8/2/95

SANDIA REPORT

SAND95-1614 • UC-905

Unlimited Release

Printed July 1995

RECEIVED

AUG 21 1995

OSTI

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

Analysis and Implementation Phase Adaptations of the Sandia Software Guidelines: Issue A April 18, 1995

Donna Eaton, Andrea Cassidy, David Cuyler, Shelley Eaton, Scott Joyce,
Elisa Kephart, Joe Schofield, Dwayne Knirk

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-94AL85000

Approved for public release; distribution is unlimited.

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

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
US Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A05
Microfiche copy: A01

DISCLAIMER

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

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

**Analysis and Implementation Phase
Adaptations of the Sandia Software
Guidelines: Issue A
April 18, 1995**

Donna Eaton, Andrea Cassidy, David Cuyler, Shelley
Eaton, Scott Joyce, Elisa Kephart, and Joe Schofield
(Integrated Information Services)

Dwayne Knirk (Quality Engineering Department)

Sandia National Laboratories
Albuquerque, NM 87185

Abstract

This document describes the processes to be used for creating corporate information systems within the scope of the Integrated Information Services (IIS) Center. This issue A describes the Analysis and Implementation phases within the context of the entire life cycle. Appendix A includes a full set of examples of the analysis set deliverables. Subsequent issues will describe the other life cycle processes as we move toward enterprise-level management of information assets, including information meta-models and an integrated corporate information model. The analysis phase as described here, when combined with a specifications repository, will provide the basis for future reusable components and improve traceability of information system specifications to enterprise business rules.

This document is available for viewing on the Sandia Internal Web, under the URL
http://sass902d.itd.sandia.gov/SLC_Alnk.pdf

Table of Contents

Introduction	1
Software Phases	4
References	15
Acknowledgments	16
Glossary of Terms	17
Appendix A	A-1
 Problem Statement	A-2
Problem Narrative—Version 0	A-3
Problem Narrative—Version 1	A-4
Context Diagram	A-6
High-Level Function Diagram	A-7
 Business Rules	A-8
 Conceptual Information Model	A-23
Constrained Sentences	A-24
Natural Language Model Diagram	A-27
 Logical Information Model—IDEF1X Diagram	A-29
 Elementary Processes	A-31
Process Definition from Data Model	A-32
 Presentation Sets	A-38
Process/Actor State Transition Matrix	A-39
Storyboards	A-48
Event List	A-64
Process List	A-69

Introduction

This report is designed to serve two overall purposes:

- Short-term: To unify the baseline deliverables produced by the analysis process, and to provide guidance for software developers who need to prepare for conceptual design reviews
- Long-term: To move corporate software and information life cycle processes toward a unified and consistent methodology, which serves the needs of various types of software development projects at Sandia.

Scope

Short-Term Scope—Deliverables

This report defines processes and checklists that will move the IIS organization toward an enterprise-level specification management system which will document:

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

This report includes a helpful, non-inclusive set of materials that describe the conceptual design of a system. A complete set of these deliverables would provide a development team with an understanding of the desired information system. This set of information could guide the customer, software developer, review team, or 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" on page 15). This report adapts the Sandia Software Guidelines to this class of software systems. It is written not to replace those documents, but to position Sandia to take advantage of new technologies and tools. It also supports current three-layer or full three-tiered architecture and future distributed computing environments.

This work is on-going; it will continue to evolve and parallel the dynamic process of applying new technologies within Integrated Information Services (IIS). We publish our initial efforts here as issue A, to describe the Conceptual Design and Analysis phase within the context of the entire software life cycle. In this effort to achieve enterprise-level management of information assets, we will move toward information meta-models, rapid prototyping, and an integrated corporate information model. We will produce later issues approximately quarterly.

Desired Future State

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

Principles

We have incorporated the principles stated in the Information Architecture document (see "References" on page 15). In addition, a series of "core beliefs" or "guiding principles" contributed to the interaction of this team. These principles capture how the team attempted to understand and advance the software engineering process within the organization. Their inclusion below may contribute to the understanding of this document.

- We take most seriously this opportunity to improve the software development process (hereafter referred to as process.)
- The process is our foundation for productivity, improvement, and measurement.
- The process is defined with adequate rigor to earn the confidence of customers.
- A repeatable, documented process will enable us to reduce variation in the quality of our software products.
- The process includes embedded validation and verification techniques eventually eliminating the need for management review.
- Management leadership is desirable in the strategic planning process establishing priorities and fostering integration.
- Responsibility and accountability for conformance to specification for software rests with the software engineer.
- Each task and deliverable is assessed by the value added to the process.

Impact

This document represents a plan for moving the development of integrated information systems from our current reality to a preferred future state. The process changes by which we attain our desired future state are only possible if management accounts for full life cycle software costs and takes responsibility and ownership for actively balancing initial delivery schedules with total software quality.

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)
- Reduce overall costs throughout the life cycle (thus allowing for workforce redeployment or acquisition of additional work).

To these goals, we have provided the basis of a process which will shift costs and time from coding and nonproductive rework toward the analysis and design phases of the life cycle, addressed in this report. In addition, there are start-up costs which will be evident

in defining and building or acquiring and populating a repository. Overall, we expect total development costs to decline as requirements are gathered correctly and completely the first time.

The success of applying these processes and methodologies and thus, attaining management's goals, will be possible only if:

- Management enforces this life cycle process, or
- This life cycle process is enforced by a tool requiring changes only be made to the specifications.

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, subject to the risk that development entails and to the current stage of implementation of architecture and data administration tools.

Before the desired future state is reached, real problems will surface with loosely interpreted application of this process. Issue B of this document will address the actual development activities and the tailoring of this process to the area of enhancements to existing systems.

Software Phases

Planning Phase

This phase involves the conceptual development of a computer solution to a business need. The planning phase may be very short if the project is described in an overall business strategic plan. However, often the project is dynamically created as an evolution of business changes in reaction to changing circumstances. Many projects result from a formal change control system, where changes and enhancements require extensive work.

The planning necessary for initiating an information system is sometimes included as part of the scoping, problem statement, or early assessment in the analysis phase. A preferred future state encompasses enterprise-wide planning and modeling. Where applicable, strategic information planning for business systems within the corporate information model includes prioritizing, sequencing, and allocating resources. A stronger emphasis on strategic and integrated planning is advocated as the number of systems, platforms, and technologies increase.

Planning Phase Steps

1. Business managers and staff (process owners) define the business need and often describe a set of solutions, of which one is chosen to be studied and developed into a system proposal.
2. Information system analysts begin to scope the problem, assess the current systems, describe a set of alternative solutions with estimates of costs and schedules for each, and recommend a solution.
3. Business managers decide on a solution to fit within the overall plan for the enterprise. Once approved, the recommended solution is documented in procedures and policy documents as the business rules for the enterprise. The IIS is requested to build or acquire a system to support these rules.

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 presented in reference 5 (see "References" on page 15). 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. It will eventually support automatic generation of information systems and business rule descriptions.

Analysis Phase Steps

1. A scope or "universe of discourse" is first defined to set boundaries on the problem. In addition, a problem statement is developed, including a problem narrative, context diagrams, and other descriptive materials.

2. A version-controlled repository will be used to manage all model constructs. An enterprise-level tool will help manage reusable information and functions for multiple models and systems. The repository will be queried 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 to add them to the repository if they are not already defined and used in other systems. Negotiations 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. A natural language (English sentences) model is created 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. A natural language process model and graphical presentation model are produced to precisely define the processes that will act on the information. (This can include storyboards, known as presentation sets or prototypes.) Constraints and derivation rules will be 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. Steps 3 and 4 are repeated 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 analyst accept this reviewed analysis set by formal sign-off.
6. A conceptual review (see "Glossary of Terms" on page 17) will be held to verify the specifications produced in this analysis of this application or sub-system. The review may also be attended where appropriate 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.
7. An ongoing goal of this overall process is to shift the traditional notion of testing earlier in the life cycle through the verification and validation of specifications. Examples play a critical role in the verification process. At some future point, confidence in the rigor of the specification and development process should reduce our reliance on classical unit, integration, system, and acceptance testing. Reasons for shifting this emphasis include:
 - The growing sophistication and reliability of software analysis and design 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.
 - The traditional "acceptance test plan" can be derived and the validation completed from the English syntax used to capture requirements and examples.
 - 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.

The following tables indicate the specification deliverable set (Table 1) and extended descriptions of these deliverables (Table 2). In Table 1, numbered items refer to their corresponding definitions in Table 2.

Table 1 Critical Specification Deliverable Set

	Plan Set [†]	Analysis Set	Physical Design Set	Implementation Set	Operational Engineering Set
For the customer	Information systems plan	(1) Problem statement (2) Business rules (6) Presentation sets		(to be discussed in future release)	(to be discussed in future release)
For the software engineering team	Reuse plan Change control plan Configuration management plan	(3) Conceptual model (4) Logical model (5) Elementary processes	(7) CRUD matrix (8) Extended presentation sets (9) DB schema (10) Dependencies & implementation constraints	(Process Checklist discussed and presented in this issue)	(to be discussed in future release)
For the review team	Metric plan Corporate model synthesis plan				
Phase Review		CDR "Conceptual Review" or "Preliminary Design Review"	DDR "Detailed Design Review" or "Critical Design Review"	FDR "Final Design Review" or "Pre-Production Review"	

†. Outside this project's scope, but required to guide the proper execution of the process

Table 2 Descriptions of Deliverables Produced by the Processes in Table 1

Deliverable	Primary Communication Roles	Dependencies	Description
1 Problem Statement	Customer : Analyst	(none)	Consists of: <ul style="list-style-type: none"> • Problem narrative—a textual deliverable that states the business problem in concise terms • Context diagram—may be included to show the scope of the system and which objects flow between the system and an external system interface. Large problems should be decomposed into subsystems that can be defined in two to three pages.
2 Business Rules	Customer : Analyst	(1)	Formal natural language sentences with examples specify aspects of each object to be maintained in the information system. Initial storyboards or cartoons and flow diagrams are also used to document the business rules and verify them with the customer. General constraints and derivation rules are included in textual form.
3 Conceptual Information Model	Analyst : Designer	(2)	Constrained sentences in textual form and in a natural language model diagram communicate precisely the information required by the designer.
4 Logical Information Model	Analyst : Designer	(3)	The 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).
5 Elementary Processes	Analyst : Designer	(4, 2)	Elementary processes that 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.

