

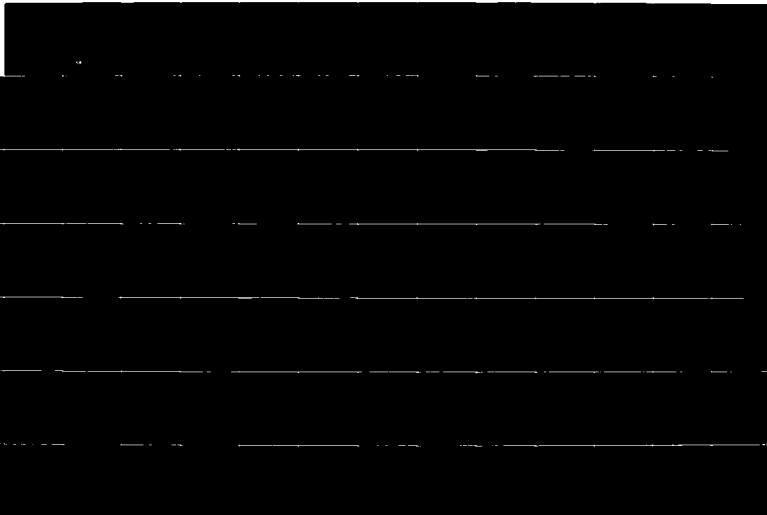
AD-A173 993

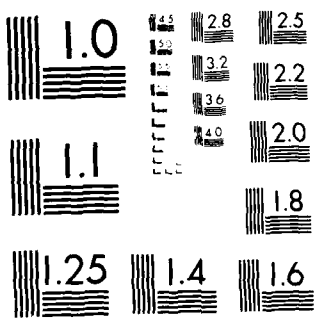
EXPERT SYSTEM FOR SOFTWARE QUALITY ASSURANCE(U) MCLEAN 1/2
RESEARCH CENTER INC VA W E BAUM ET AL. 12 NOV 86
DAAR70-84-D-0052

UNCLASSIFIED

F/G 9/2 NL

47





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A173 993

EXPERT SYSTEM FOR
SOFTWARE QUALITY ASSURANCE

FINAL REPORT

November, 1986

DTIC FILE COPY

DTIC
EXETER
B

DISTRIBUTION STATEMENT

Approved for public release;
Distribution Unlimited



McLEAN RESEARCH CENTER, INC.
1483 Chain Bridge Road, Suite 205
McLean, Virginia 22101

86 11 13 028

EXPERT SYSTEM FOR
SOFTWARE QUALITY ASSURANCE

FINAL REPORT

November, 1986

Prepared for

US Army Belvoir Research, Development and Engineering Center
Ft. Belvoir, Virginia 22060

by

William E. Baum
Judith H. Podell
G. Neil Romstedt

McLean Research Center, Inc.
1483 Chain Bridge Road, Suite 205
McLean, Virginia 22101 (703) 734-1410

DISTRIBUTION STATEMENT

Approved for public release
Distribution Unlimited

Authorization for this research was contract no. DAAK70-84-D-0052, Task no. 0018 (Task Order to a competitive contract awarded on a technical basis). The views, opinions, and/or findings contained in this report are those of the authors and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) MRC (DAAK70-84-D-0052 task 0018)			5. MONITORING ORGANIZATION REPORT NUMBER(S)		
6a. NAME OF PERFORMING ORGANIZATION McLean Research Center, Inc.		6b. OFFICE SYMBOL (if applicable)		7a. NAME OF MONITORING ORGANIZATION DCASMA	
6c. ADDRESS (City, State, and ZIP Code) 1483 Chain Bridge Rd., Suite 205 McLean, VA 22101			7b. ADDRESS (City, State, and ZIP Code) 300 E. Joppa Rd., Room 200 Towson, MD 21204		
8a. NAME OF FUNDING/SPONSORING ORGANIZATION US Army Belvoir Research, Development, Engineering Center		8b. OFFICE SYMBOL (if applicable) STRBE-TQR		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DAAK70-84-D-0052, Task 0018	
8c. ADDRESS (City, State, and ZIP Code) Ft. Belvoir, VA 22060			10. SOURCE OF FUNDING NUMBERS		
			PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.
			WORK UNIT ACCESSION NO.		
11. TITLE (Include Security Classification) Expert System for Software Quality Assurance					
12. PERSONAL AUTHOR(S) W. Baum, J. Podell, G.N. Romstedt					
13a. TYPE OF REPORT Final		13b. TIME COVERED FROM 4/86 TO 10/86		14. DATE OF REPORT (Year, Month, Day) 86 November 12	
15. PAGE COUNT 186					
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP	Expert System		
			Software Quality Assurance		
			Statement of work Preparation		
19. ABSTRACT (Continue on reverse if necessary and identify by block number)					
This report describes the development of an expert system for software quality assurance(SQA). The expert system was designed to facilitate the process of tailoring statements of work by capturing the knowledge of SQA engineers. This task was undertaken in order to alleviate the problems of staff turnover and inexperience and to ensure that the standards and requirement of an adequate SQA Program are enforced, thereby improving the level of performance of the BRDEC SQA mission.					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a. NAME OF RESPONSIBLE INDIVIDUAL Lydia Carrasquillo			22b. TELEPHONE (Include Area Code) 703-664-5988		22c. OFFICE SYMBOL STRBE-TQR

STUDY GIST

Software Quality Assurance (SQA) Expert System

Principal Findings

1. The development of the SQA expert system will facilitate the process of tailoring statements of work by having the knowledge of SQA engineers readily available to ensure that all necessary specifications and requirements are included in order to legally obligate contractors to initiate and maintain an adequate SQA program.
2. The expert system consistently applies expert knowledge and can be used to train new staff members and to improve the level of performance of the BRDEC SQA mission.

Main Assumptions

1. An expert system can be designed to simulate the decision rationale of human experts.
2. The requirements and specifications of an adequate SQA program can be effectively captured by a computer model.

Principal Limitations

1. The construction of the knowledge base requires intensive and time-consuming research to properly capture the appropriate information.
2. The field of expert systems is relatively new and still in the developmental stages.

Scope of the Effort

The development of a system to establish a Software Quality Assurance Program.

Objective

To provide system/hardware integration and management science support necessary for the implementation of the initial phase of a Software Quality Assurance (SQA) expert system for use on mission critical computer software development efforts by the Combat Engineering Directorate.

Basic Approach

1. Information was gathered on the standards, requirements, and specifications necessary to provide an adequate SQA program.
2. Commercially available software was reviewed and analyzed for their adequacy in this initial program and future developments, and based upon agreement with the Government, the most attractive software package was obtained and used to construct the expert system.
3. The expert system was integrated with other programs to create a tailored statement of work.

Reason for Performing the Study

To alleviate the problems of staff turnover and inexperience, and to ensure that the standards and requirements of an adequate SQA program are enforced, thereby improving the level of performance of the BRDEC SQA mission.

Sponsor

Belvoir Research, Development and Engineering Center

Performing Activity and Principal Investigator

McLean Research Center, Inc.
Principal Investigator: G. Neil Romstedt

Comments and Questions

McLean Research Center, Inc.
1483 Chain Bridge Road, Suite 205
McLean, Virginia 22101

DLSIE Accession Number of Final Report

N/A



FOREWORD

This report is submitted to the US Army Belvoir Research, Development, and Engineering Center, Fort Belvoir, Virginia by the McLean Reserach Center, Inc., 1483 Chain Bridge Road, Suite 205, McLean, Virginia. This report summarizes the system/hardware integration support for a Software Quality Assurance (SQA) System.

TABLE OF CONTENTS

	<u>Page</u>
STUDY GIST	i
FOREWORD	iii
1. INTRODUCTION	1 - 1
1.1 BACKGROUND	1 - 1
1.2 OBJECTIVE	1 - 2
1.3 TECHNICAL APPROACH	1 - 3
1.4 SUMMARY OF RESULTS	1 - 5
2. FUNCTIONAL ANALYSIS	2 - 1
2.1 CONCEPTUAL BASIS FOR THE EXPERT SYSTEM	2 - 1
2.2 FUNCTIONAL CONTEXT OF SQA	2 - 2
2.3 INITIAL DESIGN CONSIDERATIONS	2 - 6
3. SOFTWARE SURVEY	3 - 1
3.1 GENERAL	3 - 1
3.2 OBJECTIVE	3 - 2
3.3 RESULTS	3 - 4
3.4 OUR SELECTION	3 - 9
3.5 AI PROGRAMMING LANGUAGES	3 - 10
4. MAJOR SYSTEM COMPONENTS	4 - 1
4.1 SQA.BAT	4 - 1
4.2 THE EXPERT SYSTEM	4 - 2
4.3 SOW	4 - 3
4.4 READER.PAS	4 - 4
5. DESCRIPTION OF KNOWLEDGE BASE	5 - 1
5.1 GENERAL	5 - 1
5.2 OBJECTIVE	5 - 2
5.3 DEVELOPMENT	5 - 2
5.4 INTERRELATIONSHIP OF AVAILABLE DATA	5 - 9
5.5 DESCRIPTION OF KNOWLEDGE BASE	5 - 9
5.6 SUMMARY OF KNOWLEDGE BASE DEVELOPMENT EFFORT	5 - 11
6. ASSESSMENT OF SYSTEM CAPABILITIES	6 - 1
6.1 GENERAL	6 - 1
6.2 BENEFITS OF EXSYS	6 - 1
6.3 LIMITATIONS OF EXSYS	6 - 3
6.4 CONCLUSIONS	6 - 5
6.5 RECOMMENDATIONS	6 - 7

Appendices

- A. INTERNAL LOGIC ANALYSIS
- B. MAINTENANCE MANUAL
- C. SOW FILE
- D. PROGRAM LISTING - READER.PAS

1. INTRODUCTION

1.1 BACKGROUND

The Product Assurance and Testing Directorate of the Belvoir Research, Development and Engineering Center (BRDEC) is responsible to ensure that an adequate Software Quality Assurance (SQA) Program is conducted by the Government and the contractor for each contract issued by BRDEC that includes mission critical computer software development. SQA is an exceedingly complex discipline to apply, requiring many highly trained and experienced engineers from the Directorate staff. An initial and crucial aspect of their job is to ensure that necessary performance specifications, plans, reviews and tests are included in the Statement of Work (SOW) and other sections of contracts in order to legally obligate contractors to initiate and maintain an adequate SQA program. Unfortunately, it may not be until many months after contract award that the SQA engineer learns of any contract defects. Staff turnover compounds this problem because of the length of time needed to adequately train and develop SQA engineers.

In addition, the mission of the Product Assurance and Testing Directorate is wide ranging. Broad expertise in the areas of science, technology and regulation is required among the staff to adequately address the diverse requirements of that mission. Unfortunately, staffing levels within the Directorate are such that individual staff engineers must be competent in many different functional capacities, and performance of those functions can be irregular or infrequent.

An expert system which captures the knowledge of highly skilled SQA engineers and consistently applies that knowledge is looked to as a means to train new staff members and improve the level of performance of the BRDEC SQA mission. An expert system is a computer-based tool which applies expert knowledge in the form of decision rules to the solution of real world problems. Depending upon the complexity of the problems and the capability of the system, it may be used as a time saving support tool for an expert or as a means to substitute less experienced personnel for the expert. It is hoped that expert systems can be integrated into a number of PC-based tools, each supporting a different aspect of the Directorate mission.

1.2 OBJECTIVE

The objective of this task was to demonstrate the feasibility of employing expert systems capability in the accomplishment of a specific product assurance function. Software Quality Assurance (SQA) provisions in Statements of Work (SOW) were selected as the vehicle for this demonstration because of the clear need for improved capabilities in this area. This task provided system/hardware integration and management science support necessary for the implementation of an expert system for use on mission critical computer software development efforts. It focused on implementing the SQA guidelines of DOD-STD-2167.

1.3 TECHNICAL APPROACH

In the course of this demonstration, the following technical approach was employed:

Task 1 -- Initial functional analyses were performed to define the expert system requirements for SQA. The primary emphasis of these analyses was to determine the interaction of SOW preparation with other aspects of SQA so that a comprehensive SQA expert system be built. Preliminary information was gathered on the standards, requirements, and specifications necessary to ensure an adequate quality assurance program and on the characteristics of the software development effort which warrants the invocation of those standards, requirements, and specifications. This task resulted in an understanding of the general structure of the SQA program and a knowledge base of generally accepted practice in applying DOD-STD-2167.

Task 2 -- Commercially available expert system software and AI programming languages were reviewed and analyzed to determine their adequacy for this initial program and future developments. Software was limited to that which can be operated on a Wyse Personal Computer with 640 KB of random access memory (RAM) with a 10 MB hard disk. A total of nine (9) expert systems and four (4) languages were reviewed. These ranged in price from \$100 to \$4500 per copy. In general, the expert system shells were found to be limited by:

- (1) confining screen formats which were difficult to interpret
- (2) a minimal ability to communicate textual data via files. Most shells are oriented around interacting with the user and providing relatively simple (one or two sentences) conclusions
- (3) the number of decision rules and control over those rules. All expert system shells impose overhead along with each decision rule added to the knowledge base, and eventually the 640 KB of RAM is used up.

Task 3 -- Based upon mutual agreement between the Government and the contractor, the most flexible and affordable of the surveyed expert system development package EXSYS was used to construct the SQA expert system, capturing the knowledge obtained in Task 1. The expert system is able to query the operator and, based on his responses, stipulate the necessary correct wording in the SOW. To implement this system, some additional software was developed (the READER program) to access the designated SQA paragraph numbers from a master data base of SQA provisions. This software was also able to provide a consolidated list of CDRL items for the preparation of DD 1423s that accompany the SOW.

Task 4 -- A Software Quality Assurance Program was established in accordance with MIL-S-52779A to assure that all deliverable computer software and documentation under this task order complies with the task order and MIL-S-52779A requirements.

1.4 SUMMARY OF RESULTS

The developed SQA expert system does facilitate the process of preparing statements of work by having the knowledge of SQA engineers readily available to ensure that all the necessary specifications and requirements are included in the correct manner. The developed expert system can also be a training tool to educate less experienced staff members in the proper procedures. Additionally, the consolidated knowledge base reduces the problems associated with job turnover and diminishes the burdens placed on the human experts.

This SQA expert system has been an effective demonstration of the capabilities and limitations of expert systems in product assurance and testing. The significant lessons learned from this effort are:

- (1) Commercial expert system shells for PC's are not yet mature. They are limited by their text handling capabilities and interactive design to applications which draw conclusions which can be simply expressed. External programs are necessary to manage larger conclusions. These shells also contain latent defects which are being corrected by their developers.
- (2) Expert systems suffer somewhat from high expectations caused by overblown advertising and media hype. Expert systems are only the embodiment of decision rules. Programming decision rules into computers is nothing new.

(3) Development of a knowledge base is inherently time consuming and expensive. This effort has only begun the process of SQA knowledge base development. Future work could be pursued along many different paths. A particularly productive path might be to monitor and document the thought processes and actions of skilled SQA engineers in the act of preparing SOWs. A retrospective evaluation of prior SOWs might also yield some insights into the decisions that were made and their effectiveness.

(4) SQA is a relatively new field. With the exception of the DOD Handbook 287, it is largely undefined and undocumented. There is not a large body of published and accepted knowledge from which to draw. Although individual experts exist, individual expertise is hard to ascertain. There is no formal licensing process to certify practicing SQA engineers in the preparation of Statements of Work. As a result, there is no easy source of reference in this application for the development of a knowledge base.

These lessons need to be recognized in future work in applying expert systems technology to Product Assurance and Testing Directorate missions.

2. FUNCTIONAL ANALYSIS

2.1 CONCEPTUAL BASIS FOR THE EXPERT SYSTEM

The Software Quality Assurance (SQA) Expert System was undertaken because of the relatively high turnover among Government personnel involved in SQA and the long lead time required to train personnel in the complex and involved requirements for SQA. This could result in the inconsistent application of SQA standards, which in turn might lead to over/under specification of SQA provisions in software development contracts, and attendant excessive costs/performance risks.

The objective of this system is to provide consistent, high quality guidance to the SQA engineer to assist him in all phases of the SQA process. Particular emphasis will be placed on guiding the novice engineer and providing essential training to rapidly improve his performance. It is recognized that a fully trained SQA expert may surpass the capabilities of the SQA Expert System, but he should be able to impart his knowledge to improve the system.

Since the SQA process is quite complex, only the portions of the process relating to the preparation of SQA provisions in a Statement of Work (SOW) were attempted in this initial effort. Subsequent work will enhance this system with other phases of the SQA process.

2.2 FUNCTIONAL CONTEXT OF SQA

Software Quality Assurance is a continual process which requires constant attention on the part of SQA engineers. The fact that software is the product makes quality assurance more important, due to the difficult testability conditions inherent in most software.

Figure 2-1 illustrates quite simply the steps in the SQA process. Each of these functional steps consists of considerable planning which is governed by numerous regulations and standards. Presuming that there is some materiel development requirement, the first question is to determine whether there is a software component to the materiel product. The software component must be interpreted broadly in this context. It might have many purposes and forms such as:

- (1) Software embedded in a final product which provides some mission critical function or some system monitoring function.
- (2) Software used in the development, test, or simulation of a materiel system, but which is not incorporated directly into the developed hardware.
- (3) Software which becomes "firmware" - hard wired instructions in silicon chips or specifically designed chips to carry out specific functions.
- (4) Software as an end product, not related to a materiel system, which is used for technical or administrative purposes.

The key ingredient is that there is a software component which deserves quality assurance attention consistent with its role.

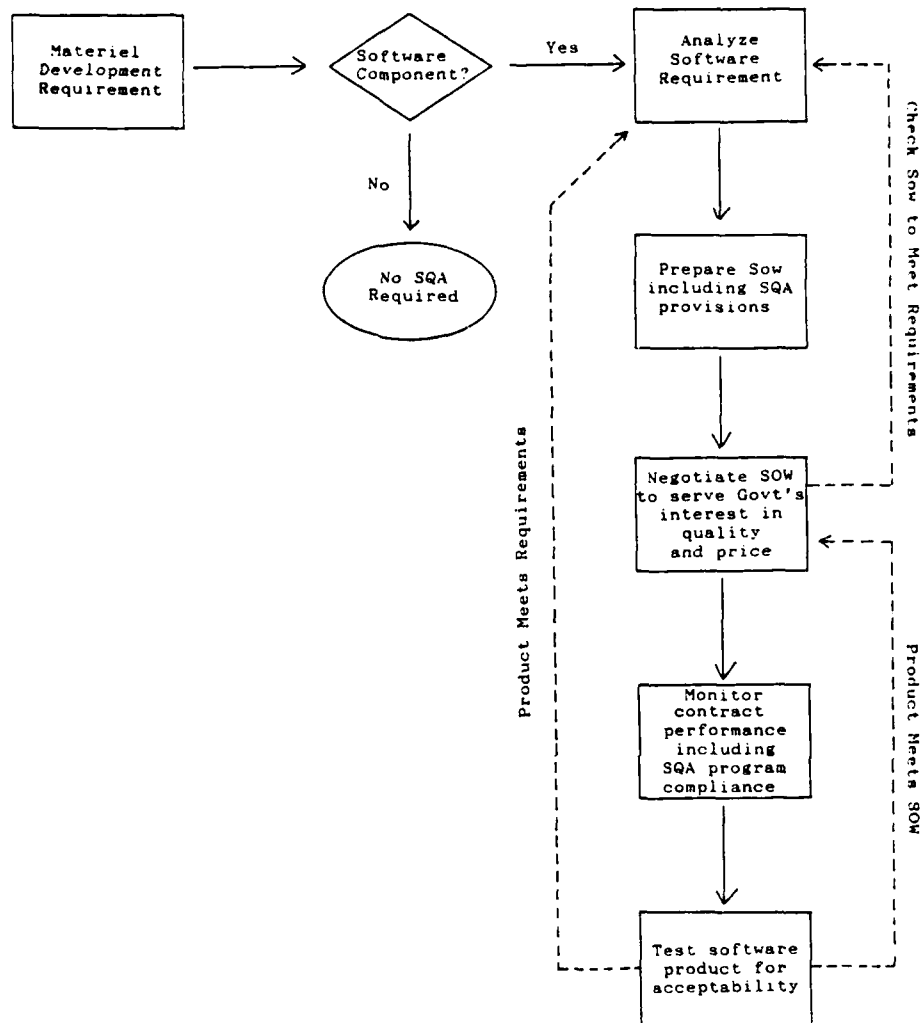


Figure 2-1 Functional Context of SQA

The software requirement then undergoes analysis to determine its contribution to and purpose in the materiel being developed. For most systems, the major software components are explicitly and formally defined in the program plan and/or functional block diagram of the system. Other software components are implied through the incorporation of subsystems (e.g., GPS) and ancillary systems (e.g., ATE) whose software might be considered to be already established and tested. The SQA engineer becomes familiar with each component and through discussions with the project technical staff arrives at an understanding of the software complexity, innovativeness, and significance to the materiel system and of the resources available to develop the software.

The SQA engineer uses this knowledge to prepare SQA provisions to be included in the software development SOW. In the SOW, the SQA engineer is afforded much latitude by regulation to tailor the standard SQA provisions to the specific software development task. He can make the provisions stringent, obligating the contractor and the Government to perform many SQA tasks when the role of the software product demands such attention. Conversely, the provisions can be lax when the software product is technically unchallenging, simple to test, or already developed. The SOW is prepared with a clear statement of the software to be developed and the SQA provisions which will be applied, and forwarded for procurement action.

When software development contracts are negotiated, the contractor often proposes alternative wording to the product definition and to the SQA provisions. At these points, the SQA engineer is called upon to exercise his judgement to protect the Government's best interests in both product quality and price. The contractor's counteroffer, while typically less expensive, may

pose some additional risk to the software product and subsequently to the materiel system it supports. The SQA engineer recommends acceptable contract language which allows an adequate SQA program. Since the negotiated contract defines the product which will be delivered, it is essential that it accurately embody the true software requirement. The SQA engineer ensures that the software product is clearly understood and accurately defined in the contract language.

During the contract performance period, the SQA engineer performs such duties as may be stipulated by the SQA program. As a minimum, these will include monitoring of the contractor's compliance with the SQA program. Other specific reviews of the product might also be required, as well as formal reviews and approval of SQA documents. These tasks will be defined in the contract.

When the software product is completed, it must undergo some acceptability testing. This testing phase will be appropriate to the complexity of the product and to its operational role within the materiel system. Usually, the testing program is explicitly defined in the SQA provisions within the contract. Two checks are made in this phase. First and foremost is to ensure that the contractor has delivered software and SQA documentation which meets the standards specified in the contract. If the contract adequately reflected the true software requirements, then the final product will meet that need. However, it could also be that in the preparation and negotiation of the contract that the software requirement was obscured. The result might be a product which complies with the contract but fails to address the true requirements, in which case corrective action must be pursued and lessons learned.

2.3 INITIAL DESIGN CONSIDERATIONS

As stated earlier, this effort will focus on the development of an expert system to aid in the specification of SQA provisions within a Statement of Work. This is only one small aspect of the total SQA process, although an important one. In addition, the SQA provisions are only a portion of the SOW and the contract package. So it is critical that the scope and objectives of this expert system be kept in perspective with respect to the overall SQA mission.

The primary consideration in designing this system was the conclusion that decision rules can be formulated to accommodate the process of developing SOWs. Systematic approaches can be taken by SQA experts with a minimum of intuitive or arbitrary considerations. This conclusion was demonstrated in the earlier SQA Tailoring Handbook produced by BRDEC, which provided decision rules to select among five SOW templates. It is important that the process be inherently rational and non-stochastic if consistent applications are to be achieved.

The second consideration was that unknown data should be conservatively accommodated. A relatively inexperienced SQA engineer might be unable to provide meaningful answers to many of the expert system's questions. He should not be penalized for his lack of experience. Instead, the system should delve further into the issues with more specific or precise questioning. When the answer is simply not known, the expert system should make the most stringent interpretation of the SQA provisions.

Because of the limited scope of the expert system, manual review and editing of the SOW will be necessary. This review and editing will ensure the integration of the SQA provisions with the other elements of the SOW and contract package. It will also allow the proper managerial supervision of the expert system output. To minimize the difficulties inherent in transferring data among various incompatible computer systems and software packages, it was determined that a commercial word processor be used to perform the editing tasks and that the SOW files created by the expert system use standard ASCII characters only. Figure 2-2 depicts the integration of the word processor into the development of a finished SOW.

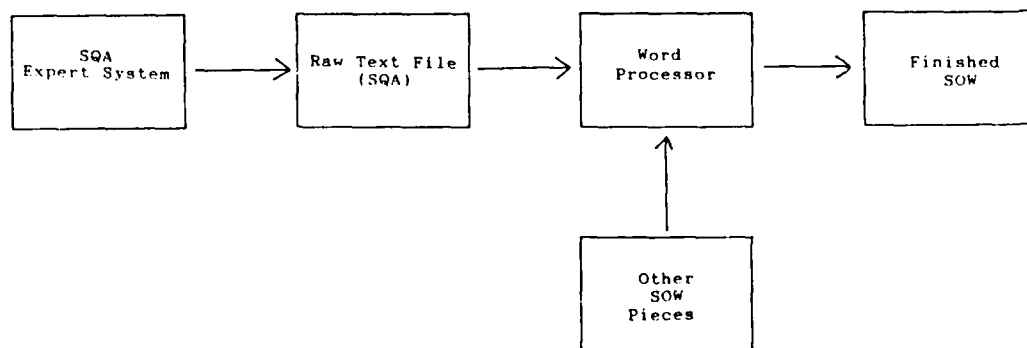


Figure 2-2 Initial Design Concept

Finally, to allow the system to conveniently grow and yet still fit within the capacity of existing commercial expert system shells, the expert system would have to operate as a "filter" on an all-inclusive SQA SOW. The expert system will select paragraphs from the comprehensive SQA SOW which are appropriate to the software development program currently being considered. The result will be tailored SQA provisions. System growth and enhancement can be accommodated by the addition of paragraphs (of any length) to the master SOW file, and by the incorporation of new selection criteria into the expert system shell.

3. SOFTWARE SURVEY

3.1 GENERAL

The interest in artificial intelligence (AI), getting computers to think like human beings, has been around nearly as long as computers themselves. Most of the work in this area has been purely theoretical and very few practical applications have emerged. However, in the recent past, with the advancement of technology and our own knowledge of human intelligence, several subsets of AI have achieved commercial success; among them, the development of expert systems. An expert system is a computer-based model that simulates the decision rationale of human experts.

The early expert systems were developed using the popular programming languages available at that time, but soon other languages were created, like Pascal and LISP, that were more suitable to the inference structure of expert systems. Then, as the success of expert systems grew, development packages known as shells began to emerge. These shells facilitate the construction of expert systems by providing a framework for the decision rationale.

Today, the field of expert systems is still in its infancy and so is the associated software. Within the past few years, commercially available shells and programming languages have infiltrated the marketplace. However, because this is such a new field, the majority of the software is still in the developmental stages and is geared toward the more general applications. To find the appropriate software for our application required an extensive search and analysis of the available products.

3.2 OBJECTIVE

In investigating the available software for designing an expert system, several criterion had to be met to satisfy the requirements of our application. The primary requirement of the software was that it had to be compatible with a WYSE PC with 640 KB of RAM and a 10 MB hard disk. With this in mind, our search included commercially available expert system shells, programming languages specifically designed for expert system applications, and general purpose programming languages.

Our search began by reading the available literature found in computer-oriented magazines and journals. The opinions of users and critics were studied, as well as the current theories and doctrines of artificial intelligence in general. Based on these discoveries, we then contacted many of the software vendors to learn more about the specifics of their packages, and when available, obtained demonstration disks to gain first hand exposure to the product.

The first criteria used in evaluating the software was ease of use because the system is intended to be used by people with a limited knowledge of the technology involved, so we wanted it to be as user-friendly as possible. Additionally, it had to be relatively easy to develop because it was necessary to deliver a demonstration model within a short amount of time. We were also constrained by the cost of the software. Therefore, we tended to focus our search on moderately priced expert system shells.

Another consideration was the availability of liberal licensing arrangements. The final product will be used at multiple work stations and this requires several copies of the model to be available. To obtain unlimited use of the final program, a flexible and inexpensive licensing policy is desired.

Other characteristics focused on the technical capabilities of the software. To implement the system the way that it was envisioned required the ability to interface with the operating system in order to create and use external files. It was also vital that the system be able to interface with a word processor. Additionally, the manner that the rules were developed and processed and the availability of forward and backward chaining were key considerations.

Because of the nature of our application, certain elements of the software were of little concern to us. For example, sophisticated co-processors to perform complicated mathematical calculations are not required, nor is the ability to infer rules based on previous dialogues with the system. These capabilities added greatly to the cost of the package but would not enhance our application.

With these considerations in mind, our search produced a number of likely candidates. Although no package matched our requirements exactly, some came quite close. We will now examine the results of our software search.

3.3 RESULTS

After investigating almost fifty potential software packages, about a dozen qualified as deserving further attention. These packages varied greatly in some areas. For example, the cost ranged from \$100 to \$4500. However, they all were compatible with the designated computer hardware. Each one seemed to have some unique feature to differentiate it from the others. The following are descriptions of these shells and programming languages:

Human Edge Software -- Expert Ease

Price \$695

Requires 128K memory and two disk drives.

Uses spreadsheet format to structure its examples and attributes, or knowledge base, from which the program draws conclusions.

Inductive logic program.

Limited in large applications because it can address only 128K of memory.

Capable of 255 examples with 31 attributes and 31 decisions in each example.

Demands consistency; Can't have two identical examples leading to different conclusions.

Written in UCSD Pascal.

Human Edge Software -- Expert Edge

Price \$795

Requires a minimum of 256K memory but 512K is recommended, Two disk drives.

Rule-based, uses deductive reasoning inference.

Forward and backward chaining.

The rules can incorporate calculations, equations, logical reasoning, judgement, fact and uncertainties.

Supports data entry through a natural language interface.

Capable of up to 500 rules (needs approximately 1K of memory for each rule).

Exsys, Inc. -- EXSYS

Price \$295
Requires PC with 256K RAM.
Uses straight text presentations that ask multiple choice questions.
Forward and backward chaining.
Capable of using external files.
Interface with word processors and data bases.
Good use of color.
700 rules per 64K of memory over 192K of RAM.
Allows probabilities.
Weak manual.
Liberal licensing arrangements.

McDonnell Douglas -- REVEAL

Price \$4500
Requires 640K RAM, hard disk drive, 8087 coprocessor recommended.
Forward and backward chaining
Knowledge based rules.
Fuzzy logic.
Mandatory one week training course.
Best suited to financial planning and analysis.
Written in FORTRAN.

Expert Systems International -- ES/P Advisor

Price \$895
Requires 256K RAM, Dos 2.x, two disk drives.
Can handle applications that guide a user step by step through a complete process giving all essential information at each stage.
Prolog-based - Has an interface to that language for the PROLOG programmer. This interface makes it one of the very few expert system development packages with an open-ended architecture, in the sense that a qualified programmer can use a fully fledged programming language to add to the system features it currently lacks.
Capable of 500 rules.

California Intelligence -- XSYS

Price \$995

Requires IBM PC/XT/AT with at least 640K of memory, Dos 2.x, and the programming language IQLISP.

Forward and backward chaining.

Interfaces with external file for input to a spreadsheet or as a database inquiry.

Uses certainty factors and weights (8087 chip needed).

Arity -- Expert System Development Package

Price \$295

Requires IBM PC or equivalent with 512K memory and hard disk Rule based system.

Forward and backward chaining.

Interfaces with other languages.

4 mb of memory for storage of rules which allows 5000-20000 rules.

This product requires either the Arity/Prolog Compiler and Interpreter (\$795) or the Arity/Prolog Interpreter (\$350) for development.

They do not require you to purchase a run-time license or to pay royalties in order for you to distribute compiled applications that you build.

Radian -- Rulemaster

Price \$995

Requires IBM PC/XT, PC/AT or equivalent, Dos system, 640K memory.

Automatic generation of rules from examples; declarative (examples) in addition to procedural (rules).

Capable of interfacing to other existing programs.

Forward and backward chaining.

MDBS -- Guru

Price \$2995

Requires IBM PC or compatible, MS Dos 2.x, 512K memory (640K recommended).

Capable of 3000 rules.

Forward and backward chaining.

Can handle incomplete or uncertain information.

Interface with databases and spreadsheets.

Natural language interface.

Menu guided interaction.

Graphics.

Input/output controlled by user.

Arity -- Prolog

Price \$350 Prolog interpreter
\$795 Native -code compiler with interpreter
Requires MS Dos 2.x.
No license fees for stand-alone programs.
Good documentation included.
I/O support is adequate for most AI applications but is limited for general applications that require formatted output.

Chalcedony Software -- Prolog V Plus

Price \$99.95
More than 100 predefined predicates and operators.
Double precision floating point arithmetic.
Able to use up to 640K RAM.
Text and graphic screen manipulation.
Documentation.

Gold Hill Computers -- Golden Common Lisp

Price \$495
Requires IBM PC or 100% PC compatible with 512K memory, MS Dos 2.0 or higher, one DSDD disk drive.
Corporate and educational licenses available.
The interpreter supports over 400 primitives.
The editor features multiple buffers and windows, parenthesis matching, automatic code indentation, and form evaluation within the edit buffer.
Speed and space limitations - Best for small size problems, Not suitable for large applications.

