Div. 642

NBSIR 79-1940 (R)

ICAM Software Documentation Standards

Center for Programming Science and Technology
Institute for Computer Sciences and Technology
National Bureau of Standards
Washington, D.C. 20234

Final Report

February 1980

Prepared for

**Air Force Materials Laboratory
Wright-Patterson Air Force Base
Ohio 45433**

NBSIR 79-1940 (R)

ICAM SOFTWARE DOCUMENTATION STANDARDS

Center for Programming Science and Technology
Institute for Computer Sciences and Technology
National Bureau of Standards
Washington, D.C. 20234

Final Report

February 1980

Prepared for
Air Force Materials Laboratory
Wright-Patterson Air Force Base
Ohio 45433

PREFACE

In October of 1977 the National Bureau of Standards agreed with the Air Force Wright Aeronautical Laboratories to develop documentation standards for the Integrated Computer-Aided Manufacturing (ICAM) Program. Under MIPR FY1457-77-02054 NBS's Institute for Computer Sciences and Technology undertook and completed this project. The documentation standards included in this final report--NBSIR 79-1940 (Air Force) (R)--have given NBS the opportunity to adapt and substantially extend Federal Information Processing Standards Publications 30, 38, and 64; moreover, they provide the ICAM Program with the means of ensuring high-quality software products.

TABLE OF CONTENTS

ICAM SOFTWARE DOCUMENTATION STANDARDS

Application Systems Division
Institute for Computer Sciences and Technology

The ICAM Program Office requires all contractors who develop ICAM software to comply with the following standards governing style and content of ICAM Software Documents. The Style Guide, which covers writing and programming style, applies to all documentation, and individual Content Guides apply to particular software documents. All of these guides function within the overall framework defined by the IDEF Functional Model and its glossary.

Key words: Computer-aided manufacturing; computer standards; programming style; software documentation; structured analysis.

1. APPLICATION INFORMATION

1.1 Summary

This report provides specifications and procedures for implementing software documentation standards in ICAM projects. These standards adapt and substantially extend Federal Information Processing Standards Publications 30, 38, and 64, and DOD 7935.1-S, for use in ICAM software contracts.

1.2 Environment

These instructions are directed to a technical staff member of the ICAM Program Office who serves as the Contract Officer to define requirements for ICAM software development and to monitor contractor performance of those requirements. The materials provided are to be implemented within any contract for ICAM software development.

1.3 References

This report includes four items. First is the contract statement of work attachment that defines the applicability of the standards and the pertinent contractor responsibilities. The others are the ICAM Software Documentation Style Guide, the ICAM Software Documentation Content Guides, and the IDEF Functional Model of the ICAM Software Documentation System. The Contract Officer and prospective contractor are referred as well to the FIPS PUBS 30, 38, and 64, and DOD 7935.1-S. (For complete references, see the introduction to the Content Guides, section 1.3, p. 23).

2. SCOPE

The statement of work presented below addresses only the documentation requirements associated with software development, and so is intended to supplement other contractual specifications. It provides space to record Contract Officer's decisions regarding the following: the packaging of standard documents, the contractor-proposed documents to be delivered, and the scheduled delivery dates for documentation items.

The standard documents are applicable and required for any contract that includes the activity which produces them; see the IDEF Model. But different options for packaging the documents as separate volumes may be chosen in order to match document scope and size to the scope and complexity of the project or software product.

The statement of work is written for a contract that covers only one system of related computer programs. The Contract Officer must adapt the wording appropriately if two or more distinct system developments are being undertaken in one contract.

3. IMPLEMENTATION

To incorporate the statement of work into a contract, the Contract Officer first assesses the proposed project using the Table of Software Categories on page 6 to choose one of the packaging options A, B, or C. Noting the start and end points of the project within either the software life cycle or the IDEF Functional Model, he marks (S) for single volume documents in the selected column A, B, or C of Table 1. If he chooses A, he marks (M) for those documents that would be acceptable as multiple volumes. He then completes the Table by filling in the scheduled delivery dates for outlines and full copy. Finally, he completes the Additional Requirements section on page 5 to include any contractor-proposed documents that are accepted for the contract, any other desired documents, and any special conditions on preparation.

The remainder of these instructions is the statement of work attachment.

4. STATEMENT OF WORK FOR SOFTWARE DOCUMENTATION REQUIREMENTS

The contractor shall prepare and deliver technical documents (software documentation) as required herein, for all computer programs (software) that are the subject of defined tasks or original work as identified elsewhere in this contract. The requirements below address objectives and general requirements, required documents, their information elements, quality and style of preparation, delivery schedule, and other pertinent factors. The following standards or guidelines are incorporated in this contract by reference and made a part of it for the purposes defined below: IDEF Functional Model of the ICAM Software Documentation System; ICAM Style Guide for Software Documentation; ICAM Software Documentation Content Guides.

Objectives and General Requirements

Among its important objectives, the ICAM program seeks to produce and to disseminate innovative computer programs that are readily transferred between different computer systems and are useful to many manufacturing organizations. Quality software documentation is essential for accomplishing diverse tasks, including installing, operating, maintaining, and using computer programs, as well as transferring design principles and methodologies to many parties. The contractor shall prepare software documents according to the requirements set forth herein, and with the technical accuracy, completeness, and other quality attributes that will best meet ICAM objectives for the software involved. The stated requirements are minimum essential characteristics. The contractor shall meet these fully and shall provide his best technical effort and judgment in proceeding beyond the stated requirements to produce highly effective software documents.

The IDEF Functional Model for the ICAM Software Documentation System shall be used by the contractor as basic guidance on document preparation, review, and acceptance procedures. Acceptance of software documents shall be based upon clarity and effectiveness of presentation, accuracy, completeness for the intended purpose, and conformance to requirements.

Required Standard Documents

The required documents consist of ICAM standard documents and additional contractor-proposed documents as stated under Additional Requirements below.

Standard software documents are those defined in the ICAM Software Documentation Content Guides. These are listed in Table 1--Standard Documents (pages 7-9), which also shows the activities that produce each document, as depicted in the IDEF Functional Model.

In Table 1, the three columns labeled A, B, and C provide for a choice among alternative configurations or packaging of the standard documents as separate volumes. One such column of Table 1 is marked to indicate the configuration that shall apply to this contract. The individual entries marked (S) or (M) in this column indicate the standard documents that must be provided because the pertinent life cycle tasks are encompassed by this contract. Documents marked "S" are acceptable only as single volumes.

If column A has been marked for this contract, then a mark (M) indicates that the pertinent document may be provided as a multiple volume document, at the discretion of the contractor or writer, following the guidance in the ICAM Content Guides.

Wherever a box marked in column B or C includes two or more consecutive documents, the contractor shall prepare one volume that provides each of these included documents as a separate part, and the parts shall be in the order shown in Table 1 (e.g., see the Feasibility Study and Cost/Benefit Analysis in columns B and C). See the ICAM Content Guides for further specification.

Table 1 further states the required delivery schedule for document outlines, review copy, and final copy.

Use of Style Guide

All documents pertaining to software shall be prepared according to the ICAM Software Documentation Style Guide, which governs narrative style, format, and other conventions that are independent of specific technical content.

The Style Guide also provides guidelines for computer programming style which shall be followed as a minimum to assure that program listings are readable source material to supplement narrative documents.

Use of Content Guides

Each ICAM Software Documentation Content Guide begins with a brief statement of the document's nature and purpose, and then provides an annotated outline of required contents. The information content of each standard document shall conform, as a minimum, to the pertinent annotated outline. The contractor shall use each Content Guide as a basic reference to determine the relevance and scope for information to be included. The contractor shall provide in each document all information that is essential and effective for understanding of the software by other parties and for fulfilling the purpose of the document, even if such information is not explicitly indicated by the pertinent Content Guide.

The Content Guide governs only the main body of the standard document. The Style Guide specifies how the main body shall be incorporated into a complete document that includes other essential features such as title page, table of contents, etc.

Additional Requirements

(The Contract Officer shall include here the appropriate terms and guidelines to incorporate the accepted contractor-proposed documents as additional contract data items, and also shall state the requirements for outlines, draft reports or partial drafts, revisions of previously delivered and accepted documents, number of copies, official distribution, and similar requirements.)

Table of Software Categories

Category	Decision Factors
C	Special purpose software for limited application
B	General purpose software adaptable to several applications throughout industry
A	General purpose software especially designed for central support of many CAM applications throughout industry

EXAMPLES OF SOFTWARE CATEGORIES

Category C:

COBOL program to translate a special data code (e.g. for part numbers) to a standard code devised for ICAM.

Program or subroutine to compute an engineering property (e.g. breaking limit of a given metal alloy) under special conditions.

Category B:

Programs or subroutines for use with AUTOIDEF in order to operate a new or unusual type of computer terminal in place of the graphics CRT.

GPSS simulation programs for existing sheet metal operations of a particular manufacturer.

Category A:

Library of FORTRAN programs for computing the principal engineering parameters for all standard alloys and metal forming methods used in aerospace manufacturing.

AUTOIDEF

IDSS

Table 1

Standard Documents

Life Cycle Stage	Standard Document Types	Document Selection (Use One Column Only)			Schedule of Deliverable Items			IDEF Diagram
		C	B	A	Augmented Outline	Complete Review Draft	Final Copy	
Needs	Definition Plan							A111
	Feasibility Study							A111
Analysis	Cost/Benefit Analysis							A111
Requirements Definition	Requirements Document							A112
Design	Problem Report							A12
	Software Change Proposal							A12
	Software Change Notice							A12
	Development Plan							A113
	System/Subsystem Specification							A113
	Program Specification							A113
	Data Specification							A113
	Strategic Test Plan							A113

Table 1

Standard Documents

(page 2)

Life Cycle Stage	Standard Document Types	Document Selection (Use One Column Only)			Schedule of Deliverable Items			IDEF Diagram
		C	B	A	Augmented Outline	Complete Review Draft	Final Copy	
Construction and Verification	Problem Report							A12
	Software Change Proposal							A12
	Software Change Notice							A12
	Detailed Test Plan							A114
Integration, Testing, And Validation	Problem Report							A12, A115
	Software Change Proposal							A12
	Software Change Notice							A12
	System Test Analysis Report							A115
	Software Validation Report							A115
	Software Summary							A115
	User's Manual							A115
	Operations Manual							A115
	Program Maintenance Manual							A115
	Installation Guide							A115

Table 1

Standard Documents

(page 3)

Life Cycle Stage	Standard Document Types	Document Selection (Use One Column Only)			Schedule of Deliverable Items			IDEF Diagram
		C	B	A	Augmented Outline	Complete Review Draft	Final Copy	
	Problem Report							A12, A116
Operation	Software Change Proposal							A12
	Software Change Notice							A12, A116
and	Maintenance Plan							A116
Maintenance	Acceptance Test Analysis Report							A116
	Application Report							A116

THIS PAGE DELIBERATELY

LEFT BLANK

ICAM

Software

Documentation

Style Guide

THIS PAGE DELIBERATELY

LEFT BLANK

1. GENERAL INFORMATION

1.1 Summary

Used in conjunction with the content guides, the style guide helps control the quality of ICAM Software Documents. While each content guide specifies the content of a single document, the style manual gives guidelines for the writing of all documents. Besides discussing good writing and programming style, it specifies acceptable format features like typography and graphics. The writer of an ICAM document uses the style guide with the individual content guides to prepare the main body of the document. He then incorporates this main body in the apparatus--title page, table of contents, glossary, etc.--specified by the style guide.

1.2 Environment

The ICAM Software Documentation Style Guide applies to all ICAM software development projects.

1.3 References

Language and Format

Air Force Wright Aeronautical Laboratories, "Preparation of Technical Reports," AFWALP 80-1, June 24, 1977.

Fowler, H. W., _Modern English Usage_, 2nd ed., rev. by Gowers, Sir Ernest, Oxford U P, 1965.

Hodges, John C., and Whitten, Mary E., _The Harbrace College Handbook_, 8th ed., Harcourt Brace Jovanovich, 1977.

Leggert, Glenn H., _et al._, _The Prentice-Hall Handbook for Writers_, 7th ed., Prentice-Hall, 1978.

Strunk, W. S., Jr., and White, E. B., _The Elements of Style_, 2nd ed., Macmillan, 1972.

Watkins, Floyd C., and Dillingham, William B., _Practical English Handbook_, 5th ed., Houghton Mifflin, 1977.

Programming Style

IBM Corp., Data Processing Division, _Improved Programming Technologies--An Overview_, Installation Management Report GC20-1850-0, October 1974, 19p.

Kernighan, B. W. and Plauger, P. J., _The Elements of Programming Style_, 2nd ed., McGraw-Hill Book Company, New York, 1974, 168p.

McCracken, Daniel D., and Weinberg, Gerald M., "How to Write a Readable FORTRAN Program," _Datamation_, v. 18, n. 10, October 1972, 73-77.

Mills, Harlan, "Software Development," _IEEE Transactions on Software Engineering_, December 1976, 265-273.

Myers, Glenford J., _Software Reliability: Principles and Practices_, John Wiley and Sons, New York, 1976, 360p.

National Bureau of Standards, _Guidelines for Documentation of Computer Programs and Automated Data Systems_, Federal Information Processing Standards Publication 38, 15 February 1976, 55p.

Van Tassel, D., _Program Style, Design, Efficiency, Debugging, and Testing_, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1974, 256p.

Yourdon, E., _Techniques of Program Structure and Design_, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1975, 364p.

2. Specification

The ICAM Style Guide consists of the Air Force Wright Aeronautical Laboratories Pamphlet entitled "Preparation of Technical Reports" (AFWALP 80-1), together with the changes and additions that follow. The first part gives further guidelines for standard English usage, the second specifies changes in AFWALP 80-1, and the third adds recommendations for programming style specifically addressed to the ICAM community, which will use FORTRAN 77.

3. Compliance with the Style Guide

Both AFWALP 80-1 and its ICAM additions repeatedly use the imperative mood to instruct writers of technical reports. For example, the first sentence of Chapter 1 states, "Do not overestimate the reader's knowledge of the subject matter of the report." Such statements shall have the force of contractual requirements.

4. ADDENDUM TO AFWALP 80-1

4.1 English Usage

Add the following material to Chapter 1 under the designated sections:

1-1 Choose an appropriate organizing principle for the document--e.g., logical development, chronological sequence, or progression from general concepts to specific instances or examples.

1-2 Every paragraph shall have a single thesis supported both by concrete examples from the software field and by specific and general reasoning. Paragraphs shall be clear, unified, and logically consistent. In addition, the writer shall make clear, logical transitions between paragraphs and develop each paragraph coherently from one statement to the next.

1-3 Add these sections:

g. Use active voice to make clear who or what is performing each action. Such information is important, often crucial, to the reader's understanding.

h. Use parallel structure to organize words, phrases, and clauses into understandable sentence units. For example, in the phrase "not only . . . but also . . ." the words between "only" and "but" should be syntactically equivalent to the words following "also." If the first group of words forms a clause, the second should also be a clause (with subject and verb) and not just a word or phrase.

i. Use works like Fowler's Modern English Usage and a standard dictionary to ensure good usage and proper diction. For further suggestions about style, consult The Elements of Style, by William S. Strunk and E. B. White.

j. Use correct punctuation to help the reader understand the logic of each statement. Make use of the differences between commas, dashes, parentheses, colons, and semicolons. For further reference consult any of the following handbooks: _Practical English Handbook, 5th ed._, ed. Watkins and Dillingham; _The Harbrace College Handbook, 8th ed._, ed. Hodges and Whitten; or _The Prentice-Hall Handbook for Writers, 7th ed._, ed. Glenn Leggert _et al_.

k. When possible, substitute verbal forms (infinitives, participles, gerunds) for nouns--especially long nouns with repetitious suffixes like "tion," "ness," and "ment." Verbal forms read more easily and help the writer avoid long strings of abstract nouns--like "computer program configuration item identification documentation." An infinitive is the "to" form of a verb, as in "to walk"; a gerund is the "-ing" form of a verb, used as a noun, as in "seeing"; a participle is either the present or the past participle of a verb (for regular verbs the "-ing" or "-ed" forms), used as an adjective. In the following sentence "to understand" is an infinitive, "Documenting" is a gerund, and "confused" is a participle: Documenting computer programs enables new or confused programmers to understand the intent of the code.

4.2 Changes to AFWALP 80-1

Make the following changes in the existing text of AFWALP 80-1.

2-5. Table of Contents Because the Table of Contents lists page numbers for sections of the document, the last sentence of this section--"(Page numbers . . . numerals.)"--is ambiguous. Substitute this sentence: "(Number the Table of Contents itself with lower-case Roman numerals, beginning with page 'v.')"

2-6. List of Illustrations Substitute this sentence for the first sentence: "If the document has illustrations, provide a list of illustrations giving figure number, title, and page number for each illustration."

2-7 List of Tables Substitute this sentence for the first sentence: "If the document has tables, provide a list of tables giving table number, caption, and page number for each table."

Chapter 3: Body of Report Delete the text of this entire chapter. Substitute the content guide for the particular document being prepared.

4-1. Report Text Replace all of this section with the following paragraph: "Number and label sections as specified by the individual content guides. Center first-level headings (like "1. GENERAL INFORMATION"); then triple space and indent the first line of text. Second-level headings shall be left-justified. Then double space and indent the first line of text. In all documents third-level and fourth-level subdivisions are optional. When used, third-level headings shall be left-justified, numbered with appropriate three-digit numbers (like "2.2.3"), and underlined. The text shall follow on the same line. When fourth-level headings seem appropriate, number and indent but do not underline them. The text shall follow on the same line, and subsequent lines shall be indented the same as the heading. (For an example of text formats, see the replacement for Attachment 14 that follows the recommendations for programming style. Its title is "Example of Numbered Paragraph Headings and Subheadings.)

4-6. Computer Runs For the first sentence substitute the following sentence: "Submit a printer original of computer runs."

4.3 Programming Style

The following material constitutes Chapter 10 of AFWALP 80-1.

10. Programming Style

The recommendations that follow focus on features of programming style that make programs in higher level languages easier to read. To some extent they imply a methodology that includes such principles as structured programming; however, they do not constitute a methodology. Do not consider these recommendations as a substitute for a complete methodology of programming practices. Instead, view them as an assortment of principles that will enable other programmers and analysts to understand programs better and maintain them more easily. In general, remember that software costs now greatly exceed hardware costs, so that program code should cater to human readers rather than to machines. Programming practices that obscure the purpose and function of the code are usually not worth whatever increase in performance they may achieve. The categories that follow group related requirements for better programming style.

10.1. Commentary

Every program module shall begin with a narrative that explains the purpose and method of the module. Add further narrative comments within the module to explain a particular statement or group of statements. Comments shall not merely repeat formulas from the code, but they must agree with the actual function of the code. Avoid excessive comments that obscure rather than clarify the function of the program or statement. Use complete, concise sentences to clarify the code. It should be easier to understand the function of a block of code by reading the comments than by reading the code itself.

10.2 Redundancy

To avoid unnecessary redundancy, use library functions and write subroutines or function calls instead of repeating code. Do not make the reader of a program interpret the same expression or function more than once.

10.3 Naming Conventions

Use mnemonic names and labels like "YMAX" or "BSQR" to help the reader remember the identity and purpose of variables, statements, functions, and subroutines. Number labeled statements in ascending order so that their relative positions are immediately clear. Declare all data, even if default types are correct. Do not prefix a mnemonic name with a default type letter. For example, use "COUNT" rather than "ICOUNT" for an integer variable, and "JOULES" rather than "RJOULES" for a real variable.

10.4 Structured Code

Limit all program modules to about fifty (50) lines of executable code. Each module shall have a single function, and couplings between modules shall be clear. Insofar as possible, modules shall be independent so that maintenance will be easier. If a mathematical expression requires a continuation line, break the expression into more than one part, using mnemonic variables for intermediate results.

Use single entry, single exit structures whenever possible. For example, the Block IF Statement of FORTRAN 77 simulates a structured "IF . . . THEN . . . ELSE," and the new expressive powers of the FORTRAN 77 DO loop can simulate a structured "While . . . DO." Give each DO statement its own CONTINUE statement. Minimize the use of COMMON blocks.

Avoid FORMAT statements; in FORTRAN 77, READ, WRITE, and PRINT statements can incorporate formatting information and thus make a labeled FORMAT statement unnecessary.

10.5 Miscellaneous

Indent code to show nesting levels.

Write straightforward rather than clever code.

EXAMPLE OF
NUMBERED PARAGRAPH HEADINGS AND SUBHEADINGS

2. REQUIREMENTS

This page is an example of properly numbered and formatted headings and subheadings. Only two levels of heading are required; when third and fourth levels seem appropriate, they should follow the format shown and explained below. For first level headings, the text is indented on the first line but not on subsequent lines.

2.1 Performance

The text for second-level headings is indented on the first line but not on subsequent lines.

<u>2.1.1 Timing</u> The text for third-level headings begins on the same line and is not indented on subsequent lines.

2.1.1.1 Response Time. The text for fourth-level headings begins on the same line and is indented the same as the heading.

Attachment 14

ICAM

Software

Documentation

Content Guides

THIS PAGE DELIBERATELY

LEFT BLANK

1. GENERAL INFORMATION

1.1 Summary

The following introduction to the ICAM Software Documentation Content Guides explains how to use them. The purpose of the guides is to specify the organization and content of the information that contractors shall provide in particular ICAM Software Documents.

1.2 Environment

The ICAM Software Documentation Content Guides apply to all ICAM software development projects, and are established as contractual requirements for Air Force sponsored projects. The contractor shall treat the preparation of documents as a continuing effort that covers preliminary drafts, changes and reviews, and the final documents in their deliverable forms.

1.3 References

Department of Defense, Automated Data Systems Documentation Standards, Standard 7935.1-S, 13 September 1977, 124p.

ICAM Program Office, ICAM Software Documentation Style Guide, 30 April 1979.

National Bureau of Standards, Guidelines for Documentation of Computer Programs and Automated Data Systems, Federal Information Processing Standards Publication 38, 15 February 1976, 55p.

National Bureau of Standards, Guidelines for Documentation of Computer Programs and Automated Data Systems For the Initiation Phase, Federal Information Processing Standards Publication 64, 1 August 1979, 54p.

National Bureau of Standards, Software Summary for Describing Computer Programs and Automated Data Systems, Federal Information Processing Standards Publication 30, 30 June 1974, 4p.

2. DOCUMENT AUDIENCES

Each Content Guide specifies a particular audience that the document shall address. This audience may be a person or a group of persons who will use the document to perform a function--e.g., design, programming, operation, maintenance. In presenting the information specified by the Content Guide, the writer of the document shall use terminology and a level of detail appropriate to this audience.

3. REDUNDANCY

he Content Guides for ICAM Software Documents are redundant in two ways. Each of them includes introductory material to give the reader a frame of reference independent of other documents. Secondly, several documents may provide the same information but differ in context, terminology, and level of detail in order to address different audiences at different points in the software life cycle.

4. FLEXIBILITY

The following paragraphs explain how document writers or contractors can use the flexible features of the ICAM Software Documentation Standards.

4.1 Length of Documents

Each Content Guide applies to documents ranging from a few to several hundred pages. The length depends on the size and complexity of the project and the judgment of the writer as to the level of detail necessary for the environment in which the software will be developed or run.

4.2 Combining Document Types

If the contract statement of work specifies one or more combined volumes, the contractor shall provide all the information required by the Content Guides for the individual documents being combined in a given volume. The contractor shall use the appropriate Content Guides as directions for preparing specific parts of the combined document. For example, he may prepare one volume containing in three individual parts the information specified by the Content Guides for the Feasibility Study, the Cost/Benefit Analysis, and the Requirements Document, respectively. The parts shall appear in the volume in the same order as the pertinent Content Guides appear herein.

4.3 Expanding Document Types

Whenever a particular document has too much information to fit conveniently into a single volume, the contractor can propose to divide it into appropriate units. Depending on the nature of the system, the contractor may prepare a separate document for each module or subsystem , or he may divide the contents between sections. For example, the Requirements Document could become three volumes, one for Functional Requirements, one for Data Requirements, and a third for Performance and Other Requirements. Acceptance of contractor proposals of this type will be indicated by pertinent Additional Requirements in the contract statement of work for software documentation.

4.4 Format

The Content Guides have been prepared using a generally consistent format. Each guide begins with a brief statement of the document's nature and purpose, and then provides an annotated outline that governs the main body of the standard document. The Style Guide specifies how the writer shall incorporate the main body into a complete document that includes other essential features such as title page, table of contents, etc.

4.5 Sequence of Contents

Each Content Guide specifies a broad outline for presenting required information. The writer shall follow exactly the numbered topics and use the designated topic headings. Where the Content Guide does not specify structure--e.g., within second-level headings--the writer shall decide how best to subdivide and organize the required information.

4.6 Flowcharts, Tables, and Forms

Since graphical representations often clarify narrative, the writer shall supplement the information specified in the Content Guide by any graphs, charts, or tables that he deems necessary or desirable.

PROBLEM REPORT

The Problem Report formally documents software problems and alerts appropriate personnel. These problems may range from serious (not routine) bugs to software failures or major problems with the software or the software development project. The developer prepares the report for his own use as well as for the ICAM Program Office. A user may prepare it and send it to the ICAM Program Office or maintenance contractor. In either case the audience may be assumed to be both legally and technically competent to judge the importance of the problem. Problem Reports may be issued at any time in the software life cycle after the definition phase.

ANNOTATED OUTLINE

1. GENERAL INFORMATION

1.1 Summary

(Summarize the nature of the problem, telling where and when it occurs and suggesting possible sources of the problem.)

1.2 Environment

(Identify the developer or user reporting the problem, and tell when and where the problem was first detected. Describe the software system and its operating environment, including software interfaces as well as the hardware on which the software runs.)

1.3 References

(List applicable references, such as:

* Related Problem Reports;
* Affected project documentation;
* Related Software Change Documents;
* Documents from related projects;
* Relevant standards or guidelines, including FIPS publications, DIDs, and DoD manuals.)

2. DETAILS

2.1 Background

(Trace the history of the problem from its initial detection to the present. Give the dates and contents of Problem Reports or Software Change Documents that address closely related problems.)

2.2 Circumstances

(Describe in detail the circumstances under which the problem occurred. If the affected software is not yet operational, explain what factors are in conflict--e.g., contradictory requirements or specifications. If the software is being tested or operated, tell which inputs produce the problem. In either case, state explicitly what the problem is and, if possible, why it occurs.)

2.3 Recommendations

(Judge the seriousness of the problem and recommend possible actions to deal with it.)

THIS PAGE DELIBERATELY

LEFT BLANK

SOFTWARE CHANGE PROPOSAL

The Software Change Proposal recommends changes to the requirements, specifications, documentation, or actual operating characteristics of the software system. In any of these cases the Software Change Proposal shall completely specify both the old and the new versions of the software, including necessary changes to software documentation. In addressing it to the contractor's own project supervisor as well as to the project manager from the ICAM Program Office, the writer may assume that his audience is both legally and technically competent to judge the importance of the software change. Either the contractor or the ICAM Program Office can produce the Software Change Proposal during any phase of the software life cycle, but both parties must agree to the change before the document becomes legally binding.

ANNOTATED OUTLINE

1. GENERAL INFORMATION

1.1 Summary

(State the purpose of the Software Change Proposal--i.e., whether the changes affect the requirements, specifications, documentation, or operation and maintenance of ICAM software.)

1.2 Environment

(Name the developer, the potential manufacturing functions, the hardware and software interfaces, and the possible operating sites of the software system.)

1.3 References

(List applicable references, such as:

* Related Problem Reports;
* Affected project documentation;
* Related Software Change Proposals or Notices;
* Documents from related projects; and
* Relevant standards or guidelines, including FIPS publications, DIDs, and DoD manuals.)

2. DESCRIPTION

2.1 Function

(Describe the purpose of the software that is affected by the proposed change.)

2.2 Old Version

(Indicate the software version which will be replaced if the proposed change becomes effective.)

2.3 New Version

(Indicate the software version which would go into effect with the proposed change.)

3. JUSTIFICATION

3.1 Background

(Trace the history of the situation which this particular Software Change Proposal addresses. Recount the events which make the change advisable, giving dates and contents of previous Problem Reports or Software Change Proposals or Notices that addressed the same or a closely related issue.)

