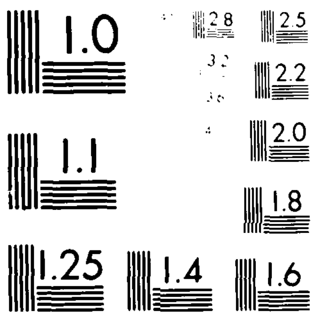
AD-A113 423 COMPUTER SCIENCES CORP FALLS CHURCH VA F/6 5/1
RECOMMENDED METHODOLOGY FOR INTER-SERVICE/AGENCY AUTOMATED MESS--ETC(U)
FEB 82 F23613-77-D-0011

UNCLASSIFIED NL

1 OF 2
AD A
113423



MICROCOPY RESOLUTION TEST CHART

RECOMMENDED METHODOLOGY FOR INTER-SERVICE/AGENCY AUTOMATED MESSAGE PROCESSING EXCHANGE (I-S/A AMPE) COST AND SCHEDULE ANALYSIS OF SECURITY ALTERNATIVES

Prepared for

U.S. AIR FORCE COMMUNICATIONS COMMAND
Scott AFB, Illinois 62225

FEBRUARY 23, 1982

CSC

COMPUTER SCIENCES CORPORATION

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A113 423	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) RECOMMENDED METHODOLOGY FOR INTER-SERVICE/AGENCY AUTOMATED MESSAGE PROCESSING EXCHANGE (I-S/A AMPE) COST AND SCHEDULE ANALYSIS OF SECURTIY ALTERNATIVES		5. TYPE OF REPORT & PERIOD COVERED FINAL 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)		8. CONTRACT OR GRANT NUMBER(s) Contract F23613-77-D-0011 Task 81-4
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer Sciences Corporation 6565 Arlington Boulevard Falls Church, Virginia 22046		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS HQ Air Force Communications Command Dir of Contracting, Engineering, Plans, and Acq Scott Air Force Base, IL 62225		12. REPORT DATE February 23, 1982 13. NUMBER OF PAGES 160
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Air Force Automated Systems Project Office Directorate of System Engineering for I-S/A AMPE Gunter Air Force Station, AL 36114		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution A. Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer security; software cost estimation; Automated Message Processing Exchange (AMPE); Inter-Service/Agency AMPE (I-S/A AMPE); Computer; PRICE estimation models; life-cycle cost estimations.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes a methodology to project the development cost and schedule for proposed implementation techniques to provide a multi-level secure environment for the Inter-Service/Agency Automated Message Processing Exchange (I-S/A AMPE). This report documents the development of the cost and schedule methodology, identifies associated inputs, and sets forth the rationale and criteria used to construct the methodology. The major distinction between this work and normal life-cycle estimation techniques is that this methodology considers all system hardware and software security factors.

DD FORM 1 JAN 73 1473

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

The methodology draws from four models chosen from the Programmed Review of Information for Costing and Evaluation (PRICE) family of models and the Software Life-Cycle Management (SLIM) model for cross-checking software factors. First, the automated models are easy to use, and secondly as shown during the comparative analysis, these models are superior to other models based on evaluation criteria and desirable attributes. Third, the incororation of these models into the methodology provides the generality needed by the Air Force to apply this approach in the evaluation of security alternatives. The analytical processes surrounding the execution of the models provides the additional framework needed to complete the approach.

Recommended Methodology

for

Inter-Service/Agency Automated
Message Processing Exchange
(I-S/A AMPE)

Cost and Schedule Analysis
of Security Alternatives

Prepared for

U.S. AIR FORCE COMMUNICATIONS COMMAND
Scott AFB, Illinois 62225

by

COMPUTER SCIENCES CORPORATION
6565 Arlington Boulevard
Falls Church, Virginia 22046

February 23, 1982

TABLE OF CONTENTS

INTRODUCTION

PART I - COST AND SCHEDULE MODEL ANALYSIS

TABLE OF CONTENTS (Cont'd)

PART III - APPLICATION OF THE METHODOLOGY

LIST OF ILLUSTRATIONS

Figure

LIST OF TABLES

Table

INTRODUCTION

INTRODUCTION

Under Contract F23613-77-D-0011, Task 81-4, Computer Sciences Corporation (CSC) was tasked to study and evaluate three Government-provided security alternatives for the Interservice/Agency Automated Message Processing Exchange (I-S/A AMPE) with respect to cost and schedule factors. The results of this task will help the Phase IV Program Management Office (PMO) determine the cost-effectiveness of each alternative proposed for the I-S/A AMPE program. Included in the study and evaluation effort are three subtasks:

1. Establishment of a cost and schedule evaluation methodology to be applied to the security alternatives
2. Cost and schedule analysis of the Government-provided security alternatives
3. Support for the PMO at technical evaluation meetings.

The first subtask required conducting a comparative analysis of cost and schedule models, developing a methodology incorporating appropriate models, and providing the framework for information that will be obtained for each security alternative.

The second subtask will use the methodology developed in the first subtask to analyze the three security alternatives. The third subtask will yield specific information on each of the proposed alternatives.

The objective of the first subtask was to develop a methodology that:

1. Yields commensurable information on the three security alternatives; that is, compare "apples to apples"
2. Incorporates and handles factors unique to security technology
3. Accommodates software and hardware development, operation, and maintenance on a turnkey basis
4. Includes both cost and schedule functions.

The immediate results of this effort have been to:

1. Establish a comprehensive list of security-related factors to quantitatively evaluate the proposed alternatives

2. Provide a basis to analyze commensurably and evaluate different security alternatives.

This report documents the development of the cost and schedule methodology and identifies associated inputs and information of the first subtask. It also sets forth the rationale and criteria used to construct the methodology and select the cost and schedule tools.

The methodology draws from four models chosen from the Programmed Review of Information for Costing and Evaluation (PRICE) family of models and the Software Life-Cycle Management (SLIM) model for cross-checking software estimates. The recommendation of these automated models is based on several factors. First, the automated models are easy to use, and secondly as shown during the comparative analysis, these models are superior to other models based on evaluation criteria and desirable attributes. Third, the incorporation of these models into the methodology provides the generality needed by the Air Force to apply this approach in the evaluation of security alternatives. The analytical processes surrounding the execution of the models provide the additional framework needed to complete the methodological approach.

This report is presented in three parts.

Part I documents the analysis of cost and schedule models. It presents the evaluation criteria used in the analysis, tabular descriptions and comparative analysis of the candidate models, and the selection of the most suitable models.

Part II describes the developed methodology.

Part III addresses the activities needed to apply the methodology. Included are the steps comprising the methodology, a detailed discussion of the PRICE models, models the input information requirement, additional factors that are unique to the security alternatives, the calibration and execution procedures, and the procedure for integration and analysis of model outputs to formulate the final cost and schedule estimates.

Appendix A describes the three proposed security alternatives which are the secure operating system, hardware separation, and end-to-end encryption. To illustrate the cost and schedule effects, example scenarios are given for each alternative.

Appendix B provides a list of applicable references.

Appendix C is included as a reference to model-specific information. It includes a glossary of model terminology.

PART I

COST AND SCHEDULE
MODEL ANALYSIS

SECTION 1 - BACKGROUND

Part I describes the analysis process used to establish the methodology. It begins by establishing the model evaluation criteria and continues through the consideration and comparative analysis of available models.

The following paragraphs describe the major considerations during this analysis; that is security technology, and its effect on cost and schedule estimation.

1.1 SECURITY CONSIDERATIONS

The I-S/A AMPE is required to handle both General Service (GENSER) and Defense Special Security Communications System (DSSCS) information including a multilevel security capability that satisfies specific accreditation criteria. For the purpose of this report, Department of Defense (DoD) Manual C-5030-58-M (July 1978), Defense Special Security Communications System, Security Criteria, and Telecommunications Guidance, sets the guidelines for system planning, design, and determination of security acceptability.

The major distinction between this task and normal system life-cycle estimation techniques is that this methodology must consider all system hardware and software security factors.

Three basic security alternatives are being considered for I-S/A AMPE. These are:

1. Secure operating system
2. Hardware separation
3. End-to-end encryption.

For each of these alternatives, the security factors that affect cost and schedule must be identified explicitly. Descriptions of possible scenarios are given in Appendix A to this report and illustrate the application of each alternative. A brief description of each alternative is given below.

The development of a secure operating system is one alternative for providing security in the I-S/A AMPE. Characteristics and features relative to the development of secure software include enforcement of security policy by hardware and software mechanisms, software development methodology, formal specifications, formal verification, and language considerations.

A second alternative ensures separation of multilevel information within the I-S/A AMPE processor by hardware, that is, different levels are processed by different hardware. This can be implemented by separate, dedicated processors with one processor dedicated to a single information level or with a multimicroprocessor architecture.

End-to-end encryption is a third alternative for providing security in the I-S/A AMPE. Information is encrypted by the sender and is not decrypted until it reaches its destination. This alternative incorporates encryption devices into the system and could be used with the other alternatives.

1.2 COST ESTIMATION

Due to the intrinsic properties of the three security alternatives, the person performing the exercise must have a thorough understanding and knowledge of the system security. In addition, a qualified person must be trained and experienced in using the particular algorithms or models before there can be confidence in the results.

Another qualification, as most costing experts will attest, is in the use of more than one model to estimate the costs. This is the result of the different methods being employed by the various models. Some use the top-down approach, that generally can be used early in the costing exercise because of less stringent parameter requirements. Others use the bottom-up approach that requires a specific knowledge of the development project components. These two methods are often used as cross-checking mechanisms for a given project.

The costing techniques selected for this task are a combination of different processes because there is no single general-purpose tool or model that approaches the full capabilities required by the task. For instance, some of the capabilities that are required include software and hardware costs, development and life-cycle costs, and special security features, such as formal specification and verification, and greater overall development complexity, and personnel clearance costs.

1.3 SCHEDULING

Whereas costing is quantitative in nature, scheduling reflects the qualitative aspects of the development project. That is, scheduling focuses more on how the project is to be developed rather than on how much is to be expended on the project. Each function complements the other in determining the time and effort for completing a given project.

The development and implementation of this methodology, which is oriented specifically toward the system security evaluation, will give the Air Force a new approach that is more comprehensive than traditional single-purpose estimation techniques.

The scheduling for the alternatives must take into account any special security requirement, such as formal verification that affects the development time. The scheduling function can be handled in a straightforward manner with consideration of these factors.

SECTION 2 - OVERVIEW OF MODEL ANALYSIS APPROACH

The selected approach was based on an investigation of existing cost and schedule models potentially suitable for application to the I-S/A AMPE task. A core of candidate models was established from the most suitable model(s) that could be selected by:

1. Establishing model evaluation criteria
2. Arranging a checkoff list of desirable model attributes for comparative analysis.

The core of candidate models was selected by analyzing studies performed by reputable authorities that compare the accuracy and functionality of the more widely accepted models and arranging personal contact meetings with model suppliers to obtain additional information.

2.1 MODEL EVALUATION CRITERIA

Before the comparative analysis of the models, a set of model evaluation criteria was developed to guide the selection process. Input from the Phase IV PMO, along with CSC's understanding of the development process for secure systems, led to establishing the following:

1. Ability to handle special security considerations
2. Life-cycle modeling capability
3. Use of state-of-the-art concepts
4. Automated operation
5. Comparative accuracy
6. Established reputation
7. Availability
8. Ease of use.

The following paragraphs describe the relevance of each of the criteria.

2.1.1 Ability to Handle Special Security Considerations

For an existing model to be suitable for application to this task, it must be flexible to incorporate and process security-related factors. These factors include modification of the development phasing, complexity, and reliability requirements. The model must also account for additional tasks that are required in a secure system development. The existing models were evaluated for their capability to accommodate such variations in the development process.

2.1.2 Life-Cycle Modeling Capability

While development phase costs are significant, the maintenance and operational phases greatly add to overall cost. Even so, these phases could be ignored if they were expected to remain constant among the three security alternatives. However, because the three alternatives differ radically in terms of operational considerations, maintenance costs will also differ. Therefore, models selected must accommodate both software and hardware life-cycle costing.

2.1.3 Use of State-of-the-Art Concepts

The technology of cost and schedule modeling has been evolving since the mid-1960s. Models have become more sophisticated in terms of ability to assess the effects of a growing set of cost-related factors. In addition, data bases of historical data on software projects have been accumulated providing a firm basis for calibration.

Software development practices have also been evolving. Thus, in a very real sense, the models have been aiming at a moving target. Only the most recent models can be expected to come close to this target. Because the development of the security alternatives is expected to incorporate the most advanced software engineering methods, it is particularly important that the selected model be as up-to-date as possible.

2.1.4 Automated Operation

A few of the earliest models were amenable to manual operation. Currently, manual operation is no longer consistent with state-of-the-art modeling concepts. The large number of parameters, the complexity of the mathematics, and the volume of output all require an automated system.

From an operational standpoint, automation is equally essential. Rapid and reliable turnaround is necessary to:

1. Ensure reproducible results
2. Perform sensitivity analyses
3. Provide adequate calibration
4. Accommodate modifications to parameters in a timely manner.

2.1.5 Comparative Accuracy

Comparative studies indicate considerable differences in cost estimates. Mohanty's experiments [3] show a 6:1 ratio between the costs estimated by the most conservative and least conservative models. The major source of this variation is environmental. That is, each model has been generated based on particular historical data, and therefore reflects these data attributes. Most of these data bases are used in company-specific environments, and thus represent specific development practices and quality standards.

Mohanty was careful to emphasize that there is no single model that can be considered to be the best. None of the models successfully quantifies development practices and quality to the extent that the model becomes environment-independent. For the present task, it is essential to choose a middle-of-the-road model that is based on a broadly drawn data base to minimize this dependence.

2.1.6 Established Reputation

It is important to select a model that has an established reputation in the field of cost and schedule modeling. An established model has a number of major advantages:

1. The model is easier to calibrate. Drawing on past experience with the model makes it possible to estimate input parameters with greater confidence. The technical complexity parameters are a particularly important example of this.
2. The model permits an apples-to-apples comparison with previous costing exercises. Both the inputs and the outputs are directly comparable.
3. The model will be believable. The fact that the model is widely accepted in both Government and industry will lead to a greater confidence in its cost and schedule estimates.

2.1.7 Availability

The model must be readily available on a cost-effective basis. While there are a number of state-of-the-art costing algorithms requiring significant effort to implement, there is no reason to expect that they would outperform readily available, off-the-shelf models.

2.1.8 Ease of Use

The ease of using a model is related directly to the ease of preparing the requisite input parameters and analyzing the outputs.

Most of the manual effort required in using any automated model consists of estimating a variety of input parameters. These models differ in the range of input parameters that they process. Thus, there is a tradeoff between ease of use and flexibility. The more flexible model requires that the user estimate more parameters. The burden that is placed on the user is significant. The accuracy of the model's output totally depends on the accuracy of the user's input estimates. A highly flexible model requires that the user have considerable insight into the development process under analysis.

2.2 COST AND SCHEDULE MODEL ATTRIBUTES

This paragraph presents descriptions of the candidate cost and schedule models. The models are best characterized by descriptive attributes, indicating the model's ability to handle various cost and schedule factors.

The areas covered by the attributes include cost, manpower, personnel and productivity, schedule, system and program characteristics, development environment, status, operations and maintenance data, and additional costs, such as documentation and travel. Within each of these areas the attributes can be classified as principal, secondary, or informational attributes. Principal attributes are those that bear a strong relationship to properties needed in a model for this task. Secondary attributes are those that are not essential to satisfy task requirements but did contribute to the analysis. Informational attributes provide additional data about the models but did not play a significant role in the comparative analysis.

To facilitate comparison of the candidate models, Table 2-1 lists each of the models in terms of these attributes. Principal and secondary attributes are identified. The table lists standard entries which, because of limited space, have been assigned keys. These keys are identified and described below:

1. OUTPUT designates the parameters estimated by the model

2. INPUT OR OUTPUT indicates that the parameter, when known, can be input to estimate other parameters

3. INCLUDED indicates that a provision for the parameter was built-in.

THIS PAGE INTENTIONALLY LEFT BLANK

ATTRIBUTES	PRICE (1977)	[illegible]M 1979	DOTY (1977)	WOLVERTON (1974)	KUSTANOWITZ (1977)	AEROSPACE (1977)	GR (1974)	SDC (1967)	ESI (1975)
SIZE AND STRUCTURE OF DATA BASE									PRIMARY INPUT PARAMETER (MEASURE IS UNDEFINED)
SIZE OF MODULES	OPTIONAL INPUT	MEMORY SIZE IN WORDS OR OBJECT CODE (OPTIONAL OUTPUT)							
PROGRAM STRUCTURE			(NPP WAS NOT INCLUDED)		SUBJECTIVE ADJUSTMENT TO INPUT				FURTHER RESEARCH REQUIRED
*NEW PROGRAM	END CODE % (INPUT)			"OLD, NEW" (INPUT)					DEGREE OF FAMILIARITY
**TYPE OF PROGRAM OR SYSTEM	7 CATEGORIES (INPUT)	4 CATEGORIES INCLUDED	5 CATEGORIES (INPUT)	6 CATEGORIES (INPUT)		"REAL-TIME OR SUPPORT" (INPUT)		FACTOR FOR BUSINESS APPLIC (INPUT)	REAL-TIME OR NOT (INPUT)
**STATE OF TECHNOLOGY	SUBJECTIVE ADJUSTMENT TO INPUT	FACTOR-REQUIRES DATA BASE FOR CALIBRATION (INPUT)	(ADVANCED TECHNOLOGY WAS NOT INCLUDED)	SUBJECTIVE ADJUSTMENT TO INPUT					FURTHER RESEARCH REQUIRED
**COMPLEXITY FACTOR (DIFFICULTY)	INPUT	IS COMPUTED (COULD BE INPUT)		"EASY, MED, HARD" (CITES FACTORS TO CONSIDER) (INPUT)			INCLUDED - UNDEFINED MEASURE	INTERFACE - SCALED 0-2 (INPUT)	FURTHER RESEARCH REQUIRED ELIMINATED WBR BRANCHES, PATHS INTERFACES
FREQUENCY OF OPERATION								SCALE 0-5 (INPUT)	
ERROR CONTENT						ACCURACY REQUIREMENTS INCLUDED			
DEVELOPMENT ENVIRONMENT		FURTHER RESEARCH REQUIRED	4 CATEGORIES BELOW						SUBJECTIVE ADJUSTMENT TO OUTPUT
*COMPUTER CONFIGURATION	INPUT FOR CROSS-CHECKS ONLY	SHOULD BE CONSIDERED IN FUTURE						RANDOM ACCESS, YES/NO (INPUT)	
SPECIAL DISPLAYS			YES/NO (INPUT)					YES/NO (INPUT)	
DETAILED DEF OF HDWT			YES/NO (INPUT)						
**CHANGE TO HDWT	COMPLICATION INPUT		YES/NO (INPUT)			INCLUDED			
**REAL-TIME OPERATION	INSTR. MIX INPUT	INPUT	YES/NO (INPUT)						

LEGEND
** PRINCIPAL ATTRIBUTE
* SECONDARY ATTRIBUTE

Table 2-1. Principal Attributes of Candidate Models (1 of 4)

MODELS / ATTRIBUTES	PRICE (1977)	SLIM (1979)	DOTY (1977)	WOLVERTON (1974)	FUSTANOWITZ (1977)	AEROSPACE (197[illegible])	GRC (1974)	SDC (1967)	ESD (1975)
*PRODUCTIVITY RATE	INCLUDED (RESOURCE)	IS COMPUTED (COULD BE INPUT)	INCLUDED	INCLUDED	LINES PER DAY BASED ON EXP (INPUT)	INCLUDED	INCLUDED	INCLUDED	INCLUDED
SCHEDULE									
**PROJECT LIFE TIME	INPUT OR OUTPUT	IN YEARS (INPUT OR OUTPUT)							
**PROJECT DEVELOPMENT TIME	OUTPUT	IN YEARS (INPUT OR OUTPUT)	OUTPUT (IN MONTHS)		OUTPUT (IN MONTHS)				
**PROJECT MILESTONES	OUTPUT	OUTPUT		INPUT					
**DEVELOPMENT PHASES	INCLUDED	INCLUDED		INPUT 7 PHASES (INPUT)	INPUT	FACTOR FOR DISORGANIZED PHASES INCLUDED			
ADDITIONAL COSTS									
*DOCUMENTATION	INCLUDED	OUTPUT	CONSIDER IN SECONDARY COSTS RATIO	(ADD TO COST RATE)	(ADD TO MANPOWER REQUIREMENTS			WBS PAGES (EXTERNAL) (INPUT)	
*TRAVEL			CONSIDER IN SECONDARY COSTS RATIO	(ADD TO COST RATE)					
*COMPUTER USAGE	INCLUDED (RESOURCE)	ESTAB A DATA BASE/HOURS PER MO	CONSIDER IN SECONDARY COSTS RATIO	ESTAB A DATA BASE, HRS BY CATEGORY					
SYSTEM/PROGRAM CHARACTERISTICS									
**NUMBER OF INSTRUCTIONS	DELIVERABLE EXECUTABLE MACHINE INSTRUCTIONS	IN SOURCE STATEMENTS (INPUT OR OUTPUT)	IN THOUSANDS- SOURCE OR OBJECT (INPUT)	OBJECT- BY CATEGORY (INPUT)	LINES OF CODE (INPUT)	INPUT- CATEGORY IS UNDEFINED	MACH LANGUAGE (INPUT)	(1) % CLERICAL (2) % INFO STORAGE & RETR (INPUT)	DELIVERED - SOURCE, EXCL COMMENTS (INPUT)
**PROGRAMMING LANGUAGE	ANY LANGUAGE BY RATIO INPUT	INPUT			SHOULD BE CONSIDERED IN FUTURE	LANGUAGE INTERFACE INCLUDED	INCLUDED "HOL" OR MACHINE		CATEGORIES ARE UNDEFINED (INPUT)
*NUMBER OF FUNCTIONS	INPUT OR OUTPUT	OPTIONAL SIZING INPUT	(CAN BE USED TO COMPUTE SIZE) (OPTIONAL INPUT)						
*NUMBER OF PROGRAMS OR MODULES	RUN OF MODEL OR PROGRAM LEVEL	OPTIONAL INPUT							
NUMBER REPORTS & FILES		OPTIONAL INPUT							

