SANDIA REPORT

SAND99–0647
Unlimited Release
Printed March 1999
Supersedes SAND95-2760,
Dated October 15, 1995

Software and Information Life Cycle (SILC) for the Integrated Information Services Organization

Adaptations of the Sandia Software Guidelines: Issue C March 23, 1999

Donna Eaton, David Cuyler, Scott Joyce, Mary Lynn Clark, Don Flores, Irene Thurston, and Joe Schofield

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

SAND99-0647
Unlimited Release
Printed March 1999
Supercedes SAND95-2760,
dated October 15, 1995

Software and Information Life Cycle (SILC) for the Integrated Information Services Organization

Adaptations of the Sandia Software Guidelines: Issue C March 23, 1999

Donna Eaton, David Cuyler, Scott Joyce, Mary Lynn Clark, Don Flores,
Irene Thurston, and Joe Schofield
(Integrated Information Services)

Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0660

Abstract

This document describes the processes to be used for creating corporate information systems within the scope of the Integrated Information Services (IIS) Center. Issue C describes all phases of the life cycle, with strong emphasis on a Planning phase followed by the interweaving of the Analysis and Design phases. This Issue C supersedes Issue B, which concentrated on the Analysis and Design phases within the context of the entire life cycle.

Use of this process is most beneficial for in-house developed software. It is recognized that components of a particular automated information system may be commercial off the shelf (COTS) products. If a rigorous methodology is used in conjunction with the purchased product, some recommended SILC steps might not be performed. Nevertheless, it is important to evaluate the applicability of this SILC process to any area the purchased product's methodology does not adequately address to ensure the quality

of the resulting system. For example, system implementation/deployment in Sandia's environment is likely to be a unique process that must incorporate the specific steps described in this document.

Appendix A	Full set of examples of the deliverables, excerpted from the Network Information System
Appendix B	One Page Description of each SILC Deliverable
Appendix C	IIS Design Review Process Guidelines
Appendix D	Objectives of SILC Project Planning
Appendix E	Uniform Resource Locator (URL) List
Appendix F	Future Needs in SILC D

Future issues will further develop these life cycle processes as we move toward enterprise-level management of information assets, including information meta-models and an integrated corporate information model. The phases described here, when combined in the future with a specifications repository, will provide the basis for future reusable components and improve traceability of information system specifications to enterprise business rules. Reference Appendix E, URL 1.

SILC Issue C - Update Summary

Major Updates to Existing Features . . .	Location/Summary Actions
the deliverables described for software development projects, based on best practices used on IIS projects during the past two years (since SILC Issue B was published)	• Table 1, p.6 • Appendix B, p.B-1
the example in Appendix A reflecting changes or total rewrites of the deliverables	• Appendix A: New Deliverable 0, A-2 through A-6 • Appendix A: Deliverable 3 & 4 combined into 3, A-17 through A-26 • Appendix A: New Deliverable 4, A-27 through A-39 • Appendix A: New Deliverable 7, A-48 through A-52 • Appendix A: Deliverables 8, 10, &13 combined into 8, A-53 through A-62 • Appendix A: Deliverables 10 and 13 deleted • Appendix A: Deliverable 12 updated, A-71 through A-75 • Appendix A: Deliverable 16 updated A83 through A88 • Appendix A: Deliverable 17 updated, A-89 through A-92

inclusion of current practices using "Object" techniques	• various wording changes no current "best practice" examples
Major Additions This Issue . . .	**Location**
configuration management guidelines from the work of the team producing software testing guidelines	• pp.12 through 15 • Figure 2, p.28 • Appendix A: Deliverable 4, A-27 through A-38 • Deliverable 8; A-52 through A-61 • Deliverable 12, A-70 through A-74 • Deliverable 17, A-85 through A-88 • Appendix B: B-6, B-10, or B-14, B-19
"IIS Design Review Guidelines" as an appendix and update the document to include the Utility Managers functions.	Appendix C, C-1 through C-20
One-page description of each "Deliverable" as an appendix	Appendix B, B-1 through B-20
more information and guidelines with respect to the project planning phase	• p.7 • Appendix A: Deliverable 0, A-2 through A-6 • Appendix B: B-2 • Appendix D: D-1 through D-7
up-front project planning for implementation	• pp.12 through 15 • Appendix A: Deliverable 7, A-48 through A-52 Deliverable 26, A-83 through A-88
all referenced URLs in an appendix for ease of update	Appendix E, E-1 through E-2
continuing work topics for the ongoing improvement of the SILC processes	Appendix F, F-1

Test Plan, Test Database, and Test Data A-71

External Interfaces A-77

List of Reused Elements A-81

User Help Documentation A-82

Test Results A-88

Appendix B B-1

Deliverable 0: Project Proposal/Project Plan B-2

Deliverable 1: Problem Statement B-3

Deliverable 2: Information Business Facts and Source Documents B-4

Deliverable 3: Conceptual and Logical Information Model B-5

Deliverable 4: Software Verification and Validation Plan B-6

Deliverable 5: Elementary Processes B-7

Deliverable 6: Conceptual Business Functions User Model B-8

Deliverable 7: Implementation Strategy B-9

Deliverable 8: Implementable Business Functions (Presentation Model) B-10

Deliverable 9: Database Schema B-11

Deliverable 11: Implementable Business Functions (Process or Function Model) B-13

Deliverable 12: Test Plan, Test Database, and Test Data B-14

Deliverable 14: External Interfaces B-16

Deliverable 15: List of Reused Elements B-17

Deliverable 16: User Help Documentation B-18

Deliverable 17: Test Results B-19

Appendix C	C-1
Integrated Information Systems (IIS) Design Review Process Guidelines	C-1
Conceptual Design Review (CDR) Guidelines	C-3
Detailed Design Review (DDR) Guidelines	C-6
Final Design Review (FDR) Guidelines	C-8
IIS Design Review Process for Low Risk Applications	C-13
Appendix D	D-1
Objectives of SILC Project Planning	D-1
Scope of this Plan	D-1
Where to start	D-1
Approaches to consider and incorporate for project planning	D-4
What to deliver - Plan Set Components	D-4
Appendix E	E-1
Uniform Resource Locator (URL) List	E-1
Appendix F	F-1
Future Needs in SILC D	F-1

Introduction

This document represents a plan for moving the development of integrated information systems from our current reality to a desired future state.

The perceived goals of the IIS organization are to:

- build systems that serve the needs of the enterprise,
- speed the deployment of new applications or enhancements,
- improve traceability of the design specifications at a sharable enterprise-level, and
- reduce overall costs throughout the life cycle (thus allowing for workforce redeployment or acquisition of additional work).

To further these goals, we provide a process that will shift costs and time from coding and nonproductive rework toward the analysis and design phases of the life cycle, addressed in this document. We recommend quality processes to manage the documentation and code using configuration management software and through approved validation and verification procedures.

Purpose

SILC is a rigorous, defined, and repeatable software quality improvement process designed by software engineers for use by software engineers. It is comprised of essential proven components required to produce high quality, maintainable, value-added software, not SILC deliverables, and to perform critical processes, not project oversight. These processes enable developers to reduce variation in their software products and provide the foundation for productivity, improvement, and measurement.

Scope

This document is designed to serve two overall purposes:

- Short-term:
  - To unify the baseline deliverables produced by the software development processes
  - To provide guidance for software developers who need to prepare for formal design reviews

- Long-term:
  - To move corporate software and information life cycle processes toward a unified and consistent methodology, serving diverse software development projects at Sandia

Short-Term Scope—Deliverables

This document defines processes and checklists that will move the Integrated Information Services (IIS) organization toward an enterprise-level specification management process that will clarify and document

- the rules for business operations, and
- the corresponding requirements for the automated information systems.

Each task and deliverable was assessed by the SILC team for the value added to the process. SILC deliverables for a software project result from the completion of a process. Application of the process is the goal of SILC, as opposed to the completion of a deliverable. An alternative representation of a deliverable is NOT a deviation from SILC.

Although it is recognized that components of a particular automated information system may be commercial off the shelf (COTS) products, use of this SILC process is most beneficial for in-house developed software. If a reasonable methodology is used in conjunction with the purchased product, it is understood that some recommended SILC steps may not be performed. Nevertheless, it is important to evaluate the applicability of this SILC process to any area the purchased product's methodology does not adequately address to ensure the quality of the resulting system. For example, system implementation/deployment in Sandia's environment is likely to be a unique process that must incorporate the specific steps described in this document. For more information about procurement of commercial applications, see The Make or Buy Policy and Model Appendix E, URL 2.

This document prescribes a useful set of materials that describe the life cycle of a system. A complete set of these deliverables will provide a software team with an understanding of the desired information system. This set of information, maintained in a configuration and software quality management process, will guide the customer, software developer, review team, production team, and management to comprehend system goals, background, and solutions to business problems.

Long-Term Scope—Methodology

The life cycle of a software product has been defined in many classical software engineering documents (see "References"). This document adapts the Sandia Software Guidelines to the IIS organization class of information systems. It is written not to replace those other documents, but to position Sandia to take advantage of new technologies and tools. It also supports multi-tiered architecture and future distributed computing environments.

This document, Issue C, supersedes Issue B (Reference 19), the initial effort to describe the software process. Issue A concentrated on the analysis phase. The analysis and design phases described in Issue B are not necessarily two separate processes. They are strongly coupled and may be completed concurrently. The steps require constant evaluation and iteration until the required result is achieved, i.e., the final evolutionary prototype and/or the critical deliverable set. Issue C adds detail and guidance for a strong plan set. In addition, implementation phase processes and deliverables are expanded and extended in the areas of testing, network description and software configuration management requirements.

This work is on going; it will continue to evolve in parallel with the dynamic process of applying new technologies within Integrated Information Services (IIS). We publish our efforts here as Issue C, to describe the entire software life cycle. In this effort to achieve enterprise-level management of information assets, we will move toward information metamodels, rapid prototyping, and an integrated corporate information model.

Desired Future State

Automatic generation of application systems, based on precisely defined specifications, is the ultimate goal for future information systems. Full software configuration management of all specifications at the enterprise level will support dynamic systems and ensure information timeliness, as well as data reliability.

Flexibility

The methodology and deliverable set presented here are designed as a core set to accomplish future enterprise-sharing and requirements traceability goals. The methodology and deliverables can and should be scaled for use in projects of any size. Specific IIS scaling guidelines are described in Appendix C.

A document set selection process is described in Sandia Software Guidelines Volume 2 - Documentation. In general, waiving production of any deliverables requires a management decision. For a software development project, the determination metric is based on weighted scores for size, development cost, customer, and life cycle characteristics, platform, team experience, political risk, and special requirements from the customer or from the project environment.

Before the desired future state is reached, real problems will surface with loosely interpreted application of this process. Issue C extends Issue B to stress the importance of the planning phase and to add vigor to implementation through the use of thorough testing procedures and controlled changes using formal software configuration management.

Usage

The software and information life cycle is normally described as a series of phases. Although we would prefer that these phases be simplified, we cannot imagine valid justification for skipping any activity or deliverable they prescribe. Overall, we feel that this methodology does not call for any more work than is already applied in current projects. It simply codifies and documents that work.

A compressed time-scale and improved quality result from a more consistent and defined methodology. The evolutionary prototype shortens development time and allows continuous improvement in understanding for both the customer and the software developer. Quality will improve through continuous iteration and refinement of requirements modeling and customer/subject expert evaluations.

The corporate repository for specifications allows the developer access to all models and future functions available for reuse and sharing at a level that is detailed and precise. Rapid application development should result from applying previously developed functions and framework.

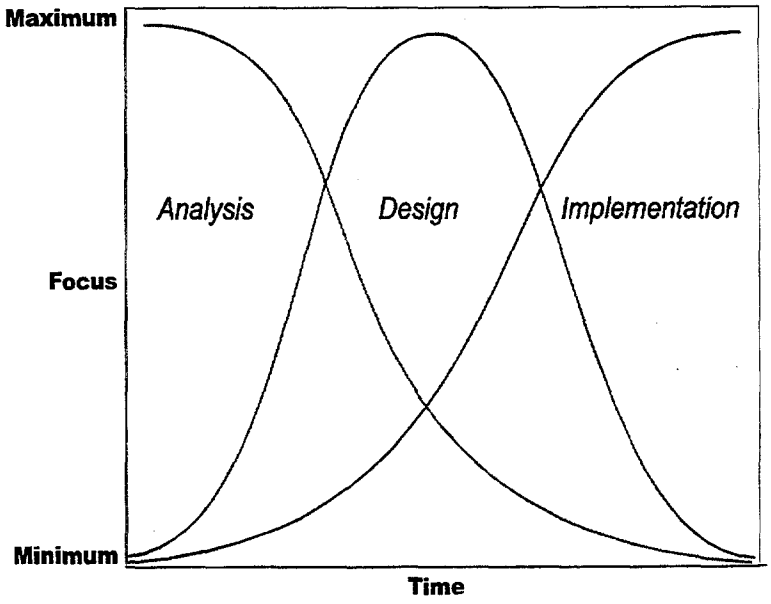
A diagram of the software process using prototyping comes closest to describing the actual steps for development of new software:

The classic phases are described as steps to apply and deliverables to produce. During the production of these deliverables, evaluation is the key to the completion of that phase or to cycle through the steps again. Both software engineers and customers need constant feedback, which is the role of evaluation. A formal evaluation may be required to end each phase, depending on the level of risk associated with the software product

and its development. The formal design review is currently used for all enterprise-wide software implemented by the IIS organization. Management of the IIS organization must choose to waive the formal design reviews (see Appendix C).

Even though we discuss phase completion, phases overlap when considered as an iterative development architecture. Figure 1 shows how the phases of a complex software development interweave.

Figure 1 Software Phase Interweaving



(Adapted from Booch and Grady)

Software Phases

Deliverables

The activities described in the software phases lead to specific software development project deliverables. The following table indicates the specification deliverable set. The numbered items are covered in more detail in the example shown as Appendix A.

Table 1 Critical Specification Deliverable Set

Primary Beneficiary	Plan Set	Analysis Set	Physical Design Set	Implementation Set	Operational Engineering Set
Customer	• Information systems plan • Project proposal (0) Project plan	(1) Problem statement (2) Information Business Facts and Source Documents (6) Conceptual Business Functions User Model	(8) Implementable Business Functions Developers Model (including flows, controls, and screen templates)	• Data layer production data-base (16) User documentation	
Software engineering team		(3) Conceptual information model, Logical information model (4) Verification & Validation Plan (5) Elementary processes (7) Implementation Strategy	(9) DB schema (11) Process Model (12) Test plan, test database, and test data (14) External and internal interfaces (15) List of reused functions	• Implementation checklist • Presentation layer application • Functional layer components	
Operations team	• System-level requirements				• operational specification
Review team	• Metric plan	• CDR "Conceptual Design Review" or "Preliminary Design Review"	• DDR "Detailed Design Review" or "Critical Design Review"	(17) Test results "Final Design Review" or "Pre-Production Review"	

Planning Phase

Enterprise-Wide Business Planning

Enterprise-wide business planning and information modeling are the preferred front-end to all information system development.

Sometimes called Information Strategic Planning (ISP), this process identifies the state of and the needs for data, process, and organizational

alignment. Strategic information planning for business systems within the corporate information model includes prioritizing, sequencing, and scoping of business areas. This planning provides the foundation for future decisions regarding the selection of business areas for development, redevelopment, and business process re-engineering. A stronger emphasis on strategic and integrated planning is advocated as the number of systems, platforms, and technologies increase.

Project Planning - A SILC First Step

Project planning, which may include on the front-end the development of a project proposal, is a necessary first step in the software and information life cycle. Project planning, while being neither new nor a science, provides a snapshot of early project expectations, resources, and scope. While this snapshot may evolve greatly over time, the project plan provides a baseline for prioritizing business opportunities, determining the allocation of resources, and coordinating the deployment of hardware, software, and communications support throughout the enterprise.

Where available, either a business plan or an Information Systems (sometimes referred to as Strategy) Plan (ISP) is the drivers for software development projects. Within an ISP are the priorities for satisfying enterprise-wide information needs. "Governance" structures are popular mechanisms for establishing these priorities in lieu of an ISP, which can be difficult to champion and develop. Whether using an ISP or a "governance" board, the project proposal or plan is eventually a product of an upper level endorsement for considering an information system project.

With the addition of Project Planning to SILC, templates, models, and reuse can be more easily applied to project documentation. References to Quality Assurance and Software configuration management plans can replace the creation of these documents by each project. Project planning templates will minimize the need to develop new project plan formats and SILC users will be able to tie project-planning tasks to other SILC deliverables in the analysis, design, and implementation phases.

With organizational change and technological advancement, project planning must include some common foundational aspects as well as implementation specific requirements. Foundational aspects are found in the ISP or its equivalent. Implementation specific requirements are represented in the Information Systems Implementation Process (Reference Appendix E, URL 3).

Analysis Phase

Analysts perform this phase by working directly with the customer, precisely defining the information facts managed by this information system (application). The preferred methodology for standard specifications is one of those mentioned in the Sandia Software Guidelines Volume 5. This methodology has been further developed to support modeling processes and events, in addition to information modeling. Because the natural language of both customer and analyst is used, communication is greatly enhanced.

The quality of a rigorous analysis phase produces the desired specifications to be managed at the enterprise level. In the future, analysts will use a version-controlled repository to manage all model constructs. This enterprise-level tool helps manage reusable information and functions for multiple models and systems. The repository will eventually support automatic generation of information systems and business rule descriptions.

Analysis Phase Steps

1. The analysts define a scope to set boundaries on the problem. In addition, they develop a problem statement, including a problem narrative, context diagrams, and other descriptive materials.

2. The analysts query the repository for existing building blocks to support the software engineering reuse of model constructs. As new objects are created in the natural language models, they must be coordinated with the data administrator. The data administrator will add these objects to the repository if they are not already defined and used in other systems. Negotiations with the data administrator will be necessary for classifying new objects and functions, i.e., to leave them as local definitions (for this application) or to promote them to enterprise-shareable status.

3. The analysts create a natural language (English sentences) model from the building blocks selected from the enterprise model repository and from newly defined constructs resulting from analyst-customer reviews. This model rigorously defines the information, its meanings, identifiers, relationships, and constraints (especially mandatory and uniqueness requirements).

4. The analysts produce a natural language process model and graphical presentation model to define precisely the processes that will act on the

information. (The model can include storyboards, known as presentation sets or prototypes.) Constraints and derivation rules are expressed textually and through concrete examples. Each analyst-customer review of the specifications should produce further refinements to the model as more areas of interest are discovered, and all business rules are brought out and explicitly documented, using a set of customer-provided examples.

5. The analysts repeats Steps 3 and 4 until the model completely specifies the problem scope and until the customer and process owner agree on its completeness. The customer or process owner and analysts accept this reviewed analysis set by formal sign-off.

6. The analysts schedule a conceptual review (see "Glossary of Terms") to verify the specifications produced in this analysis of this application or subsystem. The review may also be attended by business experts and technology experts, including the developers. Other aspects of the system may be addressed in this review, such as training and special requirements.

Future Goals

An ongoing goal in the evolution of this methodology is to continue to shift the traditional notion of testing to earlier phases in the life cycle. The examples collected during the analysis play a critical role in the validation process. At some future point, confidence in the rigor of the specification and development should reduce our reliance on classical unit, system, and acceptance testing. Reasons for shifting this emphasis include:

- The growing sophistication and reliability of software analysis, design, and testing tools will assure that generated code performs in accordance to the specification from which the code was generated. We will only need to test that the code generated from a specification of ideal behavior actually does what is intended in the real computing system, interacting with hardware the operating system components, and other applications.
- As object-oriented analyses matures analyses deliverables will be modified to better support object-oriented implementations.
- Presentation sets developed from prototyping tools serve as training and communication vehicles, establishing a foundation for eventual customer acceptance.
- By managing the design intent, traceability to business practices can be maintained for expected gains in business process improvement and business re-engineering.

Initially these documents will be manually generated, with text generation development eventually progressing to the point of automatic generation of the information system specifications and automatic generation of the business-rule documents.

Physical Design Phase

This step will use the specifications and extend them into a physical database schema and either an extended specification or an evolutionary prototype. The system software engineers should be knowledgeable developers with experience in building and using preferred technology information systems tools, as bounded by the standard architecture of the IIS systems. The software engineers should also have skills in human factors engineering as well as the classical programming training and skills.

The software engineers will specify the amount of flexibility of design format regarding the presentation sets transmitted to the development team. Aiming for consistency and user-centered design, human factors guidelines will affect the screen design and planned system behaviors. The software engineers will also consider implementation perspectives. Future presentation, functional, and data layer guidelines, along with assistance from the data administrator, will direct the choice of the components of the multi-tier model. Choice of certain vendors' tools may impose physical constraints on the implementation. The design will address those issues that were not specified in the analysis phase. The risks associated with the software product and its development affect the choice of which of the following optional documentation elements the team will produce and review:

- Decomposition. IDEF0 is a format in this example, as it is the Government standard for documenting process decomposition. Other acceptable representations include structured English state-transition diagrams, process hierarchy diagrams, and data flow diagrams.
- Detail Description and Algorithms. Data derivations, known as derived facts.

Physical Design Steps—Presentation Layer

1. The software engineers extend the presentation sets and screens, considering human factors and following standard templates and guidelines. In either hard copy form or software, this deliverable evolves into a prototype. Based on the information model that contains the information requirements, a prototype can resemble the

final product for data input as described by the elementary processes from the analysis phase.

2. The software engineers document flows and controls (which may also be in the functional layer) and include a list of edits. These edits control the occurrence of triggering events. As presentation elements, they indicate points of user input. Here the software engineers also include guidelines for choosing pick lists and checklists over user input, based on known factors such as data changeability and volume.

Physical Design Steps—Functional Layer

1. The software engineers and data administrator identify all reusable facts and functions.

2. The software engineers describe the physical design. External and internal critical interfaces include:

   - (external) control data from other sources, working data, and time-dependent triggers, and
   - (internal) data and message passing in the system.

Physical Design Steps—Data Layer

1. The software engineers plan for the production of the physical schema by proposing extensions to the logical schema such as:

   - add the surrogate (generated) keys,
   - denormalize selected tables to enhance performance and ease of use,
   - apply estimates and statistics, which indicate where additional alternate keys should be placed and indexes used,
   - add protection and controls by examining data sensitivity and adding row level protections where necessary,
   - design access paths, which may include more indexes, key placement in tables, etc., and
   - add and document additional requirements to the schema, for such functions as tracing or archiving—these should be developed from functional processes rather than information requirements at this phase.

2. Software engineers create a test database in accordance with the test plan. Future plans call for the development of a utility to automate this function.

3. The software engineers and customer formulate a conversion map for external interfaces. The map will be used to populate the database with existing legacy data or any continuing external controls and output formats to interface with other systems.

Physical Design Steps—Completion

1. The software engineers document and cross-reference all design decisions and the processes affected by them.

2. The software engineers add the physical model and general constraints (other rules and information, not contained in the model) to the repository.

3. The software engineers hold a Detailed Design Review to verify the physical design.

Implementation Phase

The implementation of a system requires skilled developers and the coordination of many interfaces to produce a new system and to put it into production. As system code is completed, the coordination of interfacing workgroups will take on even more importance to

- complete the system,
- test the system, and
- set up such components as networks, security plans, training, and production procedures.

Implementation Phase Steps

1. The database administrator provide a working test schema, a test database to the customer, and the implementers. Test data is created using the test data map provided by the software engineers and data from the customer.

2. The implementers build the presentation layer components according to the current software and hardware architecture and standards of the IIS systems. Where possible, the functional layer programs will be generated directly from the specifications.

3. The implementers build the functional layer components not already available as shared reusable code, according to the current software/hardware architecture and standards of the IIS systems.

Where possible, the code will be generated directly from the specifications.

4. The implementers perform integrated testing against the test database.

5. The test team performs system testing on the corporate test bed system.

6. The test team holds an as-built design review for the pre-production application.

7. The implementers perform the steps of the deployment process to move the system into a fully operational production system.

Implementation Steps—Deployment

This process or sub-phase includes verification, training, and production definition to move the system into a production phase where the application will be available as a corporate asset. The development team deployment coordinator should possess the skills necessary to manage enormous amounts of data and place the required modules and components in a software configuration management system. The deployment coordinator must also be able to perform multiple tasks and manage coordination of schedules of many customer and IIS groups. Guidelines for Central Services Production Support are documented in the IIS Application Support Guide (Reference Appendix E, URL 4).

1. The analysts and production specialists define the processing duplication, i.e., parallel runs for validation and verification requirements.

2. The analyst and customer provide the static data and legacy data to populate the production database.

3. The network administrator and production specialist connect the networks to access related and dependent data from other systems.

4. The analyst, customer, and training services develop the training classes and train users, as well as production support staff and help desk personnel. This team also develops user's documentation.

5. The production specialist and analysts develop operational procedures, including network support guide and company asset protection processes and plans.

6. The analysts and production specialist perform a customer acceptance review of the Final Design Review, with an official sign-off for the system.

Information Systems Implementation Checklist

Table 2 indicates the checklist as currently defined. The checklist is also available on the internal web: (Reference Appendix E, URL 5).

The checklist is a standard that incorporates required sign-offs necessary for moving new systems into production and enhancing existing ones within the Integrated Information Services Organization. The deployment process ensures continued support without dependence on the original developer. It is used by developers and the production representative to identify stakeholders who need to be contacted as products are developed. The upper portion of the checklist (above the double line) contains early life cycle considerations. Items in the lower portion of the checklist (below the double line) are coordinated before the product is released.

The Production Center Implementation Coordinator is the single point-of-contact for questions about this process. In addition, this Coordinator will serve as or provide a production representative as a full member of the project team. This representative will:

- coordinate with all production support stakeholders who are not project team members,
- ensure that production requirements are included in the project plan with reasonable milestone schedules and ensure that these milestones are met (including responsibility for writing some of the required documents), and
- write the service-level agreements documenting operational and performance requirements.

Additional information on transitioning the system to a support team can be found in the Application Support Transition Guide (Reference Appendix E, URL 6).

Table 2 Information Systems Implementation Checklist

Consult with	Customer about:	System Administration about:	Application Support about:	Technical Test Bed about:	Database Administration about:	Network Communications, Security, and Production Services about:
While performing project planning	• application requirements • performance requirements • access control	• server software requirements • server hardware requirements	• transition plan	• workstation and PC installation and execution requirements • architectural compliance including where application resides and executes • system design reviews	• data administration modeling requirements • data requirements: backup and recovery, security, conversion, volumes and growth, source/target archive/delete	• Production Services documentation requirements • Security data security classifications and security plans • Network Communications traffic tolerances and network needs • IIS Application Release Process
While performing implementation planning	• early life cycle aspects above • customer acceptance and verification testing • application roll-out communication plan	• early life cycle aspects above • backup and recovery requirements • scheduling and compatibility of system software installs and upgrades (software integration) • stress test • integrated system test	• early life cycle aspects above • change control procedures • negotiated level of support hardware and software training for supporting staff	• early life cycle aspects above • product executes on standard platforms • software distribution needs • reviewing test execution reports	• early life cycle aspects above • conformance to database standards • database performance considerations • database implementation	• early life cycle aspects above • Production Services service level agreements • Production Services Help Desk needs • Network Services network usage diagram • Production Services final sign-off of completed checklist

Operation Phase

The operation phase, traditionally called the production phase and maintenance phase, will be viewed as static, operational processes until the business rules change. This phase requires:

- quality management techniques to assure the integrity and protection of valuable corporate assets and to maintain customer delight during day-to-day business operations,
- extensive software configuration management tools, and
- intensive commitment to the maintenance of the specifications and constructs in the corporate repository.

An application in the operation phase requires the entire software life cycle process to implement new requirements. To the extent required, all procedures will be performed as outlined in the life cycle phases: Planning, Analysis, Physical Design, and Implementation.

Termination Phase

The system will be closed out after it is no longer needed.

Termination Phase Steps

1. The production specialist moves any legacy data with continuing requirements to other systems.

2. Corporate Records Management evaluates the transfer of specifications and historical information to be retained for proper disposition and thereafter saved and managed by that organization.

3. The data administrator removes from the corporate repository constructs and process specifications that are no longer required for this or other systems.

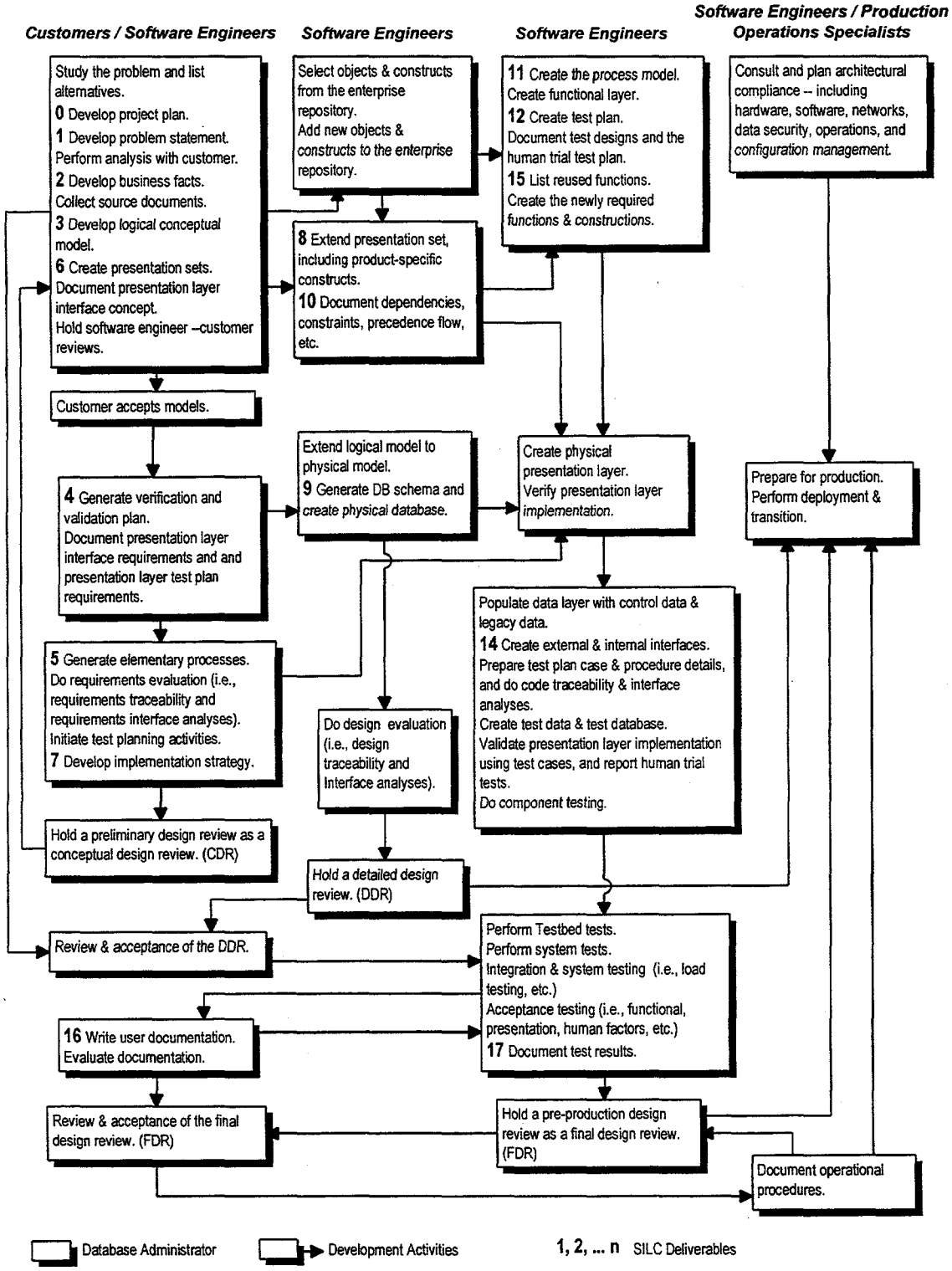
Summary

The software process we have described will be used, evaluated, and modified in a continuing process improvement cycle. As technology changes and tools and architecture become more stable, this life cycle process will be modified to incorporate current best practices.

