

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

ADA 083208

ASD-TR-78-44

②
NW

LEVEL II

SOFTWARE COST MEASURING AND REPORTING
One of the Software Acquisition
Engineering Guidebook Series

DIRECTORATE OF EQUIPMENT ENGINEERING
DEPUTY FOR ENGINEERING

JANUARY 1979

TECHNICAL REPORT ASD-TR-78-44
Final Report

DTIC
ELECTE
APR 17 1980
S D
B

Approved for public release; distribution unlimited.

AERONAUTICAL SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

FILE COPY

80 4 17 084

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.



RICHARD W. ITTELSON,
Technical Advisor
Directorate of Equipment Engineering



RICHARD J. SYLVESTER,
ASD Weapon Systems Computer Resource
Focal Point
Deputy for Engineering

FOR THE COMMANDER



JOHN S. KUBIN, Colonel, USAF
Director, Equipment Engineering

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify _____, W-PAFB, OH 45433 to help us maintain a current mailing list".

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ASD/TR-78-44	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SOFTWARE COST MEASURING AND REPORTING Software Acquisition Engineering Guidebook Series		5. TYPE OF REPORT & PERIOD COVERED Final Repts
7. AUTHOR(s) D.L./Bowie R.B./Doggett D./Fowler		6. PERFORMING ORG. REPORT NUMBER D180-22813-1
9. PERFORMING ORGANIZATION NAME AND ADDRESS Boeing Aerospace Company P.O. Box 3999 Seattle, Washington 98124		8. CONTRACT OR GRANT NUMBER(s) F33657-76-C-0723
11. CONTROLLING OFFICE NAME AND ADDRESS HQ. ASD/ENE Wright-Patterson AFB, OH 45433		10. PROGRAM ELEMENT PROJECT, TASK AREA & WORK UNIT NUMBERS PE64740F Project 2238
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 12) 977		12. REPORT DATE 2 Jan 79
		13. NUMBER OF PAGES 123
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) 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) Software Acquisition, Acquisition Engineering, Cost Estimating, Cost Measuring, Cost Reporting, Cost/Schedule Status Report, Estimating Model, Life Cycle Analysis, Computer Program Development Cycle, Contract Funds Status Report		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report is one of a series of guidebooks whose purpose is to assist Air Force Program Office Personnel and other USAF acquisition engineers in the acquisition engineering of software for Automatic Test Equipment and Training Simulators. This guidebook describes cost considerations associated with USAF and contractor cost estimating, cost distribution and cost reporting. Also, computer programming cost monitoring and control tasks are described.		

one of the

DTIC
ELECTE
APR 17 1980
S D B

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

059610

FOREWORD

This guidebook was prepared as a part of the Software Acquisition Engineering Guidebooks contract, F33657-76-C-0723. It describes the cost considerations associated with the Air Force/Contractor software procurement, including cost estimating, cost distribution, and cost reporting, as applied to Training Simulators and Automatic Test Equipment. Acquisition engineering tasks are defined and described for computer program cost monitoring and control.

This guidebook is one of a series intended to assist the Air Force Program Office and engineering personnel in software acquisition engineering for automatic test equipment and training simulators. Titles of other guidebooks in the series are listed in the introduction. These guidebooks will be revised periodically to reflect changes in software acquisition policies and feedback from users.

This guidebook reflects an interpretation of DOD directives, regulations and specifications which were current at the time of guidebook authorship. Since subsequent changes to the command media may invalidate such interpretations, the reader should also consult applicable government documents representing authorized software acquisition engineering processes. This guidebook contains alternative recommendations concerning methods for cost-effective software acquisition. The intent is that the reader determine the degree of applicability of any alternative based on specific requirements of the software acquisition with which he is concerned. Hence, the guidebook should only be implemented as advisory rather than as mandatory or directive in nature.

This guidebook was prepared by the Boeing Aerospace Company.

ACCESSION for	
NTIS	White Section <input checked="" type="checkbox"/>
DDC	Buff Section <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION _____	
BY _____	
DISTRIBUTION/AVAILABILITY CODES	
Dist. AVAIL. and/or SPECIAL	
A	

This Software Acquisition Engineering Guidebook is one of a series prepared for Aeronautical Systems Division, Air Force Systems Command, Wright-Patterson AFB OH 45433. Inquiries regarding guidebook content should be sent to ASD/ENE, Wright-Patterson AFB OH 45433. The following list presents the technical report numbers and titles of the entire Software Acquisition Engineering Guidebook Series. Additional copies of this guidebook or any other in the series may be ordered from the Defense Documentation Center, Cameron Station, Alexandria VA 22314.

ASD-TR-78-43,	Computer Program Maintenance
ASD-TR-78-44,	Software Cost Measuring and Reporting
ASD-TR-78-45,	Requirements Specification
ASD-TR-78-46,	Computer Program Documentation Requirements
ASD-TR-78-47,	Software Quality Assurance
ASD-TR-78-48,	Software Configuration Management
ASD-TR-78-49,	Measuring and Reporting Software Status
ASD-TR-78-50,	Contracting for Software Acquisition
ASD-TR-79-5042,	Statements of Work (SOW) and Requests for Proposal (RFP)
ASD-TR-79-5043,	Reviews and Audits
ASD-TR-79-5044,	Verification, Validation and Certification
ASD-TR-79-5045,	Microprocessors and Firmware
ASD-TR-79-5046,	Software Development and Maintenance Facilities
ASD-TR-79-5047,	Software Systems Engineering
ASD-TR-79-5048,	Software Engineering (SAE) Guidebooks Application and Use

TABLE OF CONTENTS

<u>Section</u>	<u>Title</u>	<u>Page</u>
1.0	INTRODUCTION	1
1.1	Purpose	1
1.2	Scope	1
1.3	TS and ATE Overview	2
	1.3.1 TS System Characteristics	2
	1.3.2 ATE System Characteristics	2
1.4	Guidebook Organization	5
2.0	APPLICABLE DOCUMENTS	7
3.0	SOFTWARE COST PERSPECTIVE	9
3.1	Software Life Cycle Analysis	9
	3.1.1 Development Cost Estimating Models	9
	3.1.2 Operation and Support (O&S) Cost Models	10
3.2	Cost Estimating Methods and Assumptions	11
	3.2.1 Methods of Estimating	12
	3.2.2 Work Breakdown Structure (WBS)	13
3.3	High Leverage Cost Factors	14
3.4	Software Cost Reporting and Regulations, Specifications and Standards (RSS)	16
	3.4.1 Documentation Overview	16
	3.4.2 Contractors AFSCP/AFLCP173-5 Implementation System	19
	3.4.3 ATE & TS Consideration	19
3.5	Unique Considerations for ATE	21
	3.5.1 Trade Consideration	21
	3.5.2 Time Consideration	21
	3.5.3 Documentation Required	22
	3.5.4 Additional Consideration	23
3.6	Unique Consideration for Training Simulators	23
	3.6.1 Frequent or Unnecessary Changes	23

TABLE OF CONTENTS (Cont)

<u>Section</u>	<u>Title</u>	<u>Page</u>
	3.6.2 Incremental Procurement	23
	3.6.3 Concurrent Procurement	23
	3.6.4 Evolutionary Change	23
	3.6.5 Inadequate Preparation	24
	3.6.6 Use of Existing Software	24
	3.6.7 Additional Considerations	24
4.0	SOFTWARE DEVELOPMENT COSTS	25
	4.1 Analysis	32
	4.2 Software Design	36
	4.3 Coding and Checkout	37
	4.4 Integration and Test	41
	4.5 Total Development Costs	41
5.0	SOFTWARE MAINTENANCE COSTS	43
	5.1 Software Maintenance Facilities & Activities	43
	5.2 Change Estimating	44
	5.2.1 Step 1-Estimating Difficulty of Task	44
	5.2.2 Step 2-Estimating the Change Rate	44
	5.2.3 Step 3-Estimating the Labor Expenditure	45
	5.2.4 Step 4-Estimating Personnel Turnover	45
	5.2.5 Step 5-Estimating Labor Cost Rates	45
	5.2.6 Step 6-Estimating Total Labor Costs	45
	5.3 Total Maintenance Costs	45
6.0	SOFTWARE COST REPORTS	47
	6.1 Report Requirements	47
	6.2 Cost Reports Description	48
7.0	BIBLIOGRAPHY	57
8.0	MATRIX: GUIDEBOOK TOPICS VS GOVERNMENT DOCUMENTS	59
9.0	GLOSSARY OF TERMS	61

TABLE OF CONTENTS (Cont)

<u>Section</u>	<u>Title</u>	<u>Page</u>
10.0	ABBREVIATIONS AND ACRONYMS	65
11.0	INDEX OF DOCUMENT TOPICS	67
APPENDIX A	SUMMARY OF EIGHT COST MODELS	A-1
APPENDIX B	WORK BREAKDOWN STRUCTURE TASKS	B-1
APPENDIX C	INTEGRATED MANAGEMENT SYSTEM	C-1