LEGEND
** PRINCIPAL ATTRIBUTE
* SECONDARY ATTRIBUTE

Table 2-1. Principal Attributes of Candidate Models (2 of 4)

MODEL:

ATTRIBUTES	PRICE (1977)	SLIM (1979)	DOTY (1977)	WOLVERTON (1974)	KUSTANOWITZ (1977)	AEROSPACE (1977)	GRC (1974)	SDC (1967)	ESD (1975)
**CPU MEMORY CONSTRAINT	INPUT	SHOULD BE CONSIDERED IN FUTURE	YES/NO (INPUT)			INCLUDED FOR REAL-TIME	FRACTION OF UTILIZATION (INPUT)		
**CPU TIME CONSTRAINT	INPUT		YES/NO (INPUT)			INCLUDED	FRACTION OF UTILIZATION (INPUT)		
1ST TIME ON CPU	INPUT		YES/NO (INPUT)					YES/NO (INPUT)	
CONCURRENT H/W DEVEL	INPUT		YES/NO (INPUT)						
**TIME SHARE VS BATCH	INPUT	SHOULD BE CONSIDERED IN FUTURE	YES/NO (INPUT)		SUBJECTIVE ADJUSTMENT TO INPUT				
*DEVELOPER USING ANOTHER FACILITY	INPUT		YES/NO (INPUT)						
*DEVELOPMENT AT OP SITE	INPUT		YES/NO (INPUT)						
DEVEL COMPUTER NOT TARGET COMPUTER	INPUT		YES/NO (INPUT)						
**DEVEL AT MORE THAN 1 SITE	INPUT		YES/NO (INPUT)					NBR SITES (INPUT)	
*PROGRAMMER ACCESS TO COMPUTER	INPUT	SHOULD BE CONSIDERED IN FUTURE	YES/NO (INPUT)						
STATUS STATUS OF PROGRAMS WHEN VALIDATION BEGINS	INPUT								
**STATUS VALIDATION TOOLS	INPUT								
STATUS COMPUTER & OP SYSTEM							SHOULD BE CONSIDERED IN FUTURE		
O & M DATA	OUTPUT	ESTABLISH DATA BASE (CITES ITEMS TO CONSIDER)	EXCLUDED	EXCLUDED					
**O & M MANPOWER	OUTPUT	OUTPUT							
**O & M SIZE	OUTPUT	OUTPUT							
*ERROR OCCURRENCE TIME	INPUT								
*ERROR CORRECTION TIME	INPUT								

LEGEND
** PRINCIPAL ATTRIBUTE
* SECONDARY ATTRIBUTE

Table 2-1. Principal Attributes of Candidate Models (3 of 4)

MODELS / ATTRIBUTES	PRICE (1977)	SLIM (1979)	DOTY (1977)	WOLVERTON (1974)	KUSTANOWITZ (1977)	AEROSPACE (1977)	GRC (1974)	SDC (1967)	ESD (1975)
COST									
**LIFE CYCLE COST	OUTPUT	OUTPUT							
**DEVELOPMENT LC COST	OUTPUT	OUTPUT	OUTPUT	OUTPUT	OUTPUT $ PER MM	OUTPUT	OUTPUT IN 1973 $		OUTPUT
*AVERAGE COST PER UNIT	OPTIONAL INPUT ($ PER MM)	$ PER MAN YEAR (INPUT)	1) $ PER MM INCLUDING OVERHEAD, ETC. 2) RATIO PRIMARY TO SECONDARY (INPUT)	$ PER INSTRUCTION BY CATEGORY (INPUT)	$ PER MM	$ PER MM			$ PER INSTRUCTION (INPUT)
MINIMUM COST FOR ERROR CORRECTION		OUTPUT (1) TIME LIMIT FOR PRGRM CORRECTION (2) OPTIMUM SAMPLE SIZE							
MANPOWER									
*LIFE CYCLE MANPOWER		IN MAN YEARS (INPUT OR OUTPUT)							
*DEVELOPMENT MANPOWER	IN MAN MONTHS	IN MAN YEARS (INPUT OR OUTPUT)	OUTPUT (IN MAN MONTHS)		OUTPUT (IN MAN DAYS)	OUTPUT (IN MAN MONTHS)		OUTPUT (MM PER 1000 INSTR)	
PERSONNEL/PRODUCTIVITY									
*PROGRAMMER EXPERIENCE	INCLUDED IN COMPLEXITY TABLE	INPUT		ELIMINATED AS UNRELATED					SUBJECTIVE ADJUSTMENT TO OUTPUT
PRGMR PARTICIPN IN DESIGN								RATIO (INPUT)	
*PERSONNEL CONTINUITY	INPUT							RATIO (INPUT)	
*AVERAGE STAFF SIZE	OPTIONAL OUTPUT	PROJECTED STAFF ON BOARD (INPUT OR OUTPUT)	MANLOADING SHOULD BE CONSIDERED IN FUTURE						

LEGEND
** PRINCIPAL ATTRIBUTE
* SECONDARY ATTRIBUTE

Table 2-1. Principal Attributes of Candidate Models (4 of 4)

SECTION 3 - COMPARATIVE ANALYSIS

The basis for the comparative analysis was two-fold:

1. Satisfaction of evaluation criteria
2. Possession of desirable attributes.

The analysis was also influenced by the overall task objective to provide a comprehensive and consistent approach to the estimation process. This required consideration of the adaptability of the models to accommodate security-specific factors and the effort involved in the integration and analysis of model outputs with other factors.

These issues are addressed in the following paragraphs.

3.1 PRELIMINARY ANALYSIS

A major consideration in analyzing existing models was the ability of a model to handle special security-related factors. For example, the different development phases and their relative proportions are not directly handled by existing models. Instead, calibration of certain parameters or manual interaction is required to achieve the desired effect. Relative phasing of the development process, however, is extremely important for certain security alternatives.

The following two approaches were considered:

1. Develop a new model to directly handle secure system development methodologies
2. Develop a cost and schedule methodology that incorporates existing models, which adjusts the input parameters to properly handle secure systems development and integrates other factors not accounted for by the models.

The second approach, detailed in Part II, was found to be both possible and practical based on the following considerations:

1. Development of a completely new automated model is a major task, requiring several thousand lines of code. This effort is not feasible within the timeframe of this contract.

2. Manual operation of such a complex model may be feasible but is not practical.

3. A specially developed model would have no recognized credibility. Exposure to a host of users with varied applications and historical data is necessary to establish credibility.

4. Estimation algorithms must be calibrated to, or extrapolated from, a data base containing historical information on a number of projects. The analysis of such a data base requires a major effort.

5. There is little a priori reason to expect that a new model will generate more accurate results than an appropriately parameterized existing model.

6. A new model, incorporating security considerations, would lack baseline model for nonsecure development. This baseline is essential for calibration.

Based on this preliminary analysis, the best of the existing models, in terms of both evaluation criteria and desired attributes, would be most appropriate for the methodology. Therefore, the analysis proceeded with a detailed comparison of the existing models.

3.2 DETAILED ANALYSIS

The detailed analysis results are listed in Tables 3-1 and 3-2.

The principal attributes of the models identified in the study analysis are summarized in Table 3-1. The totals represent the number of these principal attributes possessed by each model. Table 3-2 rates each of the models against the established evaluation criteria. For each criterion, the model was scored to indicate the level of correspondence between the model's capabilities and the criterion. Totals for each model are also given.

It is apparent from these tables that the PRICE family and SLIM are clearly superior to other existing models in terms of the established evaluation criteria and possession of desired attributes.

Table 3-1. Summary of Principal Desired Attributes of Candidate Models

MODEL PRINCIPAL ATTRIBUTE	PRICE (1977)	SLIM (.979)	DOTY (1977)	WOLVERTON (1974)	KUSTANOWITZ (1977)	AEROSPACE (1977)	GRC (1974)	SDC (1967)	ESD (1975)
Cost:									
Life Cycle Cost	X	X							
Development Life Cycle Cost	X	X	X	X	X	X	X		X
Schedule:									
Project Life Time	X	X							
Project Development Time	X	X	X		X				
Project Milestones	X	X		X					
Development Phases	X	X		X	X	X			
System/Program Characteristics:									
Number of Instructions	X	X	X	X	X	X	X	X	X
Programming Language	X	X				X	X		X
Type of Program or System	X	X	X	X		X		X	X
State Of Technology	X	X			X				
Complexity Factor (Difficulty)	X	X		X			X	X	
Development Environment:									
Change to Requirements	X		X			X			
Real-Time Operation	X	X	X						
CPU Memory Constraint	X		X			X	X		
CPU Time Constraint	X	X	X			X	X		
Time Share vs. Batch	X	X	X		X				
Develop. at More Than 1 Site	X		X					X	
Status:									
Status Validation Tools	X								
O&M Data:									
O&M Manpower	X	X							
O&M Size	X	X							
TOTALS	20	16	10	6	6	8	6	4	4

Table 3-2. Correlation Matrix of Candidate Cost/Schedule Models and Evaluation Criteria

MODEL	PRICE (1977)	SLIM (1979)	DOTY (1977)	WOLVERTON (1974)	KUSTANOWITZ (1977)	AEROSPACE (1977)	GRC (1974)	SDC (1967)	ESD (1975)
EVALUATION CRITERIA:									
1) Use of state-of-the-art concepts	2	2	2	2	2	2	2	1	2
2) Automated operation	2	2	0	0	0	0	0	0	0
3) Comparative accuracy	2	2	2	2	2	2	2	2	2
4) Availability	2	2	0	0	0	0	0	0	0
5) Ease of use	2	2	1	1	1	1	1	1	1
6) Life-cycle modeling capability	2	2*	0	0	0	0	0	0	0
7) Ability to handle special security considerations	2	1	0	0	0	0	0	0	0
TOTALS	14	13	5	5	5	5	5	4	5

Scale of correlation indices:

0 = Nonexistent/indeterminate
1 = Some correspondence
2 = Direct correspondence

* Direct corespondence for software only

In addition, based on discussions with RCA, the suppliers of PRICE, we determined that the family of models could be incorporated into a comprehensive methodology and would be adaptable to security-specific factors.

In view of these considerations, the PRICE family of models has been selected as the primary estimation tool. Because SLIM utilizes a different algorithm, and can be executed at very little additional cost, it is suitable for cross-checking the software estimates generated by PRICE. Secure systems development represents a novel application for both PRICE and SLIM; therefore, this independent validation can be expected to increase confidence in the final estimates.

PART II

COST AND SCHEDULE METHODOLOGY

SECTION 1 - OVERVIEW

This part describes the composite steps that form the methodology. As a result of the analysis presented in Part I, the use of established, reliable, automated models to estimate hardware and software costs and schedules is the major tool in the methodology. This methodology incorporates automated tools and analysis techniques that quantify the input factors for the security alternatives and integrates the combined output information. To distinguish the security-related data (obtained through the alternative analysis) from the model inputs, the term factors is applied to the former and parameters to the latter.

The methodology has been divided into the following three steps:

Step 1 - Analysis of the security alternatives

Step 2 - Use of the automated models

Step 3 - Analysis of the output.

Each of the steps produce outputs that function as inputs for the following step.

Cost and schedule factors and constraints are identified and quantified through the analysis of security alternatives. The security factors identified in this report are based on the candidate scenarios that are described in Appendix A. These security factors are grouped into two different categories:

1. Factors that bear a direct relationship to or are built into the models
2. Factors not provided for explicitly in the automated models that are handled by a supplementary procedure in the methodology.

These security factors are identified and quantified to determine model parameters before execution of the models. Following the use of the automated tools, all output information is integrated and analyzed to evaluate the effect of each alternative. Through this integration and analysis, the methodology is provided to include the unique factors that are associated with any of the alternatives, and thus generally ensure that the analysis can be applied.

A basic description of each of the steps involved in the methodology is given in the following sections. Part III of this report details the applications of this methodology to the evaluation of security alternatives. Part III also describes an approach to automate the methodology.

Several operational steps are involved in the application of the methodology. Generally, these operations can be broken into two different stages: 1) Preliminary Analysis, and 2) Integration and Detailed Analysis.

During Preliminary Analysis, the methodology relies on two main sources to form its operation baseline. These two main sources are a detailed description of the security alternatives and a detailed description of the cost and schedule models, PRICE and SLIM. After the functional analysis of the alternatives to define and develop a clear understanding of each alternative, the model input parameters corresponding directly to the alternatives are identified. The next operation, Translation, involves assigning values to the input parameters identified in the previous step to describe the alternative in terms consistent with logical relationships built into the model. Once this has been accomplished (aided by a special input data worksheet), the model is invoked, starting with entry of the appropriate parameter values. Accompanying this is the selection of specific output reports to obtain the most suitable format and desired results. To establish a close relationship between what is being estimated at a given time and the accuracy of the model as an application tool, it is important to analyze the model results and determine whether calibration is required. This procedure requires a thorough working knowledge of what and how such changes to model parameters need to be invoked. The final step to the Preliminary Analysis stage is determining which model output figures are key to cost and schedule analysis integration, the second stage of the overall methodology.

The Integration and Detailed Analysis stage, in contrast to the Preliminary Analysis stage, is oriented more to human data analysis techniques rather than procedural methods for obtaining the estimates. The first action in this stage is to ensure that the cost and schedule figures (based on common input parameters) generated by the models are indeed the correct totals to be applied during this detailed stage of analysis. Because there will be two distinct model (PRICE and SLIM) outputs for cross-checking respective totals, this procedure is designed to act in parallel during this stage of the analysis. The first step to take place in this parallel mode is combining the associated component costs and schedules into an integrated summary. Following this is the inclusion of those additional security factors that are not directly specifiable to the models. These security values must be determined by some tool or technique other than the models. Such a factor, which can be considered to fit this case, is personnel clearance, that is how many personnel need to be cleared, to what classification level the person is to be cleared, and by when the respective clearances are to be finalized. After all component cost and schedule information has been allowed for by both PRICE and SLIM parallel functions, their results can be compared or cross-checked. Then the analyst must decide whether to accept the comparison (implying that the results are within a reasonable tolerance) or reject one or both of the model results. If rejected, reiteration of the analysis or calibration is required until the final comparison is deemed acceptable.

Each of these analysis stages is supported by the use of standard worksheets, as described in Part III, Section 4.

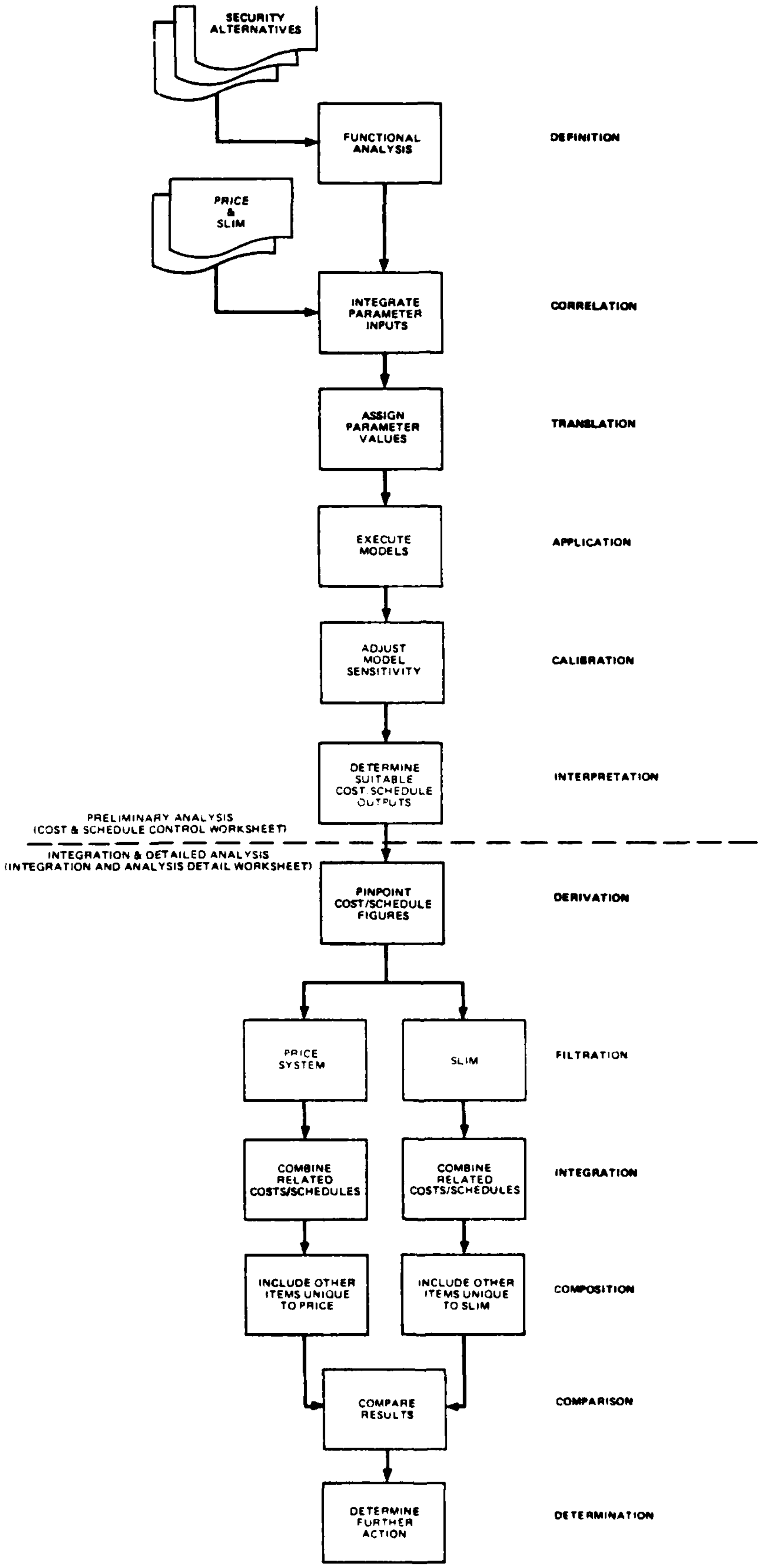


Figure 1-1. Diagram of Cost and Schedule Methodology Operations

SECTION 2 - ANALYSIS OF SECURITY ALTERNATIVES (Step 1)

The models used in the methodology take into account a large number of cost and schedule parameters. In order for the Government to supply the necessary data for the derivation of these parameters, the security factors must first be identified. Because the historical data for security alternatives under consideration is limited, the scenarios described in Appendix A have been used to delineate some of these factors. In addition, each of the security alternatives must be analyzed carefully in light of the accreditation criteria to determine the corresponding effects on the development process.

The determination of cost and schedule factors and constraints will be accomplished through analysis of each of the proposed security alternatives.

The general factors to be considered include:

1. Hardware and software sizing
2. Number and type of encryption devices
3. System security policy implementation
4. Certification and accreditation criteria
5. System configuration
6. Development phasing
7. Development and support resources requirements
8. Effect of security requirements on complexity, reliability, efficiency, and maintainability
9. Documentation requirements
10. Testing requirements.

These general factors are described in detail in Part III of this report. Although not all of this information is available at the present time, further information exchange will take place through the technical evaluation meetings.

The second stage of the analysis is the translation of appropriate security factors into the format required for use as model parameters. At present, the explicit translation process cannot be specified due to lack of comprehensive training and documentation on the automated models. Implementation of this stage of the methodology requires this training for both the PRICE and SLIM models.

Nevertheless, this step of the methodology is facilitated by the use of the Cost and Schedule Control Worksheet, shown in Figure 2-1. The directions for using this worksheet are detailed in Part III of this report. This worksheet provides a single point of reference at which all of the information for a particular alternative is available. The worksheet identifies information about the particular run, including the security factors considered, the models to be executed, and, where appropriate, historical data from previous executions. It also provides a choice of output reports. The analyst can supplement this worksheet with comments as well as attachments.

The results of the security alternative analysis process include identification and quantification of security factors and initial preparation for model executions. Security factors that are not handled directly by the models will be prepared for the integration and analysis process.

I-S/A AMPE
Cost & Schedule Analysis Control Worksheet

Date (mm/dd/yy):

Run Control No.: A -C -I

Analyst Name(s):

Security Alternative Ident: ____________

Component Ident: ____________

Objective of Analysis (what, how)

Security Issues To Be Considered:

___ 1. Security Policy Model
___ 2. Definition of Secure Software
___ 3. Level of Assurance
___ 4. Specification Language
___ 5. Development Phasing
___ 6. Complexity
___ 7. Development Machine
___ 8. Reliability
___ 9. Efficiency
___ 10. Documentation
___ 11. AMPSSO Assignment
___ 12. Personnel Clearances
___ 13. Physical Security
___ 14. COMSEC
___ 15. TEMPEST
___ 16. Information Sanitization
___ 17. Accreditation

Models To Be Executed:

Price S ___ Price SL ___ Price ___ Price L ___ SLIM ___

Model Inputs (attach corresponding Input Data Worksheets):

Input Data Worksheet Control No.: ______

Original Input Data Worksheet Control No.: ______

Prior-Run Key Sensitivity Factors and Values:

1. 2. 3.