"... Good systems are not developed by only using the correct methods. A system is developed by people who are motivated, willing, and able to use the appropriate technology," wrote Fleischmann. To manage the project, a

strong leader is vital, as well as different team structures based on the right team mix of technical and social skills and the variation in work packages for each unique software project. "Program development is done by people; there they are the most important resource," noted DeMarco (Reference 22). With the addition of the balanced mix of team members and methods and processes, plus supportive management and toolsets, the software and information life cycle may be summarized graphically as in Figure 3.

Figure 2 Roles and Responsibilities



Glossary of Terms

Term	Definition
4GL	Fourth Generation Language
CASE	Computer-Aided Software Engineering
Change Management Configuration Management	A discipline applying technical and administrative direction and surveillance to identify and document the functional and physical characteristics of a configuration item, control changes to those characteristics, record and report change processing and implementation status, and verify compliance with specified requirements
CIO	Corporate Information Officer
Conceptual Design Review	CDR; A review to verify the conceptual information model created though interactions with customers for completeness, logical integrity, and validity. Also discussed will be the traceable path from the conceptual model to targeted physical data stores, either existing or new. The conceptual model must be capable of being expressed with structured English sentences with concrete examples provided for validation. Fact-by-fact mapping to screens, reports, and databases will be examined. Mappings of facts (data attributes) to middle layer functions will also be presented. Algorithms used to derive new fact will be reviewed, and targeted implementation strategies will be discussed. Proof of the customers' acceptance of the model must exist at this time.
Conceptual schema	A formal specification of the user's information requirements; a common world model that explicitly defines the set of meaningful sentences which can be communicated via machine, or stored within the machine.
Constraint	Defines a restriction on the permitted population(s) of one or more populatable constructs in a conceptual schema.
Context Diagram	The highest level data flow diagram, depicting the system's interfaces to other systems within the enterprise and its interfaces to the outside world; defines the scope and boundary for the system. It represents the system as a single circle, hiding all information about what occurs within the system. Each external entity is depicted as a single square. These squares, and the flows into and out of them, define the context within which the system and each external entity depict flows; they are labeled with the names of objects that flow into the system from the external entity and that flow out from the system to the external entity.

Term	Definition
COTS	"Commercial Off The Shelf" software
CRUD	Create, Read, Update, Delete
CSU	Customer Service Unit
Customer	The internal or external buyer or recipient of a product.
DCE	Distributed Computing Environment
ERD	Entity Relationship Diagram; A data model that describes attributes of entities and the relationships among them.
Evolutionary prototype	A preliminary instance of a system, usually with limited interfaces and functionality, that serves as a starting point for continued development of a system until that system exists in its fully implemented form. In contrast to an exploratory prototype, an evolutionary prototype becomes an integral part of the delivered product.
Fact	A piece of information with specific meaning, communicated in a formal natural language sentence.
Final Design Review	FDR; The final opportunity to review the deliverables prior to deployment in a production status. Assure that all test bed anomalies have been resolved. All issues of support (CSU and Production Operations) should be resolved.
FIPS	Federal Information Processing Standard
Functional diagrams	Graphic depictions of functional components of a system.
ICAM	Integrated Computer-Aided Manufacturing; a definition language.
IDEF	Integrated Computer-Aided Manufacturing Definition; designated government standards, FIPS 183 and 184
IDEF0	Part of the IDEF set of standards; widely used method for graphically describing how a process works that uses functional decomposition to provide both an overall picture and a detailed view of the process.
IDEF1X	Part of the IDEF set of standards; an entity-relationship diagramming standard
IEEE	Institute of Electrical and Electronics Engineers—an organization of engineering and electronics professionals; notable for developing standards relating to software engineering

Term	Definition
IIS	Integrated Information Services
ISP	ISPDEF document
Implementation	the process of converting a software design into executable code
Methodology	A set of measurable core process definitions to be applied to software developed at Sandia
Metric	A quantitative measure to the degree to which a system, component, or process possesses a given attribute
NIAM	Natural-language Information Analysis Methodology
Project management	The processes by which projects are administered from conceptualization through maintenance and retirement
Pseudocode	A combination of programming language constructs and natural language used to express a computer program design
Quality assurance	A planned and systematic pattern of all actions necessary to provide adequate confidence that an item or product conforms to established technical requirements A set of activities designed to evaluate the process by which products are developed or manufactured
SDD	Software Design Description—A representation of software created to facilitate analysis, planning, implementation, and decision making; used for communication of software design information, and may be thought of as a blue print or model of the system
Software Management Plan	An auditable mechanism of each individual software group; describes the methods, tools, and techniques to acquire, develop and document, use, and support the software under the stewardship of that group. Development methodology documents and conduct of operations manuals are often the mechanisms used to document the Plan. The Software Quality Department distributes templates of the Plan.
Standards	Mandatory requirements employed and enforced to prescribe a disciplined and uniform approach to software development; mandatory conventions and practices.
Structure charts	Graphic depictions of the invocation control of modules which comprise a software system.
UML	Unified Modeling Language developed through the cooperative efforts of Booch, Rumbaugh, and Jacobsen. (Reference Appendix E, URL 7).
V&V	Verification and Validation

Term	Definition
Validation	Check that the software system adheres to system and software requirements.
Verification	The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.

References

1. IEEE Software Engineering Standards, Third Edition, IEEE, 1989.
2. Sandia Software Guidelines, Volume 1, Software Quality Planning, SAND85-2344, August 1987.
3. Sandia Software Guidelines, Volume 2, Documentation, SAND85-2345, in press.
4. Sandia Software Guidelines, Volume 3, Standards, Practices, and Conventions, SAND85-2346, July 1986.
5. Sandia Software Guidelines, Volume 4, Software configuration management, SAND85-2347, June 1992.
6. Sandia Software Guidelines, Volume 5, Tools, Techniques, and Methodologies, SAND85-2348, July 1989.
7. Software Development Process Methodology, Sandia, June 23, 1993.
8. Former SLP 1011, Software Management, April 1993.
9. Former SLP 2815, Preferred Processes, December 1992.
10. Computer Software Management, DOE Order 1330.1D, Office of Information Resources Management, U.S. Department of Energy, Washington, D.C., May 1992.
11. Becker, Steve D. Annotated Summary of the Information Model Design Procedure (IMDP), SAND94-0460, May 1994.
12. Nijssen, G.M. and Halpin, T.A. Conceptual Schema and Relational Database Design. Prentice Hall, 1989.
13. Myers, Glenford J. Art of Software Testing. John Wiley and Sons, 1979.
14. Software and Information Life Cycle (SILC) for the Integrated Information Services Organization; Analysis and Implementation Phase Adaptations of the Sandia Software Guidelines: Issue A. SAND95-1614, July 1995.
15. Software and Information Life Cycle (SILC) for the Integrated Information Services Organization; Analysis and Implementation Phase Adaptations of the Sandia Software Guidelines: Issue B. SAND95-2760, July 1995.
16. Booch, Grady. "Practical Software Engineering." Crosstalk, July 1995.
17. Fleischmann, Albert. Distributed Systems: Software Design and Implementation, Springer-Verlag, 1994.

18. DeMarco, T. and Lister, T. Peopleware: Productive Projects and Teams. Dorset House, 1979.

19. DOE Software Engineering Methodology (SEM), Project Plan Example, December 1995

20. Fiscal Year 1997 Enterprise Information Plan, April, 1997

21. Information Architecture Volume II, Baseline Analysis Summary, December 1996, DOE

22. The Capability Maturity Model, Software Engineering Institute, Carnegie Mellon University

Acknowledgments

Contributors—Issue A, B and C Teams and Design Review Guidelines Team

Joe Schofield	Information Technologies
Donna S. Eaton	Information Technologies
Don Flores	Implementation Processes
Irene Thurston	Information Technologies
Scott Joyce	Information Technologies
Mary Lynn Clark	Implementation Processes
David Cuyler	Data Administration
Ann Hodges	Implementation Processes
Ann Perry	Business Systems
John K. Sharp	Data Administration
Ronald Hall	Data Administration
John Hatley	Information Technologies
Shelley Eaton	Information Technologies
Gary E. Rivord	Information Technologies
Fay Tamashiro	Information Technologies
Greg Conrad	Information Technologies
Elisa Kephart	Information Technologies
Charlene Strickland	Information Technologies
Dwayne Knirk	Software Quality
Leonard Stans	Advanced Networking Integration
Al Alvarado	Program Management
Carol Jones	Administrative Systems
Mike Pendley	Administrative Systems
Duane Garrison	Administrative Systems
Jim Stromberg	Program Management
Andrea Cassidy	Test Bed System
Bill Mertens	Database Administration
T. C. Adams	Production Services
Nancy Marsh	The Production Process
Spencer Nelson	Telecommunications Ops II
David Evans	Telecommunications Ops II

Appendix A

Example of Deliverable Set to Complete the Analysis, Design, and Implementation Phases and to Use at the Design Reviews

This example from the Network Information System Project (NWIS) includes Deliverables 0 through 17, to demonstrate the deliverable set (considered the minimum critical set) for the analysis, design, and implementation phases of a software development project.

We chose this example because it is a real system, currently in production. As a teaching tool, it uses concepts familiar to all computer users at Sandia, namely User IDs and accounts on those computer systems for which access has been approved.

This Network Information System contains extensive network and security management concepts, which have been left out of our example to simplify the content to a manageable size for both the reader and the authors. As shown in the Problem Statement and Context Diagram, the concepts demonstrated are all aspects of the customer and logins/accounts, plus the document, "Reauthorization Forms."

Only customers who are Sandia employees are considered here. The entire NWIS specifications will be available in the corporate repository. The browsable functions, are available to any Sandian through the internal Web.

Project Plan

Deliverable 0

A Planning Example

The following project plan/proposal provides a simple yet complete sample of the planning components described in the SILC Planning phase. This is not the actual plan for the NWIS project, since this format was not in use at that time. This is, however, intended to serve as a clear example of how a project plan might look.

Project Plan: Network Information System

Introduction. This project plan describes the activities, deliverables, schedule, and resources to complete the Network Information System. Feedback and comments to this plan should be sent to Linda Garcia who is both the author and contact. The Network Information System project is being sponsored by Mike Eaton, CIO, through Dorothy Rarick of Database and Application Support. Stakeholders include software engineers and their managers within Integrated Information Systems (IIS), and Sandians performing the functions of Password Administration, LAN Points of Contact, Customer Service Units, Central Service Administrators, Network Software engineers, Network Design Team, Computer Security Administrators, and Computer Security Representatives. All future users and their clients are beneficiaries of this project. (Version 1 was issued on May 15th, 1995.)

Linkages to goals, objectives, critical success factors, standards and other planning instruments. The NWIS project supports the following related initiatives:

a) Corporate goal IS-A95.3 - Establish and maintain reliable corporate networks to support centralized data services, including access to and update of corporate databases, general use of e-mail, and access to Internet and Intranet resources.

b) Corporate goal IS-C95.6 - Pursuant to security regulations and proper network management, maintain complete and up-to-date records of all network connections to Sandia corporate networks, and all users authorized to access centrally-managed information systems.

c) Corporate goal IS-A95.10 - Provide, to the extent possible, a uniform, interchangeable "Sandia desktop" workstation configuration, in order to maximize the ease with which Sandians may exchange computer hardware and computerized information, and to minimize the cost of maintaining and upgrading Sandia's computing capabilities.

d) SLP 1011 for the protection and utilization of Sandia-acquired or developed software.

Project Metrics. The business of the Network Information System is the management of Sandia networks and electronic desktops. Metrics include a variety of documents that use NWIS data to indicate the health of Sandia networks, desktops, and central service accounts. Examples might be "the number of Central Service Units accessing the Network Information System", "the number of secretaries who do not have the Electronic Timecard package deployed", "the number of non-Sandians with accounts on the Image Management System", "the number of network connections that have not been upgraded to TCP/IP", etc. To the extent that these documents indicate improvements in Sandia's network picture over time, they will reflect the successful use of NWIS as a management tool. Short term project-related metrics are milestones met on schedule over total milestones, and resources expended over resources allocated. These two key metrics will be maintained and accessed from the project's "homepage".

Initial Problem Statement and Scope. The Network Information System is to replace and improve upon the existing "Network Information System" system, and is to provide the following functions for Sandia:

- **Manage Networks** - Keep track of Sandia's overall network configuration, including all of the logical LANs defined under the four major protocols "TCP/IP", "appletalk", "DECNET", and "IPX", the placement of those logical LANs on physical LANs and networks, and all machine connections to those logical and physical LANs;
- **Manage Central Service Accounts** - Keep track of customers authorized to access Sandia networks or any centrally managed services on those networks. In this sense a "customer" is anyone who has a legitimate need to access Sandia networks, whether they are Sandians or not, and whether they are U.S. citizens or not; and
- **Manage Security Plans** - Provide a means to create and manage machine security plans. Also, create portions of the major security plans that cover physical LANs, networks, and multi-user machines.

Reengineering Opportunities: The following re-engineering goals are being incorporated into the NWIS project:

- **Make NWIS the primary information system for Customer Service Units (CSUs) -** Up to now, CSUs have had to work out of several databases, and many have not even used the Network Information System. This project will combine all of the CSU data needs into NWIS, providing for the expanded data that CSUs now need about hardware configurations and software deployment, and providing a focal point for efforts to automate the collection of desktop information over the network.
- **Drive the Domain Name Service (DNS) from NWIS -** Up to now the Domain Name Service has been maintained manually, and inconsistently, by a large number of people, and is now known to contain many out-of-date entries. Use of NWIS to drive DNS will keep the data current and consistent, and will combine the DNS function with the normal maintenance of desktop information.
- **Simplification of Security plans -** As part of this effort, Computer Security will simplify machine security plans for non-classified machines to little more than the ability to associate what DOE calls "sensitive applications" (such as "performance appraisal") with such machines. Computer Security will also no longer do major security plans for LANs, networks, or machines on which there is no classified. These two policy changes will reduce the volume of one-page security plans from about 2900 to 900, and the volume of major security plans from about 150 to 50.

Current State vs. Future State: The current state of the Network Information System is documented in the "Network Information System User Guide" (May, 1985, no version), and as a variety of comment sections embedded in existing source code. The future state of NWIS is being developed through numerous meetings with the various user groups and subject matter experts, and through consultation with the current Network Information System gurus David Duggan and Debby Potter. The future state of NWIS is being captured in the standard SILC documentation set.

From a user point of view, we are essentially going to transform an aging, difficult-to-use, hard-to-modify network information system full of incomplete and incorrect data into a modern, GUI-driven easy-to-use, evolvable network information system with a variety of much-needed new features, and the ability to ensure the quality of its own data through a large number of controls and edits built into the software. We are attempting to transform software that many users view as an extra and burdensome chore

into a system that people will welcome as a useful, "one-stop shopping" network information tool.

Business Case: Development of NWIS will lead to several direct economic benefits including:

- elimination of many redundant network information sources (such as the "CSU Database") that have sprung up to compensate for what is missing from the current system,
- lower-cost maintenance of the standard "Sandia desktop" by CSUs,
- lower-cost response to network and desktop trouble calls, and
- improved guidance for Sandia network infrastructure development efforts.

Budget Cycles: The NWIS project has been funded through FY 1995. Decisions relating to additional funding desired for FY 1996, or the need to "carryover" funding into 1996, must be made by June 1.

High-level deliverables and expected outcomes: Completion of NWIS will occur as three major deliverables. Basic functionality - sufficient to replace the existing "Network Information System" system - is scheduled for October of 1995. The suite of enhancements intended to improve management of central service accounts is scheduled to be implemented in February of 1996. The suite of enhancements intended to improve support for CSU services is scheduled to be implemented in July of 1996.

An expected outcome of the NWIS is improved auditing and documenting of Sandia networks in response to DOE's need for effective oversight. Other advantages are noted in the Business Case.

Table A.0.1 Resources, Roles, & Responsibilities. Summarized as follows:

Resource	Role	FTE	Responsibility
Linda Garcia	Project lead	.75	Develop plan, administer project and meetings, communicate project needs and status, identify and retain resources
Scott Joyce	Information software engineer	1.0	Prepare the logical specification of NWIS information structures and user functions. Evaluate ongoing enhancement requests.
Ron Hall	Software developer	1.0	Write "C" code and XVT code to implement NWIS online screens and batch programs.
Mike Hagengruber	Database administrator	.25	Provide database support for the NWIS Sybase database implementation, including maintaining separate production and test databases, and doing regular backups.
Debbie Potter	Test phase coordinator	.5	Conduct detailed testing of pre-production NWIS software. Consult on requested NWIS enhancements.

Specific and Unique Requirements: The NWIS project plan has interfaces to other planning documents such as the Enterprise Information Plan, the project Software configuration management Plan, the project Quality Assurance Plan, the Computer Security Plan, and the corporate Strategic Plan. Specifically these encompass scheduling priorities, available technologies and infrastructures, funding, and people resources.

No unique legal, networking or security requirements are known to exist.

References:

- Network Information System User Guide, May, 1985, no version
- Example, Existing CSU Metrical Documenting, March, 1995
- Specification for the Network Information System, version 3.1, May, 1985
- Password Administration Enhancements to the Network Information System, version 1.3, May 1985
- CSU Enhancements to the Network Information System, version 1.3, May 1985
- Fiscal Year 1997 Enterprise Information Plan, April, 1997
- Information Architecture Volume II, Baseline Analysis Summary, December 1996, DOE

Problem Statement

Deliverable 1

The problem statement contains a problem narrative, a textual deliverable that states the business problem in concise terms. A context diagram shows the scope of the system and which objects flow between the system and an external system interface. A high-level function diagram shows the decomposition of the system into major sub-processes. Large problems should be decomposed into subsystems that can be defined in two to three pages each.

Problem Statement—Purpose

The intent of a problem statement is to establish an initial and mutually acceptable understanding between the software engineer and the customer(s) as to the nature and scope of the system. It establishes a boundary (fuzzy, to be sure) to the capabilities of the system. It gives the software engineer a starting point for determining the major components of the system. It also launches the customer/software engineer modeling sessions to clarify and verify all information and processes that the customer will require and agree to own.

The problem statement has value beyond the development period. It is a project resource that can and should be reused in socializing and training for application of the system; it is a good introduction for user manuals. Versions of this statement should be maintained to keep it current with the system "as built."

Content Guide

The problem statement should include:

- a high-level, simplified description of the major functions the system is to perform, its boundaries, and its interfaces;
- identification of users and operators, including their departmental (individual) business or operations; and
- description of phasing if intended (large or complex systems may be developed in several phases).

Problem Narrative—Version 1

Deliverable 1 - NWIS Example

The Network Information System is the framework for administering an enterprise-wide information system consisting of distributed computers and groups and multiple levels of networks. It covers these major functions:

- *Accounts*—Keeps track of "customers" authorized to access Sandia networks or any centrally managed services on those networks. In this sense a "customer" is anyone who has a legitimate need to access Sandia networks, whether they are Sandians or not, and whether they are U.S. citizens or not.
- *Networks*—Keeps track of Sandia's overall network configuration, including all of the logical LANs defined under the four major protocols "TCP/IP," "Appletalk," "DECNET," and "IPX," the placement of those logical LANs on physical LANs and networks, and all machine connections to those logical and physical LANs.
- *Security Plans*—Provides a means to create and manage machine security plans. Also, manages and create portions of the major security plans that cover physical LANs, networks, and multi-user machines.

These major functions, and the user groups that perform them, are described in more detail in Table 1. The functions describe the complete system, to be developed in two phases.

NOTE: The shaded functions represent the focus of this example, all pertaining to the accounts function.

Table A.1.1 Network Database Functions by User Group

User Group	Function
Password Administration	Registration of customers
	Generation of Login Names and Unix user Ids
	Registration of corporate-level central services
	Setting up customers and entity accounts on central services
	Expiration and reauthorization of accounts

Table A.1.1 Network Database Functions by User Group

User Group	Function
	Placement of customers in NWIS user groups
	Registration of entities
	Setting up customer membership in named Unix groups
	Registration of physical LANs and logical LANs
	Administration of control information concerning user groups, countries, sites, e-mail types, and classification ranks
NWIS Administrators (LAN Points of Contract)	Management of LAN connection information
	Management of basic machine information
	Registration of LAN-level central services
	Setting up customers and entity accounts on LAN central services
	Verification of LAN points of contact
Customer Service Units	Maintenance of desktop configuration information
	Management of LAN connection information
	Management of basic machine information
	Generation of CSU charges
Central Service Administrators	Maintenance of central service accounts
	Administration of control information concerning Internet domains
	Feeding Sandia's Domain name Servers
	Registration of Cryptographic Public Keys
Network Analysis	General network troubleshooting
Network Design Team	Management of LAN connection information
	Management of basic machine information
	Network troubleshooting
	ISN control Files
Computer Security Representatives	Preparation of machine security plans
	Maintenance of LAN connection information
	Maintenance of basic machine information
Physical Security Administrators	Registration of Vaults
E-Mail Support	Registration of E-Mail Post Offices
	Generation of Login Names for the LIS System

Table A.1.1 Network Database Functions by User Group

User Group	Function
Network Database Administrators	General troubleshooting Administration of control information concerning networks, security environments, network interface types, tech control centers, protocols, TCP/IP network implementations, vendors, machine makes and types, operating systems, and network functions

The following functions have been chosen for detailed treatment to demonstrate the software life cycle:

Password Administration

Registration of Customers

People who are in the Human Resources file automatically have much of their data reflected into the Network Information System, but certain data, such as country of citizenship, Sandia sponsor, and network access expiration date must be entered by hand even if the customer is in Human Resources. For people who are not in the Human Resources file, including visitors" from throughout the world, Password Administration registers all data. There are currently 9,256 Sandians (8,527 regular and 729 non-regular, all of which are in Human Resources), 3,083 contractors (of which 2,214 are in Human Resources), and 487 visitors (none of which are in Human Resources), listed as active in the Network Information System.

Generation of Login Names and Unix User IDs

The "Login Name" (such as "meaton" or "mjmurph") is now in use throughout Sandia to identify users on computerized systems, and is also a standard part of everyone's e-mail address. For accounts on Unix systems, Login Names are paired with numeric Unix User IDs, with different aspects of the Unix account controlled by one or the other. The Network Information System contains an algorithm to generate unique Login Names and Unix User IDs (up to three pairs for each person or entity—one pair each for the unclassified, confidential, and secret classification levels). Generation of unclassified Login Names and Unix User IDs is done automatically for people in the Human Resources file when they become active employees. All other such generation is hand-initiated by Password Administration. There are currently 14,328 active pairs of Login Names and Unix User IDs in the Network Information System.

Registration of corporate-level central services

"Corporate-level central services" are major computing resources and services to which access can be restricted by password, and for which it has been decided that Password Administration will formally control access.

When each service is registered, the highest classification rank allowed is recorded, whether or not it is available for access by named Unix groups, whether or not it requires file disposition for terminated accounts, whether

or not it requires a case for account charges, and where it is located as far as what server machines, physical LANs, and/or networks on which it is available.

Setting up customer and entity accounts on central services

Password administration uses the Network Information System to set up accounts on central services they manage before they give out passwords. (Passwords are generated outside the Network Information System.)

The Network Information System applies numerous rules to make sure that account data is complete and that a given customer's access to a service does not violate security considerations before an account may be activated.

If the service requires a case number for account charges, it is entered here and checked against the list of valid cases in HRIS.

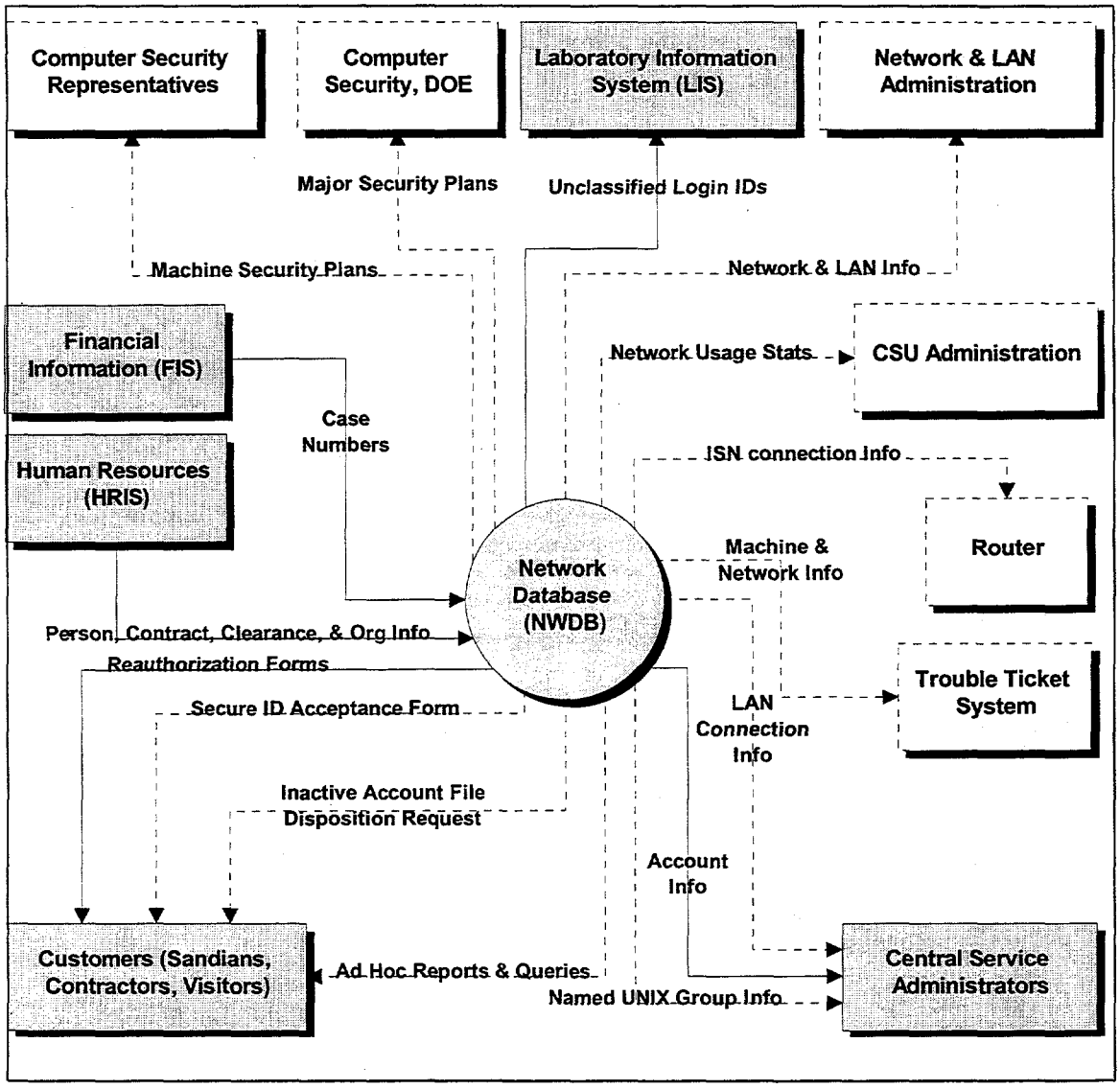
There are currently 9,382 active corporate-level accounts.

Expiration and reauthorization of accounts

The Network Information System determines the expiration date of accounts held by any customer or entity based on a number of factors such as the classification level of the accounts, a contractor's expiration date, a visitor's access expiration date, and so on. When it is time for a set of accounts belonging to a customer or entity to expire, the system will generate a Reauthorization Form, which is automatically delivered via e-mail to the account holder or entity owner if an entity. The customer can then print it out, get it signed, and return it to set the account expiration dates ahead.

Context Diagram

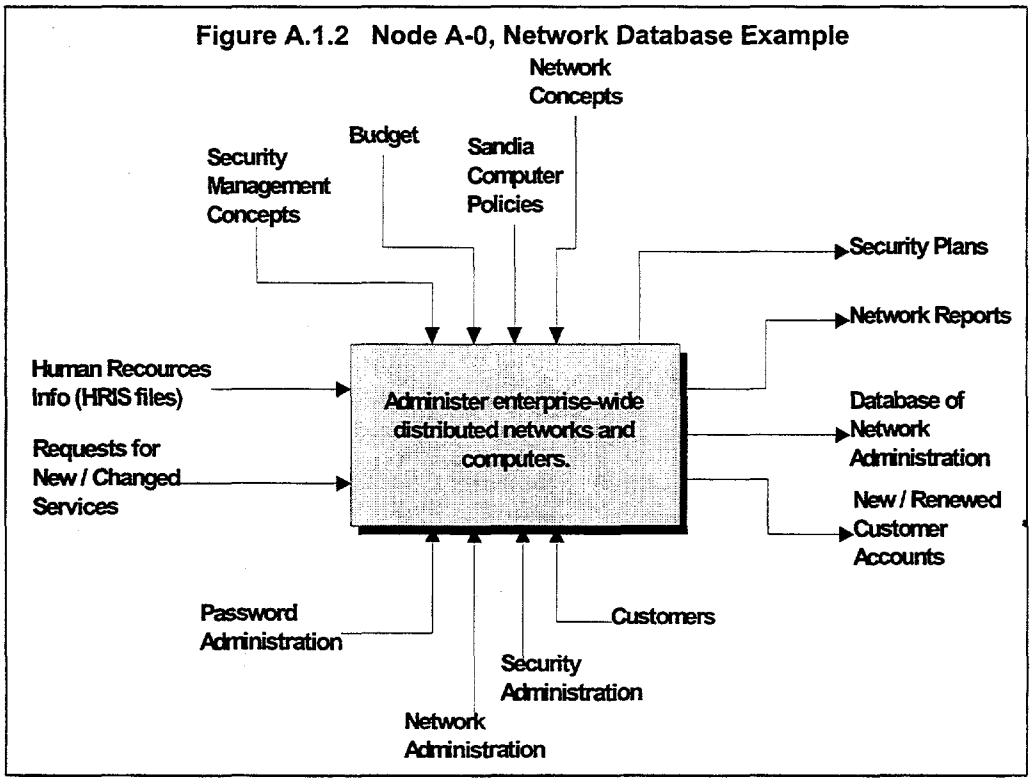
Figure A.1.1 Context Diagram—Network Database Example



High-Level Function Diagram

Deliverable 1

This example shows the major functions of the NWIS. This depicts the central part of Figure A.1.1



Reading from the top, clockwise:

- The arrows entering the top of the box containing the process description show "controls" on this process.
- The arrows exiting to the left show the outputs
- The bottom arrows show mechanisms, i.e., answers to who or what will act on this process
- Finally the arrows entering from the right show those inputs that will be affected by the processes and transformed to an output

Information Business Facts

Deliverable 2

Formal natural language sentences with examples specify aspects of each object to be maintained in the information system. These sentences, together with initial storyboards or cartoons and flow diagrams, are also used to document the business rules and verify them with the customer. The initial storyboards or cartoons and flow diagrams with process and event information (not shown here) are refined throughout the analysis and design processes into Deliverable 6, presentation sets with event and process lists (see "Presentation Sets"). General constraints and derivation rules are included in textual form or as a reference to the process list.