Table 2 Descriptions of Deliverables Produced by the Processes in Table 1 (Continued)

Deliverable	Primary Communication Roles	Dependencies	Description
6 Presentation Sets	Customer : Analyst Analyst : Designer	(5, 2)	Consists of: <ul style="list-style-type: none"> • Process/Actor State Transition Matrix—adds rigorous definitions of the processes related to each object. In addition, from the customer's viewpoint, it helps integrate the actions performed and the state of the information system for each elementary process. • 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 feel" with other systems. • Event and process lists define the flow as related to the storyboards.
7 CRUD Matrix	Designer : Implementer	(4, 5, 6)	The CRUD (Create, Read, Update, Delete) matrix identifies each elementary process used on each core object of the information system. This assists in verifying that each object is supported by a create, read, update, and delete process—it also helps identify redundancy of processes for an object.
8 Extended Presentation Sets	Designer : Implementer	(6)	Storyboard diagrams that include vendor-specific and implementation-specific features, as well as application of standard template features for the enterprise systems.
9 DB Schema	Designer : Implementer	(4)	The Physical Database, using the constructs of the vendor software and the additional information needed by the implementer of the system, is documented with the database schema diagram in the form of IDEF1X. Additionally, the SQL listings may be used.
10 Process Dependencies/ Flow	Designer : Implementer	(6, 2)	A formal definition of precedence required in the flow steps.

Deliverables 7-10, below the double line, will be discussed in Issue B.

Figure 1 Roles and Responsibilities

Physical Design Phase

This step will use the specifications and extend them into a physical database schema and prototype. The system designer should be a knowledgeable developer with experience in building and using preferred technology information systems tools, as bounded by the architecture of the IIS organization. The designer should also have skills in human factors engineering as well as the classical programming training and skills.

Physical Design Phase Steps

1. If produced, a prototype system will be used for customer verification and specification refinement.
2. All changes will be made to the natural language models and new versions of the specifications and repository constructs will be generated before the prototype is refined.
3. The physical model will be developed as a natural language model. This model will be the specifications for the physical database and the program processes.
4. Presentation, function, and data layer elements will be determined and identified.
5. The physical model and general constraints (other rules and information, not contained in the model) will be added to the repository.
6. All reusable facts and functions will be identified and provided to the application developer.
7. A Detailed Design Review will be performed to verify the physical design.

The preceding tables, Table 1 and 2, indicate the design process deliverable set, as currently defined.

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 future systems code is automatically generated, the coordination of interface groups 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 physical database performance, distributed location and features, access requirements, etc. will be built by the database administrator.
2. A cloned database will be provided to the customer and the developer for testing. Other test data may be created using the business rules information model with its corresponding examples.
3. The presentation layer components will be built according to the current software and hardware architecture and standards of the IIS organization. Where possible, the functional layer programs will be generated directly from the specifications.
4. The functional layer components not already available in the corporate repository will be built according to the current software/hardware architecture and standards of the IIS organization. Where possible, the code will be generated directly from the specifications.
5. Integrated testing will be performed against the cloned test database.

-
6. System testing will be performed on the corporate test bed system.
 7. An as-built design review will be performed for the pre-production application.
 8. The steps of the Deployment Process will be performed to move the system into a fully operational production system.

Deployment Process

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 deployment coordinator should possess the skills necessary to manage enormous amounts of data with great care for each detail. The deployment manager must also be able to perform multiple tasks and manage coordination of schedules of many groups within both the IIS organization and the customer organization.

- a. Define the processing duplication, i.e., parallel runs for validation and verification requirements.
- b. Populate the Production database with the static data and legacy data.
- c. Connect the networks to access related and dependent data from other Distributed Computing Environment (DCE) systems.
- d. Develop the training classes and train users, as well as production support staff and help desk personnel.
- e. Develop operational procedures, including company asset protection processes and plans.
- f. Perform a Customer Acceptance Review, with an official sign-off for the system.

Table 3 on page 13 indicates the process checklist, as currently defined.

Information Systems Implementation Checklist

(Also available on the World Wide Web:

http://www_irm.sandia.gov/organization/div13000/isiproc/isistart.htm)

Three things you need to know about this implementation checklist:

- It is a standard which incorporates required sign-offs necessary for productionizing new products and enhancing existing ones within the Integrated Information Services Organization. This process ensures continued support without dependence on the original developer.
- 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)
 - Write the service-level agreements documenting operational and performance requirements.
- 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.

Table 3 Information Systems Implementation Checklist

Consult with End Customer about ...	Consult with System Administration about...	Consult with Application Support about...	Consult with Technical Test Bed about...	Consult with Data and Database Administration about...	Consult with Network Communications, Security, and Production Services about...
☐ 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 applications reside and execute ☐ system design reviews	☐ Data Administration data modeling requirements ☐ Data requirements including backup and recovery, security, conversion, volumes and growth, source/target and archive/delete	☐ Production Services documentation requirements ☐ Security data security classifications and security plans ☐ Network Communications traffic tolerances and network needs
☐ early life cycle aspects above ☐ customer acceptance and verification testing	☐ early life cycle aspects above ☐ backup and recovery requirements ☐ scheduling and compatibility of system software installs and upgrades (software integration) ☐ stress test	☐ early life cycle aspect 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	☐ 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 ☐ 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 dynamic operational and development processes. This phase requires great skills in quality management techniques to assure the integrity and protection of valuable corporate assets and to maintain customer delight during day-to-day business operations.

Operation Phase Steps

1. The operation phase is a continual development process for new requirements on the application system. This phase will require extensive configuration management tools and an intensive commitment to the maintenance of the specifications and constructs in the corporate repository. To the extent required, all procedures will be performed as outlined in the life cycle phases: Planning, Analysis, Physical Design, and Implementation.
2. Where possible, all programs will be modified automatically from changes to the system specifications.
3. All changes to application system specifications will result in new business rule documents being generated from those specifications. 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.

Termination Phase

The system will be closed out after it is no longer needed. More details and other processes will be included in a later issue of this report.

Termination Phase Steps

1. Any legacy data, with continuing requirements, will be moved to other systems.
2. Corporate Records Management will evaluate the transfer of specifications and historical information to be retained for proper disposition and thereafter saved and managed by that organization.
3. The constructs and process specifications which are no longer required for this system, nor others, are removed from the corporate repository.

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 3, Standards, Practices, and Conventions, SAND85-2346, July 1986.
4. Sandia Software Guidelines, Volume 4, Configuration Management, SAND85-2347, June 1992.
5. Sandia Software Guidelines, Volume 5, Tools, Techniques, and Methodologies, SAND85-2348, July 1989.
6. Software Development Process Methodology, Sandia, June 23, 1993.
7. SLP 1011, Software Management, Draft.
8. SLP 2815, Preferred Processes, December 1992.
9. Computer Software Management, DOE Order 1330.1D, Office of Information Resources Management, U.S. Department of Energy, Washington, D.C., May 1992.
10. Becker, Steve D. Annotated Summary of the Information Model Design Procedure (IMDP), SAND94-0460, May 1994.
11. Nijssen, G.M. and Halpin, T.A. Conceptual Schema and Relational Database Design. Prentice Hall, 1989.
12. Nijssen, G.M. "The Essence of Universal Informatics." Class given in Albuquerque, New Mexico, July 1992.
13. Nijssen, G.M. and Twine, S.M. "A Summary of the Conceptual Schema Design Procedure (CSDP)." Nijssen Adviesbureau voor Informatica, 1990.
14. Nijssen, G.M. and Twine, S.M. "A Framework for Information System Development Methodologies." Nijssen Adviesbureau voor Informatica, 1990.
15. Organization 13316. "Department 13316 Home Page." Sandia Web, IRN.
http://sass902d.itd.sandia.gov/Dept13316_Home.html
16. Organization 13316. "Natural-language Information Analysis Methodology (NIAM)." Sandia Web, IRN.
<http://sass902d.itd.sandia.gov/Niam/NiamIntro.html>
17. Organization 13000. "Information Architecture." Sandia Web, IRN.
http://www_irm.sandia.gov/organization/div13000/ctr_13200/dpt13202/ouia_94.htm

Acknowledgments

Contributors—Issue A Team

Duane Garrison	Administrative Systems
Jim Stromberg	Administrative Systems
Andrea Cassidy	Administrative Systems
David Cuyler	Data Administration
John K. Sharp	Data Administration
Ronald Hall	Data Administration
Mary Lynn Clark	Electronic Commerce
Scott Joyce	Information Technologies
Irene Thurston	Information Technologies
Donna S. Eaton	Information Technologies
John Hatley	Information Technologies
Shelley Eaton	Information Technologies
Joe Schofield	Information Technologies
Gary E. Rivord	Information Technologies
Greg Conrad	Information Technologies
Elisa Kephart	Information Technologies
Charlene Strickland	Information Technologies
Dwayne Knirk	Software Quality
Al Alvarado	Program Management
Carol Jones	Program Management
Mike Pendley	Test Bed System
Bill Mertens	Database Administration
T. C. Adams	Production Services
Nancy Marsh	The Production Process

Glossary of Terms

Table 4 Glossary

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
Cloned database	An exact replica of a database, created for testing purposes
Conceptual Design Review	CDR; A review to verify the conceptual information model created through 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 facts 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

Table 4 Glossary (Continued)

Term	Definition
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 must operate. Arrows drawn between 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.
CRUD	Create, Read, Update, Delete
CSU	Customer Service Unit
Customer	The internal or external buyer or recipient of a product
DCE	Distributed Computing Environment
Detailed Design Review	DDR; To review the implementation information model. The implementation model may include implementation-specific facts (such as surrogate keys, internal objects, etc.) used to optimize physical implementation. Full compliance of the physical structures to the model specifications will be reviewed. Any constraints implemented in the schema will be tested. Specific use of data by the functional layer will be reviewed. Production support/operation documents should exist.
DOE	Department of Energy
ER	A data model that describes attributes of entities and the relationships among them
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
Flow charts	A control flow diagram in which suitable annotated geometrical figures are used to represent operations, data, or equipment, and arrows indicate the sequential flow from one to another
Functional diagrams	Graphic depictions of functional components of a system
IDEF	Integrated computer manufacturing Definition

Table 4 Glossary (Continued)

Term	Definition
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
IEF	Information Engineering Facility—a software development tool
IIS	Integrated Information Services
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	<ul style="list-style-type: none"> • 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
Rapid prototyping	A type of prototyping using powerful tools and processes to accelerate the task; emphasis is placed on developing prototypes early in the development process to permit early feedback and analysis in support of the development process
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 blueprint or model of the system
SLP	Sandia Laboratories Policy
Software engineering	The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software

Table 4 Glossary (Continued)

Term	Definition
SRS	Software Requirements Specification—Documentation of the essential requirements (functions, performance, design constraints, and attributes) of the software and its external interfaces
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 modules which comprise a software system
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

Appendix A

Example of Deliverable Set to Complete the Analysis Phase and to Use at the Conceptual Design Review

This example includes Deliverables 1 through 6. They meet the guidelines for reviews with customers or technical staff, as described in the Software and Information Life Cycle analysis phase.