4. 5. 6.

Data Validation (full printout of Model Inputs data file)

Calibration (attach operations printout of changes or indicate changes)

Figure 2-1. Cost and Schedule Control Worksheet (1 of 2)

Changes: 1. ____________ 2. ____________ 3. ____________

4. ____________ 5. ____________ 6. ____________

Expected results of these changes: ________________________________

____

____

Data Validation (full or partial - showing changes only - printout of calibration operations)

Model Outputs Desired:

Price S

__ 1. Cost & Sched. Det. Rpt. -Partial, Full
__ 2. RESO, CPLX Sens. Data Rpt.
__ 3. APPL, INST Sens. Data Rpt.
__ 4. Monthly Progress Summary
__ 5. Schedule Effect Summary

Price SL

__ 1. Cost Detail Rpt.-Partial, Full
__ 2.

PRICE

__ 1. Cost & Sched. Det. Rpt.-Partial, Full
__ 2.

PRICE L

__ 1. Cost Detail Rpt.-Partial Full,
__ 2.

SLIM

__ 1. Simulation
__ 2. Manloading
__ 3. Cashflow
__ 4. Code Production
__ 5. Life Cycle
__ 6. Milestones
__ 7. Front End
__ 8. Risk Analysis
__ 9. Pert Sizing
__ 10. Documentation
__ 11. Benefit Analysis
__ 12. CPU Usage
__ 13. Linear Program
__ 14. Interactive Linear Prog.
__ 15. Design-to-risk
__ 16. Design-to-cost
__ 17. Design-to-Schedule
__ 18. Best Bid

Other Considerations:

1. ____
2. ____
3. ____

Comments: ____

____

____

Figure 2-1. Cost and Schedule Control Worksheet (2 of 2)

SECTION 3 - AUTOMATED MODELS USAGE (Step 2)

3.1 OVERVIEW

Automated model usage requires preparation of input parameters, initial calibration of the models, and execution of the models. These activities are briefly described below. Part III provides details on the procedures involved in automated models usage.

Having determined the cost and schedule factors and constraints (Step 1), Step 2 begins by preparing inputs for using the automated models. Each of the factors identified and quantified in Step 1 must be analyzed for transformation into input parameters for the models. Some of these factors, such as estimates of hardware and software size, become input parameters directly. Others, such as organizational efficiency and project complexity, are analyzed in light of previous experience with the model to quantify them as input parameters. Additional factors that do not translate into model parameters are also identified.

Certain elements of the cost and schedule parameters are used to establish the environment to be used as the basis for estimation. Usually, models are calibrated to historical data from past projects within a particular organization. Because this organization cannot yet be specified for the security alternatives, calibration for the present purpose will use the following information:

1. Historical data from reasonably similar projects.
2. An understanding of the special exigencies entailed by secure system development.

Once the input has been prepared and the models have been calibrated, they are executed.

3.2 MODEL CAPABILITIES

As discussed in Part I, the PRICE family models has been selected as the principal cost and scheduling evaluation tool. SLIM will be used to cross-check software estimates.

A brief description of each of these models is presented below.

3.2.1 PRICE-S

The PRICE Software (PRICE-S) Model uses parametric relationships to relate new software projects to costs and schedules that are typical of the work to be accomplished. Organizational performance factors are adjusted in a calibration mode to fit the model to specific environments.

PRICE-S is an interactive model. Following the entry of project descriptions, the model derives and displays projected costs for each of three development phases. These phases are Design, Implementation, and Test and Integration.

The model also computes typical schedules for the size, type, and difficulty of the project described. If desired, manpower and scheduling constraints that apply to the software development effort can be specified. Table 3-1 lists the software development factors that PRICE-S addresses, either as input or output.

3.2.2 PRICE-SL

PRICE-SL is used to estimate post-development support costs. PRICE-SL can be calibrated to match a particular organization and project. The major activities that PRICE-SL considers are maintenance, enhancement, and anticipated growth.

The majority of PRICE-SL input parameters are identical with PRICE-S input.

The Support Economics and Environment data is new information used to define the cost level, economic scale, escalation considerations, support length, number of support locations, and level of anticipated growth.

The basic PRICE-SL output report provides cost estimates for the specified support life, along with a record of the project descriptions. A table of costs for each year of the support life, with costs distributed among the maintenance, enhancement, and growth activities is also possible.

Table 3-1. Summary of Cost and Schedule Elements Addressed by PRICE-S (Software Development)

Project size	Intensity of effort
Project type (MIS, radar, telemetry, etc.)	Changing requirements
Operational customer environment	Programming language
Hardware constraints (system loading)	Compiler power and efficiency
Existing design	Development location (in-house or on-site)
Existing code	Project complexity
External interfaces (type and quantity)	Engineering requirements
Hierarchical design functional flow structure	Programming requirements
Number of functions performed	Configuration control
Amount of code per function	Documentation
Schedule constraints, lead times and overlaps	Program management
Resource constraints	Design phase activities
Engineering Change Notice effects	Implementation activities
Economic trends	Test and Integration activities
Technology growth	Integration of independent projects
Fee, profit, and G&A	Verification and validation
Computer operation costs	Multiple test beds/installations
Overhead	Government-furnished software
Organizational efficiency	Purchased software (such as, subcontracts)
Skills	Design-to-cost
Project familiarity	Resource allocation with respect to time

3.2.3 PRICE

The PRICE model derives cost estimates of hardware assemblies and systems. The basic PRICE model provides estimates of system acquisition costs based on physical parameters such as quantity, size, weight, power consumption, environmental specification, type of packaging, and level of integration; and schedule parameters such as months to first prototype, manufacturing rate, and amount of new design. PRICE is particularly useful in developing relative costs of competitive systems. Table 3-2 lists fundamental PRICE parameters.

PRICE estimates the cost associated with design, drafting, project management, documentation, sustaining engineering, special tooling and test equipment, material, labor, and overhead. Costs to integrate subassemblies into a system and to test the system for required operation are also estimated by the model. Costs for field test, site construction, and software can be processed by the PRICE hardware model.

PRICE generates costs for the development and production phases. PRICE can also develop an engineering schedule or measure the reasonableness of an input schedule. Variations of parameters such as physical features, component configuration, percentage of new design, reliability, and Mean-Time-Between-Failure (MTBF) can be quickly assessed. Integration and test costs for both engineering and production can be developed by PRICE at any level of a work breakdown structure.

PRICE has provisions to include the costs for Government-Furnished Equipment (GFE) and purchased items. It can also evaluate the costs of related testing, modification, and integration.

3.2.4 PRICE-L

The PRICE Life-Cycle Cost (PRICE-L) Model computes support costs for hardware systems. PRICE-L operates in conjunction with the basic PRICE model.

PRICE-L user inputs can be limited to factors for the equipment deployment, maintenance policy and levels of support capability, equipment and maintenance locations, and equipment life span. All other required inputs are

Table 3-2. Fundamental Parameters to the PRICE Model (Hardware Development)

1. Quantities of equipment to be developed, produced, modified, purchased, furnished and/or integrated and tested.

2. Schedules for development, production, procurement, modification integration and testing, including lead time for set-up, parts procurement, and redesign.

3. Hardware geometry consisting of size, weight of electronic and structural elements, and electronic packaging density.

4. Amount of new design required and complexity of the development engineering task.

5. Hardware structural and electronic design repeat.

6. Operational environment and specification requirements of the hardware.

7. Type and manufacturing complexity of the structural/mechanical and and electronics portions of the hardware.

8. Fabrication process to be used for production.

9. Pertinent escalation rates and markups for general and administrative charges, profit, IR&D, cost of money, and purchase item handling.

10. Technological improvement.

11. Yield considerations for hardware development.

developed by the PRICE model. During the use of the PRICE Model, the user may generate a life-cycle cost (LCC) data file consisting of these required life-cycle cost variable inputs.

Values developed by PRICE for input to PRICE-L include:

1. Number of module, part and the weight, volume, and cost of modules and parts

2. Development and production costs and schedules

3. MTBF and Mean-Time-To-Repair (MTTR) for all repairable assemblies

4. Test equipment costs.

In addition, PRICE-L incorporates many global values that can be changed to represent various service maintenance and supply organizations.

Costs for training, field installation and testing, site preparation and maintenance, operations, software, and energy, can be processed to be included in the LCC totals.

3.2.5 SLIM

SLIM is a cost and schedule tool for software life-cycle estimation. Using the Performance Evaluation and Review Technique (PERT) algorithm, linear programming, and Monte Carlo simulation and sensitivity profiling techniques, SLIM provides cost, time, personnel, and machine projections for developing software systems. SLIM identifies the limiting constraints that can block or alter development plans. Confidence levels and risk factors are calculated by SLIM.

SLIM requires the following inputs:

1. Environment and Technology Constant - Accounts for development environment factors, such as language, tools, development machine, target machine, modern programming practices (MPP), skills of people, complexity of task, and others

2. Degree of concurrency in executing phases and subtasks - accounts for the difficulty gradient or level

3. System size - Entered as ranges to determine uncertainty

4. Cost elements - Such as, labor rate and uncertainty

5. Management constraints - Such as, maximal allowable cost and permissible time, minimal and maximal peak manpower, percent of risk of not exceeding a specific delivery date.

SLIM provides information such as:

1. Identification of minimum cost, minimum time, and feasible solutions for a particular software development project

2. Optimal risk-protected schedule for completion with associated milestones

3. Manloading and cashflow projections on a monthly, quarterly, or yearly basis for the entire life cycle with appropriate uncertainty measures

4. Risk profiles for schedule, effort, inflated and uninflated costs, manpower, and budgets

5. Identification of constraints that may affect manpower application and completion schedules.

A correlation of PRICE and SLIM inputs and outputs is provided in Appendix C of this report.

SECTION 4 - OUTPUT ANALYSIS (Step 3)

The outputs of the automated models must be integrated with any supplemental cost and schedule data to provide a comprehensive final report for each security alternative.

For each of the alternatives, separate but comparable reports will be prepared. These reports will summarize the integrated results of the models and identify those other aspects not measured by the models that affect costs and schedule, such as the actual accreditation process system deployment, and system operation.

The integration and analysis process is facilitated through the Integration and Analysis Detail Worksheet, as shown in Figure 4-1. This worksheet provides for a summary of both cost and schedule results for a given security alternative. The cost part of the worksheet includes output from the models used as well as additional security factors not addressed by the models. The schedule portion serves to combine the schedule outputs of the models. Procedures for using this worksheet are detailed in Part III of this report.

I-S/A AMPE
Integration & Analysis Detail Worksheet
Part 1 - Costs

Date (mm/dd/yy):

Run Control No.: A_-C_-I_

Analyst Name(s):

Security Alternative Ident: ______________________

Component Ident: ______________________

Model or Models Employed This Session:

Phase / Division	Development	Support
Software Hardware	Price S ___ SLIM ___ Price only ___	Price SL ___ SLIM ___ Price L only ___

COSTS		Units	Subtotal	Total
Software:				
	Development --	X		
	Support --	X		
			XX	
Hardware:				
	Development --			
	Engineering	X		
	Manufacturing	X		
			XX	
	Support --			
	Equipment	X		
	Support Equip.	X		
	Supply	X		
	Supply Admin.	X		
	Manpower	X		
	Contactor Support	X		
	Other	X		
			XX	
	Additional			
	Energy	X		
	Training	X		
	Other	X		
			XX	
				XXX

Figure 4-1. Sample Integration and Analysis Detail Worksheet (1 of 3)

I-S/A AMPE
Integration & Analysis Detail Worksheet
Part 1 - Costs (Cont'd)

Security Specific:

A. Secure Operating System

AMPSSO Assignment	X	
Personnel Clearances	X	
Physical Security	X	
COMSEC	X	
TEMPEST	X	
Sanitization of Info	X	
Accreditation	X	
Other	X	
		XX

B. Hardware Separation

AMPSSO Assignment	X	
Personnel Clearances	X	
Physical Security	X	
COMSEC	X	
TEMPEST	X	
Sanitization of Info	X	
Accreditation	X	
Other	X	
		XX

C. End to End Encryption

AMPSSO Assignment	X	
Personnel Clearances	X	
Physical Security	X	
COMSEC	X	
TEMPEST	X	
Sanitization of Info	X	
Accreditation	X	
Other	X	
		XX

Figure 4-1. Sample Integration and Analysis Detail Worksheet (2 of 3)

I-S/A AMPE
Integration & Analysis Detail Worksheet
Part 2 - Schedule

Date (mm/dd/yy):

Run Control No.: A_-C_-I_

Analyst Name(s):

Security Alternative Ident: ____________________

Component Ident: ____________________

Model or Models Employed This Session:

Phase / Division	Development	Support
Software	Price S ___ Price A ___ SLIM ___	Price SL ___ SLIM ___
Hardware	Price only ___ Price A ___	Price L ___ Price A ___

SCHEDULES

		Start Date	End Date
Software:			
	Development --		
	Design	MMYY	MMYY
	Implementation	MMYY	MMYY
	Test & Integ.	MMYY	MMYY
	Support --	MMYY	MMYY

		Start	1st Item	Finish
Hardware:				
	Development --			
	Development	MMYY	MMYY	MMYY
	Production	MMYY	MMYY	MMYY
	Support --	N/A	N/A	N/A

Note: Separate or combined activity profiles can be generated by Price A (Activity Distribution Model) from the schedule information shown above.

Comments: ____________________

Figure 4-1. Sample Integration and Analysis Detail Worksheet (3 of 3)

SECTION 5 - JOB STREAM APPLICATION

In the development of this methodology, certain facts were noted that led to the examination of the possibility for automating the methodology. This approach would use a job-stream application to consolidate the input analysis, model execution, and integration and analysis processes. A brief description of this optional approach is given below.

The consideration for a job-stream approach arose from the choice of the PRICE family of models as the primary tool in the methodology. The members of the PRICE family that are to be used are:

PRICE-S - Software Development

PRICE-SL - Software Life Cycle (Operations and Maintenance)

PRICE - Hardware Development

PRICE-L - Hardware Life Cycle (Operations and Maintenance).

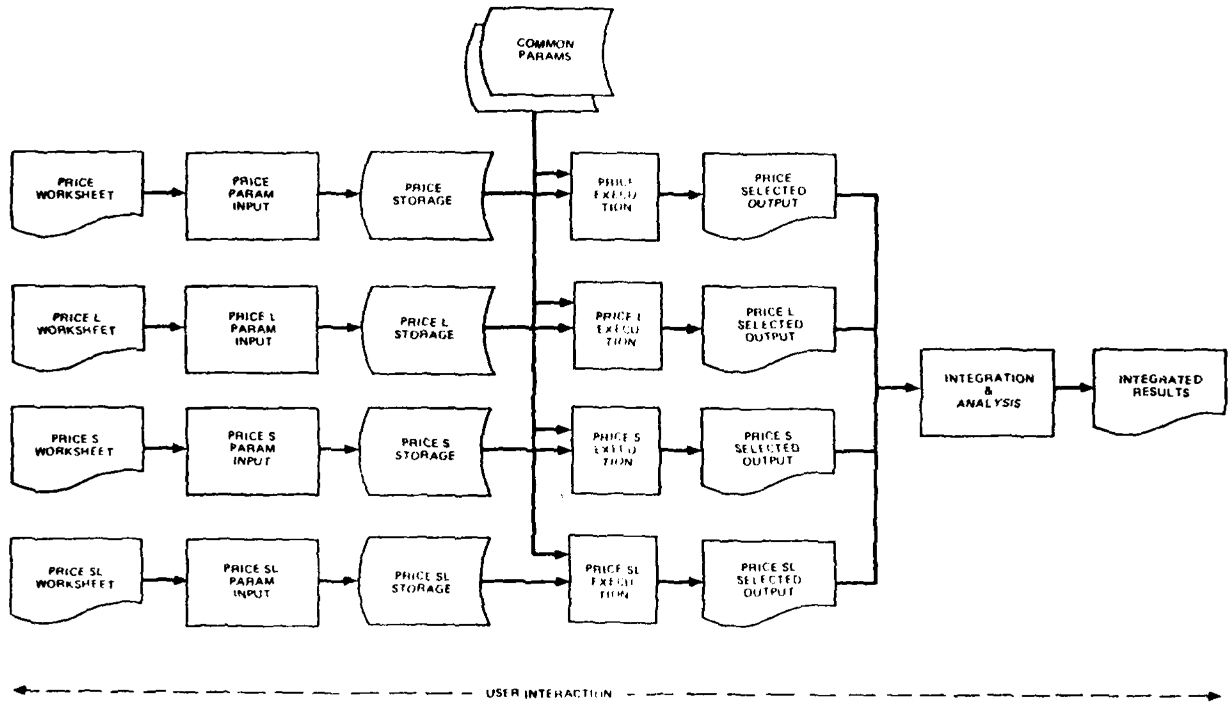
These models are functionally and logically connected to obviate the need for repetitive data manipulation operations. However, operator interaction is required for the user to move from one model to the next.

The job-stream approach unifies the procedures needed to use the PRICE family and provides the analyst with a single input and single output. The functionality of this job-stream approach can be illustrated by comparison with the existing PRICE family.

Figure 5-1 illustrates the current PRICE system model segmentation. Each model operates independently and user interaction is required throughout the process.

Figure 5-2 illustrates a possible PRICE system reconfiguration. User interaction is required here only to set up the combined input worksheet and to invoke the system driver. The models are then automatically invoked and a combined output is produced. An additional post processor is also included to give the user selected output in the desired format.

Although this basic approach appears to be both feasible and practical, the explicit design of the application program can only be completed after further details about the PRICE models are gained through training and



TP No 022 4665 B

Figure 5-1. Current PRICE System Model Segmentation

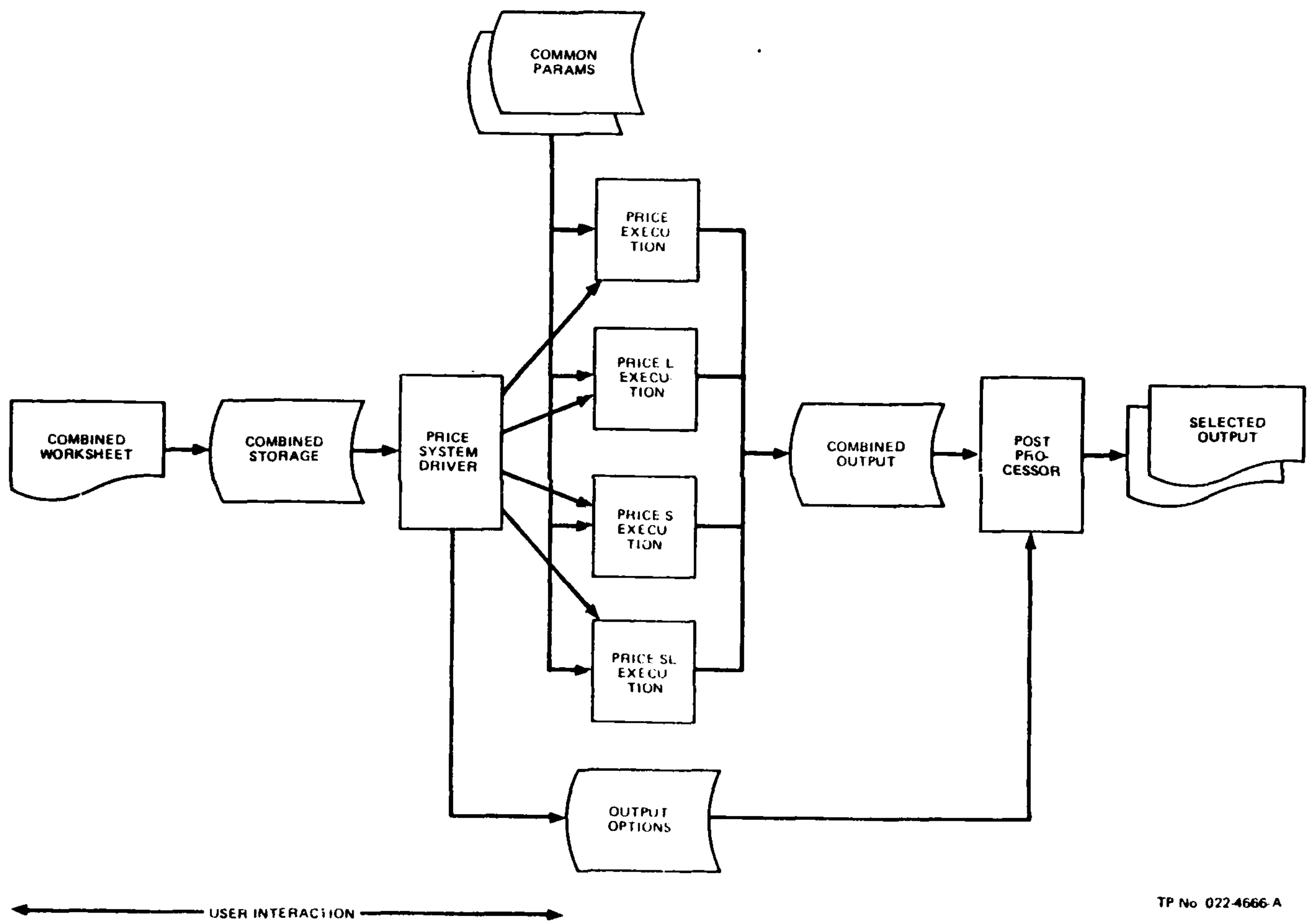


Figure 5-2. Possible PRICE System Reconfiguration

documentation. It is anticipated that the combined input worksheet will be modeled after the existing PRICE model data input worksheets and also include the information needed for the Cost and Schedule Control Worksheet, which incorporated security-specific factors.