The formal natural language sentences with examples require an object listing with definitive information about each object. The listing includes the object's label, along with examples.

NOTE: We prefer that a minimum of three examples be supplied. Where the Network Information System team had only two examples, this generated list simply displays.

Table A.2.1 Object and Fact Type Listing, With Examples

Object Type	Example Sentence
Account	Central_Service <LIS> has access authorized for Login <<a value>> Central_Service <LIS> has access authorized for Login <jschavez> Central_Service <SOMNET> has access authorized for Login <dbblack>
Agent	agent_ID <399665432 Customer> identifies a(n) Agent agent_ID <HR update> identifies a(n) Agent agent_ID <New IRN/NWIS Accts> identifies a(n) Agent
Case	Case_Nbr <3321400000> identifies a(n) Case Case_Nbr <8999111000> identifies a(n) Case Case_Nbr <<a value>> identifies a(n) Case
Central_Service	Short_Name <<a value>> identifies a(n) Central_Service Short_Name <LIS> identifies a(n) Central_Service Short_Name <SOMNET> identifies a(n) Central_Service
Clearance	Clearance_type <L> identifies a(n) Clearance Clearance_type <Q> identifies a(n) Clearance Clearance_type <none> identifies a(n) Clearance

Table A.2.1 Object and Fact Type Listing, With Examples (Continue)

Object Type	Example Sentence
Customer	Customer_ID <286449007> identifies a(n) Customer
	Customer_ID <399886113> identifies a(n) Customer
	Customer_ID <576447650> identifies a(n) Customer
Entity	Entity_ID <000031000> identifies a(n) Entity
	Entity_ID <000032007> identifies a(n) Entity
	Entity_ID <<a value>> identifies a(n) Entity
FT225	Account <LIS,dbblack> had status last changed by Agent_Type <<a value>>
FT225	Account <LIS,mfjohnson> had status last changed by Agent_Type <process>
FT225	Account <SOMNET,jschavez> had status last changed by Agent_Type <<a value>>
FT226	Account <LIS,dbblack> charges by default to Case <<a value>>
FT226	Account <LIS,mfjohnson> charges by default to Case <process>
FT226	Account <SOMNET,jschavez> charges by default to Case <<a value>>
FT227	Account <LIS,dbblack> had its status last changed by Agent <<a value>>
FT227	Account <LIS,mfjohnson> had its status last changed by Agent <New IRN/NWIS Accts>
FT227	Account <SOMNET,jschavez> had its status last changed by Agent <399665432>
FT228	Account <LIS,dbblack> had its status last changed by Date_Time <<a value>>
FT228	Account <LIS,mfjohnson> had its status last changed by Date_Time <198812110545>
FT228	Account <SOMNET,jschavez> had its status last changed by Date_Time <<a value>>
FT229	Account <LIS,dbblack> is accepted on Date <<a value>>
FT229	Account <LIS,mfjohnson> is accepted on Date <19941521>
FT229	Account <SOMNET,jschavez> is accepted on Date <19931203>
FT230	Account <LIS,dbblack> is marked with Status <<a value>>
FT230	Account <LIS,mfjohnson> is marked with Status <inactive>
FT230	Account <SOMNET,jschavez> is marked with Status <<a value>>
FT231	Customer <286449007> had status last changed by Agent <<a value>>
FT231	Customer <399886113> had status last changed by Agent <New IRN/NWIS Accts>
FT231	Customer <286449007> had status last changed by Agent <399665432 Customer>
FT232	Customer <576447650> had status last changed by Agent_Type <<a value>>
FT232	Customer <399886113> had status last changed by Agent_Type <process>
FT232	Customer <576447650> had status last changed by Agent_Type <<a value>>

Conceptual and Logical Information Model

Deliverable 3

The primary components of Deliverable 3 are the constrained sentences in textual form and in a natural language model diagram. These communicate analysis results concisely to the software engineer.

"Information systems are systems which support the effective communication of meaningful information between human beings by design. If these information systems are to be truly effective, then the principles governing their construction and use must be derived from the principles of human communication and not from the principles of computerized data processing. Our most fundamental axiom is:

> Information systems are systems, which support the communication of *linguistic* sentences (called facts) between humans.

"... in order to restrict ourselves to the effective communication of meaningful information via these information systems we must define a formal representation of the common world model to be shared by the computer and the human. This common world model is called a *conceptual schema*. For the human user, the conceptual schema explicitly defines the set of meaningful sentences which can be communicated via the machine (or, for that matter, can be stored within the machine). For the computer, the conceptual schema explicitly defines the permitted states of the database contents and the permitted transitions between those states."

A logical model may be expressed in a variety of notations, including traditional Entity-Attribute-Relationship diagrams (as in IDEF1X), and object-oriented Class diagrams (as in UML).

"Logical models generally conform to relational theory. Thus a logical model contains only fully normalized entities. Some of these may represent logical domains rather than potential physical tables. For a logical data model to be normalized, it must include the full population of attributes to be implemented and those attributes must be defined in terms of their domains or logical data types (e.g., character, number, date, picture, etc.). A logical data model requires a complete scheme of identifiers or candidate keys for unique identification of each occurrence in every entity. Since there are choices of identifiers for many entities, the logical model indicates the current selection of identity. Propagation of identifiers as foreign keys

may be explicit or implied." (Duncan Dwelle, reference Appendix E, URL 8.)

Sentences and their constraints in textual form (facts) follow.

Constrained Sentences

Deliverable 3

Table A.3.1 Objects—NWIS Example

Object Type	Description
1. Account	Nested: Central_Service has access authorized for Login
2. Agent	Object is identified by label agent_ID+
agent_ID+	AgentID is made up of either process name or customer ID plus agent type.
	Examples:[HR update;New IRN/NWIS Accts;399665432 Customer]
3. Agent_Type	values { c, p }
	Examples:[costomer;process]
4. Building_Number	Examples:[892;T-56;859]
5. Case	Object is identified by label Case_Nbr-
Case_Nbr-	Examples:[8999111000;3321400000]
6. Case_Status	Examples:[active;inactive]
7. Central_Service	Object is identified by label Short_Name-
Short_Name-	Examples:[LIS;SOMNET]
8. Classification_Level	values { u, c, s }
	Examples:[unclassified;classified;secret]
9. Clearance	Object is identified by label Clearance_type-
Clearance_type-	values { Q, L, none }
	Examples:[Q;L;none]
10. Customer	Object is identified by label Customer_ID+
Customer_ID+	Examples:[399886113;576447650;2864449007]
11. Customer_ID+	Examples:[399886113;576447650;2864449007]
12. Customer_Lookup_Name	Examples:[Johnson, Marie F.;Chavez, Jose S.;Black, Dan B.]
13. Customer_Mail_Name	Examples:[Marie F. Johnson;Jose S. Chavez;Dan B. Black]
14. Date	Examples:[19950213;19941521;19931203]
15. Date_Time	Examples:[199302211730;198812110545]

Table A.3.1 Objects—NWIS Example (Continue)

Object Type	Description
16. Email_Address	Examples:[mfjohns@snl2800.sandia.gov;jschave@E-mail_print.cs.sandia.gov]
17. Email_Post_Office_Name	Examples:[E-mail_print;SOMNET]
18. Flag	values { Y, N }
	Examples:[yes,no]
19. Login	Object is identified by label Login_Name-
Login_Name-	Examples:[mfjohnson;jschavez;dbblack]
20. Login_Holder	Object is identified by label Login_Holder_ID+
Login_Holder_ID+	Examples:[399558776;52049999;585334560]
21. Login_Holder_Type	values { customer, entity }
	Examples:[customer;entity]
22. Mail_Stop_Number	Examples:[0661;0330;0933]
23. NWIS_User_Group	Object is identified by label User_Group_Title-
User_Group_Title-	Examples:[router;NWIS Administrator;Password Administrator;Computer Security Administrator]
24. Organization	Object is identified by label Org_Nbr-
Org_Nbr-	Examples:[13211;02811;04000]
25. Phone_Number	Examples:[5058443333;5102945555;5058452222]
26. Reference_Record	Object is identified by label Ref_Rcd_Key-
Ref_Rcd_Key-	Examples:[30035;30036;30037]
27. Room_Number	Examples:[M100;268;L400]
28. SucurID_Card_Serial_Nbr	Examples:[666666;878878]
29. Site	Object is identified by label Site_code-
Site_code-	Examples:[SA,SC]
30. Site_Desc	Examples:[Sandia Labs, NM; Sandia Labs, CA]
31. Status	Object is identified by label Status_code-
Status_code-	values { a, I }
	Examples:[active;inactive]
32. Unix_User_ID	Examples:[12001;18003;30035]

Table A.3.2 Facts—NWIS Example

Fact Type	Mandatory Object	Unique Object
1. Account had status last changed by Agent_Type	Account	Account
2. Account charges by default to Case	–	Account
3. Account had its status last change by Agent	Account	Account
4. Account had its status last changed on Date_Time	Account	Account
5. Account is accepted on Date	–	Account
6. Account is marked with Status	Account	Account
7. Customer had status last changed by Agent	Customer	Customer
8. Customer changed by Agent_Type	Customer	Customer
9. Case is statused by Case_Status	Case	Case
10. Central_Service has access authorized for Login	–	Central_Service+ Login
11. Customer can be reached at Phone_Number	–	Customer
12. Customer holds Clearance	Customer	Customer
13. Customer is identified by Customer_Lookup_Name	Customer	Customer
14. Customer is identified by Customer_Mail_Name	Customer	Customer
15. Account is originally set up by Customer	–	Account
16. Customer receives electronic mail through Email_Post_Office_Name	–	Customer
17. Customer receives faxes at Phone_Number	–	Customer
18. Customer receives hard-copy mail at Mail_Stop_Number	–	Customer
19. Customer works at Site	–	Customer
20. Customer works for Organization	–	Customer
21. Customer works in Building_Number	–	Customer
22. Customer works in Room_Number	–	Customer
23. Reference_Record has as last used for a customer a Customer_ID	Reference_ Record	Reference_ Record
24. Account has its card expire on Date	–	Account
25. Customer had status last changed on Date_Time	Customer	Customer
26. Account has file disposition made Flag	–	Account
27. Customer holds a login as flagged by Flag	Customer	Customer
28. Entity flagged as holding a login by Flag	Entity	Entity
29. Entity flagged as holding a login by Flag	Customer	Customer

Table A.3.2 Facts—NWIS Example (Continue)

Fact Type	Mandatory Object	Unique Object
30. Login changed by Agent_Type	Login	Login
32. Login had its status last changed on Date_Time	Login	Login
33. Login has its accounts expire on Date	--	Login
34. Login holds Login_Holder	Login	Login
35. Login is given Unix_User_ID	Login --	Login Unix_User_ID
36. Login is held in level Classification_Level	Login	Login
37. Login is marked with Status	Login	Login
38. Login is typed by Login_Holder_Type	Login	Login
39. NWIS_User_Group can be contacted in ca through Mail_Stop_Number	--	NWIS_User_ Group
40. NWIS_User_Group can be contacted in nm through Mail_Stop_Number	--	NWIS_User_ Group
41. NWIS_User_Group contains Customer	--	NWIS_User_ Group+ Customer
42. NWIS_User_Group is ca contact for Email_Address	--	NWIS_User_ Group
43. NWIS_User_Group is nm contact for Email_Address	--	NWIS_User_ Group
44. NWIS_User_Group can be contacted in ca through Organization	--	NWIS_User_ Group
45. NWIS_User_Group can be contacted in nm through Organization	--	NWIS_User_ Group
46. NWIS_User_Group can be contacted in ca through Phone_Number	--	NWIS_User_ Group
47. NWIS_User_Group can be faxed to in ca through Phone_Number	--	NWIS_User_ Group
48. NWIS_User_Group can be contacted in nm through Phone_Number	--	NWIS_User_ Group
49. NWIS_User_Group can be faxed to in nm through Phone_Number	--	NWIS_User_ Group
50. SecurID_Card_Serial_Nbr is authorized by Account	--	SecurID_Card_ Serial_Nbr
51. Site is described by Site_Desc	Site	Site
52. Customer is marked with Status	Customer	Customer
53. Reference_Record has as last used for a customer a Unix_User_ID	Reference_ Record	Reference_ Record
54. Reference_Record has as last used for a entity a Unix_User_ID	Reference_ Record	Reference_ Record

Natural Language Model Diagram

Deliverable 3

We show two pages of Natural-language Information Analysis Methodology (NIAM) models, which is one form of a conceptual model in diagram form. This diagram provides the software engineer with another view that demonstrates the data relationships and data roles.

Notes: * denotes a "derived fact."

The **FTxx** numbers shown in the diagram reference those defined in the formal natural language sentences in Deliverable 2.

Figure A.3.1 NIAM Model—Customers

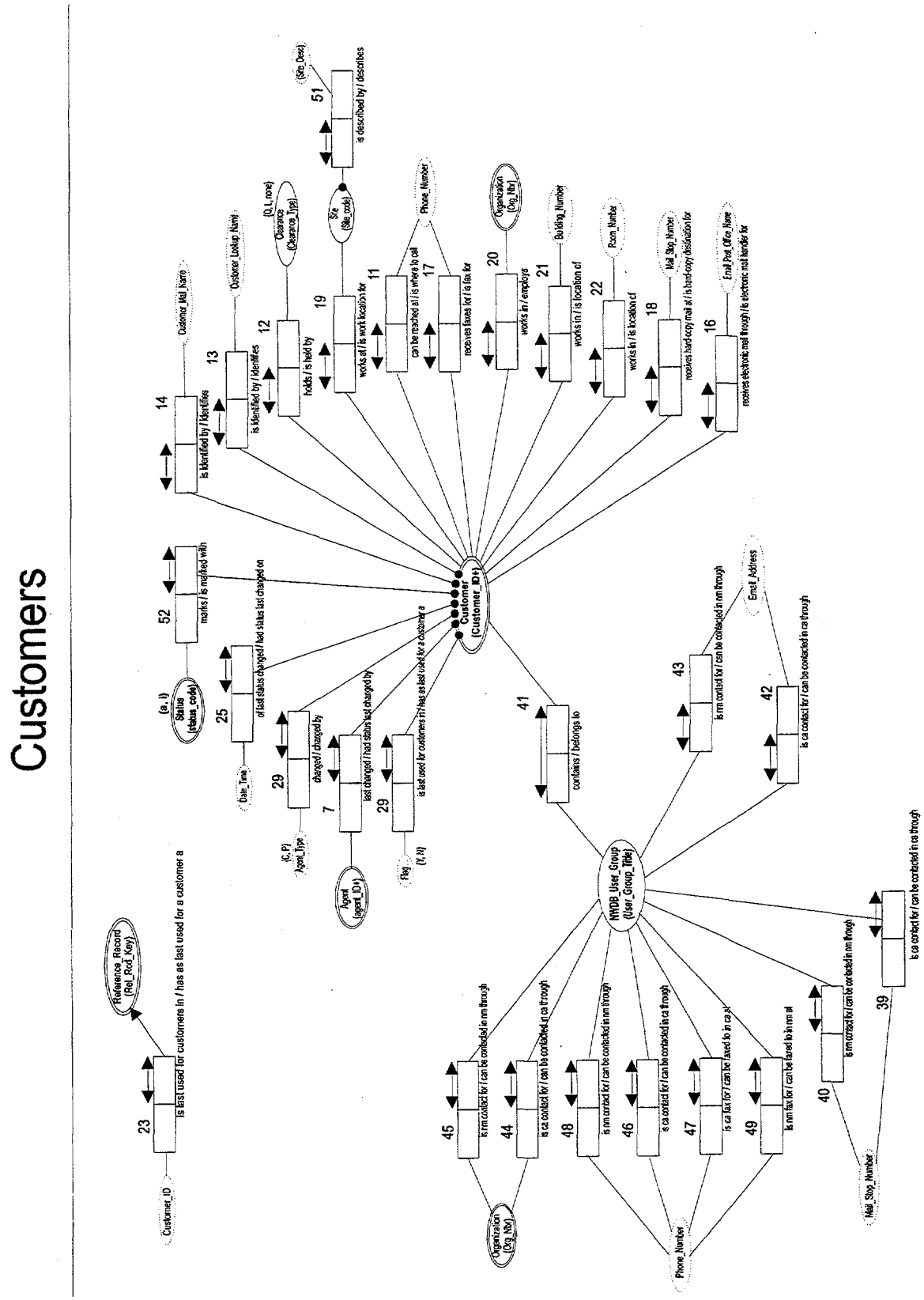
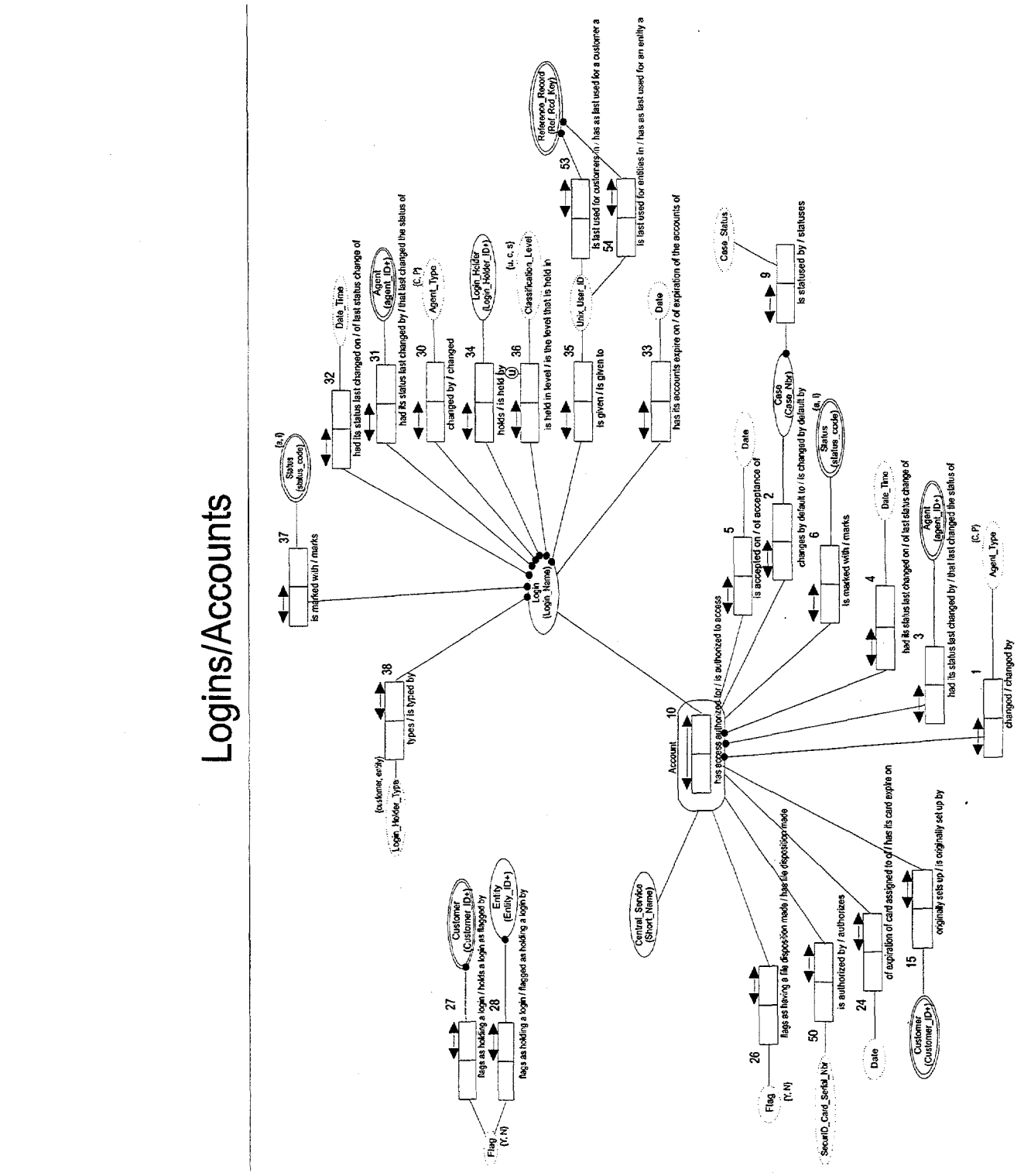


Figure A.3.2 NIAM Model—Logins/Accounts



Logical Information Model—IDEF1X Diagram

Deliverable 3

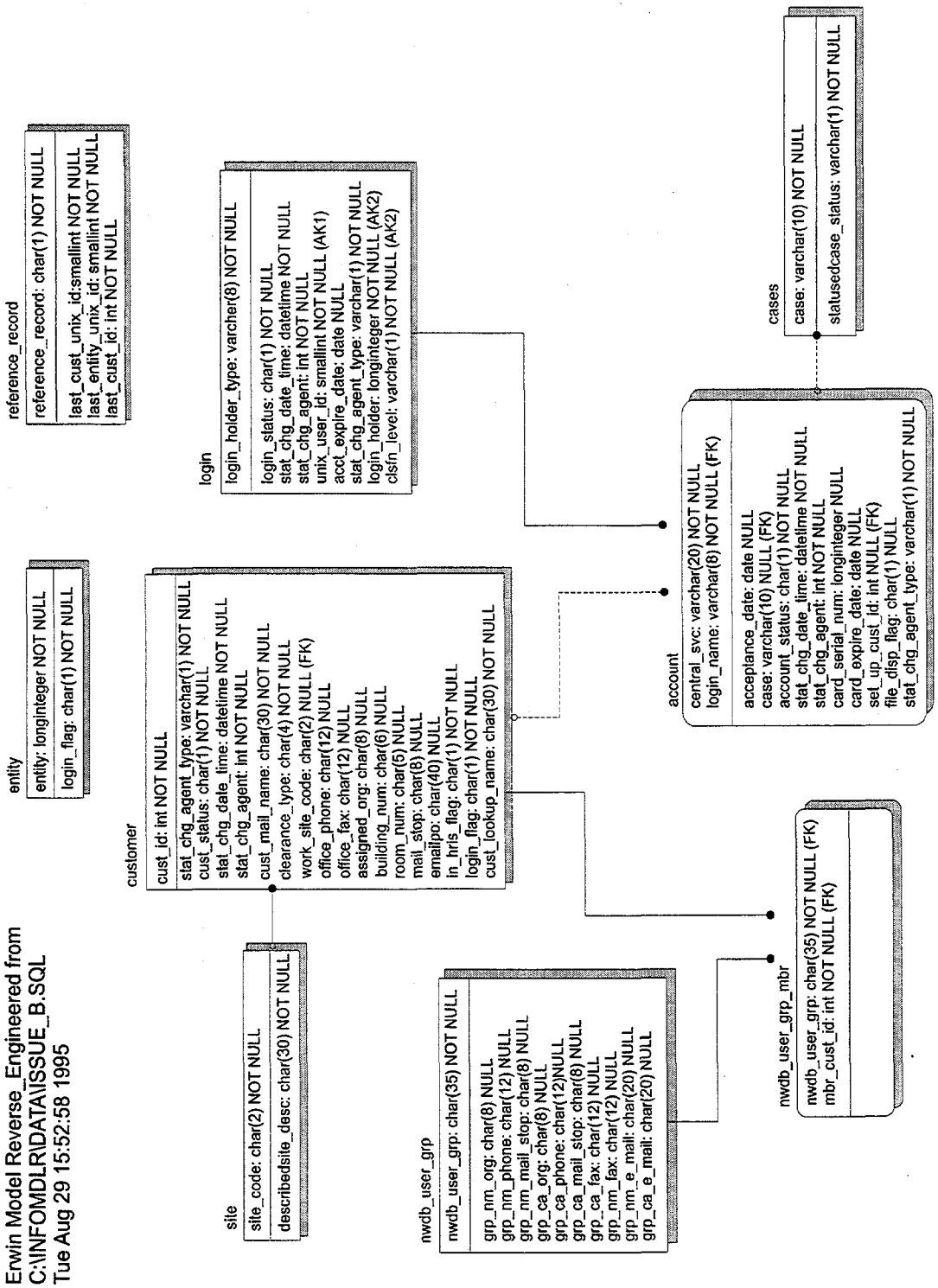
The logical database structure (without vendor software or implementation considerations) is documented with a diagram of the form used in IDEF1X (a FIPS representation of an ER model). This diagram shows the objects and attributes with foreign key relationships.

The IDEF1X format shown here is a standard chosen for government agencies.

NOTE: Keys are listed first in each table with a line following the last item in the key. Data items, which uniquely identify an object, are usually the primary key. The following describes the symbols used in an IDEF1X diagram.

PKxx	Primary key; key that uniquely identifies one object from another
AKxx	Alternate key; a secondary identifier
FKxx	Foreign key; key of another entity wen it migrates into a given entity as a result of a relationship between the two entities
Parent	An entity on the 1-end of a 1-to-many relationship
Child	An entity on the many end of a 1-to-many relationship
———	Identifying relationships (foreign key is part of primary key)
- - - - -	Non-identifying relationships (foreign key is not part of primary key)
Cardinality	The quantifier in a relationship between two entities; a quantifier answers the question, "How many?"
P ● or P ●	Existence of at least one matching child required if parent exists (1: 1 or more)
Z ● or Z ●	Existence of maximum of one child matching parent (1: 0 or 1)
● - - - ●	1:0 or more
- - - ◇	Parent-child relationship where the parent entity is optional (i.e., optional non-identifying relationship)

Figure A.3.3 NWIS IDEF1X Logical Model



Verification and Validation Plan

Deliverable 4

The software verification and validation plan specifies the tasks and resources to evaluate and test a new or enhanced software application. The verification and validation process provides insight into the state of both product and process throughout the development life cycle. Verification involves checking that products in each life cycle phase.

- adhere to prior life cycle phase requirements,
- adhere to standards, practices and conventions specified for that phase, and
- form an adequate foundation for the next life cycle phase.

Validation consists of checking that the software system adheres to system and software requirements.

A template for the use of IIS Organizations is on the web (Reference Appendix E, URL 9). A document example as used by the Electronic Timecard System is on the web (Reference Appendix E, URL 10).

Network Information System Verification and Validation Plan

1.0 Introduction

1.1 Assumptions

The assumptions upon which this software verification and validation (V&V) plan is based include the availability of

- software products,
- test tools that meet the requirements of test team personnel, and
- test team personnel (i.e., knowledgeable individuals from user, developer, test, and technical organizations).

It is also assumed that the software be controlled under configuration management.

1.2 Purpose

This plan specifies the tasks and resources for the verification and validation of the Network Information System application, and it serves as the project plan for V&V activities. Its successful use will be the basis for process owner acceptance.

The content and organization of this plan are based on

- CIO V&V Verification and Validation Plan Template, Version 1.0, November 18, 1996,
- Software and Information Life Cycle (SILC) for the Integrated Information Services Organization, Adaptations of the Sandia Software Guidelines: Issue C

Supporting plans should be prepared to specify particular tests that are to be performed on the suite of applications. At a minimum, a functional (acceptance) test should be specified and conducted to support process owner acceptance. Other tests may include, but are not limited to human factors evaluations and load or other performance tests.

1.3. Scope

The scope of this V&V effort is limited to the development life cycle and requirements documented in the references listed in this plan, summarized as follows. The time frame that this plan covers is January, 1995 through October 1997.

1.3.1 Current Network Information System Applications

There is no current application to be replaced. All specifications will be from separate functions being performed in many groups to manage data on individuals, servers, and network components and their associated access and controls data.

Network Information System "New" Applications

1. Customer List
2. Customer Details
3. Central Services List
4. Central Services Details
5. Accounts List

Web Application Function

Manage and Update Information on Network Components and Customers

1.3.2 New Network Information System Applications Functionality

See Deliverable 1 for the Context diagram and problem statement defining what the system is to do from the broad overview perspective.

New functionality:

Network Information System "New" Applications

Web Application Function

Manage and Update Information on Network Components and Customers

1. Customer List
2. Customer Details
3. Central Services List
4. Central Services Details
5. Accounts List

1.4 Objectives

The objectives to be satisfied by the Software Verification and Validation Plan include

- Verification of requirements, design, and implementation phases, and
- Validation of requirements with respect to scope.

1.5 Applicability

This plan applies to the products of the concept, requirements, design, implementation, and test phases of the software development life cycle with respect to the identified scope.

2.0 References

2.1 Applicable Standards and Policies

- Software and Information Life Cycle (SILC) for the Integrated Information Services Organization, Adaptations of the Sandia Software Guidelines: Issue C.
- Personnel Manual; replaces individual SLPs.
- Testbed Application User Interface Design Guide.

2.2 Planning Documentation

- Network Information System Project Plan (WebSPIN).
- NWIS Gantt Chart 1198.380 (WebSPIN).
- HRIS Project Gantt Chart 1198.300 (WebSPIN).

2.3 Concept, Requirements, Design, and Implementation Documentation

- Concept Design Review (CDR) briefing, November 21, 1995 (WebSPIN).
- Network Information System Attendance and Corrections Project Proposal, November 11, 1995.
- NWIS Functionality (summary) (WebSPIN).
- SILC deliverables (PVCS and WebSPIN).

2.4 Miscellaneous

- B. Beizer, *Software Testing Techniques*, 2nd edition, Van Nostrand Reinhold, 1990.
- *Configuration Management Plan.*

2.5 Business Rules and Policies for Network Information

See the Integrated Information Services Architecture Manual and Personnel Manuals and NWIS SILC deliverables. Sandia has business rules and policies that have been developed to meet our function as a government contracting agency for managing and controlling access to corporate computing resources. These rules and policies are well established and the network information control process ensures adherence to those policies.

3.0 Definitions

3.1 Validation

Checking that a software system adheres to system and software requirements.

3.2 Verification

Checking that products in each life cycle phase

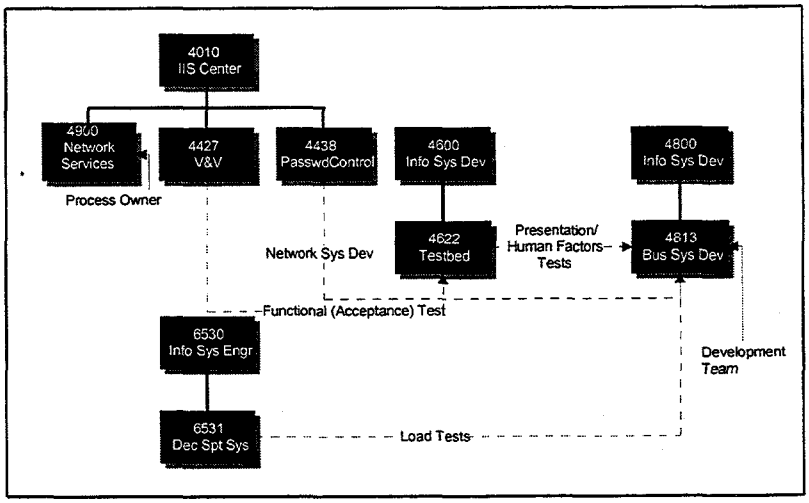
- adhere to prior life cycle phase requirements,
- adhere to standards, practices and conventions specified for that phase, and
- form an adequate foundation for the next life cycle phase.