This classroom example was chosen because it is documented thoroughly (through efforts of the X3T2 Committee of the International Standards Organization, ISO, chaired by John Sharp).

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 analyst and the customer(s) as to the nature and scope of the system. It establishes a bound (fuzzy, to be sure) to the capabilities of the system. It gives the analyst a starting point for determining the major components of the system. It also launches the customer/analyst 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
- Description of phasing if intended (large or complex systems may be developed in several phases).

Problem Narrative—Version 0

Deliverable 1

Narrative from initial meeting with the customer

Japan Wines, Inc., is a wine distribution center that needs a system for maintaining product accountability. The system should maintain accountability of products, orders, deliveries including orders from customers and orders to wineries, and updates to products, customers, and wineries.

The system does not include issues that deal with employees, product pricing, and customer and winery bills and payments.

The system should provide the means to keep a current list of the products offered by Japan Wines. It should allow orders to be taken from customers over the phone. Customer orders should be logged. The system should assign deliveries to delivery trucks, and it should be able to record which deliveries have been completed, and which have not. It should keep track of available, assigned, and total inventory levels, and automatically adjust these levels as orders are placed and product is delivered. When available inventory falls below minimum levels, the system should automatically generate orders for more wine, and fax these orders to the appropriate wineries. Wine orders should be logged. Once it is confirmed that a wine order has been received, the system should automatically update the inventory levels accordingly.

Problem Narrative—Version 1

Deliverable 1

Narrative with revisions, resulting from analysis phase

[1] Japan Wines, Inc. is a wine distribution center that needs a system for maintaining product accountability. [2] The system is to maintain accountability of products, orders, deliveries including orders from customers and orders to wineries, and updates to products, customers and wineries. [3] Issues that deal with product pricing, customer and winery bills and payments, and employees are not to be included in this system; a single order will not exceed the capacity of a truck; and the wine distribution center does not accept backorders.

[4] Customers call in to order products and when the order is completed it is confirmed. [5] Each order has a unique customer order number, and the date of the order is automatically entered. [6] The customer's telephone number is obtained either automatically by caller identification or by a direct request; it is used to assist in identifying the unique customer number, the customer name, and the customer delivery address. [7] If the customer phone number does not enable the retrieval of the customer number, then it is asked if the customer has ordered products before. [8] If the customer has, then the customer information is updated, if not, a new customer number is created and the customer name and delivery address are entered. [9] Products are ordered by first entering the product number which is used to display the product name. [10] The quantity requested is then entered. [11] The inventory is checked to see if the available inventory quantity (total inventory minus inventory assigned to pending deliveries) is sufficient for the request. [12] If the requested quantity is not currently available, then what is available is offered to the customer. [13] After all of the products are ordered and confirmed, the order is logged and the available inventory quantities are reduced to reflect the order. [14] This wine distribution center does not accept backorders.

[15] At the end of the day, winery orders are automatically faxed to the wineries when the available inventory quantity is less than the minimum stock quantity for any product. [16] The quantity ordered is the maximum stock quantity reduced by the available inventory quantity. [17] The winery order has a unique winery order number, and the date of the order is automatically entered. [18] Wineries are identified by their phone numbers. [19] The winery name and the winery fax number appear along with the phone number on the winery order. [20] Each product is obtained from only one winery, wineries receive only one order per day no matter how many products are ordered, and the winery orders are automatically logged.

[21] Each morning before customer orders are accepted, and only once per day, an inventory status report is printed and all unfilled customer orders are assigned to a delivery truck identified by a truck number. [22] The inventory status report includes all products in inventory and specifies the total, assigned, and available quantities. [23] It also specifies the minimum and maximum stock quantities and the quantities ordered the previous day. [24] Assigned quantities for each product are pulled, then each customer

order is prepared and loaded onto the assigned truck. [25] If the order is successfully delivered then the delivery instruction ticket is checked 'Completed.' [26] If the order isn't delivered then the delivery instruction ticket is checked 'Re-Deliver.' [27] The checked delivery instruction tickets are used to log the successfully delivered orders. [28] Each delivered order depletes both the total inventory and the assigned inventory; the available inventory is unchanged.

[29] New products are introduced frequently. [30] For each new product a product number is created; the product name, minimum and maximum stock quantities are set. [31] An initial inventory quantity can be entered. [32] The winery supplying the product is also entered. [33] If the product comes from a new winery, then its name, phone number, and fax number are entered. [34] Product information can also be updated to reflect modifications to product names, changes in the minimum and maximum stock quantities, or adjustments to reflect actual total inventory. [35] Wineries are also added or updated when necessary.

[36] Every product ordered on the same day from the same winery is placed on a single winery order. [37] Once receipt of a winery order is confirmed, the total and the available inventory quantities are adjusted to reflect the products that have been received; the receiving date is automatically entered and the receipt of the order is logged.

Context Diagram

Deliverable 1

This shows the scope of the system and which objects flow between the system and an external system interface. Dashed lines indicate business actions not covered in this example set.

Figure 2 Context Diagram—Japan Wines, Inc., Example

High-Level Function Diagram

Deliverable 1

This example shows the decomposition of the Wine Distribution System into major sub-processes. This depicts the central part of Figure 2 on page A-6.

Figure 3 Wine Distribution System—Japan Wines, Inc., Example

Business Rules

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” on page A-38). 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, description, type, domain, and value constraints, along with examples.

Note: In this example, objects are labeled as entities.

Entities with Label Names

ENTITY	LABEL
Address A street address or location. <i>SuperType:</i> <i>Values:</i> <i>Domain:</i> Alphanumeric characters (maximum 100)	-- <i>Examples:</i> 111 First Street 88 Easy Street 12 Carter Lane
Customer A person or company that buys wine from Japan Wines, Inc. <i>SuperType:</i> <i>Values:</i> <i>Domain:</i> Automatic numbering, starting with 1 with increments of plus 1, maximum of 999999	customer_# <i>Examples:</i> 89 92 77
Customer_Order An order that is taken from a customer for the purchase of wine from Japan Wines, Inc. <i>SuperType:</i> <i>Values:</i> <i>Domain:</i> Automatic numbering, starting with 1 with increments of plus 1, maximum of 999999	customer_order_# <i>Examples:</i> 5742 5788 5790
Customer_Order_Line A particular order line of a particular customer order (nominalization of FT7). <i>SuperType:</i> <i>Values:</i> <i>Domain:</i> Exists in Customer_Order_# and shares its domain; and exists in Product_# and shares its domain.	customer_order_#_and_product_# <i>Examples:</i> 5742 orders 3X21 5742 orders 3X33 5790 orders 3X33
Date A calendar date expressed with the day, month, year. <i>SuperType:</i> <i>Values:</i> <i>Domain:</i> [Two-digit day (1-31)]-[full month name (January, February, March, April, May, June, July, August, September, October, November, December)]-[four-digit year (1700-2100)]	date_code <i>Examples:</i> 17-May-1994 21-June-1994 03-January-1995

ENTITY	LABEL
Delivered_Customer_Order A subtype of Customer_Order, one which has been delivered to the customer. <i>SuperType:</i> Customer_Order <i>Values:</i> <i>Domain:</i> Exists in Customer_Order	customer_order_# <i>Examples:</i> 5742 5788 5790
Name A full name <i>SuperType:</i> <i>Values:</i> <i>Domain:</i> Alphanumeric characters (maximum of 20)	-- <i>Examples:</i> Joe's Bar green chili wine Sam's Winery
Phone A telephone number. <i>SuperType:</i> <i>Values:</i> <i>Domain:</i> Alphanumeric characters (maximum of 20)	phone_# <i>Examples:</i> 721 3652 484-2566 888-3866
Product An item that Japan Wines, Inc. sells to its customers. <i>SuperType:</i> <i>Values:</i> <i>Domain:</i> Alphanumeric characters (maximum of 10)	product_# <i>Examples:</i> 3X21 3X33 4X35
Quantity An amount. <i>SuperType:</i> <i>Values:</i> <i>Domain:</i> Whole numbers, minimum of -999999, maximum of 999999	quantity_# <i>Examples:</i> 79 100 4

ENTITY	LABEL
Received_Winery_Order Subtype of Winery Order, one which has been received by Japan Wines, Inc. <i>SuperType:</i> Winery_Order <i>Values:</i> <i>Domain:</i> Exists in Winery_Order	winery_order_# <i>Examples:</i> 7555 6698 8101
Truck A delivery truck owned by Japan Wines, Inc. <i>SuperType:</i> <i>Values:</i> <i>Domain:</i> Automatic numbering, starting with 1 with increments of plus 1, maximum of 999999	truck_# <i>Examples:</i> 4 1 3
Unfilled_and_Unassigned_Customer_Order_Quantity A customer order that is both unfilled (not delivered) and unassigned. This is a derived transitory fact. <i>SuperType:</i> <i>Values:</i> <i>Domain:</i> Zero or positive whole numbers, maximum of 999999	quantity_# <i>Examples:</i> 100 12 33
Winery A place of business that produces and sells their wines. <i>SuperType:</i> <i>Values:</i> <i>Domain:</i> Alphanumeric characters (maximum of 20)	phone_# <i>Examples:</i> 321 4763 555 7112 399 3092
Winery_Order An order placed by Japan Wines, Inc. to a winery. <i>SuperType:</i> <i>Values:</i> <i>Domain:</i> Automatic numbering, starting with 1 with increments of plus 1, maximum of 999999	winery_order_# <i>Examples:</i> 7555 6698 8101

Final Example Sentences by First Entity

1 Customer_Order is created on Date.

Source # 5	Customer_Order	Date
Yes	customer_order_#	date_code
1 ☒	5742	17-May-1994
2 ☐	5742	21-June-1994
3 ☒	5788	17-May-1994
4 ☒	—	03-January-1995
5 ☐	5790	—

Customer_Order must be created on exactly one Date.