LIST OF FIGURES

<u>Figure</u>	<u>Title</u>	<u>Page</u>
1.3.1	Typical Crew Training Simulator	3
1.3-2	Typical ATE Configuration	4
3.4-1	Distribution of Federal Spending	17
3.4-2	Distribution of Defense Spending	18
3.4-3	ASPR Statement Relative to C/SCSC	20
4.0-1	Trainer Simulator Software Development Process	26
4.0-2	Automatic Test Equipment Software Development Process	28
4.1-1	Typical Contractor's Engineering Estimating Form	35
4.3-1	Digital Circuit Card Test Software Generation Cost	39
6.2-1	Cost Performance Report/Work Breakdown Structure	49
6.2-2	Cost Performance Report/Functional Categories	50
6.2-3	Cost Performance Report/Baseline	51
6.2-4	Cost Performance Report/Manpower Loading	52
6.2-5	Cost Performance Report/Problem Analysis	53
6.2-6	Cost/Schedule Status Report	54
6.2-7	Contract Funds Status Report (CFSR)	55
8.0-1	Guidebook Topics Versus Government Documentation	60
C-1	Contract Responsibility Matrix (CRM)	C-2
C-2	Cost Account Definition	C-3
C-3	Cost/Schedule Indices Based on Variant Milestone BCWP Measurement Method	C-6
C-4	Cost/Schedule/At-Completion Variances Based on Variant Milestone BCWP Measurement Method	C-8
C-5	Management Estimate-At-Completion	C-9
C-6	Variance Analysis	C-11

LIST OF TABLES

<u>Table</u>	<u>Title</u>	<u>Page</u>
4.0-1	Typical 100 Instruction Software Module Development Manhour Estimation	31
4.0-2	Initial Software Estimate Example	34
4.3-1	ATE LRU Test Software Development Costs	40

Section 1.0 INTRODUCTION

Although Training Simulator (TS) and Automatic Test Equipment (ATE) ground systems are often thought of in terms of hardware capital expenditure items, one of the major single cost areas for most modern large scale TS and ATE systems is in software development. The ease of visualizing physical hardware configuration items and the corresponding relative difficulty in visualizing the software development processes, tends to obscure software procurement as a primary cost factor. This guidebook is concerned with the financial considerations pertinent to TS and ATE software.

1.1 PURPOSE

The primary purpose of this guidebook is to assist Air Force engineering personnel responsible for TS and ATE acquisition, to assure the cost considerations for software are successfully developed and monitored. This guidebook provides guidelines to familiarize Air Force personnel with the basic software cost aspects of ATE and TS. The guidebook also provides convenient references to appropriate Department of Defense (DOD), Air Force, and MIL-STD documents which bear on TS and ATE software cost functions.

1.2 SCOPE

This is one of a series of guidebooks related to the Software Acquisition Engineering (SAE) process for TS and ATE ground based systems. The SAE guidebook titles are listed below:

- Software Cost Measuring and Reporting
- Requirements Specifications
- Contracting for Software Acquisition
- Statement of Work (SOW) and Requests for Proposal (RFP)

Regulations, Specifications and Standards

Measuring and Reporting Software Status

Computer Program Documentation Requirements

Software Quality Assurance

Verification

Validation and Certification

Computer Program Maintenance

Configuration Management

Reviews and Audits

Management Reporting by the Software Director

For the purposes of this guidebook, software Cost Measuring and Reporting includes all software cost considerations from initial cost estimates developed in the software life cycle analysis through cost reports following software task accomplishment. Included are life cycle analysis; cost trades, costs per stage of software development; cost of software maintenance and changes; and appropriate cost reports. Methods and processes involved in software costing are described without reference to a specific internal contractor's methods.

Substantial commonality exists between TS and ATE software cost considerations, but development of TS or ATE unique items are specifically addressed in each document section. Although the guidebook is written primarily for technical personnel responsible for the engineering acquisition of ATE and TS systems, financial officers and managers can also make valuable use of this guidebook.

1.3 TS AND ATE OVERVIEW

1.3.1 TS System Characteristics

The TS system is a combination of specialized hardware, computing equipment, and software designed to provide a synthetic flight and/or tactics environment in which aircrews learn, develop and improve the skills associated with individual and coordinated tasks in specific mission situations. Visual, aural, and/or motion systems may be included. Figure 1.3-1 depicts a typical training simulator which employs digital processing capability.

The computer system integral to the crew training simulator, can consist of one or more general purpose computers. The computing hardware operates with floating point arithmetic and sufficient bit capacity to provide efficient use of a simulator Higher Order Language (HOL).

When a multi-processor/multi-computer system is used, it must be designed such that computers can operate simultaneously and are controlled/synchronized by a single program (supervisor/executive). The executive directs program execution and regulates priorities.

The simulator (1) accepts control inputs from the trainee (via crew station controls) or from the instructor operator station; (2) performs a real-time solution of the simulator mathematical model; and (3) provides output responses necessary to accurately represent the static and dynamic behavior of the real world system (within specified tolerance and performance criteria).

Since TS consists of interdependent hardware and software, a joint hardware/software development effort is required. As the complexity of training simulators increases, simulation software continues to grow in complexity, size, and cost. Software costs can and do exceed computer hardware costs in many cases. Therefore, it is imperative that the simulation software acquisition

engineering process be subjected to formal system engineering planning and discipline to ensure cost-effective procurement.

1.3.2 ATE System Characteristics

ATE is defined as that ground support system which performs vigorous system test with minimum manual intervention. ATE is used in place of manual devices because it is more cost effective, provides required repeatability, or repair of the item being tested requires the speed which only an automatic tester can achieve (e.g., the complex initialization routine that is normally required prior to testing a digital unit).

Figure 1.3-2 shows the typical components of an ATE system. Note that there are both hardware and software elements involved. Most of the elements shown in the figure will be found in the majority of ATE systems although the packaging and interface design may vary between specific systems.

The controls and displays section consists of the computer peripheral devices such as control panels, magnetic tape cassettes or disks, a cathode ray tube (CRT), keyboard, and small printer. The computer (normally a minicomputer), as controlled by software, operates the peripheral devices; switches test stimuli on and off; and measures responses of the Unit Under Test (UUT) (comparing to predetermined values). The operator maintains supervisory control of the testing process through the peripherals. However, his interaction is usually minimal since, by definition, the automatic test feature was selected in preference to an operator-controlled test system.

ATE is normally designed to accommodate testing several different articles of system equipment (normally one at a time). The maintenance level being supported by the ATE is determined by logistics systems analysis.