3.2 Alternatives

(Describe alternative changes in the software which could also solve the stated problem or achieve the desired improvement.)

3.3 Costs and Benefits

(Assess the costs and benefits (to both the contractor and the ICAM Program Office) of the recommended change. Justify these costs and benefits relative to those of keeping the software as it is or making one of the alternate changes. State explicitly the reasons for preferring this change to other possible actions.)

3.4 Impact

(Describe the impact of this change on the software project. Name other documents which the change will affect, and tell how to update or revise these documents. Estimate the effect of the change on the project schedule. If the change may make it difficult for the developer to meet contractual obligations, spell out anticipated problems and suggested solutions.)

SOFTWARE CHANGE NOTICE

The Software Change Notice authorizes proposed changes to the requirements, specifications, documentation, or actual operating characteristics of the software system. In any of these cases the Software Change Notice shall completely specify both the old and the new versions of the software, including necessary changes to software documentation. In addressing it to the contractor's own project supervisor as well as to the project manager from the ICAM Program Office, the writer may assume that his audience is both legally and technically competent to judge the importance of the software change. Acting together, the contractor and the ICAM Program Office can produce the Software Change Notice during any phase of the software life cycle.

ANNOTATED OUTLINE

1. GENERAL INFORMATION

1.1 Summary

(State the purpose of the Software Change Notice--i.e., whether it authorizes changes to the requirements, specifications, documentation, or operation and maintenance of ICAM software.)

1.2 Environment

(Name the developer, the potential manufacturing functions, the hardware and software interfaces, and the possible operating sites of the software system.)

1.3 References

(List applicable references, such as:

* Related Problem Reports;
* Affected project documentation;
* Related Software Change Proposals or Notices;
* Documents from related projects; and
* Relevant standards or guidelines, including FIPS publications, DIDs, and DoD manuals.)

2. DESCRIPTION

2.1 Function

(Describe the purpose of the software that the authorized change will affect.)

2.2 Old Version

(Indicate the software version which will be replaced when the change becomes effective. Summarize the former characteristics, functions, etc. that are changed, replaced, or deleted.)

2.3 New Version

(Indicate the software version which will go into effect with the change. Summarize the new characteristics, functions, etc., and reference or attach complete documentation changes effecting the change.)

3. EFFECTIVE DATE

(Give the date the Software Change Notice goes into effect.)

THIS PAGE DELIBERATELY

LEFT BLANK

DEFINITION PLAN

The Definition Plan is the first technical document in the development of a new software system. The Plan specifies the objectives, tasks, method, and schedule for analyzing application needs and defining requirements of the proposed software. This document is prepared for the ICAM Program Office and other participants in the analysis and definition effort.

ANNOTATED OUTLINE

1. GENERAL INFORMATION

1.1 Summary

(Summarize the purposes envisioned for the software.)

1.2 Environment

(Identify the analysis and definition project participants. For the software, describe potential manufacturing users, hardware and software environments, and possible operating sites.)

1.3 References

(List applicable references:

* Problem background;
* User requirements;
* Other relevant information in published documents.)

2. BACKGROUND

2.1 Problem

(State the nature of the problem or the manufacturing application that requires a software solution.)

2.2 Scope

(Define the scope of the project -- that is, establish the boundaries for what the software should and should not do.)

2.3 Objectives

(Specify the purpose, goals, and key features of the system to be defined. Describe envisioned interfaces to extant systems. Identify future plans. Present the objectives in terms of relevant manufacturing applications.)

3. PLAN

3.1 Tasks

(Describe the tasks that must be performed to initiate the development process. Specifically include feasibility analysis, cost/benefit analysis, and definition of functional, data, and performance requirements. For each task, specify goals, assumptions, constraints, and expected scope of results.)

3.2 Schedule

(Develop a schedule and associated milestones for the tasks defined. Include appropriate charts and other graphic displays, such as PERT charts.)

3.3 Approach

(Describe the strategy to be used in the analysis process, and identify software tools and methodology that must be used during the Analysis and Definition stages, such as IDEF versions 0, 1, and 2.)

3.4 Resources Required

(Estimate manpower and resources required to perform the individual analysis tasks. Identify project participants' contributions by task.)

THIS PAGE DELIBERATELY

LEFT BLANK

FEASIBILITY STUDY

This document presents concepts, broad requirements, and objectives of the proposed software system. The Feasibility Study evaluates pertinent software solutions and technology relative to application goals and needs; it compares these alternative software approaches and recommends one software design approach as best for the envisioned application. Intended for managers, this document should provide them with adequate information to make a decision with respect to the further development of the proposed system. This study should be prepared during the initial stage of software development.

ANNOTATED OUTLINE

1. GENERAL INFORMATION

1.1 Summary

(Identify systems analyzed and briefly state major conclusions.)

1.2 Environment

(Identify the analysis team, potential manufacturing users, hardware and software environment, including interfaces, potential interaction with other systems or organizations, and possible operating sites for the software system.)

1.3 References

(List applicable references, such as the Definition Plan.)

2. MANAGEMENT SUMMARY

(Summarize pertinent facts about the recommended system. State the recommendations, justification, envisioned schedule of development, and end products. Include relevant information from the Cost/Benefit Analysis. Briefly describe requirements -- e.g., new services and increased computing capacity. List the goals and objectives of the recommended system -- e.g., to automate control functions in sheet metal fabrication. Identify assumptions and constraints affecting the recommended system, such as operational life of the proposed system, available resources, and changing hardware or software environment. Summarize the methods used in performing the feasibility analysis -- e.g., survey, modelling, and simulation. Identify evaluation criteria used in arriving at recommendations.)

3. SYSTEM REQUIREMENTS AND OBJECTIVES

3.1 Requirements

(Describe the proposed system's broad requirements, and identify mandatory items. Include input/output, files descriptions, validation criteria, major processing or data flow, security, privacy, and control, and any required interfaces with other systems.)

3.2 Objectives

(Identify and explain in detail the major objectives of the proposed system -- for example, to reduce manpower and equipment cost, increase processing speed, improve efficiency of manufacturing process, and increase productivity.)

4. ANALYSIS OF EXISTING SYSTEM

(Information on the existing system provides a basis for comparison and for determining economic and management advantages of the proposed system. The Functional, Information, and User models prescribed by the IDEF methodology can be attached to this portion of the study. If the IDEF methodology is not used, details in this section provide guidance on information elements to be included in the analysis of an existing system.)

4.1 Processing/Data Flow

(Describe in detail the processing, data flow, and functions performed by the current system. This description may use graphical representations.)

4.2 Workload

(Specify the volume of work handled by the existing system.)

4.3 Costs

(Itemize, in terms of such factors as manpower and equipment, the costs incurred in operating the existing system.)

4.4 Personnel

(Identify skill categories and personnel required to operate and maintain the current system.)

4.5 Equipment

(Describe the equipment used by the existing software system.)

4.6 Limitations

(Identify and describe the limitations of the existing system, such as inadequate response time, or inability to meet new requirements.)

4.7 Special Considerations

(Identify other factors unique to this system.)

5. PROPOSED SYSTEM

5.1 Description of Proposed System

(Present the overall system concept, and describe how it satisfies the requirements defined in section 3.)

5.2 Improvements

(Describe in detail how the proposed system will meet the objectives in section 3.2)

5.3 Impact

(Describe the anticipated effect of the proposed system on the current operating environment -- equipment impact, software impact, organizational impact, and operational impact. Also include a discussion of potential conversion problems.)

6. ALTERNATIVE SYSTEMS

(Using the outline of section 5, describe each alternative software approach considered. In each case, state the reason for not choosing this approach. If no alternatives were considered, so state.)

6.1 Alternative System 1

6.2 Alternative system n

7. RATIONALE FOR RECOMMENDATION

(Justify selecting the proposed system over the alternative software solutions. Include required resources, possible effects of delay, consequences of not taking action, and all quantifiable and non-quantifiable benefits.)

8. PROPOSED SCHEDULE

(Outline a proposed schedule to include design, construction, testing, conversion, and implementation. Identify major milestones and management decision points.)

THIS PAGE DELIBERATELY

LEFT BLANK

COST/BENEFIT ANALYSIS

This document provides managers, users, designers, and auditors with estimates of costs and benefits for the alternative approaches considered. It considers recurring and non-recurring costs, as well as tangible and intangible benefits. This document is prepared during the initial phase of software development, concurrently with the Feasibility Study.

ANNOTATED OUTLINE

1. GENERAL INFORMATION

1.1 Summary

(Identify systems studied and briefly state major conclusions.)

1.2 Environment

(Identify the analysis team, the potential manufacturing functions, the hardware and software interfaces, and the possible operating sites of the software system.)

1.3 References

(List applicable references, such as:

* Definition Plan;
* Feasibility Study;
* Other relevant project documentation.)

2. MANAGEMENT SUMMARY

(Present a concise overview of the cost/benefit analysis conducted. Summarize recommendations for development and operation of the system. State the purpose and the scope of the cost/benefit analysis, the alternatives for development and operation, and major cost elements. Briefly describe the operational requirements, system life, and workload for which the cost/benefit analysis was conducted. Identify the assumptions and constraints affecting the cost/benefit analysis. Summarize the procedures for conducting the cost/benefit analysis, and the techniques for estimating and computing costs. These techniques may be detailed in an appendix. State criteria for evaluating alternative systems -- for example, organizational objectives, efficient operation, etc.)

3. DESCRIPTION OF ALTERNATIVES

(This section should be organized to provide a meaningful and logical presentation of alternatives.)

3.1 Current System

(Describe the technical and operational characteristics of the current system.)

3.2 Proposed System

(Describe the technical and operational characteristics of the proposed system.)

3.3 Alternative System 1

(Describe the technical and operational characteristics of alternative system 1.)

3.4 Alternative System n

(Describe the technical and operational characteristics of alternative systm n.)

4. COSTS

(Describe the costs of developing and operating each alternative system. If there is an existing system, include costs associated with its continuation. Where applicable, compare in-house costs with contractor costs for developing, operating, or maintaining a system.)

4.1 Non-Recurring Costs

(Present non-recurring costs of each alternative over the system's life. Include such categories of costs as:

* Capital investment cost, including the cost of acquiring, developing, and installing ADP equipment, and the cost of setting up a data processing facility.

* Other non-recurring costs, such as requirements and design studies, database preparation, ADP software conversion, and procurement planning and benchmarking.)

4.2 Recurring Costs

(Present periodic costs for operating and maintaining each alternative over the system's life -- for example, equipment, data communication, software lease and rental, personnel salaries and fringe benefits, direct support services, and travel and training.)

5. BENEFITS

(Describe non-recurring and recurring benefits which could be attained through the development of each proposed alternative. State benefits in quantifiable or non-quantifiable terms that relate to organizational objectives, goals, missions, functions, and operating environment.)

5.1 Non-Recurring Benefits

(Describe quantifiable benefits such as cost reduction, value enhancement to application systems, and others.)

5.2 Recurring Benefits

(Present periodic benefits resulting from operating and maintaining the alternative over the system's life. Include savings realized from equipment lease, data communication lease, personnel salaries and fringe benefits, and cost avoidance resulting from choosing the "best" alternative.)

5.3 Non-Quantifiable Benefits

(Describe benefits that cannot be quantified, such as improved service and reduced risk of incorrect processing. For those intangible benefits that can be assigned values in terms of estimates and tradeoffs, include boundary estimates and tradeoffs with tangible benefits.)

6. COMPARATIVE COST/BENEFIT SUMMARY

(Present the information below in a manner that facilitates comparison. Provide supporting documentation, as required, for validation and management review.)

6.1 Cost of Each Alternative Over the System Life

(For each alternative, present costs in the period in which they will be incurred. Include non-recurring costs, recurring costs, total costs, system life costs, present value costs, residual costs, and adjusted costs.)

6.2 Benefits

(Identify the period of benefits. Enter the quantifiable benefits for the period in which they accrue, and make present-value calculations.)

6.3 Net Present Value

(Calculate the net present value by subtracting the adjusted cost from the total present value of benefits.)

6.4 Benefit/Cost Ratio

(Calculate the benefit/cost ratio by dividing the total present value by the adjusted cost.)

6.5 Payback Period

(Determine when the sum of benefits first exceeds the total cost.)

7. SENSITIVITY ANALYSIS

(Sensitivity Analysis assesses the sensitivity of costs and benefits to changes in key factors. Sensitivity analyses conducted on different configurations with each alternative proposal provide a range of costs and benefits which is likely to be better than a single estimate.)

7.1 Methodology

(Describe the approach, assumptions, and model for the sensitivity analysis. Describe the analysis of sensitivity factors. For example, consider the effects of varying such factors as length of system life, volume, mix, or pattern of workload, requirements, and configuration of equipment or software; also consider the effects of alternative assumptions on such factors as inflation rate, residual value of equipment, etc.)

7.2 Sources of Data

(Identify the sources of data for the sensitivity analysis; also identify the method used for data collection and the quality of data.)

7.3 Other Factors

(Identify other factors which may affect the assessment of cost benefits, either quantitatively or qualitatively, for one or more alternatives, but are not amenable to sensitivity analysis or its implications.)

7.4 Results

(Identify and display in convenient fashion the results of the sensitivity analysis for all alternatives and factors.)

7.5 Evaluation and Conclusion

(Present the key points of the sensitivity analysis, evaluate its validity and implications, and present the conclusion.)

REQUIREMENTS DOCUMENT

The Requirements Document defines the function, data, performance and other detailed requirements of the proposed software system. This document is used to assure that software designers and prospective users are thoroughly and equally well informed of the intended capabilities and operation of the envisioned software. It is prepared during the System Definition stage, and substantially extends and refines the broad outline for the recommended software presented in the Feasibility Study.

ANNOTATED OUTLINE

1. GENERAL INFORMATION

1.1 Summary

(Summarize the general nature of the software to be developed.)

1.2 Environment

(Identify the system developer, the analysis team, the potential manufacturing users, the hardware and software environment, and the possible operating sites of the software system.)

1.3 References

(List applicable references, such as:

* Definition Plan;
* Feasibility Study;
* Cost/Benefit Analysis;
* Other related project documentation.)

2. OVERVIEW

2.1 Background

(Briefly describe the problem the software will solve, state the scope of the software, and provide any information that is relevant to the derivation of requirements for this software system.)

2.2 Objectives

(Specify the purpose, goal, and key features of the system to be developed. Describe the objectives in terms of manufacturing applications.)

2.3 Existing Methods and Procedures

(Describe how the present methods, procedures, or system fulfill current needs. Include information such as functions performed by the present system, methods and procedures used, volume and frequency of data, and deficiencies and limitations of the current system.)

2.4 Proposed Methods and Procedures

(Describe the proposed software and its capabilities. Identify techniques and procedures to be used in the new system. Identify the requirements that the proposed software will satisfy. If special software tools or methodologies are used in generating the requirements, describe them, and attach the results.)

2.5 Summary of Improvements

(Itemize improvements to be obtained from proposed software, such as acquiring new capabilities, upgrading existing capabilities, eliminating deficiencies, etc.)

2.6 Summary of Impacts

(Summarize anticipated impacts of the proposed software on the present system. Include a description of required changes to the current hardware configuration or to existing software; also describe the potential impact on the organizational structure, operational procedures, and developmental activities.)

2.7 Cost Considerations

(Describe resource and cost factors that may influence the development, design, and continued operation of the proposed software.)

3. FUNCTIONAL REQUIREMENTS

3.1 Functions

(Describe in detail the functions that are required of the proposed system, and indicate how these functions will satisfy the stated objectives.)

3.2 Processing

(Describe processing requirements, such as hardware, facilities, telecommunications, etc.)

3.3 Support

(Identify support requirements, such as software tools, data communication software, test software, etc.)

3.4 Interface

(Identify required interfaces with other systems, with components within the same system. Also describe user interfaces, where appropriate.)

4. DATA REQUIREMENTS

4.1 Data Description

(Describe the data required for the software system. If data already exists, identify the database (or the file) names, and describe the required changes, if any. Describe the characteristics of the data, such as acceptable representation, formats, relationships, logical/physical groupings, value ranges, critical values, and others.)

4.2 Input/Output

(Describe the required inputs and outputs for the proposed software. Include information about source, location, and input/output media. Tie the I/O back to particular functions or groups of functions.)

4.3 Frequency and Volume

(Indicate the expected volume of the required data, and frequency of collection and access.)

4.4 Activity

(Describe the projected data activity; include information about data access, manipulation, and update.)

4.5 Constraints

(State the constraints on the data requirements, such as limits for further expansion or utilization, maximum size, control requirements, security, and integrity of the data.)

4.6 Collection

(Describe methods of collecting data, sources of the data, storage media, data validation criteria, and data collection responsibilties. Provide specific instructions for data collection procedures.)

5. PERFORMANCE AND OTHER REQUIREMENTS

5.1 Performance

(State performance requirements for the system, including system availability, responsiveness, data accessability, and response time. If applicable, describe the use of performance monitors, benchmarking, or modelling tools for deriving performance requirements, and attach results. Tie performance criteria back to particular functions or groups of functions.)

5.2 Security

(State the security requirements for the system. Describe special procedures to be used in ensuring security.)

5.3 Integrity and Reliability

(State the system integrity and reliability requirements.)

5.4 Flexibility

(Describe the requirements for adaptability of the proposed system to changes in mode of operation, in interfacing with other software, and in adapting to integration into different operating environments.)

5.5 Failure Contingencies

(Specify possible failures of the software or the hardware, and the consequences (in terms of performance), and describe alternative courses of action that can be taken to satisfy the informational requirements. Include such techniques as back-up, fallback, and recovery and restart.)

6. OPERATING ENVIRONMENT

6.1 Equipment

(Identify the equipment required to operate the software, including new equipment, and relate it to specific functions and requirements. Include information about the processor, the size of the internal storage, online and off-line auxiliary storage media, forms, and devices, and other information.)

6.2 Controls

(Describe the operational controls imposed on the software. Identify the source of these controls.)

THIS PAGE DELIBERATELY

LEFT BLANK

DEVELOPMENT PLAN

The Development Plan describes the organization, resources, tasks, schedule, methods, and standards for development of software, beginning after the requirements definition is completed and extending through the delivery and acceptance testing of the software system. The Plan is prepared for the ICAM Program Office and all technical participants in the development work, and is used to assure that technical activities have been adequately planned and that all participants understand the working environment and objectives. Potential readers may be assumed familiar with general software design, programming, and testing activities. The Plan should be prepared soon after the Requirements Document, and should be approved by appropriate officials before significant development begins.

ANNOTATED OUTLINE

1. GENERAL INFORMATION

1.1 Summary

(Identify the software system involved and its general characteristics. Summarize the scope and purpose of this development plan, pointing out significant methods, criteria, specifications, and techniques being used to achieve timely completion of a quality software product.)

1.2 Environment

(Identify the development organization and its institutional members. Describe the envisioned utilization of the software and prospective user installations. Also describe the supporting hardware and software components, and the intended degree of portability.)

1.3 References

(List previously or concurrently issued documents that are applicable to the software and this development plan.)

2. DEVELOPMENT TASKS

(Describe the planned development work as distinct tasks and define the deliverable results of each. State the technical disciplines involved, and briefly describe those which are unique or innovative. List the individual products and milestones, and the anticipated start and completion times for each task or significant activity. Identify the responsibility of each participating organization in the completion of each milestone. State the expected allocation of personnel time and other resources to individual tasks. Show the sequence, timing, and interdependence of all tasks through appropriate charts or figures.)

3. DEVELOPMENT ORGANIZATION

(Describe fully the organizational units, principal personnel, roles and responsibilities for the development of the proposed software. Indicate the significant resources, particularly computer equipment and software, to be provided by each party. Define the work procedures and relationships planned among the parties in the development. Relate the information to the defined tasks and to the stages in the software life cycle.)

4. DEVELOPMENT AND QUALITY ASSURANCE METHODOLOGY

(Referring to the tasks defined above, describe fully the technical criteria, specific techniques, tools and support software, and development practices to be used. Define the quality assurance principles and practices. Define significant day-to-day working methods. List and describe major reviews, audits, inspections, and demonstrations. Define all quantitative and objective measurement or assessment techniques to be used.)

5. STANDARDS

(Describe the application of specific standards cited in the References and others to be developed for this project. Include standards on programming languages, development tools, program structuring and coding conventions, testing, and documentation.)

THIS PAGE DELIBERATELY

LEFT BLANK

SYSTEM/SUBSYSTEM SPECIFICATION

The System/Subsystem Specification describes in detail the functional specifications, operating environment, and design characteristics for the system or subsystem. Derived from the Requirements document, the System/Subsystem Specification specifies a design strategy for the overall system. It presents the logic flow of the entire system/ subsystem, thus showing an integrated view of the system or subsystem dynamics. This technical document is used by the software designer, the analysis team, the programmer, and the configuration manager. It is one of the first documents to be prepared during the Design stage in the software development life cycle.

ANNOTATED OUTLINE

1. GENERAL INFORMATION

1.1 Summary

(Summarize the system's specifications and other design elements for the system.)

1.2 Environment

(Identify the system developer, potential manufacturing uses, hardware and software environment, and possible operating sites for the system.)

1.3 References

(List applicable references, such as:

* Development Plan;
* Feasibility Study;
* Cost/Benefit Analysis;
* Requirements Document;
* Other relevant project documentation.)

2. SPECIFICATIONS

2.1 Description

(Describe in general terms the system or subsystem being designed. Describe how this system or subsystem relates to other systems or subsystems. Discuss potential applicability and adaptability of this system to other manufacturing activities.)

2.2 Functions

(Specify the system/subsystem functions, and show how they satisfy specific functional requirements.)

2.3 Performance

(Specify performance requirements in qualitative and quantitative terms. Describe the flexibility and adaptability of the system/subsystem to changes in performance requirements.)

3. OPERATING ENVIRONMENT

3.1 Equipment

(Identify and describe the equipment required for the operation of the system/subsystem. Include present equipment configuration, as well as new equipment required to support specific functional requirements.)

3.2 Support Software

(Describe any support software required for the system/subsystem being developed -- for example, a utility dump routine, a sort/merge package, or system testing routines. If changes are required in the support software, describe the nature, status, and availability date of such changes.)

3.3 Interfaces

(Describe relevant interfaces with other software.)

3.4 Security and Privacy

(Describe any security and privacy requirements for the system/subsystem.)

3.5 Controls

(Describe the operational controls imposed on the system/subsystem. Identify sources of these controls.)

4. DESIGN CHARACTERISTICS

4.1 Operations

(Describe operating characteristics that are specific to the computer system facility where the new system/subsystem will be operational. For example, if the computer system facility is primarily oriented toward batch processing, the design of new applications should take that operating factor into consideration.)

4.2 System/Subsystem Logic

(Describe the logic flow of the entire system/subsystem in the form of a flowchart (or other graphical representation). The flow should provide an integrated presentation of the system/subsystem dynamics, of entrances and exits, computer programs, support software, control, and data flow.)

5. PROGRAM SPECIFICATIONS

5.1 Program Specification (Identify by Name)

(Specify this program component of the System/Subsystem. Describe the system/subsystem function the program will satisfy, and describe the design characteristics of the program.)

5.2 Program Specification (Identify by Name)

(Describe the remaining computer programs in a manner similar to the paragraph above.)

THIS PAGE DELIBERATELY

LEFT BLANK

PROGRAM SPECIFICATION

The Program Specification describes program designs in sufficient detail to enable program coding. In contrast with the System/Subsystem Specification, which describes the system's programs on a general level, the Program Specification addresses each individual program and its modules. It specifies the functions, performance requirements, interfaces, operating environment, and design characteristics of each computer program to be constructed. This technical document is used by the software designer, the system analyst, and the programmer for constructing the software. The Program Specification is prepared during the Design stage of the software development life cycle, following review and conditional approval of the System/Subsystem Specification.

ANNOTATED OUTLINE

1. GENERAL INFORMATION

1.1 Summary

(Summarize the specifications of the computer program, and briefly describe the function that this program performs.)

1.2 Environment

(Identify the system designers, potential manufacturing users, hardware and software environment, and possible operating sites.)

1.3 References

(List applicable references, such as:

* Development Plan;
* Feasibility Study;
* Cost/Benefit Analysis;
* System/Subsystem Specification;
* Other relevant project documents.)

2. SPECIFICATONS

2.1 Program Description

(Provide a general description of the program, including the system/subsystem function that this program will satisfy, the algorithm and the programming language to be used, required utility program and interfaces, and relationships to other system/subsystem components. Identify any program features that would restrict modification, or any feature that may create maintenance problems.)

2.2 Functions

(Specify the functions of the program to be developed. If the program in itself does not fully satisfy a system/subsystem function, show its relationship to other programs which in aggregate satisfy that function.)

2.3 Performance

(Specify performance criteria in quantitative terms. Describe the tests that the program must be subjected to before it is deemed acceptable. Describe the flexibility and adaptability of the program to changes in performance requirements.)

3. OPERATING ENVIRONMENT

3.1 Equipment

(Identify and describe the equipment required for the operation of the program. Describe the present equipment configuration, and indicate the changes needed to support specific functional requirements, as reflected in the System/Subsystem Specification. Identify other hardware/software on which the program can run.)

3.2 Support Software

(Describe any support software required for the program being developed -- for example, utility routines and system software. Indicate if a Database Management System (DBMS) is being used, and whether the program being developed is an application program under the DBMS. If the support software requires changes, describe the nature, status, and availability date of such changes.)

3.3 Interfaces

(Describe relevant interfaces with other software components.)

3.4 Storage

(Specify the storage requirements for the program, including any constraints and conditions. Include information about internal, offline, and auxiliary storage requirements.)

3.5 Security and Privacy

(Describe any security and privacy requirements for the program.)

3.6 Controls

(Describe program controls, and identify sources of these controls.)

4. DESIGN CHARACTERISTICS

4.1 Operating Procedures

(Describe the operating procedures for the program. These procedures are specific to the computer system facility where the program will be implemented. Identify the software tools to be used in the implementation of the program, and any other special requirements that should be noted.)

4.2 Inputs

(Describe the inputs to the program. Specify the source and type of data, its expected volume and frequency, its means of entry, and provisions for maintaining necessary privacy and security.)

4.3 Program Logic

(Describe the logic flow of the program, showing the interactions among program modules, input/output relationships, and general data flow.)

4.4 Outputs

(Describe each output from the program, including desired format, expected volume and frequency, disposition of products, and security and privacy considerations.)

4.5 Data Base

(Describe the logical and physical characteristics of any database used by the program. If the database is managed by a Data Base Management System (DBMS), then describe the database in terms of the DBMS used.)

THIS PAGE DELIBERATELY

LEFT BLANK

DATA SPECIFICATION

The Data Specification is a technical document that specifies the characteristics of the data used by the programs being designed. It includes a description of the physical and logical characteristics of the data, and pertinent information about input, access, updating, and processing. Intended for use by technical personnel, such as software designers, system analysts, and programmers, the Data Specification is prepared during the Design stage of the software development life cycle. The Data Specification is particularly important for defining large aggregates of data such as parameter tables or databases.

ANNOTATED OUTLINE

1. GENERAL INFORMATION

1.1 Summary

(Summarize the scope and characteristics of the data, and briefly describe the general functions of the application software using the data.)

1.2 Environment

(Identify the system developer, the hardware and software environment, the pertinent manufacturing functions, and the possible operating sites of the software system.)

1.3 References

(List applicable references, such as:

* Requirements Document;
* System/Subsystem Specification;
* Program Specification;
* Other relevant project documents.)

2. DESCRIPTION

2.1 Identification

(Identify aggregate data structures or databases and describe the nature of the data, such as whether it is temporary, experimental, or test data.)

2.2 Using Software

(Identify all programs that will use this data. Provide the name, the release, and the version number of the software.)

2.3 Conventions

(Describe the labelling and tagging conventions that a programmer or an analyst must know to use the data specifications.)

2.4 Special Instructions

(Provide any special instructions for personnel who will be generating, accessing, or modifying the data. As necessary for brevity, provide references to the appropriate software.)

2.5 Support Software

(Give the name, function, major operating characteristics, and instructions for all support software directly related to the data, including DBMS software. Cite support software documentation. Examples of support software are:

* Database management system;
* Data dictionary/directory system;
* Storage allocation software;
* Data loading software;
* File processing software.)