This list is summarized in the following table:

	Cost	Forward Chaining	Backward Chaining	Licensing Policy	Word Processor Interface	Can Use External Files	Minimum Memory Requirement	Demo Available
Expert Ease	695	No	No	Rigid, Costly	No	No	128K	Yes
Expert Edge	795	Yes	Yes	Rigid, Costly	No	No	512K	Yes
EXSYS	295	Yes	Yes	Liberal, Inexpensive	Yes	Yes	256K	Yes
Reveal	4500	Yes	Yes	?	?	?	640K	
ES/P Advisor	895	No	No	?	?	?	256K	
XSYS	995	Yes	Yes	?	Yes	Yes	640K	
Arity Expert Sys Development Pack	1090	Yes	Yes	?	Yes	Yes	512K	
Rulemaster	995	Yes	Yes	Rigid, Costly	Yes	Yes	640K	
Guru	2995	Yes	Yes	Costly	No	No	640K	Yes
Arity Prolog	350	-	-	Inexpensive	-	-	256K	
Chalcedony Prolog V Plus	100	-	-	Flexible, low price	-	-	256K	
Golden Common Lisp	495	-	-	?	-	-	512K	

3.4 OUR SELECTION

After careful consideration and analysis of our requirements, we chose the EXSYS development package shell. Our decision was based partly on the availability of a demonstration disk which exhibited many of the features we were looking for. Additionally, this software package received favorable reviews from two of the leading computer magazines.

EXSYS is relatively easy to understand with little prior knowledge of expert systems. It interfaces well with word processors, databases and spreadsheets, and it has a built-in function which can produce external files. It can also call to already existing files and use them in its processing. The licensing policy was found to be more appealing than the other software packages, and at a cost of \$295, it provided many of the features of the more expensive packages.

Although some limitations were found after working with EXSYS for awhile, it would have been unusual not to. Because a shell package must be designed to appeal to a wide range of users, it ends up not being too specific in its applicability. In other words, unless you build it yourself for your own purpose (which would have been far too costly and time consuming at this stage) there will always be some shortcomings.

NOTE: We have recently been in touch with the developers of EXSYS to convey our evaluation of their product. Many of our problems with it have been voiced by other users and the product is currently being updated to eliminate these limitations. The new version should prove to be a far superior product and it is scheduled to be available in the next few months.

3.5 AI PROGRAMMING LANGUAGES

In addition to examining commercial expert system shells for their utility in the SQA expert system, we also looked at AI programming languages to see if they might offer some additional capability. Particularly, we were interested in how they might provide the convenient and natural method of context-sensitive help that we found lacking in the EXSYS shell. We were also attempting to determine the capacity of these languages to maintain and use rules in what may eventually become a very large system.

There are several AI programming languages available for the IBM-PC compatible computers, and many compilers offered for each language. The most common languages are LISP, PROLOG, and SMALLTALK. The use of the term "AI" is typically meant to denote:

- (1) languages which operate at a very high level. A single statement can be very powerful and cause many "hidden" actions to occur within the program.

- (2) Operate on "objects" as well as numbers and characters. These objects are things (e.g., "bird", "truck", "sky") to which are ascribed a lot of characteristics, and relate closely to the way people perceive their world.

- (3) Transparently embody advanced programming language features such as recursion and list processing to quickly and efficiently perform searches for logical relationships among objects.

Naturally, all of these features can be realized in suitable non-AI programming languages (such as PASCAL, FORTRAN, or BASIC), but these require very clever and detailed algorithms and highly skilled programmers. The objective of the AI languages is to reduce that complexity and hence to bring these capabilities to the average programmer.

This objective is not yet achieved. The conceptual basis of these languages is not well understood by programmers in general, particularly those with a traditional outlook to logical flow. AI languages appear to be more "free form" than most, so a clear idea of a program's logical flow may be difficult to obtain. In addition, the syntax of these languages is unusual and represents a barrier to learning them. Thus, the result is that these languages are appropriate for specialists who can take the time to learn them.

We selected the PROLOG language for a more detailed examination because there are several low-cost, respected compilers available for the IBM-PC. In addition, it had been described as more understandable than LISP (whose syntax is difficult) and more controllable than SMALLTALK (which typically possesses limited input/output potential). The TURBO PROLOG package from Borland International, Inc. (the makers of TURBO PASCAL) was specifically reviewed. Borland describes their product as follows:

Let's take a closer look at how Turbo Prolog differs from traditional programming languages.

Turbo Prolog is descriptive. Instead of a series of steps specifying how the computer must work to solve a problem, a Turbo Prolog program consists of a *description* of the problem. This description is made up of three components, with the first and second parts corresponding to the declaration sections of a Pascal program:

1. Names and structures of objects involved in the problem
2. Names of *relations* which are known to exist between the objects
3. *Facts* and *rules* describing these relations

The description in a Turbo Prolog program is used to specify the desired relation between the given input data and the output which will be generated from that input. *Turbo Prolog uses facts and rules.* Apart from some initial declarations, a Turbo Prolog program essentially consists of a list of logical statements, either in the form of facts such as:

it is raining today.

or in the form of rules such as:

you will get wet if it is raining
and you forget your umbrella.

Turbo Prolog can make deductions. Given the facts

john likes mary.
tom likes sam.

and the rule

jeanette likes X if tom likes X.

Turbo Prolog can deduce that

jeanette likes sam.

You can give the Turbo Prolog program a *goal*, for example

find every person who likes sam

and Turbo Prolog will use its deductive ability to find all solutions to the problem.

Execution of Turbo Prolog programs is controlled automatically. When a Turbo Prolog program is executed, the system tries to find all possible sets of values that satisfy the given goal. During execution, results may be displayed or the user may be prompted to type in some data. Turbo Prolog uses a *backtracking mechanism* which, once one solution has been found, causes Turbo Prolog to reevaluate any assumptions made to see if some new variable values will provide new solutions.

To illustrate TURBO PROLOG, a small example program is presented below:

```
/* Program 4 */
domains
    person = symbol

predicates
    male(person)
    smoker(person)
    vegetarian(person)
    sophie_could_date(person)

goal
    sophie_could_date(X) and
    write("a possible date for sophie is ",X) and nl.

clauses
    male(joshua).
    male(bill).
    male(tom).
    smoker(guiseppe).
    smoker(tom).
    vegetarian(joshua).
    vegetarian(tom).
    sophie_could_date(X) if male(X) and not(smoker(X)).
    sophie_could_date(X) if male(X) and vegetarian(X).
```

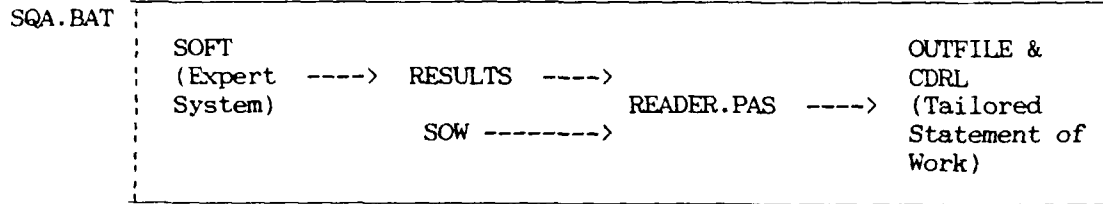
This program will determine that Sophie could date Joshua or Tom. Joshua is selected because he's male and not a smoker and then Tom is selected because he's male and a vegetarian. Even though Joshua is also a vegetarian, that fact is irrelevant because the program accepted him after it concluded that he did not smoke. This demonstrates that the order of predicate and clause presentation is critical to the logical flow in the program.

To its great credit, TURBO PROLOG provides extensive interactive output capability with windows, colors, and graphics displayed on the screen. Its external files capability appears to be limited to text data, although that data can be interpreted as objects, numbers or characters. It provides the

benefits of both interpretive and compiled program performance to assist in development and implementation of an application. With additional effort, most of the rules contained in the "clauses" section could be stored in external files. So this PROLOG version is not inherently limited by RAM space in the computer.

In an analysis of this SQA expert system task, it was determined that programming it into PROLOG would take too long for the limited scope of this effort. Most of the programming time would be taken in learning the system through trial and error. In addition, many of the fine features of the commercial expert system shell, EXSYS, would be lost without significant extra programming effort. These include the interactive editing of rules, program restart capabilities, and rule text and help screens. In a situation where a more modest software project with a longer leadtime is undertaken, it might be a proper investment of resources to apply PROLOG and realize its potential.

4. MAJOR SYSTEM COMPONENTS



4.1 SQA.BAT

As indicated above, all of the programs are directed by the batch file called SQA.BAT. The expert system is called up first and its rules and questions are stored in a file called SOFT. The report generator function is used to produce the output file RESULTS from the expert system session. This file contains the paragraph numbers that are to be included in the statement of work, as determined from the expert system. The program READER.PAS then takes this file of numbers and the file SOW, which is the master statement of work, and sorts through the numbers and constructs the tailored statement of work. The tailored statement of work is contained in the file OUTFILE and the associated contract deliverable requirement lists are in the file CDRL. The batch file then copies these two files to the word processor (in our case, Wordstar) and opens OUTFILE so the user may view and/or modify it. We shall now look at each of these components in more detail.

4.2 THE EXPERT SYSTEM

4.2.1 General

The expert system was designed using the EXSYS development package, a commercially available expert system shell. A shell program does not contain any rules, but it is designed to enable the user to create his own expert system by entering rules which will be processed and run by the shell. The objective of this expert system is to produce a list of numbers which correspond to the paragraph numbers in the SOW file (more on this in the next section). To accomplish this, the system consists of many multiple choice questions, and then for each question there are rules to determine the impact of the response. These rules are structured in an IF-THEN-ELSE type format, and if the IF part of the rule is true, the THEN part is executed. Conversely, if it is false, the ELSE part is executed. The THEN and ELSE consist of instructions: solving a mathematical expression; selecting an EXSYS-defined choice; assigning a value to a variable. (This process is further defined in the maintenance manual). If a rule determines that a paragraph is to be included in the statement of work, then that paragraph number will be assigned to a variable (that variable's value is initialized at the start of the run to -999 and if the paragraph is not to be included, then the variable's value will remain -999). Then at the conclusion of the run, only those variables with a value greater than zero will appear in the RESULTS file.

4.2.2 EXSYS Files

The questions and rules for our application are stored in the EXSYS files SOFT.TXT and SOFT.RUL, respectively. At the completion of the run, the user's responses are stored in a file called INPUT. That way, if any changes or modifications in the answers are desired they can be made without having to answer all the questions again. Additionally, if the user wished to quit his session before it was completed, his answers can be stored so that he could resume the session at a later time. The system will ask the user for a file name and his answers will be stored there.

EXSYS also has the ability to produce external output files through its report generator function. This enables the user to control what data is output to a disk file or printer. The instructions for the output's form and content are kept in a file with the same file name as the expert system (in our case, SOFT) with an .OUT extension. In our application, the file SOFT.OUT creates the file RESULTS which will contain all the paragraph numbers that are to be included in the statement of work. SOFT.OUT instructs the system, if the value of a variable is greater than zero, to write that value to the RESULTS file. It also writes the numbers of the paragraphs that are always included in the statement of work. This external file is then used in conjunction with the SOW file to construct the tailored statement of work.

4.3 SOW

The SOW is a text file containing the "master" statement of work. This "master" consists of every paragraph that could possibly appear in a statement of work. From it, the appropriate paragraphs will be selected and reorganized

into the final tailored statement of work that will fulfill the software quality assurance requirements for the specific contract it is being written for. In order to identify the paragraphs, a numbering system was used. The paragraphs are labeled with numbers ranging from 10.00 to 7700.00. These numbers are preceded by a '@' in order to ensure that other numbers within the paragraphs are not mistaken for these labels.

Also included in the SOW file is the information about the contract deliverable requirements list (CDRL). It tells which blocks the information is to go into and what the information is. This data is identified in the SOW file by a caret (^). Both the caret (prior to the CDRL data) and the '@' (prior to the paragraph numbers) are used in the program READER.PAS to construct the tailored statement of work. A complete listing of SOW is included in Appendix C of this report.

4.4 READER.PAS

The READER program was written for this specific application, as required by the overall architecture described above. READER forms the critical link between the output of the expert system shell and the master SOW file. It takes the file of paragraph numbers produced by EXSYS, finds those paragraphs in the master SOW file, and copies them to the tailored SOW file. In this process, it also produces a list of the DD Form 1423 entries which should be made to be consistent with the SQA provisions in the SOW.

READER is a very simple program. It is written in Turbo Pascal 3.0 for the IBM-PC and compatibles running the MS-DOS operating system. Commented source code for READER is included in Appendix D for in depth review. It employs several intrinsic functions of Turbo Pascal which are not ANSI or ISO

standard Pascal, including PARAMSTR, VAL, CLRSCR, TEXTBACKGROUND, and TEXTCOLOR. These may be easily replicated in other Pascal implementations.

READER first inputs the paragraph numbers produced by EXSYS and sorts them in increasing order. The master SOW file is then sequentially scanned, looking for paragraph numbers which match the sorted list. When a match is found, the paragraph body is copied from the master SOW file to the tailored SOW file. If DD Form 1423 data is indicated for a selected paragraph, then it is also copied, this time to a separate file of CDRL data. The program ends when all the paragraphs have been found or the master SOW file has been completely scanned.

Future enhancements to READER could include automated paragraph sequencing in the tailored SOW file, and automatic scheduling of deliverables on DD Form 1423 to fit within the contract performance period.

5. DESCRIPTION OF KNOWLEDGE BASE

5.1 GENERAL

This chapter presents a definition of the term "knowledge base" as used in this program effort; describes the objective of the knowledge base created in conjunction with this project; and details sources of information that forms the SQA knowledge base and how that information has been organized. The actual knowledge base that was developed under this program effort is contained in Appendix B of this report.

A knowledge base for an expert system is analogous to a data base for a traditional computer program. The results obtained from the computer program are only as good as the data used in arriving at the results. An expert system is no different. Without a reliable knowledge base, the expert system is useless.

In this program effort, the knowledge base was structured to include the pertinent basic information and the decision rational of human experts necessary to automatically create a properly tailored statement of work (SOW) to satisfy Software Quality Assurance (SQA) requirements during the development of computer software.

The body of knowledge and theory of application of that knowledge in the SQA field is constantly evolving. The expert system and knowledge base created under this program effort were intentionally configured to accomodate

inevitable change. Appendix B describes how changes to the knowledge base can be accommodated by the system.

5.2 OBJECTIVE

The objective of the knowledge base developed for SQA SOW preparation was to replicate as closely as possible the basic information available to, and the decision rationale of human experts who are knowledgeable in the preparation of SQA SOWs. The reason for this was two fold. The first reason was to ensure that the expert system would produce a competently tailored SQA SOW. The second reason was to ensure acceptance of the product of the expert system by the SQA community.

5.3 DEVELOPMENT

Two basic sources of information were exploited during the development of the SQA data base. The first source was published documents regarding SQA standards, requirements and specifications that form the basic written guidance for the manual development of SQA SOWs. The second type of information was extracted from human sources who were identified as authorities in the SQA field.

5.3.1 Published Documents

Published documents that were used in the development of the SQA data base include:

- (a) DOD-STD-2167, Defense System Software Development, 4 June 1985
- (b) DOD-STD-2168, Software Quality Evaluation, 26 April 1985 (Draft)
- (c) MIL-S-52779A, Software Quality Assurance Program Requirements, 1 August 1979
- (d) MIL-HDBK-334, Evaluation of a Contractor's Software Quality Assurance Program, 15 July 1981
- (e) DOD-STD-1467, Military Standard Software Support Environment, 18 January 1985
- (f) Technical Report on the Comparison of DOD-STD-1679A and DOD-STD-2167, Logicon, 27 September 1985 (Draft)
- (g) DOD-HDBK-287 (Draft), Defense System Software Development Handbook, 6 September 1985
- (h) Software Development Guidelines, Dynamics Research Corporation, 6 September 1985
- (i) DOD-HDBK-287 (Draft), Defense System Software Development Handbook, 23 May 1986
- (j) Joint Logistics Commanders Software Development Transition Support, Comment Analysis Report, Volume I, DOD-KDBK-287, Dynamics Research Corporation, 30 April 1986
- (k) AMC-P 702-XX, Software Quality Requirements for Software Systems Development and Production, January 1985
- (l) AMC-P 70-4, Appendix C, DD Form 1423 Preparation Instructions
- (m) Belvoir R&D Center Handbook for Determining Statements of Work for Software Development, Software Quality Evaluation, and Independent Software Verification and Validation, Teledyne Brown Engineering, January 1986
- (n) Data Item Descriptions:
 - Software Development Plan, DI-MCCR-80030
 - Software Configuration Management Plan, DI-MCCR-80009
 - Software Quality Evaluation Plan, DI-MCCR-80010
 - Software Standards and Procedures Manual, DI-MCCR-80011
 - System/Segment Specification, DI-CMAN-80008

Software Requirements Specification, DI-MCCR-80025
Interface Requirements Specification, DI-MCCR-80026
Software Top Level Design Document, DI-MCCR-80012
Software Detailed Design Document, DI-MCCR-80031
Interface Design Document, DI-MCCR-80027
Database Design Document, DI-MCCR-80028
Software Product Specification, DI-MCCR-80029
Version Description Document, DI-MCCR-80013
Software Test Plan, DI-MCCR-80014
Software Test Description, DI-MCCR-80015
Software Test Procedure, DI-MCCR-80016
Software Test Report, DI-MCCR-80017
Computer System Operator's Manual, DI-MCCR-80018
Software User's Manual, DI-MCCR-80019
Computer System Diagnostic Manual, DI-MCCR-80020
Software Programmer's Manual, DI-MCCR-80021
Firmware Support Manual, DI-MCCR-80022
Operational Concept Document, DI-MCCR-80023
Computer Resources Integ. Support Document, DI-MCCR-80024

5.3.2 Description of Published Sources of Information

(a) DOD-STD-2167, Defense System Software Development

This standard contains requirements for the development of Mission Critical Computer System software. It establishes a uniform software development process which is applicable throughout the system life cycle. The software development process defines development activities which result in: (1) the documentation, (2) the application of development tools, approaches and methods, and (3) project planning and control. It incorporates practices which have been demonstrated to be cost effective from a life cycle perspective, based on information gathered by the Department of Defense (DOD) and industry.

(b) DOD-STD-2168 (draft), Software Quality Evaluation

This standard contains requirements for evaluating the quality of software and associated documentation and activities for Mission Critical Computer Systems, and for performing the planning and follow-up activities necessary to ensure that necessary changes are made.

This standard is intended to be used in conjunction with DOD-STD-2167. These standards, together with other DOD and military specifications and standards governing configuration management, specification practices, project reviews and audits, and subcontractor control provide a means for achieving, determining, and maintaining quality in software and associated documentation.

(c) MIL-S-52779A, Software Quality Assurance Program Requirements

The purpose of this specification (which has been superseded by DOD-STD-2168) was to establish the basic requirements for Software Quality Assurance. This document was utilized in this program effort as a source of background information.

(d) MIL-HDBK-334, Evaluation of a Contractor's Software Quality Assurance Program

This document provides guidance to personnel responsible for the evaluation of a contractor's software quality program when Military Specification, MIL-S-52779A, is invoked in the contract. This document was superseded along with MIL-S-52779A, but none-the-less provides valuable background information regarding SQA.

(e) DOD-STD-1467, Military Standard Software Support Environment

This standard defines the efforts necessary to ensure the existence of a complete life-cycle software support capability for the contractually deliverable software when it enters the operational inventory.

(f) Technical Report on the Comparison of DOD-STD-1679A and DOD-STD-2167

This technical report presents the results of an in-depth comparison of software development standards DOD-STD-1679A and DOD-STD-2167 and their associated Data Item Descriptions (DIDs).

(g) DOD-HDBK-287 (Draft), Defense System Software Development Handbook, 6 September 1985

This handbook aids in the application and use of DOD-STD-2167, Defense System Software Development; MIL-STD-483, Configuration Management Practices for Systems, Equipment, Munitions, and Computer Software; MIL-STD-490, Specification Practices; and MIL-STD-1521, Technical Reviews and Audits for Systems, Equipments, and Computer Software. It contains overview material, topics of importance, tailoring methodology, and tailoring examples for the above software standards. This draft was superseded by the 23 May 1986 Draft but contains significant material regarding SOW tailoring that is considered valid by the SQA community.

(h) Software Development Guidelines, Dynamic Research Corporation, 6 September 1985.

This technical report forwarded the draft handbook DOD-HDBK-287 at (g) above to the Government.

(i) DOD-HDBK-287 (Draft), Defense System Software Development Handbook, 23 May 1986

This handbook aids in the application and use of the following standards: DOD-STD-2167, Defense System Software Development; MIL-STD-483, Configuration Management Practices for Systems, Equipment, Munitions, and Computer Software;

MIL-STD-490, Specification Practices; and MIL-STD-1521, Technical Reviews and Audits for Systems, Equipments, and Computer Software. It contains overview material, discussion of software issues, tailoring methodology, and tailoring examples for the listed software standards. This draft expands on the tailoring methodology presented in the 6 September 1985 draft of this document.

(j) Joint Logistics Commanders Software Development Transition Support

The purpose of this report is to summarize Dynamic Research Corporation's analyses and assessments of comments submitted by Government and Industry against draft DOD-HDBK-287, as a result of a review.

(k) AMC-P 702-XX, Software Quality Requirements for Software Systems Development and Production

This standard provides general requirements and specific tasks for the software quality programs during the requirements, preliminary design, detailed design, coding, testing, and initial deployment of software systems.

(l) AMC-P 70-4, Appendix C, DD Form 1423 Preparation Instructions

This document provides information regarding the preparation of DD Form 1423, Contract Data Requirements List (CDRL).

(m) Belvoir R&D Center Handbook for Determining Statements of Work for Software Development, Software Quality Evaluation, and Independent Software Verification and Validation

This document provides an approach for determining the complexity of the software in a system and for selecting software Statements of Work (SOW) appropriate for the system. This document provides an approach for estimating

the complexity of the software of Army mission critical computer system software under control of the Belvoir Research, Development, and Engineering Center (BRDEC) for Software Development (SD), Software Quality Evaluation (SQE), and Software Independent Verification and Validation (IV&V). This document provides guidance for the manual selection of an SOW for software development.

(n) Data Item Descriptions (DIDs)

These documents provide information regarding the format and content preparation instructions for data generated under applicable tasks invoked by contract for software development.

5.3.3 Human Experts

Human experts knowledgeable and authoritative in the field of SQA were interviewed to obtain basic information and decision rationale. Personnel from the Standardization and Data Management Branch, Department of the Navy, and members of the Joint Service Computer Resource Management Committee were identified as the individuals with the most expertise within the SQA community. These individuals were contacted, interviewed, and their inputs were incorporated into the knowledge base.

5.4 INTERRELATIONSHIP OF AVAILABLE DATA

The data sources were interrelated for the purposes of this program effort in the manner indicated in Figure 5-1. The published information sources were evaluated and a sample SQA knowledge base was synthesized. Current authorities were then consulted on the adequacy of the basic information and decision rationale. Their comments and suggestions were then incorporated into the knowledge base and resubmitted to them for further comment. This iterative process was continued until the conclusion of the program effort. Since the basic information and decision rationale for SQA is constantly evolving, it is anticipated that the current SQA knowledge base may ultimately need to be updated. Procedures for accomplishing this are presented in Appendix B to this report.

5.5 DESCRIPTION OF KNOWLEDGE BASE

The knowledge base for SQA SOW preparation is organized in accordance with the process developed by SQA authorities and documented for Joint Service use. The functions of the knowledge base for SQA SOW preparation are:

- (a) Selection of Appropriate Governing Standards
- (b) Classification of Required Software by Category
- (c) Selection of Applicable Contract Data Item Descriptions
 - (1) High Level Queries
 - (2) Medium Level Queries
 - (3) Detailed Queries
- (d) Tailoring of Selected DIDs
- (e) Tailoring of Required Activities, Products and Reviews for Each Software Development Phase

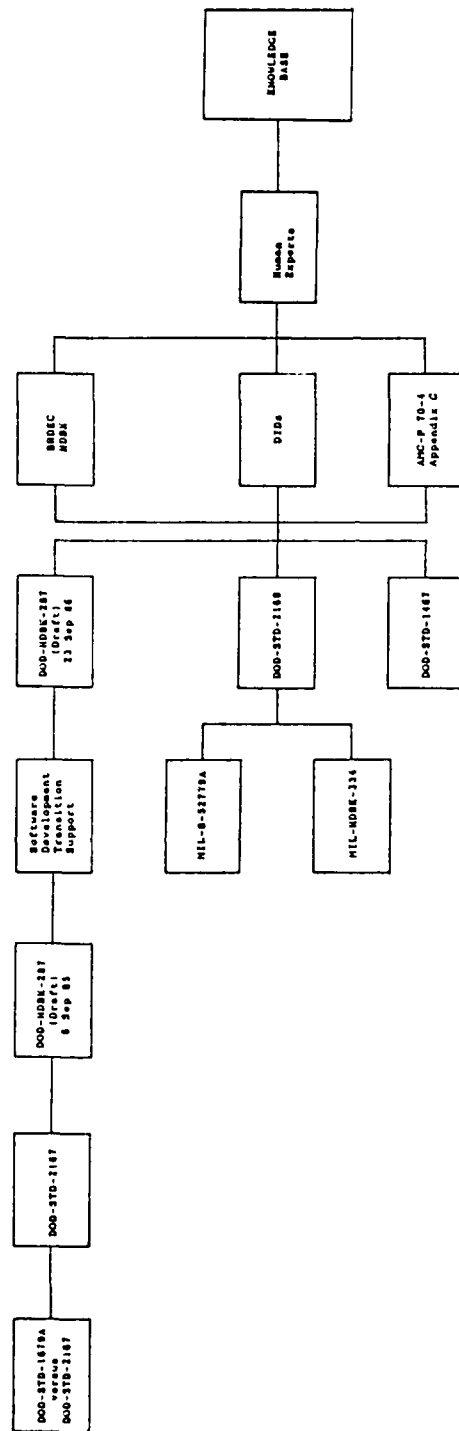


Figure 5-1 Interrelationship of Available Information

The actual basic information and decision rationale that comprise the knowledge base and perform each of these functions are contained in Appendix B to this report.

5.6 SUMMARY OF KNOWLEDGE BASE DEVELOPMENT EFFORT

The objective of the knowledge base developed for the program effort was to replicate as clearly as possible the basic information and decision rationale used by SQA experts to prepare SQA SOWs. This objective has been fully achieved. As a result, SQA SOWs can now be accurately prepared in full compliance with accepted procedures, practices, regulations and applicable guide-lines in a fraction of the time previously required.

6. ASSESSMENT OF SYSTEM CAPABILITIES

6.1 GENERAL

The development of expert systems for practical applications has come about in just the past few years and they have steadily been gaining in popularity. Part of this phenomena is due to the availability of relatively inexpensive, yet sophisticated software that enables the non-technical user to easily create models that can simulate the decision rational of human experts. This field is still in the developmental stages and new procedures are taking place almost on a day to day basis. Likewise, the expert system software is continually being changed and upgraded to keep pace with these new discoveries. Using the EXSYS development package to design our expert system, we found many benefits to this shell program, as well as some limitations. We will first examine the benefits of EXSYS.

6.2 BENEFITS OF EXSYS

Like most of the expert system shell packages, EXSYS can facilitate the development of an expert system faster and easier than using a general purpose programming language. Even programming languages specifically designed for expert system applications are not as easy to use as shell packages. When EXSYS was compared to other shell packages, here, too, we found it to be more manageable and expedient. With EXSYS, rules are constructed in a highly structured manner. The system prompts the user at each step and if a problem is encountered, help facilities are provided. This entire process can be learned quickly in a few hours.

EXSYS also has benefits directly related to our specific application. We needed a system that could create external files to store the results of each session. To achieve this, EXSYS has a built in report generator function which easily constructs external files with whatever format is designated. This capability was not included in many of the software packages we examined. In fact, many of these other packages had no procedure to preserve the results of the session. The results were merely displayed on the screen and if a hard copy was desired, the best that could be done was to use the print-screen command and send it to the printer that way.

Another positive EXSYS feature was the ability to include text notes with each rule which could be used to assist the user to better understand the questions. Although there are some problems with this procedure (which will be explained more thoroughly in the next section), this ability to help the user was cited as being very important to our client.

EXSYS also allows the user to quickly alter one or more of the responses and rerun the model while preserving the old results for comparison. In this manner, the impact of these changes can be evaluated and the sensitivity of each question on the final product can be analyzed.

If it ever becomes necessary to alter the knowledge base or reorganize the internal structure (which is quite likely since the standards and requirements change frequently within the Government), EXSYS contains an editor to facilitate this process. Modifications can be made quickly and easily without having to recreate the program from scratch.

Looking toward the future, EXSYS has a very powerful feature that enables multiple expert system programs to be linked together. This is accomplished through the use of external files. Data from one expert system can be passed to an external file which can then be used as input to a second expert system. However, although this may sound simple, it can become extremely complex. The format and sequence that the data must be in to be passed out and read back in are very rigid and the more data there is, the more difficult it is to keep track of it. Most of these difficulties will be encountered in the editing phase because the sequencing of the data will most likely be altered. Nevertheless, these problems can be overcome if the proper amount of time and care is taken.

Finally, EXSYS provides these features at a cost far below that of other comparable software packages. Additionally, the company that makes EXSYS has been very pleasant to deal with. They have been readily available for software support and they have been eager to learn about our application and suggest procedures to enhance its capabilities. But like all things, EXSYS is not perfect and we did encounter some limitations with the software. We will now look at these more closely.

6.3 LIMITATIONS OF EXSYS

A common problem among most shell packages is that they are designed with no specific application in mind, but rather they try to accommodate a broad range of uses. As a result, there are often built-in features that can actually detract from the desired performance and require additional effort to overcome. We came up against some of these obstacles and dealt with them as best we could, although some still remain. However, we recently had the

opportunity to discuss our problems with the developers of EXSYS at the Artificial Intelligence Convention held in Philadelphia, PA. Based on the meeting with us and with other users, a new, upgraded version of EXSYS will soon be released which will reduce some of these limitations.

Under normal circumstances, EXSYS will execute all of the rules in the order that it deems most important. This created quite a few problems for us as far as the structure of the rules is concerned. For one, this implies that nested IF statements cannot be used. However, we were able to overcome this by having variables serve as flags to direct the flow of questioning. Therefore, based on the response to a question, the flag variable will be set to 0 or 1, and a related question may or may not be asked depending on the value of the flag. This work-around functioned successfully but resulted in cryptic rules that can be difficult to decipher. Secondly, the order of questioning that EXSYS deems important is not necessarily the order that the user deems important. Therefore, from the user's point of view, the questions may appear in a random, incoherent sequence. Similarly, we were able to solve this problem by using variables to override this feature, but again, this distorted the appearance of the rules.

Difficulties were also encountered in trying to have help text available to the user. EXSYS does have a feature which allows text notes to be included with each of the rules. However, if the user wishes to see the text, the entire rule is displayed, as well as other rules associated with it. This causes more harm than good because the rules can be very confusing to the uneducated user. This is one of the problems we discussed with the developers of EXSYS and a solution is planned for the upcoming version. A new function

will enable external text files to be called and appear on the screen. This way, for each rule there can be an associated help text which, when a certain response is selected (presumably labeled HELP), can be viewed by the user.

Although these limitations with EXSYS required additional effort to solve, for the most part they seemed manageable and the system ultimately functioned very closely to the way it was envisioned. However, it should be kept in mind that our model was designed as a demonstrator to show what can be done with an expert system. Increasing the dimensions to a larger scale application may reveal further complications and problems.

6.4 CONCLUSIONS

The recent success of expert systems has developed because they offer a practical, meaningful solution to the problems of handling human knowledge. The ability to gather information and expertise and consolidate it in an easily accessible system allows for that knowledge to be dispersed to where it is most needed. The development of our expert system for SQA will facilitate the process of tailoring statements of work by having the knowledge of SQA engineers readily available. This will ensure that all the necessary specifications and requirements are included in the correct manner.

One of the great benefits of a consolidated knowledge base is that the real experts can spend less of their time instructing others in the correct procedures. The system is consulted instead. Additionally, the expert's knowledge has been preserved and organized so that in the event of job turnover, operations can still proceed smoothly and without interruption.

There will be no gap when someone leaves; the system will have filled it.

The expert system is also a training tool. Less experienced staff members can operate the expert system to observe and learn the proper procedures. Even those with more experience can refer to the system to refresh their skills and to keep up-to-date with any changes or modifications in the requirements.

The expert system, however, relies heavily on the knowledge base used to deduce its decision structure. The actual programming of the expert system consumes only a fraction of the effort expended; the task of data collection employs (or should employ) the majority of the resources. This process of accumulating knowledge is very difficult. Who are the true experts? What are the correct requirements and specifications? When are all avenues exhausted? These are questions that must be answered to ensure a reliable knowledge base. Even when these are satisfied, new information is constantly being found and the knowledge base must be continually updated and modified to keep pace with its dynamic environment.

Therefore, the capabilities of any expert system must be kept in their proper perspective. It can be an effective tool for decision making, for teaching, and for consolidating knowledge. But it is still only as good as the decision rules fed into it. It cannot think for itself; it must rely on its embedded knowledge base to draw conclusions. It may accomplish this much more quickly, consistently, and efficiently than most human beings, and therein lies its appeal. By automating a tedious, intricate, and error-prone job, the expert system can be a valuable tool that greatly improves the level of performance of the BRDEC SQA mission.

6.5 RECOMMENDATIONS

This SQA expert system was built with the intention of becoming a component in an overall system to encompass all aspects of the Product Assurance and Testing Directorate mission. Within the overall system, some of the components might be best implemented as expert systems, while others might take advantage of other commercial or specialized software. As a stand-alone unit, the SQA expert system has fulfilled its objective, indicating that an overall system could be beneficial and feasible. It has also demonstrated some of the real potential of expert systems. These systems are rapidly advancing in quality, flexibility, and capacity.