4.0 Verification and Validation Overview

4.1 Organization

The organization structure of test and development teams is depicted in the following figure.

Figure A.4.1 Organization Structure



4.2 Information Environment

4.2.1 Functional Interfaces

The diagram in SILC Deliverable 1 illustrates how the Network Information System application relates to other processing.

The IDEF0 diagram in SILC Deliverable 1 illustrates how the Network Information System application relates specifically to external processing. The detailed functions are described in Conceptual Business Functions User Model in SILC Deliverable 6.

4.3 Master Schedule

See the NWIS Gantt Chart 1198.380 (WebSPIN) and HRIS Project Gantt Chart 1198.300

4.4 Tools and Techniques

Table A.4.1 Tools and Techniques

Tool support for deliverables or other activities in each phase is listed in the following table.[1]

Life Cycle Phase	*Deliverable*	*Tool Support*
Concept		
	Concept phase task reporting - included in FY95 Life Cycle Review	Project for Windows® Word processor
	V&V phase summary report - included in FY95 Life Cycle Review	Word processor
Requirements		
	Requirements traceability analysis	Excel® spreadsheet or matrix Requirements semantic net
	Requirements evaluation	Word processor Excel® spreadsheet
	Requirements phase task reporting - included in FY95 Life Cycle Review	Word processor
	System Test Plan	Word processor Project for Windows®

[1] CIO V&V Verification and Validation Plan Template, Version 1.0, November 18, 1996.

Life Cycle Phase	*Deliverable*	*Tool Support*
	System Test Procedures/Cases and Execution (1) Server load testing (2) Client GUI testing (3) Client load testing (4) Middleware inspection and analysis (5) Other	 Mercury Interactive LoadRunner® Mercury Interactive WinRunner® Mercury Interactive WinRunner® Mercury Interactive WinRunner® e.g. word processor
Design		
	Design traceability analysis	Excel® spreadsheet ABC Graphics Suite® Annotated design
	Design evaluation	Word processor
	Interface analysis	Word processor User Interface: (1) Analytic evaluation - task analysis, compare UI definition with respect to industry standards and task structure. Output is quantitative data. (2) Expert evaluation - compare predefined categories with UI definition and report on differences, as well as unstructured evaluation (assessment by experts in relevant areas). Output is qualitative data. (3) Human factors evaluation - use human factors checklist; see Testbed Application User Interface Design Guide (DRAFT), December 4, 1995.
	Integration Test Plan	Word processor Project for Windows®
	Component Test Plan	Word processor Project for Windows®
	Test Design	control flow, data flow - OMT tool kit? MM?
	Integration Test Procedures/Cases and Execution (1) Server load testing (2) Client GUI testing (3) Client load testing (4) Middleware inspection and analysis (5) Other	 (1)-(4) same as client/server test suite identified in Requirements phase e.g. word processor
Implementation		
	Traceability analysis	reverse engineering tool TBD
	Code evaluation	static test tool TBD

Life Cycle Phase	*Deliverable*	*Tool Support*
	Interface analysis	User Interface: (1) Expert evaluation - refer to design phase description, and see Testbed Application User Interface Design Guide (DRAFT), December 4, 1995. (2) Observational evaluation - monitor user behavior with the system. Output consists of quantitative and qualitative data. (3) Survey evaluation - elicitation of users' subjective opinions via surveys, interviews, and questionnaires. Output consists of quantitative and qualitative data. (4) Experimental evaluation - assessment of what the interface is to do from a usability standpoint. Output consists of quantitative and qualitative data.
	Component Test Procedures/Cases and Execution (1) Server load testing (2) Client GUI testing (3) Client load testing (4) Middleware inspection and analysis (5) Other	 (1)-(4) same as client/server test suite identified in Requirements phase e.g. word processor
	Build test, audit	compiler(s), web browser, DLLs, as identified by developer
General		
	V&V Phase Summary Report	Word processor
	Issue Reports	Intersolv Tracker®
	Configuration Management	Intersolv Version Managers® (PVCS)

Since the system is a client/server application, a brief overview of client/server testing tools is presented in this plan.

Tests performed may include business rules (acceptance) testing, GUI testing, client and/or server testing, and middleware (i.e., database) inspection and analysis. Specific tests to be performed should be described in one or more supporting plans.

Some automated test tools may be appropriate for some tests, other tools appropriate for other tests. The following automated test tools, which typically provide extensive results reporting are being used:

WinRunner®, a system from the Mercury Interactive Corporation, is a software tool that can be used for GUI or regression testing and is useful to support business rules testing. Scripts captured from the application being tested enable test teams to re-test applications with little marginal effort.

LoadRunner®, also from Mercury, has been used for load testing business software. It is especially useful in identifying "bottlenecks" in the processing infrastructure. Like WinRunner, this test tool facilitates load re-tests.

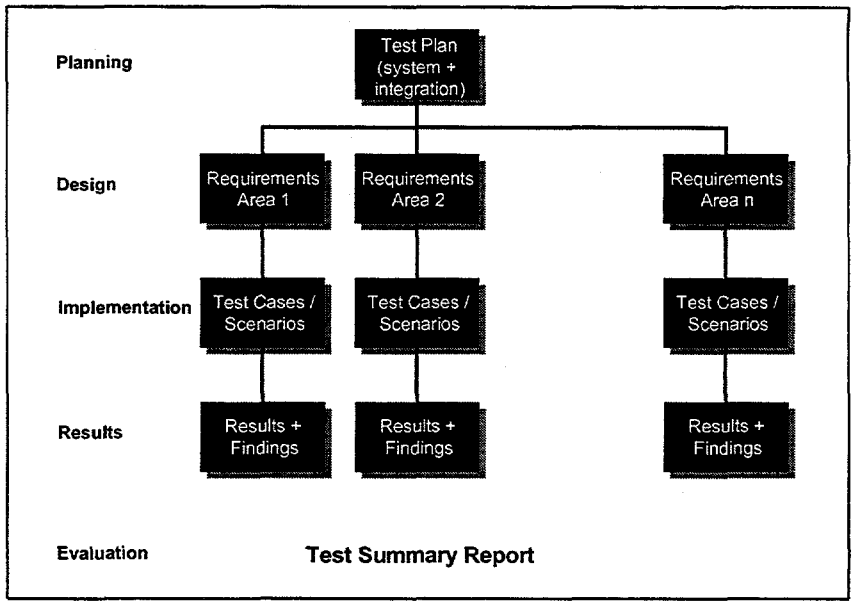
Patrol®, is used at Sandia to monitor database activity during load tests.

4.5 Test Documentation Road Map

Because test teams often want to optimize the time and resources spent on testing activities, they may want to combine system (i.e., requirements-based) testing with integration (i.e., design-based) testing, as indicated in the methodology diagram.

The following figure depicts the test documentation tree, starting with the test plan. The test plan references the test design documents, one for each requirements area, which in turn reference the test cases/scenarios for each requirements area. Test results and findings are obtained after following the test cases/scenarios. The overall evaluation of the results and findings is contained in the test summary report. The V&V life cycle phases are shown on the left side of the figure.

Figure A.4.2 Test Summary Report



5.0 Life Cycle Verification and Validation

Verification and validation deal with the issues pertinent to managing the V&V effort. Basically, these tasks are similar to tasks associated with managing any development effort: *planning*, *review* and *control*. Task details, resource allocations, and task dependencies are shown in the project plan, associated task notes, and other pertinent documentation. V&V tasks will be repeated as necessary, or new tasks initiated, in response to software changes.

6.0 Reporting

This set of tasks represents the result of the plan implementation for management and life cycle V&V and testing activities. This reporting should communicate:

- status of the V&V or testing activity,
- findings,
- added visibility of the development process, and
- increased understanding of the product.

The benefits of meeting these goals are attained only if the reporting is done on a *timely* basis. The most benefit gained is when V&V occurs in parallel to software development, and developers are notified of findings promptly.

7.0 V&V Administrative Procedures

7.1 Issue Reporting and Resolution

7.1.1 Reporting Tools

Issues or faults may be reported also by e-mail, memo, or formal report. The communication should positively identify the source of the fault, including version number, and provide details.

Test teams may use tools such as Intersolv's Tracker® bug tracking tool or Mercury Interactive's LoadRunner®, WinRunner®, or Test Director® products.

7.1.2 Issue Report Distribution and Timing

Test teams may opt to provide summaries of the issues being tracked on a recurring basis to

- management team,
- development team,
- process owner, and
- V&V project team.

7.1.3 Issue Resolution

It is not mandatory that each critical anomaly shall be resolved satisfactorily before the V&V effort can formally proceed to the next life cycle phase[2]. Benefits can be derived from test activities in later phases (e.g., evaluation), irrespective of critical problems found in an earlier phase. However, issue criticality should be estimated and reported.

Response evaluation will be a shared responsibility. Test team members evaluating responses should involve other team members as necessary. An issue report will be considered resolved if the response reflects appropriate action taken, such as a software bug fix, and is checked out by both the developer and test team.

[2]IEEE Guide for Software Verification and Validation Plans, IEEE standard 1059-1993, pg. 46.

7.1.4 Reports Timing

Issue reports should be generated within 24 hours of fault observance. The reporting tools identified in this document may be used.

- Internal reporting of issues (i.e., summarizing general findings and citing results) should be done generally within 72 hours of the analysis. Issues identified in analyses and evaluation may be delayed longer than for software errors, as the impact analysis for evaluations in the requirements and design phases may be broader in scope and require further study.

7.2 Task Iteration Policy

Software products that are input to a V&V or testing task can change due to requirements changes or clarifications, or changes as a result of an enhancement/problem report. When changes are incorporated, some tasks may need to be repeated, or new tasks introduced, depending on the modification.

Each requirement area identified in the scope has an owner responsible for their area. Any test team member may notify that owner if they suspect a change impacts that requirement area. It is the responsibility of the owner to perform the impact analysis. The results of the impact analysis will be saved along with the original notification in a central location accessible by the test team.

The sources of changes notices will be manually perused. Impending changes may be formally announced in documents placed in PVCS, WebSPIN, or other facility. System behavior may be another source of change notification that wasn't documented for some reason.

The owners will notify the test team or be notified of potential change impacts via e-mail. A copy of the e-mail may be deposited in a central location accessible by the test and development teams (e.g., PVCS, WebSPIN).

7.3 Deviation Policy

Just as requirements must be tracked under configuration management, so too must deviations from this or supporting

plans. A summary of all deviations should be included in final reports. Some deviations may require management approval. For example, the change criticality may determine the level of approval required to resolve pertinent issues.

7.4 Configuration Management

The following items will be controlled using Intersolv's Version Manager®.

- documentation,
- test procedures,
- test cases,
- test scripts,
- actual test results, and
- expected test results.

7.5 Standards, Practices and Conventions

Identify the date and version number for documents in titles or in footers or headers. Standard templates may be available in the V&V program library for different types of documents and slides.

Elementary Processes

Deliverable 5

Elementary processes represent the smallest units of work which can be executed to modify the content of the information base and which will leave the information base in a state consistent with integrity rules.

Elementary processes form the basis for analyzing the processes by beginning and ending state and recording the changes made to the data by a person or automated system.

Notes: * items are processes that include the enforcement of the mandatory relationships among tables, shown in Deliverable 4 (see "Logical Information Model—IDEF1X Diagram"). However, the following example demonstrates no mandatory relationships among tables.

This example is for the tables of the NWIS. They demonstrate the elementary processes by table.

Notes: The Optional column notes Null Allowed. The following describes the Data Entry designation:

md	Mixed determination—both the user and other data supplied by the system are involved in changing the data
derived	Only the system may change the data, using a "derivation rule"
user	A person enters or modifies the data without using other data from the database (also called strict determination)

Elementary Processes

Table A.5.1 CUD Actions by Table

CUD/Table Row	Optional	Data Entry	Comments
customer			
cust_id		derived	system-generated key
stat_chg_agent_type		derived	system-generated key
cust_status		md	user/pick list
stat_chg_date_time		derived	system clock
stat_chg_agent		derived	
cust_mail_name		user/HRIS	
clearance_type		user/HRIS	
work_site_code	✓	md/HRIS	user/pick list site.site_code
office_phone	✓	user/HRIS	
office_fax	✓	user/HRIS	
assigned_org	✓	user/HRIS	
building_num	✓	user/HRIS	
room_num	✓	user/HRIS	
mail_stop	✓	user/HRIS	
emailpo	✓	user/HRIS	
in_hris_flag		derived	
login_flag		derived	
cust_lookup-name		derived	
account			
central_svc		md	pick list
login_name		md	pick list from login table
acceptance_date	✓	user	
case	✓	HRIS/FIS	
account_status		md	pick list
stat_chg_date_time		derived	system clock
stat_chg_agent		derived	administrator login
card_serial_num	✓	user	

Table A.5.1 CUD Actions by Table (Continue)

CUD/Table Row	Optional	Data Entry	Comments
card_expire_date	✓	user	
set_up_cust_id	✓	user	
file_disp_flag	✓	md	pick list
stat_chg_agent_type		derived	administrator type
login			
login_name		md	system-generated/user-modified
login_holder_type		md	pick list
login_status		md	pick list
stat_chg_date_time		derived	system clock
stat_chg_agent		derived	administrator login
unix_user_id		md	system-generated/user-modified
acct_expire_date	✓	user	
stat_chg_agent_type		derived	administrator type
login_holder		user	
clsfn_level		md	pick list
NWIS_user_grp			
NWIS_user_grp		user	
grp_nm_phone	✓	user	
grp_nm_mail_stop	✓	user	
grp_ca_org	✓	user	
grp_ca_phone	✓	user	
grp_ca_mail_stop	✓	user	
grp_ca_fax	✓	user	
grp_nm_fax	✓	user	
grp_nm_e_mail	✓	user	
grp_cs_e_mail	✓	user	
NWIS_user_grp_mbr			
NWIS_user_grp_mbr		md	assign *customer* as member of NWIS group pick list
mbr_cust_id		md	pick list

Table A.5.1 CUD Actions by Table (Continue)

CUD/Table Row	Optional	Data Entry	Comments
cases			
case		HRIS/FIS	case must be valid case number from this list
statusedcase_status		md	pick list
site			
site_code		md	site code must be validated from this list
describedsite_desc		user	
entity			
entity		user	
login_flag		md	pick list
reference_record			
reference_record		derived	system-generated keys
last_cust_unix_id		derived	system-generated keys
last_entity_unix_id		derived	system-generated keys
last_cust_id		derived	system-generated keys

Conceptual Business Functions User Model

Deliverable 6

Presentation sets represent the user view of the business functions definition. There are many useful methods for capturing this information. All of the diagrams or storyboards must represent the users understanding of what is wanted and what is the flow.

The storyboard diagrams or (even better) a prototype system are the primary tools for conveying the user information and processes, however, other diagrams are often needed to show flows. A Hierarchy chart, menus with numbers relating to figures, data flow diagram, or event-trace diagram is recommended. Narrative flow and constraint descriptions or use-case diagrams should also be used and carefully numbered to enable the smooth development of the test plan and test cases.

Completed storyboard diagrams with annotated sets denoting groups of data items to be treated as a single unit (as in elementary processes). Standard templates, when available, will be applied to provide a consistent "look and fee" with other systems. Diagrams may be hand-drawn initially, to help the customers view their data and how they will enter data into the computer. Other diagrams will enable the customers to think about their data and how they want to use it, i.e., documents, screen displays, graphs, etc. The application development software or other vendor software may be used, starting the prototyping processes early in the customer-analyst dialogues.

Figure A.6.1 Customer Storyboard Diagram

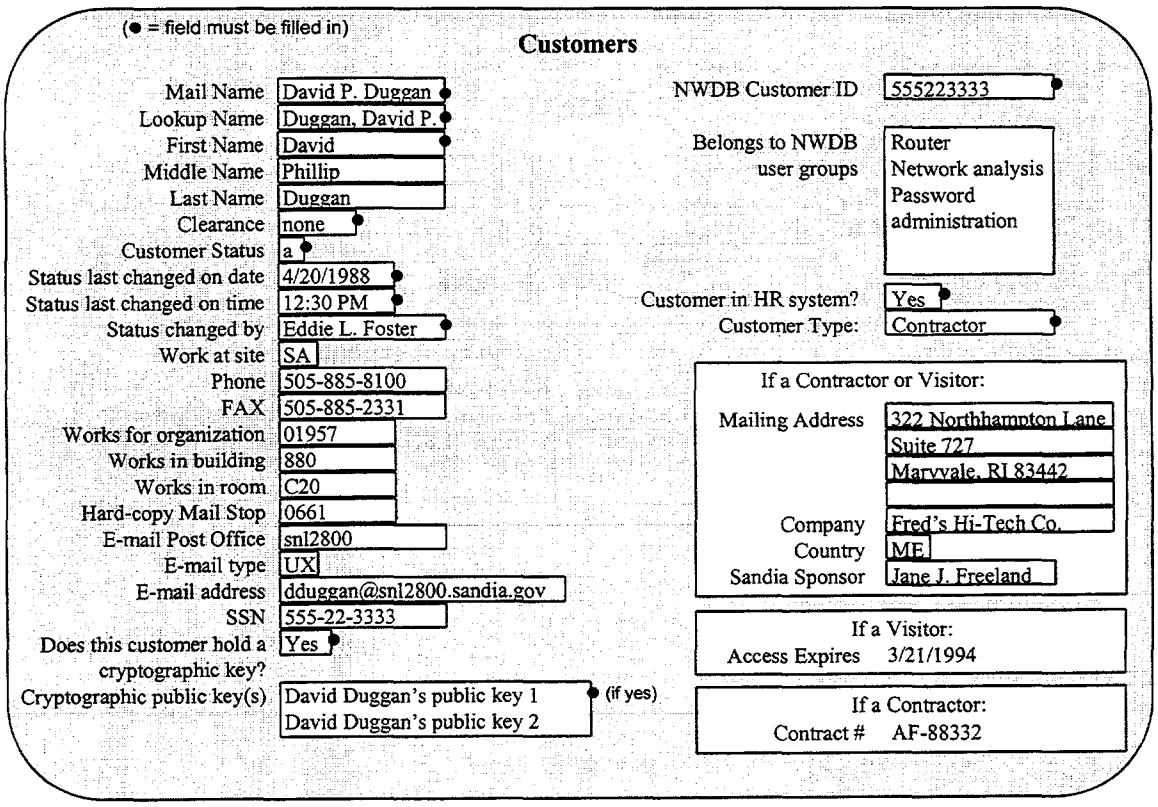


Figure A.6.2 Login Storyboard Diagram

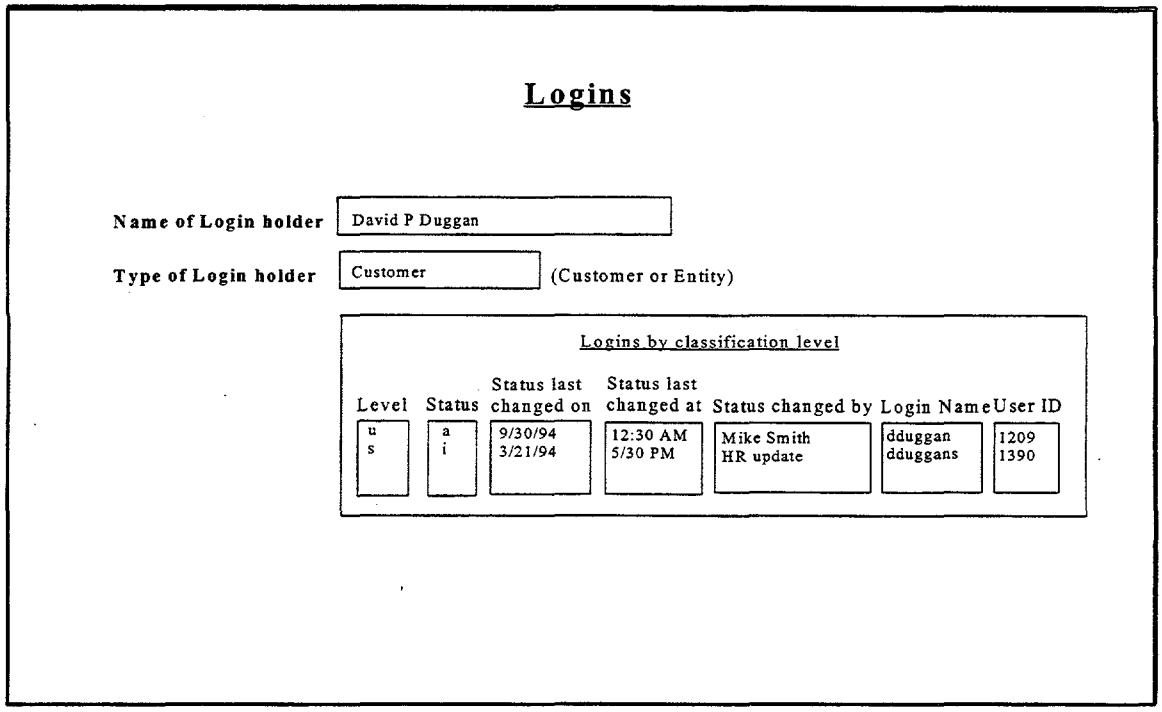
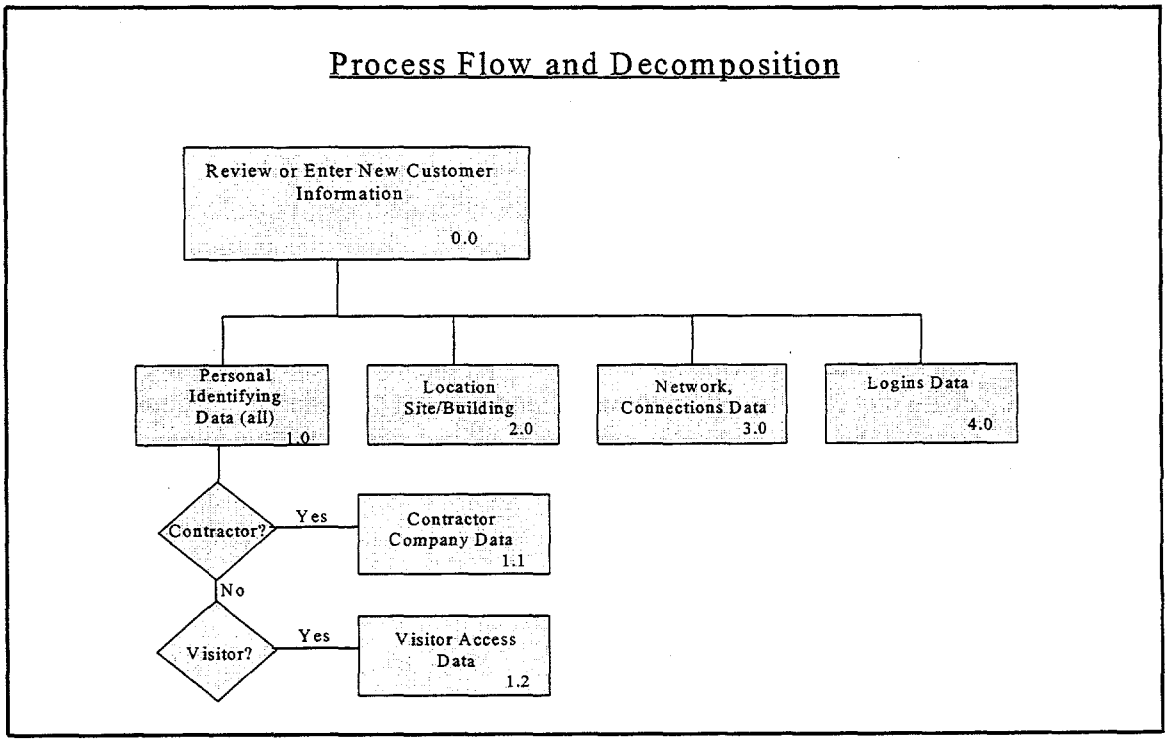


Figure A.6.3 Process Flow and Decomposition



Implementation Strategy

Deliverable 7

The Implementation Strategy is a plan to cover the physical operational proposals. It should list hardware or specific computer system to be used, and proposed software needed.

The systems and technical strategies will be described by defining architectures with brief reviews of alternatives considered. Technical discussions should address performance, security needs, and strategies. A Network Plan should be drawn in as much detail as is needed. Deployment plans should be discussed.

Implementation Strategy for the NETWORK INFORMATION System

Introduction

This implementation strategy includes settling on an underlying technical base for the system, which includes detailed specification of the hardware and software platforms to be used, and specific information about how the hardware or software platforms will affect user access, interfaces, risks, or operations.

The Network Information System (NWIS) will be managed by the Network Technologies Department. This system is being designed as a networked system to contain data collected pertinent to computer users, their network computer parameters for connections, network information, and password administration data.

Current methods of data collection will be examined and updated, as needed. The goals are:

- to use electronic collection and electronic direct transfer of person information from the HRIS to the NWIS,
- minimize use of paper source or output documents,
- to maximize response time,
- provide reliable backup, and
- serve as the record copy of all network administrative data managed by the Integrated Information systems Center for Sandia National Laboratories.

Hardware

The NWIS will be on a networked server computer, with an attached workstation. It will have an attached disk system with automated tape backup.

The specific hardware is as follows:

- SPARCserver 1000E with 256MB memory and 4.2GB internal disk and 8 mm internal tape drive
- SPARC Storage Array (RAID) Model 112 with 12.6GB (expandable)
- EXB-10E Tape Backup Unit with Internal 8 mm drive and software
- SPARC 20 Workstation
- SPARC Color Printer

Calculated capacity of the disk system includes projected 15 years growth in data storage. Disks may be added to this unit to accommodate unplanned dataset additions.

Software

The goal of modern software is to separate the information system into independent layers that may contain reusable building blocks. The three layers are data, function, and presentation. As technology changes, the layer that must be upgraded can change without much impact in the other layers. Following this design approach, the software choices are:

Data

The database software will be a commercial relational database system that is one of the corporate standards with complete database management capabilities and toolsets, SYBASE.

Function

The data retrieval applications will be written in Standard Query Language (SQL), XVT, Server Side JavaScript, Common Gateway Interface (CGI) and JAVA. The Operating System will be Solaris 2.4.5.

Presentation

The Graphical User Interface (GUI) will be HTML, geared to the Sandia standard commercial browser, NETSCAPE, version 2.1 or later.

Basic retrievals and analysis software will be written using the above tools with third party vendor software supplements to be researched and purchased during Phase II of this project.

The operating system will be UNIX-based Solaris, as provided by the hardware vendor.

User Access

The primary users of the NWIS system, in the Integrated Information Center, will be provided unclassified access in Phase I of the project. Most users are physically located in the same building as the NWIS server and the Internal Restricted Network (IRN) is available to all.

Hardware

An SNL standard personal computer or workstation may access the system. This includes the MAC, IBM PC, and UNIX stations.

Software

The user's machine will need the web browser software, NETSCAPE 2.1 or greater. The machine will also need an XVT "client" software module. This module can be automatically downloaded from a Customer Service Unit (CSU) Server. Network software will also be needed to support the corporate TCP/IP standard.

Network

A standard TCP/IP network connection will be needed to obtain access to the server through the corporate network system. Initially, the user's machine will be added as a node to the Local Area Network (LAN) in Building 880 and in Phase II will be extended to include user's on any LAN or sublan of the IRN.

See Figure A.7.1.

Password Control

Passwords and group access memberships will be administered by the Password Control group for the LAN. New Data on Networks will be entered by the Network and Communications group and by the appropriate Network Security Officers for each LAN.

Interfaces - Phase I

1. The data from the Human Resources System is incorporated into the application by a direct electronic retrieval system.

2. Data collected as paper documents (password change requests, etc.) will receive manual data input from those with group permissions for inserting data and updating data.

3. Electronic interfaces include "Re-authorization requests", "Login Names to the LIS system", "Domain name Service files", "Host Files", "LAN connection files", etc.

4. Manual interfaces include "SecurID Account Acceptance Forms" and "Message Asking What to do With Files From Inactive Accounts".

Risks Assessment

The use of the chosen presentation layer "web viewer" is dependent on solutions to hard problems involving new hardware, and new software products. Risk factors for this project's implementation involve:

- acceptance of the chosen development product,
- finding owners and maintainers of the data to populate the database enough to make it a necessary part of the network administration process (as in enforcing data and feeding it to other systems),
- long development times because XVT requires C code, and
- more complex maintenance from long development times.

Operations

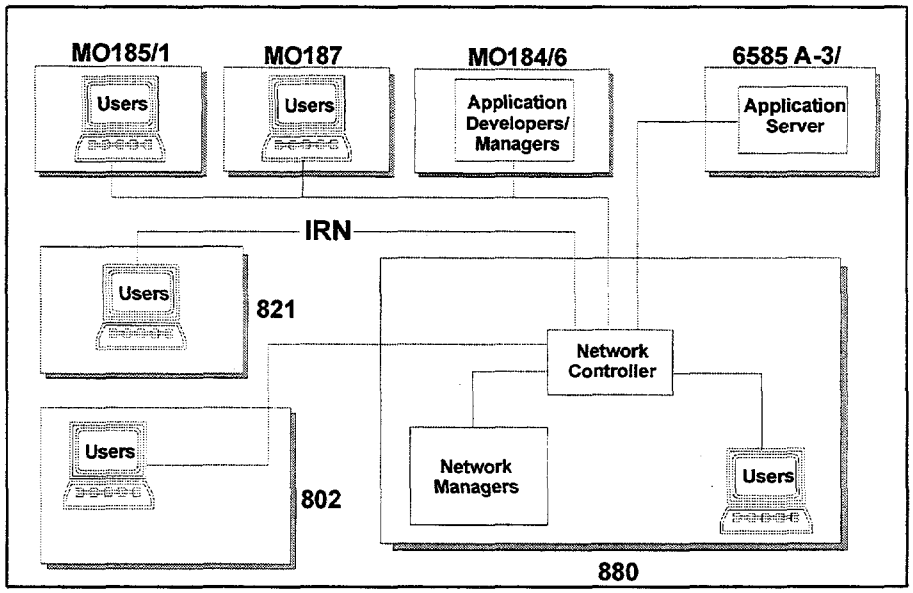
The computer system will be physically located in space owned by the Integrated Information Services organization and all equipment in that computer annex will be owned by the same group. Operations

and maintenance of the system will be performed by the Computer Operations personnel.

Deployment

Deployment will follow the guidelines in the IIS Application Support Transition Guide (Reference Appendix E, URL 6) and the Implementation Checklist (Table 2).

Table A.7.1 Network Plan



Implementable Business Functions (Presentation Model)

Deliverable 8

Extended Presentation Sets consist of storyboard diagrams that include vendor-specific and implementation-specific features, as well as application of standard template features for the enterprise systems. These are normally "extended" or evolved from initial diagrams in Deliverable 6 (see "Storyboard Diagrams"). They are often produced and managed electronically. An evolutionary prototype might begin with diagrams and electronic screens produced by the presentation layer tool. As these designs evolve, test data and coding are added for rapid production of the first product.