A second consideration in selecting this approach is retention of user interaction. The objective of the job-stream application program is to relieve the cost analyst from the more tedious and mundane tasks, while retaining the ability to use the full capabilities of the models.

It is expected that such a job-stream application will not prohibit the use of a single model nor will it interfere with the analyst's ability to calibrate the models. It will, however, provide the option of streamlining the execution of more than one model without user interaction.

PART III

APPLICATION OF THE METHODOLOGY

SECTION 1 - OVERVIEW

This part of the report provides direction to the cost and schedule analyst in applying the methodology. The following sections address:

1. Input analysis
2. Execution of the models
3. Integration and analysis.

Discussion of the input analysis process focuses on the identification and quantification of factors associated with each alternative. Areas that affect cost and schedule are listed and requirements for their specification are described. An example of a typical input data worksheet is also given.

The discussion of the model execution process details the calibration and execution of the PRICE models using illustrative examples. Finally, the integration and analysis process is described in terms of operational approach. This incorporates the results of the input analysis as well as output analysis.

SECTION 2 - INPUT ANALYSIS

To apply the automated models to the evaluation of security alternatives, it is necessary describing the alternatives in terms of common and specific input requirements. The required inputs are presented in this section.

2.1 COMMON INPUT REQUIREMENTS

For each security alternative, the following input parameters, or factors, are needed to use the PRICE models. These inputs are common to all three security alternatives but the values used may differ. For example, with the secure operating system alternative, emphasis is placed on the requirement for experienced personnel and the effort required to specify and formally verify the software system. The other two alternatives do not need such heavy emphasis in this area. Common input factors are:

1. Software size - PRICE S requires input of number of executable machine-level instructions. This can be computed from the number of expected high order language statements and a conversion factor.

2. Software Mix - This factor is the percentage of software devoted to each kind of application, such as: operating systems, online communications, and data storage and retrieval.

3. Peripheral Devices - The number and type of all interfacing equipment are required, including:

   a. Data storage and retrieval devices
   b. Online communications devices
   c. Real-time command and control devices
   d. Interactive communication devices.

4. Personnel Characteristics - Anticipated personnel-related factors, such as general level of experience, as well as experience with similar work are necessary. In most cases these characteristics can be estimated as average values. In the case of secure system development, it is expected that the general level of experience will be higher than normal, although specific experience would be lower.

5. Level of New Design - The degree to which existing designs can be incorporated into the system is required for both software and hardware estimation. It should be given as a fraction of the total effort and of the software mix only.

6. Code or Equipment Availability - This factor provides a method for incorporating existing software and equipment into the system. It should be given as a fraction of the total effort.

7. Complexity - Factors such as personnel experience, product familiarity, and nature of the system contribute to complexity factors. A standard value will be used initially subject to calibration based on other input information.

8. Schedule - As a minimum, the start and end dates for the system development and deployment must be specified. Additional dates for development, testing, and integration phases can be supplied or computed by the model. Expected system life is also needed to compute support costs.

9. Deployment and employment - This factor includes the number of installations, maintenance facilities, system usage, and availability of hardware spares.

10. Resource - This factor represents a composite value based on the organizational capabilities, experience and individual talents of the activity that is to perform the work. A standard value is used by the model, although better results may be obtained with a slightly higher value in the case of secure system implementation. This factor needs to be analyzed during model calibration to determine the precise effect of changes.

11. Platform - This major factor summarizes the operational requirements in terms of specifications and reliability. This factor takes into account the effect of changing requirements, such as evolving security accreditation criteria. A standard value of 1.2 (MIL-SPEC Ground Operating Environment) will be used in initial estimates, which are subject to calibration.

12. Overhead, Q&A - This optional factor is a linear multiplier for all project costs used to accomodate contractor burden rates.

13. Escalation rate - Estimated cost inflation factors.

14. Hardware magnitude - The quantities of equipment to be developed, produced, modified, purchased, furnished and integrated and tested must be specified. The following data is required:

    a. Number of units - The number of production units and prototype units to be built

    b. Physical characteristics - The weight and volume of each unit

    c. Electronic design characteristics - Packaging density, manufacturing complexity, percentage of new design, and reliability requirements

    d. Schedule - The start and end dates of the development phase, the completion dates for the first and last prototype units and the first and last production units.

2.2 SECURE OPERATING SYSTEM FACTORS

The following paragraphs briefly describe information needed to use the models to estimate cost and schedule for the development of a secure operating system.

1. Security Policy Model - The development of a mathematical model of the security policy that will be enforced will affect the length of requirements definition phase and the manpower required then. This affects both the cost and schedule of the entire program. It must be specified whether a formal mathematical model of security, in terms of information and control flow, will be required. If so, specification of whether an existing model, such as Bell-LaPadula [1], or a new model is to be used is required.

2. Definition of Secure Software - Portions of the system that are to be secure must be defined explicitly, for example, the access control mechanism, including the size of such software. Division of system software into verifiable code, trusted processes, and untrusted

processes is also required. The function and size of each division is also required. Expected structure of the software can also be used in this definition.

3. Level of Assurance - Specific guidance and criteria for secure software accreditation must be established, for example, a Nibaldi [4] Level-3 system implies a different level of verification than a Level-4 system. Therefore, the level of assurance should be specified in terms of the Nibaldi levels.

4. Specification Language - If a formal specification language is required, the existence and availability of verification tools for that language will influence the time and manpower needed to perform the required verification. It must be stated whether formal specification and verification will be required. If they are required, additional information, such as the availability of verification tools, is required.

5. Development Phasing - Based on the extent of required verification, the design, implementation, and testing phases of the model should be calibrated to directly reflect the particular phase distribution associated with the alternative. This factor can be determined largely from information gained in the above areas. However, for calibration purposes, past Government experience in the development of secure systems with respect to development phasing is helpful. This will allow the user to incorporate deviations from the normal (40 percent design, 20 percent code, 40 percent test) development phasing into model input data.

6. Complexity - In addition to the complexity factor for the entire system, the complexity factor for the secure software must be identified. That is, if the entire operating system is to be secure, as opposed to an access controller for the data base, the complexity factor must be increased. Information required to derive the effect of this factor is supplied primarily by the definition of the secure software and is used in model calibration.

7. Development Machine - Computer time and resource requirements will increase with the use of automated verification tools. Anticipated development environment requirements arising from the use of automated verification tools and the size of the software system can be derived from other information.

8. Reliability - The reliability factor of the models may need modification to reflect extremely high reliability requirements. For example, the software may need to recover from user and system errors without operator intervention. Reliability requirements based on security-specific requirements should be specified in the model calibration process.

9. Efficiency - The importance of efficiency in the application will influence the software requirements. Therefore, the level of expected efficiency should be specified in terms of acceptable degradation when compared to similar systems. Other performance factors, such as response time, and throughput, should also be defined.

10. Documentation - Increased documentation requirements due to the verification effort will affect the cost of the entire system. Therefore, the type and level of documentation required to support the development, testing, accreditation, and operation of secure software must be specified. This can include requirements for verification planning documents as well as modification to existing documentation standards to incorporate formal specifications.

2.3 HARDWARE SEPARATION FACTORS

The hardware separation alternative has its primary effect in terms of hardware costs and schedules. The common input factors, listed in Paragraph 2.1, need to be addressed in addition to the following issues:

1. Design - The extent to which new hardware will be designed and developed, as opposed to using existing hardware, must be determined. Availability of appropriate off-the-shelf hardware should be specified.

2. Sizing - The processing power and efficiency needed dictate the type and number of processors that comprise the system architecture. In particular, separate microprocessors to implement hardware separation has a different affect than using mainframes.

3. Software Considerations - Implications of multilevel security as it affects software requirements for this alternative should be defined and specified as indicated in Paragraph 2.2.

2.4 END-TO-END ENCRYPTION FACTORS

The end-to-end encryption alternative has its major effect in terms of hardware costs and schedules. The following issues need to be addressed in addition to the common input factors given in Paragraph 2.1:

1. Number of crypto devices - Choosing an end-to-end encryption alternative will affect the number of crypto devices required. Interface requirements to new or existing equipment must be defined based on the particular architecture.

2. Design - The extent to which new crypto devices must be designed and developed, as opposed to using existing equipment, must be defined. This can be expressed in terms of the number of new devices or as a percentage of the total number required.

3. Software Considerations - Implications of multilevel security and encryption capabilities on the system software requirements must be identified and specified as indicated in Paragraph 2.2.

2.5 ADDITIONAL REQUIREMENTS

Although the automated models are powerful tools to use when evaluating proposed security alternatives, they cannot be fully effective without a comprehensive methodology to support them. The primary requirement for this methodology is an understanding of secure system development in terms of security-related features that affect cost and schedule. This paragraph addresses the identification and quantification of security factors that are not directly handled by the models but must be considered in the analysis of the security alternatives. All of these factors must be addressed in order to satisfy the requirement for accreditation, and are described in more detail in Do" C-5030-58-M.

2.5.1 AMPSSO Assignment

An Automated Message Processing Systems Security Officer (AMPSSO) must be assigned to coordinate and monitor the enforcement of all security policies and directives. Each AMPE site may have an individual entrusted with this responsibility. The cost associated with the AMPSSO begins at the time of deployment and continues through the life cycle. The exact duties of the AMPSSO, and therefore the determination of whether one person can assume these responsibilities for more than one site, is dependent on the selected security alternative. For example, if hardware separation is provided with a dedicated switching architecture, the AMPSSO would be responsible for ensuring that no DSSCS information remain in the processor when the change is made to GENSER. Depending on the frequency of such changes, this could require an AMPSSO for each installation. If end-to-end encryption is provided at the terminal level, making the I-S/A AMPE a BLACK processor, the AMPSSO responsibilities would be reduced to monitoring the correct functioning of the crypto-equipment. Therefore, the duties and responsibilities of the AMPSSO must be specified in order to include the associated manpower costs for each option.

2.5.2 Personnel Clearances

A TOP SECRET clearance may be required for all personnel working directly with the I-S/A AMPE. Other personnel, such as system software programmers, may need to be cleared to system high to have access to all of the compartmented information handled by the I-S/A AMPE. The costs associated with providing such personnel clearances affect both development and operational phases of the system. The number and levels of personnel clearances needed vary according to the security alternative chosen and the corresponding level of required assurance. For example, if a secure operating system is provided in the I-S/A AMPE, all users are not required to be cleared to system high. For each alternative, an estimate of the number and level of clearances must be supplied. The number of personnel requiring clearances may be stated in terms of relative numbers, such as, all systems programmers.

2.5.3 Physical Security

The physical facilities must be accredited and physical access strictly controlled. Again, the level of physical protection required will vary with the chosen alternative. This effect can be seen easily in the case of hardware separation provided by dedicated processors and associated workstations, which must be physically separated and controlled. For each alternative, an estimate of the cost for physical security must be provided. Specific required information includes anticipated number of sites, physical protection mechanisms such as locks, and required security guards.

2.5.4 COMSEC

COMSEC equipment must be National Security Agency (NSA) approved. This may affect cost and schedule depending, in part, on whether existing equipment can be effectively used or whether new equipment must be designed and built. This consideration primarily affects the end-to-end encryption alternative. Therefore, an estimate of the number and types of COMSEC equipment must be specified for each alternative. The degree to which existing equipment can be used will be a major factor as will the accreditation procedures for the equipment. This factor will also influence the parameters needed for the hardware model.

2.5.5 TEMPEST

TEMPEST requirements will affect the cost and schedule of any hardware procurement. Allowances for TEMPEST testing must be considered. Multi-microprocessor implementation of hardware separation might require that new hardware be TEMPEST approved, while approved equipment for an end-to-end encryption solution might already be available. For each alternative, TEMPEST requirements must be specified in terms of equipment to be certified and the possibility of using existing off-the-shelf equipment.

2.5.6 Sanitization of Information

Sanitization and declassification of the information processed by the I-S/A AMPE will indirectly affect the cost of security (from a procedural point of view), throughout the life cycle of the system. This effect can be measured by the amount of transmitted classified traffic. The classified mix of traffic expected should be specified. An estimate of the frequency of sanitization procedures, (daily, monthly, or other specified interval) should also be given.

2.5.7 Accreditation

The major cost and schedule impact of accreditation requirements is due to the accreditation process. The accreditation criteria must be established before the development of the system. These criteria must be clearly stated and remain constant throughout the system acquisition. The criteria must indicate the level of assurance required. A statement on how the criteria will be achieved and measured must be included. Without establishing these criteria at the beginning of the acquisition, any cost or schedule impact based on design and implementation activities aimed at satisfying these criteria will not be accurately forecast. In addition, the cost and time needed to establish that the criteria are satisfied must be incorporated into the estimate. If the criteria are those established in DoD C-5030-58-M, this accreditation process includes a system security analysis, which covers personnel, physical, COMSEC, TEMPEST, procedural or administrative, and hardware/software security, together with a test plan, test design, and the system security test and evaluation. These procedures must also be updated as necessary throughout the system life cycle for reaccreditation as needed. Therefore, an outline of expected accreditation procedures to be used for the I-S/A AMPE must be specified. This should include establishing accreditation criteria and the type of certification testing to be required.

2.6 PREPARATION OF INPUT

After the security factors have been supplied, they are then quantified as model input parameters. In cases where information cannot be specified, such as personnel characteristics, an estimate to represent the average will be used.

The quantification of information that is not directly available numerically represents the most difficult portion of this step. For example, a complexity factor for each alternative must be calculated for input to PRICE. This can only be accomplished by thoroughly analyzing the differences between development of the system incorporating the alternative and development of the system without the security requirements.

Table 2-1 lists the correspondence between input information required for the Secure Operating System alternative and the input parameters required by the PRICE-S model to determine the software development costs for a given alternative. Paragraph 2.7 gives an example of PRICE-S execution that also illustrates this correspondence. Appendix C provides a glossary of PRICE-S terms.

Before operating the models, the security factors must be clearly defined. This preparatory effort pertains to PRICE, PRICE-S, PRICE-SL, and SLIM.

Input parameters for SLIM can be calculated from the information needed for the PRICE models. Many of these parameters are identical. In this case, formatting the data to be accepted by SLIM is the only additional preparation required. The correspondence between SLIM and the PRICE software model parameters is given in Appendix C.

2.7 SAMPLE MODEL INPUT DATA ANALYSIS AND PREPARATION

2.7.1 Input Data Worksheets

The quantification of system development factors into model input parameters is needed to prepare for model execution. The information needed should be recorded on input data worksheets. This paragraph addresses the input data required by the software estimation models.

Figure 2-1 shows the PRICE-S Input Data Worksheet. The Project Title and Project Category entries are used in the report headings generated by the model. The basic input set, which must be specified, consists of the descriptors INST, APPL, RESO, UTIL, PLTFM, CPLX, and the Supplemental Information YEAR and MULT. All other inputs are optional and are used to refine or modify the basic set. The definition of all of these terms is given in Appendix C. A sample showing the use of this worksheet is given in Paragraph 2.7.2. The following briefly describes the basic input set:

Table 2-1. Correlation Summary of the Secure Operating System Alternative Required Information and PRICE-S Input Parameters

REQUIRED INFORMATION (PARAGRAPH 2.2)	PRICE-S INPUT PARAMETER (PARAGRAPH 2.7)
*1. Policy Model	Schedule (DSTART)
*2. Definition of secure software	APPL, INST, CPLX, NEWD, NEWC
*3. Level of assurance	Schedule, Program Constants
*4. Specification language	RESO, NEWD, NEWC, UTIL
5. Development phasing	Schedule, Program Constants
6. Complexity of secure software	Interface types, quantities, sizing data, CPLX, APPL
7. Development machine	UTIL, Interfaces
8. Reliability	PLTFM
9. Efficiency	PLTFM, CPLX
10. Documentation	Program Constants

*Information in these areas also affects analytical cost and schedule evaluation and is not totally addressed by the models.

PRICE S Input Data Worksheet

Filename: ____________
Page ______ of ______

Project Title									
Project Category									
Descriptors	INST	APPL	RESO	UTIL	PLTFM	CPLX	NEWD	NEWC	
Schedule	DSTART	DEND	ISTART	IEND	TSTART	TEND			
Resource Constraints	DCOST	DMAX	ICOST	IMAX	TCOST	TMAX			
Mix	MDAT	MONL	MREA	MINT	MMAT	MSTR	MOPR	MAPP8	APPL8
New Design	DDAT	DONL	DREA	DINT	DMAT	DSTR	DOPR	DAPP8	
New Code	CDAT	CONL	CREA	CINT	CMAT	CSTR	COPR	CAPP8	
Interface Types	TDAT	TONL	TREA	TINT					
Interface Quantities	QDAT	QONL	QREA	QINT					
Sizing Data	FUNCT	STRU	LEVEL	CAP	SOURCE	EXPAN			
Supplemental Information	YEAR	MULT	ESC	TARCST	INTEG				
Program Constants	GTABLE=								

Notes:

GC 1610 Rev 10/79

Note: Shaded Areas Indicate Optional Inputs Used To Refine Or Modify The Basic Input Set

RCA

Figure 2-1. PRICE-S Input Data Worksheet

1. INST: Number of machine instructions for the system.

2. APPL: Provides an instruction mix based on the type of application, given as a weighted value.

3. RESO: A relative measure of the skill level, productivity, efficiency, and labor rates during development. A default value (3.5) is currently used by the model, but can be adjusted during calibration.

4. UTIL: Represents the fraction of available hardware cycle time or total memory capacity used.

5. PLTFM: A relative measure of requirements of the operational environment.

6. CPLX: Measures development environment factors, such as personnel, product familiarity, and complicating factors. A standard value of 1.0 is adjusted during calibration.

7. YEAR: Base year for economics and technology growth.

8. MULT: Linear multiplier for all costs.

Figure 2-2 illustrates an Input Worksheet for SLIM. Descriptions of the data items are given below:

1. TITLE: self-explanatory

2. START DATE: Month and year estimated start time for project.

3. Cost Elements

   a. LABOR RATE: Average cost per man-year of effort.
   b. STDDEV: Standard deviation of labor rate
   c. INFLATION RATE: Self-explanatory.

4. Environment

   a. ONLINE: The proportion of development that will occur in online, interactive mode.

   b. DEV TIME: The proportion of the development computer that is dedicated to this system development effort.

TITLE					
START DATE	MONTH	YEAR			
COST ELEMENTS	LABOR RATE	STDDEV	INFLATION RATE		
ENVIRONMENT	ONLINE	DEV TIME	PROD TIME	HDL	LANGUAGE
SYSTEM	TYPE	LEVEL	UTILIZATION	REAL TIME CODE	
MAP	STRUC PROG	DES & CODE INSP	TOPDOWN DEV	CPT USAGE	
EXPERIENCE	OVERALL	SYSTEM TYPE	LANGUAGE	HARDWARE	
TECHNOLOGY	FACTOR				
SIZE	LOW	HIGH			
OR					
FUNCTIONS	NFUNCTION				
	NAME	LOWEST	MOST LIKELY	HIGHEST	
	NAME	LOWEST	MOST LIKELY	HIGHEST	

Figure 2-2. SLIM Input Worksheet

c. PROD TIME: The proportion of the available capacity of the development computer that is used for other production work.

d. HOL: The proportion of the system that will be coded in a high order language (HOL).

e. LANGUAGE: The primary language to be used in system development.

5. System:

a. TYPE: The type of system to be developed (such as, command and control)

b. LEVEL: The level of development required (is it a new system or a redesign, etc.).

c. UTILIZATION: The proportion of memory of the target machine that will be utilized by the software system.

d. REAL TIME CODE: The proportion of real time code to the total system.

6. Modern Programming Techniques (MPP): The degree of use of the following techniques.

a. STRUC PROG: Structured Programming.

b. DESIGN & CODE INSP: Design and code inspection.

d. CPT USAGE: Chief programmer teams.

7. Experience: Personnel experience that can impact the cost and time to do a project as related to the following areas:

a. OVERALL: Overall skill and qualifications.

b. SYSTEM TYPE: Past experience with system of similar size and application.

c. LANGUAGE: Past experience with programming language.

d. HARDWARE: Past experience with development computer.

8. TECHNOLOGY: The state of use of modern technology by the development organization (can be calibrated by system).

9. **SIZE/FUNCTIONS:** An estimate of system size either by total system or by functions.

2.7.2 PRICE Sample Executions

To provide an example of the development of model input parameters for a security alternative, a few sample runs were made applying the PRICE-S and PRICE-SL models to secure communications software. The purpose of these executions was to provide an increased understanding of the model operations. This sample is not intended to be a detailed analysis of this alternative.

A sample PRICE-S Input Data Worksheet, completed to reflect the parameters and values used in the sample run, is shown in Figure 2-3. The project title was "CSC Example" and a file created (CSC1) for future reference. The project category chosen was Secure Communications.

The Descriptors entry contains eight elements, which must all be specified.