3. LOGICAL CHARACTERISTICS

(Describe each data item or field as well as the associations that constitute the logical organization of the data as viewed by prospective manufacturing users. Define and explain the structure of the data, the relationships among data items, constraints on and criteria for access and updating, and the characteristics pertaining to integrity and validity.)

4. PHYSICAL CHARACTERISTICS

4.1 Storage

(Specify the computer storage requirements, constraints, and machine-dependent factors involving the data. Include information about internal storage areas, devices required for peripheral storage, physical storage management, and offline storage requirements.)

4.2 Access

(Describe indexes, access methods, and access paths as appropriate for the intended physical storage approach.)

4.3 Design Considerations

(State design considerations for handling the data, in order to achieve such goals as efficient access and reliability. Include information about potential implementation strategies to satisfy design considerations.)

STRATEGIC TEST PLAN

The Strategic Test Plan describes concepts, criteria or standards, tools, and strategies for testing the software system. The developer produces it at the end of the Definition stage of the software life cycle and addresses it to internal management (especially programming, design, and testing supervisors) and to ICAM Program Office managers.

ANNOTATED OUTLINE

1. GENERAL INFORMATION

1.1 Summary

(Summarize the software characteristics and the developer's general testing strategy.)

1.2 Environment

(Name the testing organization or group and describe the testing environment, including kinds of personnel as well as support hardware and software. In particular, compare the testing environment to the planned operating environment.)

1.3 References

(List applicable references, including

* the Requirements Document against which the software will be tested; and
* publications that give more detail about proposed testing concepts, strategies, or methods.)

2. TESTING CONCEPTS AND CRITERIA

(Discuss the overall testing approach, concepts, and criteria that will assure production of quality software that meets its objectives and requirements. State how much of the testing will be simulation, and how much actual operation of the software. Define the level of functional testing--i.e., the features of the software the developer will test. Also define the level of structural testing--i.e., how completely the developer will examine the source code. Demonstrate that the planned testing will assure the acceptability of the software with respect to its requirements and specifications. Establish the criteria the developer will use to judge the quality, performance, and function of the software. State the quantitative criteria for determining acceptability in such major areas as timing, throughput, accuracy, and storage.)

3. TOOLS

(Describe any tools that the tester plans to use in checking out the software system. If such tools are not currently available in the testing environment, discuss plans for getting them.)

4. STRATEGIES

4.1 Kinds of Tests

(Name and describe the distinct types of testing planned for the software. Discuss the purpose and specific methodology for each of these tests, and tell how the developer will use specific tools for each of them. Discuss the applicability of different tests to general requirements of unit testing, integration testing, system testing, and acceptance testing.)

4.2 Test Data Generation

(Describe fully the methods and procedures for producing test data for each major area of testing.)

4.3 Documentation

(Discuss general plans for documenting test procedures and results, and for disseminating test sets for reuse.)

THIS PAGE DELIBERATELY

LEFT BLANK

DETAILED TEST PLAN

The Detailed Test Plan specifies detailed test data and procedures, as well as criteria for reducing and evaluating test data. Its audience includes personnel from the ICAM Program Office and managers and technical staff from the contracting company. The developer produces this plan during the Design and Programming stages of the software life cycle.

ANNOTATED OUTLINE

1. GENERAL INFORMATION

1.1 Summary

(Summarize the implementation of the general testing strategy that has been accepted for determining whether the software meets its requirements and specifications.)

1.2 Environment

(Identify the testing organization, facilities, and sites, as well as support hardware and software. Describe any prior testing not governed by this plan and state the results that have or should be obtained.)

1.3 References

(List applicable references, such as:

* Previously published documents on the project--especially the Strategic Test Plan, the Requirements Document, and the three Specifications;
* Documentation concerning related projects; and
* Standards and guidelines such as FIPS publications, DIDs, and DoD manuals.)

2. PLAN

2.1 Software Description

(Briefly describe the modular components and functions of the software system which have influenced the selection of locations, resources, and schedule described below.)

2.2 Milestones

(List the locations and dates for each major testing activity.)

2.3 Testing Activities

(Describe the activity at each test site, including the test function (e.g., system test), the test location, the participating organizations, a detailed schedule of testing events, a comprehensive list of equipment, support software, and personnel necessary to perform the test, required test materials (including the software, its documentation, test inputs and sample outputs, and test control software and worksheets), and plans for providing any training that is necessary before personnel can perform the tests.)

3. SPECIFICATIONS AND EVALUATION

3.1 Specifications

(List the scheduled tests, and state the functional requirements and specifications that each addresses. Also describe the progression from test to test.)

3.2 Methods and Constraints

(Describe and justify any modifications of the methodology and strategy specified in the Strategic Test Plan. Indicate anticipated limitations on the test due to test conditions, such as interfaces, equipment, personnel, and data bases.)

3.3 Evaluation

(Describe the rules for evaluating test results, considering such factors as the range of data values, the combinations of input types, and the maximum number of allowable halts or interrupts. Specify manual and automated techniques for manipulating the test data into a form suitable for comparison with required results.)

3.4 Documentation

(Fully describe how the developer will document the testing of the software system.)

4. TEST DESCRIPTIONS

(Fully describe each test, including the control over insertion of inputs, sequencing of operations, and recording of results, the input data and commands, the intermediate messages, the expected output data, and the step-by-step procedures for performing the test, including test setup, initialization, and termination.)

TEST ANALYSIS REPORT

The Test Analysis Report describes the tests conducted and gives the results, describes the method used in analyzing test results, presents the demonstrated software capabilities and deficiencies for review, and evaluates the status of the software in light of the test results. This report is prepared for technical and management personnel, both in the developing organization and in the ICAM Program Office. The Test Analysis Report is prepared for system testing and for acceptance testing, after each test has been completed. System Testing is designed to compare the software system to its original goals and objectives, while acceptance testing is the process of comparing the finished software to its initial requirements, as stated in the Requirements Document.

ANNOTATED OUTLINE

1. GENERAL INFORMATION

1.1 Summary

(Summarize the application and general functions of the software being tested and the test analysis documented here.)

1.2 Environment

(Identify the system developer, the potential manufacturing users, the supporting hardware and software, and the possible operating sites of the software system. Describe the manner in which the test environment may be different from the operational environment and the effects of this difference on the tests.)

1.3 References

(List applicable references, including:

* Requirements documents;
* System/Subsystem, Program, and Data Specifications;
* User's Manual;
* Operations Manual;
* Program Maintenance Manual;
* Development Plan;
* Strategic Test Plan;
* Detailed Test Plan;
* Other relevant documentation.)

2. TEST RESULTS AND FINDINGS

(Provide detailed information for each test separately in paragraphs 2.1 through 2.n.)

2.1 Test (Identify)

(Describe testing scenarios, techniques and tools, test strategies, and test results and findings.)

2.2 Test (Identify)

(Present testing information on second and succeeding tests in a manner similar to 2.1.)

3. SOFTWARE FUNCTION FINDINGS

(Identify and describe conclusions that can be drawn from all test results with respect to the functions defined in the Requirements Document and Specifications.)

3.1 Functions (Identify)

(Briefly describe the function and software being tested, and the tests themselves. Include the range of data values tested. Report findings, including deficiencies, limitations, and constraints.)

3.2 Function (Identify)

(Report findings on the second and succeeding functions in a manner similar to paragraph 3.1.)

4. ANALYSIS SUMMARY

4.1 Capabilities

(Summarize the capabilities of the software as demonstrated by the tests. If tests were conducted to demonstrate that one or more performance requirements are fulfilled, then compare the results with these requirements. Assess the effects that any difference in the test environment as compared to the operational environment may have had on this demonstration of capabilities.)

4.2 Deficiencies

(Describe the deficiencies of the software as demonstrated by the tests. Describe the impact of each deficiency on the performance of the software and the overall impact on performance of all detected deficiencies.)

4.3 Recommendations and Estimates

(Estimate the time and effort required to correct each deficiency. State how urgent the correction is, how it should be made, and who should make it.)

SOFTWARE VALIDATION REPORT

The Software Validation Report describes methods used and results obtained in final validation of software before its delivery for installation at prospective user sites. The Report is used by the ICAM Program Office to confirm that the software will serve its intended application effectively and that development has progressed satisfactorily. The Report also serves to inform software users and other ICAM participants of validation practices and their effectiveness.

ANNOTATED OUTLINE

1. GENERAL INFORMATION

1.1 Summary

(Summarize the software involved, its functions, characteristics, and intended applications. Briefly describe the scope of the validation that was performed.)

1.2 Environment

(Identify the developer and the validation participants. Define the supporting hardware and software for the system, the prospective user sites, and any qualifications or restrictions on the validation that relate to intended installations and applications.)

1.3 References

(List applicable documents, including the Requirements Document, Development Plan, and pertinent specifications.)

2. VALIDATION TASKS

(Describe the validation process as distinct tasks, including a summary of the progressive stages of testing. Summarize the disciplines and procedures of each validation task, and clearly indicate the pertinent results to be detailed later in this Report. Indicate the role or responsibility of each participating organization in each task. Summarize the resources used, and depict the sequence, schedule, and interdependence of all tasks performed.)

3. VALIDATION ORGANIZATION

(Describe fully the organizations, principal personnel, and their responsibilities in the complete validation effort. Define significant working procedures and relationships, with reference to the tasks described above.)

4. VALIDATION RESULTS

(Describe fully the results obtained from each distinct validation task, and state the conclusions warranted by these results in regard to software quality, validity, reliability, and overall readiness for delivery to prospective users. Include a detailed description of the methods actually used in each task, including personnel involved, time and other resource expenditures, support tools and background data used, progressive or partial results and their disposition in the final outcome of each task. Summarize testing results, and also define the efforts made and assurances produced from correlation of source code and documentation, audits of source code with respect to all applicable standards, physical configuration audits, interface audits, reliability measurement and prediction, correlation of test cases and requirements, simulations, shake-down or dry runs, etc.)

5. DISCREPANCIES AND VALIDATION

(List completely in summary form all the discrepancies identified by the validation process and their proposed resolution. Provide a conclusive statement concerning the present readiness of the software for ICAM use and the extent to which ICAM requirements and development objectives have been met.)

THIS PAGE DELIBERATELY

LEFT BLANK

SOFTWARE SUMMARY

The Software Summary (FIPS 30) describes a computer program or automated data system on Standard Form 185. Its audience includes technical and managerial personnel from the contractor's own company, from the ICAM Program Office, from other ICAM participants, and from other manufacturing industries. The contractor shall deliver it after he has completed testing and validating the software described by the form.

1. INSTRUCTIONS FOR COMPLETING THE SOFTWARE SUMMARY

(Obtain Standard Form 185 from the nearest GSA supply office and follow the instructions on the back of the form, with the following exceptions:

10. Application Area. Within the area labeled "Specific" describe the software's particular manufacturing application in the developer's industry.

13. Narrative. Besides the recommended information, suggest potential applications of the software in other industries.)

USER'S MANUAL

The User's Manual describes software functions in user-oriented terminology. The manual serves as the primary reference document for the user. This document is prepared to serve users with a wide range of technical sophistication and experience, including novices, managers, and computer scientists. The developer begins preparing the User's Manual during the Programming stage of the software development life cycle. The manual is revised and updated until the software has been validated, and the resulting final document is delivered with the system.

ANNOTATED OUTLINE

1. GENERAL INFORMATION

1.1 Summary

(Summarize the application and general functions of the software.)

1.2 Environment

(Identify the system developer, the potential manufacturing functions, the hardware and software interfaces, and the possible operating sites of the software system.)

1.3 References

2. DESCRIPTION OF SOFTWARE APPLICATION

(Describe the basic concepts, goals, and design philosophy of the software system. Discuss its functional and performance capabilities, its basic elements, the functions it performs, and the services it allows. Describe the structure of the software, the role of each component in the operation of the software, and the operational relationships of this software to other software. If used, graphical representations can be appended to this section. Describe in detail the system's processing operations, relationship between input and output, and databases or data files that are referenced, accessed, or maintained by the software. Describe minimal equipment configuration required for running the software.)

3. PROCEDURES AND REQUIREMENTS

3.1 Commands and Procedures

(Name and describe:

* Operating commands;
* Protocols used;
* System calls to support software or utilities, required parameters and formats;
* Characteristic responses and potential problems;
* Special operating requirements.)

3.2 Initiation

(Describe step-by-step procedures required to initiate processing.)

3.3 Input

(Describe preparation requirements for input data: frequency, volume limitation, input formats, priority and security restrictions, sample input, and all necessary parameters. In interactive processing where input is minimal, this section may simply highlight any factors--such as data standards or CRT screen positions--that may demand user attention. High-volume interactive situations may require such information as limits on file size or number of users.)

3.4 Output

(Describe the requirements relevant to each output. Include such information as use, frequency, output format, and sample output. For interactive processes, an item like output format might simply indicate CRT screen positions for data.)

3.5 Error and Recovery

(Describe conditions that will generate error codes or messages. Include simple and detailed explanations of error messages along with ways to isolate and correct particular problems. Indicate procedures to be followed by the user to ensure that restart and recovery procedures can be used.)

THIS PAGE DELIBERATELY

LEFT BLANK

OPERATIONS MANUAL

This document is used by technical staff. The Operations Manual describes in detail the software's operating characteristics and its associated environment, so that computer operators and programming personnel can run the software. The Operations Manual is drafted during the Programming stage, and it is refined and updated until the system is validated. This manual is delivered with the system.

ANNOTATED OUTLINE

1. GENERAL INFORMATION

1.1 Summary

(Summarize the general functions and operating characteristics of the software, including its purpose and use.)

1.2 Environment

(Identify the system developer, the potential manufacturing functions, the hardware and software interfaces, and the possible operating sites of the software system.)

1.3 References

(List Applicable references, such as:

* User's Manual;
* Program Maintenance Manual;
* Installation Guide;
* Other pertinent documentation on the project.)

2. OVERVIEW

2.1 Software Organization

(Describe the software's inputs, outputs, required data files, and sequence of operations. Describe, with the aid of charts or diagrams, how various modules are interrelated. Identify the software's constraints, such as priority of execution, input or output requirements, dependence on other software, and security. Describe groupings within the software.)

2.2 Program Inventory

(Identify each program by title, number, and mnemonic reference.)

2.3 File Inventory

(Identify each file that is referenced, created, or updated by the system. Include the title, the mnemonic reference, storage medium, required storage, and access information.)

3. DESCRIPTION OF RUNS

(Provide explicit instructions that the computer operator or programmer must follow for running the software. Specifically indicate where operator intervention is required. Include such information as:

* Protocol required to access a system;
* Operating system interaction, if any;
* Job control cards (or instructions) required to set up a runstream (or a session);
* Calling instructions and sequences for program modules;
* Calling instructions and sequences for support software, such as utilities, system routines, or other application routines;
* Input and output preparation requirements;
* Operator messages, and required action;
* Restart and recovery procedures.)

4. NONROUTINE PROCEDURES

(Provide any information necessary concerning emergency or nonroutine operations, such as switchover to a back-up system in case of failure, and procedures for turnover to maintenance programmers.)

5. REMOTE OPERATIONS

(Describe the procedures for running the program through remote terminals.)

PROGRAM MAINTENANCE MANUAL

The Program Maintenance Manual explains in detail the programs, their operating environment, and their maintenance procedures. This document is prepared for the maintenance programmers and the operators. It provides them with necessary technical information for understanding the programs, their operating environment, and their maintenance procedures. Preparation of the Program Maintenance Manual should begin during programming and verification, and it should be refined and updated until the system is validated. This document is delivered along with the system and other documentation.

ANNOTATED OUTLINE

1. GENERAL INFORMATION

1.1 Summary

(Summarize the characteristics, functions, special requirements, and procedures for maintaining the system.)

1.2 Environment

(Identify the maintenance team, the potential manufacturing functions, the hardware and software environment, and the possible operating sites of the software system.)

1.3 References

(List applicable references, including:

* User's Manual
* Installation Guide
* Operations Manual
* Test Plans and Reports
* Other relevant project documentation.)

2. PROGRAM DESCRIPTIONS

(In this section, identify all the programs in the software system, and individually describe each program component.)

2.1 Program Description (Identify by Name)

(Identify the program to be maintained. Describe the problem that the program solves, and the solution method used, including algorithm chosen and programming language in which the program is implemented. Describe input requirements, processing features, expected output, storage media, interfaces with other software, and procedures to run the program, including loading, terminating, and error handling. Explain programming conventions used.)

2.2 Program Description (Identify by Name)

(Provide the description, as stated above, of each program in the software system.)

3. OPERATING ENVIRONMENT

(This section describes the overall operating environment of the software system.)

3.1 Hardware

(Identify the equipment required for the operation of the system. Include information about the processor used, storage media, peripheral devices, and special features.)

3.2 Support Software

(Identify the support software required for each program. Identify the operating system, compiler or assembler, database management system, report generator, or any other software required by this program.)

3.3 Database

(Identify and describe the database used, and refer to relevant documentation on the database, including a data element directory.)

4. MAINTENANCE PROCEDURES

4.1 Verification Procedures

(Describe the verification procedures to check the performance of the program, either in general, or following modification. Include a reference to test data and procedures.)

4.2 Error Conditions

(Describe error conditions (including those not previously documented elsewhere), their origin, and the recommended method for correcting them.)

4.3 Special Maintenance Procedures

(Describe any special procedures required for maintaining the software. Include procedures and recommended frequency for preventive maintenance. Identify and describe special programs used for maintenance, such as file restoration programs and file purger.)

4.4 Listings and Flowcharts

(Refer to or else append the listing and flowcharts.)

4.5 Regression Testing

(Regression Testing is performed after a significant change is made to the software, either for functional improvement or for repair. Its purpose is to determine if the change has affected adversely other parts of the software. In this section, describe the tests performed and the test data and test cases used.)

INSTALLATION GUIDE

The Installation Guide specifies options, parameters, and procedures for configuring and loading the software in the various planned operating environments, and for performing acceptance tests on the installed software system. This Guide is prepared for computer operations technical personnel in the user's organization. Initial preparation of this document begins during the Programming and Verification stage. It is revised during Testing and Validation and delivered with the system and other system documents.

ANNOTATED OUTLINE

1. GENERAL INFORMATION

1.1 Summary

(Summarize the purpose and general functions of the system, identify the resources and materials needed, and summarize the general requirements and procedures for installing the system.)

1.2 Environment

(Identify the system developer, potential manufacturing users, hardware and software environments -- including interfaces -- and possible operating sites of the software system.)

1.3 References

(List applicable references, such as:

* User's Manual;
* Operations Manual;
* Maintenance Manual;
* Related Project Documentation.)

2. OVERVIEW

2.1 General System Description

(Describe the basic concepts, goals, and design philosophy of the software system. Discuss the functions it performs, its capabilities, and its characteristics. Explain any particular feature that will be crucial to successful and efficient installation.)

2.2 Installation Planning and Preparation

(Describe administrative and preparatory requirements and procedures, such as

* Choosing an approach to installation, i.e., installing immediately, installing in phases, or parallel processing;

* Scheduling computer time, and preparing the computer site and application environment;

* Organizing an acceptance testing team;

* Devising training plans for maintenance personnel;

* Checking that all system documentation is complete;

* Performing required conversion; and

* Making provisions for backup and recovery, safety, and security.)

3. INSTALLATION

3.1 General Requirements for Installation

(Describe the basic requirements for installing the software, such as:

* Hardware: List minimum hardware requirements for installation, e.g., mainframe (including model and series), memory requirements, storage requirements, peripheral equipment, such as disk drives, tape drives, line printers, communication processors, and other special equipment.

* Software: Specify other software required, such as operating system (include version and level required), compiler (include version and level required), utility routines, database management software, and other special purpose software.

* Data: Specify names of databases (or files) that the system will need to use; indicate sources of data and procedures for collecting data.

* Installation Options: Define and discuss options or parameters available for optimizing or tailoring the software to a given type of computer system or application environment. Provide guidelines on proper selection of parameters and options. Describe standards implications fully.

* Other Requirements: Indicate other requirements for installing the software system.)

3.2 Software System Components

(Enumerate and describe all the components of this software system. For each component, program, or module, describe its functions and its relationship to other components.)

3.3 Procedures

(Describe, in sequence, the procedures for installing the system:

* Describe required initialization processes, such as initializing files or databases, creating a dictionary or a directory, etc.

* Provide detailed instructions on how to proceed with the installation, including the order in which the software components should be installed, the parameters that must be used, and the required equipment and support software.

* Include procedures for linking components and compiling the software. Describe the sequence of execution by component names, and list and describe the database, or files, generated and accessed.

* Provide detailed procedures for correcting installation errors, and for starting over again.

* Provide directions for installing the system after the first time.

* Identify typical problems that may occur during installation, describe how each problem is solved, and recommend means of preventing future problems of this kind.)

3.4 Installation Demonstration

(Describe installation demonstrations or tests that are conducted prior to formal acceptance testing, to ensure that the system has been correctly installed and is operational.)

4. ACCEPTANCE TESTING

(Acceptance Testing is performed to verify that the software system operates successfully in a live environment, that it satisfies objectives and performance criteria defined in the Requirements Document, and that the system can serve the intended application. Describe fully the acceptance testing scenario, include requirements on the acceptance testing team, types of tests conducted, procedures, and expected results of the testing.)

5. TRAINING

(Describe in detail the types of formal training required by the operator and user of this system.)

THIS PAGE DELIBERATELY

LEFT BLANK

MAINTENANCE PLAN

The Maintenance Plan describes the objectives, organization, responsibilities, criteria, and schedule for maintenance of important ICAM software. It is prepared after completion of the software development, and is used by the ICAM Program Office to monitor maintenance effort and to inform software users of the maintenance being provided.

ANNOTATED OUTLINE

1. GENERAL INFORMATION

1.1 Summary

(Identify the software systems covered by this Plan, and their general functions and characteristics. Summarize the scope of this Maintenance Plan, pointing out the level of effort planned, objectives and criteria, and considerations important to users.)

1.2 Environment

(Identify the maintenance organization or contractor, and other participants in maintenance effort. Identify the supporting hardware and software involved in the maintenance activity.)

1.3 References

(List applicable documents, including the Maintenance Manual and other manuals issued for the software.)

2. MAINTENANCE TASKS

(Describe the planned maintenance work as distinct tasks for each software system covered by this Plan. Define the conditions and criteria applicable to initiation and completion of each task, the basic disciplines required, and the type of results expected. Identify the role and contribution of each party in the maintenance effort, and the resources provided where known or estimable. Show the sequence, schedule, and interdependence of all tasks.)

3. MAINTENANCE ORGANIZATION

(Describe fully the organizations, principal personnel, and responsibilities for maintenance. Define the necessary procedures, working relationships, and intercommunication.)

4. MAINTENANCE METHODOLOGY

(Describe the technical criteria, techniques, tools and support software, and maintenance practices to be used, unless these are fully and accurately treated in the Maintenance Manual. Define day-to-day working methods, audits, inspections, and other quality assurance practices. Describe the application of specific standards to maintenance tasks and results. Define documentation produced to inform users of software changes and to maintain accurate configuration data on the software system. Define quantitative criteria for maintenance performance and software quality.)

THIS PAGE DELIBERATELY

LEFT BLANK

APPLICATION REPORT

The Application Report tells how a given user organization applies the software system in its particular environment: what functions the software performs, what benefits it produces, and what special or general purpose changes or improvements the organization has developed. Written for both technical and managerial personnel, the Application Report is prepared during the Operation stage of the software life cycle. Its purpose is to give other potential users a clear example of the functions, flexibility, and value of the software system.

ANNOTATED OUTLINE

1. GENERAL INFORMATION

1.1 Summary

(Name the software and generally describe its application in the user's organization.)

1.2 Environment

(Identify the user's company and the operating site of the software. Name and describe the equipment on which the software runs, and describe interfaces with other software. Briefly describe the kinds of personnel who operate the software system.)

1.3 References

(List applicable references, such as:

* The Feasibility Study and Cost/Benefit Analysis for the software system;
* Related applications software or documentation;
* Documents from related projects; and
* Relevant standards or guidelines, including FIPS publications, DIDs, and DoD manuals.)

2. THE SOFTWARE PACKAGE

(Describe the software package in its delivered form. Use the general topic headings described below.)

2.1 Function

(Describe in detail the particular application of the software. What manufacturing function does it automate, and in what overall context?)

2.2 Adjustments

(Give the parameters used in installing the software on the user's own system. Specify relevant hardware and software interfaces.)

2.3 Modifications

(Describe in detail any modifications that were necessary before the new software could run on the user's system. Append or refer to the documentation for any new versions of modules or programs.)

2.4 Training

(Tell how the user organization trained its personnel to use the new software. Describe the kind of personnel who were trained and the kind, amount, and cost of training. If part of this training was necessary only for enhancements developed by the user, separate the training information into two distinct parts--one for the delivered software package, and another for user-developed enhancements.)

3. ENHANCEMENTS

(Describe any special features developed by the user to enhance the performance or usefulness of the software. Append or refer to the documentation for installing and using the software system.)

4. EVALUATION

4.1 Cost/Benefit Analysis

(Evaluate in detail the costs and benefits of the software. Include both nonrecurring costs like capital investment, software conversion, and training, and such recurring costs as salaries, maintenance contracts, and supplies. Compare these costs with quantifiable benefits like cost reduction due to reduced resource requirements, improved operating efficiency, increased productivity, and smaller error rates.)

4.2 Overall Evaluation

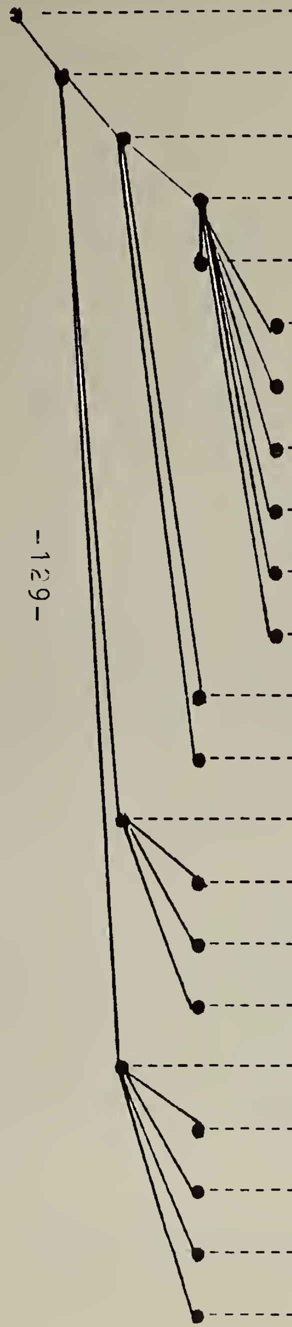
(Considering both tangible and intangible factors, estimate the value of the software system to the user organization.)

IDEF

Model

THIS PAGE DELIBERATELY

LEFT BLANK



NODE NUMBER	DIAGRAM TITLE
A-0	PROVIDE ICAM SOFTWARE DOCUMENTATION
A0	PROVIDE ICAM SOFTWARE DOCUMENTATION
A1	DEVELOP SOFTWARE AND SOFTWARE DOCUMENTATION
A11	PRODUCE ICAM SOFTWARE DOCUMENTATION
A11F	FEO DETAILING CONTROL INTERFACES
A111	ANALYZE NEEDS
A112	DEFINE SYSTEM REQUIREMENTS
A113	DESIGN
A114	PROGRAM AND VERIFY
A115	TEST, VALIDATE AND INTEGRATE
A116	OPERATE AND MAINTAIN
A12	MANAGE QUALITY AND CONFIGURATION
A13	CONFIGURE DOCUMENTS FOR ICAM
A2	ADMINISTER SOFTWARE DOCUMENTATION SYSTEM
A21	REVIEW PROJECT DOCUMENTATION
A22	EVALUATE SOFTWARE DOCUMENTATION SYSTEM
A23	DEVELOP PROPOSED ICAM SOFTWARE DOCUMENTATION STANDARDS
A3	DISSEMINATE SOFTWARE DOCUMENTS
A31	PREPARE DISSEMINATION RECORDS
A32	MANAGE DISSEMINATION TASKS
A33	PROVIDE INFORMATION SERVICES
A34	DISTRIBUTE DOCUMENTS

CONTEXT:

The ICAM Software Documentation System accomplishes the production, control, review, and distribution of ICAM results pertinent to computer programs and data, as tasks in the software life cycle. Documentation, as used here, refers to textual and associated graphic information intended for human use, as opposed to that which is solely for computer input and execution. Thus the term "documentation" does not encompass binary decks, load modules, and other strictly machine-readable information.