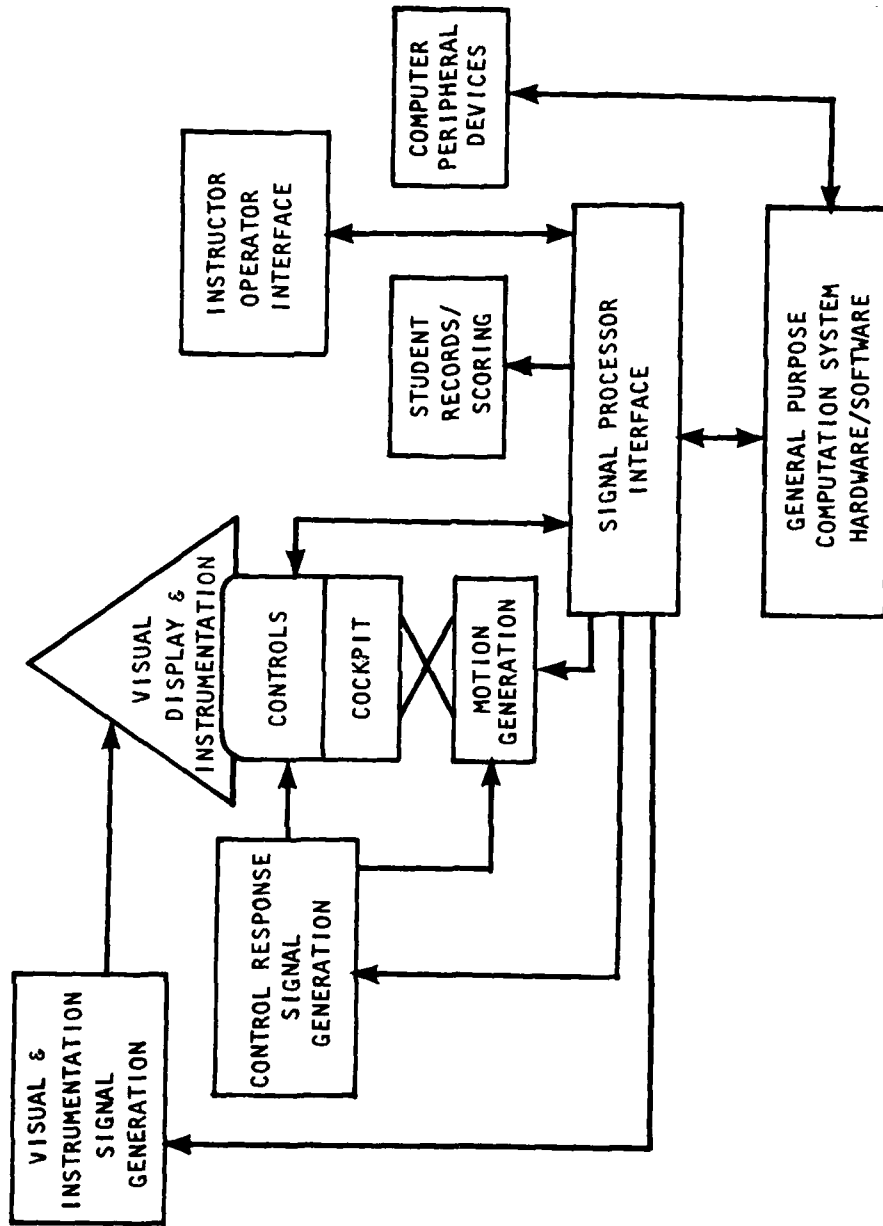


Figure 1.3-1. Typical Crew Training Simulator

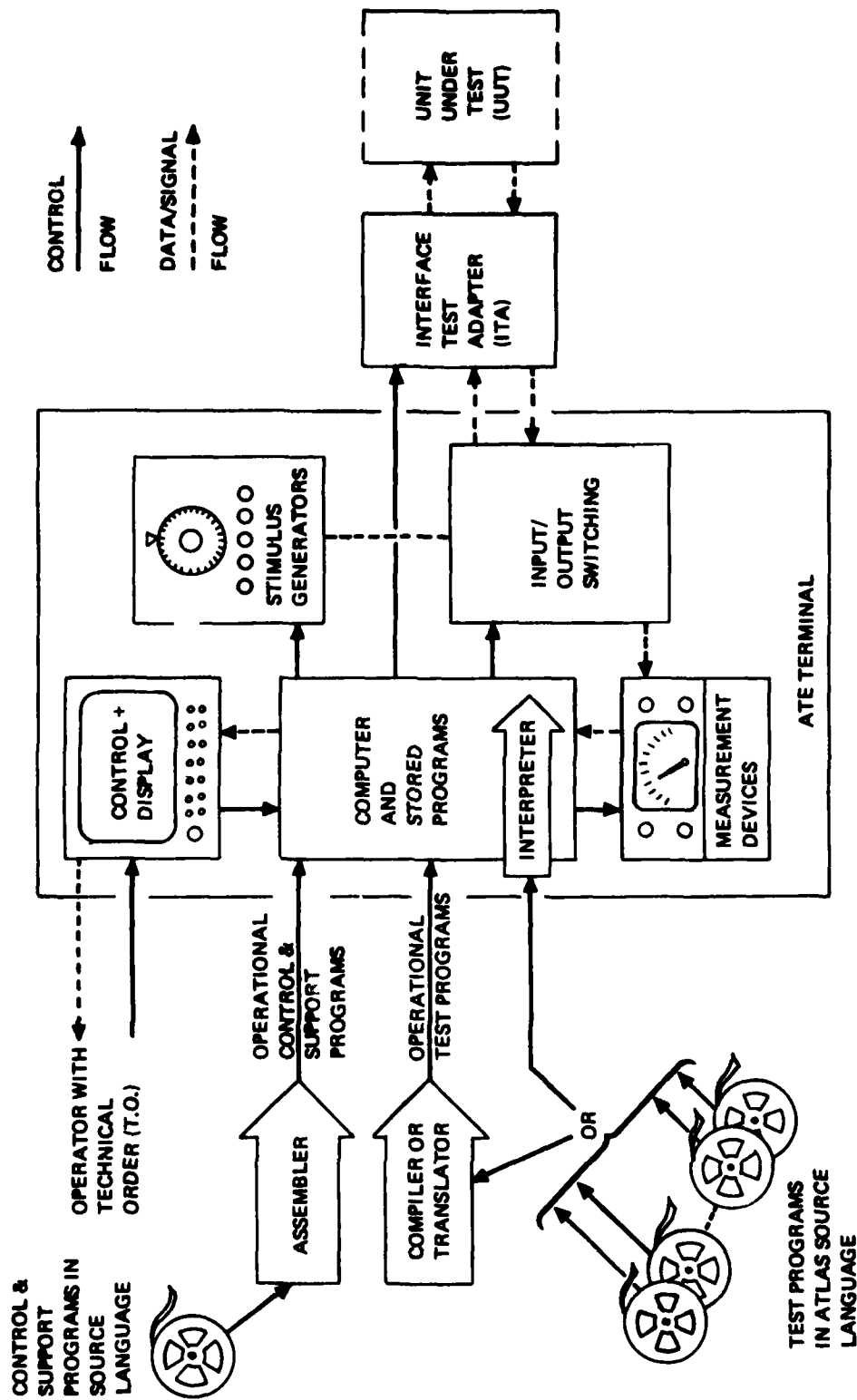


Figure 1.3-2. Typical ATE Configuration

The importance of the software portion of the ATE system should not be minimized since both the application of the test stimuli and the measurement of the result are achieved via software. Arbitrary function generation and complicated wave analysis can also be accomplished by software and is becoming more prevalent in ATE systems. The cost of ATE software is a significant component of total ATE costs and design trades can be performed to minimize ATE life cycle costs.

1.4 GUIDEBOOK ORGANIZATION

Section 1.0 of this guidebook contains introductory material about the guidebook, including guidebook purpose and scope, and the guidebook's relationship to the other SAE guidebooks. It provides a brief description of typical ATE and TS systems and describes the organization and use of the guidebook.

Section 2.0 is a list of key government documents that were referenced in the preparation of this guidebook. Sections 3 through 6 contain ground systems software cost guidelines. Section 3.0 provides an overall software cost perspective including factors and trades that should be considered while still in the pre-development phase. This section is an introduction to software costing and it contains discussion of those areas which are unique to TS and to ATE.

Section 4.0 addresses the cost associated with the various stages of software development starting with the requirement specification and continuing through the verification and validation testing. Section 5.0 considers the specific area of maintenance costs and change estimating. These subjects while not always appreciated during initial development, often have cost impacts that are a substantial portion of total software costs. Section 6.0 describes cost reports and cost requirements that are associated with software development.

Section 7.0 is a bibliography of documents that are generally applicable to the subject of software costs. This section is an expansion of Section 2.0, referenced documentation and the list includes documents not specifically noted in the text. Section 8.0 provides a matrix tabulation for the cross reference relationship between guidebook topics and corresponding government documents. Section 9.0 and 10.0 contain respectively a glossary of selected terms used in the guidebook, and the expansion of all abbreviations and acronyms used in the guidebook. Section 11.0 is a detailed subject index indicating which guidebook paragraphs address the listed subjects.

Section 2.0 APPLICABLE DOCUMENTS

The following documents and articles bear directly on the topic of cost considerations for ATE and TS software:

- AFAL report F33615-77-C-1252, developed under PE62204F, Project 2003, task 09, Work Unit 02, June-August 1976
- TRW-SS-72-01, The Cost of Developing Large Scale Software, R. W. Wolverton, March 1971 (IEEE Computer Society R-77-189)
- AFAL/AAA-3, Modified Wolverton Model
- AFSC Electronic System Division Software Workshop Summary Notes, October 1974
- A Provisional Model for Estimating Computer Program Development Costs, Brad C. Frederic (Tecalote Research, Inc.) December 1974
- 1975 Aerospace Corporation report on cost estimating
- Estimation of Computer Requirements and Software Development Costs, M. S. Taback and M. C. Ditmore (General Research Corporation) March 1974
- Preliminary Study on Estimating the Development Cost of Software, R. H. Darrow, Boeing Co., Document D180-20144-1
- A Review of Software Cost Estimation Methods, Judith A. Clapp (The MITRE Corp.)
- DOD Directive 7000.1 Resource Management Systems of DOD, 22 August 1966
- DOD Directive 5000.1 Acquisition of Major Defense Systems, 18 January 1977 (revised)
- DOD Directive 5000.2 Major System Acquisition Process, 18 January 1977 (revised)
- DOD Instruction 7000.2 Performance Measurement for Selected Acquisitions, 10 June 1977 (revised)
- MIL-STD-881A, Work Breakdown Structures for Defense Material Items, 25 April 1975
- Armed Forces Procurement Regulation (ASPR)
- AFSCP/AFLCP173-5
- AFR173-10
- DOD Instructions 7041.3, 26 February 1969
- Avionics Intermediate Support for Advanced Medium STOL Transport, Technical Memorandum ENEG-TM-77-1, May 1977
- "Handbook of Logic-Circuit Testing", Omnicomp, Inc. 1974
- MIL-STD-1513 (USAF), Trade Studies for the Selection of Avionic Test Support Systems, Criteria for 15 January 1971

Section 3.0 SOFTWARE COST PERSPECTIVE

This section describes the analysis of total life cycle costs, cost estimating methods and factors affecting cost, software cost reporting and unique considerations for ATE and TS.