The instruction (INST) count, which represents the size of the development effort, was, for this example, chosen to be 21,600, based on a real secure communications system. The APPL (application) value was chosen to be 7.5 indicating that interface and protocol requirements were considered, but timing constraints were not as stringent as for real-time applications. The default value for RESO was used. An estimate for UTIL of .5 indicates 50 percent utilization. The value of 1.7 for PLTFM indicates high reliability requirements. CPLX was initially set to .8 to indicate that the personnel, in this particular example, were among the best in the industry. NEWD and NEWC entries of 1.0 indicate a totally new design and implementation.

The next entry on the worksheet, Schedule, consists of three pairs of elements (start and end dates) for the overall development period: Design, Implementation, and Test and Integration. As shown on the worksheet, a development start date of January 1981 is provided; the model automatically calculated the remaining dates.

The Resource Constraints entry, similar to the Schedule entry, is paired off with respect to phases in the overall development effort. The first element in the pair represents the manpower cost for that phase or period.

PRICE S Input Data Worksheet

Filename: CSC1
Page ____ of ____

Project Title	CSC Example								
Project Category	Secure Communications								
Descriptors	INST 21600	APPL 7.5	RESO 3.5	UTIL .5	PLTFM 1.7	CPLX .8	NEWD 1.0	NEWC 1.0	
Schedule	DSTART 181	DEND	ISTART	IEND	TSTART	TEND			
Resource Constraints	DCOST 6000	DMAX 0	ICOST 6000	IMAX 0	TCOST 6000	TMAX 0			
Mix	MDAT 0	MONL	MREA	MINT	MMAT	MSTR	MOPR	MAPP8	APPL8
New Design	DDAT 0	DONL	DREA	DINT	DMAT	DSTR	DOPR	DAPP8	
New Code	CDAT 0	CONL	CREA	CINT	CMAT	CSTR	COPR	CAPP8	
Interface Types	TDAT 0	TONL	TREA	TINT					
Interface Quantities	QDAT 0	QONL	QREA	QINT					
Sizing Data	FUNCT 277	STRU	LEVEL 3	CAP	SOURCE	EXPAN			
Supplemental Information	YEAR 1981	MULT 1.0	ESC 0	TARCST	INTEG				
Program Constants	GTABLE=								

Notes.

GC 1610 Rev. 10/79

Figure 2-3. Simple PRICE-S Input

RCA

The value used in this case is $6000 per man-month. The second element in the pair represents maximum man-months to be expended during the respective phase. These elements were set to zero to permit the model to provide unconstrained resource estimates.

The Mix entry consists of the following elements: Data Storage/Retrieval (MDAT); On-line Communications (MONL); Real-Time Command/Control (MREA); Interactive Operations (MINT); Mathematical Operations (MMAT); String Manipulation (MSTR); Operating Systems (MOPR); and two optional elements (MAPP8, APPL8) used for special purposes. A single "0" entry in the first element, as in this sample, is used to signify unknown or not required conditions, and will effectively disable the remaining elements that follow the entry. This is an optional entry.

The New Design and New Code entries can be described in similar terms because of the close overlap of elements. In addition, there is a direct correspondence to the elements of the preceding entry, Mix. In fact, the elements for each of these entries (New Design and Code) are used to complement the corresponding element in the Mix entry. For instance, if the Operating System (MOPR) element in the Mix entry showed a .2 figure (for 20 percent), the Operating System (DOPR) element in the New Design entry would likely show a .8 figure (for 80 percent) to achieve the desired 100 percent distribution. Again, a zero value was used in the first element to disable the remaining elements because the values of 1.0 were used for NEWD, NEWC in the Descriptor line.

The Interface Types entry and the Interface Quantities entry also have similar elements. As their titles imply, each of these entries is used to quantify certain interface features included in the project. The Interface Types entry is used to measure the number of different interface types per category, such as unique disk and tape devices. The elements comprising the Interface Types entry include: Data Storage and Retrieval Devices (TDAT); On-line Communication Devides (TONL); Real-Time Command and Control Devices (TREA); and Interactive Communication Devices (TINT). The elements just described also pertain to the Interface Quantities entry elements, except that these elements are used to indicate the total number of devices existing for the type devices specified in the preceding entry. There were none for this sample case.

The Sizing Data entry elements are used for special case. The first three, function (FUNC), structure (STRU), and level (LEVEL), are normally used as alternate sources when the instruction (INST) element of the Descriptors entry is set 0. For this sample case however, respective figures are included as a fallback position to cross-check INST. The capacity (CAP) element is used in the special design-to-cost mode to answer "what-if" questions. The remaining elements, source (SOURCE) and expansion (EXPAN), are used together as an alternate approach for calculating the program size. These last three elements are not required by this sample case and are deliberately omitted.

The Supplemental Information entry is composed of several distinct elements. The year element establishes the economic/technological baseline reference points for the model; in this case it is 1981. The multiplier (MULT) element is a linear multiplier for all project costs (such as G&A and profit or fee); here, 1.0 is used to indicate a normal level. The escalation (ESC) element is used to indicate the inflation rate as an annual percentage; a 0 indicates no inflation. The target cost (TARCST) element is used only by the special calibration and Design-to-Cost modes, and is not applicable to this sample case. The last element, integration (INTEG), is applicable for such things as system integration and test file generation and verification and validation, which were not measured in this sample case.

The final entry, Program Constants, is used to set up a customized global table stored on file. This entry would be used, for example, to change the development phase proportions from the standard 40:20:40 (Design: Implementation: Test and Integration) ratio to something on the order of 40:10:50, which could be more appropriate for the special properties associated with projects involving formal specification and verification. In this sample execution, these constants were not changed.

The results of this sample run are given in Part III, Section 3 to illustrate the execution of the models.

SECTION 3 - EXECUTION OF MODELS

The second major process involved in the implementation of the methodology is the use of the automated models. This involves both calibration and execution of the models. These activities are described in this section.

3.1 CALIBRATION OF MODELS

Calibration of the models is necessary to establish a set of parameters that correspond to the anticipated development environment.

Calibration involves examining data from completed projects, estimating the differences between the development environment for those projects and the environment anticipated for the security alternatives, and deriving the parameters appropriate to the latter environment.

Proper calibration primarily affects the accuracy of the models, and secondarily affects their consistency for alternative comparison. Calibration will also provide a common basis for the execution of PRICE and SLIM.

The following calibration phases are required:

1. Resource Calibration - The resource value, in *PRICE terms*, *represents* the cost and performance efficiency of an organization. To calibrate the resource value, sample data derived from completed projects will be used when possible. The resulting resource values can be averaged and biased to develop a representative value for the anticipated development.

2. Application Calibration - The character of the software to be developed must be calibrated on the basis of the character of the software involved in completed projects for which data is known. The resulting application values must also be biased by the anticipated complexity involved in developing secure systems.

3. Phase and Cost Calibration - The proportions and respective cost multipliers for each of the development phases can also be calibrated from experience with similar projects.

4. **SLIM and PRICE Calibration** - **Results** of the two models must be calibrated so that comparable output is obtained. For example, typical output from PRICE-S is an optimal time schedule, while the SLIM output is expressed in terms of minimal schedule.

Once all the input data describing the project has been formulated and prepared, the initial procedure to execute of the model is to make the data accessible to the model. The user does this by keying data through the system editor into disk storage. The keying sequence directly corresponds to the sequence of elements on the input data worksheet. The operations used are shown in Figure 3-1.

In most cases, the model will be executed several times with different sets of parameter values. This is due to the nature of the estimation process. Each additional run of the model implies that the new parameter values are more accurate based on the increased knowledge of the user and maturity of the project. The user must be familiar with the intricacies of the project, and with the control features of the model as well. That is, with each calibration run of the model (assuming that parameter changes are to be made) the user must be aware of the relationships built into the model and how a change to one parameter may influence other parameters. Training and experience with the models will facilitate this process. All discrepancies between the estimates provided by PRICE-S, PRICE-SL and SLIM must be fully accounted. It is anticipated that most of the nontrivial discrepancies will be due to the differing degrees to which the two models can be calibrated. The analysis of discrepancies may, however, require some adjustment of parameters to ensure comparable results.

An example of the calibration process showing changes to two of the parameters is presented below.

```
ENTER CHANGES... PLTFM=1.2,RESO=2
FOLLOWING DATA CHANGES MADE:
PLTFM=1.2,RESO=2
```

LOGIN PLEASE	SYSTEM PROMPT TO LOG ON
ER! LLOOGGIINN PRICE2	USER RESPONSE
PRIMOS Version 17.3	SYSTEM ACCOUNTING INFORMATION
PRICE2 (4) LOGGED IN AT 11'50 112481	
ENTER REMOTE USER PASSWORD:	
INPUT	USER INITIATED COMMAND TO PREPARE FOR DATA INPUT
CSC EXAMPLE	DATA ENTRIES FROM INPUT DATA
SECURE COMMUNICATIONS	WORKSHEET
216000 7.5 3.5 .5 1.7 .8 1 1	
181	
6000 0 6000 0 6000 0	
0	
0	
0	
0	
0	
277 0 3	
22??1981 1	
EDIT	USER INITIATED COMMANDS TO
TOP	EDIT THE DATA INPUT FILE AND
P100	GENERATE FULL LISTING
.NULL.	LIST OF DATA FILE CONTENTS
CSC EXAMPLE	
SECURE COMMUNICATIONS	
21600 7.5 3.5 .5 1.7 .8 1 1	
181	
6000 0 6000 0 6000 0	
0	
0	
0	
0	
0	
277 0 3	
1981 1	
BOTTOM	USER INITIATED COMMAND TO DIRECT FILE POINTER TO END OF DATA FILE
FILE CSC1	GIVE THE FILE A NAME FOR FUTURE REFERENCE
OK, PRICE	INVOKE THE PRICE SYSTEM PROTOCOLS
PLEASE TYPE EXPLICIT PRICE NAME	TO START EXECUTION
OK, PRICE S3	
PRICE S3 READY	
INPUT FILENAME = CSC1	
OPTIONS :	SELECT THE OPTIONS DESIRED FOR CURRENT RUN; COMPLETE EXECUTION.
SHORT SENSIT SENSIA SCHED CURVE PRINTG PRINTP = 0 1 1	
RTABLE GTABLE UNITS OFILE POST = 0 0 1	
ALTERNATE COST UNITS = M	
SCALE = 1	
OKSKIP =	

Figure 3-1. PRICE-S System Input Operations

These changes, when compared with original values of PLTFM=1.7 and RESO=3.5, resulted in a reduction of manpower costs from 163 man-months to 41 man-months and a reduction in schedule completion time from 13 to 8 months.

One parameter that is critical to determing the overall development time and effort is the size parameter. This parameter, INST, which is the number of delivered executable machine instructions, can be changed to reflect a new order of magnitude, as given below:

ENTER CHANGES... INST=216000,RESO=3.5,CPLX=1

FOLLOWING DATA CHANGES MADE:

INST=216000,RESO=3.5,CPLX=1

This calibration run has also reset RESO to 3.5 and changed CPLX to 1.0 to indicate that the standard values are to be used.[4]

This type of parameter calibration can be done easily and shows the user the effect of modifications. A sufficient number of calibration runs must be made to ensure that the model executions will accurately reflect the system development. Note that only changed parameters need to be entered because a file (CSCI) has been established.

3.2 EXECUTION OF THE MODELS

Once calibration is completed, the models can be executed. The following paragraphs present an example of secure communications system development and describe the use of the PRICE models in estimating the cost and schedule of such an application. This discussion illustrates the general procedures involved in producing a PRICE run. Those procedures that must be considered from a user's point of view to exercise the model are described. To illustrate these procedures, a sample run using the PRICE-S (software development) and PRICE-SL (software life cycle) models is presented.

3.2.1 PRICE-S Outputs

There are several options that can be selected to suit the requirements of the current run, such as, which output reports are to be printed and whether they are to be fully or partially generated. This provides the cost analyst a method for controlling and facilitating the analysis effort by concentrating on specific component outputs.

The most frequently selected output of the model appears in Figure 3-2. This sample case uses the introductory information, originally provided as input by the user, modified by the above calibration examples.

The cost information that follows for the three development phases (Design, Implementation, Testing and Integration) is calculated by the model. These costs are also categorized by the different stages (such as systems engineering, programming) through which the project is expected to pass. For the next item, Schedule and Constraints, much of the information is provided by the model. Except for the "JAN 81" entry, all the remaining dates were computed by the model and are identified by an "*". No information was generated for the Application Categories item because of corresponding zero value input data entries.

The Sizing Data and Supplemental Information items are taken from the input data specified, except for the "*" entries that are computed by the model.

The last item in the report is a Gantt chart reflecting the start and end dates for the overall project effort, with a pictorial breakdown of the development phase ratios (which normally are measured as 40 percent design, 20 percent implementation and 40 percent testing and integration).

The model user can select other options that produce a more detailed output description. For instance, for important parameters, such as Resource (RESO) and Application (APPL), outputs can be generated that which assist in the sensitivity analysis of the estimation process. These outputs are presented in Figure 3-3 and Figure 3-4, respectively. In each case, for the values specified by the user (which are in positions near the center), the model computes deviations $\pm$ .1 in value of the specified value.

The final output included for PRICE-S, the Monthly Progress Summary, is shown in Figure 3-5. This output shows the percentage complete and percentage expended breakdowns for each of the three development phases from the date

--- PRICE SOFTWARE MODEL ---

DATE 24-NOV-81 | TIME 14:47 (181207) | FILENAME:CSC1

CSC EXAMPLE | SECURE COMMUNICATIONS

DESCRIPTORS					
INSTRUCTIONS	216000	APPLICATION	7.50	RESOURCE	[illegible]
UTILIZATION	0.50	PLATFORM	1.20	COMPLEXITY	[illegible]
NEW DESIGN	1.00			NEW CODE	[illegible]

COSTS IN DOLLARS 1000	DESIGN	IMPL	T & I	TOTAL
SYSTEMS ENGINEERING	1049.	[illegible]	[illegible]	[illegible]
PROGRAMMING	[illegible]	[illegible]	[illegible]	[illegible]
CONFIG CONTROL, Q/A	[illegible]	[illegible]	[illegible]	[illegible]
DOCUMENTATION	[illegible]	[illegible]	[illegible]	[illegible]
PROGRAM MANAGEMENT	[illegible]	[illegible]	[illegible]	[illegible]
TOTAL	[illegible]	[illegible]	[illegible]	[illegible]

SCHEDULE AND CONSTRAINTS	DESIGN	IMPL	T & I
START WORK	JAN 81	JUN 81*	[illegible]
END WORK	DEC 81*	MAY 82*	[illegible]
COST PER MAN-MONTH (1981 DOLLARS)	6000.00	6000.00	[illegible]
[illegible] MAN-MONTHS PER MONTH	[illegible]	[illegible]	[illegible]

APPLICATION CATEGORIES	MIX	NEW DEVELOPMENT DESIGN	CODE	HARDWARE INTERFACES TYPES	QUANTITY
DATA S/R	0.00	0.00	0.00	0	[illegible]
ONLINE COMM	0.00	0.00	0.00	0	[illegible]
REALTIME C&C	0.00	0.00	0.00	0	[illegible]
INTERACTIVE	0.00	0.00	0.00	0	[illegible]
MATHEMATICAL	0.00	0.00	0.00	***	***
STRING MANIP	0.00	0.00	0.00	***	***
OPR'G SYSTEMS	0.00	0.00	0.00	***	***

SIZING DATA					
FUNCTIONS	[illegible]	STRUCTURE	[illegible]	LEVEL	[illegible]
CAPACITY	[illegible]	SOURCE		[illegible]	[illegible]

SUPPLEMENTAL INFORMATION					
YEAR	1981	MULTIPLIER	1.000	ESCALATION	[illegible]
TARGET COST	0.	INTEGRATION	0.00	ESC EFFECT	[illegible]

SCHEDULE GRAPH

JAN 81

```
************ DESIGN ************
               *********** IMPLEMENT ***********
                          ***************** TEST & INTEGRATE *****************
```

Figure 3-2. PRICE-S Detailed Cost and Schedule Report

--- PRICE SOFTWARE MODEL ---

CSC EXAMPLE SECURE COMMUNICATIONS
COSTS IN MAN-MONTHS

SENSITIVITY DATA
(RESOURCE - COMPLEXITY)

RESOURCE		COMPLEXITY 0.900	COMPLEXITY 1.000	COMPLEXITY 1.100
3.400	COST	896.	986.	1079.
	MONTHS	24.5	27.6	30.7
3.500	COST	939.	1035.	1131.
	MONTHS	24.6	27.7	30.8
3.600	COST	984.	1084.	1185.
	MONTHS	24.7	27.8	31.0

Figure 3-3. PRICE-S Sensitivity Data Report for Resource and Complexity Parameters

--- PRICE SOFTWARE MODEL ---

CSC EXAMPLE
COSTS IN MAN-MONTHS

SECURE COMMUNICATIONS

SENSITIVITY DATA
(APPLICATION - INSTRUCTIONS)

INSTRUCTIONS

APPLICATION		[illegible]	[illegible]	[illegible]
7.400	COST	920.	[illegible]	[illegible]
	MONTHS	26.5	27.6	28.6
[illegible]	COST	[illegible]	1035.	1137.
	MONTHS	[illegible]	27.7	28.7
7.600	COST	944.	1048.	1153.
	MONTHS	26.8	27.8	28.8

Figure 3-4. PRICE-S Sensitivity Data Report for Application and Instructions Parameters

--- PRICE SOFTWARE MODEL ---

CSC EXAMPLE
COSTS IN MAN-MONTHS

SECURE COMMUNICATIONS

MONTHLY PROGRESS

	% COMPLETE			COST		% EXPENDED	
MONTH	DESIGN	IMPL	T & I	THIS MONTH	TOTAL	THIS MONTH	TOTAL
JAN 81	3.8	0.0	0.0	16.5	16.5	1.6	1.6
FEB 81	13.3	0.0	0.0	40.8	57.3	3.9	5.5
MAR 81	25.9	0.0	0.0	54.3	111.6	5.2	10.8
APR 81	39.7	0.0	0.0	59.5	171.1	5.8	16.5
MAY 81	53.4	0.0	0.0	58.8	229.9	5.7	22.2
JUN 81	65.9	1.4	0.0	57.0	286.9	5.5	27.7
JUL 81	76.8	5.9	0.0	55.7	342.6	5.4	33.1
AUG 81	85.6	13.3	0.0	53.0	395.6	5.1	38.2
SEP 81	92.3	23.2	0.2	49.8	445.4	4.8	43.0
OCT 81	97.0	35.1	1.1	47.5	492.9	4.6	47.6
NOV 81	99.5	48.0	2.8	43.7	536.5	4.2	51.9
DEC 81	100.0	61.1	5.2	38.4	574.9	3.7	55.6
JAN 82	100.0	73.4	8.4	37.6	612.5	3.6	59.2
FEB 82	100.0	84.0	12.3	37.4	649.9	3.6	62.8
MAR 82	100.0	92.2	17.1	35.7	685.6	3.4	66.3
APR 82	100.0	97.5	22.6	32.9	718.5	3.2	69.4
MAY 82	100.0	99.9	28.8	29.8	748.3	2.9	72.3
JUN 82	100.0	100.0	35.8	28.1	776.5	2.7	75.0
JUL 82	100.0	100.0	43.3	30.3	806.8	2.9	78.0
AUG 82	100.0	100.0	51.3	32.2	838.9	3.1	81.1
SEP 82	100.0	100.0	59.6	33.4	872.4	3.2	84.3
OCT 82	100.0	100.0	68.0	33.8	906.2	3.3	87.6
NOV 82	100.0	100.0	76.2	33.0	939.2	3.2	90.8
DEC 82	100.0	100.0	83.9	30.9	970.1	3.0	93.8
JAN 83	100.0	100.0	90.7	27.1	997.2	2.6	96.4
FEB 83	100.0	100.0	96.0	21.3	1018.5	2.1	98.4
MAR 83	100.0	100.0	99.2	13.2	1031.7	1.3	99.7
APR 83	100.0	100.0	100.0	3.0	1034.7	0.3	100.0

* LEAD =	0.60	0.20	0.15	FOR PROFILE GRAPHS
* LAG =	0.15	0.20	0.60	RESPOND OK = 7
* PEAK AV =	1.63	1.62	1.63	

END OF FILE - OK =

Figure 3-5. Monthly Progress Summary

work started in the design phase through the date work is scheduled to end for the test and integration phase.

3.2.2 PRICE-SL Outputs

The second half of the sample execution used PRICE-SL, the software life cycle model. This model continues where PRICE-S stops.

Key information for input to PRICE-SL was obtained from the PRICE-S execution. The input procedures would be basically the same as shown earlier in the first sample case, although in this CSCl example, the parameters maintain their original values as input from the PRICE-S data worksheet. The key values generated by PRICE-S and used by PRICE-SL are the total cost ($6208K) and the scheduled completion date (APR 83).

PRICE-SL also has an input data worksheet to facilitate assignment of model parameter values. This worksheet is presented in Figure 3-6. The Development Descriptor elements, instructions (INST), application (APPL), resource (RESO), utilization (UTIL), platform (PLTFM), complexity (CPLX), new design (NEWD), and new code (NEWC), correspond directly to elements found on the PRICE-S input data worksheet. The Development Economics element - reference year (YEAR), multiplier (MULT), annual escalation (ESC), design start date (DSTART), testing & integration date (TEND), and cost unit scaling (SCALE) are also derived from various entries on the PRICE-S worksheet. Total development cost (DEVCST) is an output value of the PRICE-S run.