When combined with constraints and rules, plus a flow diagram, these extended presentation sets document the business functions model for the software testing team. Flow diagram deliverables may be from any industry standard techniques, such as IDEFO, data-flow diagrams, event-trace diagrams, "if then" structured English, state transition diagram, or the UML (Unified Modeling Language) for object-method descriptions.

This example extracts specific information from the NWIS. A narrative format with cross-reference numbers is used for rules and constraints. When combined with the flow diagram using menu descriptions and constraints, this model completely describes the business processes for the developer and tester.

An example for the Electronic Timecard is on the web (Reference Appendix E, URL 12).

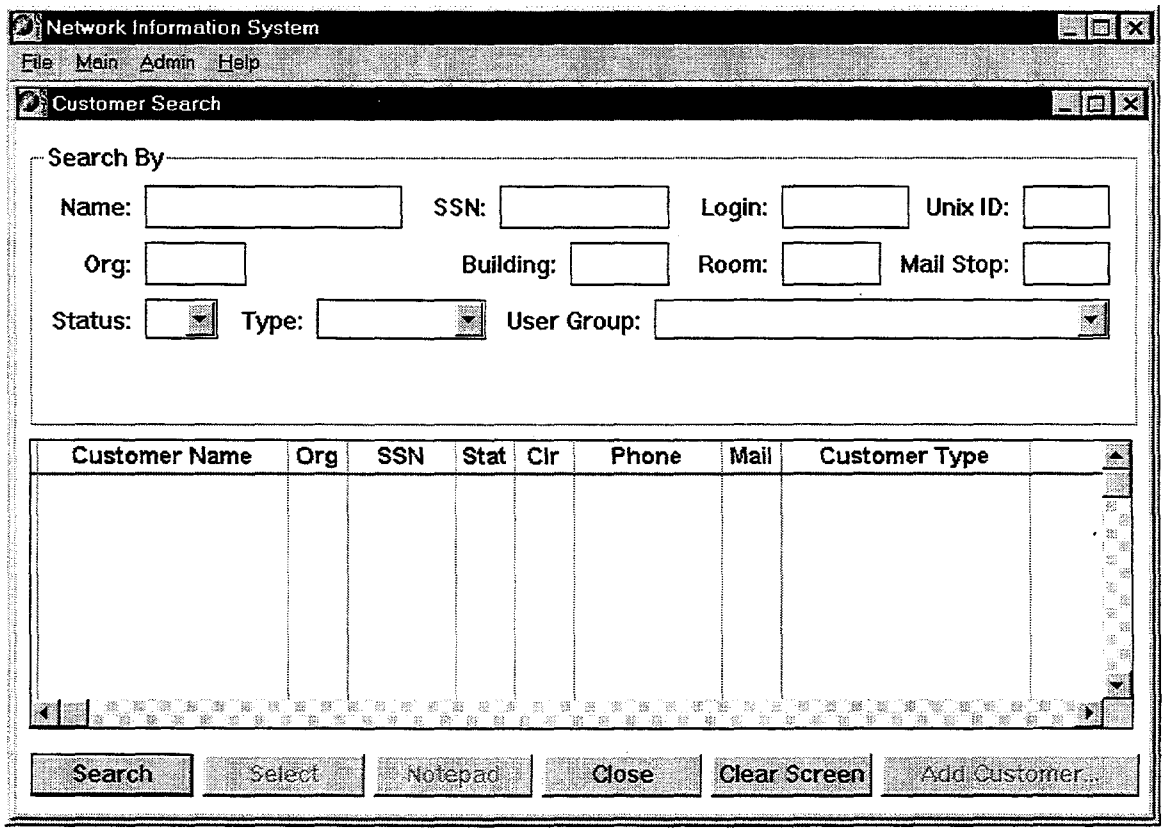
Table A.8.1 Extended Presentation Set Example

Modules

Module Identifier	Module Title
clist	Customer List Screen
cust	Customer Detail Screen
cslist	Central Services List Screen
cs	Central Service Detail Screen
alist	Accounts List Screen

Screen Details

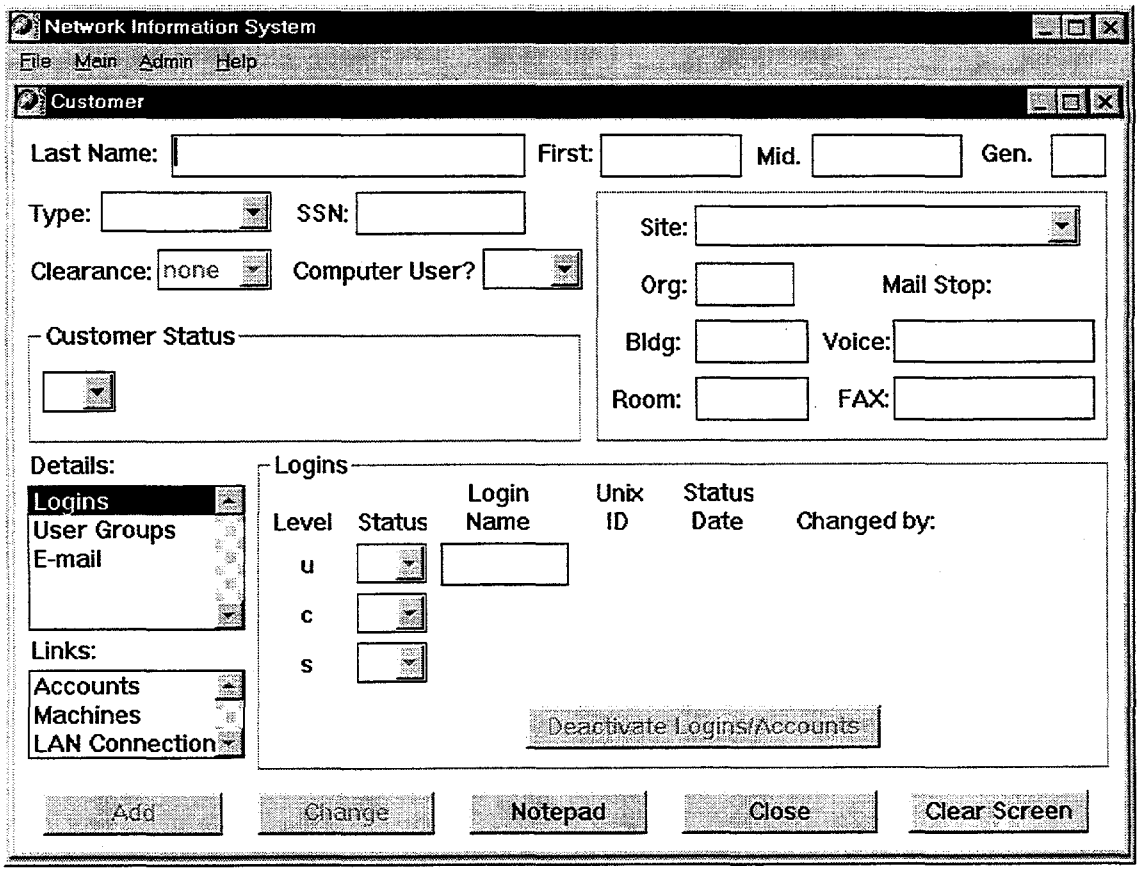
Figure A.8.1 Customer List Screen (clist)



Records Used

contract
customer
login
nwis user_grp_mbr

Figure A.8.2 Customer Detail Screen (cust)



Records Used

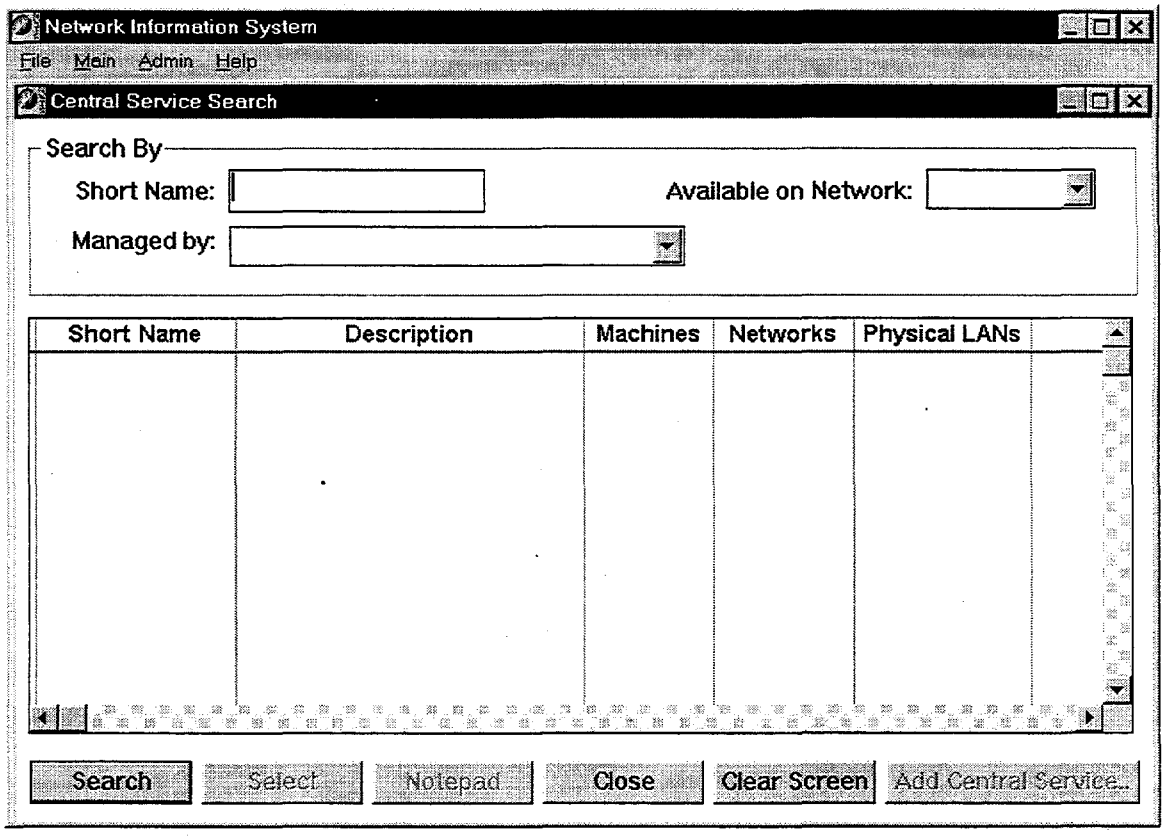
accounts (update)
customer (update)
emailpo
login (update)
nwis_user_grp
nwis user_grp_mbr (update)
reference_record (update)
site

Display Notes

The following display-only field values are updated from the HR system:

- Org
- Mail Stop

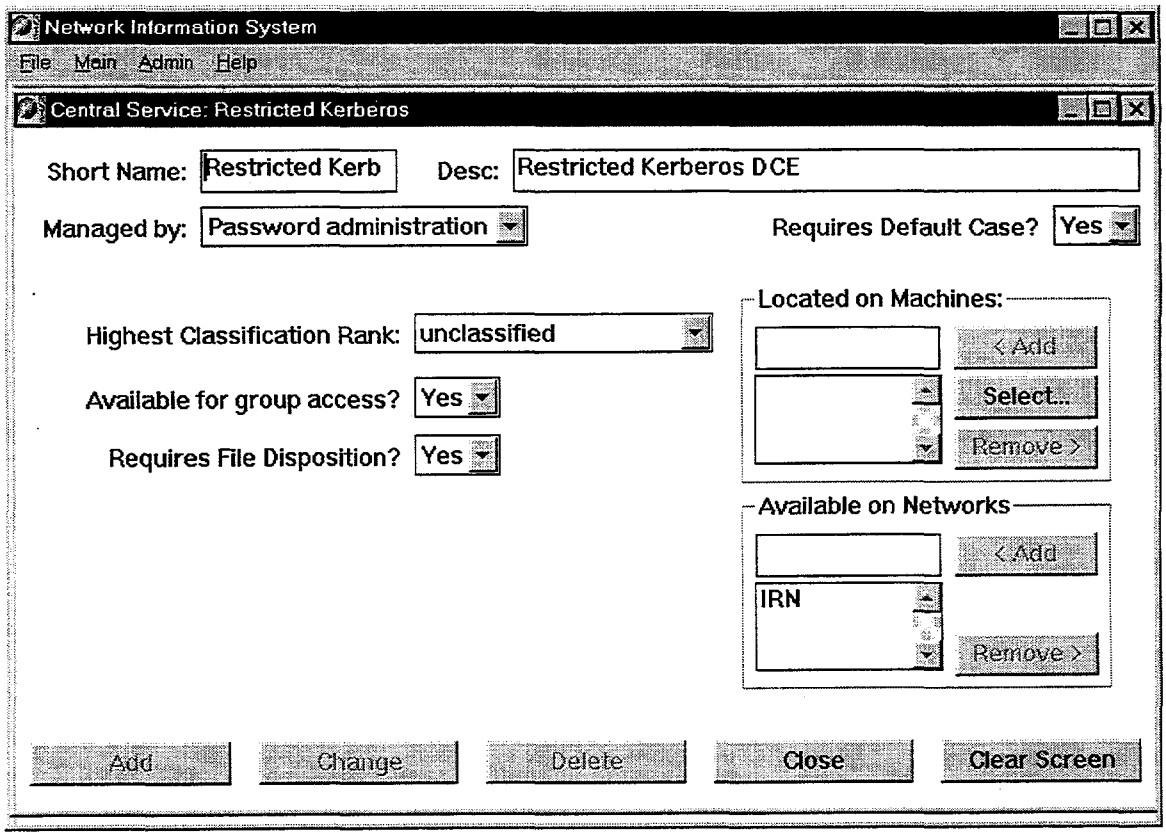
Figure A.8.3 Central Services List Screen (cslist)



Records Used

central_svc
central_svc_machine
central_svc_network
central_svc_subnet

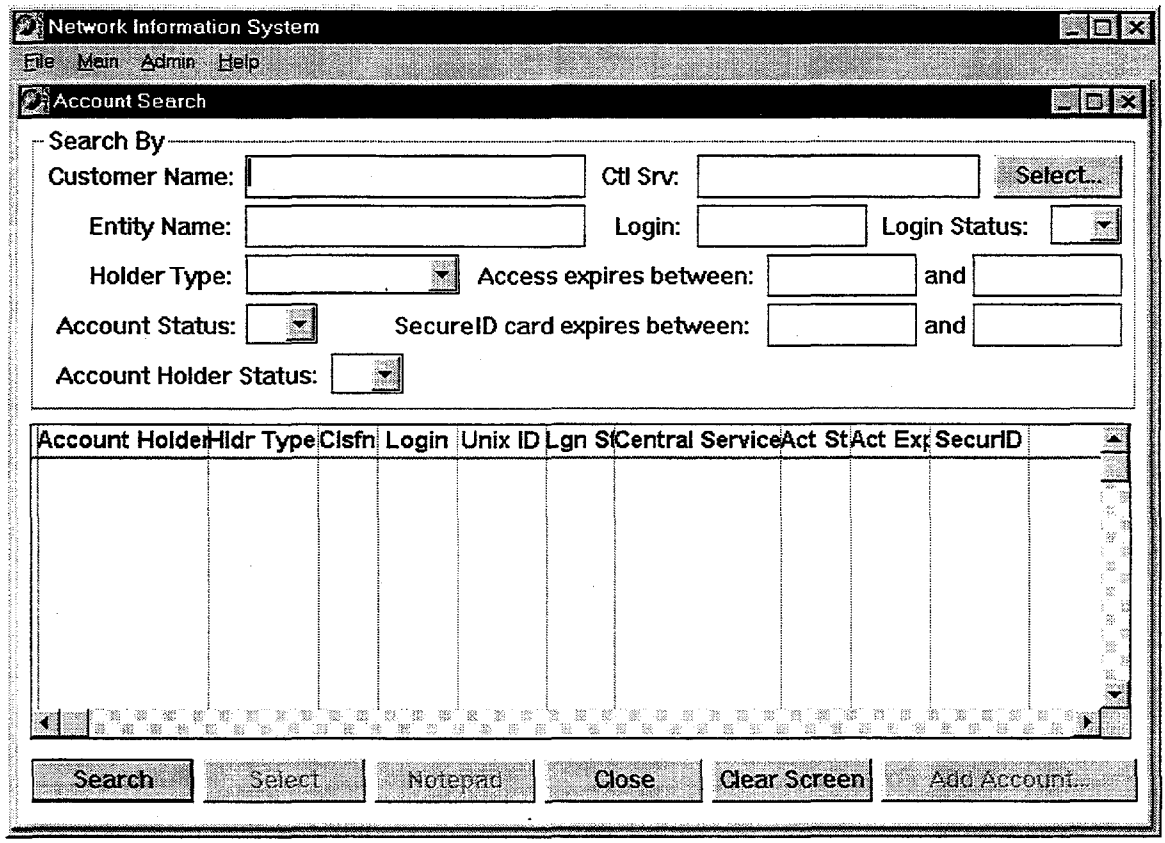
Figure A.8.4 Central Service Detail Screen (cs)



Records Used

central_svc (updated)
central_svc_machine (updated)
machine
clsfn_rank
lan_subnet
central_svc_subnet (updated)
network
central_svc_network (updated)

Figure A.8.5 Accounts List Screen (alist)



Records Used

customer
entity
login
account
central_svc

Display Notes

The fields "Account Holder" and "Central Service" in the list are truncated because of screen space limitations.

Specifications Cross-Reference Table (NWIS Access Controls and Details for modules clist, cust, cslist, cs, and alist)

This table identifies requirements details and applicability to individual modules.

- The Ref and Specification columns identify and state requirements; the remaining columns (i.e., clist, cust, etc.) identify applicability of the specification on a given row to individual modules.
- An "X" at the intersection of a requirement row and module column indicates that the requirement applies to that module. Each requirement may apply to one or more modules.
- A reference number can also be entered at the intersection of a requirement row and a module column to indicate that detailed test criteria have been developed to test the requirement; the number would identify the applicable paragraph in the test plan. Test criteria could be developed to supplement specifications for complex requirements; simple requirements could be tested sufficiently without supplemental information.

Table A.8.2 Specifications Cross Reference Table

Ref	Specification	Module Identifiers				
		clist	cust	cslist	cs	alist
10	User must belong to one of the NWIS user groups "Password administration" or "NWIS administrators" to add records.	X		X	X	
20	User must belong to one of the NWIS user groups "Password administration", "NWIS administrators", or "LAN points of contact" to add records.					X
30	User must belong to one of the NWIS user groups "Password administration" or "NWIS administrators" to change data.		X		X	
40	When a customer is added, a new customer ID must be generated; the customer ID is the primary key to customer. Do the generation as follows: 1. Get the last customer ID used from the reference record and add 1. 2. Set the last customer ID used to the new value. 3. Set the customer ID to that value in the new record.		X			

Ref	Specification	Module Identifiers				
		clist	cust	cslist	cs	alist
50	If SSN, Login name, or Unix User ID is entered, all other data should be ignored and a match made directly on that field. Otherwise, search for all matches. Entering any one of SSN, Login name, or Unix User ID will blank out the other three.	X				
60	Login name must be alphabetic and lower case.					X
70	If the search criteria yields a customer or entity without a login name or with a login name but without any accounts, display the search results (i.e., show all matches, even if the whole report line cannot be displayed).					X
80	Each login name for a login holder should display only once, even if there are multiple accounts for that login name.					X
90	Matches should be obtainable based on partial login name, lookup name, or organization. (Other matches should be by full field content. Say in Help what field can be partially entered. Also remind the user to place leading zeros on the organization if "013xx" is desired, not "13xxx").	X				
100	"Account holder type" is a combination of login holder type (i.e., customer, entity) and customer type. Convert the login holder type "customer" into the three types of customers for display.					X
110	An SSN must have 9 digits, including leading zeros. The SSN may be input with or without dashes. If entered without dashes, the SSN is altered to display the dashes in the format 999-99-9999. SSN is stored without dashes as data type int.	X	X			
120	SSN is required for customers of type "sandian", even though the field is nullable in the customer record.		X			
130	SSN, if present must be unique—no two customers may have the same SSN. SSN is not a required except for customers of type "sandian".		X			
140	Last, first, and middle name fields should be only alphabetic plus any embedded "-" or blank spaces. The first character of the name and the first character after any "-" or blank should be forced to the upper case		X			
150	If the customer first, middle, or last name is changed, the following should happen:		X			

Ref	Specification	Module Identifiers				
		clist	cust	cslist	cs	alist
	1. The customer's first name, a space, the first character of the middle name, and a space (if there is a middle name), and the last name should be concatenated to for the customer mail name. 2. The customer's last name, a comma, a space, first name, and another space (if there is a middle name), the first character of the middle name and a period should be concatenated to form the customer lookup name. Customer lookup name and mail name do not show on the screen.					
160	Clearance code should be defaulted automatically to "none" if a customer is initially added. The clearance code cannot be entered on-line; it is updated from the HR system.		X			
170	If the customer status is first entered as or is changed to "a", a login name and user ID may need to be generated.		X			
180	Site is required for customers of type "sandian", even though it is nullable in the record. The value display for Site is the site description, even though it is not the primary key.		X			
190	Values A-Z and/or 0-9. 1. When a building identifier is entered, the value should be altered to strip any leading "B". 2. Any embedded spaces, blanks, "-", or "/" characters, should be stripped. 3. Lower case a-z characters should be forced to upper case. Edited values will be displayed. The edited value must exist in the liscopy.building table, attribute bldgnum.	X	X			
200	Values A-Z and/or 0-9. 1. Any embedded spaces, blanks, "-", or "/" characters, should be stripped. 2. Lower case a-z characters should be forced to upper case. Edited values will be displayed.	X	X			
210	Phone numbers may be entered on-line for phones in other countries, so there is no set format. Phone numbers are stored without embedded dashes or spaces, so these characters should be stripped out on input.		X			

Ref	Specification	Module Identifiers				
		clist	cust	cslist	cs	alist
220	Phone numbers that consist of 10 characters should have parentheses and a dash inserted for display in the format (999) 999-9999. Phone numbers with more or less than 10 characters should display just as a number.		X			
230	The list box for the Highest Classification Rank field is not the primary key field. For display purposes, the classification rank description is being used, not it's numeric code. Users other than NWIS administrators should never see the numeric code.				X	
240	A central service may only be managed by the NWIS user groups "LAN points of contact" and "Password administration", so rather than having a list box to the nwis_user_grp table, a custom list box will display these two values.				X	
250	The flag fields (i.e., "Available for group access?", "Requires Default Case?", etc.) are displayed as "Yes" and "No", but are stored as "Y" or "N".				X	
260	A central service is a POC-managed central service when it is managed by the NWIS user group "LAN points of contact". Only a Poc-managed central service may be available on a subnet—hide this box otherwise.				X	

Database Schema

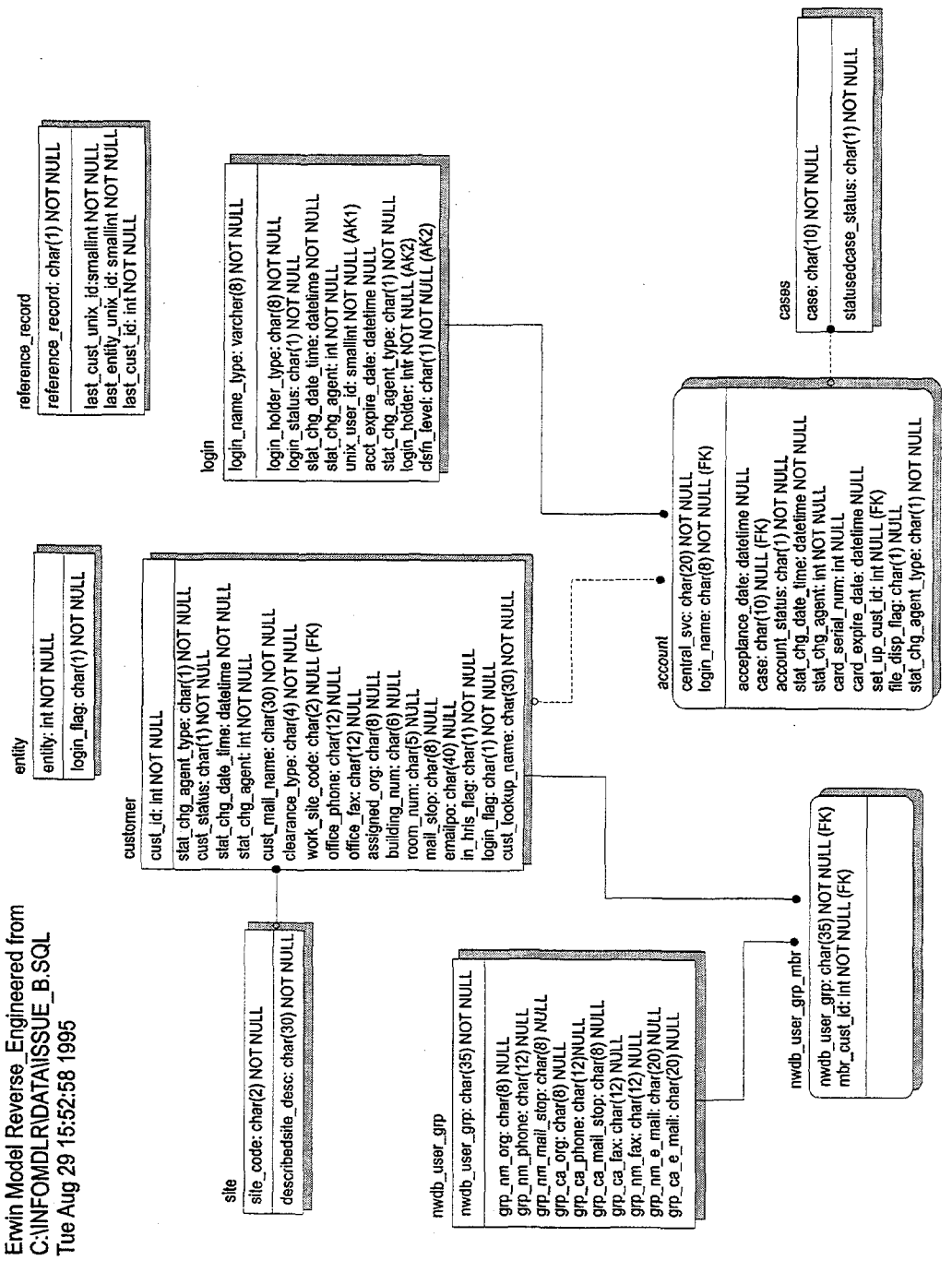
Deliverable 9

The Physical Database, using the constructs of the vendor software and the additional information needed by the implementer of the system, is documented to represent the physical design of the database (in object, relational, or other paradigms). The relational database schema diagram in the form of IDEF1X. Other physical schema diagrams will be used to report physical object. Optionally, the SQL listings may be used. The SQL to support all of the elementary processes for CRUD (Create, Read, Update, and Delete) may be automatically generated from the physical model by a tool.

NOTE: Keys are listed first in each table with a line following the last item in the key. Data items that uniquely identify an object are usually the primary key. The following describes the symbols used in an IDEF1X diagram.

PKxx	Primary key; key that uniquely identifies one object from another
AKxx	Alternate key; a secondary identifier
FKxx	Foreign key; key of another entity wen it migrates into a given entity as a result of a relationship between the two entities
Parent	An entity on the 1-end of a 1-to-many relationship
Child	An entity on the many end of a 1-to-many relationship
———	Identifying relationships (foreign key is part of primary key)
-----------	Non-identifying relationships (foreign key is not part of primary key)
Cardinality	The quantifier in a relationship between two entities; a quantifier answers the question, "How many?"
P or P	Existence of at least one matching child required if parent exists (1: 1 or more)
Z or Z	Existence of maximum of one child matching parent (1: 0 or 1)
	1:0 or more
	Parent-child relationship where the parent entity is optional (i.e., optional non-identifying relationship)

Figure A.9.1 Database Schema



Deliverable 10 (Issue B) - deleted in Issue C (Deliverable 10 content was included in Deliverable 8)

Process Model

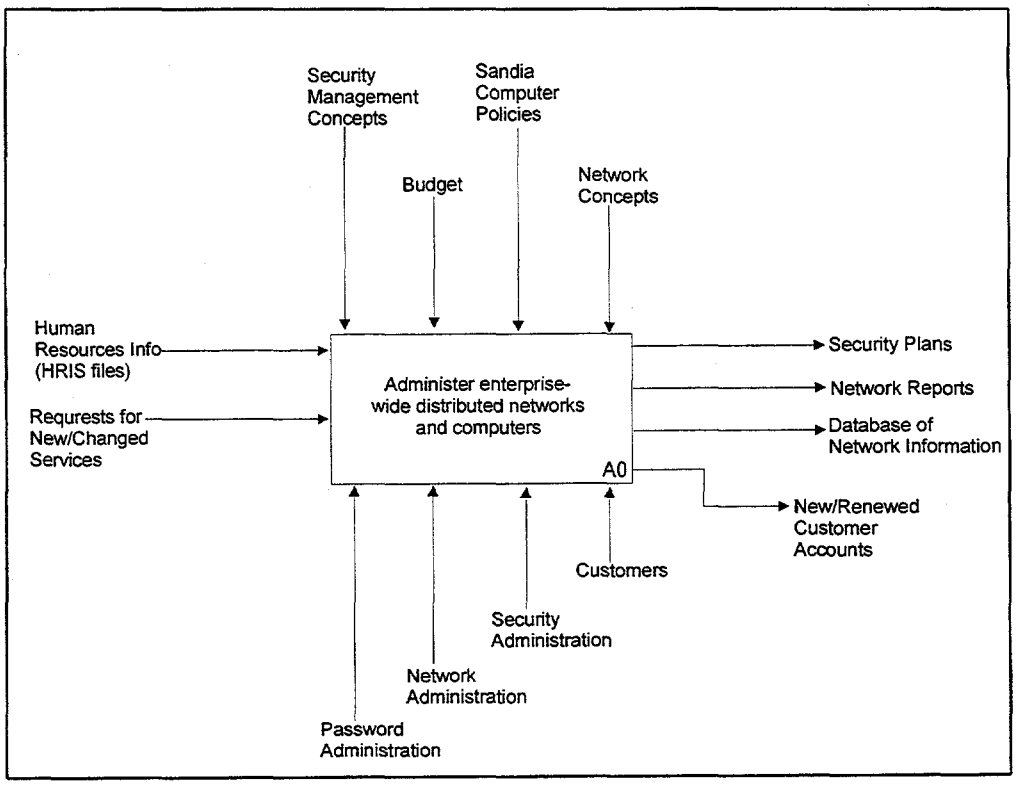
Deliverable 11

The process model will describe the system's "non-user" functionality (which was described in Deliverable 6 and 8). These functions may be specific algorithms, batch processes, operational quality checks, mirroring or logging traces for security and backup, etc.

Just as the Conceptual Business Facts User Model (Deliverable 3) evolves into a more thorough and detailed Logical Model and finally the database schema (Deliverable 9) the process model described in the "Conceptual Business Functions User Model" (Deliverable 6) in large systems will evolve into a more detailed and complete model described in the Deliverable 8, "Implementable Business Functions Developers Model". Other functions not in Deliverable 8 of the system such as Batch Processes and Interfaces to COTS may optionally be described by a Process Model. Industry standard techniques may be used: IDEF0 Decomposition (a government standard); Data Flow Diagram, Event Trace and Event Flow Diagrams, State Diagrams, Mathematical Equations, Decision Tables Natural Language Descriptions, and Pseudocode or structured text.