There is a distinction between price and cost. The only way of looking at these terms is that the price the government or customer pays the contractor equals the cost incurred plus the profit fee for doing the work. Cost does not include fee.

Software life cycle costs consists of those costs associated with software development and those associated with long term operation and support after delivery of the ground system. Several cost models applicable to ground systems are described.

Cost estimating for software is at best an inexact science. While methods and assumptions are described, these must be tempered with judgment based on knowledge of the particular ground system for which the methods are to be applied.

Thorough knowledge of the factors which significantly affect cost is a primary aid to reduction of cost estimating uncertainty. These factors are discussed through the guidebook. Experience has shown control of software costs is not fundamentally different than controlling costs for development/maintenance of any system. The primary management method is first creation of a credible plan, then continually measuring performance against the plan. Specific corrective action is taken when measured performance deviates from the plan. Application of this principal to software cost management for ATE/TS systems involves cost reporting, cost control and cost accounting. These methods are described in the following paragraph.

The nature of ATE and TS software development is such that unique acquisition factors for such systems must be considered. ATE control and support software

are normally provided with the ATE hardware procurement, whereas UUT test software is procured by separate contract. TS software costs are directly affected by the number of costly Engineering or Contract Change Proposals (ECPs or CCPs) which are implemented during TS development. This number is inversely proportioned to the quality of requirements specification and system planning which was incorporated early in the TS procurement cycle. It is also affected by the change rate of the system being simulated.

3.1 SOFTWARE LIFE CYCLE ANALYSIS

Life cycle costs are total costs associated with the software throughout its useful life. System acquisition policy regarding life cycle cost method is set forth in AFR 800-11. These costs, for purposes of Life Cycle Analysis, are separated into two categories: software development costs and software operation and support (O&S) costs. Since the activities of software development and software O&S are markedly different and, for ATE/TS systems, the division of responsibility between the USAF and its contractors normally differs between these activities, the models by which life cycle costs are estimated differ in each case. A number of mathematical models have been developed for the purpose of estimating software development costs. However, comparatively less work has been done in the area of O&S cost estimating. As a result, extensive literature search coupled with information provided by Boeing life cycle cost specialists, has failed to identify a reliable O&S cost estimating mathematical model which is applicable to ATE/TS systems. Thus, information herein relating to this area is limited to discussion of some "rules of thumb".

3.1.1 Development Cost Estimating Models

Eight currently used models are summarized in Appendix A. For a more detailed discussion on the use and limitations of the models, the reader is directed to

AFAL report F33615-77-C-1252, developed under PE62204F, Project 2003, Task 09, Work Unit 02, (during the period June 1976 to August 1976) or to the company sources identified in Appendix A. Any of these models are potentially applicable to ATE/TS software. The selection of a particular model depends upon the applicability and availability of input information required by that particular model. Therefore the selection of a model is made based upon availability of reliable information, such as estimated number of computer instructions, type of instructions, etc., associated with a particular ATE or TS application.

The eight models summarized in Appendix A are:

- a. Wolverton
- b. Modified Wolverton
- c. ESD
- d. Telecote
- e. Aerospace
- f. GRC
- g. Price
- h. Boeing

None of these models offer a reliable panacea encompassing all software projects, therefore no one model can be recommended for universal application. Software estimating is still heavily dependent on experienced judgement. However, quantitative methods, such as described in Appendix A, are valuable for bracketing an estimate. The reader is urged to peruse Appendix A to gain insight on the nature of software estimating methods.

3.1.2 Operation and Support (O&S) Cost Models

The state of the O&S cost estimating is such that mathematical model estimating of O&S costs is in its infancy. Because

of this, subjective evaluation of available estimating parameters should be made in order to project O&S costs for a particular ATE or TS.

Many factors contribute to software O&S costs. An analysis of existing O&S experience should be made to separate the cost drivers from those factors which do not play a significant role in determining software O&S costs. When looking at the data, examination should include the impact of technology factors (e.g., programming techniques), maintainability factors (e.g., documentation and design methods), functional factors (e.g., program types), complexity factors, programming language factors, hardware factors, environmental factors, scheduling factors, and staffing procedures. Some of the problems likely to be encountered in the development of a software O&S data base include inability to distinguish between development and maintenance activities, inconsistency in code types, styles and documentation, and biased activity reporting, because programmers usually prefer to be associated with software development rather than software maintenance.

Once the software program has been accepted, continual support must be furnished to modify the software package to meet changing mission and performance requirements. Besides the modifications to software programs, corrections must be made to previously undetected errors which occur. Support is required to ensure that the program performs its intended functions properly.

The software development process is typically oriented toward minimizing the total development time or maximizing the program's efficiency. In a study conducted by AFAL on the relative amount of time spent on software maintenance (see Bibliography reference 3), it was shown that most software facilities spent somewhere between 20 and 30 percent of their time on software maintenance, but some installations spent 90 to 100 percent of their time maintaining software. They concluded Air Force avionics

software is much like the latter experience and currently it costs something like \$75 per computer (object code) instruction to develop the software, but the maintenance of the software has cost up to \$4,000 per instruction. ATE/TS systems are not significantly different than avionics in this regard.

Judith A. Clapp (The MITRE Corp.) in "A Review of Software Cost Estimation Methods" noted the fact that 54% of all errors were found after acceptance tests were conducted and of these 84% were design errors. Also, of the total number errors found, 64% were attributed to mistakes in design. Throughout the development phase relatively little thought is usually given about what will happen after development is completed. According to an AFSS study, the things are likely to happen after development: (1) Another organization will want to use all or part of the software for its application, (2) the user will upgrade eventually to a new machine and will wish to convert the software, and (3) users will quite frequently want the programs changed to meet new requirements, produce new reports, accommodate new inputs, clear up inconsistencies, add new options, etc.

A search of current literature resulted in very little in the way of predicting support and maintenance cost for computer software. Unlike hardware O&S models, where the cost of spares, maintenance manhours, materials, training, etc., can be estimated based on some physical characteristics of the system, software maintenance is strictly a function of manhours to perform the necessary actions. Thus far, maintenance costs for software seem to be primarily an engineering estimate by an expert, someone familiar with the changes to be made to a program, rather than putting certain parameters into a mathematical model and calculating annual costs.

The "Aerospace Model", as summarized in Appendix A, is a total life cycle cost model. The procedure permits costs for

design and development, investment and operations and maintenance to be determined in a series of prearranged steps. The model first calculates hardware Central Processing Unit (CPU) costs, then applies factors for estimating the other Design and Development (D&D), investment, and Operation and Maintenance (O&M) costs, and finally summarizes the total program costs. The primary maintenance equations for software appear as follows:

a. Software training costs (in \$) during production phase:

Initial Civilian = number of men x 27,200

Initial Contractor = number of men x 35,598

Initial Military = number of men x 17,400

b. During the deployment phase:

$$\frac{\text{Personnel contractor support cost}}{= (\text{number of men}) \times \$48,000 \times (\text{number of years O\&M or deployment})}$$

$$\frac{\text{Military support cost}}{= (\text{number of men}) \times \$18,000 \times (\text{number of years O\&M or deployment})}$$

These equations should only be used if the estimator has no prior basis for determining costs of any of his data-processing system elements. The model, as mentioned above, calculates hardware and software costs, and is referred to as a "Data Processing System Cost Model".

Several companies, including IBM and Boeing, use a standard figure of 1 man/10,000 instructions for estimating software O&S costs. As of this writing there is no more sophisticated model for estimating O&S costs available.

3.2 COST ESTIMATING METHODS AND ASSUMPTIONS

The subject of ground system software cost estimating is an inexact science. As mentioned earlier, regardless of the

method employed, a key factor is the judgement exercised by the estimator. A particular concern to the acquisition engineer is that cost estimates must be made for software systems which will be at least partially developed by an as yet unidentified contractor. Any estimating method will require that the estimator provide information, such as the number of computer instructions or the degree of difficulty or the frequency of modification, etc., which does not exist in any precise way at the time the estimate is to be made. Hence, the element of prior experience becomes of particular importance. Significant experience data exists, for example, with the cost of prior high technology training simulators such as the Advanced Simulator in Undergraduate Pilot Training (ASUPT). This experience information would be highly applicable to similar training systems.

In general, cost information is submitted by the contractor for all USAF systems, and ATE/TS systems in particular, in accordance with the Contractor Cost Data Reporting System. This provides a source of standardized cost information which may be used to provide experience data applicable to ATE/TS systems.

3.2.1 Methods of Estimating

The following general methods are applicable to ATE and TS systems. These classical methods are used throughout industry, and compared to the gross models used for life cycle costing (Paragraph 3.1.1), provide means for carefully detailed cost estimates that are sufficiently accurate for price negotiations.