The remainder of the worksheet deals with the support aspects of the project or new data that must be determined. Two divisions, economics and environment, comprise the support criteria. For the economics division, there are six elements:

1. Reference year (SYEAR), which represents the baseline date for comparison
2. Multiplier (SMULT), which is used to markup support costs (that is, 1.35 represents a 35 percent markup)
3. Annual escalation (SESC), which indicates the inflation rate

PRICE Software Life Cycle Model Input Data Worksheet

Filename CSC1

Page ____ of ____

Development Title	Secure Comm Support			
Development Descriptors	Instructions **INST** 216000	Application **APPL** 7.5	Resource **RESO** 3.5	Utilization **UTIL** .5
	Platform **PLTFM** 1.2	Complexity **CPLX** 1	New Design **NEWD** 1	New Code **NEWC** 1
Development Economics	Reference Year **YEAR** 1981	Multiplier **MULT** 1	Annual Escalation **ESC** 0	
	Design Start Date **DSTART** 181	Test & Integ. End Date **TEND** 483	Cost Unit Scaling **SCALE** 1000	Total Dev Cost **DEVCST** 6208
Support Title	Support			
Support Economics	Reference Year **SYEAR** 1981	Multiplier **SMULT** 1	Annual Escalation **SESC** 0	
	Support Start Date **SSTART** 483	Support End Date **SEND** 488	Cost Unit Scaling **SSCALE** 1000	
Support Environment	Number of Installations **INSTAL** 20	Employment Per Installation **EMPLOY** 1	Anticipated Growth **GROWTH** .1	

Notes:

4/80

RCA

Figure 3-6. PRICE-SL Input Data Worksheet

4. Support start date (SSTART), which indicates the actual date support is to start for the project

5. Support end date (SEND), which indicates the date support is to end on the project

6. The cost unit scaling (SSCALE), which indicates the scale the report cost values are to be multiplied by to reach the final precise value.

The environment division is composed of three elements:

1. Number of Installations (INSTAL), which enumerates the total deployment sites to be considered

2. Employment Per Installation (EMPLOY), which indicates the support level

3. Anticipated Growth (GROWTH), which indicates the level of growth anticipated project manager during the support phase.

PRICE-SL uses the same approach for providing the data compiled on the worksheet to the model as does PRICE-S. The system prompts the user for each line of appropriate input data. This operation is shown in Figure 3-7 below. Each line of input directly corresponds to each division on the worksheet.

A typical report, reflecting the parameter inputs just specified, is shown in Figure 3-8. After the report title information is indicated the operational life of the project. This figure ("5-year") is calculated by the model from the start and end dates specified. The costs are divided into stages (such as systems engineering, programming) identical to the PRICE-S report preceding this run; these costs are additionally distributed over the support phases, including maintenance, enhancement, and growth. All output information that follows, through the support environment item, is derived from user input. The support resources item elements, however, are calculated by the model. The last item of the report, cost summary, provides both development cost, supplied as input to PRICE-SL, and the support costs calculated.

```
OK, PRICE-SL

* * * PRICE-SL READY * * *

INPUT FILENAME =

OPTIONS:
    SHORT TABLES  =
    RTABLE ATABLE OFILE  =

DEVELOPMENT TITLE               =SECURE COMM SUPPORT
DEVELOPMENT DESCRIPTORS(1)      =216000 7.5 3.5 .5
DEVELOPMENT DESCRIPTORS (2)     =1.2 1 1 1
DEVELOPMENT ECONOMICS(1)        =1981 1 0
DEVELOPMENT ECONOMICS(2)        =181 488 1000 6208
SUPPORT TITLE                   =SUPPORT
SUPPORT ECONOMICS(1)            =1981 1 0
SUPPORT ECONOMICS(2)            =483 488 1000
SUPPORT ENVIRONMENT             =20 1 .1

DATA CHECK  - NEXT ACTION  = R
```

Figure 3-7. PRICE-SL System Input Operations

--- PRICE SOFTWARE LIFE CYCLE COST MODEL ---

DATE 24-NOV-81 TIME 14:52 FILENAME NOT USED
(1812111)

SECURE COMM SUPPORT [illegible]

5 YEAR OPERATIONAL LIFE

COSTS IN DOLLARS 1000	MAINTAIN	ENHANCE	GROW	TOTAL
SYSTEMS ENGINEERING	1441.	180.	[illegible]	[illegible]
PROGRAMMING	721.	109.	[illegible]	[illegible]
CONFIG CONTROL, Q/A	990.	72.	[illegible]	[illegible]
DOCUMENTATION	302.	33.	[illegible]	[illegible]
PROGRAM MANAGEMENT	206.	26.	[illegible]	[illegible]
TOTAL	3749.	421.	810.	[illegible]

DEVELOPMENT DESCRIPTORS

INSTRUCTIONS	216000	APPLICATION	7.50	RESOURCE	[illegible]
UTILIZATION	0.50	PLATFORM	1.20	COMPLEXITY	1.00
NEW DESIGN	1.00			NEW CODE	[illegible]

DEVELOPMENT ECONOMICS

YEAR	1981	MULTIPLIER	1.00	ESCALATION	0.000
DESIGN START	JAN 81	TEST END	APR 83		

SUPPORT ECONOMICS

YEAR	1981	MULTIPLIER	1.00	ESCALATION	0.000
SUPPORT START	APR 83	SUPPORT END	APR 88	ESC EFFECT	1.000

SUPPORT ENVIRONMENT

INSTALLATIONS	20	EMPLOYMENT	1.00	GROWTH	[illegible]

SUPPORT RESOURCES

MAINTENANCE	1.50	ENHANCEMENT	[illegible]	GROWTH	[illegible]

COST SUMMARY

DEVELOPMENT COST IN 1981 DOLLARS 1000 ... [illegible]
SUPPORT COST IN 1981 DOLLARS 1000 .... [illegible]

Figure 3-8. PRICE-SL Detailed Cost Report

SECTION 4 - INTEGRATION AND ANALYSIS

The final process in the application of the recommended methodology is the integration and analysis of the information acquired during the processes described above.

The model output information is integrated with the additional cost factors identified in Paragraph 2.5 to provide a comprehensive cost and schedule for development of the proposed I-S/A AMPE alternative. This integration and analysis process is described below in the context of an operational approach for the methodology. This operational approach, as described in Part II, Section 1, involves both preliminary and detailed analysis. The preliminary analysis involves alternative analysis, translation of factors into model parameters, model calibration, and desired output determination. The detailed integration and analysis process includes analysis of both SLIM and PRICE outputs for compatibility, inclusion of security factors not handled by the models, and coordination of schedule outputs.

Each stage of the analysis is supported by standard worksheets, as described in the following paragraphs.

Two additional factors should also be considered in the application of this methodology. First, the integration process could be automated to provide for a single input step that would invoke each of the models in turn and then generate a single output as described in Part II, Section 5. This would provide for ease of use and minimize training requirements. Second, the integration of a family of cost and schedule models gives the user the capability to use the methodology in a very general manner. The capability is provided to estimate cost and schedule effects for the three proposed security options or other, as yet unspecified, security options.

4.1 COST AND SCHEDULE CONTROL WORKSHEET

The cost and schedule control worksheet will assist the analyst in the controlled management of establishing a uniform cost and schedule estimation strategy for each alternative. This strategy simplifies the initial process for specifying tools or techniques needed to achieve a given objective, whether in the form of a decomposed element or component of an alternative, or the entire alternative itself. In essence, the worksheet previews all considerations that must be weighed while attempting to highlight other less obvious decisions supported by the overall methodology. It also serves as a control sheet for all support documents used in the preliminary analysis stage of the methodology.

A sample worksheet showing the various control criteria is illustrated in Figure 4-1. A series of administrative control entries introduces the worksheet. The first significant entry in this section is the Run Control Number (which follows the Date entry). This control number consists of multiple elements or keys that are used to identify a particular run. These are the alternative (A1 = Secure Operating System; A2 = Hardware Separation; A3 = End-to-End Encryption), the component (C1 - software for Secure Operating System, for instance), and the iteration (I1 - represents first iteration of possibly multiple runs to uniquely identify a given run in addition to the alternative and the component identifiers). Following the Run Control Number is the Analyst Name(s) entry, identifying the person who is preparing the worksheet. Next is the entry that delineates the full identification of the security alternative addressed. The final entry of this part of the worksheet, Objective, is included as an open-ended entry of several blank lines to give the analyst freedom to identify and describe how this particular cost and schedule run will assist a particular problem, as well as prescribe the next set of directives to the overall estimating approach to the alternative. Next is a broad list of different security factors (as described in Part II, Section 2 of this report) that should be considered in the methodology as additional elements that affect cost or schedule. This arrangement makes these important factors conspicuous to the analyst at an early stage to determine which factors apply to the particular alternative and what technique or techniques should be used for quantifying these factors for later integration with the model results.

I-S/A AMPE
Cost & Schedule Analysis Control Worksheet

Date (mm/dd/yy): 11/24/81

Run Control No.: A1-C1-I1

Analyst Name(s): D. Long

Security Alternative Ident: Secure Operating System

Component Ident: Comm. Software

Objective of Analysis (what, how)

Calibration of model based on historical data.

Security Issues To Be Considered:

__ 1.	Security Policy Model	__ 10.	Documentation
X 2.	Definition of Secure Software	__ 11.	AMPSSO Assignment
__ 3.	Level of Assurance	__ 12.	Personnel Clearances
X 4.	Specification Language	__ 13.	Physical Security
__ 5.	Development Phasing	__ 14.	COMSEC
X 6.	Complexity	__ 15.	TEMPEST
__ 7.	Development Machine	__ 16.	Information Sanitization
__ 8.	Reliability	__ 17.	Accreditation
__ 9.	Efficiency		

Models To Be Executed:

Price S X Price SL X Price __ Price L __ SLIM __

Model Inputs (attach corresponding Input Data Worksheets):

Input Data Worksheet Control No.: ____

Original Input Data Worksheet Control No.: 1

Prior-Run Key Sensitivity Factors and Values:

1. PLTFM = 1.7 2. CPLX = .8 3. INST = 21600

4. 5. 6.

Data Validation (full printout of Model Inputs data file)

Calibration (attach operations printout of changes or indicate changes)

Figure 4-1. Cost and Schedule Control Worksheet (1 of 2)

Changes: 1. PLTFM = 1.2   2. CPLX = 1.0   3. INST = 216000

4. ______   5. ______   6. ______

Expected results of these changes: ______________________

____

____

Data Validation (full or partial - showing changes only - printout of calibration operations)

Model Outputs Desired:

Price S

X 1. Cost & Sched. Det. Rpt. -Partial, Full

X 2. RESO, CPLX Sens. Data Rpt.

X 3. APPL, INST Sens. Data Rpt.

X 4. Monthly Progress Summary

__ 5. Schedule Effect Summary

Price SL

X 1. Cost Detail Rpt.-Partial, Full

__ 2.

PRICE

__ 1. Cost & Sched. Det. Rpt.-Partial, Full

__ 2.

PRICE L

__ 1. Cost Detail Rpt.-Partial Full,

__ 2.

SLIM

__ 1. Simulation

__ 2. Manloading

__ 3. Cashflow

__ 4. Code Production

__ 5. Life Cycle

__ 6. Milestones

__ 7. Front End

__ 8. Risk Analysis

__ 9. Pert Sizing

__ 10. Documentation

__ 11. Benefit Analysis

__ 12. CPU Usage

__ 13. Linear Program

__ 14. Interactive Linear Prog.

__ 15. Design-to-risk

__ 16. Design-to-cost

__ 17. Design-to-Schedule

__ 18. Best Bid

Other Considerations:

1. ______________________________________

2. ______________________________________

3. ______________________________________

Comments: Development Phasing not changed - suggest next run to modify it to [illegible] ratio.
Additional Security factors not included in this run.

Figure 4-1. Cost and Schedule Control Worksheet (2 of 2)

The next section to the worksheet indicates which models must be invoked to generate cost and schedule outputs from explicitly specified parameter inputs.

NOTE: Although it is not required that PRICE and SLIM be independently processed (that is, that only PRICE or only SLIM be planned per session), it is suggested that this procedure be followed for simplicity and separation of accounting.

The fourth section of the worksheet provides a tracking mechanism from input to final, concrete figures. Highlighted in this section are those parameters (such as RESO and APPL) to which current calibration values can be compared without having to search through a host of other runs to find the appropriate figures.

The fifth section of the worksheet serves as a reminder to the analyst to attach to this worksheet a printout of the model input parameter file. This is a control mechanism to eliminate procedural problems and extra runs.

The sixth section of the worksheet deals with calibration of model input parameters. The purpose of this procedure is to gear the control mechanisms of the model as precisely as possible to the particular application being estimated. This will produce consistently accurate and reliable results. This section of the worksheet reflects the current parameter changes (which can be easily contrasted with previous settings in the model input section) in written form, or as supported by an attached printout of corresponding parameters. The second half of the calibration section is open-ended to permit the analyst to indicate what the expected results should be, either in general or specific terms. This description may compare against previous run parameters, or may be independent and generalize what the results will be.

Section seven of the worksheet is identical in function to section four, except that this section has a narrower window; it focuses on and validates only those parameters that were changed in the calibration section.

Section eight is a composite of the assorted reports that can be generated by the corresponding model. This is a planning and management mechanism for identifying only pertinent output reports.

Section nine is devoted to other considerations that might relate to this operational level of planning, such as the first date the model or models will be available (because of user scheduling constraints) and the resources for invoking the models.

The final section is reserved for pertinent comments at the level inferred by previous sections to the worksheet.

4.2 INTEGRATION AND ANALYSIS DETAIL WORKSHEET

The purpose of the Integration and Analysis Detail Worksheet is to provide the analyst with a systematic approach in the detailed evaluation of cost and schedule considerations. It consolidates, integrates, and summarizes the cost and schedule figures derived from the different estimation sources used.

A sample Integration and Analysis Detail Worksheet is illustrated in Figure 4-2.

This worksheet is divided into two separate major sections: Costs and Schedule. The analyst can focus attention on either during an analysis session. Where repetitive processing may be concerned, this division also simplifies analysis procedures.

The introduction to this worksheet, from the Date entry through the Component Ident entry, is identical in format and function to the Cost and Schedule Control Worksheet. Following the Component Ident is a user selection table for identifying the particular model or models used in deriving the cost and schedule figures for this session.

Following the model selection table is the costs section. The three categories under which the costs are allocated include Software, Hardware, and Security-specific. For software, the costs are apportioned according to development and support total life-cycle functions. The development cost figure is the total cost found either in the PRICE-S Detailed Cost and Schedule Report or in the summary section of the PRICE-SL Detailed Cost Report. When using SLIM, this figure is the total cumulative cost from the Cashflow Plan table that was generated. The second entry to the software

AD-A113 423   COMPUTER SCIENCES CORP   FALLS CHURCH VA   F/6 5/1

RECOMMENDED METHODOLOGY FOR INTER-SERVICE/AGENCY AUTOMATED MESS--ETC(U)

FEB 82   F23613-77-D-0011

UNCLASSIFIED   NL

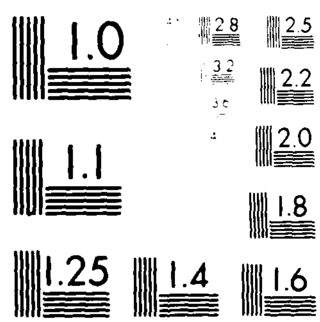
2 OF 2

END

DATE FILMED

04-82

DTIC


1.0
2.8
2.5
3.2
2.2
3.6
2.0
1.1
1.8
1.25
1.4
1.6

I-S/A AMPE
Integration & Analysis Detail Worksheet
Part 1 - Costs

Date (mm/dd/yy): 11/24/81

Run Control No.: A]-C]-I]

Analyst Name(s): D. Long

Security Alternative Ident: Secure Operating System

Component Ident: Comm. Software

Model or Models Employed This Session:

Division \ Phase	Development	Support
Software Hardware	Price S X SLIM __ Price only __	Price SL X SLIM __ Price L only __

COSTS		Units	Subtotal	Total
Software:				
	Development --	6208		
	Support --	5079		
			11287	
Hardware:				
	Development --			
	Engineering			
	Manufacturing			
			N/A	
	Support --			
	Equipment			
	Support Equip.			
	Supply			
	Supply Admin.			
	Manpower			
	Contactor Support			
	Other			
			N/A	
	Additional			
	Energy			
	Training			
	Other			
			N/A	
				11287

Figure 4-2. Sample Integration and Analysis Detail Worksheet (1 of 3)

I-S/A AMPE
Integration & Analysis Detail Worksheet
Part 1 - Costs (Cont'd)

Security Specific:

A. Secure Operating System

AMPSSO Assignment	X
Personnel Clearances	X
Physical Security	X
COMSEC	X
TEMPEST	X
Sanitization of Info	X
Accreditation	X
Other	X

N/A

B. Hardware Separation

AMPSSO Assignment	X
Personnel Clearances	X
Physical Security	X
COMSEC	X
TEMPEST	X
Sanitization of Info	X
Accreditation	X
Other	X

N/A

C. End to End Encryption

AMPSSO Assignment	X
Personnel Clearances	X
Physical Security	X
COMSEC	X
TEMPEST	X
Sanitization of Info	X
Accreditation	X
Other	X

N/A

Figure 4-2. Sample Integration and Analysis Detail Worksheet (2 of 3)

I-S/A AMPE
Integration & Analysis Detail Worksheet
Part 2 - Schedule

Date (mm/dd/yy): 11/24/81

Run Control No.: A_C_I_

Analyst Name(s): D. Long

Security Alternative Ident: Secure Operating System

Component Ident: Comm. Software

Model or Models Employed This Session:

Division \ Phase	Development	Support
Software	Price S X Price A ___ SLIM ___	Price SL X SLIM ___
Hardware	Price only ___ Price A ___	Price L ___ Price A ___

SCHEDULES

		Start Date	End Date
Software:			
	Development --	081	0483
	Design	0181	1281
	Implementation	0681	0582
	Test & Integ.	10981	0483
	Support --	0483	0488

		Start	1st Item	Finish
Hardware:				
	Development --			
	Development	N/A	N/A	N/A
	Production			
	Support --	N/A	N/A	N/A

Note: Separate or combined activity profiles can be generated by Price A (Activity Distribution Model) from the schedule information shown above.

Comments: ______________________________

Figure 4-2. Sample Integration and Analysis Detail Worksheet (3 of 3)

category is support. This is the total cost from the PRICE-SL Detailed Cost Report; with SLIM the total cost comes from the Life-Cycle Cashflow, Cumulative Cost table.

The second category under costs is Hardware. Again, this category is divided into development and support costs. Development costs for hardware are engineering costs and manufacturing costs. These costs are derived from the PRICE Detailed Cost and Schedule Report only. Included under support costs are such things as equipment, support equipment, and supply. These costs are derived from the PRICE-L Detailed Cost Report only.

The third and final category under costs is security-specific cost. These costs are not directly measurable by PRICE and SLIM models and are unique to each security alternative. Estimates for these costs are obtained through interaction with the agencies specifying the security alternatives.

Once all associated costs have been totaled (shown as a subtotal for each alternative), these software life-cycle cost figures can be compared between PRICE and SLIM. By the design of this worksheet, costs can be summarized for all three alternatives on a single worksheet, or for a single alternative on each of three separate worksheets, depending on reporting requirements.

The Schedule section to the Integration and Analysis Detail Worksheet is aligned in similar fashion, as is the Costs section, that is, by software and hardware categories broken down into development and support phases. Under development in the software category, the three development phases -- Design, Implementation, and Testing and Integration -- are delineated according to start and end dates. These dates are derived directly from the Detailed Cost and Schedule Report for PRICE-S and indirectly from the Milestones chart for SLIM. The support phase for Software is similarly arranged by start and end dates. These dates are derived directly from the Detailed Cost Report for PRICE-SL and indirectly from the Life Cycle Manpower, Effort table for SLIM. Under Hardware for Development, the breakdown is according to development and production phases. For each of these phases, the dates provided are the start date, the expected date for the first item, and the finish date. These dates are derived directly from the Detailed Cost and Schedule Report for PRICE;

SLIM is not used in this case because it deals only with software-related estimates. No dates need to be included under Hardware support, an axiom of hardware life cycle estimation. Finally, the analyst is advised that another facility exists -- in the form of the PRICE A Activity Distribution model -- allowing an individual or a totally combined activity profile schedule to be generated.

APPENDIX A

SECURITY ALTERNATIVES

APPENDIX A - SECURITY ALTERNATIVES

Certifiable multilevel security (MIS) may be the most risky area of the I-S/A AMPE design. Certifiable MLS will allow users of different security levels and access constraints to use the same system simultaneously. For example, a user with TOP SECRET files can use the same system as an uncleared user, with no concern that information will be compromised. Need-to-know criteria are also a provision of MLS systems.

Some MLS requirements are:

1. Physical security to segregate the computer and storage from the outside world
2. Administrative security to control access to secure computer facilities
3. Network security to protect information passing over communication links
4. Operating system security to control users with different clearances and need-to-know access to the system.

The first three requirements have been met with system-high secure systems, but presently the fourth has not been certified.

This appendix describes several security alternatives that might be proposed for the I-S/A AMPE program. These are secure operating system hardware separation and end-to-end encryption.

For each of these alternatives, scenarios are described to identify the security features involved and define security factors that affect cost and schedule. The descriptions given here are not intended to describe the designs expected for the I-S/A AMPE, but rather to illustrate the application of the methodology to each alternative. Operational considerations, such as efficiency, are not addressed. Information gained in this analysis provides a usuable basis for the automated models.