Date may be for creation of any number of Customer_Orders.

Comment: The date is automatically entered by the information system.

2 Customer_Order is for Customer.

Source # 4	Customer_Order	Customer
Yes	customer_order_#	customer_#
1 ☒	5742	89
2 ☐	5742	92
3 ☒	5788	89
4 ☒	—	77
5 ☐	5790	—

Customer_Order must be for exactly one Customer.

Customer may have any number of Customer_Orders.

Comment:

3 Customer has Name.

Source # 6	Customer	Name
Yes	customer_#	
1 ☒	89	Joe's Bar
2 ☐	89	Phil's Pub
3 ☒	92	Joe's Bar
4 ☒	—	Penelope Hartwig
5 ☐	77	—

Customer must have exactly one Name.

Name may be for any number of Customers.

Comment:

4 Customer is reached at Phone.

Source # 6	Customer customer_#	Phone phone_#
Yes		
1 ☒	89	721 3652
2 ☐	89	484-2566
3 ☐	92	721 3652
4 ☒	—	888-3866
5 ☐	77	—

Customer must be reached at exactly one Phone.
Phone may be for reaching exactly one Customer.

Comment:

5 Customer receives deliveries at Address.

Source # 6	Customer customer_#	Address
Yes		—
1 ☒	89	111 First Street
2 ☐	89	P.O. Box 1662
3 ☒	92	111 First Street
4 ☒	—	12 Carter Lane
5 ☐	77	—

Customer must receive deliveries at exactly one Address.
Address may be where deliveries are received by any number of Customers.

Comment:

6 Product has Name.

Source # 9	Product product_#	Name
Yes		—
1 ☒	3X21	green chile wine
2 ☐	3X21	potato wine
3 ☒	3X33	green chile wine
4 ☒	—	dandelion wine
5 ☐	4X35	—

Product must have exactly one Name.
Name may be for any number of Products.

Comment:

7 Customer_Order orders Product.

Source # 4	Customer_Order customer_order_#	Product product_#
Yes		
1 ☒	5742	3X21
2 ☒	5742	3X33
3 ☒	5788	3X21
4 ☒	—	4X35
5 ☐	5790	—

Customer_Order must order one or more Products.

Product may be ordered from any number of Customer_Orders.

Comment: This fact is nominalized to Customer_Order_Line.

8 Product is available in Quantity.

Source # 11	Product product_#	Quantity quantity_#
Yes		
1 ☒	3X21	79
2 ☐	3X21	100
3 ☒	3X33	79
4 ☒	—	4
5 ☐	4X35	—

Product must be available in exactly one Quantity.

Quantity may be available for any number of Products.

Comment: P16.

9 Product is assigned in Quantity.

Source # 11	Product product_#	Quantity quantity_#
Yes		
1 ☒	3X21	79
2 ☐	3X21	100
3 ☒	3X33	79
4 ☒	—	4
5 ☐	4X35	—

Product must be assigned in exactly one Quantity.

Quantity may be assigned for any number of Products.

Comment: Sum of all assigned quantities for unfilled orders.

10 Product has total of Quantity.

Source # 11	Product product_#	Quantity quantity_#
Yes		
1 ☒	3X21	79
2 ☐	3X21	100
3 ☒	3X33	79
4 ☒	—	4
5 ☐	4X35	—

Product must have total of exactly one Quantity.
 Quantity may be total for any number of Products.
Comment:

11 Product has minimum stock of Quantity.

Source # 15	Product product_#	Quantity quantity_#
Yes		
1 ☒	3X21	79
2 ☐	3X21	100
3 ☒	3X33	79
4 ☒	—	4
5 ☐	4X35	—

Product must have minimum stock of exactly one Quantity.
 Quantity may be minimum stock for any number of Products.
Comment:

12 Product has maximum stock of Quantity.

Source # 16	Product product_#	Quantity quantity_#
Yes		
1 ☒	3X21	79
2 ☐	3X21	100
3 ☒	3X33	79
4 ☒	—	4
5 ☐	4X35	—

Product must have maximum stock of exactly one Quantity.
 Quantity may be maximum stock for any number of Products.
Comment:

13 Winery_Order has Date.

Source # 17	Winery_Order winery_order_#	Date date_code
Yes		
1 ☒	7555	17-May-1994
2 ☐	7555	21-June-1994
3 ☒	6698	17-May-1994
4 ☒	—	03-January-1995
5 ☐	8101	—

Winery_Order must have exactly one Date.
 Date may be for any number of Winery_Orders.
Comment: The date of the winery order is entered automatically.

14 Winery_Order is for Winery.

Source # 18	Winery_Order winery_order_#	Winery phone_#
Yes		
1 ☒	7555	321 4763
2 ☐	7555	555 7112
3 ☒	6698	321 4763
4 ☒	—	399 3092
5 ☐	8101	—

Winery_Order must be for exactly one Winery.
 Winery may receive any number of Winery_Orders.
Comment:

15 Winery has Name.

Source # 19	Winery phone_#	Name
Yes		
1 ☒	321 4763	Sam's Winery
2 ☐	321 4763	Napa Valley Vineyards
3 ☒	555 7112	Sam's Winery
4 ☒	—	Sonoma Winery
5 ☐	399 3092	—

Winery must have exactly one Name.
 Name may be for any number of Wineries.
Comment:

16 Winery receives faxes at Phone.

Source # 19	Winery phone_#	Phone phone_#
Yes		
1 ☒	321 4763	721 3652
2 ☐	321 4763	484-2566
3 ☒	555 7112	721 3652
4 ☒	—	888-3866
5 ☐	399 3092	—

Winery must receive faxes at exactly one Phone.

Phone may be for receiving faxes at any number of Wineries.

Comment:

17 Product has current daily winery Quantity.

Source # 20	Product product_#	Quantity quantity_#
Yes		
1 ☒	3X21	79
2 ☐	3X21	100
3 ☒	3X33	79
4 ☒	—	4
5 ☐	4X35	—

Product must have current daily winery order of exactly one Quantity.

Quantity may be current daily winery order for any number of Products.

Comment: P23.

18 Product has existing winery on-order Quantity.

Source # 20	Product product_#	Quantity quantity_#
Yes		
1 ☒	3X21	79
2 ☐	3X21	100
3 ☒	3X33	79
4 ☒	—	4
5 ☐	4X35	—

Product must have existing winery on-order of exactly one Quantity.

Quantity may be existing winery on-order for any number of Products.

Comment: P23.

19 Product is purchased from Winery.

Source # 15, 17

	Yes	Product product_#	Winery phone_#
1	☒	3X21	321 4763
2	☐	3X21	555 7112
3	☒	3X33	321 4763
4	☒	—	399 3092
5	☐	4X35	—

Product must be purchased from exactly one Winery.

Winery may sell any number of Products.

Comment:

20 Winery_Order orders Product in Quantity .

Source # 20	Winery_Order	Product	Quantity
Yes	winery_order_#	product_#	quantity_#
1 ☒	7555	3X21	100
2 ☐	7555	3X21	12
3 ☒	7555	3X33	100
4 ☒	8101	3X21	100
5 ☐			
6 ☒	—	—	100
7 ☐	8101	3X21	—
8 ☐			

Winery_Order must order at least one Product which must be ordered in exactly one Quantity.

Quantity may be for one or more Products which may be ordered from one or more Winery_Orders.

Comment:

21 Customer_Order is sent on Date delivered by Truck .

Source # 24	Customer_Order	Date	Truck
Yes	customer_order_#	date_code	truck_#
1 ☒	5742	17-May-1994	4
2 ☐	5742	21-June-1994	1
3 ☒	5742	17-May-1994	4
4 ☒	5790	03-January-1995	4
5 ☐			
6 ☒	—	—	4
7 ☐	5790	17-May-94	—
8 ☐			

Customer_Order must be sent on at least one Date, which must be delivered by exactly one Truck.

Truck may deliver one or more Customer_Orders which are sent on at least one Date.

Comment:

22 Unfilled_and_Unassigned_Customer_Order_Quantity exists.

Source # 28 Unfilled_and_Unassigned_Customer_Order_Quantity

quantity_#

Yes

1 ☒ 1002 ☐ 123 ☒ -4 ☐5 ☐

One Unfilled_and_Unassigned_Customer_Order_Quantity may exist.

Comment: P18.

23 Delivered_Customer_Order exists.

Source # 25 Delivered_Customer_Order

customer_order_#

Yes

1 ☒ 57422 ☒ 57883 ☒ -4 ☐5 ☐

Any number of Delivered_Customer_Orders may exist.

Comment:

24 Received_Winery_Order is received on Date.

Source # 37 Received_Winery_Order

winery_order_#

Date

date_code

Yes

1 ☒ 75552 ☐ 75553 ☒ 66984 ☒ -5 ☐ 8101

Received_Winery_Order must be received on exactly one Date.

Date may be for receiving of any number of Received_Winery_Orders.

Comment:

17-May-1994
21-June-1994
17-May-1994
03-January-1995
-

25 Customer_Order_Line has ordered Quantity.

Source # 12	Customer_Order_Line	Quantity
Yes	customer_order_#_and_product_#	quantity_#
1 ☒	5742 orders 3X21	79
2 ☐	5742 orders 3X21	100
3 ☒	5742 orders 3X33	79
4 ☒	—	4
5 ☐	5790 orders 3X33	—

Customer_Order_Line must have ordered exactly one Quantity.

Quantity may be ordered for any number of Customer_Order_Lines.

Comment:

26 Customer_Order_Line has available Quantity.

Source # 11	Customer_Order_Line	Quantity
Yes	customer_order_#_and_product_#	quantity_#
1 ☒	5742 orders 3X21	79
2 ☐	5742 orders 3X21	100
3 ☒	5742 orders 3X33	79
4 ☒	—	4
5 ☐	5790 orders 3X33	—

Customer_Order_Line must have available exactly one Quantity.

Quantity may be available for any number of Customer_Order_Lines.

Comment: Total Quantity - Assigned Quantity = Available Quantity.