To achieve a comprehensive capability to automate the Product Assurance and Testing Directorate mission, it is recommended that the current incremental approach be continued. Components should be added individually and at timely intervals. This will allow the system designers to take advantage of the most recent improvements in software development rather than be committed to a package which becomes obsolete. Additionally, it will be necessary to familiarize staff engineers with the operation of the system components. This will be most effectively accomplished as an on-going process as components are added rather than trying to learn everything simultaneously.

It is also recommended that this SQA expert system for SOW development be further improved. To reiterate, the knowledge base must be continually upgraded. This can be achieved by undertaking more intensive research, specifically historical research and case studies, and by modifying the

existing package. Also, as the highly competitive marketplace of expert systems continues to develop, further enhancements to the commercial shells (including EXSYS) will be made. Full advantage should be taken of these upcoming improvements.

APPENDIX A -- INTERNAL LOGIC ANALYSIS

A.1 PARAGRAPH SELECTION CRITERIA

The end result of the expert system is a list of numbers corresponding to the paragraphs that are to be included in the Statement of Work. The status of the paragraphs is based on rules developed from the knowledge base. Each paragraph has an associated rule and selection criteria determining whether it will part of the final Statement of Work. These criteria are listed in Figure A-1.

The first column of Figure A-1 lists each paragraph number found in the master Statement of Work. An 'X' in the second column indicates that the paragraph is always included in the Statement of Work. An 'X' in the third column indicates that if the expert system determined that managerial considerations are crucial then that paragraph will be included. Likewise, the next two columns show which paragraphs will be included when system considerations are critical and when life cycle considerations are crucial. These three columns are independent of each other; therefore, as long as at least one of the criteria is satisfied, the paragraph will be included. The sixth column shows these same three criteria but indicates the combination that they must be in for a paragraph to be included. For example, 'Y N Y' means that M must equal YES (i.e., managerial considerations are crucial) and SYSCON must equal NO (i.e., system considerations are not critical) and LC must equal YES (i.e., life cycle considerations are crucial) in order for that paragraph to be included in the Statement of Work. The last column indicates the other types of selection criteria used.

Paragraph Number	Always Included	Managerial Considerations Crucial (H)	System Considerations Crucial (SYSOON)	Life Cycle Considerations Crucial (LC)	M and SYSOON and LC	Other
10.00	X					
20.00	X					
100.00	X					
200.00	X					
300.00	X					
400.00	X					
500.00	X					
600.00	X					
700.00	X					
710.00	X					
720.00	X					
730.00	X					
740.00	X					
745.00						If the support concept for the software is an agency other than the development contractor
750.00	X					
755.00						If consolidated monthly software status reports for each AND-P-702-XX task is desired
800.00	X					
900.00	X					
902.00	X					
903.00	X					
904.00		X				
905.00		X				

Figure A-1 Paragraph Selection Criteria

Paragraph Number	Always Included	Managerial Considerations Crucial (M)	System Considerations Critical (SYSOON)	Life Cycle Considerations Crucial (LC)	M and SYSOON and LC	Other
906.00						If the software requirements are broken out specifically from the system requirements AND if it is necessary for the Govt to formally review and approve what the software will do prior to its implementation
907.00			X			If the requirements include a highly involved interface with another CSCI or IMCI
908.00		X	X	X		If the design of the software needs to be documented AND if the software will be of such criticality, magnitude or complexity that the software design should be implemented, documented and reviewed as a two-step process
909.00						If the complete design of the software needs to be documented
910.00			X			
911.00			X			If the complete design of the software needs to be documented AND if more than one data base will be accessed
912.00						If some agency other than the original contractor will ever support the software
913.00	X					
914.00		X	X	X		
915.00		X	X	X		
916.00	X					
917.00		X		X		If the software, while operating in this system, requires any human interaction in the form of input, decisions to be made, or output to be evaluated

Figure A-1 Paragraph Selection Criteria (Continued)

Paragraph Number	Always Included	Material Considerations Crucial (M)	System Considerations Critical (SYSOON)	Life Cycle Considerations Crucial (LC)	M and SYSOON and LC	Other
918.00		X	X	X		If the contractor will be updating the deliverable software and distributing it to Govt agencies
919.00			X	X		If a description of the computer system's functions and user interactions into a high level, descriptive document will enhance coordination and consensus of the system concept among applicable Govt agencies
920.00			X	X		
921.00		X		X		If the computer system requires human interaction for system start up, operation, monitoring or recovery procedures
922.00		X		X		If either the computer system or associated tools provide diagnostic features to identify a malfunction and isolate a malfunctioning unit
923.00				X		If the software operating in this computer system needs to be supported by any agency other than the contractor AND if documentation is necessary to describe all of the programming features and the programming environment used in the deliverable software
924.00						If the type of firmware to be used in the system is PROM, EPROM or a combination AND if life cycle considerations are crucial OR If the software is embedded in the firmware AND if the firmware is supported by any agency other than the contractor
930.00	X					
950.00	X					

Figure A-1 Paragraph Selection Criteria (Continued)

Paragraph Number	Always Included	Managerial Considerations Critical (M)	System Considerations Critical (SYSDON)	Life Cycle Considerations Critical (LC)	M and SYSDON and LC	Other
960.00					Y Y N	
1000.00	X					
1001.00	X					
1002.00	X					
1003.00		X	X	X		
1004.00		X	X	X		
1010.00						If the software is deliverable software to be developed and designated as part of a system or hardware Configuration Item
1020.00						If the software is non-deliverable software
1030.00						If the software is unmodified commercially available and reusable software used in a deliverable (SCI or IMCI)
1040.00	X					
1050.00	X					
1060.00					Y Y N	
1070.00					Y N Y	
1080.00					Y N N	
1090.00					N Y Y	
1100.00		X				
1110.00					Y N Y	
1120.00					Y N N	
1130.00					Y Y N	
1200.00		X				
1210.00					Y N N	

Figure A-1 Paragraph Selection Criteria (Continued)

Paragraph Number	Always Included	Managerial Considerations Crucial (H)	System Considerations Critical (SYSXN)	Life Cycle Considerations Crucial (LC)	M and SYSXN and LC	Other
1300.00						If the software requirements are broken out specifically from the system requirements AND if it is necessary for the Govt to formally review and approve what the software will do prior to its implementation
1310.00					Y N Y	
1400.00			X			If the requirements include a highly involved interface with another CSC1 or IMCI
1410.00						If the complete design of the software needs to be documented
1415.00					N Y Y	
1420.00					N Y N	
1500.00		X	X	X		If the design of the software needs to be documented AND if the software will be of such criticality, magnitude or complexity that the software design should be implemented, documented and reviewed as a two-step process
1510.00					Y Y Y N	
1520.00					Y N Y	
1600.00						If the complete design of the software needs to be documented
1610.00					N Y N	
1700.00			X			
1710.00					N Y N	
1800.00			X			If the complete design of the software needs to be documented AND if more than one data base will be accessed

Figure A-1 Paragraph Selection Criteria (Continued)

Paragraph Number	Always Included	Managerial Considerations Critical (M)	System Considerations Critical (SYSOON)	Life Cycle Considerations Critical (LC)	M and SYSOON and LC	Other
1810.00					N Y Y	
1900.00						If some agency other than the original contractor will ever support the software
2000.00	X					
2010.00					N Y N	
2020.00					Y N N Y	
2030.00					N Y Y	
2040.00					Y N N N	
2100.00			X			
2200.00			X			
2300.00	X					
2400.00		X		X		
2410.00					Y N N Y	
2500.00		X	X	X		If the contractor will be updating the deliverable software and distributing it to Govt agencies
2600.00			X	X		If a description of the computer system's functions and user interactions into a high level, descriptive document will enhance coordination and consensus of the system concept among applicable Govt agencies
2610.00					Y N N Y	
2700.00			X	X		
2800.00	X					
3000.00	X					

Figure A-1 Paragraph Selection Criteria (Continued)

Paragraph Number	Always Included	Managerial Considerations Crucial (M)	System Considerations Crucial (SYSTEM)	Life Cycle Considerations Crucial (LC)	M and SYSTEM and LC	Other
3200.00						If System Considerations are critical AND if Life Cycle Considerations are crucial
3300.00	X					
3400.00	X					
3500.00		X		X		
3600.00		X		X		
3700.00	X					
3800.00			X			
3900.00	X					
4000.00		X	X	X		If there are anticipated upgrades to the software OR if there are preplanned product improvements anticipated
4100.00						
4200.00	X					
4300.00	X					
4600.00				X		If the software operating in this computer system needs to be supported by any agency other than the contractor AND if documentation is necessary to describe all of the programming features and the programming environment used in the deliverable software
4610.00					N N Y	
4700.00		X		X		If the computer system requires human interaction for system start up, operation, monitoring or recovery procedures
4710.00					Y N Y	

Figure A-1 Paragraph Selection Criteria (Continued)

Paragraph Number	Always Included	Managerial Considerations Crucial (M)	System Considerations Crucial (SYSUN)	Life Cycle Considerations Crucial (LC)	M and SYSUN and LC	Other
4800.00		X		X		If either the computer system or associated tools provide diagnostic features to identify a malfunction and isolate a malfunctioning unit
4810.00					N Y Y	
4820.00					N N Y	
4900.00						If the type of firmware to be used in the system is ROM, EPROM or a combination AND if life cycle considerations are crucial OR If the software is embedded in the firmware AND if the firmware is supported by any agency other than the contractor
4910.00						If paragraph # 924.00 and 4900.00 are included and SYSUN=NO, M=NO and LC=YES
5200.00			X	X		
5300.00	X					
5301.00	X					
5302.00	X					
5303.00	X					
5304.00		X				
5305.00	X					
5306.00			X			
5307.00			X			
5308.00	X					
5309.00			X			
5310.00	X					
5311.00	X					

Figure A-1 Paragraph Selection Criteria (Continued)

Paragraph Number	Always Included	Managerial Considerations Crucial (M)	System Considerations Critical (SYSOON)	Life Cycle Considerations Crucial (LC)	M and SYSOON and LC	Other
5312.00	X					
5313.00	X					
5314.00	X					
5400.00	X					
5500.00	X					
5600.00	X					
5700.00	X					
5710.00		X				
5720.00	X					
5730.00	X					
5740.00			X			
5750.00	X					
5760.00			X			
5765.00			X			
5770.00		X	X	X		
5780.00	X					
5800.00		X				
5900.00	X					
5910.00	X					
5920.00	X					
5930.00			X			
5940.00	X					
5950.00	X					
6000.00			X			

Figure A-1 Paragraph Selection Criteria (Continued)

Paragraph Number	Always Included	Managerial Considerations Crucial (M)	System Considerations Critical (SYSCON)	Life Cycle Considerations Crucial (LC)	M and SYSCON and LC	Other
6100.00			X			
6200.00	X					
6300.00			X			
6400.00	X					
6500.00	X					
6600.00	X					
6700.00	X					
7000.00						If there is to be an interface with independent verification and validation activities
7100.00						If design walkthroughs are required during the design phase
7200.00						If design inspection is required during the design phase
7300.00						If source code analysis is required during the coding phase
7400.00						If code walkthroughs are required during the coding phase
7500.00						If code inspection is required during the coding phase
7600.00						If software stress testing is required during the testing phase
7700.00	X					

Figure A-1 Paragraph Selection Criteria (Continued)

A.2 MULTI-LEVEL QUESTIONS

The central core of the expert system consists of questions broken down into levels. There are three Level 1 questions that must be answered. For each of these, if more information is needed to answer the question then the Level 2 questions are asked. (If the Level 1 question can be answered, the associated Level 2 questions will not be asked.) The answers to the Level 2 questions will be combined to deduce the Level 1 answer. Similarly, if more information is needed for the Level 2 questions then the Level 3 questions will be asked. The logic used to direct the flow of these questions is displayed in Figures A-2, A-3, and A-4 for the questions on managerial consideration, system consideration, and life cycle consideration, respectively.

As shown in these diagrams, the first box around the Level 1 question shows the possible responses and results of each selection. For example, in Figure A-2, if managerial considerations are crucial then the variable 'M' is assigned the value 'YES'. If more information is needed then the Level 2 questions are asked, as indicated by the arrow.

The Level 2 box encompasses all the Level 2 questions and their associated responses and results. If more information is needed for any of these questions, the corresponding Level 3 questions will be invoked, as shown by the arrows. The equations at the bottom of the box display the logic used in combining the preceding Level 2 responses to deduce the Level 1 answer.

The Level 3 boxes indicate the results of the answers to the Level 3 questions. For all the Level 3 questions, the responses are either YES, NO, or DON'T KNOW. Then, for each question there is a variable that will be assigned a 0 or 1 depending on which response is chosen. These variables are then added together to deduce the Level 2 response.

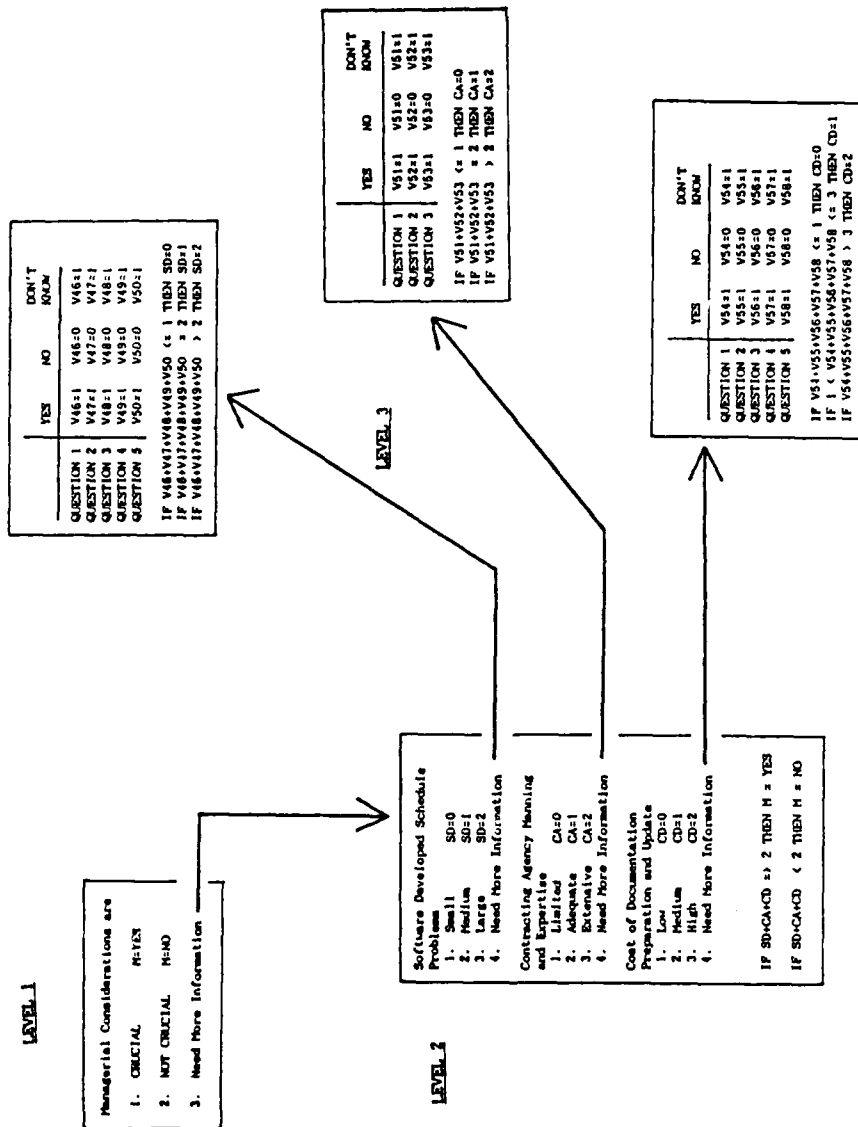


Figure A-2 Multi-Level Questions for Managerial Considerations

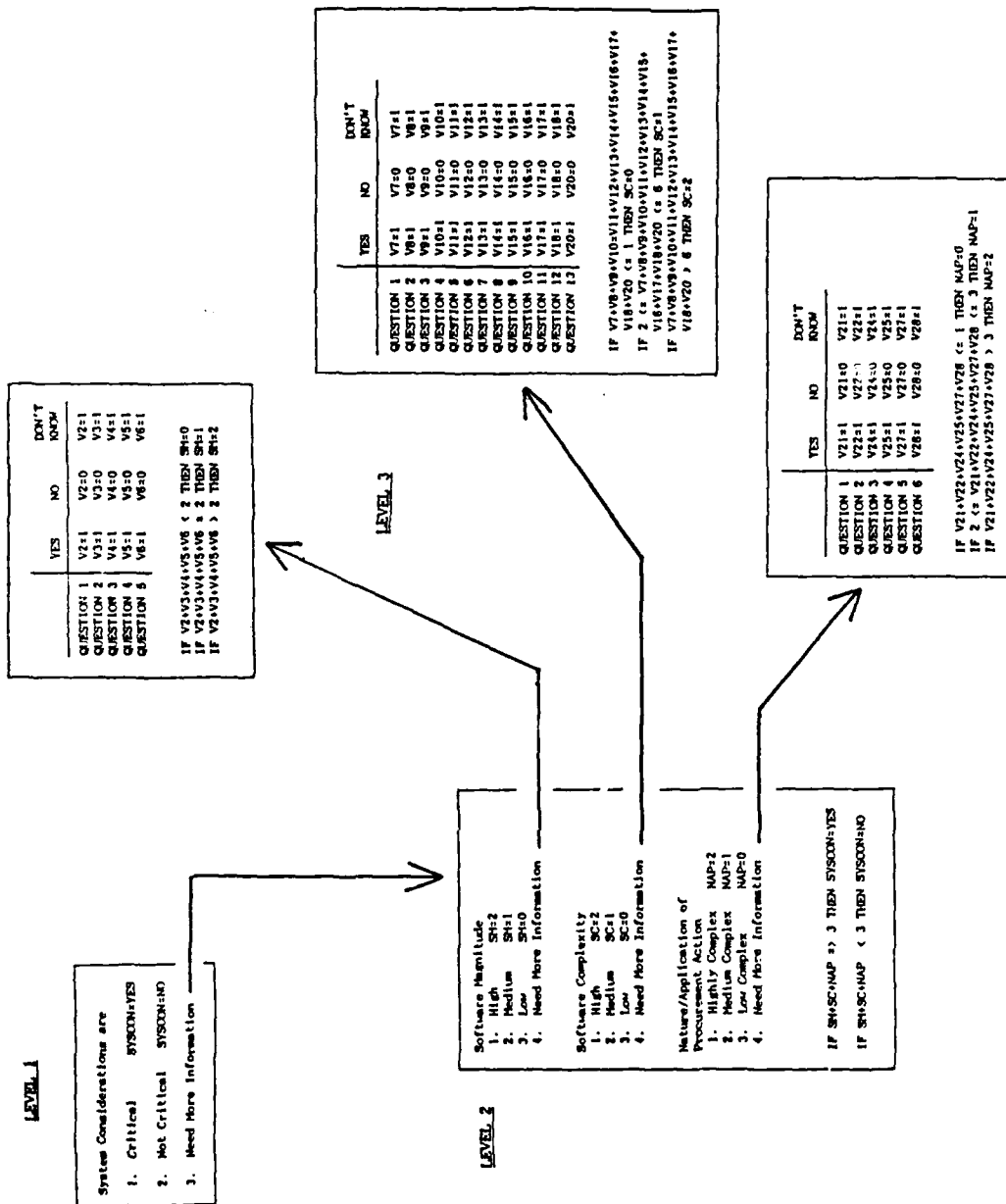


Figure A-3 Multi-Level Questions for System Considerations

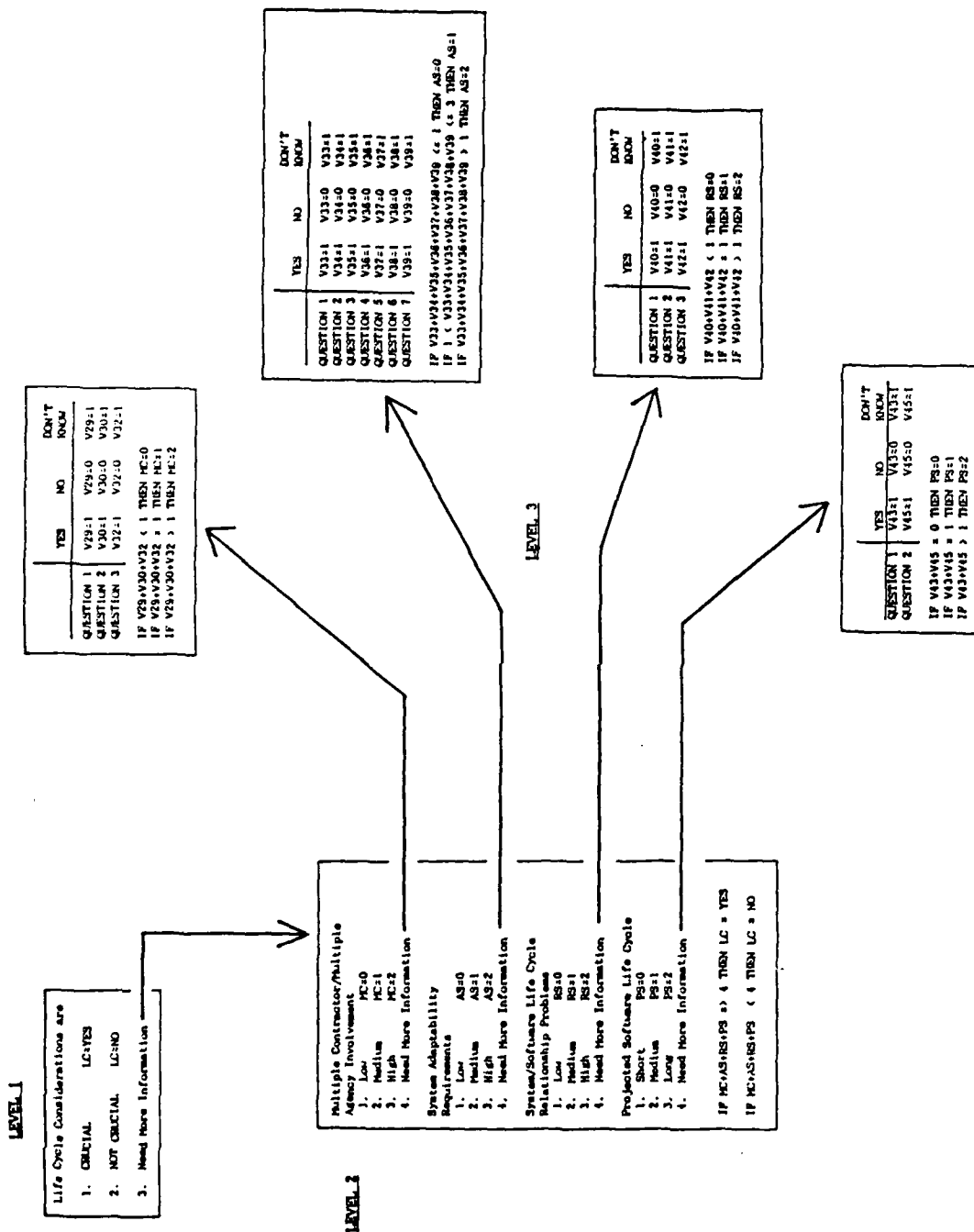


Figure A-4 Multi-Level Questions for Life Cycle Considerations

A.3 THE "DON'T KNOW" RESPONSES

As stated in the previous section, the responses for all the Level 3 questions are either YES, NO, or DON'T KNOW. Currently the expert system only keeps track of the number of DON'T KNOW responses and treats them as if they were YES's. This assumes that if you don't know if a paragraph should be included in the Statement of Work, it will be included anyway to play it safe. In the future, however, the DON'T KNOWs could be further incorporated into the expert system by adding the number of DON'T KNOWs and informing the user whether he knows enough to receive a meaningful SOW. Other logical changes could be made to treat DON'T KNOW responses differently from YES's.

In a straightforward approach, the number of DON'T KNOWs could be made to alter the impact of the Level 3 equations used to deduce the Level 2 answers. For example, if the sum of five Level 3 variables has to be greater than 3 (the threshold value) in order for a Level 2 variable to be assigned a certain value, what should happen when four of the five questions are answered with DON'T KNOW? Logically, the conclusion should depend somewhat on whether the fifth Level 3 question was answered Yes or No. If they are treated the same as YES's, the result may not properly reflect the operator's actual response. Therefore, it is necessary to include some sort of rule to manage the impact of DON'T KNOW responses.

There are several few techniques that could be used in this situation ranging from a "rule of thumb" to a very complicated, sophisticated mathematical model. In the context of this application, we recommend a simple, understandable approach.

The threshold value could be modified depending on what percent of the questions were answered with a definitive answer (i.e., YES or NO). Therefore, if all the questions are answered then the threshold remains the same (i.e., 100% X Threshold), but if only four out of five questions are answered then the threshold will be lowered (i.e., 80% X Threshold). Of course, if none of the questions are answered (or when a very low percentage has been answered) the results will still have very little relevance. It is under these circumstances that the warning messages informing the user that he does not have enough information should be invoked.

A.4 SOW FILE RATIONALE

In general, the master SOW file conforms to the content and wording of the most complete SOW template in the BRDEC Handbook, "Contracting for Computer Software Development, Software Quality Evaluation and Independent Software Verification and Validation," Report 2431, January 1986. Our research into the applicable directives and guidelines has identified a number of additional paragraphs which allow the master SOW file to be specifically tailored (either using the DOD-HDBK-287 or the the BRDEC Handbook methodologies).

The following table describes the rationale for each paragraph in the SOW file which is not included in the most severe template (#5) of the BRDEC Handbook. Most of the differences are paragraphs which exclude certain sections of cited DIDs from applicability. These exclusions were for the most part obtained from the draft DOD-HDBK-287 of 6 September 1985 which explained how to tailor the DIDs. Other differences reflect the revised wording of the less stringent templates (#1 through #4) of the BRDEC Handbook.

PARAGRAPHS IN SOW FILE NOT IN TEMPLATE #5

PARAGRAPH NUMBER	EXCLUDED IAW DOD HDBK 287	OTHER RATIONALE/DISCUSSION
745-755		Consolidation of individual reports into a single monthly Technical Operating Report for Software Status as directed by BRDEC.
902, 921-924		Common sense additions to list of prescribed software development program documents.
950		System Segment Specification IAW DOD-HDBK-287 p. 20.
960	p 85	SSS
1010-1030		Software Development Plan paragraphs IAW DOD-HDBK-287, p.27-28.
1060-1090	p 89-90	SDP
1110-1130	p 85	SCMP
1210	p 85	SQEP
1310	p 88	SRS
1410 1415, 1420	p 89	DOD-STD-2167 5.3.2.4 IRS
1510, 1520	p 86	STLDD
1610	p 90	SDDD
1710	p 89	IDD
1810	p 89	DBDD
2010-2040	p 86	STP
2410	p 87	SUM
2610	p 88	OCD

(continued)

3010		Exclusions from SDR for Templates #1 through #4. Not referenced in the Expert System. Included as directed by BRDEC.
4610	p 88	SPM
4710	p 87	CSOM
4810, 4820	p 87	CSDM
4910	p 88	FSM
8000-8630		Software Quality Evaluation paragraphs for Templates #1 and #2, which differ markedly from similar paragraphs in Templates #3, #4, and #5 (@5300 through @7700). Not referenced by the Expert System. Included as directed by BRDEC.

APPENDIX B -- MAINTENANCE MANUAL

EXSYS is a generalized expert system development package which makes decisions based on logical rules. The rules are structured in an IF-THEN-ELSE type format. The IF part contains one or more statements which can be sentences or mathematical expressions, and if they are true, the THEN part of the rule is executed. This could include selecting a choice, assigning a value to a variable and/or solving a mathematical expression. The ELSE part of the rule is optional and is executed only if the IF part is false. These rules are used to select among choices, which are typically the possible solutions to the problem. However, in our application, the choices serve as flags for directing the logical sequencing of the rules.

In EXSYS, the choices are what drive the system. The first step in building the expert system is to define the choices and then use them in constructing the rules. If a rule is written which does not address one of the choices, the question relating to that rule will not be asked since its answer will not help in selecting a choice (as far as EXSYS is concerned). In our application, the choices we want to make are what sections to include in the Statement of Work. Currently there are about 100 sections, but the full model could contain thousands. EXSYS can only accomodate thirty choices, obviously not enough for us. Consequently, we used variables to determine whether or not a section is to be included, eliminating the need for EXSYS choices. However, if there are no choices to choose among, EXSYS thinks that there is no problem to solve and it will not ask any questions. Therefore, we had to use "dummy" choices to cause the questions to be asked, and they are also used to ensure that the questions are asked in the proper sequence.

Certain conventions are followed in constructing the rules so that it will be easier to tell what is going on. First, all EXSYS choices are designated C1, C2, C3, ... , C20. Second, all variables that are used to determine if a paragraph is to be included in the statement of work begin with the letter "P" (i.e., P906, P1200, etc.). Additionally, other variables used to direct the flow of questioning begin with the letter "V". Other variable names will be described later.

Changes or modifications of the rules are accomplished by using the edit mode of EXSYS. The editor is contained in the program called EDITXS. To initiate the edit mode, simply enter EDITXS at the prompt and then enter the name of the file that is to be modified.

The following gives a detailed description of all the questions the expert system asks and how their corresponding rules are constructed.

The application is (Please enter numerical choice and hit RETURN)

- 1 statement of work preparation
- 2 RFP/RFQ preparation
- 3 source evaluation/selection
- 4 contract negotiation
- 5 in process review preparation
- 6 administrative monitoring of contract effort
- 7 technical monitoring of contract effort
- 8 product inspection/acceptance
- 9 independent verification/validation

At this time, only choice 1 is operational. Therefore, if the user selects any of the other choices, the question will be asked again, but with a warning message stating that only choice 1 can be selected. If the user still does not select choice 1, the system will assume he did and go on to the next question.

Rules 1 and 2 correspond to this question. Rule 1 is structured so that if anything but statement of work preparation is selected, C1 will be given a value of zero. Rule 2 will be invoked only when C1=0 and it will re-ask the question. If statement of work preparation is selected initially, rule 2 will be by-passed since C1 will not be equal to zero (it's value would be unassigned). Notice that Rule 2 has the statement C1=1 in its THEN section. This is the first instance of a "dummy" choice. It is used only to ensure that the question will be asked if needed. From here on out, whenever C1=1 is used, it will be used as a "dummy."

CONGRATULATIONS --- You have asked this expert system to help you quickly tailor a statement of work in strict accordance with applicable DOD standards and directives for software quality assurance. This process should take less than 5 minutes. Without this expert system, the process could take many days, and perhaps weeks.

- 1 Enter 1 and return to continue

Rule 3 corresponds to this question.

The software is

- 1 Mission Critical Computer Software (MCCS)
- 2 other than mission critical computer software

Rule 4 corresponds to this question. If the software is mission critical then C2=0; if not C2=1. C2 is used as a flag because if the software is other than mission critical, this requires another question to be asked (this is explained further with the next question). Notice also that the NOTE and REFERENCE sections of the rule contain information. This is to help the user answer the question.

Only DOD-STD-2167 is currently implemented in this Expert System. It may be applied to other than Mission Critical Computer Software, but it may result in unnecessary over-specification of the SQA tasks. What do you want to do?

- 1 Apply DOD-STD-2167
- 2 Exit the Expert System

This question is asked only if the answer to the previous question was other than mission critical computer software, that is, if C2=1. The user is getting the chance to exit the system if he did not want to apply DOD-STD-2167. Rule 5 corresponds to this question. If choice 1 is selected, the system continues. If choice 2 is selected then the expert system ends at this point and the variable END is given the value of -999. This will be used later by the READER program in order to abort the batch file.

The software is (Please indicate numerical choice(s) separated by commas and hit RETURN)

- 1 Deliverable software to be developed and designated as Computer Software Configuration Item (CSCI)
- 2 Deliverable software to be developed and designated as part of a System or Hardware Configuration Item (HWCI)
- 3 Non deliverable software
- 4 Unmodified commercially available and reusable software used in a deliverable CSCI or HWCI
- 5 Previously developed software designated as a CSCI undergoing modification

Rules 6, 7, 8 and 9 correspond to this question. Choices 2, 3, and 4 each correspond to particular paragraphs in the Statement of Work. Rule 6 states that if choice 4 is selected then paragraph 1030.00 will be included. Rule 7 states that if choice 2 is selected then paragraph 1010.00 will be included. Rule 8 states that if choice 3 is selected then paragraph 1020.00 will be included. Additionally, if only choice 4 is selected, another question will be asked. Choice 4 deals with commercial software, and so DOD-STD-2167 may be inappropriate. If this is the case, the user will be given the chance to exit the expert system, or else continue. Rule 9 deals with situation.

** LEVEL 1 ** System considerations are

- 1 critical
- 2 not critical
- 3 NEED MORE INFORMATION

This is a level 1 question. If choice 1 or 2 is selected, the program will go on to another level 1 question. However, if more information is needed, the program will go down to a series of level 2 questions which will determine the answer to this question. If at the second level, the user still needs more information, the program will step down to a series of level 3 questions which will determine the level 2 answer.

Rules 10, 11 and 12 correspond to this question. Rule 10 states that if system considerations are critical then the variable SYSCON will be assigned the value "YES". Additionally, C4 is given the value of zero and will be used as a flag to direct the flow of questioning if it is necessary to step down through the different levels of questions. Rule 11 states that if system considerations are not critical then SYSCON is given the value "NO" and C4 is again given the value zero. Rule 12 states that if more information is needed then C4 will equal 1, or else it will be zero. This is because level 2 questions concerning system considerations will only be asked if C4=1.