Software documentation typically includes application feasibility studies, software technology assessments, program specifications, manuals, program listings, and instructional material required to produce and use reliable, transferable software products. Software documentation is the technical and project information intended eventually for wide dissemination, and most directly oriented toward the development, maintenance, and use of ICAM computer programs and systems. Software documentation is only part of the total sccpe of documentation for the ICAM program. Typical information excluded from this category would be Air Force budget and program documents, contracts, management correspondence, and basic manufacturing or engineering results. The ICAM program has broader documentation needs that are not covered by the Software Documentation System model.

NODE:	TITLE:
DCC/A-0[illegible]1	CONTEXT--OPERATE ICAM SOFTWARE DOCUMENTATION SYSTEM

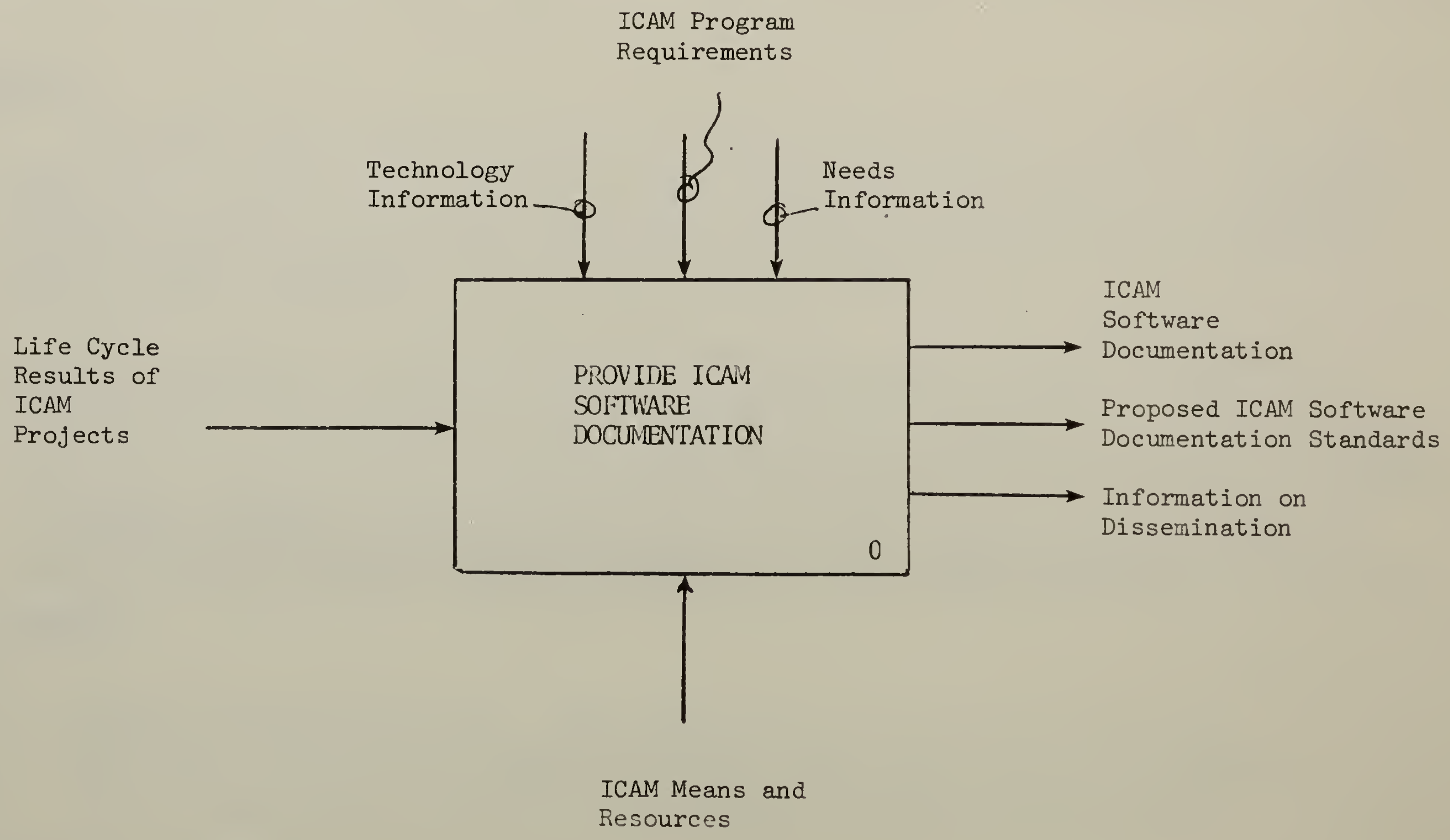
PURPOSE OF THE MODEL:

To show documenting activities, resultant documents, and the pertinent controls, especially documentation standards, in relation to key events in the life cycle of an ICAM software product. The model depicts overall requirements for the ICAM Software Documentation System. These requirements form the basis for the ICAM documentation standards.

VIEWPOINT:

The Software Documentation System is described from the viewpoint of all ICAM program participants, including AF managers, ICAM software users, and ICAM software developers.

NODE: DOC/A-0T2	TITLE: PROVIDE ICAM SOFTWARE DOCUMENTATION--TEXT

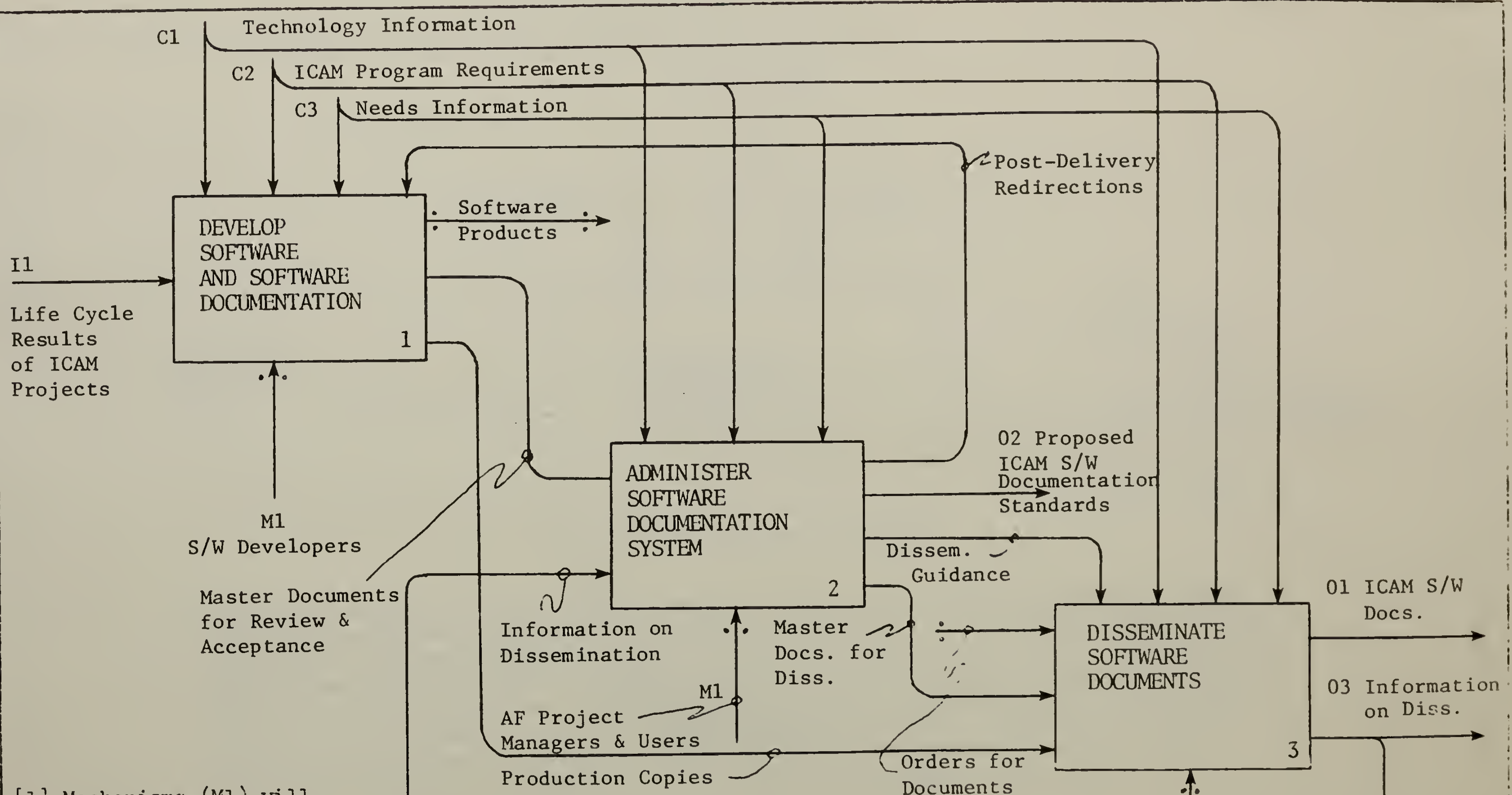

ICAM Program Requirements
Technology Information
Needs Information
Life Cycle Results of ICAM Projects
PROVIDE ICAM SOFTWARE DOCUMENTATION
0
ICAM Software Documentation
Proposed ICAM Software Documentation Standards
Information on Dissemination
ICAM Means and Resources

NODE: DOC /A-0

TITLE: PROVIDE ICAM SOFTWARE DOCUMENTATION

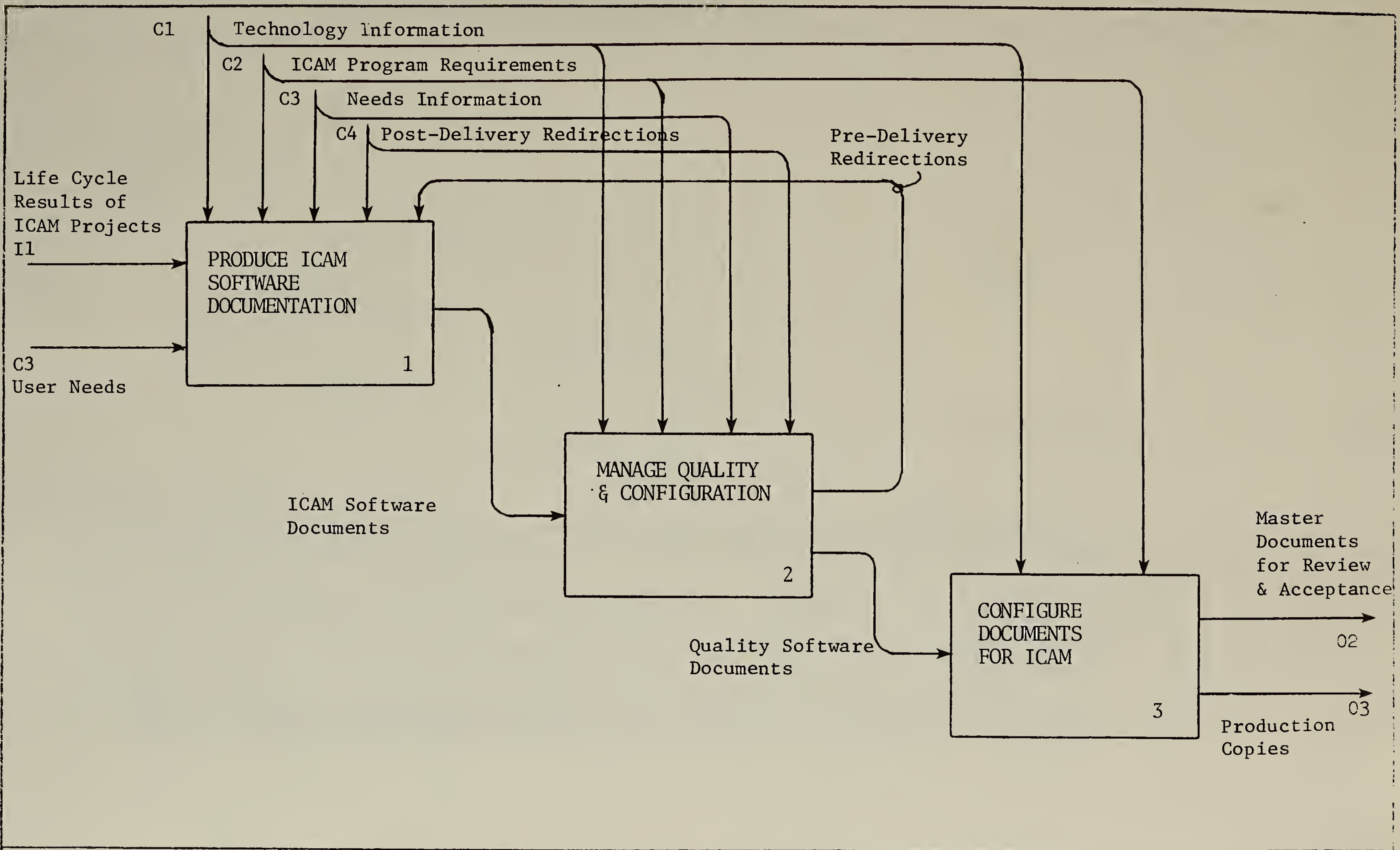
Software documentation is closely tied to software development; therefore, it is not easy to separate the documents from the functions that create them. Since the model refers primarily to the documents produced during the software development process, the A0 diagram shows a TUNNELED OUTPUT for the software produced during the software development process. Although there is a close relationship between software products and software documents, it is not within the scope and context of the present model to describe software product engineering further. The relevant software products will appear only at the A111 level, where the model details the software life cycle.

NODE:	TITLE:
DOC/AOT	PROVIDE ICAM SOFTWARE DOCUMENTATION--TEXT


C1
Technology Information
C2
ICAM Program Requirements
C3
Needs Information
Post-Delivery Redirections
I1
Life Cycle Results of ICAM Projects
DEVELOP SOFTWARE AND SOFTWARE DOCUMENTATION
1
Software Products
M1
S/W Developers
Master Documents for Review & Acceptance
ADMINISTER SOFTWARE DOCUMENTATION SYSTEM
2
O2 Proposed ICAM S/W Documentation Standards
Dissem. Guidance
Information on Dissemination
Master Docs. for Diss.
M1
AF Project Managers & Users
Production Copies
Orders for Documents
DISSEMINATE SOFTWARE DOCUMENTS
3
O1 ICAM S/W Docs.
O3 Information on Diss.
M1 AF and Contractors

[1] Mechanisms (M1) will will appear in future decompositions only if they can further enhance the diagram.

NODE: DOC/A0 | TITLE: PROVIDE ICAM SOFTWARE DOCUMENTATION


C1
Technology Information
C2
ICAM Program Requirements
C3
Needs Information
C4
Post-Delivery Redirections
Pre-Delivery Redirections
Life Cycle Results of ICAM Projects
I1
C3
User Needs
PRODUCE ICAM SOFTWARE DOCUMENTATION
1
ICAM Software Documents
MANAGE QUALITY & CONFIGURATION
2
Quality Software Documents
CONFIGURE DOCUMENTS FOR ICAM
3
Master Documents for Review & Acceptance
02
Production Copies
03

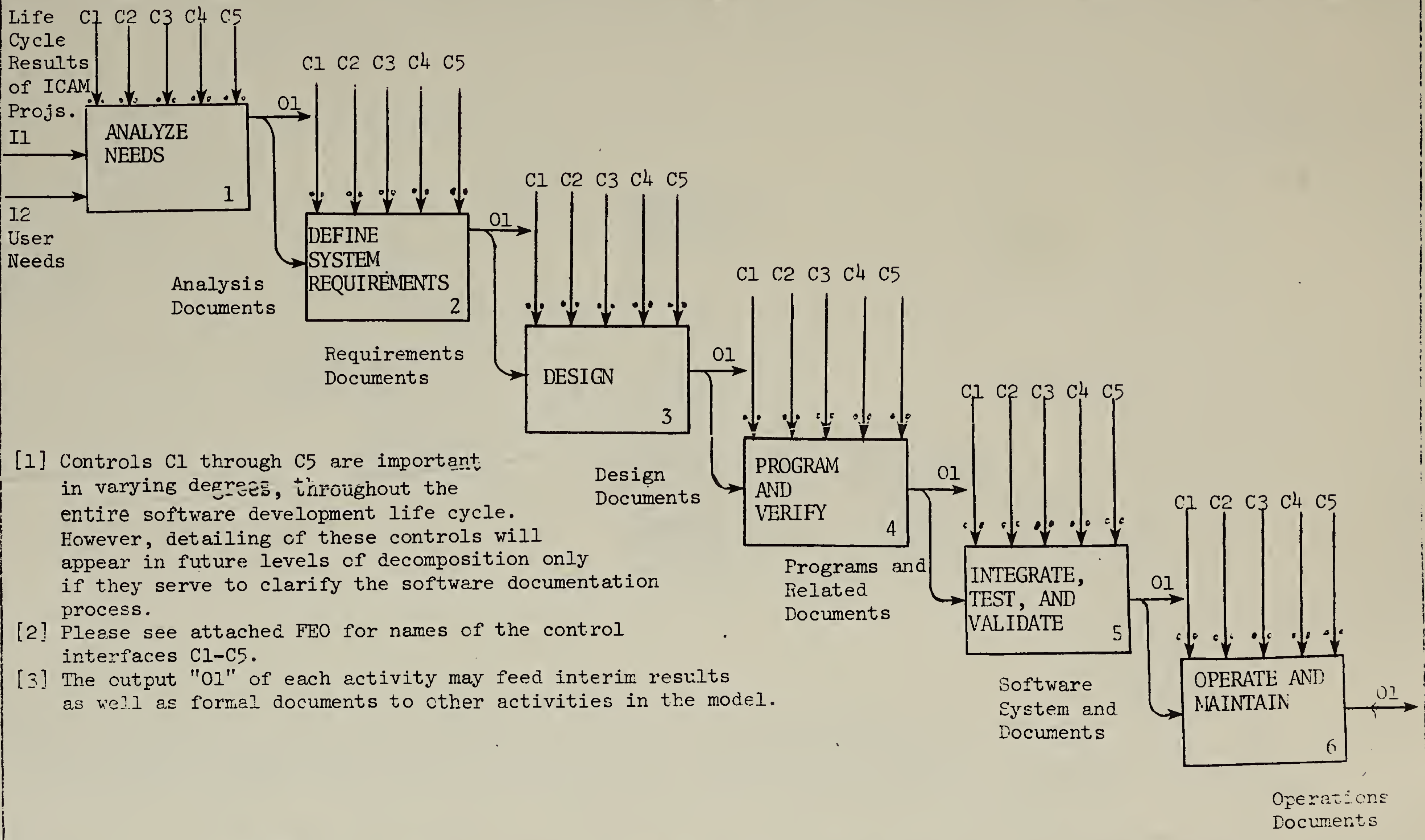
NODE: DOC/A1

TITLE: DEVELOP SOFTWARE AND SOFTWARE DOCUMENTATION

Looked at together, the decompositions of [A11]--"Produce ICAM Software Documentation"--present a highly structured view of the software life cycle. The model does not show the many informal iterations that contribute to successful software development. It shows instead a number of formal outputs, each of which may go through several levels of review. This textual explanation will substitute for the numerous feedback arrows that would only clutter the diagram. While reading the diagram, the reader should keep in mind the iterative nature of software development.

In the decompositions of [A11], and in [A12], italicized labels stand for documents that the contractor must produce in conformance to ICAM Software Documentation Standards.

NODE:	TITLE:
DOC/A11T	PRODUCE ICAM SOFTWARE DOCUMENTATION--TEXT



[1] Controls C1 through C5 are important in varying degrees, throughout the entire software development life cycle. However, detailing of these controls will appear in future levels of decomposition only if they serve to clarify the software documentation process.

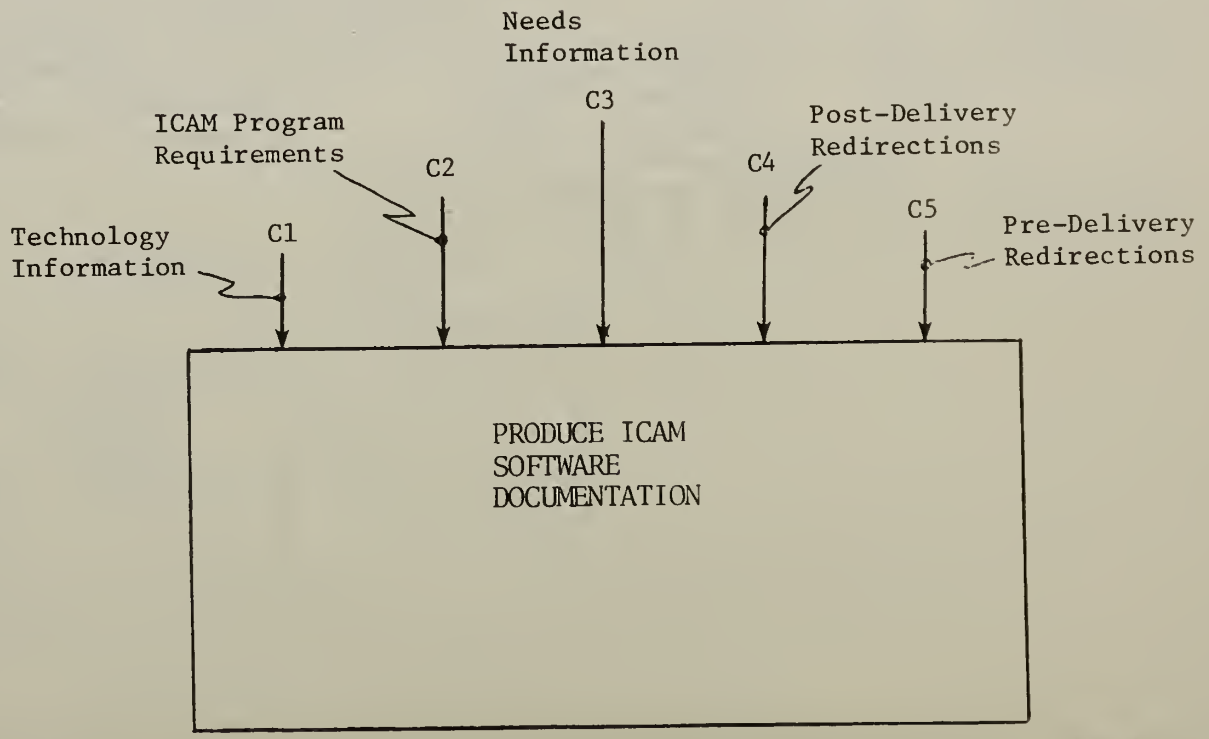
[2] Please see attached FEO for names of the control interfaces C1-C5.

[3] The output "O1" of each activity may feed interim results as well as formal documents to other activities in the model.

NODE: DOC/A11

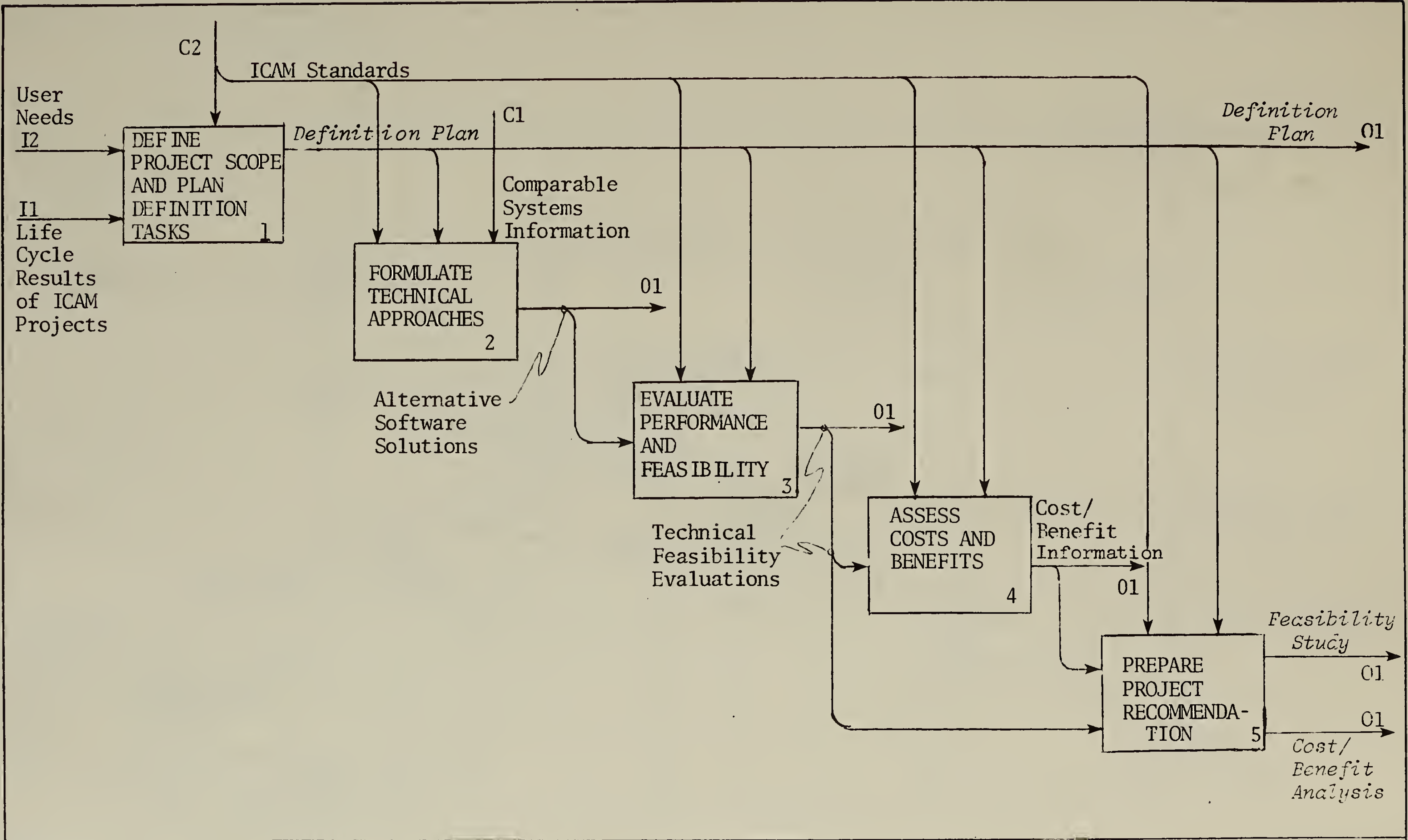
TITLE: PRODUCE ICAM SOFTWARE DOCUMENTATION

The control interfaces shown in the FEO diagram below are important to all processes in the Software Documentation System model. Since all of these controls are applicable to the entire model, they will be shown here, and will NOT appear in successive decompositions, unless a particular one contributes significantly to the document standardization process. Each control appearing below is defined in the glossary. In addition, individual outputs of each stage in the S/W Documentation life cycle may serve in certain cases as controls to the next stage. These will be shown in the appropriate places in the next decomposition.