3.2.1.1 Top-Down Estimating. The estimator relies on the total cost or the cost of large portions of previous projects which have been completed, in order to estimate the cost of the project to be estimated. History, coupled with informed opinion (or intuition), is used to allocate costs between packages. Among its many pitfalls is the substantial risk of over-

looking special or difficult technical problems that may be buried in the project tasks and the lack of details needed for cost justification.

3.2.1.2 Similarities and Differences Estimating. The estimator breaks down the jobs to be accomplished to a level of detail where the similarities to, and differences from previous projects, are most evident. Work units that cannot be compared are estimated separately by some other method. This is particularly useful if a work breakdown structure (WBS) is developed. WBS is discussed in paragraph 3.2.2.

3.2.1.3 Ratio Estimating. The estimator relies on sensitivity coefficients, or exchange ratios, that are invariant (within limits) to the details of design. The software analyst estimates the size of a module by its number of object instructions, classifies it by type, and evaluates its relative complexity. An appropriate cost matrix is constructed into a cost data base in terms of cost per instruction for that type of software, at that relative complexity level. Other ratios, empirically derived, can be used in the total estimating process; for instance, computer usage rate (based on CPU time per instruction), peripheral usage to CPU usage, engineers per secretary, and so forth. It suffers, as do all methods, from the need for a valid cost data base for many estimating situations (ATE test software versus control and support software, real time TS software vs. nonreal time printing, scoring, etc., software in a TS, etc.).

3.2.1.4 Standards Estimating. The estimator relies on standards of performance that have been systematically developed. These standards then become stable reference points from which new tasks can be calibrated. Many mature industries, such as manufacturing and construction, use this method routinely. The method is accurate only when the same operations have been performed repeatedly and good records are available. The pitfall is that custom software development is not "performed repeatedly."

3.2.1.5 Bottom-Up Estimating. The total job is broken down into relatively small work packages and work units (WBS). The work breakdown is continued until it is reasonably clear what steps and talents are involved in doing each task. Each task is then estimated and the costs are pyramided to form the total project cost. An advantage of this technique is that the job of estimating can be distributed to specialists who are most familiar with the work. One difficulty is the lack of immediate perspective of the most important parameter of all; the total cost of the project. In detailed estimates, the estimator is not sensitive to the reasonableness of this total cost of the software package. Therefore, top-down estimation is often used as a check on the bottom-up method.

3.2.1.6 Experience Method. This approach takes advantage of experience on a similar job. In order to use it, the new job must be clearly specified at least down to a major subsystem level. This permits the estimator to compare the new system to one or more completed systems. At this point, the estimator can assume that like tasks take like resources. He can obtain the base data from his own experience or from that of others as long as he knows he is comparing similar projects. If the two projects are alike in size and content, minor differences in algorithms or utility routines can be accounted for by adding a contingency factor to the total estimate. In this method, the contingency should be less than 25%. As in any method, it is wise to lay out the design in detail to permit the men who must implement the job to make their own estimates on their portion of the job. Their estimates will also be based on experience and should be more precise than the total estimate. For example, two similar training simulator systems of 150,000 instructions each may be within 25% of one another in effort. Two control sub-routines of 1000 instructions each may appear in these systems.

The major problem in the method is that it does not work on systems larger than the base used for comparison. Experience has shown complexity may grow as the square of the number of system elements. Therefore, experience with a relatively small system may not account for all the things that must be done for a large system. Neither will the Experience Method apply to systems of totally different content. The Quantitative guideline may be applicable in such cases.

3.2.1.7 Quantitative Method. The Quantitative Method is based on programmer productivity in terms of the number of instructions produced per unit of time by an average programmer. The instructions include the source statements and data descriptions written by the programmer in a HOL. The method is not precise. It is manually necessary to adjust the answer by large amounts. The estimator should not treat the results as anything other than approximate representations of system size or manpower requirements. The estimate of number of instructions, upon which the "quantitative" method is applied, is subject to serious uncertainty.

3.2.2 Work Breakdown Structures (WBS)

Several cost estimating methods involve development of WBS data. The purpose of the WBS is to identify each elemental task involved in a larger system activity. The assumption made when using the WBS is that, while estimating a larger job is difficult, the elements of the job may be estimated more easily and that the total job is then the sum of the elements. A detailed WBS model is presented in Appendix B and is applicable to ATE/TS systems in general, but should be tailored in each case (by elimination of extraneous tasks). In many cases, particularly ATE test software, a more detailed breakdown will be required.

The WBS model in Appendix B involves five phases. Both tasks and sub-tasks

are identified for each phase. The level of activity for these elements (i.e., tasks and sub-tasks) provides sufficient detail for application to the "Bottom Up" and other estimating methods previously discussed.

The principal phases are:

- a. Phase I - Requirements Definition and Analysis (R&DA)
- b. Phase II - Preliminary Design
- c. Phase III - Detailed Design
- d. Phase IV - Software Construction
- e. Phase V - Software Validation and Verification (V&V)

WBS model in Appendix B provides a detailed listing of specific tasks in software development and can serve as a valuable checklist for acquisition planning (in addition to cost estimating).

3.3 HIGH LEVERAGE COST FACTORS

In general, the highest leverage cost items for ground system software are:

a. Unique requirements. Requirements which are new to the contractor introduce an unusual element of risk and this risk is always included in his price. Such requirements as high technology visual systems for TS, particularly computer generated imagery; unusual or state-of-the-art instructional systems, etc., are significant cost impact items.

b. Changing design requirements. If the specifications are changed frequently by ECP or frequent CCPs are necessary, these become significant factors affecting development costs of ATE and TS software. These include system operational changes causing TS design modifications or the downstream determination that an ATE system does not have sufficient test elements and associated software drivers to fulfill all test requirements (possibly due to insufficient UUT test points).

The following checklist has been prepared as a guide to factors which should be considered since they impact TS and ATE software costs.

- Complexity of system
- Level of testing required
- Size of memory
- Length of total system development
- Size of executive/supervisor
- Number of interrupts
- Degree of human intervention
- Use of fixed/floating point
- Hardware deficiency
- Assembler/compiler efficiency
- Availability of debugging tools
- Skill of system people
- Size of modules
- Size of data base
- Configuration control method priority
- Target machine availability
- Target machine complexity
- Input/Output complexity
- Target machine mean time between failures
- Support equipment cost
- Parallel development costs
- Required number of iterations per second
- Throughput requirements
- Math processing requirements

- Documentation
- Number of revisions
- Number of ECPs
- Number of studies
- Designed flexibility
- Designed portability
- Work length
- Number registers
- Input/Output speed
- Degree of distribution
- Bootstrap requirements
- Memory volatility
- Multi processing requirements
- Slave/master relationships
- Data conversion requirement
- Processor interface complexity
- New system (as opposed to a modification to an existing one)
- Non-proven hardware/software
- You are required to modify someone else's programs
- Analysts have not worked on a similar application
- Designers have not worked on a similar application
- Programmers have not worked on a similar application
- Managers have not worked on a similar application
- Programers must be trained in a new coding language
- Programmers must be trained on a new computer
- You cannot make use of a proven existing operating system or input-output package
- You must provide your own support programs
- Project personnel not familiar with USAF standards, conventions, and documentation requirements
- Vague job requirements
- Multiple using commands involved
- Many policies to change
- Long system life required
- Long development cycle
- Inexperienced using command personnel
- Multiple geographic locations
- You must share computer time with other projects
- You do not have complete control of computer or keypunch resources
- User has control of computer or keypunch resources
- User will supply data base
- User will supply test data
- Data base is classified
- You must test on a computer not exactly the same as the eventual operational computer
- Your effort is split among several locations
- Computer turnaround time is greater than 2 hours

- Computer turnaround time is unpredictable
- Your designers are not doing the programming
- Your designers are not expert programmers
- Your confidence in personnel continuity is low
- You have little or no choice of personnel who work for you
- User is inexperienced in ATE or TS
- User is inexperienced in imbedded computer systems
- You expect much change
- The working environment promises many interruptions
- Incorporation of revisions to manufacturer's software program
- Slow and incorrect stenographic support
- Unavailability of required review or coordinating personnel
- Unreliable assembler or compiler
- System is real time
- Interfaces with other systems are ill-defined or complex
- The system is larger than those the contractor or user have usually worked on
- Data Base is complex or not yet defined
- Computer storage is severely limited
- Input-output is limited in terms of speed, channels, or storage capacity

- The system has a large number of functions
- Innovation required
- Programming language not high level
- Maximum program efficiency required
- High volume of data

3.4 SOFTWARE COST REPORTING AND REGULATIONS, SPECIFICATIONS AND STANDARDS (RSS)

This subsection and the two in Section 6 deal with finance in systems acquisition as compared to the other types of base finance policies and procedures. Included is an overview of how finances are handled in systems acquisition projects. Section 6 covers the "what", showing the reports that are required. The reader is given the basics of how the cost process works, but by no means enough details to make one an expert. Sources of constraints to system acquisition and information on costs and scheduling will be evident in the following paragraphs.