27 Customer_Order_Line has requested Quantity.

Source # 11, 12	Customer_Order_Line	Quantity
Yes	customer_order_#_and_product_#	quantity_#
1 ☒	5742 orders 3X21	79
2 ☐	5742 orders 3X21	100
3 ☒	5742 orders 3X33	79
4 ☒	—	4
5 ☐	5790 orders 3X33	—

Customer_Order_Line must have requested exactly one Quantity.

Quantity may be requested for any number of Customer_Order_Lines.

Comment:

28 Customer_Order is achieved.	
Source # 25	Customer_Order customer_order_#
Yes	
1	☒ 5742
2	☒ 5788
3	☒ -
4	☐
5	☐
Any number of Customer_Orders may be achieved.	
Comment:	

Conceptual Information Model

Deliverable 3

The primary deliverables are the constrained sentences in textual form and in a natural language model diagram. These communicate analysis results concisely to the designer.

“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.” (Reference 14; see “References” on page A-15).

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

Constrained Sentences

Deliverable 3

Note: * items have derivation rules.

Customer has Name

Customer is mandatory

Customer is unique

Customer has voice Phone

Customer is mandatory

Customer is unique

Phone is unique

Customer Order is created on Date

Customer Order is mandatory

Customer Order is unique

Customer Order is for Customer

Customer Order is mandatory

Customer Order is unique

Customer Order is sent on Date by Truck

Each Customer Order, Date combination is unique

Customer Order orders Product

Customer Order is mandatory

Each Customer Order, Product combination is unique

Customer Order was achieved

Customer receives deliveries at Address

Customer is mandatory

Customer is unique

Customer_Order_Line has available Quantity

Customer_Order_Line is mandatory

Customer_Order_Line is unique

Customer_Order_Line has ordered Quantity

Customer_Order_Line is mandatory

Customer_Order_Line is unique

Customer_Order_Line has requested Quantity

Customer_Order_Line is mandatory

Customer_Order_Line is unique

Delivered_Customer_Order exists

Product has current daily winery order Quantity

Product is mandatory

Product is unique

Product has existing winery on-order Quantity

Product is mandatory

Product is unique

Product has maximum stock Quantity

Product is mandatory

Product is unique

Product has minimum stock Quantity

Product is mandatory

Product is unique

Product has Name

Product is mandatory

Product is unique

Product has total Quantity *

Product is mandatory

Product is unique

Derived by rule 'total Quantity = sum(Winery Order Quantity) where Winery Order is received - sum(Customer Order Line has ordered Quantity) where Customer Order is a Delivered Customer Order'

Product is assigned in Quantity *

Product is mandatory

Product is unique

Derived by rule 'assigned Quantity = sum(Customer Order Quantity) where Customer Order Item = Item and not(Customer Order delivery was accomplished)'

Product is available in Quantity *

Product is mandatory

Product is unique

Derived by rule 'available Quantity = total Quantity - assigned Quantity'

Product is purchased from Winery

Product is mandatory

Product is unique

Received_Winery_Order is received on Date

Received_Winery_Order is mandatory

Received_Winery_Order is unique

Winery has fax Phone

Winery is mandatory

Winery is unique

Winery has Name

Winery is mandatory

Winery is unique

Winery Order has Date

Winery Order is mandatory

Winery Order is unique

Winery Order is for Winery

Winery Order is mandatory

Winery Order is unique

Winery Order orders Product in Quantity

Winery Order is mandatory

Each Winery Order, Product combination is unique

Natural Language Model Diagram

Deliverable 3

We show a Natural-language Information Analysis Methodology (NIAM) model, which is one form of a conceptual model in diagram form. This diagram provides the designer 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 4 NIAM Model

Logical Information Model—IDEF1X Diagram

Deliverable 4

The 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). IDEF stands for Integrated DEFinition method. The original concept has been expanded from the IDEF0 (Process Modeling) to other models such as IDEF1X, used to model/design processes as carried out by an automated process. 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 when 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)
———• or ——•	1: 0 or more

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.

The following elementary processes define the Create/Read/Update/Delete (CRUD) processes used on each core object. They form the basis for the CRUD matrix in Deliverable 7, and they can be generated from the Information Model.

Note: * items are processes which include the enforcement of the mandatory relationships between tables. These are shown in the IDEF1X diagram as P. (For example, the winery_order cannot be created until at least one winery_order_line exists; the customer_order cannot be created until at least one customer_order_line exists.)

Process Definition from Data Model

Create_Objects

Create_Customer

Create: Customer (id)
Create: has Name
Create: has voice Phone
Create: receives deliveries at Address

Create_Customer_Order *

Create: Customer Order (id)
Create: is created on Date
Create: is for Customer
Create: was achieved
until no more Customer Order Line to create
 Create_Customer_Order_Line for this Customer Order (id)

Create_Customer_Order_Line

Create: Customer Order (id)
Create: orders Product (id)
Create: has requested Quantity
Create: has available Quantity
Create: has ordered Quantity

Create_Customer_Order_is_sent_on_Date

Create: Customer Order (id)
Create: is sent on Date
Create: by Truck (id)

Create_Delivered_Customer_Order

Create: Customer Order (id)

Create_Product

Create: Product (id)
Create: has current daily winery order Quantity
Create: has existing winery on-order Quantity
Create: has maximum stock Quantity
Create: has minimum stock Quantity
Create: has Name
Create: has total Quantity
Create: is assigned in Quantity
Create: is available in Quantity
Create: is purchased from Winery

Create_Received_Winery_Order

Create: Winery Order
Create: is received on Date

Create_Winery

Create: Winery (Phone)
Create: has fax Phone
Create: has Name

Create_Winery_Order *

Create: Winery Order (id)
Create: is placed on Date
Create: is for Winery
until no more Winery orders Product to create
Create_Winery_Order_orders_Product for this Winery_Order (id)

Create_Winery_Order_orders_Product

Create: Winery Order (id)
Create: orders Product (id)
Create: in Quantity

Read_Objects**Read_Objects_Customer**

Read: Customer (id)
Read: has Name
Read: has voice Phone
Read: receives deliveries at Address

Read_Customer_Order *

Read: Customer Order (id)
Read: is created on Date
Read: is for Customer
Read: was achieved
until no more Customer Order Line to read
 Read_Customer_Order_Line for this Customer Order (id)

Read_Customer_Order_Line

Read: Customer Order (id)
Read: orders Product (id)
Read: has requested Quantity
Read: has available Quantity
Read: has ordered Quantity

Read_Customer_Order_is_sent_on_Date

Read: Customer Order (id)
Read: is sent on Date
Read: by Truck (id)

Read_Delivered_Customer_Order

Read: Customer Order (id)

Read_Product

Read: Product (id)
Read: has current daily winery order Quantity
Read: has existing winery on-order Quantity
Read: has maximum stock Quantity
Read: has minimum stock Quantity
Read: has Name
Read: has total Quantity
Read: is assigned in Quantity
Read: is available in Quantity
Read: is purchased from Winery

Read_Received_Winery_Order

Read: Winery Order
Read: is received on Date

Read_Winery

Read: Winery (Phone)
Read: has fax Phone
Read: has Name

Read_Winery_Order *

Read: Winery Order (id)

Read: is placed on Date

Read: is for Winery

until no more Winery orders Product to read

Read_Winery_Order_orders_Product for this Winery_Order (id)

Read_Winery_Order_orders_Product

Read: Winery Order (id)

Read: orders Product (id)

Read: in Quantity

Update_Objects**Update_Customer**

Read: Customer (id)

Update: has Name

Update: has voice Phone

Update: receives deliveries at Address

Update_Customer_Order *

Read: Customer Order (id)

Update: is created on Date

Update: is for Customer

Update: was achieved

until no more Customer Order Line Updates

Create_Customer_Order_Line for this Customer Order (id) or

Update_Customer_Order_Line for this Customer Order (id) or

Delete_Customer_Order_Line for this Customer Order (id)

Update_Customer_Order_Line

Read: Customer Order (id)

Read: orders Product (id)

Update: has requested Quantity

Update: has available Quantity

Update: has ordered Quantity

Update_Customer_Order_is_sent_on_Date

Read: Customer Order (id)

Read: is sent on Date

Update: by Truck (id)

Update_Delivered_Customer_Order

Read: Customer Order (id)

Update_Product

Read: Product (id)
Update: has current daily winery order Quantity
Update: has existing winery on-order Quantity
Update: has maximum stock Quantity
Update: has minimum stock Quantity
Update: has Name
Update: has total Quantity
Update: is assigned in Quantity
Update: is available in Quantity
Update: is purchased from Winery

Update_Received_Winery_Order

Read: Winery Order (id)
Update: is received on Date

Update_Winery

Read: Winery (Phone)
Update: has fax Phone
Update: has Name

Update_Winery_Order *

Read: Winery Order (id)
Update: is placed on Date
Update: is for Winery
until no more Winery Order orders Product to update
 Create_Winery_Order_orders_Product for this Winery_Order (id) or
 Update_Winery_Order_orders_Product for this Winery_Order (id) or
 Delete_Winery_Order_orders_Product for this Winery_Order (id)

Update_Winery_Order_orders_Product

Read: Winery Order (id)
Read: orders Product (id)
Update: in Quantity

Delete_Objects**Delete_Customer**

Delete: Customer (id)

Delete_Customer_Order *

Delete: Customer Order (id)
until no more Customer Order Line
 Delete_Customer_Order_Line for this Customer Order (id)

Delete_Customer_Order_Line *

If count > 1 or Customer Order (id) is deleted

Delete: Customer Order (id) with

Delete: orders Product (id)

Delete_Customer_Order_is_sent_on_Date

Delete: Customer Order (id)

Delete: is sent on Date

Delete_Delivered_Customer_Order

Delete: Customer Order (id)

Delete_Product

Delete: Product (id)

Delete_Received_Winery_Order

Delete: Winery Order (id)

Delete_Winery

Delete: Winery (Phone)

Delete_Winery_Order *

Delete: Winery Order (id)

until no more Winery Order orders Product to delete

Delete_Winery_Order_orders_Product for this Winery Order (id)

Delete_Winery_Order_orders_Product *

If count > 1 or Winery Order (id) is deleted

Delete: Winery Order (id) with

Delete: orders Product (id)

Presentation Sets

Deliverable 6

Presentation sets consist of one Process/Actor State Transition Matrix per core entity, the completed storyboard diagrams, and the event and process lists:

- The Process/Actor State Transition Matrix will add rigorous definitions of the processes related to each presentation set or object. In addition, from the customer's viewpoint, it helps integrate the actions performed and the state of the information system for each elementary process.
- 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 feel" with other systems.
- Event and process lists are included to define the flow, derivations, and actions as related to the storyboards.