* LEVEL 2 * Software magnitude is
1 high
2 medium
3 low
4 NEED MORE INFORMATION

Rules 13, 14, 15 and 16 correspond to this question. Rule 13 states that if the software magnitude is high then the variable SM=2. C5 is used as a flag if it is necessary to go down to level three. Rule 14 states that if software magnitude is medium then SM=1 and C5=0. Rule 15 states that if it is low then SM=0 and C5=0. If more information is needed, Rule 16 assigns the value 1 to C5, else C5=0. All these rules are invoked only if C4=1, that is, if more information was needed to answer the level 1 question about sytem considerations.

- LEVEL 3 - Will the number of unique functions apparent to the user and to be accomplished in the software be greater than 100?
1 YES
2 NO
3 DON'T KNOW

This is a level 3 question and they are all handled about the same. If the answer to a level 3 question is YES then a variable (V1, V2, ... , Vn) is assigned the value 1. If the answer is NO, that variable is given a value of 0. If the answer is DON'T KNOW, then another variable (NK1, NK2, ... , NKn) is given the value 1. In the future, the DON'T KNOW answers will be totaled up and if there are too many of them, a warning message will be displayed to user telling him he does not know enough to obtain a meaningful solution. However, at present, nothing is done with these variables and they have the same impact as answering YES.

This group of level 3 questions will be asked only if C4=1 and C5=1 and relate to the level 2 question on software magnitude.

The rules that apply to this question are Rules 17, 18 and 19 and they state the following:

Rule 17 -- If the answer is YES then V2=1
Rule 18 -- If the answer is NO then V2=0
Rule 19 -- If the answer is DON'T KNOW then NK2=1 and V2=1

- LEVEL 3 - Will there be more than one operational site that requires unique software configurations?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 20, 21 and 22 and they state the following:

- Rule 20 -- If the answer is YES then V3=1
- Rule 21 -- If the answer is NO then V3=0
- Rule 22 -- If the answer is DON'T KNOW then NK3=1 and V3=1

- LEVEL 3 - Will the software execute in a multicomputer system environment?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 23, 24 and 25 and they state the following:

- Rule 23 -- If the answer is YES then V4=1
- Rule 24 -- If the answer is NO then V4=0
- Rule 25 -- If the answer is DON'T KNOW then NK4=1 and V4=1

- LEVEL 3 - Are there more than 3 Computer Software Configuration Items (CSCIs) envisioned for the software system?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 26, 27 and 28 and they state the following:

- Rule 26 -- If the answer is YES then V5=1
- Rule 27 -- If the answer is NO then V5=0
- Rule 28 -- If the answer is DON'T KNOW then NK5=1 and V5=1

- LEVEL 3 - Is the estimated number of newly developed lines of High Order Language (HOL) code greater than 700,000?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 29, 30 and 31 and they state the following:

Rule 29 -- If the answer is YES then V6=1
Rule 30 -- If the answer is NO then V6=0
Rule 31 -- If the answer is DON'T KNOW then NK6=1 and V6=1

At this point, there is enough information from the level 3 questions to determine the software magnitude, which is the level 2 question we were trying to answer. Rules 32, 33 and 34 deal with this subject. These rules are invoked only when C4=1 and C5=1, indicating that more information was needed to answer both the level 1 and level 2 questions. The evaluation is based on the number of YES answers to the level 3 questions determined by the value of the variables (V1,V2, ...). The rules state the following:

Rule 32 -- If $V2+V3+V4+V5+V6 < 2$ then SM=0
Rule 33 -- If $V2+V3+V4+V5+V6 = 2$ then SM=1
Rule 34 -- If $V2+V3+V4+V5+V6 > 2$ then SM=2

The value for SM found here will be used later to help determine the value for system considerations, the level 1 question.

Rule 35 then evaluates the number of DON'T KNOW responses for the questions about software magnitude and if there are too many, a warning message will be called and appear on the screen. Variables are used to count up these responses (they all start with NK) and a threshold value is determined. At this point, the user can either exit the expert system or continue on. Rule 35 states the following:

If $NK2+NK3+NK4+NK5+NK6 \geq 3$
and the warning message is answered with "Exit the Expert System"

Then END = -999
STOP

The variable END is used later by READER to indicate that further processing by the batch file is not needed (since the user is exiting the system). STOP is an EXSYS variable telling the expert system not to continue asking any more questions.

* LEVEL 2 * Software complexity is

- 1 high
- 2 medium
- 3 low
- 4 NEED MORE INFORMATION

Back to a level 2 question, which relates to the level 1 question on system considerations and is only asked if C4=1. Its rules are structured in the same way as the previous level 2 question. The corresponding rules are 36, 37, 38 and 39. Rule 36 states that if software complexity is high then SC=2 and C6=0. Rule 37 states that if it is medium then SC=1 and C6=0. Rule 38 says that if it is low then SC=0 and C6=0. Rule 39 states that if more information is needed then C6=1, else C6=0.

The next set of level 3 questions are asked only if C4=1 and C6=1 and relate to the level 2 question on software complexity.

- LEVEL 3 - Will the software use intricate or complicated computational algorithms?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 40, 41 and 42 and they state the following:

- Rule 40 -- If the answer is YES then V7=1
Rule 41 -- If the answer is NO then V7=0
Rule 42 -- If the answer is DON'T KNOW then NK7=1 and V7=1

- LEVEL 3 - Will the software use computational algorithms or interfaces that apply or define the state-of-the-art?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 43, 44 and 45 and they state the following:

- Rule 43 -- If the answer is YES then V8=1
Rule 44 -- If the answer is NO then V8=0
Rule 45 -- If the answer is DON'T KNOW then NK8=1 and V8=1

- LEVEL 3 - Are there demanding throughput and/or sizing/timing (memory and real-time control) constraint requirements?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 46, 47 and 48 and they state the following:

- Rule 46 -- If the answer is YES then V9=1
- Rule 47 -- If the answer is NO then V9=0
- Rule 48 -- If the answer is DON'T KNOW then NK9=1 and V9=1

- LEVEL 3 - Will the software perform functions critical to the mission operations?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 49, 50 and 51 and they state the following:

- Rule 49 -- If the answer is YES then V10=1
- Rule 50 -- If the answer is NO then V10=0
- Rule 51 -- If the answer is DON'T KNOW then NK10=1 and V10=1

- LEVEL 3 - Will the software, while operating in this computer system, require any human interaction in the form of input, decisions to be made, or output to be evaluated?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 52, 53 and 54 and they state the following:

- Rule 52 -- If the answer is YES then V11=1; additionally, paragraph 917.00 will be included, thus P917=917.00
- Rule 53 -- If the answer is NO then V11=0
- Rule 54 -- If the answer is DON'T KNOW then NK11=1 and V11=1

- LEVEL 3 - Will the software have demanding primary (main computer memory) and secondary (tape, disk, etc.) storage requirements?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 55, 56 and 57 and they state the following:

- Rule 55 -- If the answer is YES then V12=1
- Rule 56 -- If the answer is NO then V12=0
- Rule 57 -- If the answer is DON'T KNOW then NK12=1 and V12=1

- LEVEL 3 - Will the software be required to support a large number or variety of peripheral equipment items?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 58, 59 and 60 and they state the following:

- Rule 58 -- If the answer is YES then V13=1
- Rule 59 -- If the answer is NO then V13=0
- Rule 60 -- If the answer is DON'T KNOW then NK13=1 and V13=1

- LEVEL 3 - Will the software have demanding multiprocessing, multiprogramming, or distributed processing requirements?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 61, 62 and 63 and they state the following:

- Rule 61 -- If the answer is YES then V14=1
- Rule 62 -- If the answer is NO then V14=0
- Rule 63 -- If the answer is DON'T KNOW then NK14=1 and V14=1

- LEVEL 3 - Will the software access more than one data base?

- 1 YES
- 2 NO
- 3 DON'T KNOW

If this question is answered YES then another question concerning data bases will be asked, therefore the rules will be a little different. The rules that apply to this question are Rules 64, 65 and 66 and they state the following:

Rule 64 -- If the answer is YES then V15=1; additionally, C18=1 and will be used as a flag to determine if the next question will be asked or not

Rule 65 -- If the answer is NO then V15=0

Rule 66 -- If the answer is DON'T KNOW then NK15=1 and V15=1

- LEVEL 3 - Will any of the data bases share common information?

- 1 YES
- 2 NO
- 3 DON'T KNOW

This question will only be asked if C18=1, determined from the preceding question. The rules that apply to this question are Rules 67, 68 and 69 and they state the following:

Rule 67 -- If the answer is YES then V16=1

Rule 68 -- If the answer is NO then V16=0

Rule 69 -- If the answer is DON'T KNOW then NK16=1 and V16=1

- LEVEL 3 - Will the data base(s) be primarily dynamic in nature, with changes being made internally and/or externally?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 70, 71 and 72 and they state the following:

Rule 70 -- If the answer is YES then V17=1

Rule 71 -- If the answer is NO then V17=0

Rule 72 -- If the answer is DON'T KNOW then NK17=1 and V17=1

- LEVEL 3 - Will the software be required to perform software failure detection, software fault isolation, or software recovery initiation?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 73, 74 and 75 and they state the following:

- Rule 73 -- If the answer is YES then V18=1
- Rule 74 -- If the answer is NO then V18=0
- Rule 75 -- If the answer is DON'T KNOW then NK18=1 and V18=1

- LEVEL 3 - Will the software be required to operate a large number of interfaces or complex interfaces with other systems or equipment?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 76, 77 and 78 and they state the following:

- Rule 76 -- If the answer is YES then V20=1
- Rule 77 -- If the answer is NO then V20=0
- Rule 78 -- If the answer is DON'T KNOW then NK20=1 and V20=1

At this point, there is enough information from the level 3 questions to determine the software complexity, which is the level 2 question we were trying to answer. Rules 79, 80 and 81 deal with this subject. These rules are invoked only when C4=1 and C6=1, indicating that more information was needed to answer both the level 1 and level 2 questions. The evaluation is based on the number of YES answers to the level 3 questions determined by the value of the variables (V1,V2,...). The rules state the following:

- Rule 79 -- If $V7+V8+V9+V10+V11+V12+V13+V14+V15+V16+V17+V18+V20 \leq 1$
then SC=0
- Rule 80 -- If $2 \leq V7+V8+V9+V10+V11+V12+V13+V14+V15+V16+V17+V18+V20 \leq 6$
then SC=1
- Rule 81 -- If $V7+V8+V9+V10+V11+V12+V13+V14+V15+V16+V17+V18+V20 > 6$
then SC=2

The value for SC found here will be used later to help determine the value for system considerations, the level 1 question.

Rule 82 then evaluates the number of DON'T KNOW responses for the questions about software complexity and if there are too many, a warning message will be called and appear on the screen. Variables are used to count up these responses (they all start with NK) and a threshold value is determined. At this point, the user can either exit the expert system or continue on. Rule 82 states the following:

If $NK7+NK8+NK9+NK10+NK11+NK12+NK13+NK14+NK15+NK16+NK17+NK18+NK20 \geq 7$
and the warning message is answered with "Exit the Expert System"

Then END = -999
 STOP

The variable END is used later by READER to indicate that further processing by the batch file is not needed (since the user is exiting the system). STOP is an EXSYS variable telling the expert system not to continue asking any more questions.

* LEVEL 2 * The nature/application of procurement action is
1 highly complex
2 medium complex
3 low complex
4 NEED MORE INFORMATION

Rules 83, 84, 85 and 86 correspond to this question. These rules are invoked only if C4=1 which means that more information was needed to the level 1 question about system considerations. Rule 83 states that if the nature/application of the procurement action is highly complex then the variable NAP=2 and that C7=0. Rule 84 states that if complexity is medium then NAP=1 and C7=0. Rule 85 states that if complexity is low then NAP=0 and C7=0. Finally, Rule 86 states that if more information is needed then C7=1, else C7=0. This variable directs the flow of questioning to level 3 if C7=1, which occurs when more information is needed.

The next set of level 3 questions are asked only if C4=1 and C7=1 and relate to the level 2 question on the nature/application of the procurement action.

- LEVEL 3 - Do the requirements for the software dictate software research and development because there are no similar systems?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 87, 88 and 89 and they state the following:

Rule 87 -- If the answer is YES then V21=1
Rule 88 -- If the answer is NO then V21=0
Rule 89 -- If the answer is DON'T KNOW then NK21=1 and V21=1

- LEVEL 3 - Will the software be executed in real time (as opposed to batch)?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 90, 91 and 92 and they state the following:

Rule 90 -- If the answer is YES then V22=1
Rule 91 -- If the answer is NO then V22=0
Rule 92 -- If the answer is DON'T KNOW then NK22=1 and V22=1

- LEVEL 3 - Will this software be used directly in the operation of mission-critical systems?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 93, 94 and 95 and they state the following:

Rule 93 -- If the answer is YES then V24=1
Rule 94 -- If the answer is NO then V24=0
Rule 95 -- If the answer is DON'T KNOW then NK24=1 and V24=1

- LEVEL 3 - Are the software requirements such that existing software and documentation cannot be used?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 96, 97 and 98 and they state the following:

- Rule 96 -- If the answer is YES then V25=1
- Rule 97 -- If the answer is NO then V25=0
- Rule 98 -- If the answer is DON'T KNOW then NK25=1 and V25=1

- LEVEL 3 - Are there anticipated upgrades to the software?

- 1 YES
- 2 NO
- 3 DON'T KNOW

With this question, the format of the rules for the level 3 questions changes slightly. Instead of having two separate rules for the YES and NO answers, only one will be necessary, as can be seen below. The variables will be initialized to zero so there will be no need for the rules about the NO answers. This new format will be used throughout the rest of the rule base. The rules that apply to this question are Rules 99 and 100 and they state the following:

- Rule 99 -- If the answer is YES then V27=1
- Rule 100 -- If the answer is DON'T KNOW then NK27=1 and V27=1

- LEVEL 3 - Will there be security classification requirements for the software?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 101 and 102 and they state the following:

- Rule 101 -- If the answer is YES then V28=1
- Rule 102 -- If the answer is DON'T KNOW then NK28=1 and V28=1

At this point, there is enough information from the level 3 questions to determine the nature/application of the procurement action, which is the level 2 question we were trying to answer. Rules 103, 104 and 105 deal with this subject. These rules are invoked only when C4=1 and C7=1, indicating that more information was needed to answer both the level 1 and level 2 questions. The evaluation is based on the number of YES answers to the level 3 questions determined by the value of the variables (V1,V2,...). The rules state the following:

Rule 103 -- If $V21+V22+V24+V25+V27+V28 \leq 1$ then $NAP=0$
Rule 104 -- If $2 \leq V21+V22+V24+V25+V27+V28 \leq 3$ then $NAP=1$
Rule 105 -- If $V21+V22+V24+V25+V27+V28 > 3$ then $NAP=2$

The value for NAP found here will be used next to help determine the value for system considerations, the level 1 question.

Rule 106 then evaluates the number of DON'T KNOW responses for the questions about the nature/application of the procurement action and if there are too many, a warning message will be called and appear on the screen. Variables are used to count up these responses (they all start with NK) and a threshold value is determined. At this point, the user can either exit the expert system or continue on. Rule 106 states the following:

If $NK21+NK22+NK24+NK25+NK27+NK28 \geq 3$
and the warning message is answered with "Exit the Expert System"

Then $END = -999$
 $STOP$

The variable END is used later by READER to indicate that further processing by the batch file is not needed (since the user is exiting the system). STOP is an EXSYS variable telling the expert system not to continue asking any more questions.

Now that all the values for the level 2 questions have been found, we are now ready to find the answer to the level 1 question on system considerations. Rules 107 and 108 are used for this and they are invoked only if C4=1.

Rule 107 -- If $SM+SC+NAP \geq 3$ then $SYSCON=YES$, which means that system considerations are critical.

Rule 108 -- If $SM+SC+NAP < 3$ then $SYSCON=NO$, which means that system considerations are not critical.

AD-A173 993

EXPERT SYSTEM FOR SOFTWARE QUALITY ASSURANCE(U) MCLEAN
RESEARCH CENTER INC UA W E BAUM ET AL. 12 NOV 86
DRAK78-84-D-0052

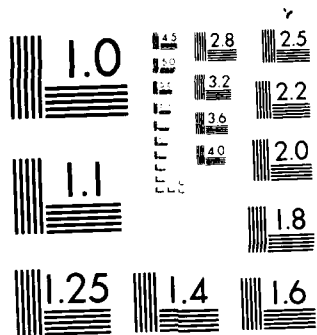
2/2

UNCLASSIFIED

F/G 9/2

NL

END
DATE
FILMED
4-87
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

** LEVEL 1 ** Life cycle considerations are

- 1 crucial
- 2 not crucial
- 3 NEED MORE INFORMATION

We are now back to the next level 1 question. Rules 109, 110 and 111 correspond to this question. Rule 109 states that if life cycle considerations are crucial then LC=YES and C8=0. Rule 110 states that if they are not crucial then LC=NO and C8=0. Rule 111 states that if more information is needed then C8=1, else C8=0. C8 is used as a flag to direct the flow of questioning down to level 2 if more information is needed, that is, when C8=1.

* Level 2 * Multiple contractor/multiple agency involvement is

- 1 low
- 2 medium
- 3 high
- 4 NEED MORE INFORMATION

This level 2 question is asked only when C8=1, that is, when more information was needed about life cycle considerations. Rules 112, 113, 114 and 115 correspond to this question. Rule 112 states that if involvement is low then MC=0 and C9=0. Rule 113 states that if involvement is medium then MC=1 and C9=0. Rule 114 states that if involvement is high then MC=2 and C9=0. Rule 115 states that if more information is needed then C9=1, else C9=0. This variable directs the flow of questioning to level 3 if C9=1, which occurs when more information is needed.

The next set of level 3 questions are asked only if C8=1 and C9=1 and relate to the level 2 question on multiple contractor/multiple agency involvement.

- LEVEL 3 - Will more than one major command, service, agency, or government department be involved?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 116 and 117 and they state the following:

Rule 116 -- If the answer is YES then V29=1

Rule 117 -- If the answer is DON'T KNOW then NK29=1 and V29=1

- LEVEL 3 - Will more than one primary contractor be involved?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 118 and 119 and they state the following:

Rule 118 -- If the answer is YES then V30=1

Rule 119 -- If the answer is DON'T KNOW then NK30=1 and V30=1

- LEVEL 3 - Will multiple sites be required for the use and storage of the requirements, design, test, and support documents?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 120 and 121 and they state the following:

Rule 120 -- If the answer is YES then V32=1

Rule 121 -- If the answer is DON'T KNOW then NK32=1 and V32=1

At this point, there is enough information from the level 3 questions to determine the multiple contractor/multiple agency involvement, which is the level 2 question we were trying to answer. Rules 122, 123 and 124 deal with this subject. These rules are invoked only when C8=1 and C9=1, indicating that more information was needed to answer both the level 1 and level 2 questions. The evaluation is based on the number of YES answers to the level 3 questions determined by the value of the variables (V1,V2,...). The rules state the following:

Rule 122 -- If $V29+V30+V32 < 1$ then $MC=0$
Rule 123 -- If $V29+V30+V32 = 1$ then $MC=1$
Rule 124 -- If $V29+V30+V32 > 1$ then $MC=2$

The value for MC found here will be used later to help determine the value for life cycle considerations, the level 1 question.

Rule 125 then evaluates the number of DON'T KNOW responses for the questions about multiple contractor/multiple agency involvement and if there are too many, a warning message will be called and appear on the screen. Variables are used to count up these responses (they all start with NK) and a threshold value is determined. At this point, the user can either exit the expert system or continue on. Rule 125 states the following:

If $NK29+NK30+NK32 \geq 2$
and the warning message is answered with "Exit the Expert System"

Then $END = -999$
STOP

The variable END is used later by READER to indicate that further processing by the batch file is not needed (since the user is exiting the system). STOP is an EXSYS variable telling the expert system not to continue asking any more questions.

* LEVEL 2 * System adaptability requirements are
1 low
2 medium
3 high
4 NEED MORE INFORMATION

This level 2 question is asked only when C8=1, that is, when more information was needed about life cycle considerations. Rules 126, 127, 128 and 129 correspond to this question. Rule 126 states that if the requirements are low then $AS=0$ and $C10=0$. Rule 127 states that if the requirements are medium then $AS=1$ and $C10=0$. Rule 128 states that if the requirements are high then $AS=2$ and $C10=0$. Rule 129 states that if more information is needed then $C10=1$, else $C10=0$. This variable directs the flow of questioning to level 3 if $C10=1$, which occurs when more information is needed.

The next set of level 3 questions are asked only if C8=1 and C10=1 and relate to the level 2 question on system adaptability requirements.

- LEVEL 3 - Will more than half of the computer system's capacity be used for this software's timing (speed) and sizing (memory) requirements?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 130 and 131 and they state the following:

Rule 130 -- If the answer is YES then V33=1

Rule 131 -- If the answer is DON'T KNOW then NK33=1 and V33=1

- LEVEL 3 - Are there any requirements which limit the expansion of the software?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 132 and 133 and they state the following:

Rule 132 -- If the answer is YES then V34=1

Rule 133 -- If the answer is DON'T KNOW then NK34=1 and V34=1

- LEVEL 3 - Will flexibility requirements be waived? (Flexibility is a measure of the effort required to enhance the software or to modify it to meet new requirements.)

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 134 and 135 and they state the following:

Rule 134 -- If the answer is YES then V35=1

Rule 135 -- If the answer is DON'T KNOW then NK35=1 and V35=1

- LEVEL 3 - Will modularity requirements be waived?
 - 1 YES
 - 2 NO
 - 3 DON'T KNOW

The rules that apply to this question are Rules 136 and 137 and they state the following:

- Rule 136 -- If the answer is YES then V36=1
- Rule 137 -- If the answer is DON'T KNOW then NK36=1 and V36=1

- LEVEL 3 - Will the requirement for the contractor to recommend approaches for expansion be waived for the RFP?
 - 1 YES
 - 2 NO
 - 3 DON'T KNOW

The rules that apply to this question are Rules 138 and 139 and they state the following:

- Rule 138 -- If the answer is YES then V37=1
- Rule 139 -- If the answer is DON'T KNOW then NK37=1 and V37=1

- LEVEL 3 - Will portability requirements be waived? (Portability is a measure of the effort required to transfer the software from one hardware or software environment to another.)
 - 1 YES
 - 2 NO
 - 3 DON'T KNOW

The rules that apply to this question are Rules 140 and 141 and they state the following:

- Rule 140 -- If the answer is YES then V38=1
- Rule 141 -- If the answer is DON'T KNOW then NK38=1 and V38=1

- LEVEL 3 - Is this an upgrade to, modification to, or conversion from an existing software system?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 142 and 143 and they state the following:

Rule 142 -- If the answer is YES then V39=1
Rule 143 -- If the answer is DON'T KNOW then NK39=1 and V39=1

At this point, there is enough information from the level 3 questions to determine the system adaptability requirements, which is the level 2 question we were trying to answer. Rules 144, 145 and 146 deal with this subject. These rules are invoked only when C8=1 and C10=1, indicating that more information was needed to answer both the level 1 and level 2 questions. The evaluation is based on the number of YES answers to the level 3 questions determined by the value of the variables (V1,V2,...). The rules state the following:

Rule 144 -- If $V33+V34+V35+V36+V37+V38+V39 \leq 1$ then AS=0
Rule 145 -- If $1 < V33+V34+V35+V36+V37+V38+V39 \leq 3$ then AS=1
Rule 146 -- If $V33+V34+V35+V36+V37+V38+V39 > 3$ then AS=2

The value for AS found here will be used later to help determine the value for life cycle considerations, the level 1 question.

Rule 147 then evaluates the number of DON'T KNOW responses for the questions about system adaptability requirements and if there are too many, a warning message will be called and appear on the screen. Variables are used to count up these responses (they all start with NK) and a threshold value is determined. At this point, the user can either exit the expert system or continue on. Rule 147 states the following:

If $NK33+NK34+NK35+NK36+NK37+NK38+NK39 \geq 4$
and the warning message is answered with "Exit the Expert System"

Then END = -999
 STOP

The variable END is used later by READER to indicate that further processing by the batch file is not needed (since the user is exiting the system). STOP is an EXSYS variable telling the expert system not to continue asking any more questions.

* LEVEL 2 * The system/software life cycle relationship problems are

- 1 low
- 2 medium
- 3 high
- 4 NEED MORE INFORMATION

This level 2 question is asked only when C8=1, that is, when more information was needed about life cycle considerations. Rules 148, 149, 150 and 151 correspond to this question. Rule 148 states that if the relationship problems are low then RS=0 and C12=0. Rule 149 states that if the problems are medium then RS=1 and C12=0. Rule 150 states that if the problems are high then RS=2 and C12=0. Rule 151 states that if more information is needed then C12=1, else C12=0. This variable directs the flow of questioning to level 3 if C12=1, which occurs when more information is needed.

The next set of level 3 questions are asked only if C8=1 and C12=1 and relate to the level 2 question on life cycle relationship problems.

- LEVEL 3 - Is there a planned difference in the projected life time of the hardware system and the software to be developed?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 152 and 153 and they state the following:

- Rule 152 -- If the answer is YES then V40=1
- Rule 153 -- If the answer is DON'T KNOW then NK40=1 and V40=1

- LEVEL 3 - Is there a planned difference in the projected life time of the hardware system and the software to be developed?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 154 and 155 and they state the following:

- Rule 154 -- If the answer is YES then V41=1
- Rule 155 -- If the answer is DON'T KNOW then NK41=1 and V41=1

- LEVEL 3 - Is the software scheduled to be developed and finished before the hardware?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 156 and 157 and they state the following:

Rule 156 -- If the answer is YES then V42=1
Rule 157 -- If the answer is DON'T KNOW then NK42=1 and V42=1

At this point, there is enough information from the level 3 questions to determine the system/software life cycle relationship problems, which is the level 2 question we were trying to answer. Rules 158, 159 and 160 deal with this subject. These rules are invoked only when C8=1 and C12=1, indicating that more information was needed to answer both the level 1 and level 2 questions. The evaluation is based on the number of YES answers to the level 3 questions determined by the value of the variables (V1,V2,...). The rules state the following:

Rule 158 -- If $V40+V41+V42 < 1$ then RS=0
Rule 159 -- If $V40+V41+V42 = 1$ then RS=1
Rule 160 -- If $V40+V41+V42 > 1$ then RS=2

The value for RS found here will be used later to help determine the value for life cycle considerations, the level 1 question.

Rule 161 then evaluates the number of DON'T KNOW responses for the questions about system/software life cycle relationship problems and if there are too many, a warning message will be called and appear on the screen. Variables are used to count up these responses (they all start with NK) and a threshold value is determined. At this point, the user can either exit the expert system or continue on. Rule 161 states the following:

If $NK10+NK41+NK42 \geq 2$
and the warning message is answered with "Exit the Expert System"

Then END = -999
STOP

The variable END is used later by READER to indicate that further processing by the batch file is not needed (since the user is exiting the system). STOP is an EXSYS variable telling the expert system not to continue asking any more questions.

* LEVEL 2 * The projected software life cycle is

- 1 short
- 2 medium
- 3 long
- 4 NEED MORE INFORMATION

This level 2 question is asked only when C8=1, that is, when more information was needed about life cycle considerations. Rules 162, 163, 164 and 165 correspond to this question. Rule 162 states that if the projected life cycle is short then PS=0 and C13=0. Rule 163 states that if it is medium then PS=1 and C13=0. Rule 164 states that if it is long then PS=2 and C13=0. Rule 165 states that if more information is needed then C13=1, else C13=0. This variable directs the flow of questioning to level 3 if C13=1, which occurs when more information is needed.

The next set of level 3 questions are asked only if C8=1 and C13=1 and relate to the level 2 question on the projected software life cycle.

- LEVEL 3 - Is the expected operational life for the software at least 5 years?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 166 and 167 and they state the following:

Rule 166 -- If the answer is YES then V43=1

Rule 167 -- If the answer is DON'T KNOW then NK43=1 and V43=1

- LEVEL 3 - Are there explicitly stated maintainability requirements?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 168 and 169 and they state the following:

Rule 168 -- If the answer is YES then V45=1

Rule 169 -- If the answer is DON'T KNOW then NK45=1 and V45=1

At this point, there is enough information from the level 3 questions to determine the projected software life cycle, which is the level 2 question we were trying to answer. Rules 170, 171 and 172 deal with this subject. These rules are invoked only when C8=1 and C13=1, indicating that more information was needed to answer both the level 1 and level 2 questions. The evaluation is based on the number of YES answers to the level 3 questions determined by the value of the variables (V1,V2,...). The rules state the following:

Rule 170 -- If $V43+V45 = 0$ then $PS=0$

Rule 171 -- If $V43+V45 = 1$ then $PS=1$

Rule 172 -- If $V43+V45 > 1$ then $PS=2$

The value for PS found here will be used next to help determine the value for life cycle considerations, the level 1 question.

Rule 173 then evaluates the number of DON'T KNOW responses for the questions about the projected software life cycle and if there are too many, a warning message will be called and appear on the screen. Variables are used to count up these responses (they all start with NK) and a threshold value is determined. At this point, the user can either exit the expert system or continue on. Rule 173 states the following:

If $NK43+NK45 \geq 1$

and the warning message is answered with "Exit the Expert System"

Then $END = -999$

STOP

The variable END is used later by READER to indicate that further processing by the batch file is not needed (since the user is exiting the system). STOP is an EXSYS variable telling the expert system not to continue asking any more questions.

Now that all the values for the level 2 questions have been found, we are now ready to find the answer to the level 1 question on life cycle considerations. Rules 174 and 175 are used for this and they are invoked only if C8=1.

Rule 174 -- If $MC+AS+RS+PS \geq 4$ then $LC=YES$, which means that life cycle considerations are crucial

Rule 175 -- If $MC+AS+RS+PS < 4$ then $LC=NO$, which means that life cycle considerations are not crucial.

**** LEVEL 1 **** Managerial considerations are

- 1 crucial
- 2 not crucial
- 3 NEED MORE INFORMATION

We are now back to the next level 1 question. Rules 176, 177 and 178 correspond to this question. Rule 176 states that if managerial considerations are crucial then M=YES and C14=0. Rule 177 states that if they are not crucial then M=NO and C14=0. Rule 178 states that if more information is needed then C14=1, else C14=0. C14 is used as a flag to direct the flow of questioning down to level 2 if more information is needed, that is, when C14=1.

*** LEVEL 2 *** Software developed schedule problems are

- 1 small
- 2 medium
- 3 large
- 4 NEED MORE INFORMATION

This level 2 question is asked only when C14=1, that is, when more information was needed about managerial considerations. Rules 179, 180, 181 and 182 correspond to this question. Rule 179 states that if the problems are small then SD=0 and C15=0. Rule 180 states that if the problems are medium then SD=1 and C15=0. Rule 181 states that the problems are large then SD=2 and C15=0. Rule 182 states that if more information is needed then C15=1, else C15=0. This variable directs the flow of questioning to level 3 if C15=1, which occurs when more information is needed.

The next set of level 3 questions are asked only if C14=1 and C15=1 and relate to the level 2 question on the software developed schedule problems.

- LEVEL 3 - Has the proposed "need by" time frame for the software been deemed realistic?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 183 and 184 and they state the following:

Rule 183 -- If the answer is YES then V46=1
Rule 184 -- If the answer is DON'T KNOW then NK46=1 and V46=1

- LEVEL 3 - Will this software/system replace an existing system?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 185 and 186 and they state the following:

Rule 185 -- If the answer is YES then V47=1
Rule 186 -- If the answer is DON'T KNOW then NK47=1 and V47=1

- LEVEL 3 - Is the software development for this system on the critical path of the system development?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 187 and 188 and they state the following:

Rule 187 -- If the answer is YES then V48=1
Rule 188 -- If the answer is DON'T KNOW then NK48=1 and V48=1

- LEVEL 3 - Is the software documentation for this system on the critical path of the system development?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 189 and 190 and they state the following:

Rule 189 -- If the answer is YES then V49=1
Rule 190 -- If the answer is DON'T KNOW then NK49=1 and V49=1

- LEVEL 3 - Is there a sufficient number of contracting agency personnel to thoroughly review all documentation required?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 191 and 192 and they state the following:

Rule 191 -- If the answer is YES then V50=1
Rule 192 -- If the answer is DON'T KNOW then NK50=1 and V50=1

At this point, there is enough information from the level 3 questions to determine the software developed schedule problems, which is the level 2 question we were trying to answer. Rules 193, 194 and 195 deal with this subject. These rules are invoked only when C14=1 and C15=1, indicating that more information was needed to answer both the level 1 and level 2 questions. The evaluation is based on the number of YES answers to the level 3 questions determined by the value of the variables (V1,V2,...). The rules state the following:

Rule 193 -- If $V46+V47+V48+V49+V50 \leq 1$ then $SD=0$
Rule 194 -- If $V46+V47+V48+V49+V50 = 2$ then $SD=1$
Rule 195 -- If $V46+V47+V48+V49+V50 > 2$ then $SD=2$

The value for SD found here will be used later to help determine the value for managerial considerations, the level 1 question.

Rule 196 then evaluates the number of DON'T KNOW responses for the questions about software developed schedule problems and if there are too many, a warning message will be called and appear on the screen. Variables are used to count up these responses (they all start with NK) and a threshold value is determined. At this point, the user can either exit the expert system or continue on. Rule 196 states the following:

```
If    NK46+NK47+NK48+NK49+NK50 >= 3
      and the warning message is answered with "Exit the Expert System"

Then  END = -999
      STOP
```

The variable END is used later by READER to indicate that further processing by the batch file is not needed (since the user is exiting the system). STOP is an EXSYS variable telling the expert system not to continue asking any more questions.