3.4.1 Documentation Overview

Military development programs are increasingly constrained by money. Consider Figures 3.4-1 and 3.4-2. While the DOD budget is diminishing, pay escalates leaving a progressively smaller amount for development of new weapon systems. All decisions, whether to start, continue, or stop a project, depend on the money spent, the money presently available, or the money hoped for in the future. Where this money is spent is determined by Congressional appropriations. Acutely aware of this problem, the DOD looks very closely at two areas in the systems acquisition process: cost and schedule. DOD Directive 7000.1, "Resource Management Systems of DOD" dated 22 August 1966, requires performance measurement. DOD Directive 5000.1, "Acquisition of Major Defense Systems"

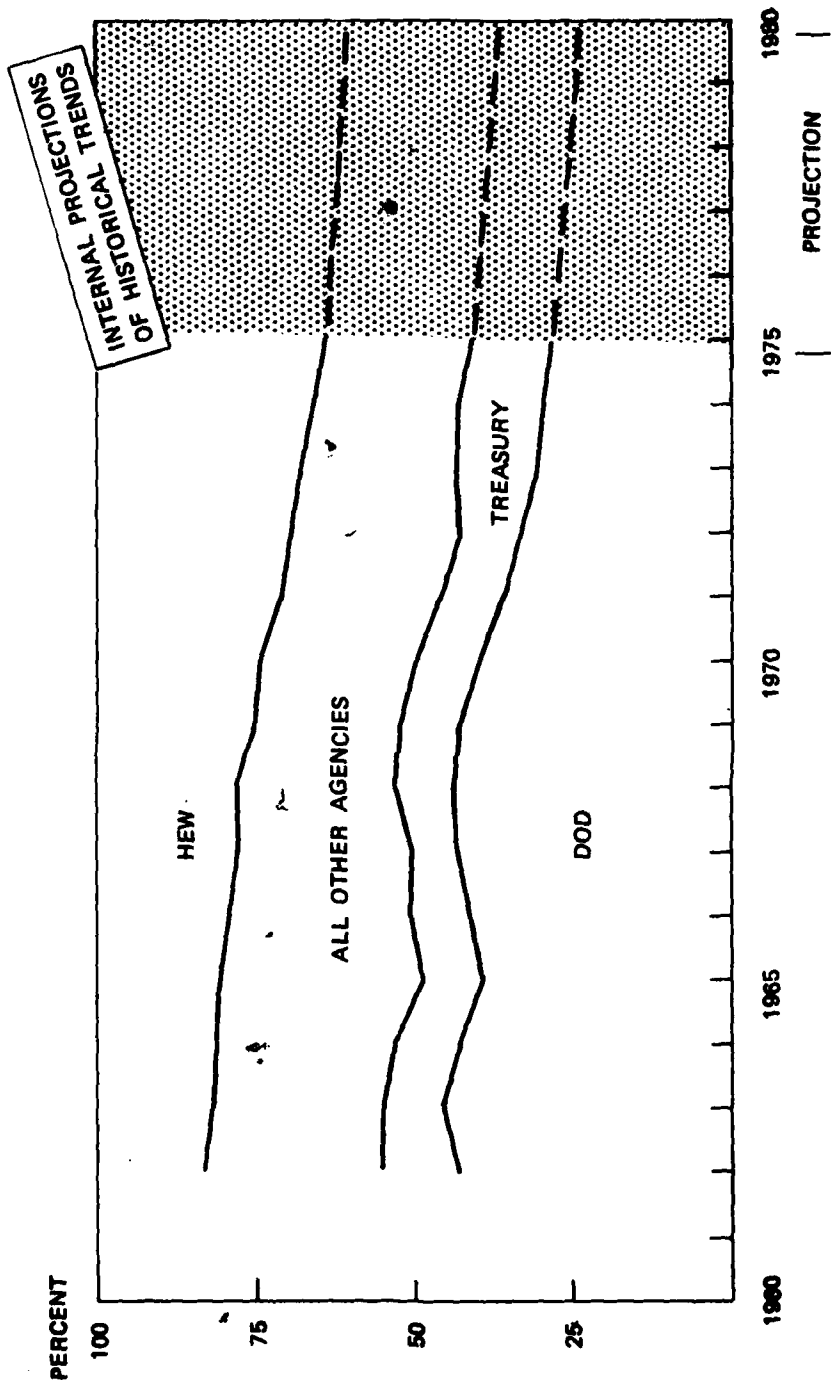


Figure 3.4-1 Distribution of Federal Spending

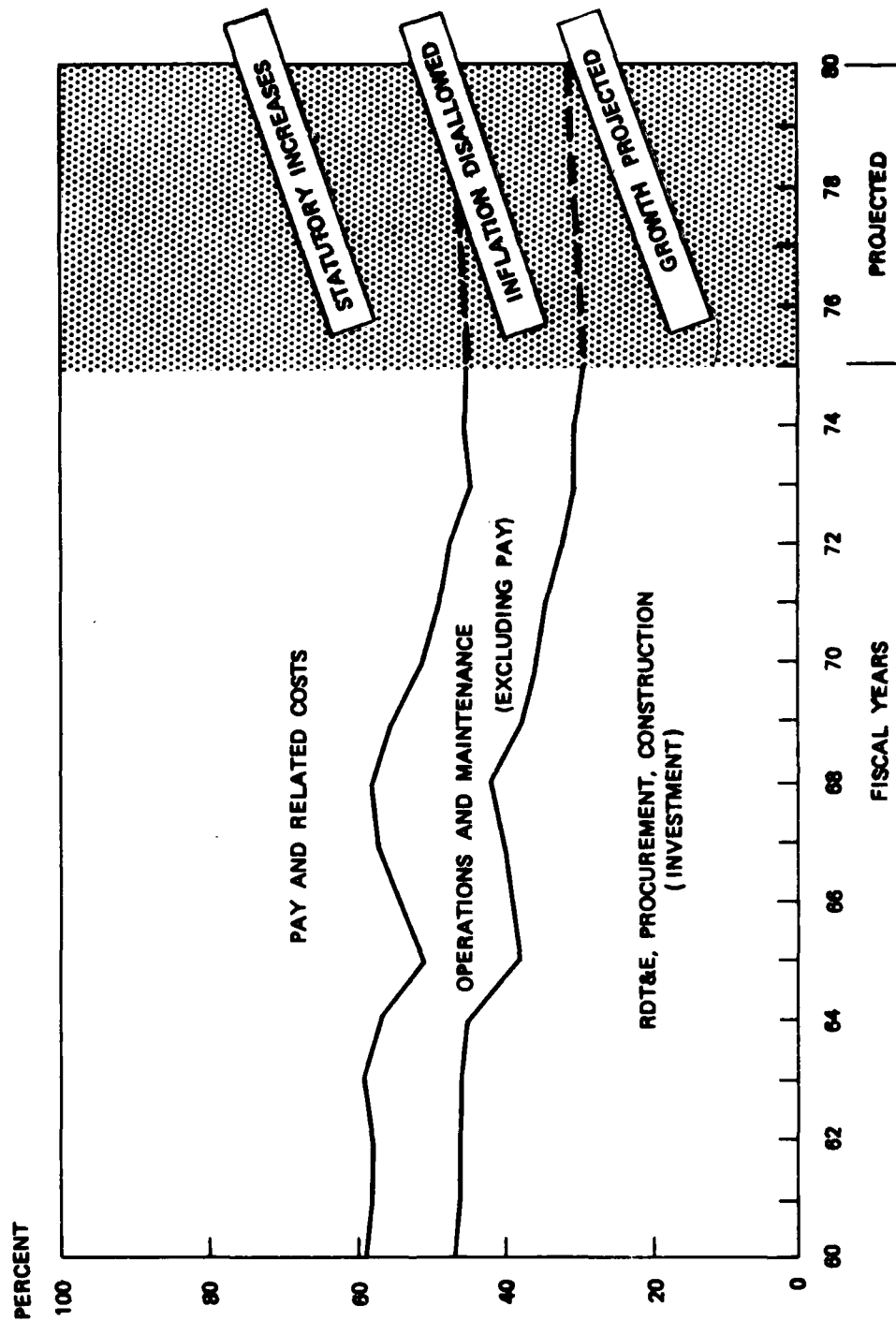


Figure 3.4-2 Distribution of Defense Spending

dated 18 January 1977 (revised), covers management information and program control. DOD Directive 5000.2, "Major System Acquisition Process" dated 18 January 1977 (revised), covers acquisition. These three basic directives are implemented by DOD Instruction 7000.2, "Performance Measurement for Selected Acquisitions" dated 10 June 1977 (revised).