Each lowest-level function description is related back to its identifying event, i.e., the event that occurred in the systems environment that caused that function to be needed.

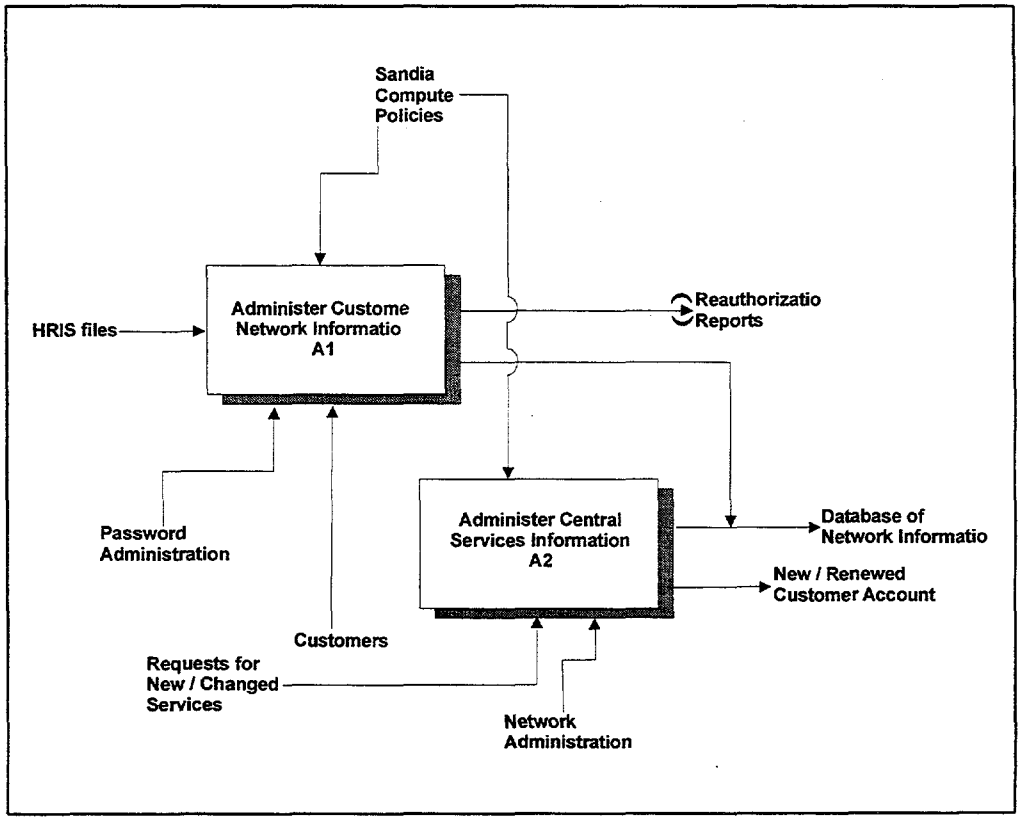
Figure A.11.1 Node A-0, Network Database Example



Reading from the top, clockwise:

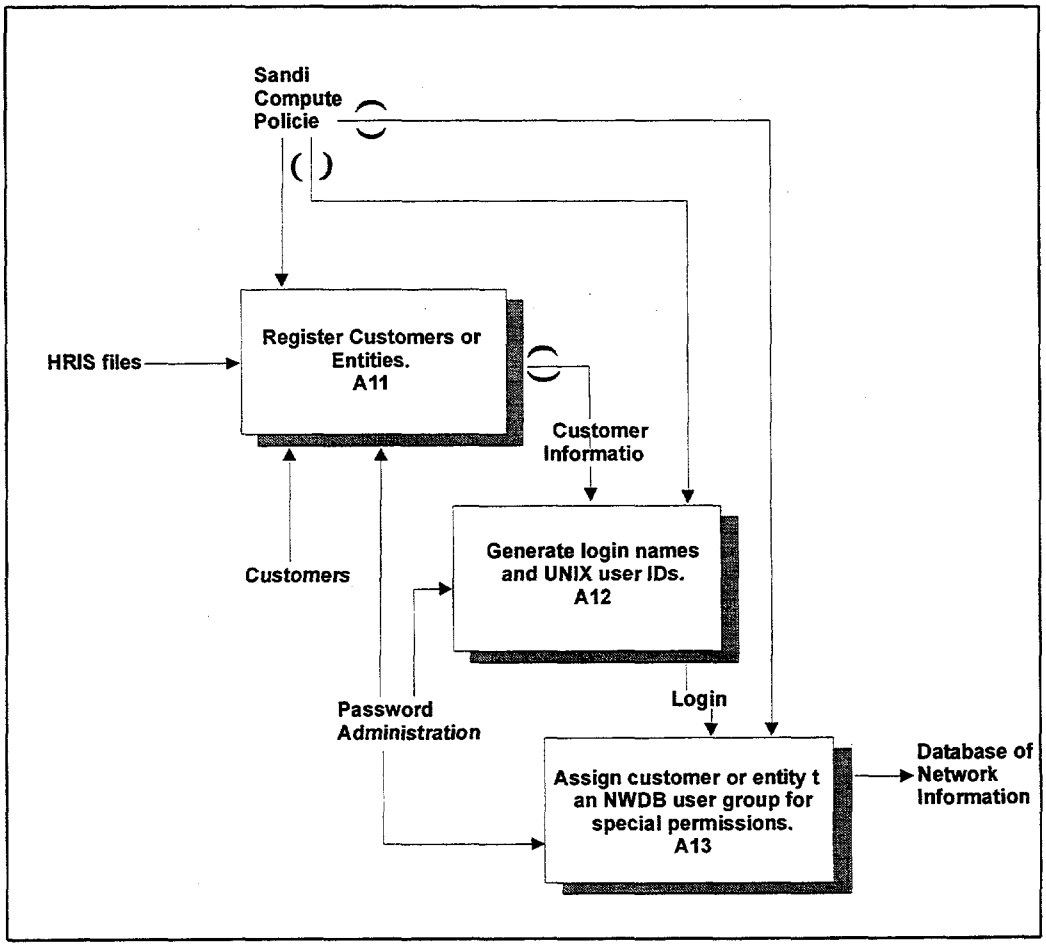
- the arrows entering the top of the box containing the process description show "controls" on the process.
- the arrows exiting to the left show the outputs,
- the bottom arrows show mechanisms, i.e., answers to who or what will act on this process, and
- finally the arrows entering from the right show those inputs that will be affected by the processes and transformed to an output.

Figure A.11.2 Node A0, Administer Enterprise-Wide Distributed Networks



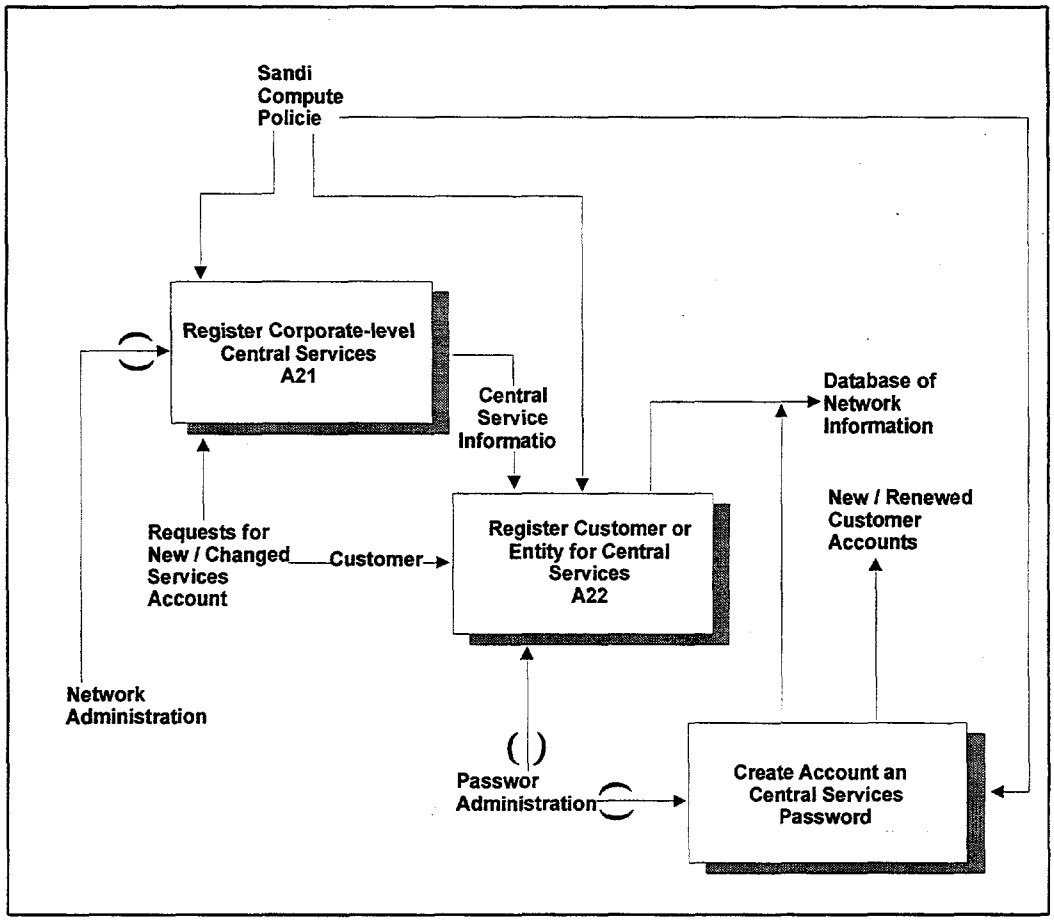
The A0 process shown here is decomposed into two subprocesses, A1 and A2. If an external entity was not noted with exact wording in the higher-level diagram, a tunnel marker displays around the arrowhead.

Figure A.11.3 Node A1, Administer Customer Network Information



The A1 subprocess is further broken down into activities A11, A12, and A13. Each of these can be decomposed again until the granularity is such as to describe the software processes to be managed in writing and deploying the system.

Figure A.11.4 Node A2, Administer Central Services Information



The A2 subprocess is further broken down into activities A21, A22, and A23.

Test Plan, Test Database, and Test Data

Deliverable 12

Test Data and Test Database

Test data may be acquired in the modeling phase as valid and invalid examples of each fact. Testing will check application algorithms and output, and it will check the results of application processes. Testing will also attempt to validate the modeling examples developed early in the SILC process.

The test database will typically be a significant data set to establish and carry out the test plan. It must be a reusable baseline database, and it should be established early in the development process.

One or more versions of test data and the test database may be needed. It is recommended that frequent backups be performed of the database during testing, with each backup having a unique version identifier. The following sequence of events illustrates a possible database backup schedule for a functional testing scenario.

- Save all data and data structures to record the pre-test environment.
- Conduct tests using data inputs expected to fail.
- Save the database data affected by the processing.
- Examine the data to verify expected failures from the test.
- Conduct tests using data inputs expected to process successfully.
- Save the database data affected by the processing.
- Examine the data to verify expected successful processing.

Test Plan

One or more software test plans, designed to be used with the V&V Plan, provide details that apply to testing one or more components of a software system. They may cover a particular kind of testing, such as requirements-based testing (i.e., system testing, business rules testing, performance testing), or design-based testing (i.e., integration testing, link testing). Plans may also consist of multiple levels of plans, such as those for a subsystem or subsystem component unit.

Test plan details will typically address the kind of test being performed, objectives, features to be tested, features not to be tested, approach (i.e.,

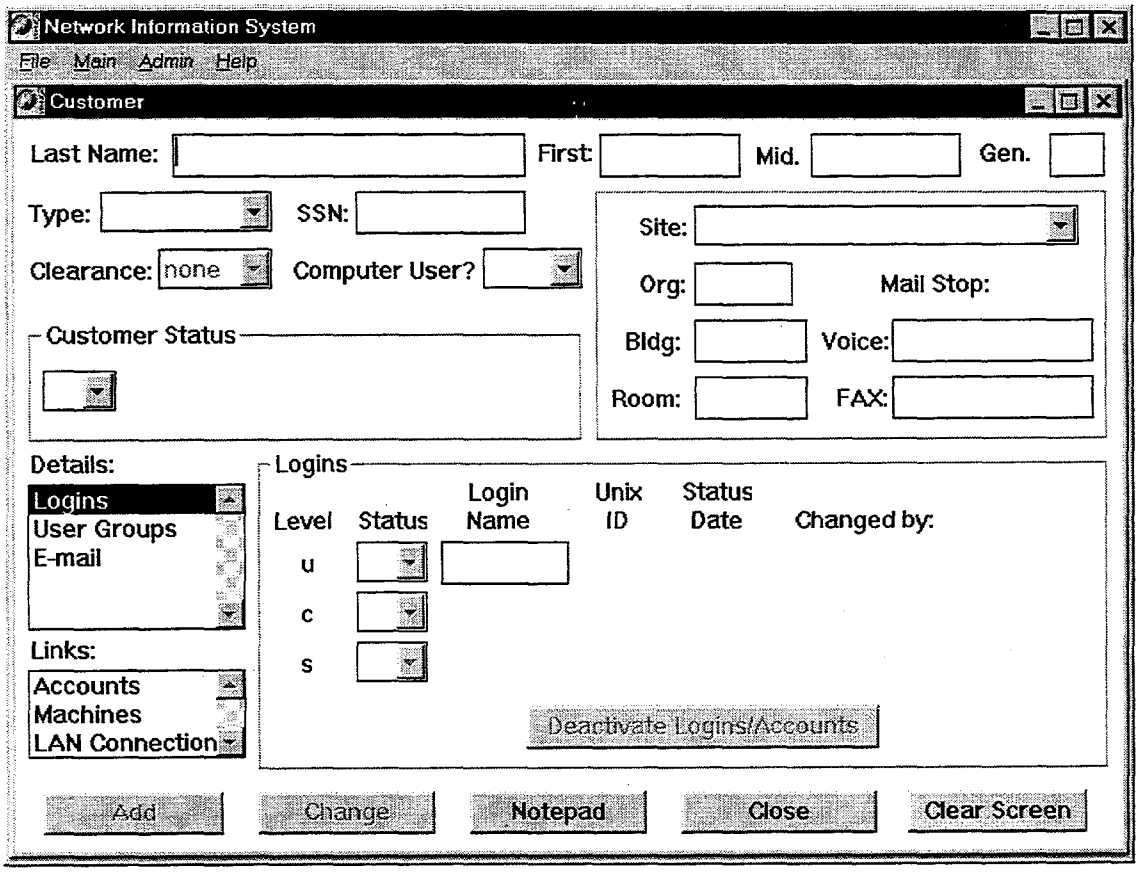
tools, strategy, constraints), item pass/fail criteria, suspension criteria and resumption requirements, test deliverables, testing tasks, test environment, responsibilities, staffing and training requirements, schedule, risks and contingencies, approvals, and references. The use of automated test tools (i.e., Mercury Interactive's WinRunner, LoadRunner, and Test Director) will influence the nature of the test plan significantly.

The use of automated test tools can assist in the management of thorough testing activities, but these tools typically require a significant degree of expertise, and they compound the complexity of testing activities. Manual testing reduces the complexity of testing activities, but it also limits the scope of testing activities while placing greater demands on the time of test team members. A combination of automated and manual testing may be appropriate in some instances.

Test plans that have been placed on the web include examples of WinRunner and LoadRunner testing scenarios. Reference Appendix E, URL 11, URL 12, URL 13, and URL 14.

The detailed specifications of screen functions and constraints established for each process, as recorded in the developer's model, SILC Deliverable 8, become the foundation for functional or business rules testing. The following example illustrates the addition of test criteria to deliverable 8 specifications.

Figure A.12.1 Customer Detail Screen (cust)



Records Used

accounts (update)
customer (update)
emailpo
login (update)
nwis_user_grp
nwis user_grp_mbr (update)
reference_record (update)
site

Access Controls & Details

See Detail Equivalent Reference Table cust items.

Display Notes

The following display-only field values are updated from the HR system:

- Org
- Mail Stop

Specifications Cross-Reference Table

This table identifies requirements details and applicability to individual modules.

- The Ref and Specification columns identify and state requirements; the remaining columns (i.e., clist, cust, etc.) identify applicability of the specification on a given row to individual modules.
- An "X" at the intersection of a requirement row and module column indicates that the requirement applies to that module. Each requirement may apply to one or more modules.
- A reference number can also be entered at the intersection of a requirement row and a module column to indicate that detailed test criteria have been developed to test the requirement; the number would identify the applicable paragraph in the test plan. Test criteria could be developed to supplement specifications for complex requirements; simple requirements could be tested sufficiently without supplemental information.

Table A.12.2 Specifications Cross Reference Table

Ref	Specification	Module Identifiers				
		clist	cust	cslist	cs	alist
110	An SSN must have 9 digits, including leading zeros. The SSN may be input with or without dashes. If entered without dashes, the SSN is altered to display the dashes in the format 999-99-9999. SSN is stored without dashes as data type int.	X	X			
	Test Criteria Attempt entry of invalid SSNs: SSN having less than 9 digits SSN having more than 9 digits SSN that includes alphabetic characters Attempt entry of valid SSNs.	**Expected Result** Reject Reject Reject Accept Accept		**Test Data Reference** colson (ssn 111223333) hdclay (ssn 111423333)		
120	SSN is required for customers of type "sandian", even though the field is nullable in the customer record.		X			
	Test Criteria Attempt entry of a record having type "sandian", leaving the SSN field blank.	**Expected Result** Reject		**Test Data Reference** dhflore (ssn 999441111)		

Ref	Specification	Module Identifiers				
		clist	cust	cslist	cs	alist
	Attempt entry of a record having type "sandian", entering a valid SSN.	Accept		npzucho (ssn 112223333)		
130	SSN, if present must be unique—no two customers may have the same SSN. SSN is not a required except for customers of type "sandian".		X			
	Test Criteria	**Expected Result**		**Test Data Reference**		
	Attempt entry of a record having type "sandian" and an SSN identical to a record already stored in the customer database.	Reject		chradfo (ssn 222113333)		
	Attempt entry of a record not having type "sandian" and an SSN identical to a record already stored in the customer database.	Reject		jestrom (ssn 333112222)		
	Attempt entry of a record not having type "sandian" and an SSN not identical to any record already stored in the customer database.	Accept		cbuckm (ssn 014443333)		

The foregoing example illustrates the addition of test criteria directly to the specification tables contained in deliverable 8. Close coordination between the developer and the test team enables this approach, which results in optimum modularity in the documents. The developer has published the specifications; the test team simply adds to those specifications, repeating nothing that has been published. An alternative to this approach would have a copy of the specification tables being extracted from deliverable 8 and inserted into a separate test plan. In either case, it is important to place all documents under configuration management, with each document version being identified uniquely with a version number. The test report can then positively link a given version of specifications to a given version of test criteria and data to the version of the software tested.

A test plan template for the use of IIS organizations is on the web (Reference Appendix E, URL 15).

Other examples of test plans have also been placed on the web. These examples illustrate both functional and performance test plans. Reference .....

Deliverable 13 (Issue B) deleted
Deliverable 13 content is included in Deliverable 8.

External Interfaces

Deliverable 14

Most systems rely on data and controls from other systems, either by direct network links or file transfer. The interfaces must be documented to the level of detail required for future maintenance of the software. If the data must be converted for this application, a conversion map will show the input and transformation of each field, and the mapping to the new software.

The Conversion Map is optional.

This example extracts specific information from the NWIS.

External Interfaces

The NWIS example includes these interfaces:

- *Inputs from Other Systems*—The Network Database receives, from Human Resources, basic data about customers who are registered in Human Resources (note that many customers are not registered in Human Resources), a list of valid contracts with begin and end dates, a list of valid buildings, a list of valid cases and case statuses, and the clearance level ("L" or "Q") of each cleared person.
- *Outputs to Other Systems*—The Network Database generates and sends Login service accounts, LAN connections, and CSU services to the Financial Information System, sends six types of files to the Domain Name servers to drive Internet traffic, and sends basic user, account, machine, and LAN connection data to the Trouble Ticket system for reference purposes.
- *Other Outputs*—The Network Database also generates about 30 other types of output, many of which are data files to feed various hosts and other network services.

The data must be converted for this application. The following conversion map illustrates the input and transformation of each field, and the mapping to the new software.

Table A.14.1 Conversion Map

	Old Tables			New Tables		
Facts	**Tables**	**Column Name of Sentence Subject**	**Column Name of Sentence Object**	**Table(s)**	**column Name of Sentence Subject**	**Column Name of Sentence Object**
A customer has a first name. On HR load, derive by parsing out name	user_info rpt_person	ussno ssnum	fname name	customer	cust_id	first_name
A customer has a middle name. On HR load, derive by parsing out name	user_info rpt_person	ussno ssnum	mname name	customer	cust_id	middle_nam e
A customer has a last name. On HR load, derive by parsing out name.	user_info rpt_person	ussno ssnum	lname name	customer	cust_id	last_name

	Old Tables			New Tables		
Facts	**Tables**	**Column Name of Sentence Subject**	**Column Name of Sentence Object**	**Table(s)**	**column Name of Sentence Subject**	**Column Name of Sentence Object**
A customer is identified by a mail name. If the customer first, middle, or last name is changed, do the following: The customer's first name, a space, the first character of middle name and a space (if there is a middle name), and last name should be concatenated to form the customer mail name.	user_info rpt_person	ussno ssnum	fname mname lname name	customer	cust_id	cust_mail_ name
A customer is identified by a lookup name. If the customer first, middle, or last name is changed, do the	user_info rpt_person	ussno ssnum	fname mname lname	customer	cust_id	cust_lookup_ name
following: The customer's last name, a comma, a space, first name, and if there is a middle name, another space, the first character of the middle name and a period should be concatenated to form the customer lookup name.			name			
A customer holds a clearance. Clearance code must be either "Q," "L," or "none." Clearance code should automatically default to "non" unless otherwise obtained from the security file.	clearance	ssn	clearance_ type	customer	cust_id	clearance_ type
A customer is typed by a customer type. Note: Customer type.	user_info rpt_person	ussno ssnum	emp_type pertype nemtype	customer	cust_id	cust_id
A customer is identified by a SSN. SSN, if present, must be unique. If not blank, check that no two customers may have the same one, and report error if do!	user_info rpt_person	ussno ssnum	ussno ssnum	customer	cust_id	ssnum
A customer works at a site, NWIS values not the same as HR. Need translate to HR codes. Must be upper case. Site is required for customers of type "sandian."	user_info rpt_person	ussno ssnum	site wrsitcd	customer	cust_id	work_site_ code
A customer answers at a phone number. Phone numbers from HR should be stored without embedded dashes—just as a number.	user_info rpt_person	ussno ssnum	uphone offcphone	customer	cust_id	office_phone
A customer receives FAXes at a phone number.	rpt_person	ssnum	faxnum	customer	cust_id	office_fax
A customer works for an organization. A customer must work for an organization if the customer type is "sandian." Org. must be a five-digit number, with a leading zero as necessary. In the HR files org. has often not been zero-filled to the left. All org. numbers should be zero-filled to 5 places, e.g., "1" should become "00001," "1956" should become 01956." Also remove any section numbers, i.e., dashes and any numbers following the dashes.	user_info rpt_person	ussno ssnum	uorg asorg	customer	cust_id	assigned_org

	Old Tables			New Tables		
Facts	**Tables**	**Column Name of Sentence Subject**	**Column Name of Sentence Object**	**Table(s)**	**column Name of Sentence Subject**	**Column Name of Sentence Object**
A customer works in a room. Strip any embedded spaces, blanks, "-" or "/."	user_info rpt_person	ussno ssnum	room ofcroom	customer	cust_id	.room_num
A customer receives internal hard-copy mail at a hard-copy mail stop. Mail stop is required for customers of type "sandian." NWIS has 5 characters—need chop off 1.	user_info rpt_person	ussno ssnum	mailstop mailstop	customer	cust_id	mail_stop
A customer may be registered in the HR system. This flag should be set to "Y" if and only if the customer appears in an HR file. This flag will automatically be set to "N" anytime a customer is added directly on-line, and then automatically set to "Y" only when and if they appear in an HR file.	- keep in future	derive on HR load		customer	cust_id	in_hris_flag
A customer is marked with a status. Note: Customer status	user_info rpt_person	ussno ssnum	status perstat	customer	cust_id	cust_status
A customer had their status last changed on a date.	user_info	ussno	datetrans	customer	cust_id	stat_chg_ date_time
A customer had their status last changed at a time.	- keep in future			customer	cust_id	stat_chg_ date_time
A customer had their status last changed by an agent. whotrans is a Login Name. Some may be missing. Convert to customer ID for new system.	user_info	ussno	whotrans	customer	cust_id	stat_chg_ agent
none	user_info	acct_status		customer		
An NWIS user group may be contacted in California through a hard-copy mail stop.	- hand enter			NWIS_user_grp	NWIS_user_grp	NWIS_user_grp
A customer belong to an NWIS user group. (load LAN POCs only)	- hand enter			NWIS_user_grp	NWIS_user_grp	NWIS_user_grp

List of Reused Elements

Deliverable 15

Sharable routines such as functions and methods will be documented. Those existing functions found in the repository will be listed, along with new sharable routines that are additions to the repository.

NWIS Example

Reused Object Types and Fact Types

HRIS information as shown in the conversion map (see "External and Internal Interfaces" on page A-79). In summary, the following HRIS tables are reused: case, building, clearance, contract, and basic person data.

Reused Functions

XVT/C function libraries

User Help Documentation

Deliverable 16

User's Guide

The purpose of user documentation is to present information in a concise, helpful, and organized manner that is necessary for accomplishing tasks. This resource includes information pertaining only to the software tool. Other documentation, such as policies, procedures, and tutorials, is outside the scope of the software project. The user documentation set includes:

- System Overview
  - Purpose
  - Scope
  - Intended audience
- Glossary of Terms
  - Field definitions with examples
- Task Procedures
  - Steps required to accomplish a task or solve a problem
- Quick Reference
  - Mechanics of operating the system, such as navigating, editing, entering special characters, searching, keyboard shortcuts
  - Structure of the user interface, such as flow or order of screens, general explanation of screen design such as menu bar symbols
- FAQ (Frequently Asked Questions)
  - Simple troubleshooting procedures
  - Pitfalls to avoid
  - Who to call for help
  - Documented bugs
  - Enhancements for future versions

Information pertaining to field domains, such as field length and type of characters allowed, should be embedded in the screen design.

The following steps are provided in lieu of an extensive NWIS example.

1. Decide which documentation tool(s) will be used, based on the chosen platform(s) and database. The selected tool(s) will determine the documentation style. If necessary, a tool should be selected that will accommodate multiple platforms so only one version of the

documentation is managed. The software developer shall inform the writer of any special considerations for interfacing the completed documentation with the source code for on-line help.

2. Match the documentation set to the level of user audience, as defined in the Problem Statement (Deliverable 1).

3. Decide which format(s) will be used to present the information. The on-line help format will always be employed. Other formats, such as hard copy documentation or HTML on the World Wide Web, may be required. If other formats are required, they should be automatically produced from the original documentation, so only one version is managed.

4. Create the System Overview, based on the Problem Statement (Deliverable 1).

5. Automatically extract object definitions and examples from the Conceptual Information Model (Deliverable 3) to create the Glossary of Terms. Considering the level of user audience, it may not be necessary to define all of the objects.

6. Extract procedural information from the Business Functions User Model (Deliverable 6) to create the Task Procedures. This should be combined with information presented in the Implementable Business Functions Developers Model (Deliverable 8) to completely define the steps for each task.

7. Create the Quick Reference, which will include mechanics of the operating the system and the structure of the user interface. Some of this information will be found in Implementable Business Functions Developers Model (Deliverable 8). Other information will be obtained from the software developer. The standard GUI template or guideline may also have a Quick Reference or provide information for the application reference.

8. Create the FAQ (Frequently Asked Questions) from various sources, such as the software developers, system testers, help desk, and users of the system. This document should be updated frequently. The system should be designed so that the Corporate Computer Help Desk can respond to the user questions first and refer calls to application maintenance personnel second. Access to system change request procedures and browsing to check for request status is one way to provide information on bugs and future enhancement information.

Future State Goal: Design user interface so that no additional user documentation is required.

The User Guide for the NWIS is an excellent on-line Guide and Help System using Web-based technologies (Reference Appendix E., URL 16).

User's Network/Hardware Support Guide

The following description of what should be provided is given here in lieu of an NWIS example.

Distributed applications of varied types have been deployed at Sandia, some purchased, some written in-house, some multi-tier client-server applications and many WWW based. All run in a distributed manner, utilizing various network services scattered across the Sandia enterprise network. Support personnel must manage and support these applications, but they often lack the broad but detailed knowledge of the various components and services being used. For support purposes, the following process will characterize and document the operation of these distributed applications. Users need to be able to recognize the normal variations in application response so they can report conditions that are not normal. Developers and purchasers to assess the network effect and response of an application during the initial implementation stages.

The analysis of an application network should:

- Make decisions about the distributed deployment of the applications. For example, should executable code be resident on the end-user computer or downloaded each execution from a server. (Reference Deliverable 7).
- Identify network services used by the application and allow verification that the services are operated at compatible time schedules and response levels. (Reference Deliverable 4, 12, & 17).
- Allow for incorporation of built-in measures of key application responses and error conditions. (Reference Deliverables 4, 12, & 17).
- Provide the basic information that will be needed by the help desk and trouble shooting personnel should any problems with the application be reported by the end user. (Reference Deliverable 16).

The basic steps in documenting an application for network support are listed here.

1. **Create a list of contacts and their roles/responsibilities.**

It is critical to know who is involved and how. The list could include programmers, database administrators, key users, project leaders, LAN managers, server managers, CSU personnel and any other key players. The list will evolve as the investigator becomes increasingly knowledgeable about the application.