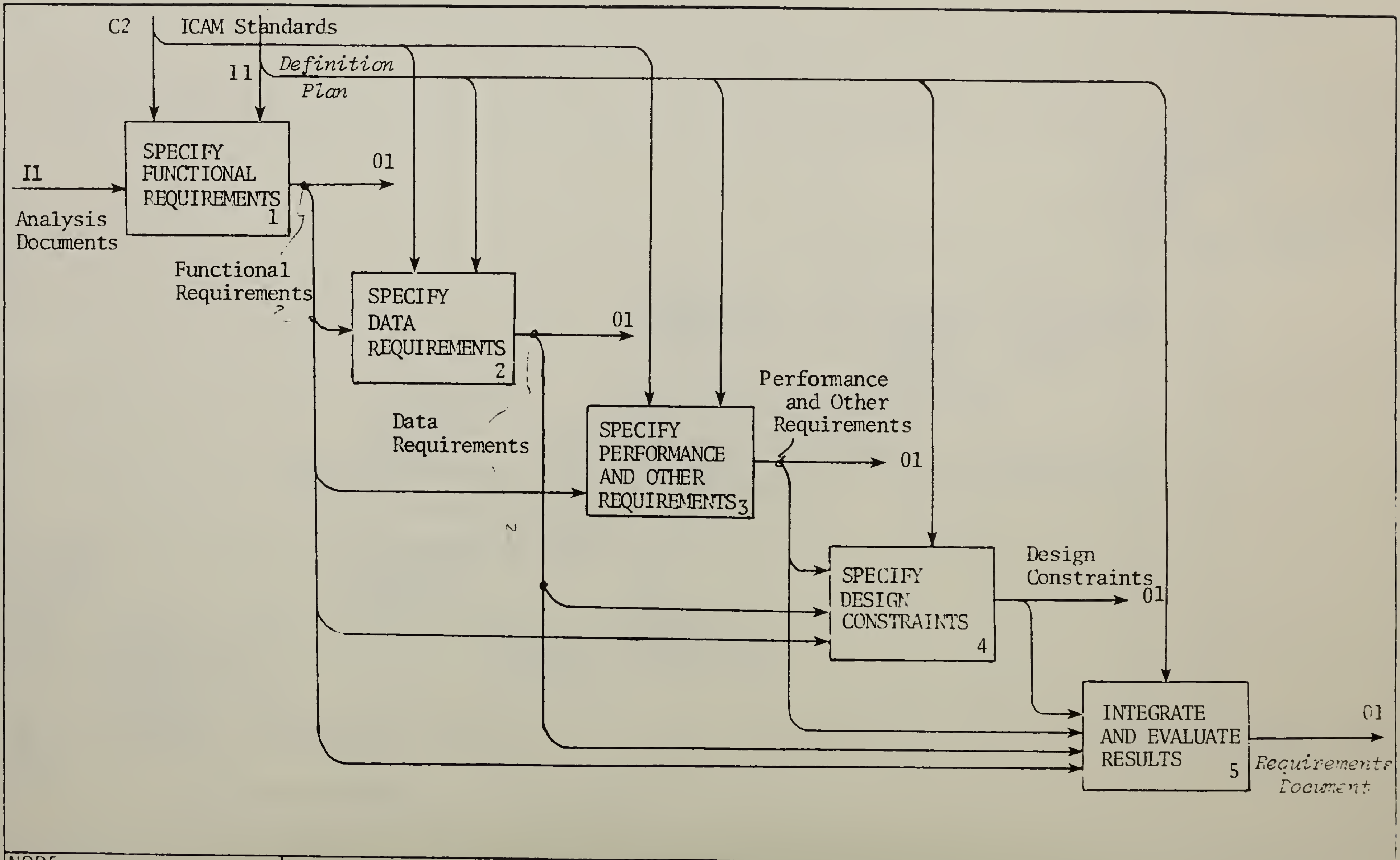

Needs Information
C3
ICAM Program Requirements
C2
Post-Delivery Redirections
C4
Technology Information
C1
C5
Pre-Delivery Redirections
PRODUCE ICAM SOFTWARE DOCUMENTATION

NODE: DOC/A11F

TITLE: FEO DETAILING CONTROL INTERFACES

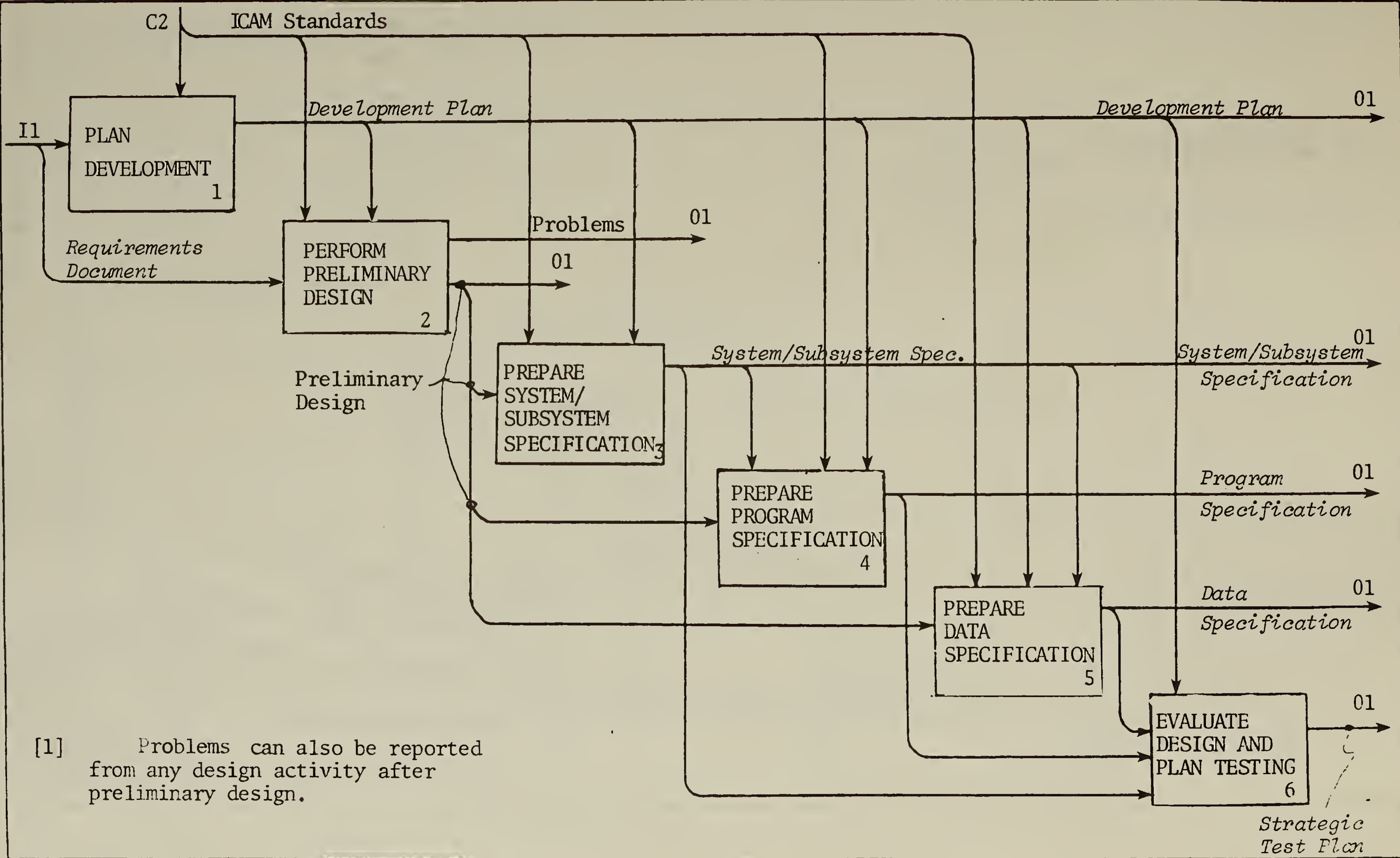

C2
ICAM Standards
User Needs I2
I1 Life Cycle Results of ICAM Projects
DEFINE PROJECT SCOPE AND PLAN DEFINITION TASKS 1
Definition Plan
C1
Comparable Systems Information
FORMULATE TECHNICAL APPROACHES 2
01
Alternative Software Solutions
EVALUATE PERFORMANCE AND FEASIBILITY 3
01
Technical Feasibility Evaluations
ASSESS COSTS AND BENEFITS 4
Cost/ Benefit Information
01
Definition Plan
01
PREPARE PROJECT RECOMMENDA-TION 5
Feasibility Study
01
01
Cost/ Benefit Analysis

NODE: DOC/A111 | TITLE: ANALYZE NEEDS


C2
ICAM Standards
I1
Definition Plan
I1
Analysis Documents
SPECIFY FUNCTIONAL REQUIREMENTS 1
01
Functional Requirements
SPECIFY DATA REQUIREMENTS 2
01
Data Requirements
SPECIFY PERFORMANCE AND OTHER REQUIREMENTS 3
Performance and Other Requirements
01
SPECIFY DESIGN CONSTRAINTS 4
Design Constraints
01
INTEGRATE AND EVALUATE RESULTS 5
01
Requirements Document

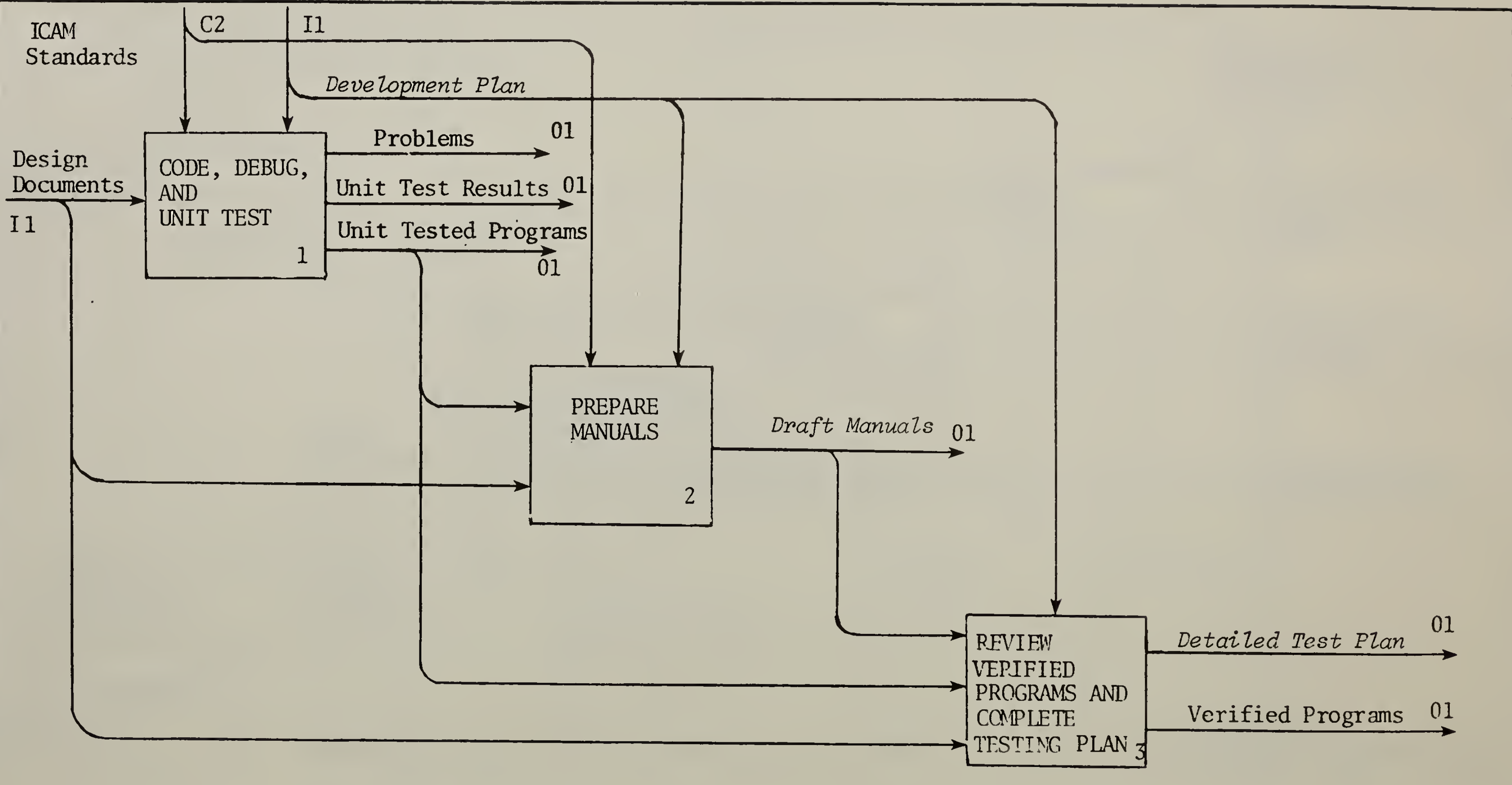
NODE DCC/A112

TITLE DEFINE SYSTEM REQUIREMENTS



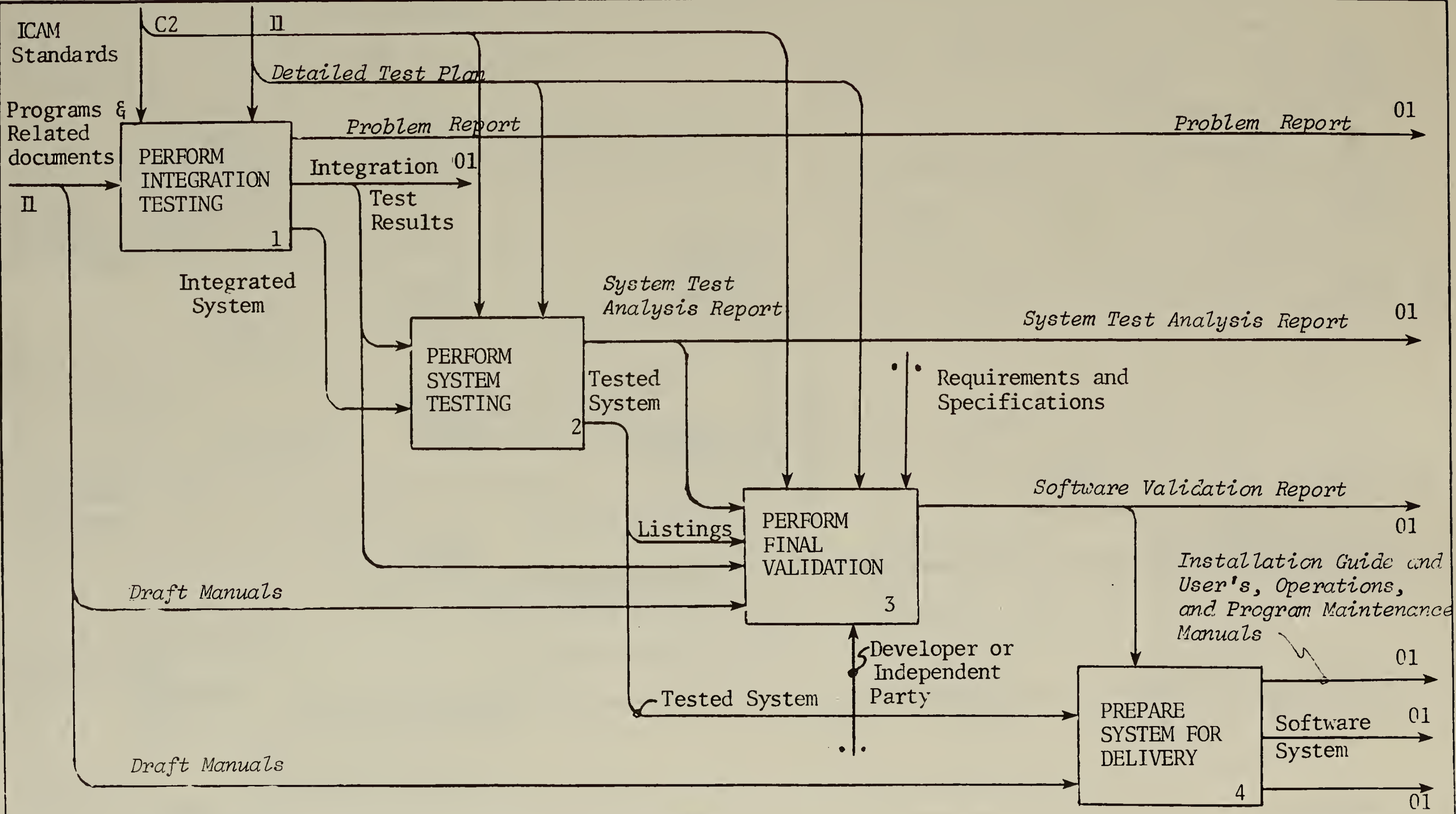
[1] Problems can also be reported from any design activity after preliminary design.

NODE DCC/A113	TITLE: DESIGN



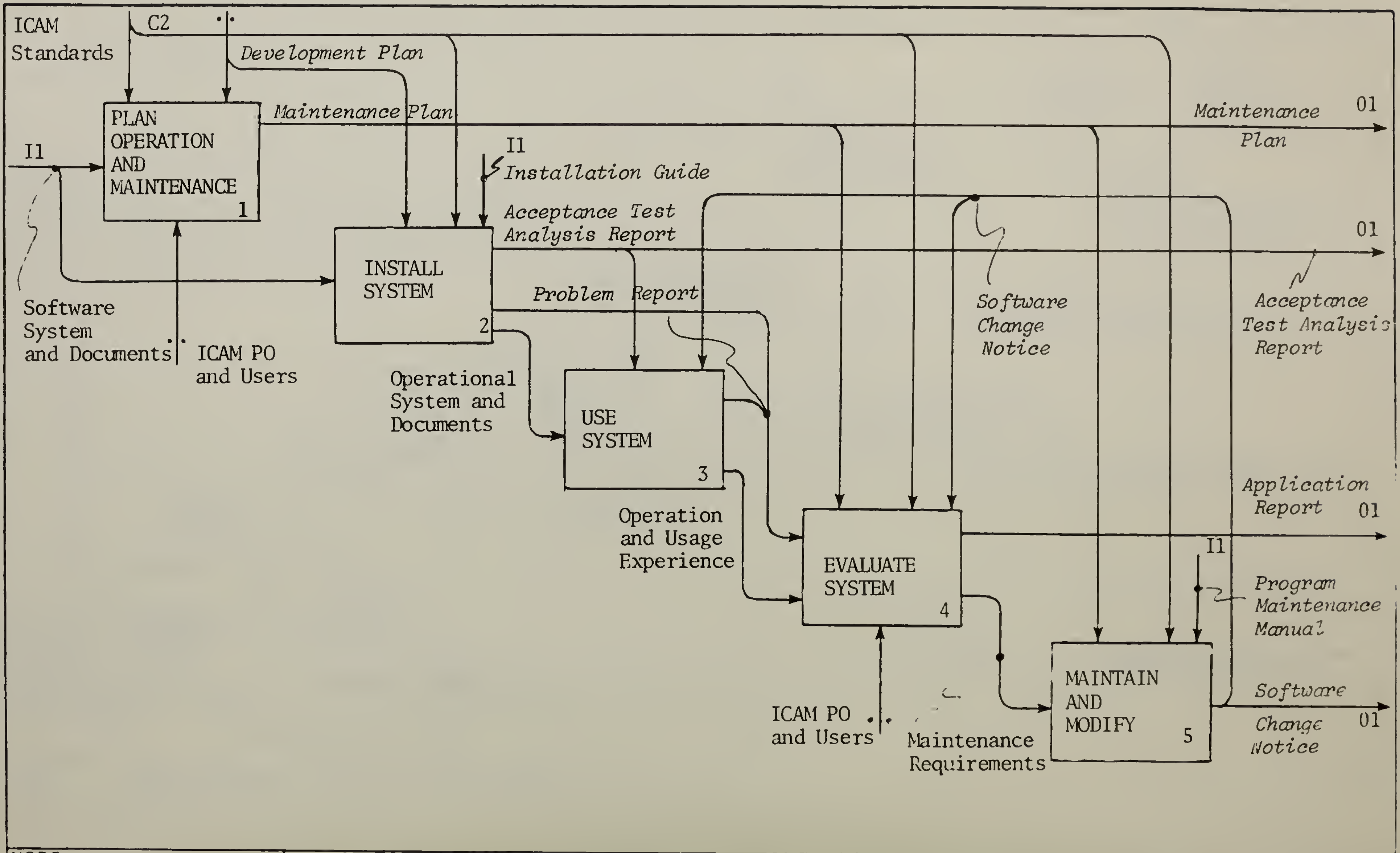
[1] Problems can also be reported from programming activities after unit testing.

NODE	TITLE:
DOC/A114	PROGRAM AND VERIFY

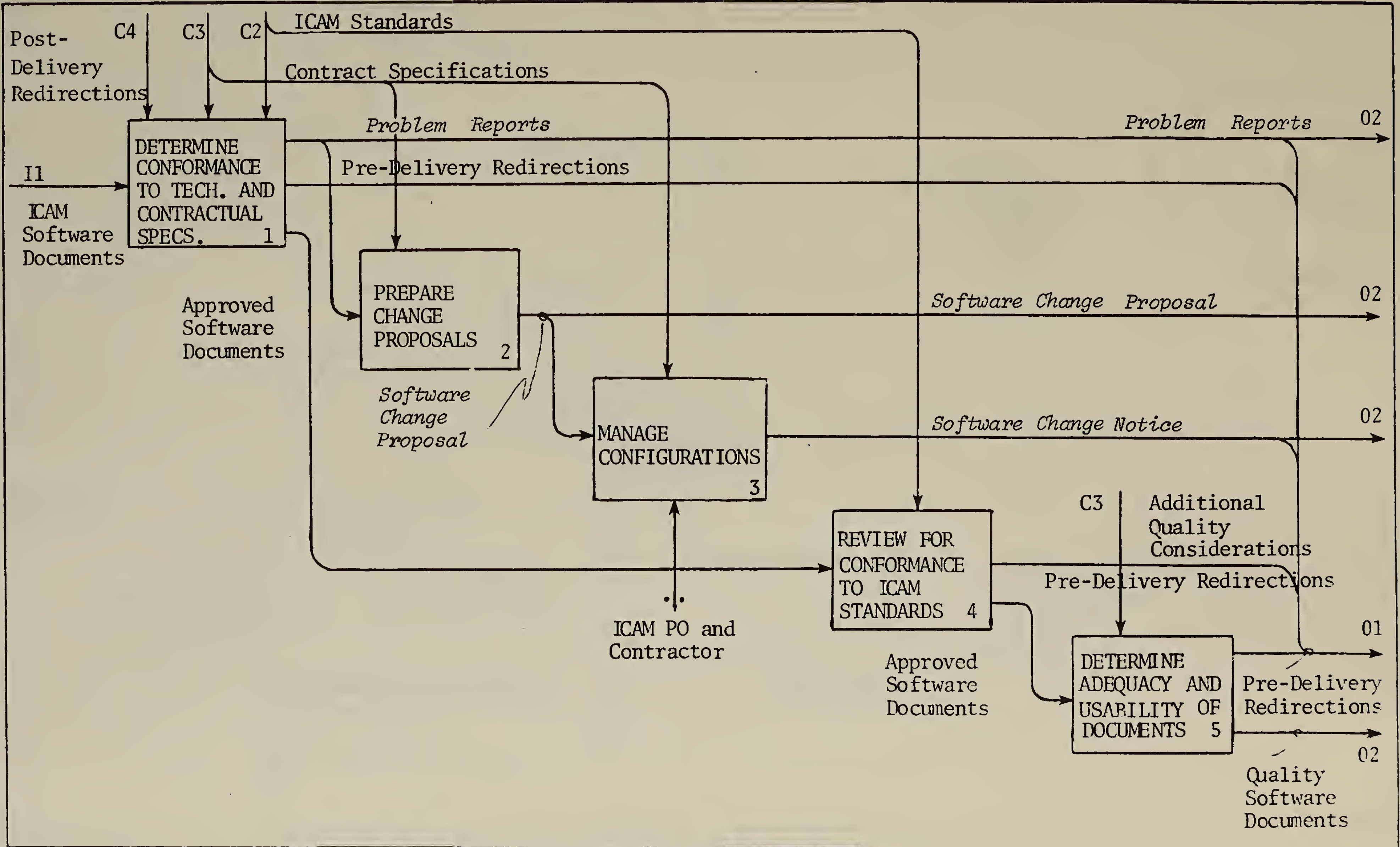

ICAM Standards
C2
I1
Detailed Test Plan
Programs & Related documents
I1
PERFORM INTEGRATION TESTING 1
Problem Report
Integration 01
Test Results
Integrated System
Problem Report 01
PERFORM SYSTEM TESTING 2
System Test Analysis Report
Tested System
System Test Analysis Report 01
Requirements and Specifications
Listings
PERFORM FINAL VALIDATION 3
Software Validation Report 01
Draft Manuals
Developer or Independent Party
Tested System
Installation Guide and User's, Operations, and Program Maintenance Manuals 01
PREPARE SYSTEM FOR DELIVERY 4
Software System 01
Draft Manuals
Software Summary 01

[1] *Problem Reports* can also be filed during system testing and final validation.

NODE	TITLE:
DOC/A115	INTEGRATE, TEST, AND VALIDATE



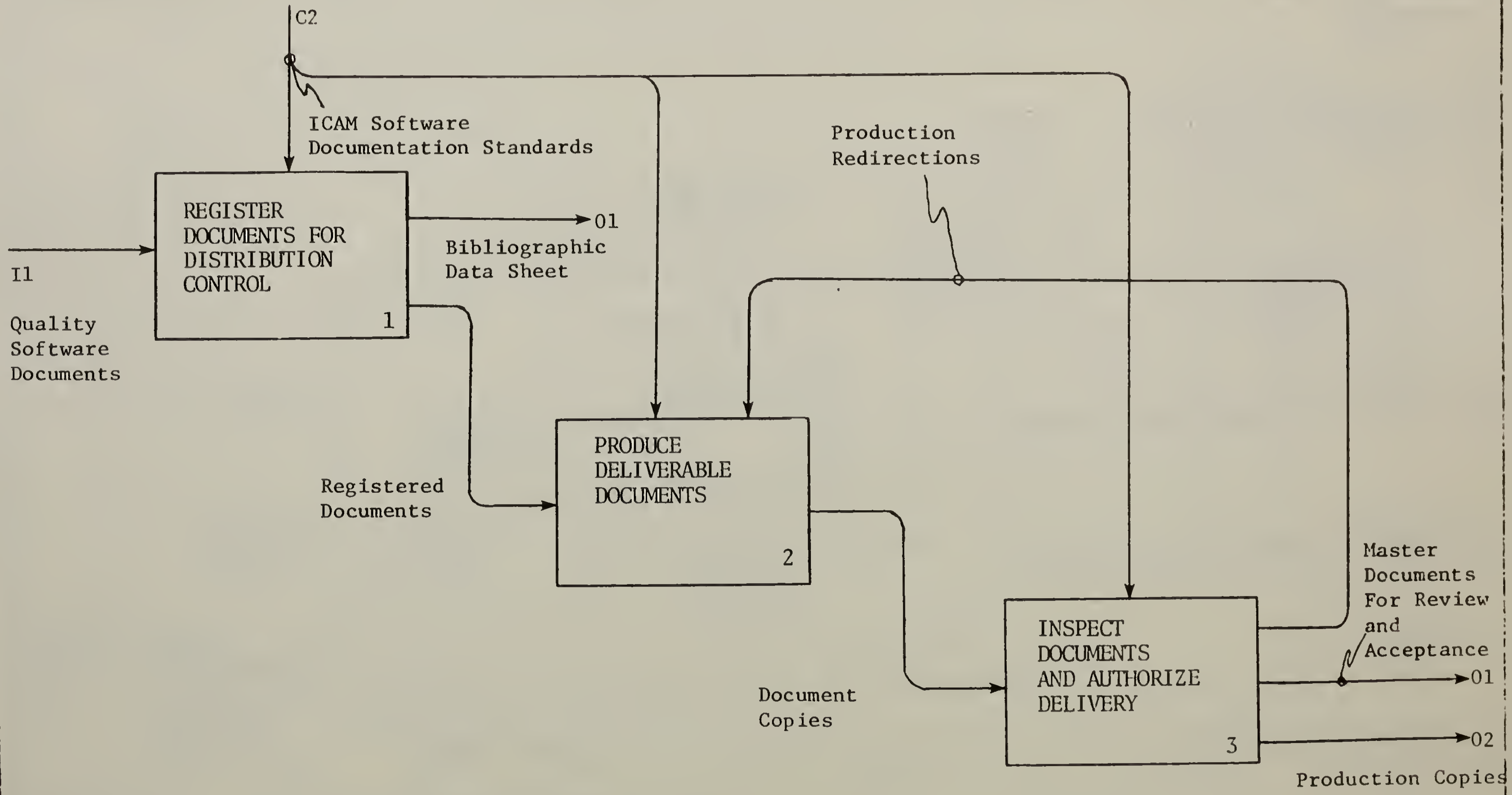
NODE:	TITLE:
DOC/A116	OPERATE AND MAINTAIN