A.1 SECURE OPERATING SYSTEM

The development of a secure operating system is one alternative for providing security in the I-S/A AMPE. The requirements for a secure operating system must be incorporated into the overall development methodology required for the system. Characteristics and features unique to the development of secure software are detailed below. The requirements include system development techniques, enforcement of security policy by hardware and software mechanisms, and establishment of the appropriate level of assurance.

A.1.1 Secure System Development

Although there are many variations among the approaches to system software development, there is also substantial commonality. This commonality can be described in terms of the phases of the software development cycle: statement of operational requirements, system requirements specification, design of the system, and implementation of the system.

The approach to developing secure software is to formalize this process. This includes controlling the system evolution by expanding each level of development to allow formal verification that each level has met specifications of the previous level. The application of automated tools, whenever available, facilitates this process. The following paragraphs describe this formal approach.

First, a formal security implementation directive is produced stating the general security requirements expected of the system. This directive is the accepted standard, based on the system security policy. DoD Directive 5200.28 is the accepted policy for DoD security systems.

A formal security model formalizes the directive in precise, unambiguous terminology and forms the basis for subsequent rigorous analysis. There are currently no formal techniques for verifying that a security model conforms to a directive, so the verification, at this level, is performed by review and arbitration by designers and Government representatives.

A formal specification is a rigorous, unambiguous statement about the required behavior of the security-relevant parts of the system. These specifications are stated in some formal specification language, such as SPECIAL, Ina Jo, or Gypsy. The formal top-level specifications, namely, the specification of the security-relevant portions of the system at its points of interface to its external environment, are verified to satisfy the security model. This is referred to as specification verification.

These formal top-level specifications are then expanded to incorporate design decisions made during the software design process. These more detailed formal specifications are verified against the top-level specifications. This process is referred to as design verification.

The code for the security-relevant parts of the system is written in a high-level language, such as JOVIAL, C, MODULA, EUCLID, Gypsy, or Ada. The next step in the verification is to ensure that the high-level language program meets its formal specifications. This is called code verification.

The final step is to translate the high-level language program into a low-level language that will run on the chosen hardware. The low-level program should be verified to be equivalent to the high-level language program. The problem can be addressed by program equivalence techniques such as those used to validate compilers.

At this point, the verification chain is complete. It could be concluded that the secure software will operate in conformity with the original directive.

A.1.2 Security Policy Enforcement

The most important requirement for the security-related software is that of enforcing the DoD security policy. Three factors of the approach to satisfy this requirement are:

1. A clear and consistent statement of security policy that can be supported by software.
2. Mechanisms, both hardware and software, that explicitly enforce the policy. These are called protection mechanisms.

3. Specific techniques, tools, procedures, and disciplines that provide a satisfactory level of assurance that the mechanisms do, in fact, enforce the policy.

The demonstration of the first factor is a rigorous mathematical model that reflects the required security policy and is used as a guide in the design and implementation of the security controls.

Demonstration of the second factor will include a demonstration that protection mechanisms are present. This demonstration will establish that:

1. The mechanisms that enforce the policy are clearly identified
2. The protection mechanisms are isolated from the remaining software for self-protection
3. All accesses to information objects are mediated
4. The security software will support a soft crash, namely, that the system will be able to restart at some checkpoint location with data in a consistent state in the face of hardware error
5. All identified communication channels are audited
6. The denial of service problem has been a primary design consideration.

Satisfaction of the third factor should include a verification plan for demonstrating that the access controls and mechanisms function correctly. The plan demonstrates:

1. That attention has been paid to protection during system design
2. That extensive testing of the security controls is provided
3. That penetration testing is included
4. That top-level specifications describing the external interface of the security mechanisms to the rest of the system are given in a formal specification language
5. That the use of the formal specification language will allow the potential for formally verifying that the security policy model is transferred correctly into the top-level specifications

6. That the formal specifications will allow the potential for formally verifying that the security control design corresponds to the Security Policy Model

7. That configuration management of the security software has been incorporated into the verification plan.

A.1.3 Levels of Assurance

The cost and schedule impact of this alternative is dependent on the level of assurance required and proposed. Seven levels of assurance have been defined by Nibaldi [3] of MITRE Corporation. Each of these levels, as outlined below, identifies an increased level of internal protection:

1. Level 0: Mistake Protection. The Level 0 system is primarily designed to protect against inadvertent compromise, rather than a determined subverter. There is little attempt to control users.

2. Level 1: Discretionary Controlled Sharing. Discretionary protection is supported and system integrity is promoted in an attempt to prevent users from interfering with the operating system or with each other. There is no formal attempt to validate the protection mechanisms.

3. Level 2: Mandatory Controlled Sharing. Mandatory as well as discretionary security is included at this level. Attention is given to denial of service and protection-related events are audited. Extensive testing is relied on for assurance.

4. Level 3: Isolated Protection Mechanism. The protection mechanisms are identified, isolated, and made independent of other software, allowing for ease of verification and analysis. The software implementing security controls must be developed using a methodological approach. Testing is the primary means of assurance.

5. Level 4: Design Verification. The validity of the design with respect to a security model must be proven. This involves the formal expression of both the security policy and the design to facilitate the verification.

6. Level 5: Source Code Verification. Design-to-model and code-to-design proofs must be completed satisfactorily for a system to reach this level. Denial of service is addressed carefully.

7. Level 6: Object Code Verification. To attain this level, the object code must be proven to correspond with the security policy model.

The proposed development of a secure system can be evaluated in terms of these criteria. The requirement for formal specification and verification techniques will have an effect on the relative cost and schedule phasing of the software system development and must be incorporated in the application of the methodology. For example, if automated tools are used in the verification process, the additional cost due to increased computer usage and possible schedule impact become factors.

Within this context of secure system development, the options for the level of assurance are incorporated easily to provide for various secure software development scenarios. The required level of assurance, for example, design verification, can be established.

Another option which could lead to different system costs and schedule is the requirement for a formal mathematical model of the required security policy. A mathematical model may need to be developed to reflect an informational flow policy in the I-S/A AMPE; this will affect both costing and scheduling parameters.

An additional factor to be considered in the development of secure software is the distribution of formal specification efforts. It has been demonstrated that one company can produce traditional specifications while a second company produces the formal specifications. This could have major cost effect on the I-S/A AMPE program although the schedule effect would be minimal.

The number of possible scenarios for the development of a secure operating system is dependent on decisions in each of the above areas. For each of these possibilities, security-relevant factors can be identified and quantified.

A.2 HARDWARE SEPARATION

This section describes three methods for ensuring separation of information of different security classification levels within the I-S/A AMPE processor by hardware. Illustrated is separation at two levels: General Service (GENSER) and Defense Special Security Communications System (DSSCS) message traffic, although these scenarios also apply to multiple levels within GENSER and DSSCS.

A.2.1 Separate Processors

Figure A-1 illustrates the first alternative. This requires separate, identical, dedicated processors for each level. The processors and their associated workstations are in separate rooms or otherwise separated by physical security methods. The systems are more or less redundant copies of each other. The main advantage of this approach is that no verified or trusted software is required within the processors. The MLS capability is provided through physical separation.

The major effect in this scenario is the cost of duplicate hardware. The effect of this factor is dependent on the processing power needed for the I-S/A AMPE and the availability of appropriate hardware.

A.2.2 Dedicated Switching Architecture

The second alternative, shown in Figure A-2, uses a single I-S/A AMPE processor, that is dedicated to one level of traffic at any one time. A secure, manually operated switch is set to route DSSCS message traffic through the crypto-device (KG) when I-S/A AMPE is dedicated to DSSCS processing. This switch would require deliberate action to alter its setting. The MLS capability is provided by separating processing time.

The processor and its workstations are sanitized before initiating a work session at another level. Physical security is responsible for ensuring that access to I-S/A AMPE is in accordance with the classification level of any particular session.

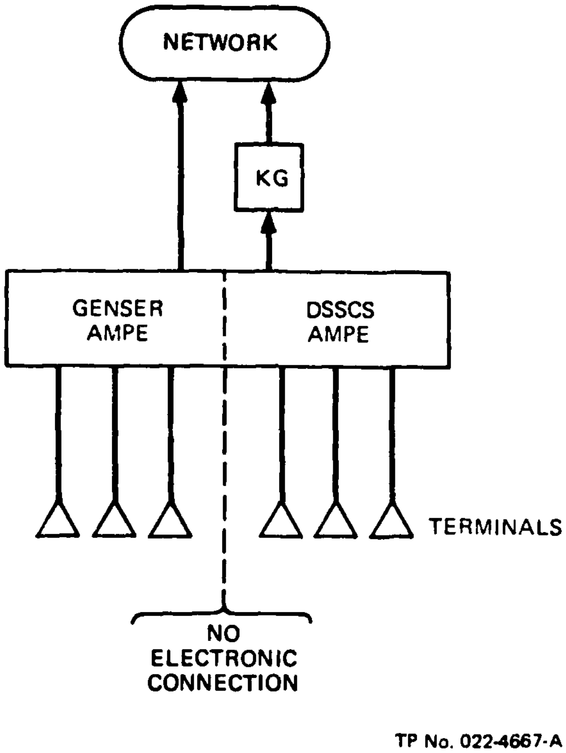


Figure A-1. Separate AMPE Processor for Each Classification Alternative

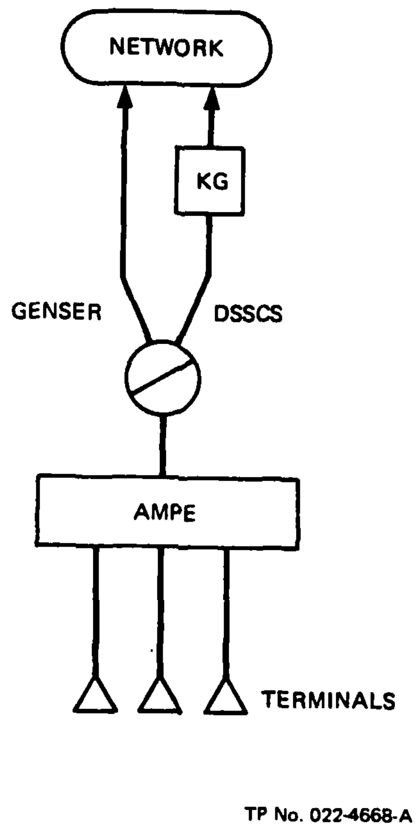


Figure A-2. Dedicated Switching Architecture Alternative

The major cost and schedule effect of this alternative is due to the incorporation of the secure switch. Design and development factors need to be considered if the use of an existing switch is not possible. The software or firmware implementation of the control is also a factor. Additional costs during operational use arise from the additional sanitization and physical security requirements.

A.2.3 Multimicroprocessor Architecture

A third architecture is a multimicroprocessor configuration. Figure A-3 shows the functional architecture of such an I-S/A AMPE processor. It consists of electronically separate circuit boards, each of which is a microprocessor central processing unit (CPU) along with the associated memory. Each board executes programs stored in its local memory independently of the other boards. These boards are referred to as processing units (PUs). The example I-S/A AMPE processor, handling the two levels DSSCS and GENSER, consists of seven processors:

1. GENSER Line Processing Unit (GENSER LPU) - This is the communications interface to the GENSER terminals. It contains hardware/software to accumulate input characters into local memory buffers to form messages.

2. GENSER Message Processing Unit (GENSER MPU) - This unit performs required checks on the message for formatting and other considerations.

3. DSSCS LPU - Aside from being dedicated to DSSCS traffic, this LPU performs the same functions as a GENSER LPU.

4. DSSCS MPU - Handles DSSCS messages only and is identical to the GENSER MPU in function.

5. Common Storage (CS) - This is the CS area that provides auxiliary memory to the local memory on the other PUs, and contains the interprocess (PU to PU) communication areas.

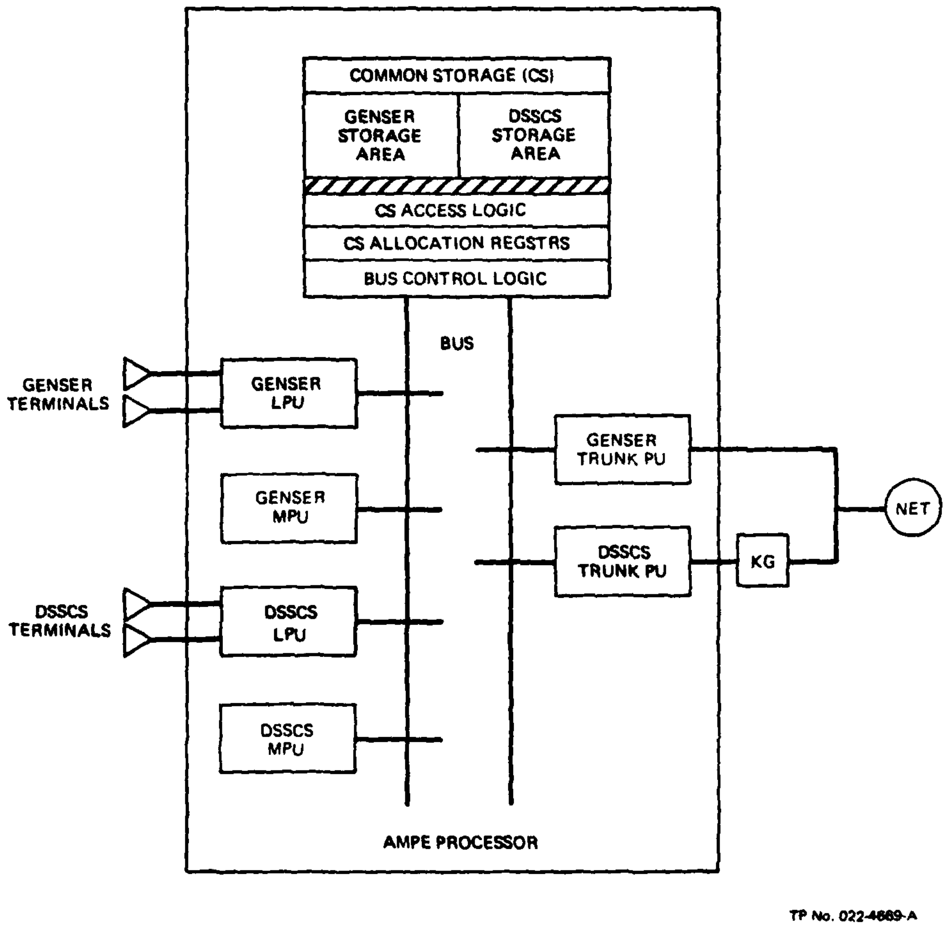


Figure A-3. Multimicroprocessor Architecture Alternative

6. GENSER Trunk Processing Unit (GENSER TPU) - Handles communications protocol with the network and GENSER message transmission to the I-S/A AMPE processor.

7. DSSCS TPU - Handles communications protocol with the network and DSSCS message transmission to the I-S/A AMPE processor.

The boards are not connected either directly or logically, but can only indirectly communicate through the control data bus. Each PU has a local memory bank on board that it alone can access. In addition to its local memory, each PU, or possibly group of PUs, has a portion of common storage allocated to it that it alone can access. This memory access restriction, along with the separation of the boards, provides the MLS capability.

The Access control mechanism that enforces the above described memory access restrictions is the memory addressing logic (MAL). MAL checks all addressing attempts to see if the memory address requested is within the range indicated by a pair of registers that contain the highest and lowest address allowed for a particular PU. A bank of these boundary registers is maintained by the MAL.

The boundary registers are static in nature, that is, they are not changed dynamically once the processor begins operation. They are loaded at initial program load (IPL) time or the addresses are permanently tuned in to restrict the contents to read-only operations. This, in effect, realizes a static memory allocation scheme that is enforced by hardware addressing logic. In the example shown in Figure A-3, the static allocation partitions common storage into GENSER and DSSCS storage areas. Only GENSER LPU, GENSER MPU, and GENSER TPU can access the GENSER side and only the DSSCS associated PUs may access the DSSCS-related areas.

The PUs operate in parallel, independent of each other. When a particular PU wants to do I/O to the CS, it sets a flag on its associated bus interface. The bus control logic continually polls these indicator flags and allows one bus transfer or memory access to occur per board and then continues the polling with a predetermined priority scheme.

This type of architecture is very flexible, for example, if the number of users increased, one or several PU boards could be added (multiple MPU boards are also possible) without major architecture revisions, and with no software changes required. If workload increases and performance begins to suffer, more PUs can be added. Multiple PUs can also provide redundant backup, making a highly reliable architecture.

Primary cost and schedule impact of implementing such a multimicroprocessor architecture is caused by the initial design and development of the hardware, firmware, and software. Accreditation of this alternative will also affect cost and schedule.

A.3 END-TO-END ENCRYPTION

Three basic architectures are presented to describe the use of end-to-end encryption in a secure I-S/A AMPE.

A.3.1 Private Line Interface

The first architecture involves the use of a Private Line Interface (PLI) in the AUTODIN II or similar network, and is illustrated in Figure A-4.

With this architecture, the I-S/A AMPE routes GENSER traffic to a separate port ensuring that DSSCS traffic is not misrouted to the GENSER port. The I-S/A AMPE itself will handle traffic of all levels and compartments. Access to the processor will be controlled by physical security procedures. MLS will be provided in the I-S/A AMPE for information flow to terminal operators and to the network. This MLS protection can take the form of software protection, hardware separation, or, in all probability, a combination of both, in addition to the encryption protection.

The primary cost and schedule factor in this scenario is due to the PLI itself, although other hardware, as well as software, factors need to be considered. A determination of the type and availability of both PLI and crypto-devices is needed to accurately cost this alternative.

A.3.2 Key Distribution Center Mediation

A second architecture is illustrated in Figure A-5. In this case, an end-to-end transaction is made through Key Distribution Center (KDC)

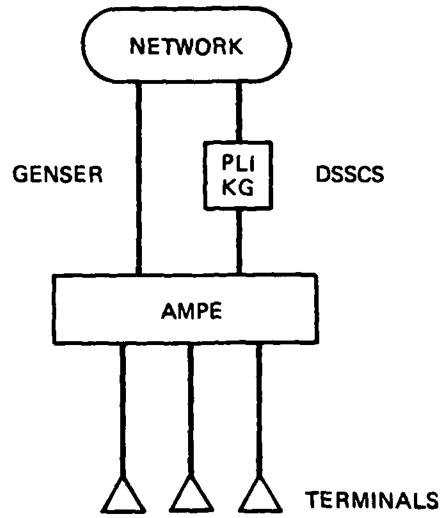


Figure A-4. Incorporation of PLI

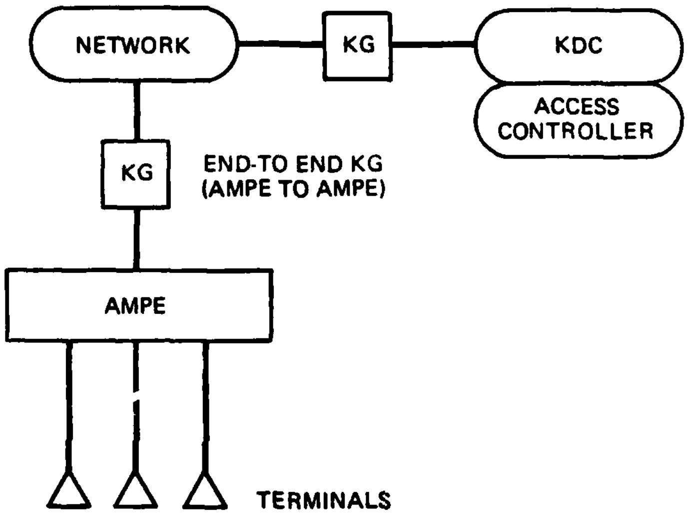


Figure A-5. KDC Mediation

mediation. If DSSCS data is to be passed, the access controller can authenticate the receiving party before distributing the variable for use in the call. The I-S/A AMPE will have access to all information levels and will require terminal security considerations. The I-S/A AMPE will not be concerned with misrouting since the access controller will authenticate and distribute the crypto-variable to the receiving processor after checking for the proper levels of access. MLS is provided by both the encryption devices and the software in the KDC.

In addition to the number and type of crypto-devices required, the cost and schedule effect of providing the necessary hardware and software for the KDC must be evaluated. To accomplish this, a more detailed definition of the KDC functionality is required.

A.3.3 I-S/A AMPE as a BLACK Processor

The third alternative architecture is shown in Figure A-6. In this scenario, the terminals will be end-to-end keyed by the KDC once the proper authorization is established so that the I-S/A AMPE will not handle DSSCS traffic. There may be a need to separate multilevel GENSER traffic in the I-S/A AMPE. If all terminals have their own cryptos, then the processor will contain only BLACK (unclassified) information. The I-S/A AMPE could serve as an electronic mailbox. The MLS capability is provided, therefore, by terminal-to-terminal encryption and control of the crypto-variables.

In this scenario, the major cost and schedule impact is due to the increased number of crypto-devices required. KDC implementation is another factor to be considered.