* LEVEL 2 * The contracting agency manning and expertise is

- 1 limited
- 2 adequate
- 3 extensive
- 4 NEED MORE INFORMATION

This level 2 question is asked only when C14=1, that is, when more information was needed about managerial considerations. Rules 197, 198, 199 and 200 correspond to this question. Rule 197 states that if the answer is limited then CA=0 and C16=0. Rule 198 states that if the answer is adequate then CA=1 and C16=0. Rule 199 states that if the answer is extensive then CA=2 and C16=0. Rule 200 states that if more information is needed then C16=1, else C16=0. This variable directs the flow of questioning to level 3 if C16=1, which occurs when more information is needed.

The next set of level 3 questions are asked only if C14=1 and C16=1 and relate to the level 2 question on the contracting agency manning and expertise.

- LEVEL 3 - Are there experienced software engineering personnel available at the contracting agency to review technically complex documentation?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 201 and 202 and they state the following:

- Rule 201 -- If the answer is YES then V51=1
- Rule 202 -- If the answer is DON'T KNOW then NK51=1 and V51=1

- LEVEL 3 - Are there experienced review and audit personnel at the contracting agency to attend all reviews and perform all audits?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 203 and 204 and they state the following:

- Rule 203 -- If the answer is YES then V52=1
- Rule 204 -- If the answer is DON'T KNOW then NK52=1 and V52=1

- LEVEL 3 - Are personnel available at the contracting agency for those time-intensive tasks, such as quality evaluation?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 205 and 206 and they state the following:

- Rule 205 -- If the answer is YES then V53=1
- Rule 206 -- If the answer is DON'T KNOW then NK53=1 and V53=1

At this point, there is enough information from the level 3 questions to determine the contracting agency manning and expertise, which is the level 2 question we were trying to answer. Rules 207, 208 and 209 deal with this subject. These rules are invoked only when C14=1 and C16=1, indicating that more information was needed to answer both the level 1 and level 2 questions. The evaluation is based on the number of YES answers to the level 3 questions determined by the value of the variables (V1,V2,...). The rules state the following:

Rule 207 -- If $V51+V52+V53 \leq 1$ then CA=0

Rule 208 -- If $V51+V52+V53 = 2$ then CA=1

Rule 209 -- If $V51+V52+V53 > 2$ then CA=2

The value for CA found here will be used later to help determine the value for managerial considerations, the level 1 question.

Rule 210 then evaluates the number of DON'T KNOW responses for the questions about the contracting agency manning and expertise and if there are too many, a warning message will be called and appear on the screen. Variables are used to count up these responses (they all start with NK) and a threshold value is determined. At this point, the user can either exit the expert system or continue on. Rule 210 states the following:

If $NK51+NK52+NK53 \geq 1$

and the warning message is answered with "Exit the Expert System"

Then END = -999

STOP

The variable END is used later by READER to indicate that further processing by the batch file is not needed (since the user is exiting the system). STOP is an EXSYS variable telling the expert system not to continue asking any more questions.

* LEVEL 2 * The cost of documentation preparation and update is

- 1 low
- 2 medium
- 3 high
- 4 NEED MORE INFORMATION

This level 2 question is asked only when C14=1, that is, when more information was needed about managerial considerations. Rules 211, 212, 213 and 214 correspond to this question. Rule 211 states that if the answer is low then CD=0 and C17=0. Rule 212 states that if the answer is medium then CD=1 and C17=0. Rule 213 states that if the answer is high then CD=2 and C17=0. Rule 214 states that if more information is needed then C17=1, else C17=0. This variable directs the flow of questioning to level 3 if C17=1, which occurs when more information is needed.

The next set of level 3 questions are asked only if C14=1 and C17=1 and relate to the level 2 question on the cost of documentation.

- LEVEL 3 - In terms of budgeted project dollars, does the benefit outweigh the cost of documenting the software requirements?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 215 and 216 and they state the following:

Rule 215 -- If the answer is YES then V54=1

Rule 216 -- If the answer is DON'T KNOW then NK54=1 and V54=1

- LEVEL 3 - In terms of budgeted project dollars, does the benefit outweigh the cost of documenting software design?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 217 and 218 and they state the following:

Rule 217 -- If the answer is YES then V55=1

Rule 218 -- If the answer is DON'T KNOW then NK55=1 and V55=1

- LEVEL 3 - In terms of budgeted project dollars, does the benefit outweigh the cost of documenting the formal testing program?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 219 and 220 and they state the following:

Rule 219 -- If the answer is YES then V56=1
Rule 220 -- If the answer is DON'T KNOW then NK56=1 and V56=1

- LEVEL 3 - In terms of budgeted project dollars, does the benefit outweigh the cost of documenting the internal management practices?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 221 and 222 and they state the following:

Rule 221 -- If the answer is YES then V57=1
Rule 222 -- If the answer is DON'T KNOW then NK57=1 and V57=1

- LEVEL 3 - In terms of budgeted project dollars, does the benefit outweigh the cost of documenting the support and operation procedures?

- 1 YES
- 2 NO
- 3 DON'T KNOW

The rules that apply to this question are Rules 223 and 224 and they state the following:

Rule 223 -- If the answer is YES then V58=1
Rule 224 -- If the answer is DON'T KNOW then NK58=1 and V58=1

At this point, there is enough information from the level 3 questions to determine the cost of documentation preparation and update, which is the level 2 question we were trying to answer. Rules 225, 226 and 227 deal with this subject. These rules are invoked only when C14=1 and C17=1, indicating that more information was needed to answer both the level 1 and level 2 questions. The evaluation is based on the number of YES answers to the level 3 questions determined by the value of the variables (V1,V2,...). The rules state the following:

Rule 225 -- If $V54+V55+V56+V57+V58 \leq 1$ then $CD=0$
Rule 226 -- If $1 < V54+V55+V56+V57+V58 \leq 3$ then $CD=1$
Rule 227 -- If $V54+V55+V56+V57+V58 > 3$ then $CD=2$

The value for CD found here will be used next to help determine the value for managerial considerations, the level 1 question.

Rule 228 then evaluates the number of DON'T KNOW responses for the questions about the cost of documentation preparation and update and if there are too many, a warning message will be called and appear on the screen. Variables are used to count up these responses (they all start with NK) and a threshold value is determined. At this point, the user can either exit the expert system or continue on. Rule 228 states the following:

```
If    NK54+NK55+NK56+NK57+NK58 >= 2
      and the warning message is answered with "Exit the Expert System"

Then  END = -999
      STOP
```

The variable END is used later by READER to indicate that further processing by the batch file is not needed (since the user is exiting the system). STOP is an EXSYS variable telling the expert system not to continue asking any more questions.

Now that all the values for the level 2 questions have been found, we are now ready to find the answer to the level 1 question on managerial considerations. Rules 229 and 230 are used for this and they are invoked only if C14=1.

Rule 229 -- If $SD+CA+CD \geq 2$ then M=YES, which means that managerial considerations are crucial

Rule 230 -- If $SD+CA+CD < 2$ then M=NO, which means that managerial considerations are not crucial.

This completes the section of questions that are structured in levels. Based on the values to the level 1 question variables, specific paragraphs will be included in the statement of work. The following rules illustrate the relationship between the paragraphs and the responses:

```
Rule 239 -- If M=YES then the following paragraphs are included:
          904.00, 905.00, 908.00, 914.00, 915.00, 917.00, 918.00,
          921.00, 922.00, 1003.00, 1004.00, 1100.00, 1200.00,
          1500.00, 2400.00, 2500.00, 3500.00, 3600.00, 4000.00,
          4700.00, 4800.00, 5304.00, 5710.00, 5770.00, 5800.00
```

Rule 240 -- If SYSCON=YES then the following paragraphs are included:
 907.00, 908.00, 910.00, 911.00, 914.00, 915.00, 918.00,
 919.00, 920.00, 1003.00, 1004.00, 1400.00, 1500.00,
 1700.00, 1800.00, 2100.00, 2200.00, 2500.00, 2600.00,
 2700.00, 3800.00, 4000.00, 5200.00, 5306.00, 5307.00,
 5309.00, 5740.00, 5760.00, 5765.00, 5770.00, 5930.00,
 6000.00, 6100.00, 6300.00

Rule 241 -- If LC=YES then the following paragraphs are included:
 908.00, 914.00, 915.00, 917.00, 918.00, 919.00, 920.00,
 921.00, 922.00, 923.00, 1003.00, 1004.00, 1500.00, 2400.00,
 2500.00, 2600.00, 2700.00, 3500.00, 3600.00, 4000.00,
 4600.00, 4700.00, 4800.00, 5200.00, 5770.00

Rule 242 -- If SYSCON=YES and LC=YES then include paragraph 3200.00

Rule 245 -- If paragraphs 924.00 and 4900.00 are included and SYSCON=NO
 and M=NO and LC=YES then include paragraph 4910.00

Rule 246 -- If SYSCON=YES and M=YES and LC=NO then include paragraphs
 960.00, 1130.00, 1060.00, 1510.00

Rule 247 -- If SYSCON=NO and LC=YES and M=YES then include 1110.00,
 1520.00, 1310.00, 1070.00, 2020.00, 4710.00, 2410.00, 2610.00

Rule 248 -- If SYSCON=NO and LC=NO and M=YES then include 1120.00,
 1210.00, 2040.00, 1080.00

Rule 249 -- If SYSCON=YES and LC=NO and M=NO then include 2010.00,
 1420.00, 1710.00, 1610.00

Rule 250 -- If SYSCON=YES and LC=YES and M=NO then include 2030.00,
 4810.00, 1415.00, 1810.00, 1090.00

Rule 251 -- If SYSCON=NO and M=NO and LC=YES then include 4820.00 and
 4610.00

The next group of questions are all structured in the same way. The user can answer with YES, NO or DON'T KNOW. Answering YES results in a certain paragraph being included in the statement of work. Answering NO results in no additional paragraphs included in the statement of work. In the future, the DON'T KNOW answers will be totaled up and if there are too many of them, a warning message will be displayed to user telling him he does not know enough to obtain a meaningful solution. However, at present, nothing is done with these answers and they have the same impact as answering YES.

Is there to be an interface with independent verification and validation activities?

- 1 YES
- 2 NO
- 3 DON'T KNOW

This question corresponds to Rule 231. If the answer is YES or DON'T KNOW then paragraph 7000.00 will be included in the statement of work.

Are you requiring design walkthroughs during the design phase of the project?

- 1 YES
- 2 NO (or no design phase envisioned for this project)
- 3 DON'T KNOW

This question corresponds to Rule 232. If the answer is YES or DON'T KNOW then paragraph 7100.00 will be included in the statement of work.

Are you requiring design inspection during the design phase of this project?

- 1 YES
- 2 NO (or no design phase envisioned for this project)
- 3 DON'T KNOW

This question corresponds to Rule 233. If the answer is YES or DON'T KNOW then paragraph 7200.00 will be included in the statement of work.

Are you requiring source code analysis during the coding phase of this project?

- 1 YES
- 2 NO (or no coding phase envisioned for this project)
- 3 DON'T KNOW

This question corresponds to Rule 234. If the answer is YES or DON'T KNOW then paragraph 7300.00 will be included in the statement of work.

Are you requiring code walkthroughs during the coding phase of this project?

- 1 YES
- 2 NO (or no coding phase envisioned for this project)
- 3 DON'T KNOW

This question corresponds to Rule 235. If the answer is YES or DON'T KNOW then paragraph 7400.00 will be included in the statement of work.

Are you requiring code inspection during the coding phase of this project?

- 1 YES
- 2 NO (or no coding phase envisioned for this project)
- 3 DON'T KNOW

This question corresponds to Rule 236. If the answer is YES or DON'T KNOW then paragraph 7500.00 will be included in the statement of work.

Will software stress testing be required during the testing phase of this project?

- 1 YES
- 2 NO (or no testing phase envisioned for this project)
- 3 DON'T KNOW

This question corresponds to Rule 237. If the answer is YES or DON'T KNOW then paragraph 7600.00 will be included in the statement of work.

Are there anticipated upgrades to the software or is a preplanned product improvements program anticipated for the software?

- 1 YES
- 2 NO
- 3 DON'T KNOW

This question corresponds to Rule 238. If the answer is YES or DON'T KNOW then paragraph 4100.00 will be included in the statement of work.

The type of firmware to be used in the system is

- 1 none
- 2 non-programmable
- 3 PROM
- 4 EPROM
- 5 combination

This question corresponds to Rule 243. If the answer is PROM, EPROM or combination then a variable called FLAG1 will be set to 1. This variable is used in conjunction with variable LC to determine if paragraphs 924.00 and 4900.00 should be included in the statement of work. That rule is structured as follows:

Rule 244 -- If Flag1=1 and LC=YES then include paragraphs 924.00 and 4900.00

The support concept for the software is

- 1 the development contractor
- 2 an agency other than the development contractor

This question corresponds to Rule 252. If the answer is an agency other than the development contractor then paragraph 745.00 is to be included in the statement of work.

Will a description of the computer system functions and user interactions into a high level, descriptive document enhance coordination and consensus of the system concept among applicable Government agencies?

- 1 Yes
- 2 No

The question corresponds to Rule 253. If the answer is YES then paragraphs 919.00 and 2600.00 will be included in the statement of work.

Will the computer system require any human interaction for system start-up, operation, monitoring or recovery procedures?

- 1 Yes
- 2 No

The question corresponds to Rule 254. If the answer is YES then paragraphs 921.00 and 4700.00 will be included in the statement of work.

Do either the computer system or associated tools provide diagnostic features to identify a malfunction and isolate a malfunctioning unit?

- 1 Yes
- 2 No

The question corresponds to Rule 255. If the answer is YES then paragraphs 922.00 and 4800.00 will be included in the statement of work.

Will the software operating in this computer system need to be supported by any agency other than the contractor?

- 1 Yes
- 2 No

The question corresponds to Rule 256. If the answer is YES then the variable VAR1 is given a value of 1. This variable will be used with the variable from the next question because both of these questions have to be answered YES to have paragraphs included in the statement of work.

Is documentation necessary to describe all of the programming features and the programming environment used in the deliverable software?

- 1 Yes
- 2 No

The question corresponds to Rule 257. If the answer is YES then the variable VAR2 is given a value of 1. This variable will be used with the variable from the preceding question because both of these questions have to be answered YES to have paragraphs included in the statement of work.

Now that the values for VAR1 and VAR2 have been determined, Rule 258 can be executed. It states that if VAR1=1 and VAR2=1 then include paragraphs 923.00, 4600.00, 5100.00, 5110.00, 5120.00, 5130.00, and 5140.00

Will any of the software operating in this computer system be embedded in the firmware?

- 1 Yes
- 2 No

The question corresponds to Rule 259. If the answer is YES then the variable VAR3 is given a value of 1. This variable is used to determine if the next question will be asked. It is only asked if the answer to this question is YES.

Will the firmware be supported by any agency other than the contractor?

- 1 Yes
- 2 No

This question is asked only if the preceding question was answered YES, that is, if VAR3=1 and corresponds to Rule 260. If the answer is YES then the variable VAR4 is given a value of 1. This variable is used in conjunction with VAR3 to determine if a paragraph will be included in the statement of work.

Now that the values for VAR3 and VAR4 have been determined, Rule 261 can be executed. It states that if VAR3=1 and VAR4=1 then include paragraphs 924.00 and 4900.00.

Will the software requirements be broken out specifically from the system requirements?

- 1 Yes
- 2 No

This question corresponds to Rule 262. If the answer is YES then the variable VAR5 is given a value of 1. This variable will be used with the variable from the next question because both of these questions have to be answered YES to have paragraphs included in the statement of work.

Is it necessary for the Government to formally review and approve what the software will do prior to its implementation?

- 1 Yes
- 2 No

The question corresponds to Rule 263. If the answer is YES then the variable VAR6 is given a value of 1. This variable will be used with the variable from the preceding question because both of these questions have to be answered YES to have paragraphs included in the statement of work.

Now that the values for VAR5 and VAR6 have been determined, Rule 264 can be executed. It states that if VAR5=1 and VAR6=1 then include paragraphs 906.00 and 1300.00.

Do the requirements include a highly involved interface with another CSCI or HSCI?

- 1 Yes
- 2 No

This question corresponds to Rule 265. If the answer is YES then paragraphs 907.00 and 1400.00 will be included in the statement of work.

Does the design of the software need to be documented?

- 1 Yes
- 2 No

The question corresponds to Rule 266. If the answer is YES then the variable VAR7 is given a value of 1. This variable will be used with the variable from the next question because both of these questions have to be answered YES to have paragraphs included in the statement of work.

Will the software be of such criticality, magnitude or complexity that the software design should be implemented, documented and reviewed as a 2-step process?

- 1 Yes
- 2 No

The question corresponds to Rule 267. If the answer is YES then the variable VAR8 is given a value of 1. This variable will be used with the variable from the preceding question because both of these questions have to be answered YES to have paragraphs included in the statement of work.

Now that the values for VAR7 and VAR8 have been determined, Rule 268 can be executed. It states that if VAR7=1 and VAR8=1 then include paragraphs 908.00 and 1500.00.

Will the complete design of the software need to be documented?

- 1 Yes
- 2 No

This question corresponds to Rule 269. If the answer is YES then paragraphs 909.00 and 1600.00 will be included in the statement of work. Additionally, the variable VAR10 will be given the value of 1 and used with another variable to determine if other paragraphs will be included.

Once VAR10 has been found, it is used in conjunction with the variable V15 (determined in the question asking if more than one data base will be accessed). Rule 270 states that if VAR10=1 and V15=1 then include paragraphs 911.00 and 1800.00 in the statement of work.

VAR10 is also used in Rule 274 which states that if VAR10=1 then include paragraph 1410.00 in the statement of work.

Will the contractor be updating the deliverable software and distributing it to Government agencies?

- 1 Yes
- 2 No

The question corresponds to Rule 271. If the answer is YES then paragraphs 918.00 and 2500.00 will be included in the statement of work.

Will some agency other than the original contractor ever support the software?

- 1 Yes
- 2 No

The question corresponds to Rule 272. If the answer is YES then paragraphs 912.00 and 1900.00 will be included in the statement of work.

Do you want consolidated monthly software status reports or individual reports for each AMC-P-702-XX task?

- 1 Consolidated
- 2 Individual
- 3 Don't know

The question corresponds to Rule 273. If the answer is consolidated then paragraph 755.00 will be included in the statement of work.

This is the final question that is asked. EXSYS then executes its report generator function which writes to an external file all the paragraph numbers to be included in the statement of work. When a paragraph is to be included in the statement of work, its number is assigned to a variable (all these variables begin with the letter "P"), otherwise that variable has a value of zero. The report generator then writes only those variables with a value greater than zero to the external file. At the same time, it also writes the numbers of the paragraphs that always appear in the statement of work and are not decided within the expert system. This external file is called RESULTS and it is then further processed by the batch file to construct the tailored statement of work.

APPENDIX C -- SOW FILE

@10.00

STATEMENT OF WORK

@20.00

BRDEC SOFTWARE DEVELOPMENT AND SOFTWARE QUALITY EVALUATION

@100.00

SCOPE OF SOFTWARE DEVELOPMENT AND SOFTWARE QUALITY ASSURANCE TASKS

@200.00 General

This document delineates the Government's requirements for scientific, engineering, analysis, and technical services to support software development and software quality evaluation for BRDEC (Belvoir Research, Development and Engineering Center) mission-critical computer systems software.

@300.00 Scope of Work

The contractor shall provide all necessary personnel, supervision, management, materials, services, equipment, and facilities to perform software development and software quality evaluation for tactical systems being developed, managed, or supported by BRDEC.

The efforts shall include the application of proven methodologies and tools for software development and software quality evaluation, software documentation preparation, configuration management procedures, software test and evaluation, and technical writing for mission-critical software.

@400.00 Personnel Requirements

Personnel requirements may vary during the period of actual contract performance, and therefore, the contractor may be required to adjust both the extent and the composition of the actual work team(s) in order to effectively handle the then current workload.

The Government reserves the right, at any time, to confirm that all personnel assigned to the contract meet the minimum requirements in the contract.

@500.00 Phase-In, Phase-Out

The phase-in period for any subsequent contractor may not exceed ninety (90) calendar days prior to the expiration of this contract. During this time, the current contractor shall continue to meet full manning requirements for active delivery orders.

@600.00 Media for Deliverable Documentation

The contractor shall prepare and deliver each item (specification, description, procedure, report, manual, or document) required by this SOW, (unless stated otherwise in that particular item), in NUMBER copies (hardcopy)

and on electronic media in accordance with the related CDRL. The electronic media shall be TYPE compatible with a SYSTEM host system format.

@700.00 Standards and Specifications

The Standards and Specifications of the latest issue in effect on the date of invitation for bids or Request for Proposal (RFP) shall be used to the extent specified within this Statement of Work.

@710.00 Software Development Files

The contractor shall develop and maintain software development files for vertical integration of the documentation pertaining to each software unit. These files shall include as a minimum: software unit requirements, unit detailed design, software source code listing, and unit test case descriptions, unit test procedures, unit requirements, and unit test results. The software development files shall be made available to the Government, or a designated Government representative, for review and audit. The software development files shall be maintained, and stored by the contractor until three years after completion of the contract for possible Government use or reference.

@720.00 IV&V Support

The Government, or a designated representative, may perform Independent Verification and Validation (IV&V) of the software under development.

@730.00 Commercially Available, Reusable, and Government Furnished Software

The contractor shall identify any limited or restricted rights to any commercially available or reusable software. All commercially available or reusable software code, documentation, or data rights shall be negotiated and agreed to by the software development contractor prior to delivery of the software to the Government. When previously developed software is included in the software system, the software development contractor shall deliver, to the Government, current technical and user documentation for the previously developed software. The software development contractor shall modify the Software Development Plan (SDP) to contain plans for certification of the commercially available or reusable software and the proposed schedule for software development shall reflect time allocated for execution of these plans. The software development contractor shall modify the Software Requirements Specification (SRS) or Interface Requirements Specification (IRS) to reflect all interface requirements between the commercially available, reusable, or Government furnished software and any new software to be developed. The software development contractor shall modify the Software Detailed Design Document (SDDD) or Interface Design Document (IDD) to contain all interface design information for interfacing the commercially available, reusable, or Government furnished software with any new software to be developed.

@740.00 DD Form 1423 Block 16 Explanation

Block 16 of DD Form 1423 is used to identify when a draft of the document is due (if required). The Government review period is defined if a draft is produced. The due date of the revised document is then provided. After the revised document is approved then any changes or revisions to the

approved document shall be made in the form of change pages. The approved revised document with all the change pages (if any) will constitute the final document at the end of the contract. The "Reproducible; Electronic Media" comment in Block 16 reflects how the data is to be delivered. The reproducible and electronic media form requirements for delivery may be in Block 14 or Block 16. The actual number of copies will be found in Block 14.

@745.00 Delivery Requirements for Software Documentation

Support documentation tasked out of DOD-STD-2167 must be developed as part of the system.

@750.00 Software Status Reports

The contractor shall report on the status of the quality of the software being produced. This report shall describe software development progress, identify problems and solutions, and report on the findings of AMC-P-702-XX Tasks, report on the actions and efforts resulting from application of the Software Quality Evaluation/Assurance Program and Plan and report other pertinent information. These tasks are described in a "Technical Operating Report for Software Status" (DID # DI-S-30559) and submitted in accordance with the CDRL.

^ 1423 data for paragraph 750.00

BLOCK 2 -- Software Status Reports

BLOCK 4 -- DI-S-30559

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- DD

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Submittal to Government at initiation of contract and each month thereafter. Changes/revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@755.00

The software status reports called out under the various provisions of this statement of work shall be consolidated for delivery to the Government.

@800.00 Software Development

The following paragraph and subparagraphs delineates the Government requirements for software development.

@900.00 Requirements for Software Development

The contractor shall establish and follow a software development program in accordance with the requirements of DOD-STD-2167 and as specified herein. This program shall include, but not be limited to, preparation of the following documents described in DOD-STD-2167:

@902.00 System Segment Specification

- @903.00 Software Development Plan
- @904.00 Software Configuration Management Plan
- @905.00 Software Quality Evaluation Plan
- @906.00 Software Requirements Specification
- @907.00 Interface Requirements Specification
- @908.00 Software Top Level Design Document
- @909.00 Software Detailed Design Document
- @910.00 Interface Design Document
- @911.00 Data Base Design Document
- @912.00 Software Product Specification
- @913.00 Software Test Plan
- @914.00 Software Test Description
- @915.00 Software Test Procedure
- @916.00 Software Test Report
- @917.00 Software User's Manual
- @918.00 Version Description Document
- @919.00 Operational Concept Document
- @920.00 Computer Resources Integrated Support Document
- @921.00 Computer System Operator's Manual
- @922.00 Computer System Diagnostic Manual
- @923.00 Software Programmer's Manual
- @924.00 Firmware Support Manual

@930.00 Specific details applicable to each of the above tasks, along with other tasks not identified above, are provided in the individual statements of work for each task and are included below.

@950.00 System Segment Specification

A System Segment Specification (SSS) shall be prepared in accordance with paragraph 3.1.1 of MIL-STD-483, paragraphs 5.1.1.1 and 5.1.2.1 of DOD-STD-2167 and DID # DI-CMAN-80008. The SSS shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 950.00

BLOCK 2 -- System Segment Specification

BLOCK 3 -- SSS

BLOCK 4 -- DI-CMAN-80008

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- DD

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Draft specification shall be submitted NLT 30 days after contract award. Allow 30 days for Gov't review/comments. Revised SSS due NLT 30 days after receipt of Gov't comments. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@960.00

Paragraphs 10.2.5.1, 10.2.5.2, 10.2.6.1, 10.2.6.2, and 10.2.6.4 of the System Segment Specification DID # DI-CMAN-80008 are excluded from this statement of work and do not form part of the required effort under this contract.

@1000.00 Software Development Plan

A Software Development Plan (SDP) shall be prepared in accordance with paragraph 3.1.1 of MIL-STD-483, paragraphs 5.1.1.1 and 5.1.2.1 of DOD-STD-2167, and DID # DI-MCCR-80030. The SDP shall be submitted in accordance with the CDRL. In addition to the requirements of the Software Development Plan described in DI-MCCR-80030, the contractor shall include the following:

^ 1423 data for paragraph 1000.00

BLOCK 2 -- Software Development Plan

BLOCK 3 -- SDP

BLOCK 4 -- DI-MCCR-80030

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- DD

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Draft plan shall be submitted NLT 30 days after contract award. Allow 30 days for Gov't review/comments. Revised SDP due NLT 30 days after receipt of Gov't comments. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@1001.00

1). A detailed description of any other software development activities not included in the tasks specified above but which are specified elsewhere in this contract.

@1002.00

2). A detailed description of other contractor initiated software development activities, which are deemed necessary by the contractor, and which will be performed as part of the overall software development effort.

@1003.00

3). A detailed description of the manloading projected for each software development task. This shall include the type of personnel and the length of time required for each type.

@1004.00

4). The methods for monitoring, assessing, and reporting associated with each task (including frequency and type of report submittals).

@1010.00

The contractor shall identify the applicable requirements in the SDP and apply these requirements subject to contracting agency approval. As a minimum, the requirements shall include the procedures for:

- (a) Requirements analysis and allocation
- (b) Design and coding
- (c) Hardware and software integration and test
- (d) Coordination of hardware and software design
- (e) Documentation

- (f) Configuration management
- (g) Software quality evaluation

@1020.00

The contractor shall describe in the SDP the controls to be imposed on all non-deliverable software, firmware, and hardware used in the development and acquisition of deliverable software. As a minimum, the contractor shall describe the provisions for:

- (a) Modification (as applicable)
- (b) Documentation
- (c) Configuration management
- (d) Design and coding standards
- (e) Testing
- (f) Quality evaluation
- (g) Certification

@1030.00

The contractor shall perform the following activities prior to incorporating commercially available software, reusable software, GFS, or any combination of these, into the design:

(a) Describe in the SDP the data rights and documentation the contractor plans to provide the contracting agency regarding the commercially available and reusable software.

(b) Evaluate the commercially available, reusable, or GFS to determine whether it performs as documented.

(c) Describe in the SDP the contractor's plans for certifying the commercially available or reusable software.

(d) Obtain explicit contracting agency approval for use of commercially available software.

@1035.00

1) Paragraphs 10.2.5.3.1 through 10.2.5.3.3 and 10.2.7.2.1 through 10.2.7.2.7 are tailored out of the SDP since the configuration management information is to be provided in the corresponding Software Configuration Management Plan (SCMP), DI-MCCP-80009 or a System Configuration Management Plan, DI-E-3108.

@1036.00

2) Paragraphs 10.2.5.4.1 through 10.2.5.4.3 and 10.2.7.3.1 through 10.2.7.3.2.2.4 are tailored out of the SDP since the software quality evaluation information is to be provided in the corresponding Software Quality Evaluation Plan (SQEP), DI-MCCR-80010.

@1037.00

3) Paragraphs 10.2.7.1.1 through 10.2.7.1.7 are tailored out of the SDP since the software standards and procedures are to be provided in the corresponding Software Standards and Procedures Manual (SSPM), DI-MCCR-80011.

@1040.00

Upon approval by the Government, the Software Development Plan shall

be a contractual requirement and the contractor shall manage and conduct the development of the software in accordance with it. The approved Software Development Plan shall be the basis for determining contractual compliance with software development requirements.

@1050.00

Revisions to the Software Development Plan shall incorporate Government approved changes, additions, or deletions which have evolved during the development of the software since the previous issue of the plan.

@1060.00

Paragraph 10.2.7.1 of the Software Development Plan DID # DI-MCCR-80030 is excluded from this statement of work and do not form a part of the required effort under this contract.

@1070.00

Paragraph 10.2.7.2 of the Software Development Plan DID # DI-MCCR-80030 is excluded from this statement of work and do not form a part of the required effort under this contract.

@1080.00

Paragraphs 10.2.7.3, 10.2.7.4, 10.2.7.7, and 10.2.7.8 of the Software Development Plan DID # DI-MCCR-80030 are excluded from this statement of work and do not form a part of the required effort under this contract.

@1090.00

Paragraphs 10.2.7.9 and 10.2.7.10 of the Software Development Plan DID # DI-MCCR-80030 are excluded from this statement of work and do not form a part of the required effort under this contract.

@1100.00 Software Configuration Management Plan

A Software Configuration Management Plan (SCMP) shall be prepared in accordance with paragraph 3.1.1 of MIL-STD-483, paragraphs 5.1.1.1 and 5.1.2.1 of DOD-STD-2167 and DID # DI-MCCR-80009. The SCMP shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 1100.00

BLOCK 2 -- Software Configuration Management Plan

BLOCK 3 -- SCMP

BLOCK 4 -- DI-MCCR-80009

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- DD

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Draft plan shall be submitted NLT 60 days after contract award. Allow 45 days for Gov't review/comments. Revised SCMP due NLT 30 days after receipt of Gov't comments. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@1110.00

Paragraphs 10.2.5.1, 10.2.5.3, 10.2.6.2, and 10.2.6.6 of the Software

Configuration Management Plan DID # DI-MCCR-80009 are excluded from this statement of work and do not form a part of the required effort under this contract.

@1120.00

Paragraphs 10.2.5.2, 10.2.6.3, 10.2.6.4, and 10.2.6.5 of the Software Configuration Management Plan DID # DI-MCCR-80009 are excluded from this statement of work and do not form a part of the required effort under this contract.

@1130.00

Paragraph 10.2.6.1 of the Software Configuration Management Plan DID # DI-MCCR-80009 is excluded from this statement of work and do not form a part of the required effort under this contract.

@1200.00 Software Quality Evaluation Plan

A Software Quality Evaluation Plan (SQEP) shall be prepared in accordance with paragraphs 5.1.1.1 and 5.1.2.1 of DOD-STD-2167 and DID # DI-MCCR-80010. The SQEP shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 1200.00

BLOCK 2 -- Software Quality Evaluation Plan

BLOCK 3 -- SQEP

BLOCK 4 -- DI-MCCR-80010

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- DD

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Draft plan shall be submitted NLT 60 days after contract award. Allow 30 days for Gov't review/comments. Revised SQEP due NLT 30 days after receipt of Gov't comments. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@1210.00

Paragraphs 10.2.5, 10.2.6, and 10.2.7 of the Software Quality Evaluation Plan DID # DI-MCCR-80010 are excluded from this statement of work and do not form a part of the required effort under this contract.

@1300.00 Software Requirements Specification

A Software Requirements Specification (SRS) shall be prepared in accordance with paragraphs 3.4.2 and 3.4.7.1 of MIL-STD-483, paragraph 3.1.3.2.5.1 of MIL-STD-490, paragraphs 5.1.1.6 and 5.1.2.4 of DOD-STD-2167, and DID # DI-MCCR-80025. The SRS shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 1300.00

BLOCK 2 -- Software Requirements Specification

BLOCK 3 -- SRS

BLOCK 4 -- DI-MCCR-80025

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- DD

BLOCK 8 -- A
 BLOCK 10 -- ONE/R
 BLOCK 12 -- See Item 16
 BLOCK 13 -- See Item 16
 BLOCK 15 -- Total
 BLOCK 16 -- Draft due 45 days after contract award. Allow 30 days for Gov't review/comments. Revised SRS due 30 days after receipt of Gov't comments. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@1310.00

Paragraph 10.2.7 of the Software Requirements Specification DID # DI-MCCR-80025 is excluded from this statement of work and do not form a part of the required effort under this contract.