Post-Delivery Redirections
C4
C3
C2
ICAM Standards
Contract Specifications
I1
ICAM Software Documents
DETERMINE CONFORMANCE TO TECH. AND CONTRACTUAL SPECS. 1
Problem Reports
Pre-Delivery Redirections
Approved Software Documents
PREPARE CHANGE PROPOSALS 2
Software Change Proposal
MANAGE CONFIGURATIONS 3
ICAM PO and Contractor
REVIEW FOR CONFORMANCE TO ICAM STANDARDS 4
Approved Software Documents
C3
Additional Quality Considerations
Pre-Delivery Redirections
DETERMINE ADEQUACY AND USABILITY OF DOCUMENTS 5
Problem Reports 02
Software Change Proposal 02
Software Change Notice 02
01
Pre-Delivery Redirections
02
Quality Software Documents

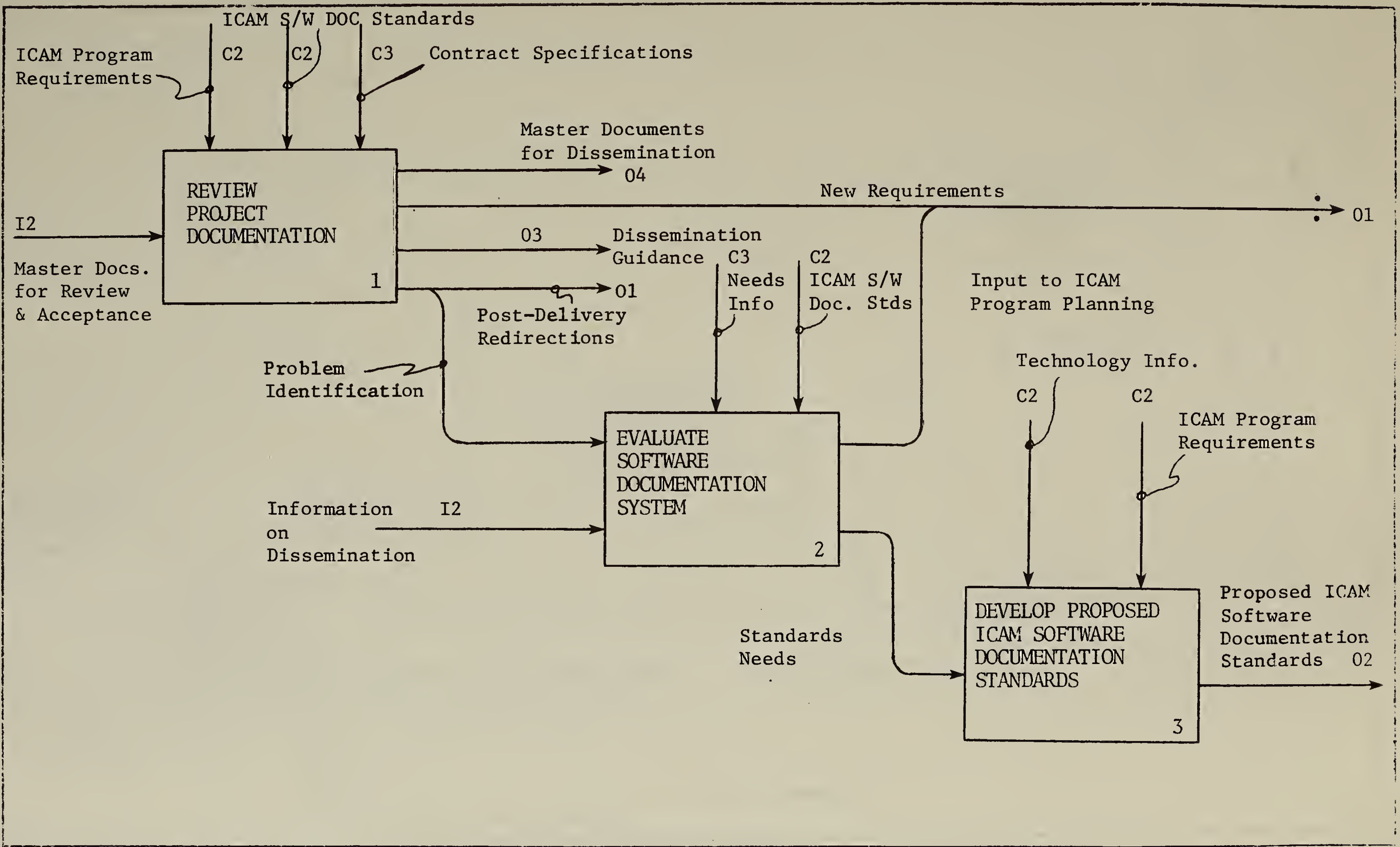
NODE: DOC/A12

TITLE: MANAGE QUALITY AND CONFIGURATIONS


C2
ICAM Software Documentation Standards
Production Redirections
REGISTER DOCUMENTS FOR DISTRIBUTION CONTROL
1
01
Bibliographic Data Sheet
I1
Quality Software Documents
PRODUCE DELIVERABLE DOCUMENTS
2
Registered Documents
Master Documents For Review and Acceptance
INSPECT DOCUMENTS AND AUTHORIZE DELIVERY
3
01
02
Document Copies
Production Copies

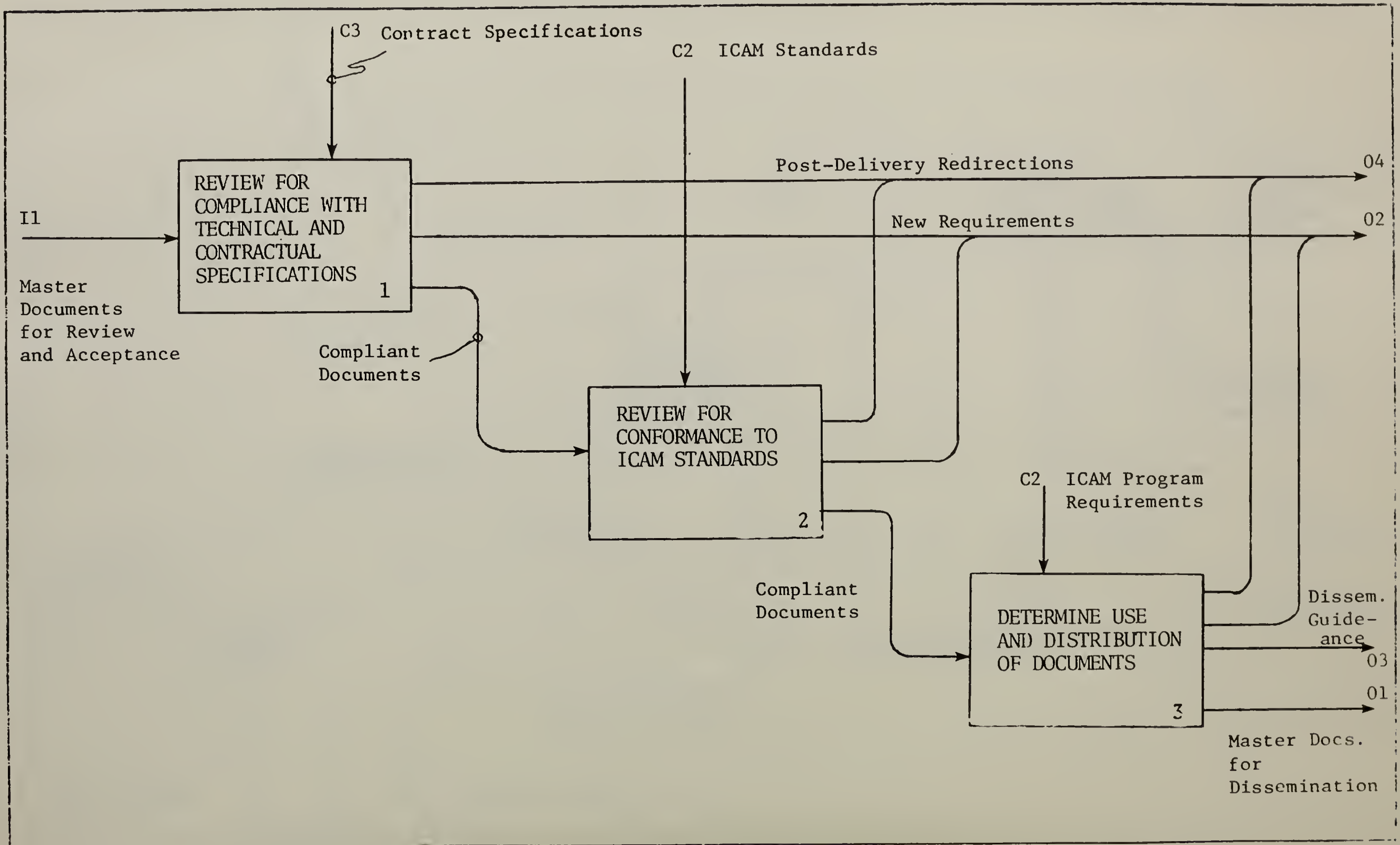
NODE: DOC/A13

TITLE: CONFIGURE DOCUMENTS FOR ICAM


ICAM S/W DOC Standards
ICAM Program Requirements
C2
C2
C3
Contract Specifications
Master Documents for Dissemination
04
New Requirements
01
I2
REVIEW PROJECT DOCUMENTATION
1
Master Docs. for Review & Acceptance
03
Dissemination Guidance
01
Post-Delivery Redirections
C3
Needs Info
C2
ICAM S/W Doc. Stds
Input to ICAM Program Planning
Problem Identification
Technology Info.
C2
C2
ICAM Program Requirements
EVALUATE SOFTWARE DOCUMENTATION SYSTEM
2
Information on Dissemination
I2
Standards Needs
DEVELOP PROPOSED ICAM SOFTWARE DOCUMENTATION STANDARDS
3
Proposed ICAM Software Documentation Standards 02

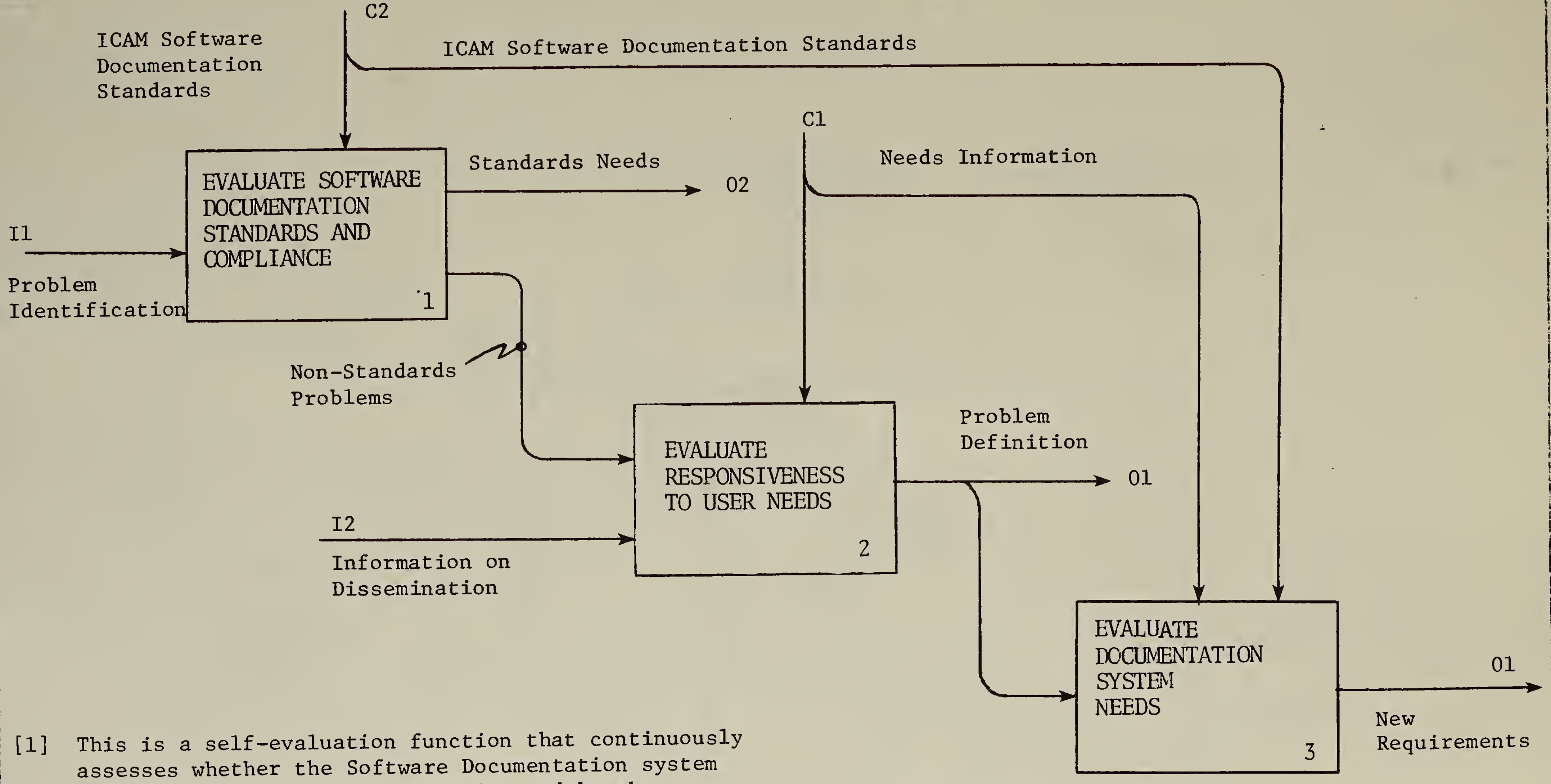
NODE: DOC/A2

TITLE: ADMINISTER SOFTWARE DOCUMENTATION SYSTEM


C3 Contract Specifications
C2 ICAM Standards
I1
Master Documents for Review and Acceptance
REVIEW FOR COMPLIANCE WITH TECHNICAL AND CONTRACTUAL SPECIFICATIONS
1
Compliant Documents
Post-Delivery Redirections
04
New Requirements
02
REVIEW FOR CONFORMANCE TO ICAM STANDARDS
2
Compliant Documents
C2 ICAM Program Requirements
DETERMINE USE AND DISTRIBUTION OF DOCUMENTS
3
Dissem. Guide-ance
03
01
Master Docs. for Dissemination

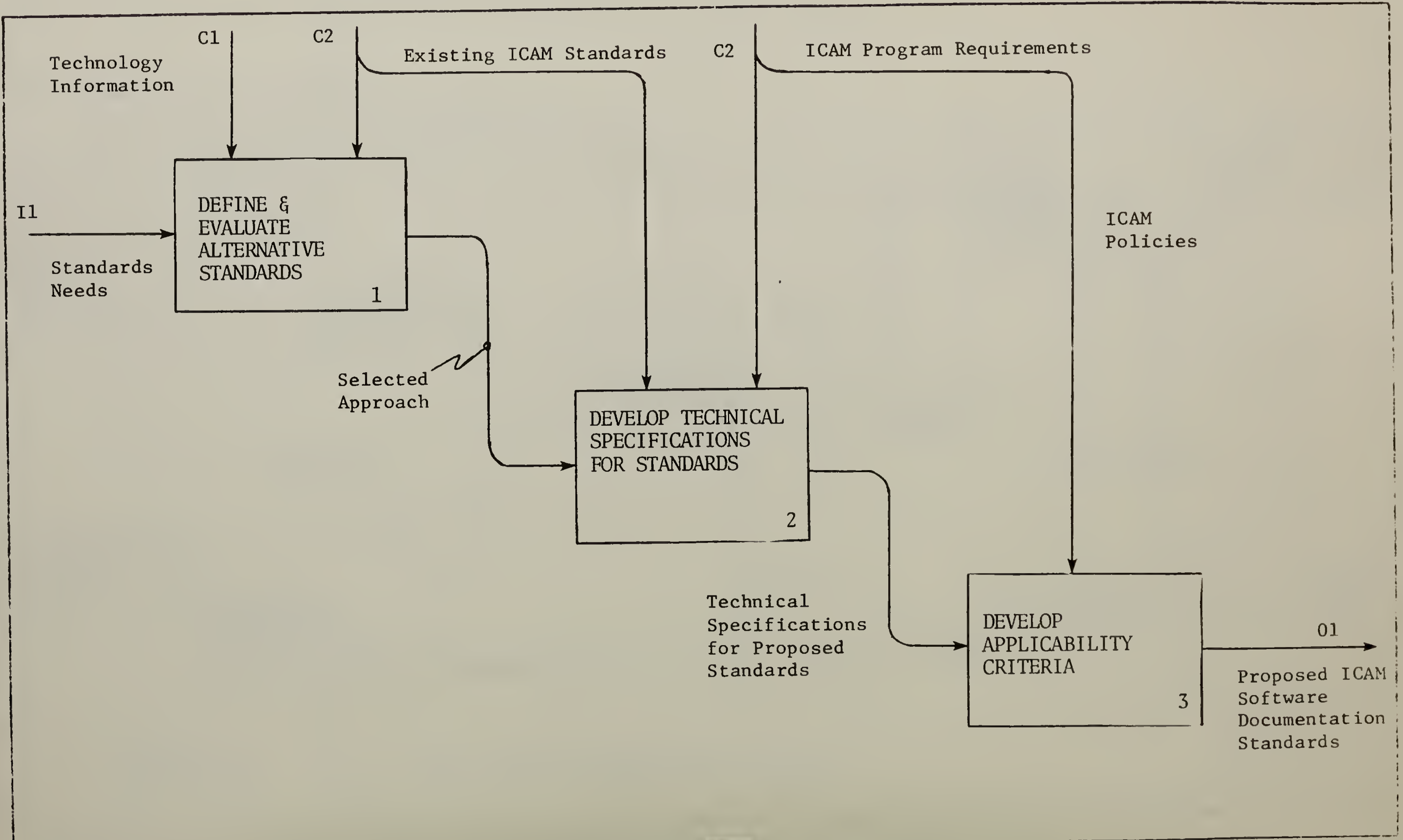
NODE: DOC/A21

TITLE: REVIEW PROJECT DOCUMENTATION


C2
ICAM Software Documentation Standards
ICAM Software Documentation Standards
C1
Needs Information
Standards Needs
02
I1
Problem Identification
EVALUATE SOFTWARE DOCUMENTATION STANDARDS AND COMPLIANCE
1
Non-Standards Problems
EVALUATE RESPONSIVENESS TO USER NEEDS
2
Problem Definition
01
I2
Information on Dissemination
EVALUATE DOCUMENTATION SYSTEM NEEDS
3
01
New Requirements

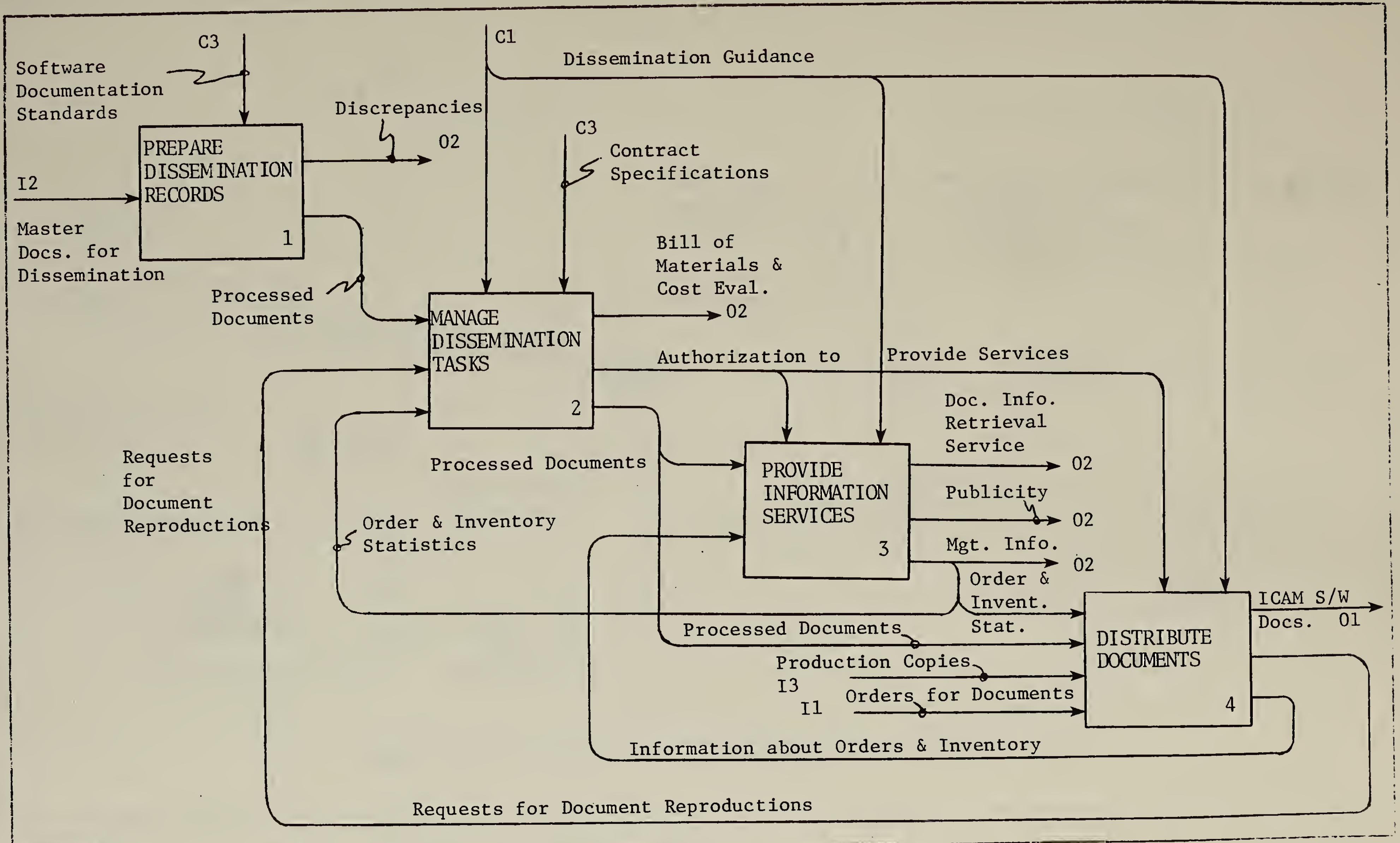
[1] This is a self-evaluation function that continuously assesses whether the Software Documentation system works. In particular, it is triggered by the identification of a problem.

NODE: DOC/A22 | TITLE: EVALUATE SOFTWARE DOCUMENTATION SYSTEM


C1
C2
Existing ICAM Standards
C2
ICAM Program Requirements
Technology Information
I1
Standards Needs
DEFINE & EVALUATE ALTERNATIVE STANDARDS
1
ICAM Policies
Selected Approach
DEVELOP TECHNICAL SPECIFICATIONS FOR STANDARDS
2
Technical Specifications for Proposed Standards
DEVELOP APPLICABILITY CRITERIA
3
O1
Proposed ICAM Software Documentation Standards

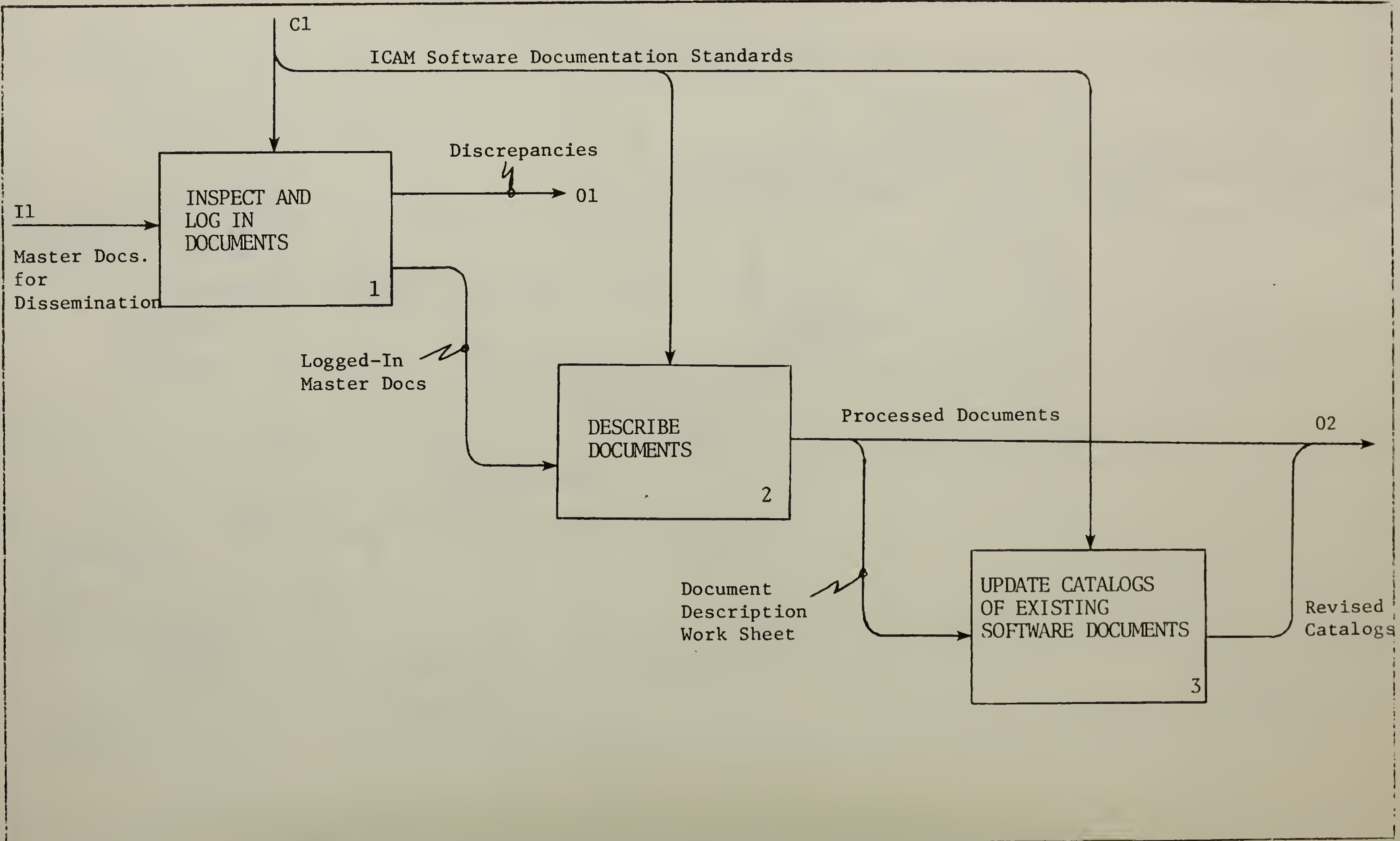
NODE: DOC/A23

TITLE: DEVELOP PROPOSED ICAM SOFTWARE DOCUMENTATION STANDARDS

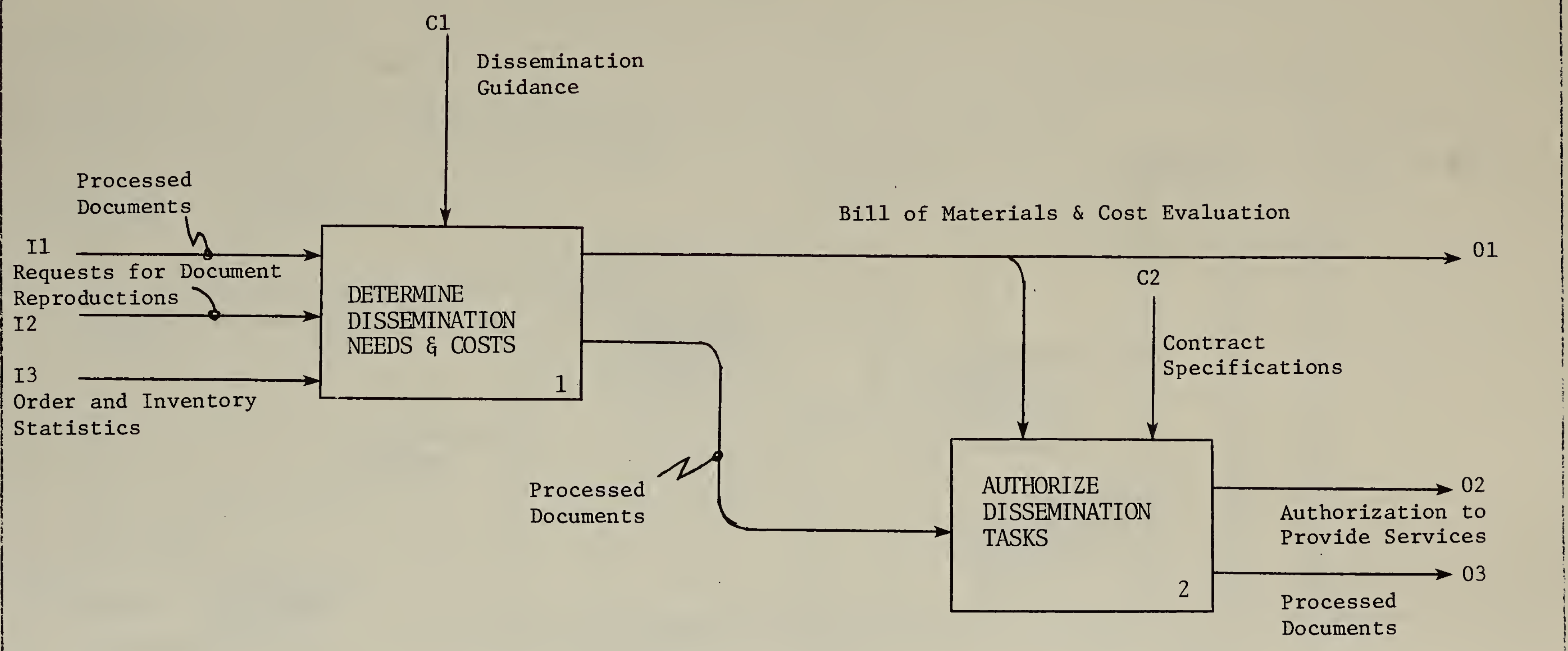

C3
Software Documentation Standards
I2
Master Docs. for Dissemination
PREPARE DISSEMINATION RECORDS
1
Discrepancies
02
Processed Documents
C1
Dissemination Guidance
C3
Contract Specifications
MANAGE DISSEMINATION TASKS
2
Bill of Materials & Cost Eval.
02
Authorization to Provide Services
Requests for Document Reproductions
Processed Documents
Order & Inventory Statistics
PROVIDE INFORMATION SERVICES
3
Doc. Info. Retrieval Service
02
Publicity
02
Mgt. Info.
02
Order & Invent. Stat.
Processed Documents
Production Copies
I3
I1
Orders for Documents
DISTRIBUTE DOCUMENTS
4
ICAM S/W Docs.
01
Information about Orders & Inventory
Requests for Document Reproductions