Process/Actor State Transition Matrix

Deliverable 6

Description

- Add-on tool for your favorite Systems Analysis and Design methodology
- One matrix for each major database entity, or each major object (see “Logical Information Model—IDEF1X Diagram” on page A-29)
- One column for each major actor (actors can be people or software modules/systems)
- One row for each major customer or system task (each thing that can happen to the entity/object)
- Column for Beginning state (status) on the left; Ending state (status) on the right
- Not every row results in a state (status) change
- Cells show what each actor does

Uses

- Help customers and developers understand the system
- Help define the requirements
- Help think through business needs
- Provide systematic, structured framework for process analysis
- Get more detail in specifications
- Help all parties communicate about requirements and specifications
- Capture knowledge about the system for managers, customers, future maintainers

Inputs Needed

- Context diagram
- Core requirements (more may be found later)
- Core data entities (more may be found later)
- List of players (actors)
- List of tasks or events
- List of processes

Skills/Guidelines for Writing

- Listen
- Understand customers' needs
- Communicate clearly in writing, with precision and conciseness
 - Use customer's terminology
 - Avoid computer jargon
 - Avoid methodology's jargon

Table 5 Customer Process/Actor State Transition Matrix
 (Valid Status List: Null, Active, Inactive)

Object Life Cycle Event (Task)	Beginning State	Actions Taken—By Business Function					Elementary Processes	Ending State
	Incoming Status	Customer	Order Clerk	Dispatcher	Stock Clerk	Japan Wines Inc. Application Program		Resulting Status
Add a new customer	Null	Phones Japan Wines, Inc. to begin doing business	Enters phone number, If not found, enters customer info	n/a	n/a	Attempts retrieve of customer info If not found, create customer	Read customer Create customer	Active
Confirm customer info	Active	Phones Japan Wines, Inc. to ask if address is correct	Enters phone number	n/a	n/a	Retrieves customer info	Read customer	Active
Change info on existing customer	Active	Phones Japan Wines, Inc. with a name or address change	Enters phone number Enters new name or delivery address	n/a	n/a	Retrieves customer info Updates customer	Read customer Update customer	Active
Mark customer as Inactive	Active	n/a	n/a	n/a	n/a	Reads all customers If any line had no orders in 5 years, marks customer as Inactive	Read customer Update customer	Active
Delete a customer, or periodic purge of all inactive customers	Inactive	Outside the project's scope						Null

Table 6 Customer Order Process/Actor State Transition Matrix
(Valid Status List: Null, Accepted, Achieved, Delivered)

Object Life Cycle Event (Task)	Beginning State	Actions Taken—By Business Function					Elementary Processes	Ending State
	Incoming Status	Customer	Order Clerk	Dispatcher	Stock Clerk	Japan Wines Inc. Application Program		Resulting Status
Initiate order	Null	Phones Japan Wines Inc. to place an order	Enters phone number on Customer Order screen If not found, enters customer info	n/a	n/a	Attempts retrieve of customer info If found, creates order If not found, creates customer & creates order	Read customer Create customer order Create customer Create customer order	Accepted
Initiate, change, or delete line items for order	Note: See the Process/Actor State Diagrams for the Customer Order Line (Table 7) and Product (Table 10) objects							
Answer inquiry about order	Accepted	Calls to ask when order will be sent	Enters phone number or customer order number	n/a	n/a	Retrieves order info	Read customer and/or Read customer order	
Mark order assigned for delivery (achieved)	Accepted	n/a	n/a	Enters a delivery truck number for a date on Delivery Assignment screen	n/a	Updates order by changing was_achieved	Update customer order	Achieved
Mark order as delivered (accomplished)	Achieved	n/a	n/a	Enters order number on Delivery Logged screen	n/a	Updates order by changing was_delivered	Update customer order	Delivered
Delete Order		Outside the project's scope						

Table 7 Customer Order Line Process/Actor State Transition Matrix
(Valid Status List: Null, Active)

Object Life Cycle Event (Task)	Beginning State	Actions Taken—By Business Function					Elementary Processes	Ending State
		Customer	Order Clerk	Dispatcher	Stock Clerk	Japan Wines Inc. Application Program		
	Incoming Status							Resulting Status
Initiate line item info	Null	Calls & gives product number & how many of it he/she wants	Enters quantity ordered Note: If not enough stock is available, the Adjust Quantity Requested screen will pop up & allow the quantity to be changed	n/a	n/a	Retrieves product name	Create customer order line	Active
Answer inquiry about order line	Active	Calls to ask how many of a product were ordered	Enters phone number or order number	n/a	n/a	Retrieves line items for this order	Read customer or Read customer order	Active
Change line item quantity because of customer request	Active	Calls to change quantity on a line	Selects line to change Enters new quantity on line item	n/a	n/a	Retrieves line item(s) for order Verifies that order is not yet delivered Calculates quantity available Updates line item	Update customer order line	Active
Adjust Line Item screen pops up as clerk tabs out of quantity when taking an order	Active	n/a	Tells customer how many are available & inputs new quantity or cancels line	n/a	n/a	Calculates available quantity May create line item	Update customer order line	Active
Delete Line Item		Calls to say product is not wanted, or to change product to another	Selects line to delete Clicks delete button	n/a	n/a	Retrieves line items for customer order Deletes old line Possibly creates new line	Delete customer order line May create customer order line	Null Old line Null, new line Active

Table 8 Customer Order Shipment Process/Actor State Transition Matrix
 (Valid Status List: Null, Active)

Object Life Cycle Event (Task)	Beginning State	Actions Taken—By Business Function					Elementary Processes	Ending State
	Incoming Status	Customer	Order Clerk	Dispatcher	Stock Clerk	Japan Wines Inc. Application Program		Resulting Status
An order is placed on a truck to be sent out for delivery	Null	n/a	n/a	n/a	n/a	Records date & truck number for an attempted delivery of a customer order	Create customer order shipment	Active
Correct data entry error-truck number	Active	Outside the project's scope						Active
Answer customer request for when/how order was shipped	Active	Calls to find out status of order	n/a	n/a	n/a		Read customer order shipment	Active
Delete erroneous customer order shipment	Active	Outside the project's scope						Null

Table 9 Winery Process/Actor State Transition Matrix
 (Valid Status List: Null, Active)

Object Life Cycle Event (Task)	Beginning State	Actions Taken—By Business Function						Elementary Processes	Ending State
	Incoming Status	Customer	Winery Sales Staff	Order Clerk	Dispatcher	Stock Clerk	Japan Wines Inc. Application Program		Resulting Status
Add a new winery	Null	Calls to order a product from a winery we don't do business with	Visits or sends catalog with products we want to carry	Responds to customer by entering winery phone number Enters winery name & fax/ phone	n/a	Responds to winery staff by entering winery phone number	Attempts retrieve of winery phone number, but it is not found, Creates winery	Create winery	Active
Change info on existing winery	Active	n/a	Fax/phone or winery name has changed	n/a	n/a	Enters changed winery info	Retrieves winery info Updates winery	Update winery	Active
Delete a winery, or periodic purge of all inactive wineries		Outside the project's scope							

Table 10 Product Process/Actor State Transition Matrix
(Valid Status List: Null, Active)

Object Life Cycle Event (Task)	Beginning State	Actions Taken—By Business Function						Elementary Processes	Ending State
		Customer	Winery Sales Staff	Order Clerk	Dispatcher	Stock Clerk	Japan Wines Inc. Application Program		
	Incoming Status								Resulting Status
Add a new Product	Null	Phones to order a product we don't carry	Visits or sends catalog with new product	Enters product name & info,	n/a	Enters new product name & info	Adds a new product	Create product	Active
Assign minimum & maximum stock quantities	Active	n/a	May recommend stock quantities	n/a	n/a	Enters product name Enters quantities	Retrieves product, changes quantities	Update product	Active
Create & print product inventory status report	Active	n/a	n/a	n/a	n/a	n/a	Each weekday at 7:30 AM, retrieve info on each product & print report Calculate total number of product on hand Calculate total product committed to customer orders Calculate available product	Read product	Active
Delete product		Outside the project's scope							

Table 11 Winery Order Process/Actor State Transition Matrix
 (Valid Status List: Null, Active, Faxed)

Object Life Cycle Event (Task)	Beginning State	Actions Taken—By Business Function					Elementary Processes	Ending State
	Incoming Status	Customer	Order Clerk	Dispatcher	Stock Clerk	Japan Wines Inc. Application Program		Resulting Status
Place line items on order	Null	Note: See the Process/Actor State Diagrams for the Winery Order Line (Table 12) and Product (Table 10) objects						Active
Mark order as faxed to winery	Active	Fax order to winery—verify that fax winery order report was successfully sent to winery Update order placed date					Create winery order	Faxed
Receive shipment	Faxed	n/a	n/a	n/a	Enters quantities received on Receive Winery Shipment screen	Updates order received date	Create received winery order	Faxed
Delete order		Outside the project's scope						

Table 12 Winery Order Line Process/Actor State Transition Matrix
(Valid Status List: Null, Active)

Object Life Cycle Event (Task)	Beginning State	Actions Taken—By Business Function						Elementary Processes	Ending State
		Customer	Winery Sales Staff	Order Clerk	Dispatcher	Stock Clerk	Japan Wines Inc. Application Program		
Create line item for each product to be ordered	Null	n/a	n/a	n/a	n/a	n/a	At 5:30 PM each Wednesday, as part of creating winery order, read each product by winery Calculate quantity to order based on available quantity, on-order quantity, & minimum & maximum stock levels	Read winery order line Create winery order line	Active
Change line item quantity	Active	Outside the project's scope							Active
Answer inquiries from winery	Active	n/a	Phones to request new copy of order	n/a	n/a	n/a	Retrieves info about products on order from a winery	Read winery order line	Active
Delete line item		Outside the project's scope							

Storyboards

Deliverable 6

The format for these diagrams may be penciled or computerized drawings or electronic prototype screens. Development of new requirements should be added easily to the format chosen with the complete diagrams available in hard copy, at a minimum. These diagrams evolve from the initial storyboards or cartoons in Deliverable 2 throughout the analysis and design processes.