Instruction 7000.2 specifically requires the use of what is called Cost/Schedule Control Systems Criteria (C/SCSC)..." in selected major systems acquisitions programs under all types of contracts ... normally exceeding \$75 million RDT&E or \$300 million production...except firm-fixed-price and firm-fixed-price-with-economic-price adjustment ... The objective of C/SCSC is to provide contractor cost/schedule systems that provide an adequate data basis for responsible decision making by both contractor management and DOD components ...Accordingly, contractors' internal management control systems must provide data which indicates work progress; which properly relates cost, schedule, and technical accomplishment; which are valid, timely, and auditable; and which supply DOD managers with information at a practical level of summarization..." (7000.2). It does not require changes of the contractors' cost/schedule management system except to meet any equitable contract cost distribution C/SCSC requirements, or any Cost Accounting Standards Board procedures that are not presently being met.

USAF inputs to MIL-STD-881A "Work Break-down Structures for defense Material Items", dated 25 April 1975, and to the Armed Forces Procurement Regulation (ASPR) have further defined the instructions for C/SCSC of 7000.2. The ASPR is particularly important because this regulation is the basis from which government contracts are formed. See Figure 3.4-3 which is one of the ASPR provisions relative to C/SCSC.

Air Force Systems Command Pamphlet 173-5 precisely defines the criteria

against which a contractor's cost/schedule control system must be evaluated. There seems to be as many acronyms for C/SCSC (i.e., C/S-Square, C-Square, CSPEC) as there are presently used versions of C/SCSC. Rather than reviewing AFSCP/AFLCP 173-5, a typical aerospace contractor's implementation system, hereafter referred to as an Integrated Management System (IMS), will be presented for a better understanding of an applied C/SCSC. It would have been formally approved via an Air Force validation letter. This is not to say that the contractor had no way, prior to C/SCSC, of controlling management tools of cost and schedule. But aerospace contractors in general have varied systems covering cost and schedules, ranging from poor to good, with no standardized output form for USAF buyers to analyze in the detail needed.

3.4.2 Contractor's AFSCP/AFLCP 173-5 Implementation System

The final output of the IMS to USAF buyers is contained in Section 6 and the inputs and the inner workings of the IMS are explained in Appendix C. Specifically, the IMS, to integrate cost and schedule data into a management tool, functions around the five major criteria sections of the C/SCSC: (1) organization; (2) scheduling and budgeting; (3) accounting; (4) analysis; and (5) revisions/access to data. These criteria are molded into a working system prior to contract award or start via the RFP which the USAF uses in the contractor selection process.

The IMS is a rather intricate system warranting detailed explanation and is therefore treated at length in Appendix C. The material in Appendix C is organized according to the five criteria identified above and a thorough reading is recommended.

3.4.3 ATE and TS Considerations

Now that you have seen the flow-down of financial RSS concerning major acquisitions from DOD through USAF to the

7-104.87 Cost/Schedule Control Systems. In accordance with 1-331(h), insert the following clause:

COST/SCHEDULE CONTROL SYSTEMS (1973 APR)

(a) The Contractor shall establish, maintain and use in the performance of this contract Cost/Schedule Control Systems meeting the attached criteria (DODI 7000.2 Performance Measurement for Selected Acquisitions). Prior to acceptance by the Contracting Officer and within ninety* (90) (*or as otherwise agreed to by the parties) calendar days after contract award, the Contractor shall be prepared to demonstrate the operation of his systems to the Government to verify that the proposed systems meet the established criteria set forth above. As a part of the demonstration, review and acceptance procedure, the Contractor shall furnish the Government a description of the Cost/Schedule Control Systems applicable to this contract in such form and detail as indicated by the AFSCP/AFLCP 173-5, AMCP 37-5, NAVMAT P-5240 Cost Schedule Control Systems Criteria Joint Implementation Guide hereinafter referred to as the guide, or required by the Contracting Officer. The Contractor agrees to provide access to all pertinent records, data and plans as requested by representatives of the Government for the conduct of the review.

(b) The description of the management systems accepted by the Contracting Officer, identified by title and date, shall be referenced in the contract. Such systems shall be maintained and used by the Contractor in the performance of this contract.

(c) Contractor changes to the accepted systems shall be submitted to the Contracting Officer for review and approval. The Contracting Officer shall advise the Contractor of the acceptability of such changes within sixty (60) days after receipt from the Contractor. When systems existing at time of contract award do not comply with the criteria, adjustments necessary to assure compliance will be effected at no change in contract price or fee.

(d) The Contractor agrees to provide access to all pertinent records and data requested by the Contracting Officer or his duly authorized representative for the purpose of permitting Government surveillance to insure continuing application of the accepted systems to this contract. Deviations from accepted systems discovered during contract performance shall be corrected as directed by the Contracting Officer.

(e) The Contractor shall require that each selected subcontractor, as mutually agreed to between the Government and the Contractor and as set forth in the schedule of this contract, shall meet the Cost/Schedule Control Systems criteria as set forth in the guide and shall incorporate in all such subcontracts adequate provisions for demonstration, review, acceptance and surveillance of subcontractors' systems, to be carried out by the Government when requested by either the prime or subcontractor.

(f) If the Contractor or subcontractor is utilizing Cost/Schedule Control Systems which have been previously accepted, or is operating such systems under a current Memorandum of Understanding, the Contracting Officer may waive all or part of the provisions hereof concerning demonstration and review.

(End of clause)

Figure 3.4-3 ASPR Statement Relative to C/SCSC

contractor, and the contractor's implementation of these RSS, four more areas need be covered.

First, whether or not the acquisition of ATE and TS software would be a major acquisition, does not matter. Your exposure to C/SCSC lends itself very well to understanding acquisition cost and schedule management for other than major acquisitions. A system called the Cost/Schedule Status Report (C/SSR) is applicable to contracts of \$2 million or over and of more than one year duration, and for which a CPR is not a requirement. C/SSR is almost a mini-C/SCSC which is less costly to administer and less burdensome data-wise because performance measurement is conducted at higher WBS and organizational levels. However, the basic type of data used is still the same (Budgeted Cost of Work Scheduled (BCWS), Budgeted Cost of Work Performed (BCWP), Actual Cost of Work Performed (ACWP), Estimate At Completion, etc.).

Second, ATE and TS cost data can be separated to a large extent from overall system costs by using the C/SCSC WBS and functional breakdown structure. The Cost Performance Report (CPR), which is the end result of C/SCSC, can have its supporting computer data sorted for every possible way that anyone would ever need or want to analyze cost breakdowns. Should the normal data supplied in the CPR or C/SSR not suffice for your needs, it means that the Contract Data Requirements List (CDRL) attached to the contract, as well as listed on the SOW and the Data Item Description (DID), did not fully state your desires.

Third, every RFP from USAF must list whether C/SCSC, C/SSR, or some other control system is to be employed on the contract. The RFP also includes the SOW, DID, and CDRL.

Fourth, there is a distinction between price and cost. The only way of looking at these terms is that the price the government or customer pays the contractor equals the costs incurred plus the

profit fee for doing the work. Cost does not include fee.

3.5 UNIQUE CONSIDERATIONS FOR ATE

The critical ingredient with regard to ATE is the cost of developing and maintaining the test programs. The cost examples given at the end of paragraph 4.3, for Line Replaceable Units (LRU) of various complexity, indicates the high costs associated with the development of just one test program; and an ATE station can have several hundred test programs.

3.5.1 Trade Consideration

The first cost consideration in a new weapon system are the trades to determine the amount of general purpose test equipment, special purpose support equipment, and ATE required to support the weapon system maintenance. The prime consideration usually becomes: to what extent will the cost of test program development vs. operational returns in manpower, allow the use of ATE over manual test methods, i.e., test software development costs now vs. manual test costs later. Many such trades are based on MIL-STD-1513, (USAF) Trade Studies for the Selection of Avionic Test Support Systems, Criteria for, 15 January 1971. From these studies the determination of items such as; what special testing requirements exist, how much ATE control is wanted, to what extent the software should self-document, and which HOL should be used; should be determined.

3.5.2 Time Consideration

The next consideration that must take place is the time at which an ATE system is to be introduced into the Air Force inventory. Ideally the ATE system, in conjunction with available test equipment, would be developed very early in the program and be utilized in the UUT supplier's factory for new equipment acceptance tests. This gives the double benefit of requiring only one configuration ATE system that is utilized both in the factory and in the field; plus it

would reduce the operational software maintenance because of the previous factory utilization. The trade, however, is that this concept requires development of the test software very early in the weapon system program, when the UUTs are still under development. This in turn may provide increased ATE test software development costs, due to the inherent variations in UUT design and configuration in the weapon system development phase.