NODE: DOC/A3

TITLE: DISSEMINATE SOFTWARE DOCUMENTS

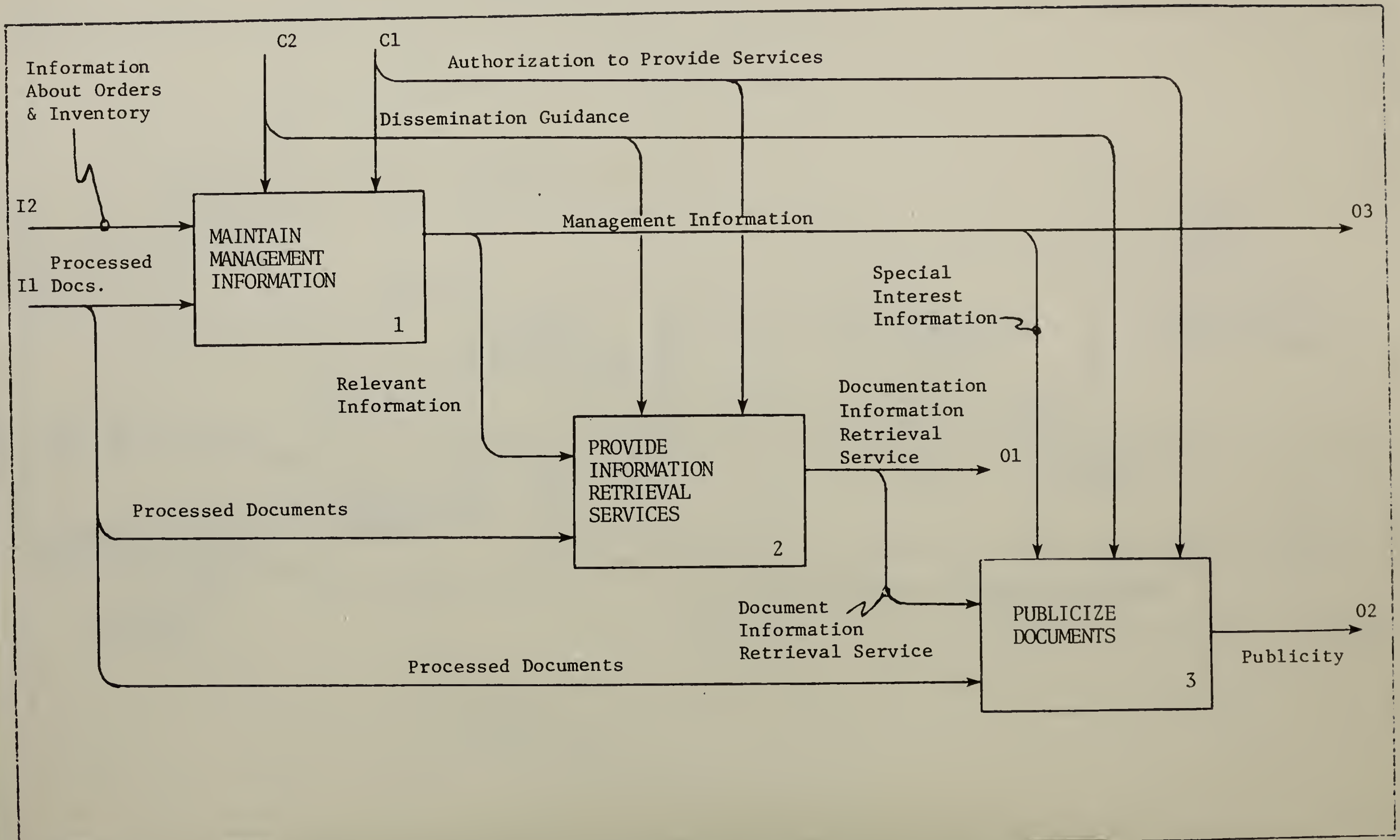

C1
ICAM Software Documentation Standards
I1
Master Docs. for Dissemination
INSPECT AND LOG IN DOCUMENTS
1
Discrepancies
01
Logged-In Master Docs
DESCRIBE DOCUMENTS
2
Processed Documents
02
Document Description Work Sheet
UPDATE CATALOGS OF EXISTING SOFTWARE DOCUMENTS
3
Revised Catalogs

NODE: DOC/A31 TITLE: PREPARE DISSEMINATION RECORDS


C1
Dissemination Guidance
Processed Documents
I1
Requests for Document Reproductions
I2
I3
Order and Inventory Statistics
DETERMINE DISSEMINATION NEEDS & COSTS
1
Bill of Materials & Cost Evaluation
01
C2
Contract Specifications
Processed Documents
AUTHORIZE DISSEMINATION TASKS
2
02
Authorization to Provide Services
03
Processed Documents

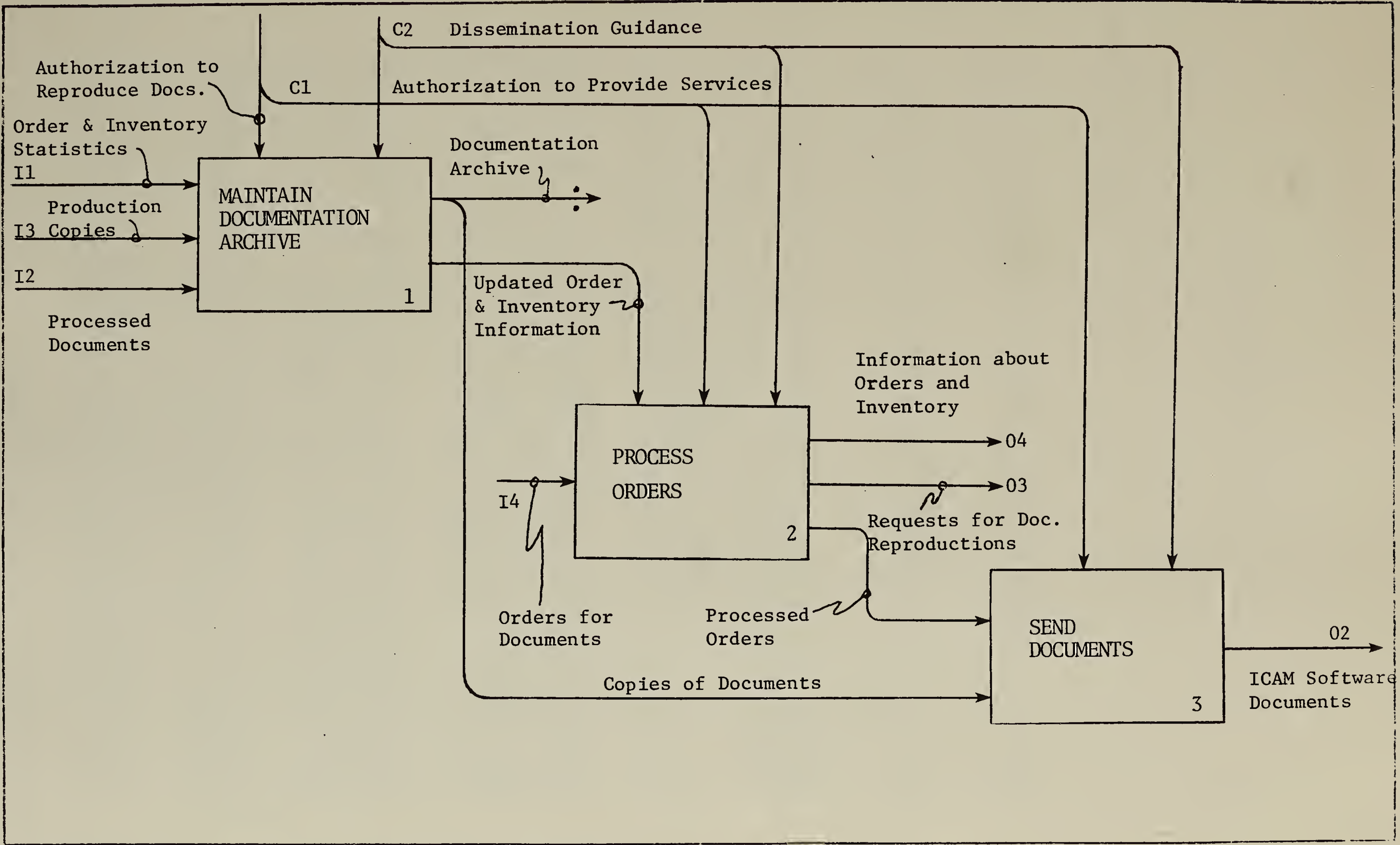
NODE: DOC/A32

TITLE: MANAGE DISSEMINATION TASKS


C2
C1
Authorization to Provide Services
Information About Orders & Inventory
Dissemination Guidance
I2
MAINTAIN MANAGEMENT INFORMATION
1
Management Information
03
Processed I1 Docs.
Special Interest Information
Relevant Information
Documentation Information Retrieval Service
01
PROVIDE INFORMATION RETRIEVAL SERVICES
2
Processed Documents
Document Information Retrieval Service
PUBLICIZE DOCUMENTS
3
02
Publicity
Processed Documents

NODE: DOC/A33

TITLE: PROVIDE INFORMATION SERVICES


C2 Dissemination Guidance
Authorization to Reproduce Docs.
C1
Authorization to Provide Services
Order & Inventory Statistics
I1
Production
I3 Copies
I2
Processed Documents
MAINTAIN DOCUMENTATION ARCHIVE
1
Documentation Archive
Updated Order & Inventory Information
Information about Orders and Inventory
PROCESS ORDERS
2
04
03
I4
Requests for Doc. Reproductions
Orders for Documents
Processed Orders
SEND DOCUMENTS
3
02
ICAM Software Documents
Copies of Documents

NODE: DOC/A34 | TITLE: DISTRIBUTE DOCUMENTS

THIS PAGE DELIBERATELY

LEFT BLANK

GLOSSARY FOR DOC-MODEL

GLOSSARY FOR DIAGRAM A-0

Box [A-0.0] PROVIDE ICAM SOFTWARE DOCUMENTATION: Develop, produce, control, review, and distribute ICAM Software Documentation.

[A-0.0I1] LIFE CYCLE RESULTS OF ICAM PROJECTS: Results such as software products, documents, reports, etc. that are pertinent to the current project, but were produced by other ICAM projects.

[A-0.0C1] TECHNOLOGY INFORMATION: State of the art information in pertinent technology, especially in computer and software technology. This type of information provides guidance on methodology, quality criteria, feasibility of an approach, etc.

[A-0.0C2] ICAM PROGRAM REQUIREMENTS: Standards, plans and policies, and resources constraints, as applied to software development.

STANDARDS: Standards that are already approved, and are applicable to ICAM software development. These standards require technical compliance. They may be Federal standards or guidelines, Air Force or ICAM standards, etc. For example, MIL-STD 490, FIPS 38; DoD Standard 7935.1-S, the ICAM Software Documentation Style Guide and Content Guides, etc.

POLICIES AND PLANS: Plans, organizational policies, and regulations that may govern the software operations of affected organizations. These require general compliance, and may originate from the Federal Government, the Air Force, ICAM organizations, contractor or subcontractor organizations, etc.

RESOURCES CONSTRAINTS: Constraints due to limitations of manpower, money, time, hardware/software facilities, etc. Other resources constraints include budgetary, contractual, or managerial considerations.

[A-0.0C3] NEEDS INFORMATION: Information that indicates approaches and needs for software documentation. Included would be contract specifications, technical articles on documentation, conference results from document users, and special task force reports on user needs.

CONTRACT SPECIFICATIONS: Contract specifications consist of specific requirements which are dictated by the sponsor funding the project. These specifications may state in very general terms the nature, objectives, and evaluation criteria for the project. They may also contain specific requirements for compliance with ICAM policies, standards, and regulations.

USER NEEDS: Project requests, statements of users' requirements, problem statements, etc.

[A-0.001] ICAM SOFTWARE DOCUMENTS (SOFTWARE DOCUMENTATION): Textual information and associated graphics intended for human use, as opposed to those which are solely for computer input and execution. Software documentation is the technical and project information intended eventually for wide dissemination, and specifically oriented toward the development, design, maintenance, and use of ICAM computer programs and systems. Specifically included are: feasibility studies, program specifications, manuals, program listings, and instructional material required to produce and use reliable and transferable software products. Excluded from this class of documents are the budget and program documents, contracts, management correspondence, and basic manufacturing and engineering results.

[A-0.002] PROPOSED ICAM SOFTWARE DOCUMENTATION STANDARDS: Technical specifications for software documentation that are proposed by ICAM organizations after evaluating the effectiveness of the Software Documentation System and identifying problems in responding to users' needs.

[A-0.003] INFORMATION ON DISSEMINATION: Management information on the complete body of ICAM Software Documents that are available for dissemination, including, for example, indexes to available software documents.

[A-0.OM1] ICAM MEANS AND RESOURCES: All resources needed or used to perform a task, including people that do tasks--e.g., developers or ICAM managers--and automated tools--e.g., a FORTRAN analyzer, IDEF, or a report generator.

GLOSSARY FOR AO

BOX [AO.1] DEVELOP SOFTWARE AND SOFTWARE DOCUMENTATION: The emphasis in this stage is on the software documents produced in conjunction with software development. Good software management practices dictate that documentation should be produced during the software development process, and not as an afterthought; this box reflects that philosophy by showing the two processes as integrated.

BOX [AO.2] ADMINISTER SOFTWARE DOCUMENTATION SYSTEM: Performed by the ICAM Program Office, this function encompasses such oversight activities as formally reviewing for compliance with contracts and standards, overseeing and evaluating the effectiveness of the operations of the Software Documentation System, and proposing technical specifications for documentation standards.

BOX [AO.3] DISSEMINATE SOFTWARE DOCUMENTS: This activity includes not only the distribution of software documents, but also the management of information about the software collection.

[AO.1C4] POST-DELIVERY REDIRECTIONS: Feedback from the ICAM Program Office, affecting ICAM Software Documentation. Redirections are usually issued after the Software Documentation has been delivered to the ICAM Program Office and undergone formal review for compliance. Redirections provide guidance to developers for restructuring or modifying noncompliant documents--those that do not conform to technical or contractual requirements.

[AO.101] SOFTWARE PRODUCTS: Actual programs, listings, tapes, associated data files, etc.

[AO.102] MASTER DOCUMENTS FOR REVIEW AND ACCEPTANCE: These documents have been delivered by a contractor or developer for ICAM review and acceptance. After final acceptance, they are placed under configuration control.

[AO.103] PRODUCTION COPIES: These reproductions of master documents are prepared for dissemination.

[AO.203] DISSEMINATION GUIDANCE: Directions for dissemination of ICAM Software Documents and related services. Such guidance might include dissemination lists, schedules, and requirements.

[A0.204] MASTER DOCUMENTS FOR DISSEMINATION: These documents comply with contract specifications and ICAM standards, and are therefore reproduced for dissemination to other program participants and to the public.

[A0.3I1] ORDERS FOR DOCUMENTS: Formal requests for documents from the ICAM user community.

[A0.3M1] AIR FORCE AND CONTRACTORS: These people are responsible for disseminating the software documents. The activity may be performed by a group within ICAM or delegated to a responsible organization such as NTIS.

[A0.2M1] AIR FORCE PROJECT MANAGERS AND USERS: The ICAM Program Office within the Air Force has the primary responsibility for administering the Software Documentation System. It is aided by feedback from the user community.

[A0.1M1] SOFTWARE DEVELOPERS: These people are responsible for developing software and software documentation. They may be either contractors or members of ICAM organizations.

GLOSSARY FOR A1

BOX [A1.1] PRODUCE ICAM SOFTWARE DOCUMENTATION: The actual production of documentation during the software life cycle. Note that the focus is on software documents, not the actual software products.

BOX [A1.2] MANAGE QUALITY AND CONFIGURATION: This is the "formal" internal review cycle performed by the contractor. Technical quality control for documentation is exercised throughout software development. Redirection, in the form of suggested modifications or guidance, is supplied to the developing group before final delivery to ICAM sponsors.

BOX [A1.3] CONFIGURE DOCUMENTS FOR ICAM: The final preparation of documents for delivery to the ICAM Program Office and subsequent dissemination. Preparation focuses on physical organization and control requirements for ICAM.

[A1.1I2] USER NEEDS: See [A-0.0C3]: Needs Information. Included are project requests, statements of users' requirements, problem statements, etc.

[A1.1C5] PRE-DELIVERY REDIRECTIONS: The internal technical guidance issued by the developers/contractors, due to inconsistencies, problems, or inadequacies found in the software documentation processes. Such redirection is given before the documents are submitted for formal review and acceptance by the ICAM Program Office.

[A1.2O2] QUALITY SOFTWARE DOCUMENTS: These documents have undergone thorough inspection as well as quality assurance review, and have been judged not only to comply with technical and contractual requirements, but also to meet the contractor's highest criteria for excellence.

GLOSSARY FOR A11

BOX [A11.1] ANALYZE NEEDS: The developer studies the existing system to determine how new software can solve current problems. This initial stage in the software life cycle involves an assessment of present technology and the formulation of various approaches based on existing technology and on descriptions of comparable systems. After evaluating each of these systems for technical feasibility and economic benefits, the contractor recommends one of them for development.

BOX [A11.2] DEFINE SYSTEM REQUIREMENTS: Using the software solution that was recommended at the conclusion of the analysis stage, the contractor specifies detailed requirements for the functions, data, performance, portability, and modularity of the software. He will also spell out auxiliary software requirements such as utility routines, specific compilers, or a database management system. He will often consult future users of the system, and he may simulate the operating environment in order to "test out" requirements.

BOX [A11.3] DESIGN: Design consists of preliminary design, detailed design, and the planning of these activities. The preliminary design covers the overall system and produces preliminary specifications for system/ subsystems, programs, and data. Feedback from a review process helps refine these documents, which then guide the development of detailed specifications for each aspect of the software system. Using these specifications, the design team prepares a Strategic Test Plan outlining testing strategies and methodology.

BOX [A11.4] PROGRAM AND VERIFY: Following standard programming practices, the developer writes, debugs, and tests the computer programs designed to satisfy ICAM needs. At the same time he begins preparing system documentation, including drafts of the Operations Manual, User's Manual, Program Maintenance Manual, and Installation Guide. When he has tested all program units, he prepares a Detailed Test Plan according to the methodology and strategy outlined in the Strategic Test Plan.

BOX [A11.5] INTEGRATE, TEST, AND VALIDATE: Perform tests, analyze test results, and validate them against system specifications. If tests are unsuccessful, request reprogramming or file a Problem Report. When tests are successful, release Test Analysis Reports on integration testing and system testing, and final drafts of the User's Manual, Operations Manual, Program Maintenance Manual, and Installation Guide.

BOX [A11.6] OPERATE AND MAINTAIN: At this stage the developer works with the user to install, operate, and maintain the developed software. After installing the system, he prepares a Test Analysis Report on acceptance testing. The organization responsible for maintaining the system prepares a Maintenance Plan for the user. After adapting the software to his own environment, the user evaluates the system and prepares an Application Report on his particular applications of the software and on any enhancements he may have developed.

[A11.101] ANALYSIS DOCUMENTS: The analysis of needs produces a Definition Plan, a Feasibility Study, and a Cost/Benefit Analysis. Informal analysis data include Alternative Software Solutions and Technical Feasibility Evaluations.

[A11.201] REQUIREMENTS DOCUMENT: The Requirements Document defines functional, data, performance, and other requirements of the proposed software system.

[A11.301] DESIGN DOCUMENTS: Specific design documents include a Development Plan, a System/Subsystem Specification, a Program Specification, a Data Specification, and a Strategic Test Plan. Informal design data include Problems and a Preliminary Design.

[A11.401] PROGRAMS AND RELATED DOCUMENTS: Actual programs and listings, along with a formal Detailed Test Plan, informal documents noting Problems and Unit Test Results, and Draft Manuals.

[A11.501] SOFTWARE SYSTEM AND DOCUMENTS: Formal test and integration documents include Problem Reports, a System Test Analysis Report, a Software Validation Report, a Software Summary, an Installation Guide, a User's Manual, an Operations Manual, and a Program Maintenance Manual. Informal data include Integration Test Results.

[A11.601] OPERATIONS DOCUMENTS: These documents include a Maintenance Plan, a Test Analysis Report on acceptance testing, an Application Report, and Problem Reports. Informal operations data include Maintenance Requirements and Operation and Usage Experience.

GLOSSARY FOR A111

BOX [A111.1] DEFINE PROJECT SCOPE AND PLAN DEFINITION TASKS: After receiving a statement of user needs, an analysis project team meets to define the scope of the problem. The project team may vary throughout the life cycle of the project development effort, but at this stage it generally involves users as well as developers. This team has the responsibility of determining whether software can solve the problem as stated, whether the organization has the resources to do the job, and whether the problem needs to be subdivided (or subcontracted) for development purposes. The team refines the user's statement of needs before formulating a plan and setting the bounds for the problem under study. It describes the objectives, states the assumptions, and formulates a set of evaluation criteria for further analysis. Of primary interest during this planning stage are the documenting activities. The project team prepares a Definition Plan to cover the activities of needs analysis and requirements definition.

BOX [A111.2] FORMULATE TECHNICAL APPROACHES: Having determined that the problem is solvable, the project team next formulates technical approaches for satisfying the objectives of the software development project. It assesses current technology in the relevant field, noting in particular any descriptions of comparable systems. The team then documents possible technical approaches, including in its report the applicable software assessments.

BOX [A111.3] EVALUATE PERFORMANCE AND FEASIBILITY: The project team evaluates each of the alternative software solutions to determine its technical feasibility--i.e., its capability of meeting user requirements with available technology and methods of operation. In addition, the team determines the performance and operational feasibility of each solution--i.e., its ability to fit the operational pattern and resources of the organization. In some cases, the team may use simulation, or modeling, to "exercise" each software solution, and then use the simulation reports in evaluating performance and operational feasibility.

BOX [A111.4] ASSESS COSTS AND BENEFITS: For each of the alternative software solutions, the developer must evaluate not only recurring and nonrecurring costs, but also quantifiable and intangible benefits.

BOX [A111.5] <u>PREPARE PROJECT RECOMMENDATION</u>: After the project team evaluates the technical feasibility and benefits of each of the alternative software solutions, it compares their risks, uncertainties, and sensitivity to changes. It then prepares a Feasibility Study recommending one software solution to the development team and to management.

[A111.1C1] <u>ICAM STANDARDS</u>: Applicable ICAM Standards include general technical guidelines as well as the ICAM Software Documentation Style Guide and the ICAM Software Documentation Content Guides for the Definition Plan, Feasibility Study, and Cost/Benefit Analysis.

[A111.101] <u>DEFINITION PLAN</u>: The Definition Plan covers all the activities of needs analysis and requirements definition. It tells whether these activities will involve automated tools or formal reports and other documentation. It identifies resources, organizations, methodology, and standards for specific definition tasks, and it schedules milestones, reviews, and delivery dates for major ICAM Software Documents.

[A111.2C3] <u>COMPARABLE SYSTEMS INFORMATION</u>: Information about systems which resemble alternative software solutions, but which may have been used in a different environment or designed for different purposes.

[A111.201] <u>ALTERNATIVE SOFTWARE SOLUTIONS</u>: Alternative structures and methods for satisfying the identified objectives of the system development project.

[A111.301] <u>TECHNICAL FEASIBILITY EVALUATIONS</u>: This technical part of the Feasibility Study evaluates the abilities of the alternative software solutions to satisfy user requirements with available technology and methods of operation.

[A111.401] <u>COST/BENEFIT ANALYSIS</u>: The Cost/Benefit Analysis gives managers, designers, and users detailed estimates of projected costs and benefits of alternative pieces of software. It analyses both recurring and nonrecurring costs, and tangible and intangible benefits.

[A111.501] <u>FEASIBILITY STUDY</u>: The Feasibility Study summarizes the findings of the analysis activity. It discusses the goals and needs of an organization and compares possible ways to reach those goals. It records the technical feasibility evaluations of alternative software solutions and recommends a particular software system as the best to meet the technical and economic requirements of the software development project.

[A111.502] <u>COST/BENEFIT ANALYSIS</u>: The formal document presenting the information identified in [A111.401]: Cost/Benefit Analysis.

GLOSSARY FOR A112

BOX [A112.1] SPECIFY FUNCTIONAL REQUIREMENTS: Identify and define the system functions that are required of the proposed software. These requirements may indicate the intended operating environment and the user activities to be supported.

BOX [A112.2] SPECIFY DATA REQUIREMENTS: Identify relevant data items or aggregates, and describe the necessary technical characteristics for data collection and storage.

BOX [A112.3] SPECIFY PERFORMANCE AND OTHER REQUIREMENTS: Define such performance characteristics as timing for job turnaround, response time in an interactive environment, and requirements for accuracy, validation, editing, and security. These requirements should reflect results of any simulations performed during this stage.

BOX [A112.4] SPECIFY DESIGN CONSTRAINTS: After defining the system requirements, the project team specifies any fundamental design standards, and identifies and documents any constraints on the design process. For example, some mathematical software to be used might require a particular FORTRAN compiler.

BOX [A112.5] INTEGRATE AND EVALUATE RESULTS: Using the information produced during requirements analysis, refine cost estimates for the project and combine the information about requirements into the Requirements Document.

[A112.1C1] ICAM STANDARDS: These standards include general guidelines for requirements analysis as well as the ICAM Software Documentation Style Guide and the ICAM Software Documentation Content Guide for the Requirements Document.

[A112.1C2] DEFINITION PLAN: See [A111.101].

[A112.101] FUNCTIONAL REQUIREMENTS: The Functional Requirements initially define the software, including its requirements and operating environment. They compare existing and proposed methods of performing the required functions, and they identify projected improvements, possible impacts, and specific functions of proposed software products.

[A112.201] DATA REQUIREMENTS: The Data Requirements describe the data in technical terms. They identify static and dynamic data elements and tell what kind of information the user and the developer collect to characterize specific data elements.