Corporate Functions

Product Accountability Icon—I1

Product
Accountability

Product Accountability Menu—M1

Japan Wines, Inc.
Product Accountability

Customer Order

Assign Truck

Delivery Accomplished

Receive Winery Shipment

Add/Update Product

Add/Update Customer

Add/Update Winery

Cancel

Customer Order Screen—S1

Customer Order														
Customer Order No.	<input type="text" value="5472"/>	Date <input type="text" value="7-Sep-94"/>												
Customer No.	<input type="text" value="89"/>													
Customer Name	<input type="text" value="Joe's Bar"/>													
Phone	<input type="text" value="721 3652"/>													
Delivery Address	<input type="text" value="111 First St."/>													
<table><thead><tr><th>Product No.</th><th>Product</th><th>Quantity</th></tr></thead><tbody><tr><td>3X21</td><td>green chile wine</td><td>76</td></tr><tr><td>5271</td><td>red wine</td><td>21</td></tr><tr><td>7631</td><td>white wine</td><td>5</td></tr></tbody></table>			Product No.	Product	Quantity	3X21	green chile wine	76	5271	red wine	21	7631	white wine	5
Product No.	Product	Quantity												
3X21	green chile wine	76												
5271	red wine	21												
7631	white wine	5												
<input type="button" value="Cancel"/>		<input type="button" value="OK"/>												

Adjust Quantity Requested Screen—S2

Adjust Quantity Requested

Product No.

7631

Product

white wine

Quantity Requested

5

Quantity Available

3

Quantity Ordered

3

Cancel

OK

Inventory Report—R1**Inventory Status**

Date 7-Sep-94

Product #	Name	Available Quantity	Assigned Quantity	Total Quantity	Minimum Stock Qty.	Maximum Stock Qty.	Existing On-Order Qty.
3X21	green chile wine	100	0	100	50	100	0
5721	red wine	50	20	70	60	150	105
7631	white wine	5	55	60	30	75	40

Delivery Truck Assignment Screen—S3

Delivery Truck Assignment														
Unfilled & Unassigned Customer Orders		<input style="width: 50px;" type="text" value="12"/>												
Truck	<input style="width: 80px;" type="text" value="T1"/>	Date <input style="width: 100px;" type="text" value="8-Sep-94"/>												
Customer Order No.	<input style="width: 100px;" type="text" value="5472"/>													
Customer No.	<input style="width: 100px;" type="text" value="85"/>													
Customer Name	<input style="width: 100px;" type="text" value="Joe's Bar"/>													
Phone	<input style="width: 100px;" type="text" value="721 3652"/>													
Delivery Address	<input style="width: 100px;" type="text" value="111 First St."/>													
<table style="width: 100%; border-collapse: collapse;"><thead><tr><th style="text-align: left; width: 20%;">Product No.</th><th style="text-align: left; width: 40%;">Product</th><th style="text-align: left; width: 40%;">Quantity</th></tr></thead><tbody><tr><td>3X21</td><td>green chile wine</td><td>76</td></tr><tr><td>5271</td><td>red wine</td><td>21</td></tr><tr><td>7631</td><td>white wine</td><td>5</td></tr></tbody></table>			Product No.	Product	Quantity	3X21	green chile wine	76	5271	red wine	21	7631	white wine	5
Product No.	Product	Quantity												
3X21	green chile wine	76												
5271	red wine	21												
7631	white wine	5												
<input style="width: 100px;" type="button" value="Cancel"/>		<input style="width: 100px;" type="button" value="OK"/>												

Delivery Instructions Report—R2

Delivery Instruction Ticket		
Truck	<input type="text" value="T1"/>	Date <input type="text" value="8-Sep-94"/>
Customer Order No.	<input type="text" value="5472"/>	
Customer No.	<input type="text" value="85"/>	
Customer Name	<input type="text" value="Joe's Bar"/>	
Phone	<input type="text" value="721 3652"/>	
Delivery Address	<input type="text" value="111 First St."/>	
Product No.	Product	Quantity
3X21	green chile wine	76
5271	red wine	21
7631	white wine	5
Completed	☐	Re-Deliver ☐

Delivery Accomplished Screen—S4

A screenshot of a software dialog box titled "Delivery Accomplished". The dialog box has a light gray background and a black border. It contains the following elements:

- The title "Delivery Accomplished" is centered at the top.
- Below the title, the text "Customer Order No." is followed by a text input field containing the value "5472".
- At the bottom, there are two buttons: "Cancel" on the left and "OK" on the right.

Log Another Delivery Screen—S5

A rectangular dialog box with a thin black border. Inside the box, the text "Log Another Delivery?" is centered at the top. Below the text, there are two rectangular buttons. The button on the left is labeled "Cancel" and the button on the right is labeled "OK". Both buttons have a thin black border and are positioned side-by-side.

Winery Order Report—R3

Winery Order Japan Wines, Inc.			
Customer Order No.	7555	Date	7-Sep-94
Winery	Sam's Winery		
Phone	321 9987		
Fax	321 8789		
Product No.	Product	Quantity	
3X21	green chile wine	100	
5271	red wine	121	
7631	white wine	75	

Receive Winery Shipment Screen—S6

Receive Winery Shipment		
Order Request No.	<input type="text" value="7555"/>	Date <input type="text" value="9-Sep-94"/>
Winery Name	<input type="text" value="Sam's Winery"/>	
Winery Phone	<input type="text" value="321 9987"/>	
Product No.	Product	Quantity
3X21	green chile wine	100
5271	red wine	121
7631	white wine	75
<div><input type="button" value="Cancel"/> <input type="button" value="OK"/></div>		

Add/Update Product Information Screen—S7

Add/Update Product Information

Product No.

Product Name

Assigned Quantity

Total Quantity

Minimum Stock Quantity

Maximum Stock Quantity

Purchased from
Winery (phone)

Add Winery Screen—S8

Add Winery Screen

Winery (phone)

Winery Name

Winery Fax

Cancel

OK

Add/Update Customer Information Screen—S9

Add/Update Customer Information	
Customer No.	<input type="text" value="89"/>
Customer Name	<input type="text" value="Joe's Bar"/>
Phone	<input type="text" value="721 3652"/>
Delivery Address	<input type="text" value="111 First St."/>
<div><input type="button" value="Cancel"/><input type="button" value="OK"/></div>	

Add/Update Winery Information Screen—S10

Add/Update Winery Information

Winery (phone)

Winery Name

Winery Fax

Event List

Deliverable 6

The event list, together with the storyboard diagrams and process list, defines the flow of control. The **Sxx**, **Mxx**, and **Rxx** numbers refer to the screen layouts, menus, and reports in the storyboard diagrams (see “Storyboards” on page A-48). The **Pxx** numbers refer to the processes defined in the process list (see “Process List” on page A-69). Events are numbered, **Exx**, for traceability of the implementation.

Note: We include the event list in Issue A for completeness. The format may change in Issue B, possibly into a matrix that shows event numbers, process numbers, and fact numbers. This format may better communicate to developers the relationships between the lists.

Events

E1

In <Corporate Functions> if <Product Accountability Icon - I1 is clicked> then
<display Product Accountability Menu - P33>

E2

In <Product Accountability Menu - M1> if <Customer Order button is clicked> then
<Initiate Customer Order - P1>

E3

In <Product Accountability Menu - M1> if <Assign Truck button is clicked> then
<Initiate Truck Assignment - P2>

E4

In <Product Accountability Menu - M1> if <Delivery Accomplished button is
clicked> then <Specify Delivery Accomplished - P3>

E5

In <Product Accountability Menu - M1> if <Receive Winery Shipment button is
clicked> then <Receive Winery Shipment - P4>

E6

In <Product Accountability Menu - M1> if <create/update Product button is clicked>
then <Initiate create/update Product - P5>

E7

In <Product Accountability Menu - M1> if <create/update Customer button is
clicked> then <Initiate create/update Customer - P6>

E8

In <Product Accountability Menu - M1> if <create/update Winery button is clicked>
then <Initiate create/update Winery - P7>

E9

In <Customer Order Screen - S1> if <Customer Phone Number entered> then <read
Customer Information - P8>

E10

In <Customer Order Screen - S1> if <Retrieve Customer Information - P8 did not
return a Customer #> then <create Customer Address - P9>

E11

In <Customer Order Screen - S1> if <Product No. entered> then <read Product Name
- P10>

E12

In <Customer Order Screen - S1> if <Quantity entered> then <read Available Quantity - P11>

E13

In <Customer Order Screen - S1> if <Available Quantity less than requested Quantity> then <Initiate Adjust Quantity Requested - P12>

E14

In <Adjust Quantity Requested Screen - S2> if <OK Button Clicked and Quantity Ordered is not zero> then <Adjust Quantity Requested - P13>

E15

In <Adjust Quantity Requested Screen - S2> if <Cancel Button Clicked> then <Delete Line Item - P14>

E16

In <Adjust Quantity Requested Screen - S2> if <OK Button Clicked and Quantity Ordered is equal to zero> then <display Customer Order Screen - S1 & Delete Line Item - P14>

E17

In <Customer Order Screen - S1> if <OK Button Clicked> then <Accept Customer Order - P15>

E18

In <Customer Order Screen - S1> if <Cancel Button Clicked> then <display Product Accountability Menu - M1>

E19

In <Accept Customer Order - P15> if <Assigned Quantity changes> then <Adjust Available Quantity - P16>

E20

In <Product Accountability System> if <7:30 am on Weekday> then <Print Inventory Report - P17>

E21

In <Delivery Truck Assignment Screen - S3> if <Delivery Date Entered> then <Initiate Delivery Truck Assignment - P19>

E22

In <Delivery Truck Assignment Screen - S3> if <OK Button Clicked & FT22 greater than one> then <Initiate Delivery Truck Assignment - P19>

E23

In <Delivery Truck Assignment Screen - S3> if <OK Button Clicked & FT22 equals one> then <display Product Accountability Menu - M1>

E24

In <Delivery Truck Assignment Screen - S3> if <OK Button Clicked> then <Assign Delivery Truck - P20>

E25

In <Delivery Truck Assignment Screen - S3> if <OK Button Clicked> then <Print Delivery Instruction Ticket - P21>

E26

In <Delivery Truck Assignment Screen - S3> if <Cancel Button Clicked> then <display Product Accountability Menu - M1>

E27

In <Delivery Accomplished Screen - S4> if <OK Button Clicked> then <display Log Another Delivery Screen - S5 & Log Delivery Accomplished - P22>