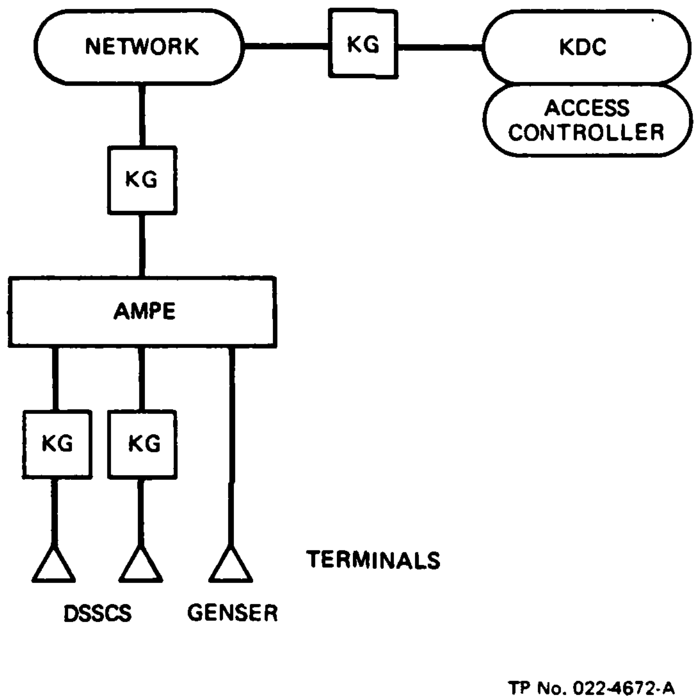


Figure A-6. AMPE as a BLACK Processor

APPENDIX B - REFERENCES

1. Bell, D.E. and L.J. LaPadula, "Secure Computer Systems", ESD-TR-73-278, Volume I-III, The MITRE Corporation, Bedford, MA, November 1973 - June 1974.

2. Freiman, Frank R. and Robert E. Park, "The PRICE Software Cost Model", PRICE Systems, RCA Corp., Cherry Hill, NJ, IEEE Transactions on Software Engineering, 1979.

3. Mohanty, Siba N., "Software Cost Estimation: Present and Future", in Software - Practice and Experience, John Wiley and Sons, Ltd., 1981, pp. 103-121.

4. Nibaldi, G., "Proposed Technical Evaluation Criteria for Trusted Computing Systems", M79-225, The MITRE Corporation, October, 1979.

5. Steffey, Raymond E. Jr., "An Analysis of the RCA PRICE-S Cost Evaluation Model as it Related to Current Air Force Computer Software Acquisition and Management". Master's thesis. Air Force Institute of Technology, Wright-Patterson AFB, OH, December, 1979.

6. Airborne Systems Software Acquisition Engineering Guidebook, "Software Cost Analysis and Estimating", prepared by TRW (R. W. Wolverton) under contract to Aeronautical Systems Division, Wright-Patterson AFB, OH, September, 1980.

7. Data and Analysis Center (DACS) for Software, "Quantitative Software Models", No. SRR-1 prepared by IIT Research Institute under contract to RADC, Griffiss AFB, NY, March, 1979.

8. PRICE Parametric Cost Models. Promotional Literature, RCA PRICE Systems, Cherry Hill, NJ, September, 1981.

9. PRICE & PRICE-S Pocket Operating Guides. RCA PRICE Systems, Cherry Hill, NJ, November, 1981.

10. SLIM (Software Life Cycle Management). Promotional Literature, Quantitative Software Management, Inc., McLean, VA, September, 1981.

APPENDIX C

Model Terminology

APPENDIX C - MODEL TERMINOLOGY

This appendix provides reference material on the software cost models. Section C.1 is a glossary of PRICE-S and PRICE-SL terms. Section C.2 provides parameter calibration tables for major parameters for PRICE-S and PRICE-SL. Section C.3 is a comparative summary of SLIM and PRICE inputs and outputs.

C.1 GLOSSARY OF MODEL TERMS

This section provides a comprehensive reference of terms used by the PRICE-S and PRICE-SL models. Because SLIM operates in a dedicated interactive, self-descriptive mode, such a list of references is not required in this context. General reference can be made, however, to SLIM's basic inputs by referring to Section C.3 of this appendix.

This glossary is separated by terms that directly apply to the Input Data Worksheet (Section I) and terms that are related to the controlled operation of the model (Section II). The operations terminology is subdivided according to the particular operational function to which it is related.

PRICE-S TERMS

I. Input Data:

ITEM	DEFINITION
APPL	Character of the software
APPL8	User defined optional APPL category - MIX line
CAP	Memory capacity in terms of machine level executable instructions
CAPP8	New Code for MAPP8 - New Code line
CDAT	Data Storage and Retrieval - New Code line
CINT	Interactive Operations - New Code line
CMAT	Mathematical Applications - New Code line
CONL	On-Line Communications - New Code line
COPR	Operating Systems - New Code line
CPLX	Development environment/task descriptor
CREA	Real Time Command and Control - New Code line
CSTR	String Manipulation - New Code line
DAPP8	New Design for MAPP8 - New Design line
DCOST	Average cost per man-month or man-hour - Design phase
DDAT	Data Storage and Retrieval - New Design line
DEND	Design phase end date
DINT	Interactive Operations - New Design line
DMAT	Mathematical Operations - New Design line
DMAX	Maximum man-months or man-hours per month - Design phase
DONL	On-Line Communications - New Design line
DOPR	String Manipulation - New Design line
DREA	Real Time Command and Control - New Design line
DSTART	Design phase start date
DSTR	String Manipulation - New Design line
ESC	Escalation
EXPAN	Expansion ratio from High-Order Language to Machine-Order Language
FUNCT	Number of functions in a functional flow diagram
GTABLE	Global Constants file - Contains ATABLE, CTABLE, and DTABLE
ICOST	Average cost per man-month or man-hour - Implementation phase
IEND	Implementation phase end date
IMAX	Maximum man-months or man-hours per month - Implementation phase
INST	Instructions - Project magnitude
INTEG	Integration and Test factor
ISTART	Implementation phase start date
LEVEL	Average level of the Work Breakdown Structure displayed in an equivalent functional tree diagram
MAPP8	User defined optional MIX element - MIX line
MDAT	Data Storage and Retrieval - MIX line
MINT	Interactive Operations - MIX line
MMAT	Mathematical Operation - MIX line
MONL	On-Line Communications - MIX line
MOPR	Operating Systems - MIX line
MREA	Real Time Command and Control - Mix line
MSTR	String Manipulation - Mix line

PRICE-S TERMS (CONT.)

I. Input Data (Cont.):

ITEM	DEFINITION
MULT	Linear multiplier for all costs
NEWC	Amount of New Code required - Input line to PRICE-S
NEWD	Amount of New Design required - Input line to PRICE-S
PLTFM	Customer specifications and reliability requirements factor
QDAT	Data Storage and Retrieval devices - Interace Quantity line
QINT	Interactive Devices - Interface Quantity line
QONL	On-Line Communication Devices - Interface Quantity line
QREA	Real Time Command and Control devices - Interface Quantity line
RESO	Skill level, productivity, efficiency, labor rates, overhead of crew
SOURCE	Number of source statements
STRU	Structure - Calculated value for functional flow diagram
TARCST	Input to Design-to-Cost, APPL, and RESO calibration modes
TCOST	Average cost per man-month or man-hour - Test and Integration phase
TDAT	Data Storage and Retrieval devices - Interface Types line
TEND	Test and Integration phase and date
TINT	Interactive devices - Interface Types line
TMAX	Maximum man-months or man-hours per month - Test and Integration phase
TONL	On-Line Communications devices - Interface Types line
TREA	Real Time Command and Control - Interface Types line
TSTART	Test and Integration phase start date
UTIL	Fraction of available hardware cycle time or total memory capacity used
YEAR	Base year for economics and technology growth

GTABLE = (Program Constants)

ATABLE	Linear multipliers for cost elements in Global Constants
ACD	Design phase - column multiplier
ACI	Implementation phase - column multiplier
ACT	Test & Integration phase - column multiplier
CON	Configuration Control - row multiplier
DOC	Documentation - row multiplier
PGM	Program Management - row multiplier
PRO	Program Management - row multiplier
SYS	System Engineering - row multiplier
CTABLE	Curve Controls in Global Constants
DTABLE	Descriptor global table in Global Constants
RTABLE	Header Dialog control option that allows a customer inflation rate table to be used for the entire run

PRICE-S TERMS (CONT.)

II. Operations Control:

ITEM	DEFINITION
DES	Descriptors - Input line of PRICE-S
INSPF	Instructions per function - Global Constants
OFILE	Output filename - Header Dialog control option
OKSKIP	Header Dialog control option
PRINTG	Header Dialog control option that activates printing of all Global Constants
PRINTP	Header Dialog control option that activates printing of Resource Allocation Profiles
POST	Header Dialog control option that creates a post-processor file
SCH	Schedule - Input line to PRICE-S
SCHED	Schedule Effect Summary - Header Dialog control option
SENSIA	Header Dialog control option that activates printing of INST/APPL Sensitivity matrix
SENSIT	Header Dialog control option that activates printing of RESO/CPLX Sensitivity matrix
SHORT	Header Dialog control option that determines the output print format
SUPP	Supplemental Informtion - Input line to PRICE-S
TOL	Level of tolerance for error messages in PRICE-S
TYPES	Interface Types - Input line to PRICE-S
UNITS	Header dialog control option that controls cost units. Default value: DOLLARS

SENSIA Header Dialog control option that activates printing of INST/APPL Sensitivity matrix

- ASTEP Step size for APPLICATION in SENSIA option
- ISTEP Step size for INSTRUCTIONS in SENSIA option

SENSIT Header Dialog control option that activates printing of RESO/CPLX Sensitivity matrix

- CSTEP Step si~e for COMPLEXITY in SENSIT option
- RSTEP Step size for RESOURCE in SENSIT option

PRICE-S TERMS (CONT.)

Resource Allocation Profile:

ITEM	DEFINITION
DLAG	Lag parameter for Resource Allocation Profile - Design phase
DLEAD	Lead parameter for Resource Allocation Profile - Design phase
ILAG	Lag parameter for Resource Allocation Profile - Implementation phase
ILEAD	Lead parameter for Resource Allocation Profile - Implementation phase
LAG	Parameter for shaping Resource Allocation Profile
LEAD	Parameter for shaping Resource Allocation Profile
TLAG	Lag parameter for Resource Allocation Profile - Test and Integration phase
TLEAD	Lead parameter for Resource Allocation Profile - Test and Integration phase

C.2 PARAMETER CALIBRATION TABLES

This appendix provides reference tables of parameters for PRICE-S and PRICE-SL that are used to calibrate the model to a particular project being estimated. These tables consist of range values that quantify and describe the given project or project component.

These tables describe the following major parameters: APPL (Application) and CPLX (Complexity).

The APPL parameter provides an instruction mix based on different types of applications, such as operating systems and interactive operations, and weighted values to guide the analyst in determining an APPL value appropriate to the particular application.

The CPLX parameter measures development environment factors. Based on a normalized value of 1.0 as a general defense industry average, one or more adjustments can be applied depending on the specific nature of the project and related implications.

Table C-1. Instruction Mix

APPLICATION TYPE	WEIGHT	IDENTIFYING CHARACTERISTICS
OPERATING SYSTEMS	10.95	Task management. Memory management. Heavy hardware interface. Many interactions. High reliability and strict timing requirements.
INTERACTIVE OPERATIONS	10.95	Real time man/machine interfaces. Human engineering considerations and error protection very important.
REAL TIME COMMAND AND CONTROL	8.46	Machine to machine communications under tight timing constraints. Queuing not practicable. Heavy hardware interface. Strict protocol requirements.
ON-LINE COMMUNICATIONS	6.16	Machine to machine communications with queuing allowed. Timing restrictions not as restrictive as with real time command and control.
DATA STORAGE AND RETRIEVAL	4.10	Operation of data storage devices. Data base management. Secondary storage handling. Data blocking and deblocking. Hashing techniques. Hardware oriented.
STRING MANIPULATION	2.31	Routine applications with no overriding constraints. Not oriented toward mathematics. Typified by language compilers, sorting, formatting, buffer manipulation, etc.
MATHEMATICAL OPERATIONS	.86	Routine mathematical applications with no overriding constraints.

Table C-2. Typical CPLX Adjustments

	CPLX ADJUSTMENT
PERSONNEL	
Outstanding Crew, Among Best In Industry	-.2
Extensive Experience, Some Top Talent	-.1
Normal Crew, Experienced	0
Mixed Experience, Some New Hires	+.1
Relatively Inexperienced, Many New Hires	+.2
PRODUCT FAMILIARITY	
Old Hat, Redo Of Previous Work	-.2
Familiar Type Of Project	-.1
Normal New Project, Normal Line Of Business	0
New Line Of Business	+.2
COMPLICATING FACTORS	
First Time With Language	+.1
First Time With Processor	+.1
New Language	+.2 to +.3
New Hardware	+.2 to +.3
More Than One Location/Organization	+.2
Multinational Project	+.4
Hardware Developed In Parallel Or Many Changing Requirements	+.2 to +.3

PRICE-SL - TERMS

I. Input data:

ITEM	DEFINITION
APPL	Character of the software
CPLX	Development environmental task descriptor
DEVCST	Software development cost
DSTART	Design phase start date
EMPLOY	Employment factor per installation
ESC	Escalation
GROWTH	Anticipated growth factor
INST	Instructions - Project magnitude
INSTAL	Number of installations
MULT	Linear multiplier in development cost
NEWC	Amount of New Code included in development
NEWD	Amount of New Design included in development
PLTFM	Customer specification and reliability requirements factor
RESO	Skill level, productivity, efficiency, labor rates, overhead crew during development
SCALE	Development cost unit scale
SEND	Support end date
SESC	Support escalation
SMULT	Linear multiplier of support costs (G&A, profit as fee)
SSCALE	Support cost unit scale
SSTART	Support start date
SYEAR	Base year for support economics
TEND	Test and Integration end date
UTIL	Fraction of available hardware cycle time or trotal memory capacity used
YEAR	Base year for economics and technology growth for development

II. Operations Control:

ITEM	DEFINITION
OFILE	Output filename - Header Dialog control option
RTABLE	Inflation rate table
SHORT	Header dialog print format control option
SKIP PROMPT	Header dialog processing control option
TABLES	Header dialog additional output control option

C.3 COMPARATIVE SUMMARIES OF PRICE AND SLIM

Tables that compare software aspects of the selected models are found in this section. This includes the PRICE-S, PRICE-SL, and SLIM models.

Two distinct summaries, one for inputs and one for outputs, are provided. In both cases, the summaries show how SLIM parameters correspond to PRICE-S and PRICE-SL parameters.

The first summary examines the SLIM inputs and interprets each of these specifications into one or more PRICE-S or PRICE-SL input parameters. The PRICE-S or PRICE-SL interpretation can take different forms. It is:

1. A parameter acronym (which can be found in the PRICE-S or PRICE-SL glossary preceding this)
2. A concise description
3. A combination of both.

Certain symbols are used to represent meaning or action. The right directional arrow (---½) is used to symbolize a parameter specification that is common to both PRICE-S and PRICE-SL. A plus or positive (+) sign indicates that in addition to the PRICE-S parameter specification, PRICE-SL employs a complementary parameter.

The Procedure Name column identifies the particular procedural control mechanism under which are included the specifications shown in the second column. The calibration procedure is the initial main procedure activated to establish the technology factor. The remaining specifications described in the second column are controlled by the build procedure, implying that a parameter data file is to be established on a file storage device to facilitate possible changes to some of the parameters. A procedure enclosed by parentheses indicates that the procedure is a subset of the main procedure.

The second summary provides similar informtion to identify and describe the associated outputs.

INPUT SUMMARY

Procedure Name		SLIM Input Description	PRICE-S (S/W Dev.)		PRICE-SL (S/W Support)
Calibrate	1.	To arrive at Technology factor; Inputs = size, months, man-months			
Build	2.	Assign data file (input & output) names	Specified during data input ops		Can use Development descriptors & economics from PRICE-S
	3.	Start date	YEAR	---½	+ SYEAR
	4.	Monetary unit	Program Constants (Units option)	---½	
	5.	Fully burdened labor rate	Resource Constraints	---½	
	6.	Standard deviation of labor rate			
	7.	Anticipated inflation rate	ESC		+ SESC
	8.	Proportion of development for on-line, interactive mode	MIX		
	9.	Proportion of development computer dedicated to system development	MIX		
	10.	Proportion of development computer dedicated to production work	MIX		
	11.	HOL (high order language)	SOURCE & EXPAN		
	12.	DBMS used in development	APPL		
	12a.	With DBMS, what percent of system in this language			
	13.	Report writer used in development	System Integration (RESO calibration)	---½	

INPUT SUMMARY (CONT.)

Procedure Name		SLIM Input Description	PRICE-S (S/W Dev.)	PRICE-SL (S/W Support)
Build (Cont.)	14.	Type of system	APPL (or sum of MIX if APPL=0)	---½
	14a.	Attributes (new design, interfaces, etc.) of this system	New Design (or NEWD), New Code Code (or NEWC), Interface Types and Quantities	---½
	15.	Proportion of memory of target machine utilized by system	UTIL	
	16.	Proportion of real time code	APPL	
	17.	Proportion of Modern Programming Practices (MPP) used: a. Structured Programming b. Design & Code Implementation c. Top-down development d. Chief programmer teams	RESO, UTIL, CPLX, Schedule (all to some degree)	
	18.	Level of programmer experience: a. Overall skill and qualifications b. With development computer c. With programming practices d. With system or one of similar size & application	CPLX, Schedule	---½
	19.	State of technology	RESO, APPL, DSTART	---½
	20.	Sizing a. By overall system b. Module by module 1. Name of smallest, most likely, and largest possible number	a. INST (or SOURCE & EXPAN) b. Separate run required per component	---½

OUTPUT SUMMARY

Procedure Name		SLIM Output Description	PRICE-S (S/W Dev.)	PRICE-SL (S/W Support)
Estimate (Implementation)	1.	Summary of Input data: Cost elements, Environment, System, MPP, Experience, Technology, Size	Option can be selected after data entered from worksheet	" "
	2.	Simulation - standard deviation for size, time, effort, cost; Sensitivity Profile for Minimum Time Solution; Consistency check on RADC data base; Size-Time-Effort Trade-off plot	Relative to Sensitivity Reports for either CPLX/RESO or INST/APPL; Consistency checks on INST & APPL	
	3.	Manloading - Manpower X Development Time (Staffing Plan) Plot, Staffing Plan chart (by month)	Can run in ECIRP mode to derive RESO; schedule adjusted in form of Schedule Effect Summary Report	
	4.	Cashflow - $ per year X Development Time (cashflow Plan) Plot, Cashflow Plan chart (by month)	Relative to Monthly Progress Summary	
	5.	Code Production - Code Production chart (by month) assuming coding begins at detailed design time	Programming effort/cost by Design, Implementation, T&I in standard report	Program effort/cost by maintenance, enhancement, growth
	6.	Life Cycle - Manpower and Cashflow plots and charts for entire life cycle of project	In standard report for development only	In standard report for support only

OUTPUT SUMMARY (CONT.)

Procedure Name		SLIM Output Description	PRICE-S (S/W Dev.)	PRICE-SL (S/W Support)
Estimate (Implementation) (Cont.)	7.	Milestones - Based on many comparable systems, estimates broken down by: critical design review, system integration test, prototype test, start installation, full operational capability	In standard report by Design, Implementation, T&I	In standard report based on date. Testing ends for development.
	8.	Front End - Minimum time for Feasibility Study and Functional Design, by time and effort		
	9.	Risk Analysis - Risk Analysis profiles in plot and chart form by time, effort, and cost for possible cost/bid/evaluation strategies	Combination of standard & special (most notably) reports in normal mode	
(Misc.)	10.	Pert Sizing: Documentation; Benefit Analysis	Sizing by ECIRP; Documentation in standard report	Documentation effort/cost by maintenance, enhancement, growth
	11.	CPU Usage - in hours by month for entire development cycle	Embedded in UTIL as an input	
(What If)	12.	Linear Program - Minimum Cost and Time solutions, Cost/time tradeoffs, as managerial function	In general by changing Development descriptors & rerunning model	Affected by change in Development descriptors
	13.	Interactive Linear Program - Based on sets of values entered for time and effort, plot is generated showing minimums and maximums for time, effort, and cost. Values can be changed for constraint tradeoffs	Same as above	Same as above

OUTPUT SUMMARY (CONT.)

Procedure Name		SLIM Output Description	PRICE-S (S/W Dev.)	PRICE-SL (S/W Support)
Estimate (Cont.) (What If) (Cont.)	14.	Design to Risk - Trade-off analysis by entering maximum development time and amount (very high = 99 percent, high = 95 percent, medium = 90 percent of risk measured against existing minimum time parameters. In time, effort, and cost.	Chiefly by changing schedule end work figures (for stretchout) and then comparing resultant model standard outputs	Affected by PRICE-S changes to extent of final cost & schedule start supply date
	15.	Design to Cost - Trade-off analysis by changing development effort; requires running man-loading and cashflow for analysis. Consistency check against independent data base.	Special mode (using standard report output) for working around given target cost for investigative feasibility & scope of work	No direct effect except for size of Devopment effort
	16.	Design to Schedule - Same as for #15 above, except for a change to development time versus effort	Directly related to Schedule Effect Summary report as function of changes to CPLX	Influences schedule for start support
	17.	Best Bid - Using maximum development time and cost entries, computes best bid solution and gives probabilities of not exceeding cost and/or schedule. Consistency check performed against independent data base.	In ECIRP mode can use TARCST to derive RESO and CPLX (the ½ it is, the ½ cost). Size can also be derived from TARCST through design-to-cost.	
(Maintenance)	18.	Modify - Capability to change any variable on file.	Has front-end interactive edit function for changing any variable & option.	Can use Development descriptors plus some economic descriptions from PRICE-S data file; has edit capability also.

END

DATE
FILMED

04-82

DTIC