@1400.00 Interface Requirements Specification

An Interface Requirements Specification (IRS) shall be prepared in accordance with paragraph 3.4.2 and 3.4.7.1 of MIL-STD-483, paragraph 3.1.3.2.5.2 of MIL-STD-490, paragraphs 5.1.1.6 and 5.1.2.4 of DOD-STD-2167, and DID # DI-MCCR-80026. The IRS shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 1400.00

BLOCK 2 -- Interface Requirements Specification
 BLOCK 3 -- IRS
 BLOCK 4 -- DI-MCCR-80026
 BLOCK 6 -- STRBE-TQR
 BLOCK 7 -- DD
 BLOCK 8 -- A
 BLOCK 10 -- ONE/R
 BLOCK 12 -- See Item 16
 BLOCK 13 -- See Item 16
 BLOCK 15 -- Total
 BLOCK 16 -- Draft due 60 days after contract award. Allow 30 days for Gov't review/comments. Revised IRS due 30 days after receipt of Gov't comments. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@1410.00

An Interface Design Document (IDD) is required for each IRS.

@1415.00

Paragraphs 10.2.5.1 and 10.2.5.2 of the Interface Requirements Specification DID # DI-MCCR-80026 are excluded from this statement of work and do not form a part of the required effort under this contract.

@1420.00

Paragraphs 10.2.5.3, 10.2.6.1, and 10.2.6.2 of the Interface Requirements Specification DID # DI-MCCR-80026 are excluded from this statement of work and do not form a part of the required effort under this contract.

@1500.00 Software Top Level Design Document

A Software Top Level Design Document (STLDD) shall be prepared in accordance with paragraph 3.4.7.2 of MIL-STD-483, paragraph 3.1.3.3.5.1 of MIL-STD-490, paragraphs 5.2.1.2 and 5.2.2.3 of DOD-STD-2167, and DID # DI-MCCR-80012. The STLDD shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 1500.00

BLOCK 2 -- Software Top Level Design Document

BLOCK 3 -- STLDD

BLOCK 4 -- DI-MCCR-80012

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- DD

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Draft due 90 days prior to PDR. Allow 30 days for Gov't review/comments. Revised STLDD due 30 days prior to PDR. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@1510.00

Paragraphs 10.2.5.4, 10.2.5.5, and 10.2.5.6 of the Software Top Level Design Document DID # DI-MCCR-80012 are excluded from this statement of work and do not form a part of the required effort under this contract.

@1520.00

Paragraph 10.2.5.7 of the Software Top Level Design Document DID # DI-MCCR-80012 is excluded from this statement of work and do not form a part of the required effort under this contract.

@1600.00 Software Detailed Design Document

A Software Detailed Design Document (SDDD) shall be prepared in accordance with paragraph 3.4.7.2 of MIL-STD-483, paragraph 3.1.3.3.5.2 of MIL-STD-490, paragraphs 5.3.1.2 and 5.3.2.3 of DOD-STD-2167, and DID # DI-MCCR-80031. The SDDD shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 1600.00

BLOCK 2 -- Software Detailed Design Document

BLOCK 3 -- SDDD

BLOCK 4 -- DI-MCCR-80031

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- DD

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Draft due 90 days prior to CDR. Allow 30 days for Government review/comments. Revised SDDD due 30 days prior to CDR. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@1610.00

Paragraphs 10.2.5.1, 10.2.5.2, and 10.2.5.3 of the Software Detailed Design Document DID # DI-MCCR-80031 are excluded from this statement of work

and do not form a part of the required effort under this contract.

@1700.00 Interface Design Document

An Interface Design Document (IDD) shall be prepared in accordance with paragraph 3.4.7.2 of MIL-STD-483, paragraph 3.1.3.3.5.4 of MIL-STD-490, paragraphs 5.3.1.2 and 5.3.2.4 of DOD-STD-2167, and DID # DI-MCCR-80027.

The IDD shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 1700.00

BLOCK 2 -- Interface Design Document

BLOCK 3 -- IDD

BLOCK 4 -- DI-MCCR-80027

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- DD

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Draft due 90 days prior to CDR. Allow 30 days for Government review/comments. Revised IDD due 30 days prior to CDR.

Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@1710.00

Paragraph 10.2.5 of the Interface Design Document DID # DI-MCCR-80027 is excluded from this statement of work and do not form a part of the required effort under this contract.

@1800.00 Data Base Design Document

A Data Base Design Document (DBDD) shall be prepared in accordance with paragraph 3.4.7.2 of MIL-STD-483, paragraph 3.1.3.3.5.3 of MIL-STD-490, paragraphs 5.3.1.2 and 5.3.2.5 of DOD-STD-2167, and DID # DI-MCCR-80028.

The DBDD shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 1800.00

BLOCK 2 -- Data Base Design Document

BLOCK 3 -- DBDD

BLOCK 4 -- DI-MCCR-80028

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- DD

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Draft due 90 days prior to CDR. Allow 30 days for Government review/comments. Revised DBDD due 30 days prior to CDR.

Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@1810.00

Paragraphs 10.2.5.1 and 10.2.5.4 of the Data Base Design Document DID # DI-MCCR-80028 are excluded from this statement of work and do not form a

part of the required effort under this contract.

@1900.00 Software Product Specification

A Software Product Specification (SPS) shall be prepared in accordance with paragraph 3.4.7.3 of MIL-STD-483, paragraph 3.1.3.3.5 of MIL-STD-490, paragraph 5.6.2.5 of DOD-STD-2167, and DID # DI-MCCR-80029. The SPS shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 1900.00

BLOCK 2 -- Software Product Specification

BLOCK 3 -- SPS

BLOCK 4 -- DI-MCCR-80029

BLOCK 5 -- STRBE-TQR

BLOCK 7 -- DD

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Draft due 90 days prior to PCA. Allow 30 days for Government review/comments. Final SPS due 30 days prior to PCA. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@2000.00 Software Test Plan

A Software Test Plan (STP) shall be prepared in accordance with paragraphs 5.2.1.6 and 5.2.2.5 of DOD-STD-2167, and DID # DI-MCCR-80014. The STP shall be submitted in accordance with the CDRL.

^ 1123 data for paragraph 2000.00

BLOCK 2 -- Software Test Plan

BLOCK 3 -- STP

BLOCK 4 -- DI-MCCR-80014

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- DD

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Draft due 90 days prior to PDR. Allow 30 days for Government review/comments. Revised STP due 30 days prior to PDR. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@2010.00

Paragraphs 10.2.5.1 and 10.2.5.2 of the Software Test Plan DID # DI-MCCR-80014 are excluded from this statement of work and do not form a part of the required effort under this contract.

@2020.00

Paragraphs 10.2.5.3, 10.2.6.1, and 10.2.6.10 of the Software Test Plan DID # DI-MCCR-80014 are excluded from this statement of work and do not form

a part of the required effort under this contract.

@2030.00

Paragraphs 10.2.6.1, 10.2.6.3, 10.2.6.5, and 10.2.6.6 of the Software Test Plan DID # DI-MCCR-80014 are excluded from this statement of work and do not form a part of the required effort under this contract.

@2040.00

Paragraph 10.2.6.7 of the Software Test Plan DID # DI-MCCR-80014 is excluded from this statement of work and do not form a part of the required effort under this contract.

@2100.00 Software Test Description

A Software Test Description (STD) shall be prepared in accordance with paragraphs 5.3.1.14 and 5.3.2.8 of DOD-STD-2167, and DID # DI-MCCR-80015. The STD shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 2100.00

BLOCK 2 -- Software Test Description

BLOCK 3 -- STD

BLOCK 4 -- DI-MCCR-80015

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- DD

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Draft due 90 days prior to the CDR. Allow 30 days for Government review/comments. Revised STD due 30 prior to CDR. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@2200.00 Software Test Procedure

A Software Test Procedure (STPR) shall be prepared in accordance with paragraphs 5.4.1.12 and 5.4.2.6 of DOD-STD-2167, and DID # DI-MCCR-80016. The STPR shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 2200.00

BLOCK 2 -- Software Test Procedure

BLOCK 3 -- STPR

BLOCK 4 -- DI-MCCR-80016

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- DD

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Draft due 90 days prior to the start of integration testings. Allow 30 days for Government review/comments. Revised STPR due 30 days after receipt of Government comments. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@2300.00 Software Test Report

A Software Test Report (STR) shall be prepared in accordance with paragraphs 5.6.1.3 and 5.6.2.3 of DOD-STD-2167, and DID # DI-MCCR-80017. The STR shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 2300.00

BLOCK 2 -- Software Test Report

BLOCK 3 -- STR

BLOCK 4 -- DI-MCCR-80017

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- DD

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Each STR due 30 days after completion of test. All STR due 30 days prior to PCA. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@2400.00 Software User's Manual

A Software User's Manual (SUM) shall be prepared in accordance with paragraphs 5.2.1.8 and 5.2.2.6 of DOD-STD-2167 and DID # DI-MCCR-80019. The SUM shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 2400.00

BLOCK 2 -- Software User's Manual

BLOCK 3 -- SUM

BLOCK 4 -- DI-MCCR-80019

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- DD

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Draft due 30 days prior to PDR. Revised SUM due 30 days prior to PCA. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@2410.00

Paragraphs 10.2.5, 10.2.6, and 10.2.7 of the Software Users Manual DID # DI-MCCR-80019 are excluded from this statement of work and do not form a part of the required effort under this contract.

@2500.00 Version Description Document

A Version Description Document (VDD) shall be prepared in accordance with paragraph 80.5.4 (Appendix VIII), of MIL-STD-483, paragraphs 5.6.1.5 and 5.6.2.6 of DOD-STD-2167 and DID # DI-MCCR-80013. The VDD shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 2500.00

BLOCK 2 -- Version Description Document

BLOCK 3 -- VDD
 BLOCK 4 -- DI-MCCR-80013
 BLOCK 6 -- STRBE-TQR
 BLOCK 7 -- DD
 BLOCK 8 -- A
 BLOCK 10 -- ONE/R
 BLOCK 12 -- See Item 16
 BLOCK 13 -- See Item 16
 BLOCK 15 -- Total
 BLOCK 16 -- Draft due 90 days prior to PCA. Allow 30 days for Government review/comments. Revised VDD due 30 days prior to PCA. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@2600.00 Operational Concept Document

An Operational Concept Document (OCD) shall be prepared in accordance with paragraphs 5.1.1.4 and 5.1.2.2 of DOD-STD-2167 and DID # DI-MCCR-80023. The OCD shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 2600.00

BLOCK 2 -- Operational Concept Document
 BLOCK 3 -- OCD
 BLOCK 4 -- DI-MCCR-80023
 BLOCK 6 -- STRBE-TQR
 BLOCK 7 -- DD
 BLOCK 8 -- A
 BLOCK 10 -- ONE/R
 BLOCK 12 -- See Item 16
 BLOCK 13 -- See Item 16
 BLOCK 15 -- Total
 BLOCK 16 -- Draft due 60 days after contract award. Allow 30 days for Government review/comments. Revised OCD due 30 days after receipt of Government comments. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@2610.00

Paragraph 10.2.7 of the Operational Concept Document DID # DI-MCCR-80023 is excluded from this statement of work and do not form a part of the required effort under this contract.

@2700.00 Computer Resources Integrated Support Document

A Computer Resources Integrated Support Document (CRISD) shall be prepared in accordance with paragraphs 5.2.1.10 and 5.2.2.6 of DOD-STD-2167 and DID # DI-MCCR-80024. The CRISD shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 2700.00

BLOCK 2 -- Computer Resources Integrated Support Document
 BLOCK 3 -- CRISD
 BLOCK 4 -- DI-MCCR-80024
 BLOCK 6 -- STRBE-TQR
 BLOCK 7 -- DD
 BLOCK 8 -- A
 BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16
 BLOCK 13 -- See Item 16
 BLOCK 15 -- Total
 BLOCK 16 -- Draft due 45 days prior to PDR. Revised CRISD due 30 days prior to CDR. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@2800.00 Technical Reviews

The purposes of design reviews, audits, and meetings are to review the system requirements and capabilities and to review the contractor's system engineering efforts as the Software Development Program proceeds from conceptual development to operational deployment. Unless otherwise stated, all reviews, audits, and meetings shall be conducted in the contractor's facilities. The following table outlines the review process and responsibilities:

Specification -----	Event -----	Presenter -----
Type A Specification	SDR	Contractor
Type B Specification	PDRs	Contractor
Draft Type C Specification	CDRs	Contractor
Final Type C Specification	PCAs	Government

The contractor shall plan, support, and conduct the following reviews, audits, and meetings as described below.

@3000.00 System Design Review (SDR)

The purpose of the SDR is to review and validate the Type A Specification provided by the contractor. The contractor shall present the Type A Specification and demonstrate that all requirements have been properly translated to the specification. The preliminary Software Requirements and Interface Requirements Specifications will be reviewed and validated. The contractor shall present the preliminary Software Requirements Specification and the preliminary Interface Requirements Specification and compare them with the System/Segment specifications to determine if all requirements which are to be fulfilled by the software have been properly translated to these software specifications. The final Functional Baseline will be composed of the completed System Specification (i.e., the approved Type A Specification). This approval will occur upon conclusion of the System Design Review and will mark the point at which the System Specification will establish the Functional Baseline. The contractor shall also present, during the SDR, briefings in accordance with Appendix B of MIL-STD-1521B as applicable.

The SDR shall be documented in accordance with DID # DI-A-7088,

"Conference Agenda" (SDR); DID # DI-E-5423, "Design Review Data Package" (SDR), changing "MIL-STD-1521 and Appendices B, C, D, and G", in item 10 of DI-E-5423 to "MIL-STD-1521B and Appendix B", and with DID # DI-A-7089, "Conference Minutes" (SDR). All DIDs shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 3000.00

BLOCK 2 -- Conference Agenda
BLOCK 3 -- System Design Review (SDR)
BLOCK 4 -- DI-A-7088
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- LT
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 -- (a) Draft to Government by 35 days before initiation of SDR
(b) Comments to contractor 15 days after receipt by Gov't
(c) Revised to Gov't by 5 days before initiation of SDR
Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

BLOCK 2 -- Conference Minutes
BLOCK 3 -- System Design Review (SDR)
BLOCK 4 -- DI-A-7089
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- LT
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 -- Submittal to Government by 15 days after conclusion of SDR.
Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

BLOCK 2 -- Design Review Data Package (SDR)
BLOCK 4 -- DI-E-5423
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- LT
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 -- Submitted to Government by 30 days prior to SDR.
Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@3010.00

The following paragraphs of Appendix B of MIL-STD-1521 are excluded from the SDR briefings:

(a) 20.3.1 n

- (b) 20.3.1 p
- (c) 20.3.1 x
- (d) 20.3.2 j
- (e) 20.3.2 n
- (f) 20.3.7 k (2)
- (g) 20.3.11
- (h) 20.3.12.13

@3200.00 Software Specification Review (SSR)

The SSR shall be a formal review of a CSCI's requirements as specified in the Software Requirements Specification and the Interface Requirements Specification(s). Normally, it shall be held after System Design Review but prior to the start of CSCI preliminary design. A collective SSR for a group of configuration items, treating each configuration item individually, may be held when such an approach is advantageous to the contracting agency. Its purpose is to establish the Allocated Baseline for preliminary CSCI design by demonstrating to the contracting agency the adequacy of the Software Requirements Specification (SRS), Interface Requirements Specification(s) (IRS), and Operational Concept Document (OCD). The contractor shall present briefings and the items to be reviewed at the SSR as specified in Appendix C of MIL-STD-1521B as applicable.

The SSR shall be documented in accordance with DID # DI-A-7088, "Conference Agenda" (SSR); with DID # DI-E-5423, "Design Review Data Package" (SSR), changing "MIL-STD-1521 and Appendices B, C, D, and G", in item 10 of DI-E-5423 to "MIL-STD-1521B and Appendix C", and with DID # DI-A-7089, "Conference Minutes" (SSR). All DIDs shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 3200.00

- BLOCK 2 -- Conference Agenda
- BLOCK 3 -- Software Specification Review (SSR)
- BLOCK 4 -- DI-A-7088
- BLOCK 6 -- STRBE-TQR
- BLOCK 7 -- LT
- BLOCK 8 -- A
- BLOCK 10 -- ONE/R
- BLOCK 12 -- See Item 16
- BLOCK 13 -- See Item 16
- BLOCK 15 -- Total
- BLOCK 16 -- (a) Draft to Government by 35 days before initiation of SSR
 - (b) Comments to contractor 15 days after receipt by Gov't
 - (c) Revised to Gov't by 5 days before initiation of SSR
 Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

- BLOCK 2 -- Conference Minutes
- BLOCK 3 -- Software Specification Review (SSR)
- BLOCK 4 -- DI-A-7089
- BLOCK 6 -- STRBE-TQR
- BLOCK 7 -- LT
- BLOCK 8 -- A
- BLOCK 10 -- ONE/R
- BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 -- Submitted to Government by 15 days after conclusion of SSR.
Changes/Revisions shall be submitted as change pages for
approval. Reproducible; Electronic Media.

BLOCK 2 -- Design Review Data Package (SSR)
BLOCK 4 -- DI-E-5423
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- LT
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 -- Submitted to Government by 30 days prior to SSR.
Changes/Revisions shall be submitted as change pages for
approval. Reproducible; Electronic Media.

@3300.00 Preliminary Design Reviews (PDRs)

PDRs are the initial technical reviews of the major Configuration Items (CIs). PDRs will be used to review and validate the Type B Specifications and supporting documentation submitted by the contractor. The contractor shall conduct the PDRs to demonstrate that all the functions of the System Specification have been properly, completely, and accurately allocated to the CI specifications. The initial Allocated Baseline is set upon conclusion of the initial PDR. The Allocated Baseline will be expanded incrementally (i.e., by the addition of Type B Specifications as they are approved) and subjected to Government configuration control with the conclusion of each PDR. Each PDR will be held for the purpose of approving Type B Specifications for one or more individual major CIs. Changes to individual approved Type B Specifications must be made through the Engineering Change Proposal process. The final Allocated Baseline will be set upon successful conclusion of the last PDR. This will mark the point at which the entire package of approved Type B Specifications will be termed the Development Specification which forms the Allocated Baseline. The contractor shall also present, during the PDR, briefings on the subjects specified in Appendix D of MIL-STD-1521B as applicable.

The PDR shall be documented in accordance with DID # DI-A-7088, "Conference Agenda" (PDR); with DID # DI-E-5423, "Design Review Data Package" (PDR), changing "MIL-STD-1521 and Appendices B, C, D, and G", in item 10 of DI-E-5423 to "MIL-STD-1521B and Appendix D", and with DID # DI-A-7089, "Conference Minutes" (PDR). All DIDs shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 3300.00
BLOCK 2 -- Conference Agenda
BLOCK 3 -- Preliminary Design Review (PDR)
BLOCK 4 -- DI-A-7088
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- LT
BLOCK 8 -- A
BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 -- (a) Draft to Government by 35 days before initiation of PDR
(b) Comments to contractor 15 days after receipt by Gov't
(c) Revised to Gov't by 5 days before initiation of PDR
Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

BLOCK 2 -- Conference Minutes
BLOCK 3 -- Preliminary Design Review (PDR)
BLOCK 4 -- DI-A-7089
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- LT
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 -- Submitted to Government by 15 days after conclusion of PDR.
Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

BLOCK 2 -- Design Review Data Package (PDR)
BLOCK 4 -- DI-E-5423
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- LT
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 -- Submitted to Government by 30 days prior to PDR.
Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@3400.00 Critical Design Reviews (CDRs)

CDRs are held to review and validate the specific system design before detailed coding of software is begun. The contractor shall present the draft Type C Specification(s) at each CDR which will be reviewed to ensure that the functions allocated by the Development Specification are properly addressed at the "build to" level. Coding of software for each draft Type C Specification, approved upon conclusion of an individual CDR, will then begin. One of the purposes of the CDR is to assure that each Type C Specification is supportive of and consistent with previously approved Type B Specifications. The contractor shall also present, at the CDRs, briefings on the subjects specified in Appendix E of MIL-STD-1521B as applicable.

The CDR shall be documented in accordance with DID # DI-A-7088, "Conference Agenda" (CDR); with DID # DI-E-5423, "Design Review Data Package" (CDR), changing "MIL-STD-1521 and Appendices B, C, D, and G", in item 10 of DI-E-5423 to "MIL-STD-1521B and Appendix E", and with DID # DI-A-7089,

"Conference Minutes" (CDR). All DIDs shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 3400.00

BLOCK 2 -- Conference Agenda
BLOCK 3 -- Critical Design Review (CDR)
BLOCK 4 -- DI-A-7088
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- LT
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 -- (a) Draft to Government by 35 days before initiation of CDR
(b) Comments to contractor 15 days after receipt by Gov't
(c) Revised to Gov't by 5 days before initiation of CDR
Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

BLOCK 2 -- Conference Minutes
BLOCK 3 -- Critical Design Review (CDR)
BLOCK 4 -- DI-A-7089
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- LT
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 -- Submitted to Government by 15 days after conclusion of CDR.
Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

BLOCK 2 -- Design Review Data Package (CDR)
BLOCK 4 -- DI-E-5423
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- LT
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 -- Submitted to Government by 30 days prior to CDR.
Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@3500.00 Test Readiness Review (TRR)

The TRR shall be a formal review of the contractor's readiness to begin formal CSCI testing. It is conducted after software test procedures are available and CSC integration testing is complete. The purpose of the TRR is for the contracting agency to determine whether the contractor is in fact ready to begin CSCI testing. A technical understanding shall be reached on the informal test results, and on the validity and the degree of

completeness of the Computer System Operator's Manual (CSOM), Software User's Manual (SUM), and Computer System Diagnostic Manual (CSDM). The contractor shall present briefings and the items are to be reviewed at the TRR as specified in Appendix F of MIL-STD-1521B as applicable.

The TRR shall be documented in accordance with DID # DI-A-7088, "Conference Agenda" (TRR); with DID # DI-E-5423, "Design Review Data Package" (TRR), changing "MIL-STD-1521 and Appendices B, C, D, and G", in item 10 of DI-E-5423 to "MIL-STD-1521B and Appendix F", and with DID # DI-A-7089, "Conference Minutes" (TRR). All DIDs shall be submitted in accordance with the CDRL.

^ 1123 Data for paragraph 3500.00

BLOCK 2 -- Conference Agenda
BLOCK 3 -- Test Readiness Review (TRR)
BLOCK 4 -- DI-A-7088
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- LT
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 -- (a) Draft to Government by 35 days before initiation of TTR
(b) Comments to contractor 15 days after receipt by Gov't
(c) Revised to Gov't by 5 days before initiation of TTR
Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

BLOCK 2 -- Conference Minutes
BLOCK 3 -- Test Readiness Review (TRR)
BLOCK 4 -- DI-A-7089
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- LT
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 -- Submitted to Government by 15 days after conclusion of TRR.
Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

BLOCK 2 -- Design Review Data Package (TRR)
BLOCK 4 -- DI-E-5423
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- LT
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 -- Submitted to Government by 30 days prior to TRR.
Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@3600.00 Functional Configuration Audit (FCA)

The objective of the Functional Configuration Audit (FCA) shall be to verify that the configuration item's actual performance complies with its hardware development or Software Requirements and Interface Requirements Specifications. Test data shall be reviewed to verify that the hardware or computer software performs as required by its functional/allocated configuration identification. For configuration items developed at Government expense, an FCA shall be a prerequisite to acceptance of the configuration item. For software, a technical understanding shall be reached on the validity and the degree of completeness of the Software Test Reports, and as appropriate, Computer System Operator's Manual (CSOM), Software User's Manual (SUM), and Computer System Diagnostic Manual (CSDM). The contractor shall present briefings and the items to be reviewed at the FCA as specified in Appendix G of MIL-STD-1521B as applicable.

The FCA shall be documented in accordance with DID # DI-A-7088, "Conference Agenda" (FCA); with DID # DI-E-5423, "Design Review Data Package" (FCA), changing "MIL-STD-1521 and Appendices B, C, D, and G", in item 10 of DI-E-5423 to "MIL-STD-1521B and Appendix G", and with DID # DI-A-7089, "Conference Minutes" (FCA). All DIDs shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 3600.00

BLOCK 2 -- Conference Agenda
BLOCK 3 -- Functional Configuration Audit (FCA)
BLOCK 4 -- DI-A-7088
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- LT
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 -- (a) Draft to Government by 35 days before initiation of FCA
(b) Comments to contractor 15 days after receipt by Gov't
(c) Revised to Gov't by 5 days before initiation of FCA
Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

BLOCK 2 -- Conference Minutes
BLOCK 3 -- Functional Configuration Audit (FCA)
BLOCK 4 -- DI-A-7089
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- LT
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 -- Submitted to Government by 15 days after conclusion of FCA.
Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

BLOCK 2 -- Design Review Data Package (FCA)

BLOCK 1 -- DI-E-5423
 BLOCK 6 -- STRBE-TQR
 BLOCK 7 -- LT
 BLOCK 8 -- A
 BLOCK 10 -- ONE/R
 BLOCK 12 -- See Item 16
 BLOCK 13 -- See Item 16
 BLOCK 15 -- Total
 BLOCK 16 -- Submitted to Government by 30 days prior to FCA.
 Changes/Revisions shall be submitted as change pages for
 approval. Reproducible; Electronic Media.

@3700.00 Physical Configuration Audits (PCAs)

PCAs are formal technical examinations of CIs to ensure that each
 CI complies with the technical documentation and the provisional Type C
 Specifications. Each provisional Type C Specification will become a final
 Type C Specification upon approval during the PCA in which it is reviewed.
 Upon successful completion of the final FCA and PCA, the Product Specification
 (composed of all final Type C Specifications) will be approved and become the
 final Product Baseline. The contractor shall support this effort in accordance
 with Appendix H of MIL-STD-1521B, as applicable.

The CDR shall be documented in accordance with DID # DI-A-7088,
 "Conference Agenda" (PCA); with DID # DI-E-5423, "Design Review Data Package"
 (PCA), changing "MIL-STD-1521 and Appendices B, C, D, and G", in item 10 of
 DI-E-5423 to "MIL-STD-1521B and Appendix H", and with DID # DI-A-7089,
 "Conference Minutes" (PCA). Documentation shall be submitted in accordance
 with the CDRL.

^ 1423 Data for paragraph 3700.00

BLOCK 2 -- Conference Agenda
 BLOCK 3 -- Physical Configuration Audit (PCA)
 BLOCK 4 -- DI-A-7088
 BLOCK 6 -- STRBE-TQR
 BLOCK 7 -- LT
 BLOCK 8 -- A
 BLOCK 10 -- ONE/R
 BLOCK 12 -- See Item 16
 BLOCK 13 -- See Item 16
 BLOCK 15 -- Total
 BLOCK 16 -- (a) Draft to Government by 35 days before initiation of PCA
 (b) Comments to contractor 15 days after receipt by Gov't
 (c) Revised to Gov't by 5 days before initiation of PCA
 Changes/Revisions shall be submitted as change pages for
 approval. Reproducible; Electronic Media.

BLOCK 2 -- Conference Minutes
 BLOCK 3 -- Physical Configuration Audit (PCA)
 BLOCK 4 -- DI-A-7089
 BLOCK 6 -- STRBE-TQR
 BLOCK 7 -- LT
 BLOCK 8 -- A
 BLOCK 10 -- ONE/R
 BLOCK 12 -- See Item 16

BLOCK 12 -- See Item 16
 BLOCK 15 -- Total
 BLOCK 16 -- Submitted to Government by 15 day after conclusion of PCA.
 Changes/Revisions shall be submitted as change pages for
 approval. Reproducible; Electronic Media.

BLOCK 2 -- Design Review Data Package (PCA)
 BLOCK 4 -- DI-E-5423
 BLOCK 6 -- STRBE-TQR
 BLOCK 7 -- LT
 BLOCK 8 -- A
 BLOCK 10 -- ONE/R
 BLOCK 12 -- See Item 16
 BLOCK 13 -- See Item 16
 BLOCK 15 -- Total
 BLOCK 16 -- Submitted to Government by 30 days prior to PCA.
 Changes/Revisions shall be submitted as change pages for
 approval. Reproducible; Electronic Media.

@3800.00 Formal Qualification Review (FQRs)

The objective of the FQR shall be to verify that the actual performance of the configuration items of the system, as determined through tests, comply with the hardware Development Specification, Software Requirements, and Interface Requirements Specifications, and to identify the test report(s) and data which document results of qualification tests of the configuration items. The point of Government certification will be determined by the contracting agency and will depend upon the nature of the program, risk aspects of the particular hardware and software, and contractor progress in successfully verifying the requirements of the configuration items. When feasible, the FQR shall be combined with the FCA at the end of configuration item or subsystem testing, prior to PCA. If sufficient test results are not available at the FCA to insure the configuration items will perform in their system environment, the FQR shall be conducted during System testing (post PCA) whenever the necessary tests have been successfully completed to enable certification of configuration items. For non-combined FCA-FQRs, traceability, correlation, and completeness of the FQR shall be maintained with the FCA and duplication of effort avoided. The contractor shall present briefings and the items to be reviewed at the FQR as specified in Appendix I of MIL-STD-1521B as applicable.

The FQR shall be documented in accordance with DID # DI-A-7088, "Conference Agenda" (FQR); with DID # DI-E-5423, "Design Review Data Package" (FQR), changing "MIL-STD-1521 and Appendices B, C, D, and G", in item 10 of DI-E-5423 to "MIL-STD-1521B and Appendix I", and with DID # DI-A-7089, "Conference Minutes" (FQR). All DIDs shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 3800.00
 BLOCK 2 -- Conference Agenda
 BLOCK 3 -- Formal Qualification Review (FQR)
 BLOCK 4 -- DI-A-7088
 BLOCK 6 -- STRBE-TQR

BLOCK 7 -- LT
 BLOCK 8 -- A
 BLOCK 10 -- CNE/R
 BLOCK 12 -- See Item 16
 BLOCK 12 -- See Item 16
 BLOCK 15 -- Total
 BLOCK 16 -- (a) Draft to Government by 35 days before initiation of FQR
 (b) Comments to contractor 15 days after receipt by Gov't
 (c) Revised to Gov't by 5 days before initiation of FQR
 Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

BLOCK 2 -- Conference Minutes
 BLOCK 2 -- Formal Qualification Review (FQR)
 BLOCK 4 -- DI-A-7089
 BLOCK 6 -- STRPE-TQR
 BLOCK 7 -- LT
 BLOCK 8 -- A
 BLOCK 10 -- CNE/R
 BLOCK 12 -- See Item 16
 BLOCK 12 -- See Item 16
 BLOCK 15 -- Total
 BLOCK 16 -- Submitted to Government by 15 days after conclusion of FQR.
 Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

BLOCK 2 -- Design Review Data Package (FQR)
 BLOCK 4 -- DI-E-5123
 BLOCK 6 -- STRBF-TQR
 BLOCK 7 -- LT
 BLOCK 8 -- A
 BLOCK 10 -- CNE/R
 BLOCK 12 -- See Item 16
 BLOCK 12 -- See Item 16
 BLOCK 15 -- Total
 BLOCK 16 -- Submitted to Government by 30 days prior to FQR.
 Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@3900.00 Program Manager Meetings (PMMs)

PMMs allow the contractor's program manager and the Government's Program Manager to discuss program status and issues. PMMs will be held monthly or more often as program progress by the contractor dictates. PMMs will normally be at a contractor or subcontractor facility. PMM agenda, meeting place, time, and location will be established on an informal basis directly between the Government and contractor program managers.

Change pages or revisions to previously approved documents (SRS, SDP, STLD, SDDD, SUM, etc.) shall be presented to the Government Program Manager for approval at PMMs. The Government will review the proposed changes within 30 days. Approval or disapproval, along with any Government comments for the change page or revisions to the previously approved documents will be provided to the Government Program Manager at a Future PMM.

The contractor's program manager shall attend and support PMs.

^ 1123 data for paragraph 3900.00

BLOCK 2 -- Conference Minutes

BLOCK 3 -- Program Manager Meetings (PM)

BLOCK 4 -- DI-A-7089

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- LT

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Submitted to Government by 15 days after conclusion of PM.
Changes/Revisions shall be submitted as change pages for
approval. Reproducible; Electronic Media.

@4000.00 Technical Interchange Meetings (TIMs)

TIMs will be held at times and locations to be mutually agreed upon by the Government and the contractor. TIMs will be held as a result of a need for clarification of specifications, requirements, information, or data. The orientation of these meetings will be flexible and encompass design reviews, specification reviews, schedule reviews, and any other subject(s) deemed appropriate by mutual agreement of the contractor and the Government.

The contractor shall plan, support, and conduct TIMs. The contractor shall record the results of the TIMs in accordance with DID # DI-A-7089, "Conference Minutes" (TIMs), and the results shall be submitted in accordance with the CDRL.

^ 1123 data for paragraph 4000.00

BLOCK 2 -- Conference Minutes

BLOCK 3 -- Technical Interchange Meetings (TIM)

BLOCK 4 -- DI-A-7089

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- LT

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Submitted to Government by 15 days after conclusion of TIM.
Changes/Revisions shall be submitted as change pages for
approval. Reproducible; Electronic Media.

@4100.00 Preplanned Product Improvement (P3I)

Since research is ongoing, the need to implement P3I features could come at any point during the life cycle. As P3I features are approved and validated by the Government, they may be incorporated into the software. Consequently, the software design will explicitly include a means of incorporating P3I capabilities. Once the technology is available, and has been "reduced-to-practice," recommendations for implementing a feature will be assessed regarding cost, schedule, and performance impacts during the

Government's formal review process.

@4200.00 Engineering Change Proposal (ECP)

The contractor shall prepare an Engineering Change Proposal (ECP) in accordance with DOD-STD-480 or MIL-STD-481 and DID # DI-E-3128 to propose each change to the Government that impacts the CSCI's cost, schedule, interfaces, or Government-controlled baselines.

@4300.00 Specification Change Notice (SCN)

The contractor shall prepare a Specification Change Notice (SCN) in accordance with MIL-STD-490 and DID # DI-E-1126A to describe changes to Government-controlled baselines. Preliminary SCNs shall accompany ECPs as applicable. Additional guidance may be found in MIL-STD-483 and MIL-STD-490.