E28

In <Delivery Accomplished Screen - S4> if <Cancel Button Clicked> then <display Product Accountability Menu - M1>

E29

In <Log Another Delivery Screen - S5> if <OK Button Clicked> then <display Delivery Accomplished Screen - S4>

E30

In <Log Another Delivery Screen - S5> if <Cancel Button Clicked> then <display Product Accountability Menu - M1>

E31

In <Product Accountability System> if <5:30 PM on Weekday> then <Calculate Product Order Amounts - P23>

E32

In <Product Accountability System> if <Calculate Product Order Amounts - P23 completed> then <Fax Winery Order Report - R3 - P24>

E33

In <Receive Winery Shipment Screen - S6> if <OK Button Clicked> then <Receive Winery Shipment - P25>

E34

In <Receive Winery Shipment Screen - S6> if <Cancel Button Clicked> then <display Product Accountability Menu - M1>

E35

In <create/update Product Information Screen - S7> if <Winery (phone) entered & it is New> then <Enter New Winery - P26>

E36

In <create Winery Information Screen - S8> if <OK Button Clicked> then <create Winery Information - P27>

E37

In <create Winery Information Screen - S8> if <Cancel Button Clicked> then <return to create/update Product Information Screen - S7>

E38

In <create/update Product Information Screen - S7> if <OK Button Clicked &P5 returned a Product Name> then <update Product Information - P28>

E39

In <create/update Product Information Screen - S7> if <OK Button Clicked &P5 did not return a Product Name> then <create Product Information - P29>

E40

In <create/update Product Information Screen - S7> if <Cancel Button Clicked> then <display Product Accountability Menu - P33>

E41

In <create/update Customer Information Screen - S9> if <OK Button Clicked &P6 returned a Customer Name> then <update Customer Information - P30>

E42

In <create/update Customer Information Screen - S9> if <OK Button Clicked &P6 did not return a Customer Name> then <create Customer Information - P31>

E43

In <create/update Customer Information Screen - S9> if <Cancel Button Clicked> then <display Product Accountability Menu - P33>

E44

In <create/update Winery Information Screen - S10> if <OK Button Clicked &P7 returned a Winery Name> then <update Winery Information - P32>

E45

In <create/update Winery Information Screen - S10> if <OK Button Clicked &P7 did not return a Winery Name> then <create Winery Information - P27>

E46

In <create/update Winery Information Screen - S10> if <Cancel Button Clicked> then <display Product Accountability Menu - P33>

Process List

Deliverable 6

The process list, together with the storyboard diagrams and event list, defines the flow of control. The **Sxx**, **Mxx**, and **Rxx** numbers refer to the screen layouts, menus, and reports in the storyboard diagrams (see "Storyboards" on page A-48). The **FTxx** numbers refer to the facts in the formal natural language sentences (see "Business Rules" on page A-8). Processes are numbered, **Pxx**, for traceability of the implementation.

Note: We include the process list in Issue A for completeness. The format may change in Issue B, possibly into a matrix that shows event numbers, process numbers, and fact numbers. This format may better communicate to developers the relationships between the lists. Also, in the future we will reference the CRUD processes generated and shown in Deliverable 5.

Processes

- P1 Initiate Customer Order
 display Customer Order Screen - S1
 establish unique Customer Order #
 establish Order Date
 establish Customer Phone {from caller id, if available}
 enter Customer Phone
- P2 Initiate Truck Assignment
 display Delivery Truck Assignment Screen - S3
 establish Delivery Date
- P3 Specify Delivery Accomplished
 display Delivery Accomplished Screen - S4
 enter Customer Order No.
- P4 Receive Winery Shipment
 display Receive Winery Shipment Screen - S6
 establish Date
 enter Winery Order Request No.
 read FT14
 read FT15
 read each FT20
 read each FT6
 validate quantities received match order

-
- P5 Initiate create/update Product
display create/update Product Information Screen - S7
enter Product No.
read FT6
read FT9
read FT10
read FT11
read FT12
read FT19
if desired
enter Product Name
enter Assigned Quantity
enter Total Quantity
enter Minimum Stock Quantity
enter Maximum Stock Quantity
enter Winery (phone)
- P6 Initiate create/update Customer
display create/update Customer Information Screen - S9
enter Phone
read FT3
read FT4
read FT5
if desired
enter Phone
enter Name
enter Delivery Address
- P7 Initiate create/update Winery
display create/update Winery Information Screen - S10
enter Phone
read FT15
read FT16
if desired
enter Name
enter Fax
- P8 read Customer Address
read F4
read F3
read F5

-
- P9 create Customer Address
 generate unique Customer No.
 create F4
 create F3
 create F5
- P10 read Product Name
 read F6
- P11 Check Available Quantity
 read F8
- P12 Initiate Adjust Quantity Requested
 display Adjust Quantity Requested Screen - S2
 copy Product No.
 copy Product
 copy FT25 to FT27
 read F8, copy to FT26
 copy FT26 to FT25
 Enter FT25 (if needed and less than or equal to FT26)
- P13 Adjust Quantity Requested
 display Customer Order Screen - S1
 copy FT25 to Quantity for Product No.
- P14 Delete Line Item
 display Customer Order Screen - S1
 For Product No.
 clear Product No.
 clear Product Name
 clear Quantity
- P15 Accept Customer Order
 create FT1
 create FT2
 create Each FT7, FT25, FT26, FT27
 update Each FT9, Assigned Quantity = Assigned Quantity + FT25
- P16 Available Quantity (FT8)
 read FT9
 read FT10
 Available Quantity = Total Quantity - Assigned Quantity
 update FT8
-

-
- P17 Print Inventory Report
 establish Date
 read Each FT6
 read Each FT8
 read Each FT9
 read Each FT10
 read Each FT11
 read Each FT12
 read Each FT18
 Print R1
- P18 Count Unfilled Customer Orders (FT22)
 Count --- Orders minus (Delivered Orders plus Assigned Orders)
- P19 Initiate Delivery Truck Assignment
 establish FT22 (Not delivered and not assigned)
 establish Customer Order Number (Not delivered and not assigned)
 read FT2
 read FT3
 read FT4
 read FT5
 read Each FT7, FT25
 read Each FT6
 enter Truck Number
- P20 Assign Delivery Truck
 create FT21
- P21 Print Delivery Instruction Ticket
 copy FT2
 copy FT3
 copy FT4
 copy FT5
 copy Each FT7, FT25
 copy Each FT6
 copy FT21
 Print R2
- P22 Log Delivery Accomplished
 create FT23

-
- P23 Calculate Product Order Amounts (FT17 and FT18)
 for every item
 read FT9
 read FT10
 read FT11
 read FT12
 Current Daily Winery Order Quantity = Maximum Stock Quantity -
 (Available Quantity + Existing Winery On-Order Quantity)
 if Available Quantity < Minimum Stock Quantity
 else Winery Order Quantity = 0
 Existing Winery On-Order Quantity = Existing Winery On-Order Quantity +
 Current Daily Winery Order Quantity
 update FT17
 update FT18
- P24 Fax Winery Order Report
 for each Winery with Product that has Current Daily Winery Order
 Quantity > 0
 establish unique Order Request No.
 establish Date
 read FT15
 read FT16
 read each FT17
 read each FT6
 create FT13
 create FT14
 create each FT20
 Fax
- P25 Receive Winery Shipment
 create FT24
 for each product ordered by Winery Request No.
 read FT10
 Total Quantity = Total Quantity + Quantity
 update FT10
 calculate FT8
 update FT8
 read FT18
 Existing Winery On-Order Quantity = Existing Winery On-Order Quantity -
 Quantity
 update FT18

-
P26	Enter New Winery
display create Winery Information Screen - S9
copy Winery (phone)
enter Winery Name
enter Winery Fax
P27	create Winery Information
create FT15
create FT16
display create/update Product Information Screen - S7
P28	update Product Information
update FT6
update FT9
update FT10
update FT11
update FT12
update FT19
P29	create Product Information
create FT6
create FT9
create FT10
create FT11
create FT12
create FT19
P30	update Customer Information
update FT3
update FT4
update FT5
P31	create Customer Information
create FT3
create FT4
create FT5
P32	update Winery Information
update FT15
update FT16
P33	display Product Accountability Menu
display Product Accountability Menu - M1

Distribution:

	MS 9018	Central Technical Files, 8523-2
5	MS 0899	Technical Library, 13414
	MS 0619	Print Media, 12615
2	MS 0199	Document Processing, 7613-2 For DOE/OSTI
	MS 0630	Michael J. Eaton, 13100
	MS 0803	John F. Jones, 13200
	MS 0629	Michael G. Robles, 13300
	MS 0622	L. Herbert Pitts, 13400
	MS 0801	Melissa J. Murphy, 13900
	MS 0898	R. Dennis Rowley, 13312
	MS 1098	Thomas L. Ferguson, 13313
	MS 1090	Joseph A. Ruggles, 13314
	MS 1090	Grant C. Claycomb, 13315
	MS 0661	Gary Rivord, 13316
	MS 0661	Michael H. Pendley, 13212
	MS 0809	Melville E. Mefford, 13913
	MS 0805	Dorothy S. Rarick, 13911
	MS 0813	Jim Hamilton, 13412
	MS 0807	R. Michael Cahoon, 13918
	MS 0622	Hank Witeck, 13902
	MS 1090	William D. Swartz, 13411
	MS 0806	Michael O. Vahle, 13216
	MS 0811	C. Douglas Brown, 13221
	MS 1098	Jim Stromberg, 13313
	MS 1098	Andrea Cassidy, 13313
	MS 0661	David K. Cuyler, 13922
	MS 0801	John K. Sharp, 13922
	MS 0801	Ronald Hall, 13922
	MS 1090	Mary Lynn Clark, 13411
	MS 0661	Scott Joyce, 13316
	MS 0661	Irene Thurston, 13316
10	MS 0661	Donna S. Eaton, 13316
	MS 0661	John Hatley, 13316
	MS 0661	Shelley Eaton, 13316
	MS 0661	Joe Schofield, 13316
	MS 0661	Elisa Kephart, 13316
	MS 0661	Charlene Strickland, 13316
	MS 0638	Dwayne Knirk, 12326
	MS 0629	Al Alvarado, 13300
	MS 0898	Carol Jones, 13312
	MS 0805	Bill Mertens, 13911
	MS 0813	T. C. Adams, 13913
	MS 1090	Nancy Marsh, 13314