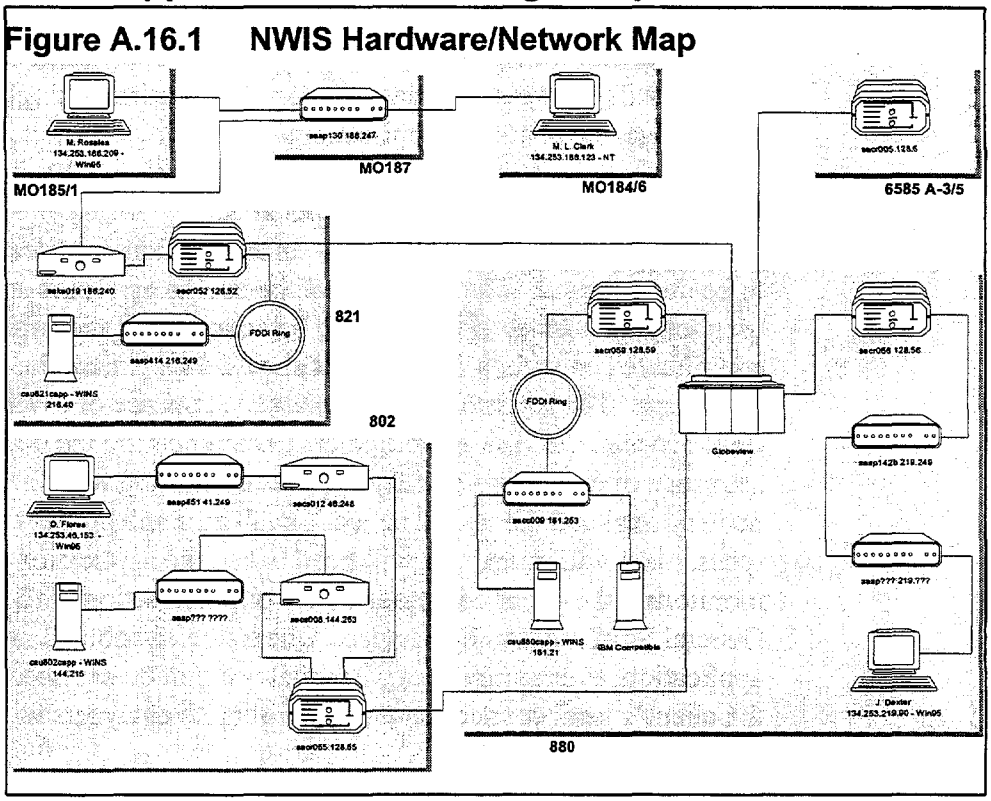
2. **Document how the application operates.**
   There are two related views of how an application operates which need to be documented. The first is what the user experiences and expects. A user account/password will need to be acquired. If the application is complicated enough, a formal class should supplement the user document. The second view is the actual sequence of events and data flow between the various computers. Documenting the detailed sequence of operations usually requires consulting with the application developer(s) and the use of network analyzers and probes of various types. At a minimum, you will need to run the application while monitoring the client computer's network connection with a Network General Sniffer or similar device. Due to the distributed nature of the application, events may take place that will not be captured; looking at the client's local connection and other remote analyzers will need to be used.

3. **Make a list of all network components used.**
   Include all computers as well as any switches, hubs, routers or other components. The information should include IP addresses, domain names, aliases and computer contact information such as owner, location, emergency contact, hardware type, OS, etc.

4. **Create an application map.**
   Using Visio Professional drawing software, create a map of the network as used by the application. Include all components from step 3 above. Note that while a detailed map containing all components is extremely useful to someone who needs to fully understand the application's operation, a simpler conceptual diagram can also be useful when interacting with end-users.

NWIS Application and Message Sequence

Figure A.16.1 NWIS Hardware/Network Map



When a user signs on to his desktop client machine to use this application, the following message sequence takes place (authenticating the client machine - its domain, and the user; then running the application). The major parts of the message sequence are:

1. The user turns on the client computer which registers with the WINS server and obtains the address of the authenticating domain controllers (PDC).

2. The user enters a name and password which are authenticated by the PDC - this step will allow access to the NWIS executable code held on the CAPPS server.

3. The user clicks on the NWIS icon which causes executable code and library routines to be loaded from the CAPPS server to the client computer.

4. Start-up of the NWIS executable requires the client computer to request a name to IP address translation from the DNS computer.

5. The user provides a username and password which are passed-to and authenticated by the sybbravo computer.

6. Subsequent queries and use of the executable sends queries to the sybbravo database computer with answers to the client.

Test Results

Deliverable 17

The test team follows the test plan and documents the results of testing activities. A detailed test plan that specifies all tests performed for an application, with specific test data. This plan can reduce the details in a test report in which no deficiencies were identified to confirm that tests were performed successfully. Other particulars such as variances from the test plan, a comprehensiveness assessment, an evaluation of the application, recommendations, and findings should be reported as appropriate.

For load testing, various tools can be used to measure traffic patterns and response times. For a baseline characterization to be used as a reference for future comparison, a simple Sniffer capture noting traffic, error rates, transaction response times seen by the client, etc. might suffice. More detailed information may be required if the purpose is to look for flaws and bottlenecks in the application. In this case, RMON2 probes and/or other application layer probes can be deployed across the network and used to create traffic maps, response profiles and other reports as needed.

One technique found to be valuable to some customers is to track application response time and availability from the end user's perspective. This can be done using a Windows scripting tool like Mercury Interactive's WinRunner. A script is developed that mimics a user repetitively logging in and running the application. WinRunner then keeps track of application behavior and response times.

The use of automated test tools (i.e., Mercury Interactive's WinRunner, LoadRunner, and Test Director) will influence the nature of the test reports significantly. Typically, automated test tools provide a wealth of output recording test results and statistics. Test results that have been placed on the web include instances where WinRunner and LoadRunner test tools were used in testing activities. Reference Appendix E, URL 17, URL 18, and URL 19.

NWIS Test Results

For network load tests (network assessment) the following example illustrates an NWIS results report.

NWIS Load Tests

Tests and analysis have shown that the time necessary to load and start-up the NWIS application are acceptable and range between 6 and 25 seconds. The size of the executable and library files are small enough to fit acceptably onto any NT computer that meets Sandia's common operating environment specifications. Query times in the most common functions of the NWIS range from 2 to 10 seconds. Once the application is started and the user has logged-on, there is no large data transmission load - messages to or from the client range from 1,000 to 10,000 bytes for each user query. No large files are transmitted except in the start-up phase.

For manual tests, the following NWIS example illustrates the reporting of deficiencies.

Test Incident Report - NWIS Example

tir#:	1000
date:	06/27/95
platform:	pc
application:	NWIS.exe
version:	
function:	NWIS User Group
tester:	fay
role:	NWISdev
retest date:	
retester:	
retest result:	
legend:	
severity level:	

0- informative
1- user interface not friendly (ex: buttons, menu not arranged well)
2- user interface is confusing (ex: directions not clear)
3- data is confusing (ex: information has to be derived)
4- data is incorrect
5- application response is inappropriate or incorrect
6- response is inordinately slow (ex: database access inefficient)
7- does not conform to software specifications
8- other

9- does not conform to datamodel
10- application aborts or does not return to user
****

incident#: 1
spec#: none
error message:
severity: 0- informative
description: I was able to invoke two copies of this screen, and query "User Group" == 'Test Group' on both screens. I then changed the "mail stop" on one screen. The second screen was not refreshed with the new information. This may not be in error but may confuse the user. (Should the user not be allowed to query the same entity on two different screens?)

incident#: 2
spec#: gen-70
error message:
severity: 7- does not conform to software specifications
description: The system did not request a confirmation for a delete (see steps to duplicate error below)

incident#: 3
spec#: none
error message: 'wugrps_pbdelete' dependent foreign key constraint...constraint name = 'central sc952442517'...command has been aborted...sql server error...failed deleted NWIS_user_grp
severity: 1- user interface not friendly
description: The above error messages appeared when I attempted to delete a user group.
(see steps to duplicate error below)

incident#: 4
spec#: usergrp-30
error message:
severity: 7- does not conform to software specifications
description: mail stop is required to be 4 digits upon entry. I entered 2 digits and the system left filled with zeroes instead of displaying an error message.
(see steps to duplicate error below)

incident#: 5
spec#: usergrp-40
error message:
severity: 7- does not conform to software specifications
description: Since telephone numbers allow '-' and '(' and ')' I was able to input a value that only consists of '-' and '(' and ')'. Ex: (----)—
(see steps to duplicate error below)

****

steps to duplicate error

incident#: 2
–add a test row
–query the row to confirm that it was added
–delete the row
–notice that there is no confirmation message
–query to confirm that the row was in fact deleted

incident#: 3
–query NWIS User group == 'Test Group'
–select delete
–notice the succession of error messages

incident#: 4
–add or change the "mail stop" as a two digit value
–click on "add" or "change"
–notice that the value is left padded with zeroes

incident#: 5
–"add" or "change" a test "user group"
–enter "voice", "fax" values as '(----)—'
–click on "add" or "change"
–notice the successful update

****

programmer's notes

incident#:

****

retest notes

incident#:

****

Appendix B

One Page Description of SILC Deliverables

Conceptual Design Review

- Deliverable 0: Project Proposal/Project Plan
- Deliverable 1: Problem Statement
- Deliverable 2: Information Business Facts
- Deliverable 3: Conceptual and Logical Information Models
- Deliverable 4: Software Verification and Validation Plan
- Deliverable 5: Elementary Processes
- Deliverable 6: Conceptual Business Functions User Model

Detail Design Review

- Deliverable 7: Implementation Strategy
- Deliverable 8: Implementable Business Functions (Presentation Model)
- Deliverable 9: Database Schema
- Deliverable 11: Implementation Business Functions (Process or Function Model) (OPTIONAL)
- Deliverable 12: Test Plan, Test Database, and Test Data
- Deliverable 14: External and Internal Interfaces
- Deliverable 15: List of Reused Functions

Testbed and Final Design Review

- Deliverable 16: User Documentation
- Deliverable 17: Test Results

**NOTE:* Not all of these deliverables are required. During project management, they are negotiated among the sponsor(s), project management, and the project team. Results will be sent to the IIS stakeholders. The deliverables make reference to the IIS Software and Information Life Cycle (SILC) Issue C. This document can be viewed on the Web (Reference Appendix E, URL 1).

Deliverable 0: Project Proposal/Project Plan

Description

Strategic information planning for systems within the corporate model includes conformance to standard architectures, integration and resource allocation. Each project must identify its position/role as it relates to the overall corporate environment. In addition each project must identify the customer expectations in terms of resources, performance and scope.

Purpose

To ensure that the project team understands the current corporate strategies, principles, and objectives. Describe the project in terms of customer needs and discuss implementation alternatives which will meet the customer's requirements and align with corporate requirements. Negotiate performance, cost, and schedule.

Example:
- A project proposal (ref. CIO Project Template in PDI).
- SILC Appendix A, Deliverable 0.

Questions answered from the analysis
- Who is the customer/funding source for this project?
- Why is this project needed?
- How does this project meet corporate objectives?
- Who are the major stakeholders? Users ?
- Who has the responsibility for verifying and validating requirements?
- What are the major milestones and deliverables?
- What is the project's cost?
- What software tools, DBMS are proposed?
- Proposed Network requirements?
- Which hardware platforms and configuration?
- Who's going to take care of the system when completed? Cost?
- Any major purchases required?
- Any major obstacles or risks?

Deliverable 1: Problem Statement

Description

The problem statement defines the problem space in concise terms and establishes the boundaries of the problem (i.e., what is part of the project vs. what is not). The problem statement should contain three elements; problem narrative, a context diagram, and a high-level function diagram.

Purpose

The primary purpose of the problem statement is to establish an initial and mutually acceptable understanding between the software engineer and the customer as to the nature and scope of the problem. It gives the software engineer a starting point for determining the major components of the system. The documents produced in this exercise should be viewed as the primary sources for management briefings and training. Versions of these documents should be maintained to keep them current with the system "as built".

Example:
- SILC document Appendix A, Deliverable 1.

Questions answered from the analysis
- Specifically, what is the problem you are trying to solve?
- What are the major functions, objects?
- Is this a new system?, conversion?, modification?
- Which existing systems will be affected? In what way?
- Will this system require any information from existing systems?
- Must this system provide data to any other system?
- Will this project be completed in phases? What are they?

Deliverable 2: Information Business Facts and Source Documents

Description

The process of clearly establishing a complete (or at least significant) list of business requirements is iterative. Deliverable 2 is the first iteration of this process and will provide a foundation for more detailed analysis in subsequent deliverables. A simple, yet structured approach for communicating and validating these rules with customers, subject experts, and potential users will be used : English sentences and concrete examples (pictures, drawings, existing documents, documents, etc.). This information is collected through interviews, JAD sessions, existing systems or documents. Care should be taken to work within the problem boundaries as stated in Deliverable 1. If the boundaries need to expand, based on new knowledge from this deliverable, the project plan needs to be re-negotiated and modified (performance, cost, schedule).

Purpose

The purpose of this deliverable is to begin a series of dialogue with the customer, users, subject experts and major stakeholders on the specifics of their problem area. The output from this deliverable (sentences, diagrams, examples) will serve as input to detailed analysis in Deliverable 3.

Examples:
- Existing screens, documents, documents, database definitions.
- SILC Appendix A, Deliverable 2

Output from JAD sessions:
- Bulleted statements of fact
- Diagrams
- Concrete examples

Questions answered from the analysis
- What sources of information was used for the analysis?
- What type of information is to be included in the system?
- Can you provide me an example of what you mean?
- What are some of the information objects of the system?
- About how big will the database be?

Deliverable 3: Conceptual and Logical Information Model

Description

Once a set of significant information is assembled from Deliverable 2, the generalized sets of information must be examined for complete understanding including any limitations on the populations (constraints), any rules for modification, and proper identification schemes. Checks against the initial problem statement to test completeness must be performed. Validation of the understanding must be provided by customers, users, or subject experts. Traceability to the items collected in Deliverable 2 must be maintained. Upon completion of this analysis, a complete information model exists including: all data constructs necessary to satisfy the customer's needs of the system, all of the rules/constraints relevant to the population, modification, or deletion of information from the system, concrete examples (both good and bad) of information contained in the model, verbalizations of all information contained in the model. Multiple iterations (versions) of the model will be required to achieve the final level of completeness. Implementation specific structures (i.e. surrogate keys, aliases, and computational tables) are not included at this time.

Purpose

A completed information model has two primary purposes:

- Provide a clear, unambiguous representation of the information relevant to the problem area which can be easily understood, verified, and validated by the customer, users, and subject experts.
- Provide formal non-vendor related specifications for the implementation of these constructs into database management systems

Examples:

- SILC Appendix A, Deliverable 3
- IDEF1X, ER Diagram, Object Diagram

Questions answered from the analysis

- How is any object identified? Attributes identified?
- What is required to create any new object? Modify it? Delete it?
- Can you give me an example of that rule?
- What did the user provide you in order to create that object?

Deliverable 4: Software Verification and Validation Plan

Description

The software verification and validation plan specifies the tasks and resources required for adequate evaluation and testing of a new or enhanced software application.

Verification involves checking that products in each life cycle phase:
- adhere to prior life cycle phase requirements,
- adhere to standards, practices and conventions for that phase, and
- form an adequate foundation for the next life cycle phase.

Validation consists of checking that the software system adheres to system and software requirements.

Purpose

The primary purpose of the verification and validation plan is to consider and document how the quality of a new or enhanced software application will be proven. Depending on the size and criticality of the application, various levels of evaluation and testing can occur. The verification and validation plan specifies the level of effort required and plans for an appropriate set of resources to complete the tasks.

Example
- SILC Appendix A, Deliverable 4

Questions answered:
- What is the scope of the verification and validation effort?
- What assumptions are being made about the product/process?
- What resources, roles, and responsibilities exist in the plan?
- What evaluation techniques/tools are used on each deliverable?
- What techniques will be used for traceability among deliverables?
- How will issues/problems be documented and resolved?
- What method of software configuration management is used?

Deliverable 5: Elementary Processes

Description

Elementary processes represent the smallest units of work which can be executed to modify the content of the information base and which will leave the information base in a state consistent with the prescribed integrity rules. At this time, we are only concerned with the minimum amount of work required to create new objects defined in the information model (deliverable 3).

Purpose

The purpose of this activity is to clearly understand the dependencies (constraints) of the objects defined in the information model and to identify the sequence of actions (processes) necessary to create new instances of these objects. Suggested grouping of objects and related facts will be uncovered leading to screen content for the entry of new information.

Examples
- SAND95-2013 "Systems Requirements Specification for Data Accumulation and Analysis Information Management System", section 3.5.
- SILC Appendix A, Deliverable 5.

Questions answered
- What are the minimum set of processes required to create any new object instance?
- Which objects should be considered for grouping?
- Which facts are provided by the user? system?

Deliverable 6: Conceptual Business Functions User Model

Description

When developing a new information system, it is important to understand the work functions of the user and how they will interact with the information system during the course of their work processes. This activity attempts to identify those work processes in which the computer may provide either a complete or partial role. The work processes, who performs them (actors), and the beginning and ending state of the database is described.

Purpose

The purpose of Deliverable 6 is to:

- help define system requirements and functions,
- better understand the expected interactions between users and the computer,
- capture knowledge about the system for managers, users, and future maintainers,
- provide a systematic, structured framework for process analysis, and
- assure that the system remains in the scope of the initial problem statement.

Examples:
- Use Case Diagrams.
- SILC Appendix A, Deliverable 6.
- AT&T Process/Actor diagrams.
- Cartoons depicting the grouping of information on screens, documents, etc. (optional).
- Storyboards showing information flow.

Questions answered from the analysis
- Who will be using the system?
- Which processes will be supported by the system? Manually?
- Which roles will be played by which people?
- Who will create/update the information?
- Who needs documents/queries from the system? Why?

Deliverable 7: Implementation Strategy

Description

The Implementation Strategy is a plan to cover the physical operational proposals. It should list hardware or specific computer system to be used, and proposed software needed.

The systems and technical strategies will be described by defining architectures with brief reviews of alternatives considered. Technical discussions should address performance, security needs, and strategies. A Network Plan should be drawn in as much detail as is needed. Deployment plans should be discussed.

Purpose

To document recommended systems strategy architectures with brief review of alternatives considered to define recommended organizational strategy and action plan.

Examples:
- SILC Appendix A, Deliverable 7

Questions answered from the analysis
- What are the projected throughput and availability requirements?
- What are the storage requirements for new data collected?
- Integration requirements?
- Response requirements?
- Security requirements?
- Location requirements?
- Proposed hardware?
- Proposed software?
- Proposed network?
- Proposed architecture?
- Who are the experts for each part of the architecture?
- Which computers are involved? A list of names and IP addresses?
- What is the sequence of communications between computers?

Deliverable 8: Implementable Business Functions (Presentation Model)

Description

Extended Presentations Sets consist of screen and document layouts in template form which incorporate vendor specific and implementation specific features. Conformance to any viewing/presentation standards should be provided in these layouts. They are often created and managed electronically. The Conceptual Business Functions User Model (Deliverable 6) provides the guidance for screen content and functionality. The screens should be capable of demonstrating menu and screen connectivity actions at this time. It is not expected that database access or extensive coding exist however. The screens and menus will evolve and iterate throughout the remainder of the project as the system is assembled.

Purpose

The purpose of the Implementable Business Functions Model is to provide the developer with the final concepts of the screens and menus intended to be developed in the presentation. The behavior of the screens/menus can be validated as well as validation with the current version of the information model (deliverable 3). Any editing or constraint enforcement at the presentation level can be verified at this time.

Examples:
- Systems Requirements Specification for Data Accumulation and Analysis Information System, SAND95-2013, Extended Presentation Sets
- SILC Appendix A, Deliverable 8

Questions answered from the analysis
- What will the system screens look like?
- How do the screens behave?
- Will they satisfy the user's business requirements?
- Are any constraints enforced in the presentation layer?
- Which ones?

Deliverable 9: Database Schema

Description

The physical database design, using the constructs of a specific vendor software product is presented in the form required for implementation: schema Data Definition Language (DDL) for relational systems, Class Type definitions for Object database, File definitions for non-database. Any extensions to the database definition, i.e. triggers, database procedures, views, etc. should be included in this specification. Full traceability back to the source information model must be demonstrated. In diagram form, this represents the latest version of Logical Information Model - Deliverable 4.

Purpose

The purpose of this task is to create a version of the database definition to be implemented for the application. This deliverable is the primary source of communication and requirements specification with the Database Administrator (DBA). Full traceability to the physical implementation must be maintained. Also, the specifications must be able to be tested for proper implementation.

Examples:
- SQL/DDL with associated trigger and/or database procedures.
- Class Type definition code.
- SILC Appendix A, Deliverable 9.

Questions answered from the analysis
- What is the implemented (physical) structure?
- How does this relate to the initial design (model)?
- Which DBMS will be used?

Deliverable 10 - deleted

Deliverable 11: Implementable Business Functions (Process or Function Model)

Description

The process model describes the system's functionality from a "non-user" perspective. (The user perspective was described by Deliverables 6 and 8.) These system functions may be specific algorithms, batch processes, operational functions, etc. Formal testing should be conducted on these detailed specifications.

Purpose

When there are complex batch processes, interfaces to Commercial Off-the Shelf (COTS) software, and complex quality checking procedures or security controls, these processes will be defined in detail in the process model. All processes should be tied to the event that occurred in the systems environment that caused that function to be needed.

Examples:
- IDEF0 Decomposition
- Decision Tables
- Event Trace, Event Flow Diagrams, and Event Process Tables
- Pseudocode, natural language, or structured text
- Mathematical equations
- SILC Appendix A, Deliverable 11.

Questions answered from the analysis
- What happens in each batch process?
- What must operational security processes do?
- What must operational quality processes do?
- What detailed algorithms are required?
- What complex decisions determine the process flow?
- What is the detailed specification for the functions performed after each event takes place?

Deliverable 12: Test Plan, Test Database, and Test Data

Description

Understanding of how the system will be tested is described in the project test plan. Four basic areas of testing include 1) All of the structures, relationships, derivations, and constraints implemented from the data model - usually associated with a Data Base Management System (DBMS), 2) Any code (database procedures, standard functions, etc.) not a part of the database definition language (DDL), 3) Items involved in the presentation layer - screens, menus, behavior, 4) Network/communications testing. Additional types of testing should be conducted on an as need basis such as volume testing and security testing. The Sandia Test Bed is primarily oriented towards test types 3 and 4. The development team is responsible for all test types, but primarily 1 and 2.

Purpose

Provide a test plan addressing the approach and processes used to conduct testing on at least the four major areas of the system. Provide a test database populated with valid examples (gathered from the modeling process) to serve a testing resource. Provide concrete examples/scenarios demonstrating each constraint identified in the analysis. Constraints such as mandatory could be grouped together and associated with appropriate schema expressions (i.e., not null). We are not attempting to test that the 'not null' expressions work in the DBMS, but rather verify that they are there.

Examples:
- System Test Plan describing the areas of testing
- SILC Appendix A, Deliverable 12

Questions answered from the analysis
- How will the system be tested?
- Who will be doing which tests?
- Where is the test database?
- How will you record/manage test results?
- Is any volume testing required?

Deliverable 13 :- deleted

Deliverable 14: External Interfaces

Description

Most systems rely on data and controls from other systems, either by direct network links or file transfer. The interfaces must be documented to provide for ongoing maintenance of the system. If data must be converted in the process, a conversion map showing the input, output and transformation process for each piece of data (fact) must be created.

Purpose

Clearly define the interfaces required between systems (old and new). Provide a detailed mapping of existing data to new data. Verify the dependencies the project has on other systems, both long and short term.

Examples:
- External and Internal Interfaces
- SILC Appendix A, Deliverable 14

Questions answered from the analysis
- Which systems provide data to your system?
- Describe the data you must provide to system X?
- What are you going to do with the data in the transfer?
- Is this a one time mapping or periodic?
- Are you going to use the network, tape, foot net?

Deliverable 15: List of Reused Elements

Description

Sharable routines such as functions and methods should be documented. The version of the shared software should be captured.

Purpose

Proper software configuration management of software.

Examples:
- No current examples exist
- SILC Appendix A, Deliverable 15

Questions answered from the analysis
- Which shared functions do you use?
- What version are they?
- How will you conduct change control?

Deliverable 16: User Help Documentation

Description

The enduring system documentation for ongoing operation and support of the software must be released and controlled throughout the life of the system.

Purpose

To provide clear, concise, accurate and up to date documentation for the long term support and operation of the software.

Examples:

- Systems Requirements Specification for Data Accumulation and Analysis Information System, SAND95-2013, Extended Presentation Sets

Questions answered from the analysis

Users Guide

- What will the system screens look like?
- How do the screens behave?
- Will they satisfy the user's business requirements?
- Are any constraints enforced in the presentation layer? Which ones?

Users Support

- Who will be using the system?
- What is their access path?
- What network components are being used?

Deliverable 17: Test Results

Description

Test results of each of the test areas will result in documents such as the Test Incident Report. These are the type of documents required by the test bed to verify that testing has been completed

Purpose

Verify the adherence of the finished system to the customer requirements. Assess the quality of the software created. Verify compliance with all known constraints. Verify derivations. Evaluate the system's "user friendliness". Verify conformance to any standards.

Examples:
- Test Incident Document
- SILC Appendix A, Deliverable 17.

Questions answered from the analysis
- How many bugs were uncovered during testing?
- Where are the test results?
- Who tested which piece of software?
- Are you releasing the software with any known bugs?

Appendix C

Integrated Information Systems (IIS) Design Review Process Guidelines

IIS System Design Review Process Guidelines have been written to assist the project leader and IIS stakeholders with design reviews of application development/enhancements and commercial software projects. Project planning includes the scheduling of the appropriate project reviews as negotiated with the utility service managers. Four possible outcomes from negotiating are:

1. a full review,

2. a partial review schedule eliminating or modifying one or more review phases,

3. and E-mail Design review for low risk applications as described in the section, entitled IIS Design Review Process for Low Risk Applications, or

4. a design review is not required. Also at a planning meeting, the project leader commits to the software life cycle deliverables and their formats from the IIS Software and Information Life Cycle (SILC) Issue C. Refer to Appendix B for a one-page description of the SILC deliverables. Results of the planning meeting will be sent to the utility service managers and IIS stakeholders.

The design review will:

- Provide a formal or E-mail process for the evaluation, approval and documentation of project milestones throughout the development life cycle.
- Provide metrics and objectives to help the project meet performance, cost and schedule requirements.

- Prepare a forum to double-check that customer requirements have been validated, and that the design meets functionality requirements.
- Prepare a forum to validate that IIS stakeholder requirements/constraints have been addressed.

Analysis/Planning: IIS Design Review Options

All corporate application development/enhancement and commercial software projects use the IIS design review process. A diagram and flow of the process is shown in Figure C.1.2

The IIS Design Review options are:

1. If a full review is required, all design review phases must be completed which are described in this document. Included in this requirement are the SILC deliverables which are determined in this planning meeting. The SILC deliverables are described in Appendix A and B.

2. A partial review is required by eliminating or modifying one or more review phases. If this occurs, the project leader will agree to the design review phases which are performed and the SILC processes which are completed. These commitments are incorporated into the project plan.

3. For low risk applications, the design review phases will be done via e-mail. Refer to IIS Design Review Process for Low Risk Applications for a copy and example of this process. Low risk applications will continue to follow the IIS Design guidelines, but do not require SILC deliverables. The project leader will be required to provide a project proposal or project plan.

4. The utility service managers and project leader can agree that a review is not required.

NOTE: For options 1 and 2, not all SILC deliverables in Appendix B are required. The deliverables must be negotiated among the customer, utility service managers, and the project leader. **The project leader should schedule and communicate the review phases in advance so the appropriate individuals will be present at the reviews. If the appropriate individuals cannot be present, they should send a representative.**

Conceptual Design Review (CDR) Guidelines

Objective Of Conceptual Design Review:

Both the project team and the IIS stakeholders should view this Design Review process as a teaming effort with the customer to baseline the project system requirements and design; thus, any new customer requests would be queued as new project and funding prioritization decisions.

In addition to providing a customer/supplier baseline for the analysis phase and documentation, the CDR serves as a forum to obtain written approval from the customer, the utility service managers, and the IIS stakeholders to continue with the project activities as planned or to identify alternative steps.

Before the CDR, project team members meet with the IIS stakeholders listed in Table C.1.1 and review their requirements/constraints. Team members should be prepared to show that all requirements/constraints have been addressed. Also refer to the top section of the Implementation Checklist in the SILC document for early life cycle requirements.

The purpose of the CDR is to ensure that the project is ready to move from the requirements phase to the design phase. The project team and IIS stakeholders need to make sure that the customer's and stakeholder(s) are clearly defined and that the project requirements are understood. The review team and project leader will agree on the requirements and design documentation. There are three possible outcomes from the review:

1. completely ready to proceed,

2. not ready to proceed,

3. can proceed upon completion of identified action items.

Conceptual Design Review Steps.

The CDR includes key customers, utility service managers, and IIS stakeholders (refer to Table C.1.1 for stakeholders). The focus will be on the customer requirements and IIS stakeholder issues.

1. Project leader schedules a review with key customers (customers who can approve requirements), the utility service managers, and appropriate

IIS stakeholders (see below list). Refer to the CDR outline Figure C.1.1 for review presentation.

2. Provide an overview of the project i.e., objectives, phases, and deliverables.

3. Identify and discuss customers and IIS stakeholders.

4. Review validated (describe how validation was done) customer requirements and obtain approval from customer Utility Service Managers.

5. Review Validation and Verification Plan.

6. Review project plan (performance, cost and schedule) for full support of the validated customer requirements.

7. Identify how the requirements and design are being documented and where the documentation is maintained.

8. Review the outcome of meetings with the utility service managers and IIS stakeholders. Show that you have satisfied their requirements/constraints/guidelines.

Table C.1.1

Application Utility	**Computing Utility**
- Design Reviews	- Helpdesk
- Applications	- Computer Security
- Testbed, Human Factors	- Systems Administration
- Training	- Database Administration
- C/S Prod Environment	
Desktop Utility	**Data Tone Utility**
- CSUs	- Communications/Network
Dial Tone Utility	
- Phone/SVIS	

9. Review action items from this review and prepare to show corrections in the Detailed Design Review. Also update project documentation.

10. Obtain Conceptual Design Review approvals. Refer to Figure C.1.3 for copy of approval sheet.

Figure C.1.1 Conceptual Design Review Outline

Conceptual Design Review Outline

I. Overview.

II. Identify the Customers.

III. Project Scope

A. Identify the project scope.
A one paragraph description of the project. Refer to SILC Deliverable 1. Identify the statements/issues that limit the scope of the effort (e.g., limited time, limited dollars, and limited utility, etc.).

IV. Customer Requirements

A. What analysis technique was used to determine the customer functional and information requirements? (the "what" of the problem/project not the "how").

B. What assumptions were made in developing the requirements?

C. Describe the results of the analysis for the customer functional and information requirements for the project (e.g., what functions or processes will this effort provide to the customer).

D. Identify and show how the customer requirements were/will be validated. How will we help to ensure that the delivered products will satisfy the customer requirements? Refer to SILC for deliverables 2, 3, 4, 5, & 6.

E. Identify how the requirements and design are being documented and where maintained.

F. How were the IIS Stakeholder requirements/constraints accommodated/satisfied? Refer to Table C.1.1

V. Design Methodologies

A. Identify the design method/s used to develop the functional design.

B. Describe what alternate designs were produced and the design decisions for design choice.

Detailed Design Review (DDR) Guidelines

Objective Of Detailed Design Review:

The DDR serves as a forum to obtain written approval from the customer, utility service managers, and the IIS Stakeholders to continue with the project activities as planned or to identify alternative steps.

The detailed design review allows the project leader to show examples of the functionality which will satisfy the customer requirements. For application development and commercial software, the examples should satisfy presentation, function, and data layers.

DDR Steps:

1. Project leader schedules the review with appropriate key customers, utility service managers, and appropriate IIS stakeholders.

2. Provide an overview of what will be presented.

3. Review corrections to action items from the CDR review.

4. Review status to development project plan (performance, cost and schedule)

5. Review conformance to design specifications.

6. Presentation layer: Show objects for presentation: actual screens, menus, buttons, and pick lists. Use HTML, Power Builder, CASE software, etc. for examples. Refer to SILC Deliverable 8. **Presentations are reviewed by Human Factors before the DDR, and an approval from this organization will be required to continue with the project.** This approval step should eliminate screen rework which occurs from testing in the Testbed.

7. Function layer: Provide a list of the functions which will support the presentation layer and a list describing all databases accessed by each function. Discuss specific use of data by functional layer. Refer to SILC Deliverable 15.