@4600.00 Software Programmer's Manual

A Software Programmer's Manual (SPM) shall be prepared in accordance with paragraphs 5.3.1.17 and 5.3.2.11 of DOD-STD-2167 and DID # DI-MCCR-80021. The SPM shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 4600.00

BLOCK 2 -- Software Programmer's Manual

BLOCK 3 -- SPM

BLOCK 4 -- DI-MCCR-80021

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- DD

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Draft due 15 days prior to PDR. Allow 15 days for Government review/comments. Final submitted NLT 30 days prior to CDR.

@4610.00

Paragraphs 10.2.5 and 10.2.6 of the Software Programmer's Manual DID # DI-MCCR-80021 are excluded from this statement of work and do not form a part of the required effort under this contract.

@4700.00 Computer System Operator's Manual

A Computer Support Operator's Manual (CSOM) shall be prepared in accordance with paragraphs 5.2.1.7 and 5.2.2.6 of DOD-STD-2167 and DID # DI-MCCR-80018. The CSOM shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 4700.00

BLOCK 2 -- Computer System Operator's Manual

BLOCK 3 -- CSOM

BLOCK 4 -- DI-MCCR-80018

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- DD

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 -- Draft due 15 days prior to PDR. Final due 30 days prior to
PCA

@4710.00

Paragraphs 10.2.5 and 10.2.6 of the Computer System Operator's Manual
DID # DI-MCCR-80018 are excluded from this statement of work and do not form
a part of the required effort under this contract.

@4800.00 Computer System Diagnostic Manual

A Computer Support Diagnostic Manual (CSDM) shall be prepared in
accordance with paragraphs 5.2.1.9 and 5.2.2.6 of DOD-STD-2167 and DID #
DI-MCCR-80020. The CSDM shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 4800.00

BLOCK 2 -- Computer System Diagnostic Manual
BLOCK 3 -- CSDM
BLOCK 4 -- DI-MCCR-80020
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- DD
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 -- Draft due 15 days prior to PDR. Final due 30 days prior to
PCA

@4810.00

Paragraph 10.2.5 of the Computer System Diagnostic Manual
DID # DI-MCCR-80020 is excluded from this statement of work and do not form
a part of the required effort under this contract.

@4820.00

Paragraph 10.2.6 of the Computer System Diagnostic Manual
DID # DI-MCCR-80020 is excluded from this statement of work and do not form
a part of the required effort under this contract.

@4900.00 Firmware Support Manual

A Firmware Support Manual (FSM) shall be prepared in accordance with
paragraphs 5.3.1.18 and 5.3.2.11 of DOD-STD-2167 and DID # DI-MCCR-80022.
The FSM shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 4900.00

BLOCK 2 -- Firmware Support Manual
BLOCK 3 -- FSM
BLOCK 4 -- DI-MCCR-80022
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- DD
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total

BLOCK 16 -- Draft due 90 days prior to CDR. Allow 15 days for Government review/comments. Final submitted NLT 30 days prior to CDR.

@4910.00

Paragraphs 10.2.5 and 10.2.6 of the Firmware Support Manual DID # DI-MCCR-80022 are excluded from this statement of work and do not form a part of the required effort under this contract.

@5100.00 Software Support Environment

The contractor shall establish and implement a program to define and provide a Software Support Environment in accordance with DOD-STD-1467 and as specified herein.

The contractor shall define a Developmental Software Support Environment (DSSE), shall ensure the compatibility of this environment with the contracting activity's designated Life Cycle Software Support Environment (LCSSSE), and shall ensure the existence of a complete contracting activity life cycle software support capability for the deliverable software of the contracted effort.

@5110.00 Developmental Software Support Environment Plan

A Developmental Software Support Environment Plan shall be prepared in accordance with DOD-STD-1467 and DID # DI-E-7140. The plan shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 5110.00

BLOCK 2 -- Development Software Support Environment Plan

BLOCK 3 -- DSSEP

BLOCK 4 -- DI-E-7140

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- DD

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Draft due 90 days prior to PDR. Allow 30 days for Government review/comments. Revised DSSEP due 30 days prior to PDR. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@5120.00 Documentation of Commercially Available or Privately Developed Software

Documentation of commercially available or privately developed software shall be in accordance with DOD-STD-1467 and DID # DI-E-7141. The documentation shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 5120.00

BLOCK 2 -- Documentation of Commercially Available or Privately Developed Software

BLOCK 4 -- DI-E-7141

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- DD

BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 -- Draft due 45 days prior to PDR. Revised documentation due 30 days prior to PCA. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@5130.00 Software Support Transition Plan

A Software Support Transition Plan shall be prepared in accordance with DOD-STD-1467 and DID # DI-E-7142. The plan shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 5130.00

BLOCK 2 -- Software Support Transition Plan
BLOCK 3 -- SSTP
BLOCK 4 -- DI-E-7142
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- DD
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 -- Draft due 90 days prior to PDR. Allow 30 days for Government review/comments. Revised SSTP due 30 days prior to CDR. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@5140.00 Life Cycle Software Support Environment Users Guide

A Life Cycle Software Support Environment Users Guide shall be prepared in accordance with DOD-STD-1467 and DID # DI-E-7143. The guide shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 5140.00

BLOCK 2 -- Life Cycle Software Support Environment Users Guide
BLOCK 3 -- LCSSEUG
BLOCK 4 -- DI-E-7143
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- DD
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 -- Draft due 45 days prior to PDR. Revised LCSSEUG due 30 days prior to PCA. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@5200.00 System Requirements Review (SRR)

The SRRs are normally conducted during the system Concept Exploration or Demonstration and Validation Phase. Such reviews normally will be conducted after the accomplishment of functional analysis and preliminary requirements allocation to determine initial direction and progress of the contractor's System Engineering Management effort and his convergence upon an optimum and complete configuration. The contractor shall present, at the SRR, briefings and review items on the subjects specified in Appendix A of MIL-STD-1521B as applicable.

The SRR shall be documented in accordance with DID # DI-A-7088, "Conference Agenda" (SRR); with DID # DI-E-5423, "Design Review Data Package" (SRR), changing "MIL-STD-1521 and Appendices B, C, D, and G", in item 10 of DI-E-5423 to "MIL-STD-1521B and Appendix A", and with DID # DI-A-7089, "Conference Minutes" (SRR). All DIDs shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 5200.00

BLOCK 2 -- Conference Agenda
BLOCK 3 -- System Requirements Review (SRR)
BLOCK 4 -- DI-A-7088
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- LT
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 -- (a) Draft to Government by 35 days before initiation of SSR
(b) Comments to contractor 15 days after receipt by Gov't
(c) Final to Gov't by 5 days before initiation of SSR
Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

BLOCK 2 -- Conference Minutes
BLOCK 3 -- System Requirements Review (SRR)
BLOCK 4 -- DI-A-7089
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- LT
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 -- Submitted to Government by 15 days after conclusion of SRR.
Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

BLOCK 2 -- Design Review Data Package (SRR)
BLOCK 4 -- DI-E-5423
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- LT
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total

BLOCK 16 -- Submitted to Government by 15 days prior to SRR.
Changes/Revisions shall be submitted as change pages for
approval. Reproducible; Electronic Media.

@5300.00 Software Quality Evaluation

The following paragraphs delineate the Government's requirements for ensuring the development of quality software.

The contractor shall establish and follow a Software Quality Evaluation Program in accordance with the requirements of paragraph 5.8, "Software Quality Evaluation," of DOD-STD-2167.

Any requirement for a specification, plan, manual, description, procedure, or document shall be omitted if it was not required in any subparagraph of 2.1, "Requirements for Software Development" of this SOW.

The contractor shall establish and follow a Software Quality Evaluation program in accordance with the requirements of DOD-STD-2167, AMC-P 702-XX, and as specified herein. This program shall include, but not be limited to, the following requirements described in DOD-STD-2167, identified DIDs, and the following Tasks of AMC-P 702-XX.

@5301.00 Task 101 Software Quality Assurance Program Establishment and Implementation

@5302.00 Task 103 Software Quality Requirements Review, and Inspection

@5303.00 Task 106 Documentation Review

@5304.00 Task 108 Technical Review and Audits

@5305.00 Task 201 Software Requirements Evaluation

@5306.00 Task 202 Software Requirements Traceability

@5307.00 Task 301 Design Analysis

@5308.00 Task 302 Design Traceability

@5309.00 Task 303 Interface Analysis

@5310.00 Task 401 Code-to-Design Traceability

@5311.00 Task 501 Unit, Module, and Subprogram Test and Evaluation

@5312.00 Task 502 Software Integration Test and Evaluation

@5313.00 Task 503 Software Performance Test and Evaluation

@5314.00 Specific details applicable to each of the above tasks are provided in the individual statements of work for each task.

NOTE: If any conflict between DOD-STD-2167 and AMC-P 702-XX exists,

then DOD-STD-2167 shall prevail.

@5400.00 Scope

The software development contractor shall provide technical expertise to conduct a Software Quality Evaluation (SQE) Program on all software including the support equipment software contained in the system. This support effort shall include assessment, verification and validation of software requirements, algorithm applicability, software design, interfaces, design implementation, computer program performance, development tests, test data, and software documentation. Software baseline control, software development methodology and activities, and product assurance activities also shall be assessed by the contractor's software quality evaluators.

@5500.00 Software Quality Assurance Program Establishment

The contractor shall establish and implement a Software Quality Assurance Program in accordance with paragraph 5.8 of DOD-STD-2167 and Task 101, "Software Quality Assurance Program Establishment and Implementation," of AMC-P 702-XX. It shall be documented according to DID # DI-S-30559, "Technical Operating Report (TOR) for Software Quality Assurance Program Establishment and Implementation", and submitted, in accordance with the CDRL, to the Government for approval. In addition to the requirements of the Software Quality Assurance Program described in paragraph 5.8 of DOD-STD-2167 and Task 101, the contractor shall include the following:

a. A detailed description of any other software quality activity not included in the tasks specified above but which are specified elsewhere in this contract.

b. A detailed description of other contractor initiated software quality activities, which are deemed necessary by the contractor, and which will be performed as part of the overall software quality effort.

@5600.00 Software Quality Requirements Review and Inspection

The contractor's software quality assurance organization shall perform a software quality requirements review and inspection of the Software Quality Program and the Software Quality Evaluation Plan in accordance with Task 103, "Software Quality Requirements Review and Inspection", of AMC-P 702-XX. A report in accordance with DID # DI-S-30559, "Technical Operating Report for Software Quality Requirements Review and Inspection", shall be prepared and shall be submitted in accordance with the CDRL.

^ 1423 data for paragraph 5600.00

BLOCK 2 -- Technical Operating Report (TOR) for Software

BLOCK 3 -- Quality Requirements Review and Inspection

BLOCK 4 -- DI-S-30559

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- LT

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Report due 45 days after publish draft SRS.

Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@5700.00 Documentation Review

The contractor software quality assurance organization shall perform software documentation reviews in accordance with Task 106, "Documentation Reviews", of AMC-P 702-XX.

The contractor's Software Quality Evaluator shall:

@5710.00 Review the updated Software Development Plan (SDP), Software Standards and Procedures Manual (SSPM), Software Configuration Management Plan (SCMP), and Software Quality Evaluation Plan (SQEP) for adherence to required format and documentation standards, compliance with contractual requirements, internal consistency, understandability, technical adequacy, appropriate degree of completeness, traceability to the SOW, consistency with each other, feasibility, appropriate level of detail and appropriate content for the intended audience.

@5720.00 Analyze the preliminary Software User's Manual (SUM) for adherence to required format and documentation standards; compliance with contractual requirements; internal consistency; understandability; technical adequacy; appropriate degree of completeness; traceability to the SRS and IRS; consistency with the STLDD, CSOM, and CSDM; appropriate level of detail; and appropriate content for intended audience.

@5730.00 Analyze the Software Product Specification (SPS) for adherence to required format and documentation standards; compliance with contractual requirements; internal consistency; understandability; technical adequacy; appropriate degree of completeness; traceability to the SRS; incorporation of STLDD, SDDD, DBDD, IDD, and software listings consistent with updated source code; and appropriate content for intended audience.

@5740.00 Analyze the Software Development Files (SDFS) for adherence to required format and documentation standards; compliance with contractual requirements; internal consistency; understandability; technical adequacy; appropriate degree of completeness; traceability to the SRS, IRS, and STP; consistency with the SDDD, IDD, and DBDD; feasibility; appropriate level of detail; appropriate allocation of timing and sizing resources; and appropriate content for intended audience.

@5750.00 Analyze the Software Test Plan (STP) for adherence to required format and documentation standards; compliance with contractual requirements; internal consistency; understandability; technical adequacy; appropriate degree of completeness; traceability to the SRS and IRS; consistency with the SDP; feasibility; appropriate level of detail; appropriate test coverage of requirements; adequacy of planned tools, facilities, procedures, methods, and resources; and appropriate content for intended audience.

@5760.00 Analyze the Software Test Description (STD) for adherence to required format and documentation standards; compliance with contractual requirements; internal consistency; understandability; technical adequacy; appropriate degree of completeness; traceability to the SRS, STP, and IRS; consistency with the SDDD, IDD, and DBDD; feasibility; appropriate level of

detail; adequate test coverage of requirements; adequacy of planned tools, facilities, procedures, methods, and resources; adequate detail in specifying CSCI test inputs, expected results, and evaluation criteria; and appropriate content for intended audience.

@5765.00 Analyze the Software Test Procedures (STPR) for adherence to required format and documentation standards; compliance with contractual requirements; internal consistency; understandability; technical adequacy; appropriate degree of completeness; traceability to the STP, STB, SRS, and IRS; consistency with the SDDD; feasibility; appropriate level of detail; adequate test coverage of requirements; adequacy of planned tools, facilities, procedures, methods, and resources; and appropriate content for intended audience.

@5770.00 Analyze the Software Test Report (STR) for adherence to required format and documentation standards; compliance with contractual requirements; internal consistency; understandability; technical adequacy; appropriate degree of completeness; traceability to the STP and STD; conformance to STP and expected results in STD; completeness of testing; acceptability of deviations; adequacy of retesting; adequacy of tested CPCI; appropriate allocation of sizing and timing resources; adequate test coverage of requirements; and appropriate content for intended audience.

@5780.00 All problem areas, potential problem areas, and errors shall be documented and proposed solutions developed and documented in a "Technical Operating Report for Documentation Review" (followed by the name of the document reviewed). This report shall be prepared according to DID # DI-S-30559 and submitted in accordance with the CDRL.

^ 1423 data for paragraph 5780.00

BLOCK 2 -- Technical Operating Report (TOR) for

BLOCK 3 -- Documentation Review

BLOCK 4 -- DI-S-30559

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- LT

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- A report shall be submitted for each software document produced.

The report is due 45 days after the release of the draft document called for in the MCCR category of 1423s.

Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@5800.00 Technical Reviews and Audits

The contractor's software quality assurance organization shall review the Software Development and Software Quality Evaluation plans, and participate in all contracted technical reviews and audits identified in this contract, in accordance with Task 108, "Technical Reviews and Audits", of AMC-P 702-XX and MIL-STD-1521.

@5900.00 Software Requirements Evaluation

The contractor's software quality assurance organization shall evaluate all software requirements to ensure that they are quantifiable, measurable, and testable in accordance with Task 201, "Software Requirements Evaluation", of AMC-P 702-XX.

The contractor's Software Quality Evaluators shall determine if the:

@5910.00 System/Segment Specification (SSS) has been prepared in accordance with paragraph 3.1.3.1 of MIL-STD-490, paragraphs 5.1 and 20.4.1 of DOD-STD-2167, DID # DI-CMAN-80008, and as required by the software development contract Statement of Work (SOW).

@5920.00 Software Requirements Specification (SRS) has been prepared in accordance with paragraphs 3.4.2 and 3.4.7.1 of MIL-STD-483, paragraph 3.1.3.2.5.1 of MIL-STD-490, paragraphs 5.1.1.6 and 5.1.2.4 of DOD-STD-2167, DID # DI-MCCR-80025, and the software development contract SOW.

@5930.00 Interface Requirements Specification (IRS) has been prepared in accordance with paragraphs 3.4.2 and 3.4.7.1 of MIL-STD-483, paragraph 3.1.3.2.5.2 of MIL-STD-490, paragraphs 5.1.1.6 and 5.1.2.4 of DOD-STD-2167, and the software development contract SOW.

@5940.00 Defined requirements are complete in scope, unambiguous, complementary, testable, feasible, consistent, and technically accurate.

@5950.00 All problem areas, potential problem areas, and errors shall be documented and proposed solutions developed and documented in a "Technical Operating Report for Software Requirements Evaluation" (DID # DI-S-30559) and submitted in accordance with the CDRL.

^ 1423 data for paragraph 5950.00

BLOCK 2 -- Technical Operating Report (TOR) for Software

BLOCK 3 -- Requirements Evaluation

BLOCK 4 -- DI-S-30559

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- LT

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Report due 45 days after publish draft SRS.

Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@6000.00 Software Requirements Traceability

The contractor's software quality assurance organization shall conduct a Software Requirements Traceability in accordance with Task 202, "Software Requirements Traceability", of AMC-P 702-XX. Analyze all Software Requirements Documentation (e.g., B5 Specs - Software Requirements Specification and Interface Requirements Specification) to ensure traceability to system level documents (System/Segment Specification). All problem areas, potential problem areas, and errors shall be documented and proposed solutions developed and documented in a "Technical Operating Report for Software Requirements Traceability" (DID # DI-S-30559) and submitted in accordance with the CDRL.

^ 1423 data for paragraph 6000.00
 BLOCK 2 -- Technical Operating Report (TOR) for Software
 BLOCK 3 -- Requirements Traceability
 BLOCK 4 -- DI-S-30559
 BLOCK 6 -- STRBE-TQR
 BLOCK 7 -- LT
 BLOCK 8 -- A
 BLOCK 10 -- ONE/R
 BLOCK 12 -- See Item 16
 BLOCK 13 -- See Item 16
 BLOCK 15 -- Total
 BLOCK 16 -- Report due 60 days after publish draft SRS.
 Changes/Revisions shall be submitted as change pages for
 approval. Reproducible; Electronic Media.

@6100.00 Design Analysis

The contractor's software quality assurance organization shall evaluate the software design during development to ensure that all software requirements are being satisfied, and that the design is being documented in accordance with DOD-STD-2167, related DIDs, and Task 301, "Design Analysis", of AMC-P 702-XY.

The contractor's Software Quality Evaluators shall:

- a. Analyze detailed software design (C5 Specifications - Software Top Level Design Document (STLDD), Software Detailed Design Document (SDDD), Interface Design Document (IDD), Data Base Design Document (DBDD), and Software Product Specification (SPS)) to ensure traceability to the software requirements (B5 Specifications - Software Requirements Specification (SRS) and Interface Requirements Specification (IRS)).
- b. Analyze the evolving Top Level Design and Software Top Level Design Documentation (STLDD) for adherence to required format and documentation standards, compliance with contractual requirements, internal consistency, understandability, technical adequacy, appropriate degree of completeness, traceability to the SRS and IRS, feasibility, appropriate design techniques used for the software, appropriate level of detail, appropriate allocation of sizing and timing resources, and appropriate content for intended audience.
- c. Analyze the evolving detailed design and the Software Detailed Design Document (SDDD) for adherence to required format and documentation standards; compliance with contractual requirements; internal consistency; understandability; technical adequacy; appropriate degree of completeness; traceability to the SRS, IRS, and STLDD; consistency with each other; feasibility; adequacy of planned tools, facilities, procedures, methods, and resources; appropriate content for intended audience. All problem areas, potential problem areas, and errors identified shall be documented and proposed solutions developed and documented in a "Technical Operating Report for a Quick Response Analysis of the Software Design" (DID # DI-S-30559) and submitted in accordance with the CDRL.
- d. Analyze the evolving data base design and Data Base Design Document (DBDD) for adherence to required format and documentation standards;

compliance with contractual requirements; internal consistency; understandability; technical adequacy; appropriate degree of completeness; traceability to the SRS, IRS, and STLDD; consistency with the SDDD and IDD; feasibility; appropriate design techniques; appropriate level of detail; appropriate allocation of sizing and timing resources; adequacy of planned tools, facilities, procedures, methods, and resources; consistency between data definition and data use; accuracy and required precision of constants; adequacy of backup procedures and mechanisms; appropriate content for intended audience. All problem areas, potential problem areas, and errors identified shall be documented and proposed solutions developed and documented in a "Technical Operating Report for a Quick Response Analysis of the Data Base Design Document" (DID # DI-S-30559) and submitted in accordance with the CDRL.

All problem areas, potential problem areas, and errors shall be documented and proposed solutions developed and documented in a "Technical Operating Report for Design Analysis" (DID # DI-S-30559) and submitted in accordance with the CDRL.

^ 1423 data for paragraph 6100.00

BLOCK 2 -- Technical Operating Report (TOR) for Design

BLOCK 3 -- Analysis

BLOCK 4 -- DI-S-30559

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- LT

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Report due 45 days after publish draft Software Detailed Design Document (SDDD). Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@6200.00 Design Traceability

The contractor's software quality assurance organization shall conduct a Design Traceability analysis in accordance with Task 302, "Design Traceability", of AMC-P 702-XX. All problem areas, potential problem areas, and errors shall be documented and proposed solutions developed and documented in a "Technical Operating Report for Design Traceability" (DID # DI-S-30559) and submitted in accordance with the CDRL.

^ 1423 data for paragraph 6200.00

BLOCK 2 -- Technical Operating Report (TOR) for Design

BLOCK 3 -- Traceability

BLOCK 4 -- DI-S-30559

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- LT

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Report due 45 days after publish draft Software Detailed Design Document (SDDD). Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@6300.00 Interface Analysis

The contractor's software quality assurance organization shall conduct an Interface Analysis in accordance with Task 303, "Interface Analysis", of AMC-P 702-XX. Analyze the evolving interface design for adherence to required format and documentation standards; compliance with contractual requirements; internal consistency; understandability; technical adequacy; appropriate degree of completeness; traceability to the SRS, IRS, and STLDD; consistency with the SDDD and DBDD; feasibility; appropriate design techniques; appropriate level of detail; appropriate allocation of sizing and timing resources; adequacy of planned tools, facilities, procedures, methods, and resources; appropriate content for intended audience. All problem areas, potential problem areas, and errors identified shall be documented and proposed solutions developed and documented in a "Technical Operating Report for a Quick Response Analysis of the Interface Design Document" (DID # DI-S-30559) and submitted in accordance with the CDRL.

^ 1423 data for paragraph 6300.00

BLOCK 2 -- Technical Operating Report (TOR) for Interface

BLOCK 3 -- Analysis

BLOCK 4 -- DI-S-30559

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- LT

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Report due 45 days after publish draft Software Detailed Design Document (SDDD) or Interface Design Document (IDD).

Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@6400.00 Code-to-Design Traceability

The contractor's software quality assurance organization shall perform a Code-to-Design Traceability analysis in accordance with Task 401, "Code-to-Design Traceability", of AMC-P 702-XX.

The contractor's Software Quality Evaluators shall:

a. Analyze the evolving and completed source code for each unit for adherence to required format and documentation standards; compliance with contractual requirements; internal consistency; understandability; technical adequacy; appropriate degree of completeness; traceability to the SDDD; and appropriate coding techniques used.

b. Verify source code compliance, with the Software Development Plan requirements, in all applicable areas such as listing format, structure commenting, naming conventions, coding standards, etc.

c. Independently assess software performance for selected areas of emphasis by using performance analysis tools on actual development contractor

coded modules. This includes inputting test case data that exercises not only normal cases, but also extreme (upper/lower boundaries) cases and exceptional cases (out of range); this case data includes volume and values. Recommend specific system level testing to verify and analyze critical software functions.

All problem areas, potential problem areas, and errors shall be documented and proposed solutions developed and documented in a "Technical Operating Report for Code-to-Design Traceability" (DID # DI-S-30559) and submitted in accordance with the CDRL.

^ 1423 data for paragraph 6400.00

BLOCK 2 -- Technical Operating Report (TOR) for Code-to-

BLOCK 3 -- Design Traceability

BLOCK 4 -- DI-S-30559

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- LT

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Report due 45 days after completion of source code and prior to start of CSCI testing. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@6500.00 Unit, Module, and Subprogram Test and Evaluation

The contractor's software quality assurance organization shall ensure that unit, module and subprogram tests are conducted in accordance with Task 501, "Unit, Module, and Subprogram Test and Evaluation", of AMC-P 702-XX. Analyze unit and CSC integration test procedures for adherence to required format and documentation standards; compliance with contractual requirements; internal consistency; understandability; technical adequacy; appropriate degree of completeness; traceability to the STP, CSC Test Cases, and Unit Test Cases; consistency with the SDDD; feasibility; appropriate level of detail; adequate test coverage of requirements; adequacy of planned tools, facilities, procedures, methods, and resources; and appropriate content for intended audience. All problem areas, potential problem areas, and errors shall be documented and proposed solutions developed and documented in a "Technical Operating Report for Unit, Module, and Subprogram Test and Evaluation" (DID # DI-S-30559) and submitted in accordance with the CDRL.

^ 1423 data for paragraph 6500.00

BLOCK 2 -- Technical Operating Report (TOR) for

BLOCK 3 -- Unit, Module, and Subprogram Test & Evaluation

BLOCK 4 -- DI-S-30559

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- LT

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Report due 30 days after completion of unit, module, and subprogram testing. Changes/Revisions shall be submitted as

change pages for approval. Reproducible; Electronic Media.

@6600.00 Software Integration Test and Evaluation

The contractor's software quality assurance organization shall ensure that Software Integration Test and Evaluation is conducted in accordance with Task 502, "Software Integration Test and Evaluation", of AMC-P 702-XX. All problem areas, potential problem areas, and errors shall be documented and proposed solutions developed and documented in a "Technical Operating Report for Software Integration Test and Evaluation" (DID # DI-S-30559) and submitted in accordance with the CDRL.

^ 1423 data for paragraph 6600.00

BLOCK 2 -- Technical Operating Report (TOR) for

BLOCK 3 -- Software Integration Test & Evaluation

BLOCK 4 -- DI-S-30559

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- LT

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Report due 30 days after completion of CSC and integration testing. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@6700.00 Software Performance Test and Evaluation

The contractor's software quality assurance organization shall ensure that software performance test and evaluation procedures are conducted in accordance with Task 503, "Software Performance Test and Evaluation", of AMC-P 702-XX. All problem areas, potential problem areas, and errors shall be documented and proposed solutions developed and documented in a "Technical Operating Report for Software Performance Test and Evaluation" (DID # DI-S-30559) and submitted in accordance with the CDRL.

^ 1423 data for paragraph 6700.00

BLOCK 2 -- Technical Operating Report (TOR) for

BLOCK 3 -- Software Performance Test & Evaluation

BLOCK 4 -- DI-S-30559

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- LT

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Report due 30 days after completion of CSC and integration testing. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@7000.00 Interface with Independent Verification and Validation Activities

The contractor's software quality assurance organization shall ensure adequate interface and data exchange between the contractor's software development organization in accordance with Task 111, "Interface with Independent Verification and Validation Activities", of AMC-P 702-XX. Compliance with Task 111 shall be documented in a "Technical Operating Report for Interface with IV&V Activities" (DID # DI-S-30559) and submitted in accordance with the CDRL.

^ 1423 data for paragraph 7000.00

BLOCK 2 -- Technical Operating Report (TOR) for
BLOCK 3 -- Interface with IV&V Activities
BLOCK 4 -- DI-S-30559
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- LT
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 -- Submittal to Government by initiation of CDR and each month thereafter. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@7100.00 Design Walk-Through

The contractor's software quality evaluators shall conduct Design Walk-Throughs periodically throughout the design phase to ensure that the development of the computer software is proceeding in accordance with the software development plan and the software being developed conforms to the software requirements. The design walk-through shall be conducted in accordance with Task 308, "Design Walkthrough", of AMC-P 702-XX. All problem areas, potential problem areas, and errors shall be documented and proposed solutions developed and documented in a "Technical Operating Report for Design Walk-Throughs" (DID # DI-S-30559) and submitted in accordance with the CDRL.

^ 1423 data for paragraph 7100.00

BLOCK 2 -- Technical Operating Report (TOR) for
BLOCK 3 -- Design Walk-Through
BLOCK 4 -- DI-S-30559
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- LT
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 -- Submittal to Government by 10 days after each Design Walk-Through. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@7200.00 Design Inspection

The contractor's software quality evaluators shall conduct Design Inspections on each Computer Software Configuration Item (CSCI) and of the software design as a whole in order to locate errors and ensure a quality product is being produced. The design inspection shall be conducted in accordance with Task 309, "Design Inspection", of AMC-P 702-XX. All problem areas, potential problem areas, and errors shall be documented and proposed solutions developed and documented in a "Technical Operating Report for Design Inspection" (DID # DI-S-30559) and submitted in accordance with the CDRL.

^ 1423 data for paragraph 7200.00

BLOCK 2 -- Technical Operating Report (TOR) for
BLOCK 3 -- Design Inspection
BLOCK 4 -- DI-S-30559
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- LT
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 -- Report submitted to Government 45 days after publish draft Software Detailed Design Document. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@7300.00 Source Code Analysis

The contractor's software quality evaluators shall analyze the source code for, but not limited to, properties such as complexity, consistency, and adherence to software development standards in order to ensure the production of quality software. The source code analysis shall be conducted in accordance with Task 402, "Source Code Analysis", of AMC-P 702-XX. All problem areas, potential problem areas, and errors shall be documented and proposed solutions developed and documented in a "Technical Operating Report for Source Code Analysis" (DID # DI-S-30559) and submitted in accordance with the CDRL.

^ 1423 data for paragraph 7300.00

BLOCK 2 -- Technical Operating Report (TOR) for
BLOCK 3 -- Source Code Analysis
BLOCK 4 -- DI-S-30559
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- LT
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 -- Report due 45 days after completion of Source code and prior to start of CSCI testing. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@7400.00 Code Walk-Through

The contractor's software quality evaluators shall conduct source code walk-throughs on each Computer Software Component (CSC) to ensure that the source code development is proceeding in accordance with the software development plan and the source code implements the software requirements. The code Walk-Through shall be conducted in accordance with Task 403, "Code Walkthrough", of AMC-P 702-XX. All problem areas, potential problem areas, and errors shall be documented and proposed solutions developed and documented in a "Technical Operating Report for Code Walk-Through" (DID # DI-S-30559) and submitted in accordance with the CDRL.

^ 1423 data for paragraph 7400.00

BLOCK 2 -- Technical Operating Report (TOR) for

BLOCK 3 -- Code Walk-Through

BLOCK 4 -- DI-S-30559

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- LT

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Submitted to Government by 10 days after each Code Walk-Through. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@7500.00 Code Inspection

The contractor's software quality evaluators shall conduct a code inspection on each Computer Software Component (CSC) in order to locate errors and ensure that quality code is being produced. The code inspection shall be conducted in accordance with Task 404, "Code Inspection", of AMC-P 702-XX. All problem areas, potential problem areas, and errors shall be documented and proposed solutions developed and documented in a "Technical Operating Report for Code Inspection" (DID # DI-S-30559) and submitted in accordance with the CDRL.

^ 1423 data for paragraph 7500.00

BLOCK 2 -- Technical Operating Report (TOR) for

BLOCK 3 -- Code Inspection

BLOCK 4 -- DI-S-30559

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- LT

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Report submitted to Government 45 days after completion of Source code and prior to start of CSCI testing. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@7600.00 Software Stress Testing

The contractor shall ensure that for certain periods during the software test, the software shall be required to operate at levels which stress the software's capabilities in terms of response times and data handling capacity. The software stress testing shall be conducted in accordance with Task 504, "Software Stress Testing", of AMC-P 702-XX. All problem areas, potential problem areas, and errors shall be documented and proposed solutions developed and documented in a "Technical Operating Report for Software Stress Testing" (DID # DI-S-30559) and submitted in accordance with the CDRL.

^ 1423 data for paragraph 7600.00

BLOCK 2 -- Technical Operating Report (TOR) for

BLOCK 3 -- Software Stress Testing

BLOCK 4 -- DI-S-30559

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- LT

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Report due 30 days after completion of stress testing and prior to start of CSCI testing. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@7700.00 Quality Assurance Program Status Reports

The contractor shall report on the status of the Quality Assurance Program to describe status, identify problems and solutions, and permit proper use and care of the product by the Government (DID # DI-S-30559).

^ 1423 data for paragraph 7700.00

BLOCK 2 -- Quality Assurance Program Status Reports

BLOCK 4 -- DI-S-30559

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- LT

BLOCK 8 -- A

BLOCK 9 -- 10

BLOCK 10 -- MTHLY

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Submitted to Government by initiation of the CDR and each month thereafter. Changes/Revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@8000.00 SOFTWARE QUALITY EVALUATION

The following paragraphs delineate the Government's requirements for ensuring the development of quality software.

@8010.00 SCOPE

The software development contractor shall provide technical expertise to conduct a Software Quality Evaluation (SQE) Program on all software

including the support equipment software contained in the system. This support effort shall include assessment, verification and validation of software requirements, algorithm applicability, software design, interfaces, design implementation, computer program performance, development tests, test data, and software documentation. Software baseline control, software development methodology and activities, and product assurance activities also shall be assessed by the contractor's Software Quality Evaluators.

@8020.00 Requirements for Software Quality Evaluation

The Contractor shall establish and follow a Software Quality Evaluation Program in accordance with the requirements of paragraph 5.8, "Software Quality Evaluation," of DOD-STD-2167. The following subparagraphs of paragraph 5.8 may be omitted:

- 5.8.1.2.5b
- 5.8.1.2.8g

Any requirement for a specification, plan, manual, description, procedure, or document shall be omitted if it was not required in any subparagraph of 2.1, "Requirements for Software Development" in this SOW.