[A112.301] PERFORMANCE AND OTHER REQUIREMENTS: A description of such required performance characteristics as timing, accuracy, security, validation, flexibility, and response to contingencies. When applicable, Performance and Other Requirements should include a simulation or analysis report.

[A112.401] DESIGN CONSTRAINTS: Constraints on software design, perhaps including language specification, basic and special design standards, and interfaces with the existing system.

[A112.501] REQUIREMENTS DOCUMENT: The Requirements Document includes the functional, data, performance, and other requirements of the proposed software system. For individual descriptions of the separate components, see [A112.101], [A112.201], and [A112.301].

GLOSSARY FOR A113

BOX [A113.1] PLAN DEVELOPMENT: Plan the development of the software project from preliminary design through acceptance testing.

BOX [A113.2] PERFORM PRELIMINARY DESIGN: The developer produces and documents an overall preliminary design of the software system, emphasizing general specifications of the system's internal construction. The project team, including intended users, validates the requirements documents, and the designers use feedback from this review cycle to sketch out software modules and components.

BOX [A113.3] PREPARE SYSTEM/SUBSYSTEM SPECIFICATION: The design team analyzes the performance trade-offs of various algorithms and prepares detailed structural and functional specifications for each system/subsystem. After defining the internal architecture of the software, the team documents the system/subsystem specification, which provides a frame of reference for the remainder of the design.

BOX [A113.4] PREPARE PROGRAM SPECIFICATION: Using the system/subsystem specification, the designers specify and document individual program modules and their interactions.

BOX [A113.5] PREPARE DATA SPECIFICATION: Define the logical and physical characteristics of the data. This definition may identify sources, methods of collection, and storage properties.

BOX [A113.6] EVALUATE DESIGN AND PLAN TESTING: Review the final design and prepare a Strategic Test Plan outlining strategies and methodology for testing the software system.

[A113.1C1] ICAM STANDARDS: These standards include general guidelines for software design as well as the ICAM Software Documentation Style Guide and the ICAM Software Documentation Content Guides for the Development Plan, the System/Subsystem Specification, the Program Specification, the Data Specification, and the Strategic Test Plan.

[A113.101] DEVELOPMENT PLAN: The Development Plan specifies the organizational units responsible for particular tasks and identifies products and milestones. It allocates necessary resources and personnel for each task, describes the methodology for development and quality assurance, and schedules reviews and delivery dates for software products and documents.

[A113.201] PROBLEMS: Problems detected during preliminary design. The project team reports them to the project's quality control personnel, who either resolve the problem or file a Problem Report. Problems can also be reported from any other design activity.

[A113.202] PRELIMINARY DESIGN: A general outline for the System/Subsystem Specification, the Program Specification, and the Data Specification.

[A113.301] SYSTEM/SUBSYSTEM SPECIFICATION: This document gives detailed information about the operating environment, design characteristics, system and subsystem interfaces and relationships, logic, dynamics, support software requirements, and control and data flow of proposed software.

[A113.401] PROGRAM SPECIFICATION: Specifically directed toward programmers, this document gives detailed characteristics of the proposed software: its functions, its operating environment, its logic and flow, the relationships among its modules, and the design features of each module.

[A113.501] DATA SPECIFICATION: This document states the standards, conventions, and quality requirements of the proposed data, and specifies its logical and physical characteristics--including methods of creation, collection, storage, access, maintenance, updating, and protection. The document also specifies the software to perform these functions.

[A113.601] STRATEGIC TEST PLAN: This document outlines the concepts, criteria or standards, tools, strategies, and methodology for testing the software system. It specifies what part of testing will be simulation and what part actual operation of the software. It identifies broad tests (like unit testing), giving their purposes and specifying what documentation they require.

GLOSSARY FOR A114

BOX [A114.1] CODE, DEBUG, AND UNIT TEST: Using standard programming languages, the programmers write executable code to satisfy the program specifications produced in the design stage. Debugging involves finding and correcting errors introduced during the coding process. When debugging is finished, the programmers test individual program modules to ensure that they are logically correct for the particular testing environment. This unit testing, which the programmer documents in the Unit Test Results, verifies that the module has no significant problem or logical error to impede its use. Errors discovered in testing may feed back to the coding, or even to the design stage. Programmers should take care to document carefully each of these three programming stages.

BOX [A114.2] PREPARE MANUALS: Using the newly tested code for individual program modules, write drafts of manuals for installation, operation, maintenance, and usage of the system. Revise the manuals each time a change in code affects their usefulness.

BOX [A114.3] REVIEW VERIFIED PROGRAMS AND COMPLETE TESTING PLAN: Review programs for correctness and conformance to programming standards in the ICAM Software Documentation Style Guide. Prepare a Detailed Test Plan.

[A114.1C1] ICAM STANDARDS: General guidelines for programming as well as the ICAM Software Documentation Style Guide and the ICAM Software Documentation Content Guides for the Unit Test Analysis Report, the User's Manual, the Operations Manual, the Program Maintenance Manual, the Installation Guide, and the Detailed Test Plan.

[A114.1C2] DEVELOPMENT PLAN: See [A113.1O1].

[A114.1O1] PROBLEMS: Problems detected during coding, debugging, or unit testing. These informal reports go to the project's quality control personnel, who either resolve the problem or file a formal Problem Report. Other activities during programming and verification can also report Problems.

[A114.1O2] UNIT TEST RESULTS: This informal document records the results of each unit test. It analyzes test results and presents demonstrated capabilities and deficiencies for review.

[A114.103] UNIT TESTED PROGRAMS: Programs (along with their listings) that have "passed" unit tests, and are considered logically correct within that testing environment.

[A114.201] DRAFT MANUALS: First drafts of a User's Manual, an Operations Manual, a Program Maintenance Manual, and an Installation Guide. Programmers will continue to update all this documentation until the software system is validated.

[A114.301] DETAILED TEST PLAN: The Detailed Test Plan reviews previous testing history, identifies particular tests, and specifies test locations and environments. It states the functions that need to be tested and describes appropriate test cases, including criteria for evaluating the thoroughness of the testing as well as the performance of the software. It matches tests to requirements, specifications, and functions of the software system. It gives a detailed schedule of events, including test dates, test sites, and any necessary special arrangements (e.g., freeing up a block of computing time). It describes the tools needed for testing, and specifies how to compile test data.

[A114.302] VERIFIED PROGRAMS: Unit tested programs that have gone through a final review for correctness.

GLOSSARY FOR A115

BOX [A115.1] PERFORM INTEGRATION TESTING: Integrate individual programs into the specified overall system and test them to make sure that each of them still works well when linked with other programs into an integrated software system.

BOX [A115.2] PERFORM SYSTEM TESTING: Test the integrated system to make sure that as a whole it will perform properly in the specified operating environment.

BOX [A115.3] PERFORM FINAL VALIDATION: Analyze system testing results in terms of design specifications. Note minor problems and errors for correction by the programming team. Major problems with the original design or functional specifications may require a Problem Report.

BOX [A115.4] PREPARE SYSTEM FOR DELIVERY: Physically prepare the system for delivery to the user. When appropriate, prepare tapes and disk packs, set up files in the proper sequence, and assemble system components for delivery.

[A115.1C1] ICAM STANDARDS: General guidelines for testing as well as the ICAM Software Documentation Style Guide and the ICAM Software Documentation Content Guides for the Test Analysis Report, the Problem Report, the Software Summary, the User's Manual, the Operations Manual, the Program Maintenance Manual, and the Installation Guide.

[A115.1C2] DETAILED TEST PLAN: See [A114.301].

[A115.101] PROBLEM REPORT: The Problem Report records any anomalies in the testing of the system. Depending on the seriousness of the problem, the report may feed back to any of the four preceding stages--needs analysis, requirements definition, design, or programming. Any testing or validation activity may result in a Problem Report.

[A115.102] INTEGRATION TEST RESULTS: This report documents how well the individual program modules perform after they have been integrated into a system.

[A115.103] INTEGRATED SYSTEM: The integrated system includes all the program modules, each of which has passed integration testing.

[A115.201] SYSTEM TEST ANALYSIS REPORT: The System Test Analysis Report records the performance of the system as a whole during its test runs. It describes the tests, inputs, outputs, and performance of the software system in the testing environment, and it compares this performance to the expected performance of the software in the actual operating environment.

[A115.202] TESTED SYSTEM: The software system (including its listings) that has passed system tests and is ready to be packaged for delivery to the user or the ICAM Program Office.

[A115.3I2] LISTINGS: The code for the tested software system.

[A115.3C3] REQUIREMENTS AND SPECIFICATIONS: Relevant ICAM standards and all previous project documents that are necessary for final validation. Included are the ICAM Software Documentation Style Guide and Content Guides, the Feasibility Study, the Requirements Document, all the specifications, and both test plans.

[A115.301] SOFTWARE VALIDATION REPORT: When the developer has successfully completed the system testing of his software, he prepares the Software Validation Report to certify that the system functions reliably and meets both its specifications and the requirements of the user environment.

[A115.401] INSTALLATION GUIDE AND USER'S, OPERATIONS, AND PROGRAM MAINTENANCE MANUALS: Although the programmers draft these manuals during programming, they do not release them until after final validation. At this point the texts of the manuals are final, but the documents themselves must still be physically prepared before delivery to the ICAM Program Office. The Installation Guide specifies how, and under what conditions, the software system may be installed. For example, it may specify that two tape drives are necessary to load all the modules in, or that it expects certain commands from the console. The Operations Manual gives directions for operating the system in its specified environment. The Program Maintenance Manual describes for the maintenance programmer the functions of the software system, its operating environment, and any necessary maintenance procedures such as programming conventions, verification procedures, and error correction procedures. The User's Manual describes for users the functions performed by the software. It should contain information on how to use the software, and it should serve as a reference document for preparing input data and parameters, and for interpreting results.

[A115.402] SOFTWARE SYSTEM: This approved and assembled software package, including all related documentation, is physically ready for delivery.

[A115.403] SOFTWARE SUMMARY: A brief description of the software system, giving its purpose, developer, and history. The description goes on the form included in FIPS 30, "Software Summary."

GLOSSARY FOR A116

BOX [A116.1] PLAN OPERATION AND MAINTENANCE: Plan routine operation and prepare a Maintenance Plan to cover all aspects of maintenance and modification of the new software system.

BOX [A116.2] INSTALL SYSTEM: Load the software system into the actual operating environment. Perform acceptance testing, and train users in the operation of the system.

BOX [A116.3] USE SYSTEM: Operate the system that has been successfully installed; exercise it at first through a shakedown period, with representative workload, in the actual environment; after the "installation bugs" are found, integrate the system into the operational environment in production mode.

BOX [A116.4] EVALUATE SYSTEM: The user periodically evaluates operating software systems to search for problems and to determine whether maintenance is satisfactory, whether modifications might improve the system, or whether new developments in hardware or software technology have made the current software system obsolete.

BOX [A116.5] MAINTAIN AND MODIFY: Maintain the system and make modifications to meet changing operational requirements.

[A116.101] MAINTENANCE PLAN: Besides scheduling routine maintenance, the Maintenance Plan includes a mechanism for evaluating the system and reporting faults or problems found after the software has been installed.

[A116.2C3] INSTALLATION GUIDE: See [A115.401]: Installation Guide and User's, Operations, and Program Maintenance Manuals.

[A116.201] ACCEPTANCE TEST ANALYSIS REPORT: This report documents the installation of the system, giving the results of the acceptance testing and noting any unusual occurrences. It reports what tests were run, what inputs were used, and what outputs were produced, and it evaluates the general capabilities of the installed software in the given operating environment.

[A116.202] PROBLEM REPORT: A formal report of problems encountered during installation of the software system.

[A116.203] OPERATIONAL SYSTEM AND DOCUMENTS: A successfully installed software system, together with the documentation necessary to use it.

[A116.3C2] SOFTWARE CHANGE NOTICE: This notice of system changes tells when the system was modified and notes required changes in existing documentation.

[A116.302] OPERATION AND USAGE EXPERIENCE: Information about the actual operation of the system.

[A116.401] APPLICATION REPORT: The user's report that evaluates and describes his experiences with the software system. It tells how the software has affected his business and what training his personnel received. It also describes adjustments or modifications, unusual applications, or enhancements developed by the user.

[A116.402] MAINTENANCE REQUIREMENTS: Either routine or special problems with the software system that require maintenance or modification.

[A116.5C3] PROGRAM MAINTENANCE MANUAL: See [A115.501]: Installation Guide and User's, Operations, and Program Maintenance Manuals.

GLOSSARY FOR A12

BOX [A12.1] DETERMINE CONFORMANCE TO TECHNICAL AND CONTRACTUAL SPECIFICATIONS: The contractor closely examines his own product to make sure that it conforms to technical and contractual specifications. He reviews Problem Reports from the development team and decides whether to propose software changes through a Software Change Proposal, or to order the development team to try again to meet the contract specifications. Approved documents go on to the next stage of internal review. Noncompliant documents return to previous stages with redirections.

BOX [A12.2] PREPARE CHANGE PROPOSALS: Working with formal Problem Reports, the developer locates the cause of the problem and prepares proposals for changing software requirements or specifications.

BOX [A12.3] MANAGE CONFIGURATIONS: After receiving a Software Change Proposal, the configuration manager negotiates with the developer to come up with new contract specifications.

BOX [A12.4] REVIEW FOR CONFORMANCE TO ICAM STANDARDS: At this stage the contractor determines whether or not his software documentation conforms to relevant ICAM standards.

BOX [A12.5] DETERMINE ADEQUACY AND USABILITY OF DOCUMENTS: As a final stage in the internal quality assurance cycle, the developer evaluates the documents for clarity, intelligibility, and modularity.

[A12.101] PROBLEM REPORTS: Formal reports of problems detected during development. These reports go both to the ICAM PO (like other ICAM Software Documents) and to the developer's personnel who propose changes to requirements or specifications.

[A12.103] and [A12.402] APPROVED SOFTWARE DOCUMENTS: These documents have passed the internal review specified by the boxes from which they emerge. The developer releases them to the ICAM Program Office only after they have passed all inspections.

[A12.201] SOFTWARE CHANGE PROPOSAL: A proposal by the contractor for changes in requirements, specifications, documentation, or actual operating characteristics of the software system. It specifies both existing and proposed versions of the requirements or specifications in question.

[A12.301] SOFTWARE CHANGE NOTICE: The Software Change Notice officially changes requirements, specifications, documentation, or actual operating characteristics of the software being developed or run. Its format is suitable for it to be appended to existing contract specifications, and if produced during development rather than operation, it becomes part of pre-delivery redirections.

[A12.5C1] ADDITIONAL QUALITY CONSIDERATIONS: Factors--like readability, attractiveness, organizational logic, and consistency of presentation--that affect the quality of ICAM Software Documents.

[A12.3M1] ICAM PROGRAM OFFICE AND CONTRACTOR: The ICAM contractor and the ICAM Program Officers in charge of configuration management.

GLOSSARY FOR A13

BOX [A13.1] REGISTER DOCUMENTS FOR DISTRIBUTION CONTROL: This is the developer's process for controlling documents. It involves assigning document identification codes as well as dating, classifying, and logging in the document. It also includes preparing an index to accompany the software document when the developer releases it to the ICAM Program Office.

BOX [A13.2] PRODUCE DELIVERABLE DOCUMENTS: Format, record, print, and bind ICAM Software Documents.

BOX [A13.3] INSPECT DOCUMENTS AND AUTHORIZE DELIVERY: Check document copies for agreement with approved texts and formats before authorizing delivery to the ICAM Program Office.

[A13.101] BIBLIOGRAPHIC DATA SHEET: A form (like DD 1473) which lists author(s), title, sponsoring and producing agencies, general bibliographic information, and a control number.

[A13.102] REGISTERED DOCUMENTS: Documents which have been categorized for distribution control.

[A13.2C2] PRODUCTION REDIRECTIONS: When the person who inspects document copies finds differences between them and the approved original text, he returns them with production redirections--specific statements of changes needed before delivery to the ICAM Program Office.

[A13.201] DOCUMENT COPIES: Master documents or production copies to be inspected before delivery to the ICAM Program Office.

GLOSSARY FOR A2

BOX [A2.1] REVIEW PROJECT DOCUMENTATION: The formal review performed by the ICAM Program Office. Primary interest is in compliance with contractual requirements and with ICAM standards and quality objectives. Compliant documents are transmitted for dissemination, while noncompliant documents are returned to the contractors with redirection, in the form of required modifications and improvements.

BOX [A2.2] EVALUATE SOFTWARE DOCUMENTATION SYSTEM: This is a continuous process conducted by the ICAM Program Office; it is especially important when a problem with the Documentation System has been identified. As a result of this evaluation, plans and schedules are prepared for documentation and dissemination of the software documents . Possible output of this stage includes the plans and schedules that go into ICAM program planning for major innovations and improvements in the Documentation System.

BOX [A2.3] DEVELOP PROPOSED ICAM SOFTWARE DOCUMENTATION STANDARDS: A problem in the Software Documentation System may lead to the identification of specific needs for standards. Technical specifications and applicability criteria for a proposed standard are developed and submitted for processing to the standard-making authority within ICAM.

[A2.102] NEW REQUIREMENTS: While monitoring a contract, the ICAM Program Office may find it necessary to modify contract specifications on a given project, or write new requirements. This may result from a variety of circumstances, ranging from redefinition of ICAM needs to changes in policies.

[A2.2I1] PROBLEM IDENTIFICATION: When ICAM Program Officers find documentation to be noncompliant, they return it to the contractors for modification and improvement. The office further examines noncompliant documents for possible problem areas affecting the operation of the Software Documentation System. The officers then use these identified problems to evaluate the overall Documentation System.

[A2.201] INPUT TO ICAM PROGRAM PLANNING: Information from the evaluation of the Software Documentation System that is pertinent to overall ICAM program planning. This kind of information is outside the scope of the Software Documentation System.

[A2.202] STANDARDS NEEDS: Identification of specific needs for documentation standards; alternatively, a statement of inadequacy in existing standards.

GLOSSARY FOR A21

BOX [A21.1] REVIEW FOR COMPLIANCE WITH TECHNICAL AND CONTRACTUAL SPECIFICATIONS: This function of the ICAM Program Office is akin to the compliance review that the contractor performs prior to delivery. (See A12.1) In formally reviewing the software documents, the PO must determine whether these documents comply with the contractual and technical requirements. If they do, then the review process continues. If they do not, then the documents go back to the contractor with redirections. In some cases, the PO may issue contract clarifications along with the post-delivery redirections.

BOX [A21.2] REVIEW FOR CONFORMANCE TO ICAM STANDARDS: The ICAM Program Office formally ensures that delivered documents conform to ICAM standards. This process is akin to the one performed by the contractor (See A12.3).

BOX [A21.3] DETERMINE USE AND DISTRIBUTION OF DOCUMENTS: In a process analogous to the one performed by the contractor (See A12.4), the ICAM PO evaluates the software documentation for clarity, traceability, portability, and modularity. After considering the potential audience for the software, the Program Office determines dissemination needs and provides dissemination guidance before releasing approved documents for dissemination.

[A21.103] and [A21.203] COMPLIANT DOCUMENTS: Documents that have gone through the evaluation process and have been certified compliant with technical, contractual, and standards requirements.

GLOSSARY FOR A22

BOX [A22.1] EVALUATE SOFTWARE DOCUMENTATION STANDARDS AND COMPLIANCE: Analyze problems with the Documentation System to see if they result from the standards themselves, from questions of compliance, or from other matters like administration or accounting.

BOX [A22.2] EVALUATE RESPONSIVENESS TO USER NEEDS: After reviewing documentation standards, the ICAM Program Office determines how well the Software Documentation System responds to user needs.

BOX [A22.3] EVALUATE DOCUMENTATION SYSTEM NEEDS: The review and evaluation of the Software Documentation System takes place continually. However, if a problem occurs in the operation of this documentation system, then the ICAM Program Office thoroughly reviews and evaluates the management, operation, and processes of the system.

[A22.102] NON-STANDARDS PROBLEMS: Problems with the Documentation System that do not require changes in standards.

[A22.201] PROBLEM DEFINITION: Specifically identified problems in the way the Documentation System responds to users' needs.

[A22.301] NEW DOCUMENTATION SYSTEM REQUIREMENTS: Needed improvements or additions to the existing Documentation System.

GLOSSARY FOR A23

BOX [A23.1] DEFINE AND EVALUATE ALTERNATIVE STANDARDS: Using identified standards needs, the ICAM Program Office evaluates existing ICAM standards for technical deficiencies, examines alternative ways to correct the particular standards problem, and selects one approach as the best.

BOX [A23.2] DEVELOP TECHNICAL SPECIFICATIONS FOR STANDARDS: ICAM standards developers either write technical specifications for new standards or revise the technical specifications of existing standards to remove identified deficiencies.

BOX [A23.3] DEVELOP APPLICABILITY CRITERIA: Develop criteria for applying the new technical standards. Add these criteria to the technical specifications to produce proposed ICAM Software Documentation Standards.

[A23.1C2] EXISTING ICAM STANDARDS: Standards that already exist in the ICAM environment and that apply to software documentation. [See also definition of Standards under A-0.0C2, ICAM Program Requirements].

[A23.101] SELECTED APPROACH: The chosen approach for developing new standards to meet existing standards needs.

[A23.201] TECHNICAL SPECIFICATIONS FOR PROPOSED STANDARDS: Technical specifications that modify existing standards or constitute new standards.

GLOSSARY FOR A3

BOX [A3.1] PREPARE DISSEMINATION RECORDS: Preparing auxiliary documents to control dissemination of approved master documents.

BOX [A3.2] MANAGE DISSEMINATION TASKS: Determine dissemination requirements, allocate resources, keep track of orders and inventory, and authorize dissemination of services.

BOX [A3.3] PROVIDE INFORMATION SERVICES: Make available such services as an online document retrieval service, or an information retrieval service.

BOX [A3.4] DISTRIBUTE DOCUMENTS: Included are the physical storage and distribution of documents.

[A3.101] DISCREPANCIES: Differences between the intended and actual documents, including missing parts and internal inconsistencies.

[A3.102] PROCESSED DOCUMENTS: Master documents together with their indexes, catalogs, abstracts, and other secondary documents.

[A3.2I2] REQUESTS FOR DOCUMENT REPRODUCTIONS: Requests for more copies of ICAM Software Documents to replenish existing inventories.

[A3.2I3] ORDER AND INVENTORY STATISTICS: Statistics calculated from information about existing inventory and incoming orders.

[A3.201] BILL OF MATERIALS AND COST EVALUATION: A breakdown of the physical elements comprising each document. For a report it would include paper size and type (i.e., white bond, red construction, glossy print, etc.). Coupled with an inventory, this breakdown makes it possible to evaluate and predict costs, to determine when supplies should be ordered, and to estimate the associated delays.

[A3.202] AUTHORIZATION TO PROVIDE SERVICES: Authority to run an information retrieval service and to distribute documents.

[A3.3I2] INFORMATION ABOUT ORDERS AND INVENTORY: Raw data on orders and inventory that can be processed into usable statistics about the Documentation System.

[A3.301] DOCUMENT INFORMATION RETRIEVAL SERVICE: A service that provides abstracts, indexes and other information enabling users to get access to ICAM Software Documents or products.

[A3.302] PUBLICITY: Catalogs, announcements, newsletters, and other documents that inform potential users of available ICAM software and documents.

[A3.303] MANAGEMENT INFORMATION: Management-oriented charts and reports that summarize a particular operation or function--e.g., the number of documents in production or storage. Examples of this kind of report are summaries, statistical reports, productivity reports, cost evaluations, and user and document profiles. This type of information is required for the effective evaluation and planning of documentation activities.

GLOSSARY FOR A31

BOX [A31.1] INSPECT AND LOG IN DOCUMENTS: Check the master document for compliance with ICAM documentation standards and assign it an accession number.

BOX [A31.2] DESCRIBE DOCUMENTS: Index the documents by subject code and key words.

BOX [A31.3] UPDATE CATALOGS OF EXISTING SOFTWARE DOCUMENTS: Use information about a new software document to update a catalog of existing software documents.

[A31.102] LOGGED-IN MASTER DOCUMENTS: Master documents which have acquired a control number and are ready for descriptive indexing.

[A31.201] PROCESSED DOCUMENTS: Primary and secondary documents for ICAM software.

[A31.3I1] DOCUMENT DESCRIPTION WORK SHEET: The secondary documents which describe and control ICAM Software Documents.

[A31.301] REVISED CATALOGS: Catalogs which have been revised to include new ICAM Software Documents.

GLOSSARY FOR A32

BOX [A32.1] DETERMINE DISSEMINATION NEEDS AND COSTS: Using information about composition and usage of documents, produce a bill of materials and a cost evaluation for dissemination.

BOX [A32.2] AUTHORIZE DISSEMINATION TASKS: Tell the dissemination personnel when and how to provide information services and documents.

GLOSSARY FOR A33

BOX [A33.1] MAINTAIN MANAGEMENT INFORMATION: Keep order and inventory statistics, information about the documentation archive, and information about special interest users. Information on new software documents will be used to update existing management information.

BOX [A33.2] PROVIDE INFORMATION RETRIEVAL SERVICES: Store and give out information about ICAM Software Documents, including indexes, abstracts, catalogs, and general bibliographic information.

BOX [A33.3] PUBLICIZE DOCUMENTS: Prepare announcements, catalogs, and newsletters to inform potential users about available ICAM Software Documents.

[A33.2I1] RELEVANT INFORMATION: Management information that may improve information retrieval services.

[A33.3C1] SPECIAL INTEREST INFORMATION: Management information about present and prospective users of ICAM Software Products and Documents.

GLOSSARY FOR A34

BOX [A34.1] MAINTAIN DOCUMENTATION ARCHIVE: Operate a library of master documents and production copies of all ICAM software. Keep track of inventories of each document, and release documents when needed for dissemination.

BOX [A34.2] PROCESS ORDERS: Handle incoming orders from ICAM users. Record orders, update order and inventory information, and release orders to be filled.

BOX [A34.3] SEND DOCUMENTS: Fill all processed orders with copies of documents from the documentation archive.

[A34.101] DOCUMENTATION ARCHIVE: The library of master documents and production copies of all ICAM software.

[A34.102] UPDATED ORDER AND INVENTORY INFORMATION: These updates of order and inventory statistics take into account any discrepancies between incoming statistics and existing inventory.

[A34.103] COPIES OF DOCUMENTS: Copies of ICAM Software Documents to be distributed to customers.

NBS-114A (REV. 9-78)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NO. NBSIR 79 - 1940 (R)	2. Gov't. Accession No.	3. Recipient's Accession No.
4. TITLE AND SUBTITLE ICAM Software Documentation Standards			5. Publication Date
			6. Performing Organization Code
7. AUTHOR(S) Center for Programming Science & Technology, Institute for Computer Sciences and Technology			8. Performing Organ. Report No.
9. PERFORMING ORGANIZATION NAME AND ADDRESS NATIONAL BUREAU OF STANDARDS DEPARTMENT OF COMMERCE WASHINGTON, DC 20234			10. Project/Task/Work Unit No.
			11. Contract/Grant No. MIPR FY1457-77-02054
12. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS *(Street, City, State, ZIP)* Air Force Materials Laboratory Wright - Patterson Air Force Base Ohio 45433			13. Type of Report & Period Covered NBSIR 10/77 - 10/79
			14. Sponsoring Agency Code

15. SUPPLEMENTARY NOTES

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

16. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)*

The ICAM Program Office requires all contractors who develop ICAM software to comply with the following standards governing style and content of ICAM Software Documents. The Style Guide, which covers writing and programming style, applies to all documentation, and individual Content Guides function within the overall framework defined by the IDEF Functional Model and its glossary.

17. KEY WORDS *(six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons)*

Computer-aided manufacturing; computer standards; programming style; software documentation; structured analysis.

18. AVAILABILITY ☐ Unlimited	19. SECURITY CLASS (THIS REPORT) UNCLASSIFIED	21. NO. OF PRINTED PAGES
[✓ For Official Distribution. Do Not Release to NTIS]		
☐ Order From Sup. of Doc., U.S. Government Printing Office, Washington, DC 20402, SD Stock No. SN003-003-	20. SECURITY CLASS (THIS PAGE) UNCLASSIFIED	22. Price
☐ Order From National Technical Information Service (NTIS), Springfield, VA. 22161		

USCOMM-DC