8. Data layer: Present an overview of the schema tables. Also present the implementation information model. Include implementation specific

facts (i.e. surrogate keys, internal objects, etc) used to optimize physical implementation. Use SILC Deliverables 7 and 9.

9. Review test plans. Refer to related SILC Deliverables.

10. Discuss Risks and Issues - Elaborate on problems that have occurred and their impact.

11. Provide the status of required documentation: Production services, help desk, training, security, installation and deployment.

12. Review action items from this review and plan to show the corrections during the Final Design Review.

13. Obtain Detailed Design Review approval.

NOTE: Steps 5, 6 and 7 only apply to application development and commercial software.

Infrastructure Testbed

Objective Of Infrastructure Test bed:

To obtain written certification that when the software is deployed, it will satisfy all the infrastructure requirements/constraints. Refer to "Appendix E", URL 20, for details.

The project leader secures certification from the testbed and has the IIS Design Review Process Approval Sheet signed to continue in the life cycle process

Final Design Review (FDR) Guidelines

Objective Of Final Design Review:

The FDR serves as a forum to obtain written approval from the customer, the utility service managers, and the IIS stakeholders to continue with the project activities as planned or identify alternative steps.

This review is the final opportunity to review the deliverables prior to deployment in a production status. The review will assure that:

- All testbed anomalies have been resolved.
- All IIS stakeholder issues have been resolved.
- A deployment plan has been created and funded and resources have been identified for the maintenance and support of the application.

FDR Steps:

1. Project leader schedules the review with appropriate customers, utility service managers, and appropriate IIS stakeholders.
2. Provide an overview of what will be presented.
3. Review corrections to action items from the DDR review.
4. Review status to development project plan (performance, cost and schedule)
5. Review conformance to design specifications.
6. Show that the implementation checklist has been completed.
7. Show that the documentation process has been completed. Refer to SILC Deliverable 16.
8. Review test results. Refer to SILC Deliverable 17.
9. Obtain FDR approval. Also update project documentation.

Production Sign-Off & Deployment

This is the final phase of the software design review process, but it will not require a forum to obtain the final approvals on the IIS Design Review

Process Approval sheet. The project leader must secure sign-offs from the following areas:

Production Environment/Operations:
All Production review phases have been completed, approved, and that all documentation has been updated (see computing Production Environment on helpdesk Web homepage).

Deployment:
A plan has been developed to fund resources and documentation for the targeted customers.

Applications Maintenance and Support:
Resources, documentation, and funding have been identified and the operation phase have been addressed and agreed on.

Utility Service Managers:
All review phases have been completed, approved, and that all documentation has been updated.

Figure C.1.2 Design Review Process

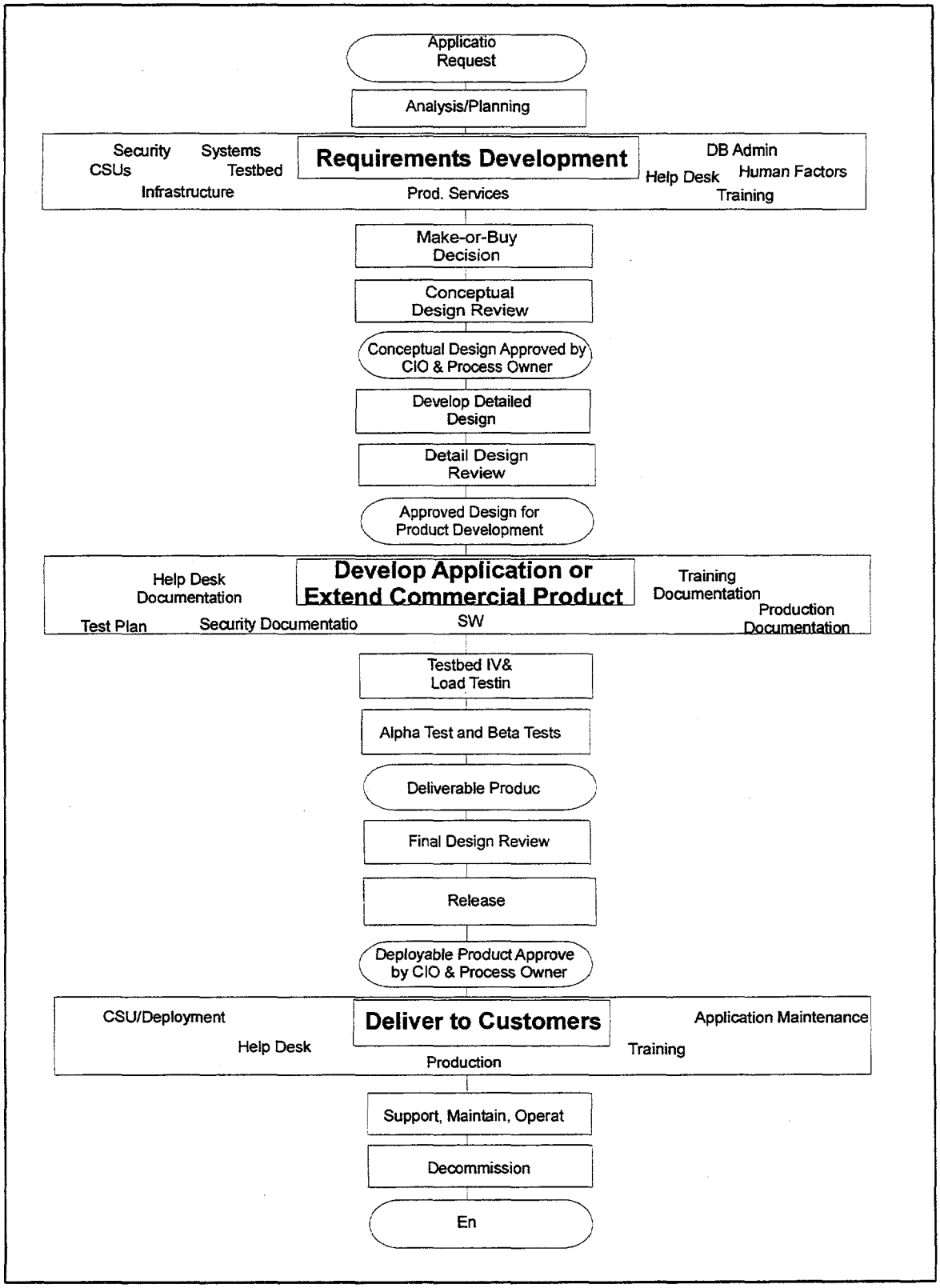


Figure C.1.3 IIS Design Review Process Approval Sheet

IIS DESIGN REVIEW PROCESS APPROVAL SHEET

Project: _____

Project Leader: _____

Conceptual Design Review (CDR) Approval/Date:

Process Owner _____

IIS Application Utility Service Manager Representative _____

IIS Computing Utility Service Manager Representative _____

IIS Desktop Utility Service Manager Representative _____

IIS Data Tone Utility Service Manager Representative _____

Detailed Design Review (DDR) Approval/Date:

Process Owner _____

IIS Application Utility Service Manager Representative _____

IIS Computing Utility Service Manager Representative _____

IIS Desktop Utility Service Manager Representative _____

IIS Data Tone Utility Service Manager Representative _____

Infrastructure Testbed/Date:

Certification Representative _____

IIS Application Utility Service Manager Representative _____

IIS Computing Utility Service Manager Representative _____

IIS Desktop Utility Service Manager Representative _____

IIS Data Tone Utility Service Manager Representative _____

Final Design Review (FDR) Approval/Date:

Process Owner _____

IIS Application Utility Service Manager Representative _____

IIS Computing Utility Service Manager Representative _____

IIS Desktop Utility Service Manager Representative _____

IIS Data Tone Utility Service Manager Representative _____

Production Sign-off & Deployment Approval/Date:

IIS Application Utility Service Manager Representative _____

IIS Computing Utility Service Manager Representative _____

IIS Desktop Utility Service Manager Representative _____

IIS Data Tone Utility Service Manager Representative _____

IIS Design Review Process for Low Risk Applications

Introduction

The following pages discuss the process for the Design Review and approval of an application which is deemed low risk according to the definition below. The following organizations may be submitting these applications:

Business Systems Development
Webco

Currently, there is a formal IIS Design review process which requires formal reviews, and approval of the development and /or enhancements for CIO corporate applications. This review process has been very beneficial with the identification and resolution of issues for projects which are high risk. For applications which are low risk (refer below for definitions), the process has not been cost-effective for the IIS utility service managers, IIS stakeholders, and the developer. To provide customer satisfaction, efficient use of IIS stakeholder resources and still meet the IIS Design Review Process guidelines, the below process will be used for low risk applications. Low risk applications will still be following the IIS Design Guidelines.

Using e-mail (until a Workflow solution is implemented) the customer, IIS utility service managers, and IIS stakeholders will be able to review, comment and approve or disapprove the customer funded project.

Low Risk Requirements:

- Software distribution to the desktop is not required.
- The application does not update official corporate data, query only on official corporate data.
- No business rules are enforced.
- The application will operate within the existing infrastructure.
- The system is less than $75,000.
- The project does not exceed the stakeholders planned budget.
- The project is not a high visibility project (political).
- The application is unclassified.

Process

Integrated Information Systems (IIS) Design Review Process Phases*

Phase I, II, III: Planning, Conceptual Design Review (CDR), Detailed Design Review (DDR) via e-mail to stakeholders
Stakeholder comments reviewed and approved

Phase IV: Unit Test & Integration Test (Code & Test), Done by developer (requires approval from IIS utility service managers)

Phase V: Testbed

Alpha Test (Use CIO staff) - In conjunction with Test Bed

Beta Test (User testing) - In conjunction with Test Bed

Phase VI: Final Design Review (FDR) - via e-mail or if necessary a meeting will be held

Phase VII: Production Deployment & Sign-Off

Phase VIII: Application Maintenance & Core Support

* All process phases should have process owner approval.

The process will be conducted via e-mail until a workflow solution is implemented. The planning meeting, the conceptual design review and the detailed design review will be combined and will be conducted via e-mail. A project plan for the application will be sent out to all IIS utility service managers and stakeholders. A reasonable length of time (minimum of one week), will be given to each stakeholder to review the project plan and provide input to the developer. If a developer does not hear from a stakeholder an attempt to contact the stakeholder via phone will be done. Ultimately, the stakeholder is responsible for responding.

If a major issue is identified by a stakeholder then a meeting will be held to resolve the issue.

The developer will incorporate the feedback that is received into their project plan and resolve conflicting issues. This section will contain the date of the review. The developer will then conduct a short meeting with one of the IIS utility service managers for certification to proceed with development.

The developer will then code and test the application. The application will then be submitted to the test bed. The developer and the Testbed stakeholders will negotiate any suggested changes and/or issues.

Once the Testbed has certified the application, the developer will send out an updated copy of the project plan for the final design review via e-mail. The stakeholders will have a reasonable amount of time (minimum of one week) to provide input. If there are no major issues identified, the developer will submit the application to production. If there are major issues a meeting will be held to resolve the issues.

If the application is a Web application that requires linkage from a corporate home page, the application will be submitted to the Webco manager. If the application requires the LIS or production services for deployment the application will be submitted to production services and the appropriate CSU.

Table C.1.2 Roles and Responsibilities

Model for Roles and Responsibilities:

Using e-mail, the roles and responsibilities for the IIS Utility Service Managers and stakeholders are:

Application Utility

Applications Support

The applications support stakeholder reviews low risk applications and provide feedback during the review sessions to the developer in a timely manner.

For corporate applications, applications support provides on-going support if the application breaks.

Human Factors

The human factors stakeholder reviews low risk applications and provide feedback during the review sessions to the developer in a timely manner. Human factors personnel will provide guidance on usability issues.

Testbed

The Testbed stakeholder reviews low risk applications and provide feedback during the review sessions to the developer in a timely manner.

The Testbed accepts low risk applications without formal signature for the design reviews, and will certify these applications in the usual fashion.

Training

The training stakeholder reviews low risk applications and provide feedback during the review sessions to the developer in a timely manner.

Data Tone Utility

Communications/Networking

The communications/marketing stakeholder assesses the need to advertise and communicate the product and provide feedback to the developer in a timely manner.

Computing Utility

Corporate Computing Help Desk (CCHD)

The CCHD stakeholder reviews low risk applications and provide feedback during the review sessions to the developer in a timely manner. The CCHD will provide the first line of support for applications.

Database Administration

The database stakeholder reviews low risk applications and provide feedback during the review sessions to the developer in a timely manner.

Production services

The production services stakeholder reviews low risk applications and provide feedback during the review sessions to the developer in a timely manner.

Security

The security stakeholder reviews low risk applications and provide feedback during the review sessions to the developer in a timely manner.

Systems Administration

The systems stakeholder reviews low risk applications and provide feedback during the review sessions to the developer in a timely manner.

Desktop Utility

Customer Service Units (CSUs)

The CSU stakeholder reviews low risk applications and provide feedback during the review sessions to the developer in a timely manner.

Developer

The developer satisfies the IIS Design Review Process for low risk applications. In addition, the developer will be responsible for negotiating with each stakeholder and the utility managers. The following are guidelines for a developer with stakeholders:

Data Tone Utility

Communications/Networking

If necessary, identify network performance requirements to the Communications/Network stakeholders.

Computing Utility

CCHD

The developer negotiates a service level agreement with the CCHD. The developer will provide the required amount of training and/or documentation that CCHD needs.

The developer notifies CCHD when the application goes into production. A deployment plan will be developed that will notify all relevant stakeholders when the application has been deployed.

Database Administration

For development efforts the developer consults with the DBAs to determine the amount of time that they will expend on the project and build their costs into the estimate for the customer.

Production Services

The developer complies with production requirements listed on the Web and submits applications to the Production Services Representative for review and approval.

Security

The developer works with the sponsor to determine the classification of the data. The classifications are unclassified/classified and limited/unlimited information. If the data does not meet the above criteria, the developer will work with security personnel and the process owner to handle the sponsor's needs and incorporate a cost and time estimate for security personnel into their project plan.

Systems Administration

The developer provides an estimate of the number of potential users and amount of usage, so systems can do capacity planning. If the application requires backup and recovery services the developer will notify systems personnel as soon as possible.

Desktop Utility

CSUs

If the application requires CSU support a service agreement will be negotiated.

Application Utility

Human Factors

During the planning phase, the developer reviews their screen designs with a member of the Human Factors team.

Testbed

The developer submits their application to the Testbed for evaluation. Any changes suggested by the Testbed will be negotiated between the developer and the Testbed stakeholders.

Training

The developer will consult with the training group as needed for their application.

Appendix D

Objectives of SILC Project Planning

The development of the SILC Plan Set supports three primary objectives identified below.

Table D.1.1 SILC Plan Set

Objective	Approach	SILC support
Minimize overhead while increasing information value	reduce paper-based documents while increasing the early use of software-based models, repositories, and automated tools	link planning processes and deliverables to subsequent SILC processes
Emphasize the capture and development of early test requirements and criteria	software-based tools and repositories, early emphasis on Design Review information, Software configuration management Plans, and Quality Assurance Plans	consider references to standards, business needs, legal and security requirements, and expected outcomes were a priority.
Size and set boundaries for labor, costs, and schedule	executive sponsorship, business case development, metrics considerations, and linking the project to business goals	establish performance measures, a metrics "dashboard", high-level deliverables, and business case

Scope of this Plan

The project plan is intended to supplement the front-end activities of the SILC process. Efforts that are candidates for using this planning instrument include new development projects and significant enhancement projects.

Where to start

Planning Phase Steps—Corporate Level

Preferably members of top-level management participate in enterprise information systems planning to develop critical success factors for the business:

1. The enterprise-level planners create association matrices that correlate organizations, data, and processes.

2. The participants identify and resolve matrix inconsistencies.

3. The participants produce a prioritized list of information development efforts. As process owners they often propose a set of solutions, from which one is chosen to be studied and developed into a system proposal by software engineers.

4. The participants produce an organizational structure established on information flows and usage.

Planning Phase Steps—Project Level

- Document and summarize the project, its background, earlier attempts, relevant pre-proposal dialogues and proposal progress. This information will contribute to the project plan introduction. Key pieces of meta data include the naming of the project, its sponsor(s), funding source(s), key customers, anticipated duration, key contributors, author, and purpose. Optionally include relevant project history and its potential impact on the success of the project.
- Develop a high-level business case using one or more of the following financial analysis tools. Consider the expenditures and savings on people, process, and technology.
  - Life cycle-based cost/benefit analysis to forecast the expected tangible costs and savings of the project over its entire life cycle as opposed to merely the development life cycle. Understandably, the further out the dollars are projected, the less confidence can be applied to the totals.
  - ROI (return on investment) to forecast the expected percent of return (dollars saved) over costs (dollars expended) for the expected life of the system. The ROI is easily derived from the totals associated with the cost/benefit analysis.
  - Payback period to determine how much time will pass before the system has paid for itself through accumulated savings. Payback can be stated in terms of number of months or years and can also be derived from the cost/benefit analysis.
- Perform the feasibility assessment. Using the following matrix as a guide, consider and document the technical, operational, and economic feasibility. The preceding business case will aid in the economic feasibility. Subgroupings by assumptions, constraints, and risks will likely add clarity to the feasibility assessment.

Table D.1.2 Feasibility Assessment Matrix

	technical	operational	economic
assumptions (will help)			
constraints (will hinder)			
risks (may hinder) (hardware & network framework - likelihood of success)			

- Determine the level of software configuration management required for the project considering size, rigor, and resources. A deliverable, the software configuration management plan, is documented in more detail below and in the references at the end of this section.

Approaches to consider and incorporate for project planning

Each of the following approaches and techniques for gathering information are documented in numerous sources, many in the references that follow this section. Their placement here is intended to serve only as a reminder, not a tutorial.

- Interviews, surveys, polling, and documentation review are data gathering techniques that are applied to collect initial information about the project. Each technique has both strengths and weaknesses in respect to time, consistency of data collected, and breadth of information gathering. Early and sustained contact with the project sponsor and stakeholders is important.
- JAD (Joint Application Design or Development) is a group decision-making process, guided by a trained facilitator, for any phase of the information life cycle including project plan planning. JADs are especially useful when groups of less than 25 are involved in the process and have an interest in understanding the perspectives and requirements of other participants. JADs are not as valuable when decisions have already committed resources, schedule, and product.
- Automated model validation offers the following advantages: it enforces consistency and completeness checks, it shifts the validation earlier in the life cycle identifying problems sooner where they are less costly to repair, it is less costly to perform than the peer review, and it requires the use of a repository where models can be reused.

What to deliver - Plan Set Components

> ***NOTE:*** some projects may include the project proposal as part of the project plan, whereas others may be expected to produce an initial proposal which when accepted, signals the start of the formal project planning phase.

The following components comprise the project plan. An example may be found in the Appendix A, Deliverable 0.

- Project metadata such as author; organization; brief overview (usually between a paragraph and a page in length); key project sponsors, stakeholders, and customers; sign-offs; issue date; comments and earlier revisions if applicable.

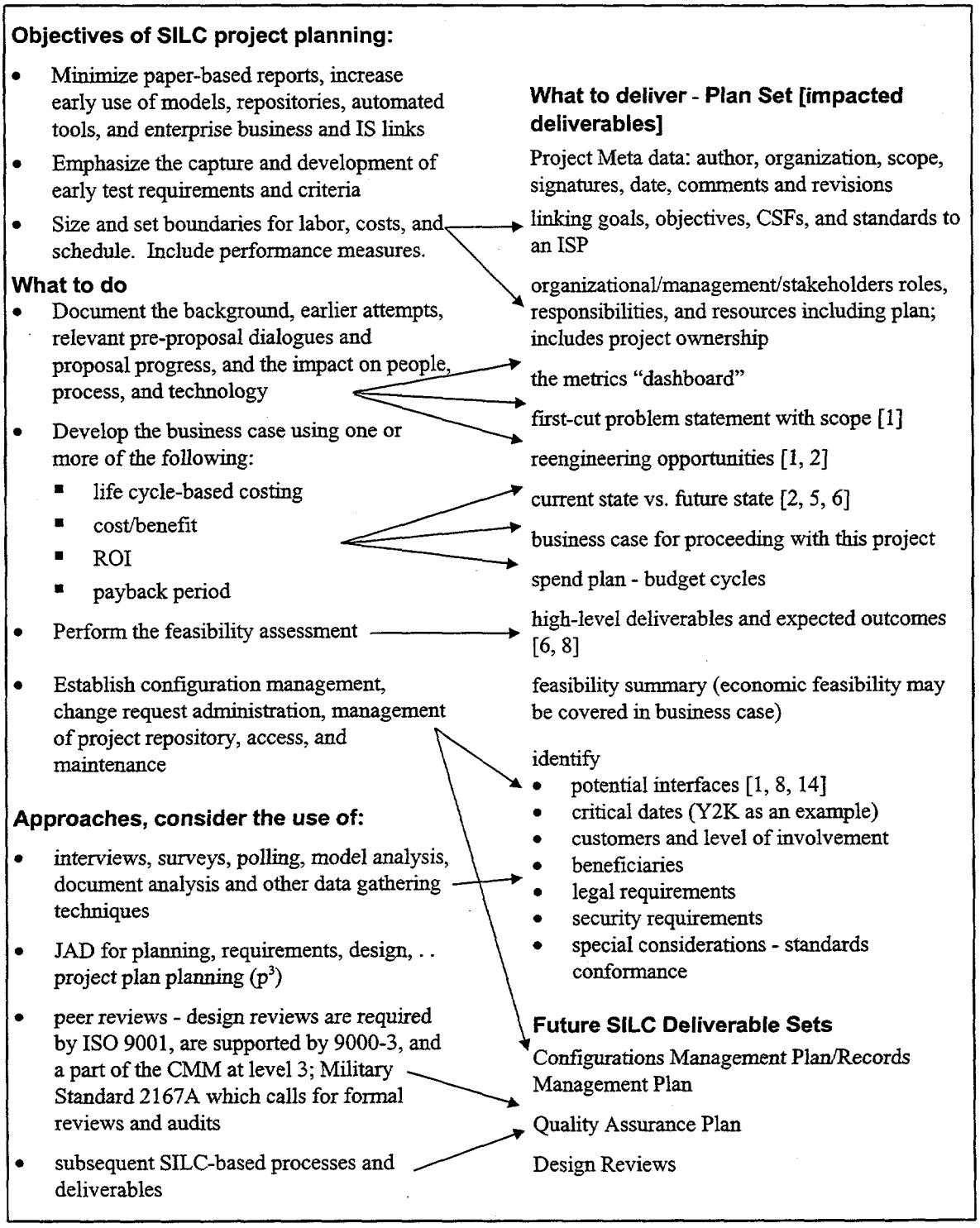
- Relationships among goals, objectives, critical success factors, and standards to an Information Systems Plan where one exists or other similar corporate or organization plans such as an Enterprise Information Plan. Specify which specific higher level planning instruments, and their subcomponents, are enacted or enabled by the project.
- Roles, responsibilities, and resources for organizations, management, project team, and stakeholders including project plan and its ownership.
- Document specific expectations, openly or carefully detailed, for the contributors and participants in the project. Where multiple people resources are allocated to a variety of responsibilities, an association matrix can often be used to simplify and summarize this information.
- The metrics "dashboard". Identify key metrics that portray the progress of the project using business-centered metrics. Describe business-centric metrics where, how, and by whom they are maintained for the life of the project.
- First-cut problem statement and scope. (These planning components will later feed a potentially more detailed problem statement in the analysis stage.) State the problem in terms of the business needs, not a technology that is expected to solve the problem. Scope minimally pinpoints the business process that triggered the proposed system and describes the information or processes that are outside the 'boundary" of the system or are terminated. Good example: Weapon design data is captured and maintained in multiple databases that add unnecessary costs to the operation. Bad example: Weapon design data needs multi-tiered applications that share interfaces and objects to reduce costs.
- Reengineering opportunities (these may emerge from the problem statement, but they may also contribute to a refined problem statement as the project progresses.) Evaluate considering deliverable 2 Information Business fats and Source Documents to assess how much effort needs to be expended to redevelop the underlying business processes.
- Current state vs. future state comparison. Compare how the business environment will change using a "snapshot" of the current state with the expected future state. These differences may be useful for building a business case for funding, or sustaining the funding of, the project.
- Business case for proceeding with this project. Articulate the expected economic impact of the project. Quantify as many intangible costs and benefits as possible. Minimize the use of intangible benefits to justify the project.
- Budget cycles. Identify the pertinent budget cycles to which the project is subject. Ensure adequate funding is available (or forthcoming) for successfully launching and sustaining the project.

- High-level deliverables and expected outcomes. List the high-level products that are expected to result from the project. Highlight major expected changes in the way business is conducted, newly implemented processes, and those that may be eliminated.

Also, identify

- potential interfaces with other business and information systems,
- critical dates (Y2K as an example),
- unique legal requirements,
- unique security requirements and constraints,
- unique network requirements and constraints, and
- special considerations - use of ISO 9000 standards conformance (used by some organizations that qualify their suppliers by that standard), QC-1, . . .

Figure D.1.3 Summary of Project Planning



Appendix E

Uniform Resource Locator (URL) List

URL 1 (http://www-irn.sandia.gov/silc/IssueC.html)..... ii
Software and Information Life Cycle Document

URL 2 (http://www-irn.sandia.gov/policy/makbypol.htm). 2
The Make or Buy Policy and Model

URL 3 (http://www-irn.sandia.gov/organization/div4000/isiproc/isistart.htm). 7
Information Systems Implementation Process and Environment

URL 4 (http://www-irn.sandia.gov/organization/div4000/isiproc/isistart.htm). 13
Production Guidelines and Environment

URL 5 (http://www-irn.sandia.gov/organization/div4000/isiproc/isistart.htm). 14
Information Systems Implementation Checklist

URL 6 (http://www-irn.sandia.gov/organization/div4000/ctr4800/astg.htm).. 14
Application Support Transition Guide

URL 7 (http://www.rational.com/UML).....21
Unified Modeling Language

URL 8 (http://www.aisintl.com/case/CDM-PDM.html) A-17
Logical Model

URL 9 (http://www-irn.sandia.gov/silc/svvptpl.pdf)..... A-27
Template for Software Verification and Validation Plan

URL 10 (http://www-irn.sandia.gov/silc/etktp01.pdf)..... A-53
Verification and Validation Plan - Electronic Timecard System

URL 11 (http://www-irn.sandia.gov/silc/etktp01a.pdf)..... A-72
Test Plan (Modular) - Web Electronic Timecard, Manual Tests

URL 12 (http://www-irn.sandia.gov/silc/etkscr~1.pdf)..... A-72
Developer Specifications - Web Electronic Timecard (Deliverable 8)

URL 13 (http://www-irn.sandia.gov/silc/etkcortp.pdf). A-72
Test Plan (Manual Tests) - Electronic Timecard Review and Approval Screen

URL 14 (http://www-irn.sandia.gov/silc/botp.pdf). A-72

Test Plan (Comprehensive) - Business Reporting System

URL 15 (http://www-irn.sandia.gov/silc/stptpl.pdf)..... A-75
Test Plan Template

URL 16 (https://www.prod.sandia.gov/NWIS/nwis.html)..... A-84
User's Guide for NWIS

URL 17 (http://www-irn.sandia.gov/silc/etktr01w.pdf)..... A-88
Test Report - Web Electronic Timecard, Manual Tests

URL 18 (http://www-irn.sandia.gov/silc/bo-winr2.pdf)..... A-88
Performance Test Report - Business Reporting System (Uses WinRunner)

URL 19 (http://www-irn.sandia.gov/silc/bo-winr.pdf)..... A-88
GUI Test Report - Business Reporting System (Uses WinRunner)

URL 20 (http://www-irn.sandia.gov/testbed)..... C-7
IIS Testbed Home Page

Appendix F

Future Needs in SILC D

1. Incorporate the interfaces with the Repository for model and metadata stores.
2. Review all the physical design set. Resolve overlaps in analysis and design sets
3. Review the implementation phase - include the variety of activities that actually occur therein.
4. Add measures and metrics for assessing quality improvements in software produced with this process.
5. Expand on references in the one-page descriptions of deliverables in Appendix B.
6. Reconsider adding object process models to the example in Appendix A.
7. Extend the Test Results example in Appendix A.
8. Review and change any processes to reflect consensus best practices of current projects.
9. Review and document best practices for commercial software implementation processes.
10. Review and document best practices for obtaining requirements pertaining to commercial software purchases.
11. Design reviews tend to be status reviews as described currently. A Post Implementation Review needs to be added and described to complete the review processes.

Distribution:

1	MS0113	Donald H. Flores, 10305
1	MS0115	Scott Joyce, 4612
1	MS0469	John F. (Jack) Jones, 4600
1	MS0629	Paul D. Merillat, 4800
1	MS0629	Al Alvarado, 4800
1	MS0638	Dwayne Knirk, 12326
1	MS0660	Linda Blevins, 4612
1	MS0660	Andrea Cassidy, 4612
1	MS0660	Marc B. Gunkel, 4612
1	MS0660	Jim D. Harmon, 4612
1	MS0660	Scott R. Joyce, 4612
1	MS0660	Timothy M. Meeks, 4612
1	MS0660	Michael H. Pendley, 4612
1	MS0660	Fay M. Tamashiro, 4612
1	MS0660	Dawn C. Abbott, 4825
1	MS0660	David L. Blackledge, 4825
1	MS0660	David S. Cuyler, 4825
1	MS0660	John H. Dexter, 4825
1	MS0660	Ronald C. Hall, 4825
1	MS0660	Pamela K. Lackey, 4825
1	MS0660	Lois J. Lauer, 4825
1	MS0660	Patrick B. Milligan, 4825
1	MS0660	Gary E. Rivord, 4825
1	MS0660	Joseph R. Schofield, 4825
1	MS0660	Benjamin Anello, 4619
1	MS0660	Sandra C. Babb, 4619
1	MS0660	Michael K. Bencoe, 4619
1	MS0660	Shelley M. Eaton, 4619
1	MS0660	Luis Hernandez, 4619
1	MS0660	Karl Horak, 4619
1	MS0660	Lilly Ingham-Hill, 4619
1	MS0660	Christopher J. Lujan, 4619
1	MS0660	Twila Parker, 4619
1	MS0660	William D. Swartz, 4619
1	MS0660	Dennis R. Tenorio, 4619
1	MS0660	Michael J. Eaton, 4622
1	MS0660	John W. Hatley, 4622
1	MS0660	Ken Keahbone, 4622
1	MS0660	Joe P. Sena, 4622
1	MS0660	John B. Whitley, 4622
1	MS0801	Melissa Murphy, 4400
1	MS0803	L. Herb Pitts, 4400
1	MS0805	Bill Mertens, 4823
1	MS0806	C. Douglas Brown, 4621

1	MS0806	Leonard Stans, 4616
1	MS0806	Michael O. Vahle, 4616
1	MS0807	R. Michael Cahoon, 4418
1	MS0811	Thomas L. Ferguson, 4813
1	MS0811	Jim Stromberg, 4813
1	MS0817	Mary Lynn Clark, 4815
1	MS0817	Lucien Don Daigle, 4815
1	MS0913	Jim Hamilton, 4411
1	MS1012	Grant C. Claycomb, 4811
1	MS9018	Central Technical Files, 8940-2
2	MS0899	Technical Library, 4916
1	MS0619	Review & Approval Desk, 15102 For DOE/OSTI