@8030.00 Software Requirements

The contractor's Software Quality Evaluators shall:

- @8040.00 a. Analyze all Software Requirements Documentation (e.g., B5 specs; Software Requirements Specification and Interface Requirements Specification) to ensure traceability to system level documents (System/Segment Specification). All problem areas, potential problem areas, and errors identified shall be documented and proposed solutions developed and documented in a "Technical Operating Report on Software Requirements Traceability" (DID # DI-S-30559) and submitted in accordance with the CDRL.

- b. Determine if the:

~ 1423 data for paragraph 8040.00

BLOCK 2 -- Technical Operating Report (TOR) for Software

BLOCK 3 -- Requirements Traceability

BLOCK 4 -- DI-S-30559

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- LT

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Report due 90 days after publish draft SRS. Changes/revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

- @8050.00 System/Segment Specification (SSS) has been prepared in accordance with paragraph 3.1.3.1 of MIL-STD-490, paragraphs 5.1 and 20.4.1 of DOD-STD-2167, DID # DI-CMAN-80008, and as

required by the software development contract Statement of Work (SOW).

@8060.00 Software Requirements Specification (SRS) has been prepared in accordance with paragraphs 3.4.2 and 3.4.7.1 of MIL-STD-483, paragraph 3.1.3.2.5.1 of MIL-STD-490, paragraphs 5.1.1.6 and 5.1.2.4 of DOD-STD-2167, and the software development contract SOW.

@8070.00 c. All problem areas, potential problem areas, and errors identified in item b. above shall be documented and proposed solutions developed and documented in a "Technical Operating Report for Software Requirements Evaluation" (DID # DI-S-30559) and submitted in accordance with the CDRL.

1423 data for paragraph 8070.00

BLOCK 2 -- Technical Operating Report (TOR) for Software

BLOCK 3 -- Requirements Evaluation

BLOCK 4 -- DI-S-30559

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- LT

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 -- Report due 45 days after publish draft SRS. Changes/revisions shall be submitted as change pages for approval. Reproducible; Electronic Media.

@8100.00 Software Preliminary Design

The contractor's Software Quality Evaluators shall:

@8110.00 a. Analyze preliminary design to ensure traceability to software requirements.

@8120.00 b. Evaluate design implementation, structure, I/O, and control flow.

@8130.00 c. Analyze interfaces and timing to ensure definition of all internal and external interfaces; ensure subsystem interface compatibility; identify critically timed interfaces and potential problems; ensure subcontractor and prime contractor software interfaces are compatible and complementary.

@8140.00 d. Analyze the evolving Top Level Design and Software Top Level Design Document (STLDD) for adherence to required format and documentation standards, compliance with contractual requirements, internal consistency, understandability, technical adequacy, appropriate degree of completeness, traceability to the SRS and IRS, feasibility, appropriate design techniques used for the software, appropriate level of detail, appropriate allocation of sizing and timing resources, and appropriate content for intended audience. All problem areas, potential problem areas, and errors identified shall be documented and proposed solutions

developed and documented in a "Technical Operating Report for a Quick Response Analysis of the STLDD" (DID-S-30559) and submitted in accordance with the CDRL.

^ 1423 data for paragraph 8140.00

BLOCK 2 -- Technical Operating Report (TOR) for
BLOCK 3 -- a Quick Response Analysis of the STLDD
BLOCK 4 -- DI-S-30559
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- LT
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 --

@8150.00 e. Inspect the Software Test Plan (STP) for adherence to required format and documentation standards; compliance with contractual requirements; internal consistency; understandability; technical adequacy; appropriate degree of completeness; traceability to the SRS and IRS; consistency with the SDP; feasibility; appropriate level of detail; adequate test coverage of requirements; adequacy of planned tools, facilities, procedures, methods, and resources; and appropriate content for the intended audience. All problem areas, potential problem areas, and errors identified shall be documented and proposed solutions developed and documented in a "Technical Operating Report for a Quick Response Analysis for the STP" (DID # DI-S-30559) and submitted in accordance with the CDRL.

^ 1423 data for paragraph 8150.00

BLOCK 2 -- Technical Operating Report (TOR) for
BLOCK 3 -- a Quick Response Analysis of the STP
BLOCK 4 -- DI-S-30559
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- LT
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 --

@8160.00 f. Analyze the preliminary Software User's Manual (SUM) for adherence to required format and documentation standards; compliance with contractual requirements; internal consistency; understandability; technical adequacy; appropriate degree of completeness; traceability to the SRS and IRS; consistency with the STLDD, CSOM, and CSDM; appropriate level of detail; and appropriate content for intended audience. All problem areas, potential problem areas, and errors identified shall be documented and proposed solutions developed and documented in a "Technical Operating Report for a Quick Response Analysis of the Software User's Manual" (DID-S-30559) and submitted in

accordance with the CDRL.

^ 1423 data for paragraph 8160.00

BLOCK 2 -- Technical Operating Report (TOR) for

BLOCK 3 -- a Quick Response Analysis of the Software User's Manual

BLOCK 4 -- DI-S-30559

BLOCK 6 -- STRBE-TQR

BLOCK 7 -- LT

BLOCK 8 -- A

BLOCK 10 -- ONE/R

BLOCK 12 -- See Item 16

BLOCK 13 -- See Item 16

BLOCK 15 -- Total

BLOCK 16 --

@8200.00 Software Detailed Design

The contractor's Software Quality Evaluators shall:

- @8210.00 a. Analyze detailed software design (C5 Specifications, Software Top Level Design Document (STLDD), Software Detailed Design Document (SDDD), Interface Design Document (IDD), Data Base Design Document (DBDD), and Software Product Specification (SPS)) to ensure traceability to the software requirements (B5 specifications - Software Requirements Specification (SRS) and Interface Requirements Specification (IRS)).
- @8220.00 b. Verify compliance with the Software Top Level Design Document requirements. Especially spare timing and memory.
- @8230.00 c. Review detailed design to assess readiness for coding.
- @8240.00 d. Conduct detailed analysis of interfaces timing and testing as identified in paragraph 2.1.2.d.
- @8250.00 e. Construct coded routines for selected critical algorithms and functions from the B5 and C5 specifications and verify specified performance.
- @8260.00 f. Review the updated Software Development Plan (SDP) for adherence to required format and documentation standards, compliance with contractual requirements, internal consistency, understandability, technical adequacy, appropriate degree of completeness, traceability to the SOW, consistency with each other, feasibility, appropriate level of detail, and appropriate content for intended audience. All problem areas, potential problem areas, and errors identified shall be documented and proposed solutions developed and documented in a "Technical Operating Report for a Quick Response Analysis of the SDP" (DID-S-30559) and submitted in accordance with the CDRL.
- ^ 1423 data for paragraph 8260.00
- BLOCK 2 -- Technical Operating Report (TOR) for
- BLOCK 3 -- a Quick Response Analysis of the SDP
- BLOCK 4 -- DI-S-30559
- BLOCK 6 -- STRBE-TQR

BLOCK 7 -- LT
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 --

@8270.00 g. Analyze the evolving detailed design and the Software Detailed Design Document (SDDD) for adherence to required format and documentation standards, compliance with contractual requirements, internal consistency, understandability, technical adequacy, appropriate degree of completeness, traceability to the SRS, IRS, and STLDD, consistency with each other, feasibility, appropriate level of detail, adequacy of planned tools, facilities, procedures, methods, and resources; and appropriate content for intended audience. All problem areas, potential problem areas, and errors identified shall be documented and proposed solutions developed and documented in a "Technical Operating Report for a Quick Response Analysis of the Software Detailed Design Document" (DID-S-30559) and submitted in accordance with the CDRL.

^ 1423 data for paragraph 8270.00
BLOCK 2 -- Technical Operating Report (TOR) for
BLOCK 3 -- a Quick Response Analysis of the Software Detailed Design Document
BLOCK 4 -- DI-S-30559
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- LT
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 --

@8280.00 h. Analyze the evolving interface design for adherence to required format and documentation standards; compliance with contractual requirements; internal consistency; understandability; technical adequacy; appropriate degree of completeness; traceability to the SRS, IRS, and STLDD; consistency with the SDDD and DBDD; feasibility; appropriate design techniques; appropriate level of detail; appropriate allocation of sizing and timing resources; adequacy of planned tools, facilities, procedures, methods, and resources; and appropriate content for intended audience. All problem areas, potential problem areas, and errors identified shall be documented and proposed solutions developed and documented in a "Technical Operating Report for a Quick Response Analysis of the Software Interfaces" (DID-S-30559) and submitted in accordance with the CDRL.

^ 1423 data for paragraph 8280.00
BLOCK 2 -- Technical Operating Report (TOR) for
BLOCK 3 -- a Quick Response Analysis of the Software Interfaces
BLOCK 4 -- DI-S-30559

BLOCK 6 -- STRBE-TQR
 BLOCK 7 -- LT
 BLOCK 8 -- A
 BLOCK 10 -- ONE/R
 BLOCK 12 -- See Item 16
 BLOCK 13 -- See Item 16
 BLOCK 15 -- Total
 BLOCK 16 --

@8290.00 i. Analyze the evolving data base design and Data Base Design Document (DBDD), if being prepared, for adherence to required format and documentation standards; compliance with contractual requirements; internal consistency; understandability; technical adequacy; appropriate degree of completeness; traceability to the SRS, IRS, and STLDD; consistency with the SDDD and IDD; feasibility; appropriate design techniques; appropriate level of detail; appropriate allocation of sizing and timing resources; adequacy of planned tools, facilities, procedures, methods, and resources; consistency between data definition and data use; accuracy and required precision of constants; adequacy of backup procedures and mechanisms; and appropriate content for intended audience. All problem areas, potential problem areas, and errors identified shall be documented and proposed solutions developed and documented in a "Technical Operating Report for a Quick Response Analysis of the Data Base Design Document" (DID-S-30559) and submitted in accordance with the CDRL.

^ 1423 data for paragraph 8290.00

BLOCK 2 -- Technical Operating Report (TOR) for
 BLOCK 3 -- a Quick Response Analysis of the Data Base Design Document
 BLOCK 4 -- DI-S-30559
 BLOCK 6 -- STRBE-TQR
 BLOCK 7 -- LT
 BLOCK 8 -- A
 BLOCK 10 -- ONE/R
 BLOCK 12 -- See Item 16
 BLOCK 13 -- See Item 16
 BLOCK 15 -- Total
 BLOCK 16 --

@8300.00 Coding, Unit Testing, and CSC Integration Testing

The contractor's Software Quality Evaluators shall:

@8310.00 a. Analyze the evolving and completed source code for each unit for adherence to required format and documentation standards; compliance with contractual requirements; internal consistency; understandability; technical adequacy; appropriate degree of completeness; traceability to the SDDD, IDD, and DBDD; and appropriate coding techniques used. All problem areas, potential problem areas, and errors identified shall be documented and proposed solutions developed and documented in a "Technical Operating Report for Source Code Analysis" (DID-S-30559) and submitted in accordance with the CDRL.

^ 1423 data for paragraph 8310.00
 BLOCK 2 -- Technical Operating Report (TOR) for
 BLOCK 3 -- Source Code Analysis
 BLOCK 4 -- DI-S-30559
 BLOCK 6 -- STRBE-TQR
 BLOCK 7 -- LT
 BLOCK 8 -- A
 BLOCK 10 -- ONE/R
 BLOCK 12 -- See Item 16
 BLOCK 13 -- See Item 16
 BLOCK 15 -- Total
 BLOCK 16 --

@8320.00 b. Analyze unit and CSC integration test procedures for adherence to required format and documentation standards; compliance with contractual requirements; internal consistency; understandability; technical adequacy; appropriate degree of completeness; traceability to the STP, CSC Test Cases, and Unit Test Cases; consistency with the SDDD, IDD, and DBDD; feasibility; appropriate level of detail; adequate test coverage of requirements; adequacy of planned tools, facilities, procedures, methods, and resources; and appropriate content for intended audience. All problem areas, potential problem areas, and errors identified shall be documented and proposed solutions developed and documented in a "Technical Operating Report for Unit and CSC Integration Testing" (DID-S-30559) and submitted in accordance with the CDRL.

^ 1423 data for paragraph 8320.00
 BLOCK 2 -- Technical Operating Report (TOR) for
 BLOCK 3 -- Unit and CSC Integration Testing
 BLOCK 4 -- DI-S-30559
 BLOCK 6 -- STRBE-TQR
 BLOCK 7 -- LT
 BLOCK 8 -- A
 BLOCK 10 -- ONE/R
 BLOCK 12 -- See Item 16
 BLOCK 13 -- See Item 16
 BLOCK 15 -- Total
 BLOCK 16 --

@8400.00 CSCI Level Testing

The contractor's Software Quality Evaluators shall:

- @8410.00 a. Perform analysis of the software and identify potential timing and saturation problems.
- @8420.00 b. Witness integration testing, identify discrepancies, and conduct independent analysis of problems to independently determine sources of problems.
- @8430.00 c. Verify that the interfaces (I/O) between CSCIs are correctly implemented in accordance with the design and requirements.

@8440.00 d. Analyze the Software Product Specification (SPS) for adherence to required format and documentation standards; compliance with contractual requirements; internal consistency; understandability; technical adequacy; appropriate degree of completeness; traceability to the SRS and IRS; incorporation of STLDD, SDDD, IDD, DBDD, and software listings consistent with updated source code; and appropriate content for intended audience. All problem areas, potential problem areas, and errors identified shall be documented and proposed solutions developed and documented in a "Technical Operating Report for a Quick Response Analysis of the Software Product Specification" (DID # DI-S-30559) and submitted in accordance with the CDRL.

^ 1423 data for paragraph 8440.00

BLOCK 2 -- Technical Operating Report (TOR) for
BLOCK 3 -- Quick Response Analysis of the Software Product Specification
BLOCK 4 -- DI-S-30559
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- LT
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total
BLOCK 16 --

@8500.00 Source Code Validation

The contractor's Software Quality Evaluators shall:

@8510.00 a. Validate the updated Source Code for adherence to required format and documentation standards; compliance with contractual requirements; internal consistency; understandability; technical adequacy; appropriate degree of completeness; compliance with language and coding standards; compliance with maintainability standards; consistency with updated SDDD, IDD, and the DBDD; appropriate coding techniques used; and appropriate application of timing and sizing resources. All problem areas, potential problem areas, and errors identified shall be documented and proposed solutions developed and documented in a "Technical Operating Report for a Quick Response Analysis of the Source Code" (DID # DI-S-30559) and submitted in accordance with the CDRL.

^ 1423 data for paragraph 8510.00

BLOCK 2 -- Technical Operating Report (TOR) for
BLOCK 3 -- Quick Response Analysis of the Source Code
BLOCK 4 -- DI-S-30559
BLOCK 6 -- STRBE-TQR
BLOCK 7 -- LT
BLOCK 8 -- A
BLOCK 10 -- ONE/R
BLOCK 12 -- See Item 16
BLOCK 13 -- See Item 16
BLOCK 15 -- Total

BLOCK 16 --

@8520.00 b. Validate that the integrated CSCIs meet all software and system requirements by witnessing system level integration, test and demonstrations.

@8530.00 c. Verify and validate that all CSCI reassemblies plus patches are reflected in documentation and satisfy requirements.

@8600.00 Reviews and Audits Support

The contractor's Software Quality Evaluators shall provide technical support, attend, and document findings for the following reviews, audits, and meetings:

@8610.00 a. Formal Reviews:

All reviews and audits identified in the software development Statement of Work.

PDR - Preliminary Design Review

CDR - Critical Design Review

@8620.00 b. Audits:

PCA - The contractor's Software Quality Evaluators shall review all C5 Specs against qualified computer programs to ensure that they agree and support, as required, MIL-STD-1521 activities.

@8625.00 b. PCA - Physical Configuration Audit - The IV&V contractor shall review all C5 Specs against qualified computer programs to ensure that they agree and support, as required, MIL-STD-1521 activities.

@8630.00 c. Working Group Meetings:

CRWG - Computer Resource Working Group

ICWG - Interface Control Working Group

TPWG - Test Plan Working Group

APPENDIX D -- READER.PAS

program reader;

{ This program is intended to be used in the SQA expert system.
Prepared in 1986 by McLean Research Center, Inc. and Copyright by MRC.

Programmed by Neil Romstedt and Judy Podell.

The program is invoked with 4 parameters. Each parameter is a file name.

1. The text file. For the SQA ES it is the file 'SOW'. This is an ascii file with some special embedded characters which imply a format to this program. An '@' in column 1 of a line denotes the beginning of a new paragraph. Immediately following the '@' will be a paragraph number (real format). The paragraph numbers are in sequential order, and any new paragraph can be inserted using fractional amounts. The remainder of that line will be the paragraph title. The body of the paragraph will consist of the lines following the title line, until another '@' is found. For paragraphs which imply a 1423 entry, a '^' in column 1 of a line of the paragraph body will begin the data that is to be entered into the 1423. Each subsequent line until another '@' will be interpreted as a 1423 entry item.
2. The numbers file. For the SQA ES it is the file 'RESULTS'. This is an ascii file of paragraph numbers to include in the final product. The paragraph numbers should have corresponding entries in the text file, but if they don't the program will note the deficiency rather than bombing. Each number is in real format with one number per line. The paragraph numbers are sorted by this program, so they may be in any order in the file. Errors in this file will cause the program to bomb.
3. The output file. For the SQA ES it is the file 'OUTFILE'. This is an ascii file of the selected paragraphs of the text file - the tailored Statement of Work. This file is appropriate for reading into many word processors, but will include hard carriage returns at the end of each line.
4. The cdrl file. For the SQA ES it is the file 'CDRL'. This is an ascii file of the form 1423 entries for the paragraphs that were selected from the text file. This data is written to a separate file because it will be handled separately in the preparation of the CDRL.

}

{ VARIABLE TYPE DECLARATIONS }

type str255=string[255];	{strings to reach each text line}
str20=string[20];	{ string for input }
rarray=array[1..100] of real;	{array to store the paragraph numbers}

{ GLOBAL VARIABLE DECLARATIONS. These variables exist everywhere. }

```
var tf,num,outfile,cdrl:text;           {file definitions}
    line,oline,tstr,system,media,ndelivery,form:str255;
    break,i,last,n:integer;
    flag,ok:boolean;
    test,textno:real;
    r:rarray;
```

{ Note: In a PASCAL program, the action begins with the last procedure.
Before that, all subroutines and functions must be defined. So what you see
here are all the utilities that must exist for this program. }

```
procedure readint(var i:integer; def,maxval:integer);
```

{ This procedure buffers the input so that only an integer can be entered.
It also provides a default value for the answer, and a maximum value for
the answer, which must be greater than zero. }

```
var num:str20; res,x,y:integer;
begin
    x:=wherex; y:=wherey;
    repeat
        gotoxy(x,y); clreol; write(def:1); gotoxy(x,y);
        repeat until keypressed; clreol; readln(num); res:=0; i:=def;
        if length(num)>0 then val(num,i,res);
        until (res=0) and (i>0) and (i<=maxval);
    end;
```

```
procedure parse(var line:str255; var r:real);
```

{ This procedure takes a line of text, looks to see if it is preceeded by a '@'
or a '^' or a '~' (outmoded), and returns the paragraph number if '@', a -1.0
if '^', or a 0.0 otherwise. In all cases the remainder of the text line is
returned in the variable 'line'. }

```
var code,i:integer; numstr:str255;
begin
    r:=0.0; {initialize the paragraph number to zero}
    if line[1]='@' then begin {for paragraph header lines}
        i:=pos(' ',line); {find the first blank space in the line}
        if i=0 then i:=length(line)+1; {if there is no blank, then there is no
            title}
        numstr:=copy(line,2,i-2); {put the number into a temporary string}
        val(numstr,r,code); {evaluate the number string to the paragraph number}
        if code>0 then r:=0.0; {if the number can't be evaluated then ignore it}
        line:=copy(line,i+1,length(line)-i); {put the title part back into 'line'}
    end
```

```

else if line[1]='^' then begin {for cdrl header lines}
  line:=copy(line,2,length(line)-1); {put the title part back into 'line'}
  r:=-1.0; {set the paragraph number to -1.0 so that it will stand out}
end
else if line[1]='~' then
  line:=copy(line,2,length(line)-1); {strip this outmoded character}
end;

procedure first_at(var textno:real; var title:str255; var ok:boolean);

{ This procedure reads and skips lines until a line with an '@' is encountered.
  This is used to skip over paragraphs which are not included in the tailored
  SOW. It returns the paragraph number in 'textno', the paragraph title in
  'title', and the end-of-file flag in 'ok'.}

var i:integer; line:str255;
begin
  ok:=false; {assume that I haven't found an '@'}
  while (not eof(tf)) and (not ok) do begin {keep looking until EOF or I have}
    readln(tf,line); {get the next text line from the text file}
    parse(line,textno); {parse it to find its paragraph number}
    if textno>0.0 then begin {if it is a paragraph title line then}
      title:=line; {set the title equal to the remainder of the line}
      ok:=true; {set the flag to true}
    end;
  end;
end;

procedure switchback(color:integer);

{ This simple procedure provides an easy way of switching the screen color,
  using the Turbo Pascal screen directives. }

begin
  textbackground(color); clrscr;
end;

function sp(i:integer):str255;

{ This function returns a string which contains a number of blank spaces. It
  is used to make screen formatting easier within the Pascal write statement.}

var st:str255;
begin
  st:='';
  while i>0 do begin st:=st+' '; i:=i-1; end;
  sp:=st;
end;

```

```
function center(st:str255; fw,l:integer):str255;
```

```
{ This function returns a string which contains an input string 'st' centered
  within a field width 'fw', and preceeded by 'l' blank spaces. It is used
  to make screen formatting easier within the Pascal write statement.}
```

```
var i:integer;
begin
  i:=(fw-length(st)) div 2;
  center:=sp(i+1)+st+sp(fw-i-length(st));
end;
```

```
procedure sort(first,last:integer);
```

```
{ This procedure implements the Quicksort Algorithm. It is used sort a real
  array 'r' in increasing order. It is recursive (i.e. it calls itself). The
  parameters are the first ('first') and last ('last') elements of the array
  that are currently being sorted. To sort 30 numbers stored in positions 1
  thru 30 of array 'r' you would call 'sort(1,30)'. The algorithm works by
  dividing array into smaller and smaller subsets, with each subset consisting
  of values smaller than subsequent subsets in the array. Eventually, the
  array is completely sorted. This is the fastest and easiest sort known. }
```

```
var i,j: integer;
    temp,dividingline: real;
begin
  i:=first;
  j:=last;
  dividingline:=r[(first+last) div 2];
  repeat
    while r[i]<dividingline do i:=i+1;
    while r[j]>dividingline do j:=j-1;
    if i<=j then begin
      temp:=r[i]; r[i]:=r[j]; r[j]:=temp;
      i:=i+1; j:=j-1;
    end
  until i>j;
  if first<j then sort(first,j);
  if i<last then sort(i,last);
end;
```

```
procedure waitout;
```

```
{ This procedure halts program execution so the user can read the screen. }
```

```
begin
  write('Press <return> to continue '); readln;
end;
```

```
procedure askquestions;
```

```
{ This procedure asks the questions at the beginning of the program about the
  deliverables, and makes sure that the user has reviewed his choices. The
  responses are global variables. No fancy processing here. }
```

```
var i,ok:integer; ntemp:str255;
```

```
    ans:char;
```

```
begin
```

```
    ndelivery:='';
```

```
    repeat
```

```
        ok:=0; i:=0;
```

```
        switchback(lightrd);
```

```
        repeat
```

```
            writeln;
```

```
            writeln(sp(2),'How many hardcopies of specifications, descriptions, ',
                    'procedures,');;
```

```
            writeln(sp(2),'reports, manuals and documents is the contractor',
                    'to deliver? ');;
```

```
            write(sp(5)); readln(ntemp);
```

```
            if length(ntemp)>0 then ndelivery:=ntemp;
```

```
        until length(ndelivery)>0;
```

```
        writeln; writeln;
```

```
        writeln(sp(2),'What type of electronic media is the contractor to',
                    'employ for');
```

```
        writeln(sp(2),'his deliverables?');
```

```
        writeln(sp(5),'1. Magnetic tape, 9 track, 1600 bpi, 3/4 inch');
```

```
        writeln(sp(5),'2. Floppy disk, CPT, 8", single sided, single density, 26',
                    'soft sectors');
```

```
        writeln(sp(5),'3. Floppy disk, MS-DOS, 5 1/4", double sided, double',
                    'density');
```

```
        writeln(sp(5),'4. Floppy disk, Apple Macintosh, 3 1/2", single sided');
```

```
        writeln(sp(5),'5. Other');
```

```
        write(sp(2),'Enter Choice: '); readint(i,5,5);
```

```
        case i of
```

```
            1: media:='magnetic tape, 9 track, 1600 bpi, 3/4 inch';
```

```
            2: media:='floppy disk, CPT, 8", single sided, single density 26 soft',
                    'sectors';
```

```
            3: media:='floppy disk, MS-DOS, 5 1/4", double sided, double density';
```

```
            4: media:='floppy disk, Apple Macintosh, 3 1/2", single sided';
```

```
            5: begin write(sp(2),'Enter media type '); readln(media); end;
```

```
        end;
```

```
        writeln; writeln;
```

```
        writeln(sp(2),'What is the host system?');
```

```
        writeln(sp(5),'1. IBM-PC or compatible');
```

```
        writeln(sp(5),'2. CPT Word Processor');
```

```
        writeln(sp(5),'3. Apple Macintosh');
```

```
        writeln(sp(5),'4. Other');
```

```
        write(sp(2),'Enter Choice: '); readint(i,4,4);
```

```

case i of
  1: system:='IBM-PC (or IBM-PC compatible)';
  2: system:='CPT 8500 Series Word Processor';
  3: system:='Apple Macintosh';
  4: begin write(sp(2),'Enter System Name '); readln(system); end;
end;
writeln; writeln;
writeln(sp(2),'What is the word processor/data format?');
writeln(sp(5),'1. Standard ASCII');
writeln(sp(5),'2. None is specified');
writeln(sp(5),'3. Word Perfect');
writeln(sp(5),'4. Wordstar');
writeln(sp(5),'5. Microsoft Word');
writeln(sp(5),'6. Volkswriter');
writeln(sp(5),'7. Displaywrite');
writeln(sp(5),'8. Multimate Advantage');
writeln(sp(5),'9. Mac Word');
writeln(sp(5),'10. Mac Write');
writeln(sp(5),'11. Other (specify)');
write(sp(2),'Enter Choice: '); readint(i,11,11);
case i of
  1: form:='Standard ASCII text';
  2: form:='';
  3: form:='Word Perfect software';
  4: form:='Wordstar software';
  5: form:='Microsoft Word software';
  6: form:='Volkswriter software';
  7: form:='Displaywrite software';
  8: form:='Multimate Advantage software';
  9: form:='Mac Word software';
  10: form:='Mac Write software';
  11: begin
    write(sp(2),'Enter word processor/data format '); readln(form); end;
end;
clrscr; writeln; writeln; writeln;
writeln(sp(2),'Number of Hardcopies: ',ndelivery:1); writeln;
writeln(sp(2),'Media to Use: ',media); writeln;
writeln(sp(2),'Host System: ',system);
writeln(sp(2),'Data Format: ',form);
writeln; writeln; writeln; writeln;
write(sp(2),'Is this data correct? (Y/N) '); readln(ans);
if upcase(ans)='Y' then ok:=1;
writeln; writeln; writeln;
until ok=1;
system:=system+', '+form;
end;

```

```

procedure writeheader;

{ This procedure writes the introductory header message on the screen.
  No fancy processing here. }

begin
  switchback(lightblue); textcolor(yellow); writeln; writeln; writeln;
  writeln(center('Expert System Statement Of Work Translator Program',79,0));
  writeln;
  writeln(center('Copyright 1986, McLean Research Center, Inc.',79,0));
  writeln; writeln;
  writeln(center(
    'This materiel may be reproduced by or for the U.S. Government',79,0));
  writeln(center(
    'pursuant to the copyright license under DoD FAR Supplement 52.227-7013.',
    79,0));
  writeln; writeln;
  writeln(sp(10),'This program is taking the SQA Expert System output, which');
  writeln(sp(5),
    'is a group of SQA paragraph numbers to include in the Statement of');
  writeln(sp(5),
    'Work, and is extracting those paragraphs in order from a master');
  writeln(sp(5),
    'data base of SQA paragraphs. The program output is a text file');
  writeln(sp(5),
    'which will be read by a word processor and incorporated into the');
  writeln(sp(5),
    'actual Statement of Work. Also produced is a second file, called');
  writeln(sp(5),
    'CDRL, which contains instructions for completing Forms 1423 as');
  writeln(sp(5),'directed by the included paragraphs. ');
  writeln; writeln; writeln; waitout;
end;

```

```

function replace(line,target,repl:str255):str255;

```

```

{ This function returns a text string, which is the 'line' string with the
  'target' substring replaced by the 'repl' substring. If 'line' does not
  contain the 'target' then it is returned unaltered. This is used to
  substitute the deliverable quantities into the text of paragraph #600.00.)

```

```

var i,lt,ls:integer;
begin
  i:=pos(target,line); {see if the 'target' is in 'line' and find out where}
  lt:=length(target);   {compute the dynamic length of 'target' for use below}
  ls:=length(line);     {compute the dynamic length of 'line' for use below}
  if i>0 then replace:=copy(line,1,i-1)+repl+copy(line,i+lt,ls-(i+lt-1))
  else replace:=line;
end;

```

```

{ This part begins the main portion of the program, where the fun begins. }

begin

  { assign the files to the parameters, and open them for reading or
    writing. Pascal used reset to read from a file (which must already exist)
    and rewrite to write to a file (which will be overwritten if it already
    exists). Paramstr is a Turbo pascal feature. }
  assign(tf,paramstr(1)); reset(tf);
  assign(num,paramstr(2)); reset(num);
  assign(outfile,paramstr(3)); rewrite(outfile);
  assign(cdrl,paramstr(4)); rewrite(cdrl);

  { read in the paragraph numbers from the text file, and sort them in
    increasing order. }
  n:=0;
  while not eof(num) do begin
    n:=n+1;
    readln(num,r[n]);
  end;
  sort(1,n);

  { the reader program should stop abruptly if there are any paragraph numbers
    with a value less than or equal to zero. This will allow the expert
    system to signal a termination of the process. }
  if r[1]<=0.0 then begin
    writeln(' The reader program has been terminated at the',
      ' direction of the expert system. ');
  end
  else begin { continue with the process }

    { start off with header and questions }
    writeheader;
    askquestions;

    { send the introductory data message to the cdrl output file }
    writeln(cdrl);
    writeln(cdrl,'The following is the data to be entered onto Form 1423');

    { advance the text file to the first '@'. Then begin a loop for each
      paragraph number. }
    first_at(textno,line,ok);
    for i:=1 to n do begin

      { advance the text file until the paragraph number in the text file
        equals or exceeds the paragraph number in the array. If they are not
        exactly equal, then the paragraph number in the array has been missed
        somehow - because it wasn't there or because it was incorrectly marked. }
      while (r[i]>textno) and (ok) do first_at(textno,line,ok);
      if r[i]<>textno then writeln(outfile,
        'text not found for paragraph ',r[i]:5:2)

```

```

{ If the paragraph number is 600.00 then special processing is needed to
substitutue the deliverable data. Each line is read in turn, and the word
'NUMBER' is replaced by the variable 'ndelivery', 'TYPE' is replaced by
'media', and 'SYSTEM' is replaced by 'system'.}
else if textno=600.0 then begin
  writeln(outfile);
  writeln(outfile,line);
  oline:='';
  repeat
    readln(tf,line);
    parse(line,textno);
    if textno=0.0 then begin
      if length(oline)>0 then line:=oline+' '+line;
      line:=replace(line,'NUMBER',ndelivery);
      line:=replace(line,'TYPE',media);
      line:=replace(line,'SYSTEM',system);
      while length(line)>79 do begin
        break:=79;
        while line[break]<>' ' do break:=break-1;
        writeln(outfile,copy(line,1,break));
        line:=copy(line,break+1,length(line)-break);
      end;
      oline:=line;
    end;
  until textno>0.0;
  writeln(outfile,oline); end

{ For all other paragraph numbers the standard processing occurs. }
else begin

  { the title line is written. }
  writeln(outfile);
  writeln(outfile,line);

  { lines from the text file are read and processed sequentially until
  the next paragraph number is encountered. }
  repeat
    if eof(tf) then ok:=false
    else begin

      { read a line and parse it }
      readln(tf,line);
      parse(line,textno);

      { if it is a paragraph body line then write it to outfile }
      if textno=0.0 then writeln(outfile,line);

```

```

    { if it is a 1423 starting line then begin the processing to the
      cdrl file until the next paragraph number is encountered. This
      begins with header data. }
    if textno<0.0 then begin
      writeln(outfile);
      writeln(cdrl);
      writeln(cdrl,'Form 1423 data: ',line);
      repeat
        if eof(tf) then ok:=false
        else begin
          readln(tf,line);
          parse(line,textno);
          if textno=0.0 then writeln(cdrl,line);
        end;
      until (textno>0.0) or (not ok);
    end;

    end; { the line which was found was not the end-of-file }
  until (textno>0.0) or (not ok); { a new paragraph number was found }

  end; { the standard processing sequence }
end; { the loop of array elements }

end; { the program is not terminated abruptly. }

{ The processing is done. Close the four external files. }
close(tf);
close(num);
close(outfile);
close(cdrl);

{ Finish out nicely. }
waitout;
switchback(black);
textcolor(white);

{ End of the main procedure. Thank you. }
end.

```

DATE
LMED
— 8