Also, the nature of weapon system contracting can interfere with efforts to develop ATE test software early in the program. Weapon system contractors normally have a Design Development Test and Evaluation (DDT&E) contract before the system is committed to production, and this contract usually involves delivery of a relatively small number of units. ATE hardware and software justification typically requires a production quantity of delivered equipment to be cost effective when compared against manual or semi-automatic testing. The normal method is to implement ATE later in a program, using special and general purpose test equipment in the early program stages.

Consideration must be given to the efficiency of the ATE control and support software, because of its down-stream impact on the test software. Trade studies during the ATE software evaluation (analysis) phase, should be undertaken to determine if the possibly more costly compilers (development time cost) may return the investment over the increased run time (operational test time) of an interpreter. Additionally, and in association with the compiler/interpreter study, the execution (run) and development time for the desired standardized HOL such as Abbreviated Test Language for all Systems (ATLAS) routine should be compared against other test languages in selecting the ATE software. Other cost influencing factors requiring particular attention by ATE acquisition engineers, arise as a result of normal contractor-contractee relationships.

As is the case with other contractor furnished software, if ATE software is not well defined before it is committed to development, the cost will be adversely affected. If the ATE engineer is not clear on precisely what will be delivered by the customer, or if precise delivery requirements are not fully specified, it is doubtful that a satisfactory ATE software system will be delivered. Further, if the ATE acquisition engineer is unclear on what should be delivered, or what tasks need to be done, it is doubtful that an effective procurement will result. Other guidebooks in this series may be consulted on such matters as specification of requirements, deliverable documentation and contracting.

3.5.3 Documentation Required

In particular, a significant portion of the costs associated with ATE software development and maintenance is the documentation procured with it. Misunderstanding can be avoided by requiring that potential contractor's, in their offering, breakdown software costs into the various elements. These elements should identify as separate cost items such activities as preparation of part II specifications, program coding, check-out, integration testing, program validation and verification, etc. This effort has the added advantage of providing a means whereby competing contractors can be compared on their knowledge of the job. If, for example, one proposal indicates a segment of cost which is disproportional to the total, this reflects directly on the proposers understanding, or lack of understanding, of the job. It also helps to identify "gold plating" on the part of the contractor.

However, it should be noted that there is very little standardization in methods of developing ATE software. Some contractors may employ the use of special facilities, such as a general or special purpose system integration laboratory, while other contractors use other schemes. Such things directly

affect such cost elements as ATE support and control software development and checkout.

3.5.4 Additional Considerations

Improvements should be made in methods of accounting for software costs. Cost data collection and analysis and tracking of actual cost vs. estimated cost is not normally done for, and made available to, the government by its contractors. More meaningful historical data would be extremely helpful in estimating ATE as well as other ground system software costs.

Finally, it is essential for ATE software that a detailed WBS is prepared and WBS cost is collected and reported by the contractor. Further, the ATE acquisition engineer should influence the contractor to prepare WBS information in one for one correspondence with the specification tree.

3.6 UNIQUE CONSIDERATIONS FOR TRAINING SIMULATORS

Several factors, unique to TS software, can affect cost in addition to those previously considered.

3.6.1 Frequent or Unnecessary Changes

As previously indicated, frequent or unnecessary changes via ECP and CCP are significant cost impact items. It should be noted that poorly defined requirements or poor planning leads either to excessive change or development of TS software which fails to meet user requirements. Particular attention should be devoted to guidelines indicated in the Requirements Specification guidebook. Clear definition of requirements, avoidance of requirements excessive to real user needs and careful planning, including consideration of all reasonable design alternatives before the RFP is released, will minimize this problem in most cases.

3.6.2 Incremental Procurement

TS which are procured incrementally give rise to particularly difficult software development and maintenance. For example, the A-10 Fighter/Attack simulator is being initially procured without a visual system. The visual system, procured under separate contract, will likely be provided by a different contractor. Consequently, the visual system contractor must integrate his visual system with another contractor's simulator providing computer-to-computer interface and providing modifications to the instructor/operators console and its associated software. The visual system contractor, therefore, works to the added difficulty of interfacing, and undoubtedly modifying, software developed by someone else.

3.6.3 Concurrent Procurement

It is particularly difficult to develop and then hold constant TS requirements for training systems procured concurrently with new weapon systems. Detailed design data for a new weapon system and specific configurational information change frequently during the Research Development Test and Evaluation (RDT&E) phase of a new weapon system. This makes it especially difficult to design a training simulator, and its software in particular, since each change in weapon system configuration may change the TS configuration.

3.6.4 Evolutionary Change

Airplane systems undergo evolutionary product improvement. The engine manufacturer may find a way to increase the performance of his product after the airplane has been placed in the operational inventory. New air-to-air or air-to-ground weapons may become available for the airplane, altering airplane performance. These things make it necessary to modify the simulator in order to keep current with system configuration. It is particularly difficult for a simulator contractor to keep up with changes implemented by the airframe manufacturer.

3.6.5 Inadequate Preparation

Unique development and maintenance problems of the type previously indicated are a result of the very nature of TS. These problems cannot be eliminated, but experience has shown their impact can be minimized by insistence on the part of the TS acquisition engineer of adherence to strict discipline in each phase of the acquisition. He should ensure complete and adequate specification before proceeding on any change. He must provide for inter-contractor communication, including classified design data. He must also carefully plan TS changes and manage his contractors to ensure they are proceeding in accordance with these plans. Above all, he should provide the contractor/contractee environment conducive to rigorous software disciplines including adequate documentation, thorough testing, and defining completely all software before proceeding to coding and checkout.

3.6.6 Use of Existing Software

Software developments costs for training simulators tend to be much less than for most newly-developed real time software. This is because of substantial repeatability in modules between different system simulators. While TS software is rarely available "off-the-shelf", many of the components and modules can be quickly adapted to satisfy new simulator

applications. Also, there are very few contractors for TS systems so adaptation of modules from system to system can usually be accomplished within a contractor's organization.

3.6.7 Additional Considerations

Some of the frequent problems associated with TS software cost estimating include:

a. Difficulty in separating hardware development cost from software development cost. The two are closely interrelated in TS systems.

b. Contractors tend to underestimate the cost of documentation and consequently the user is often delivered poor documentation. (This has serious impact on software maintenance.)

c. Detailed cost visibility is generally not available to the procurement agency. Additional detail in cost reporting, comparable to that available internally to contractor management, is desirable.

d. Cost estimates are often based on grossly-defined TS system requirements. More specific functional requirements should improve completeness and accuracy of estimates. Also, application of methods such as Wolverton's model should use factors that are specific to TS-type software.

Section 4.0 SOFTWARE DEVELOPMENT COSTS

This section is concerned with the costs associated with each development phase of producing TS and ATE software. There are three software packages that are of interest: training simulator operational and support, ATE operational and support, and ATE test software. TS software consists primarily of the weapon system simulation software modules, the associated control software, and support software for software maintenance, all of which are developed in conjunction with TS hardware. The ATE operational and support programs are developed with the ATE hardware system, whereas the ATE test software packages are individually developed for each UUT - usually after delivery of the ATE test station. Although the ATE software units are time phased, the same basic development sequence exists for all ATE and TS software packages.

The software development sequence involves four principle phases:

- a. Analysis (process of deriving and specifying requirements, software development planning, and implementation concept generation)
- b. Software Design (initial detail design)
- c. Coding and Checkout (software code generation with syntax correction and simulation runs)
- d. Integration and tests (operational de-bug and subsequent validation)

Section 4 is organized under these principal tasks with Analysis described in paragraph 4.1, Software Design in 4.2, Coding and Checkout in 4.3 and Integration and Testing, in 4.4. Paragraph 4.5 discusses the total of all costs, including considerations outside the four defined development phases.

It is important to note, that for both TS and ATE, the software and hardware development efforts are interdependent and cannot be accomplished independently. Figures 4.0-1 and 4.0-2 depict the development process of TS and ATE software including the associated required hardware support. The "Phase" portion of Figures 4.0-1 and 4.0-2 indicates a more detailed example of the tasks being accomplished. The "documentation" portion indicates some of the documents produced during each phase (a complete description of documentation in each phase is provided in the Computer Program Documentation Requirements guidebook). The "responsibilities" portion provides an indication of efforts during each phase. It should be noted that the activities reflected in these figures do not normally coincide with the equivalent weapon system development activities.

Although cost estimating varies greatly from organization-to-organization; a similarity often exists in that development estimates are normally accomplished in a three group function: analysis and design, code and checkout, and test and integration. A "rule-of-thumb" that The Boeing Company has applied on past ATE and TS programs allocates 40 percent cost to analysis and design, 20 percent to code and checkout, and 40 percent to test and integration. These costs estimates apply only to the contractor or sub-contractor portions and not to the initial requirement generation. Table 4.0-1 provides a feasible breakdown of the distribution of manhours that could be applied to the 40-20-40 estimate for a "typical" 100 instruction ATE or TS software module assuming an overall average work rate of four instructions per man day. It is emphasized that the effort allocation of Table 4.0-1 is only an example and is provided to indicate a typical cost distribution that could apply.

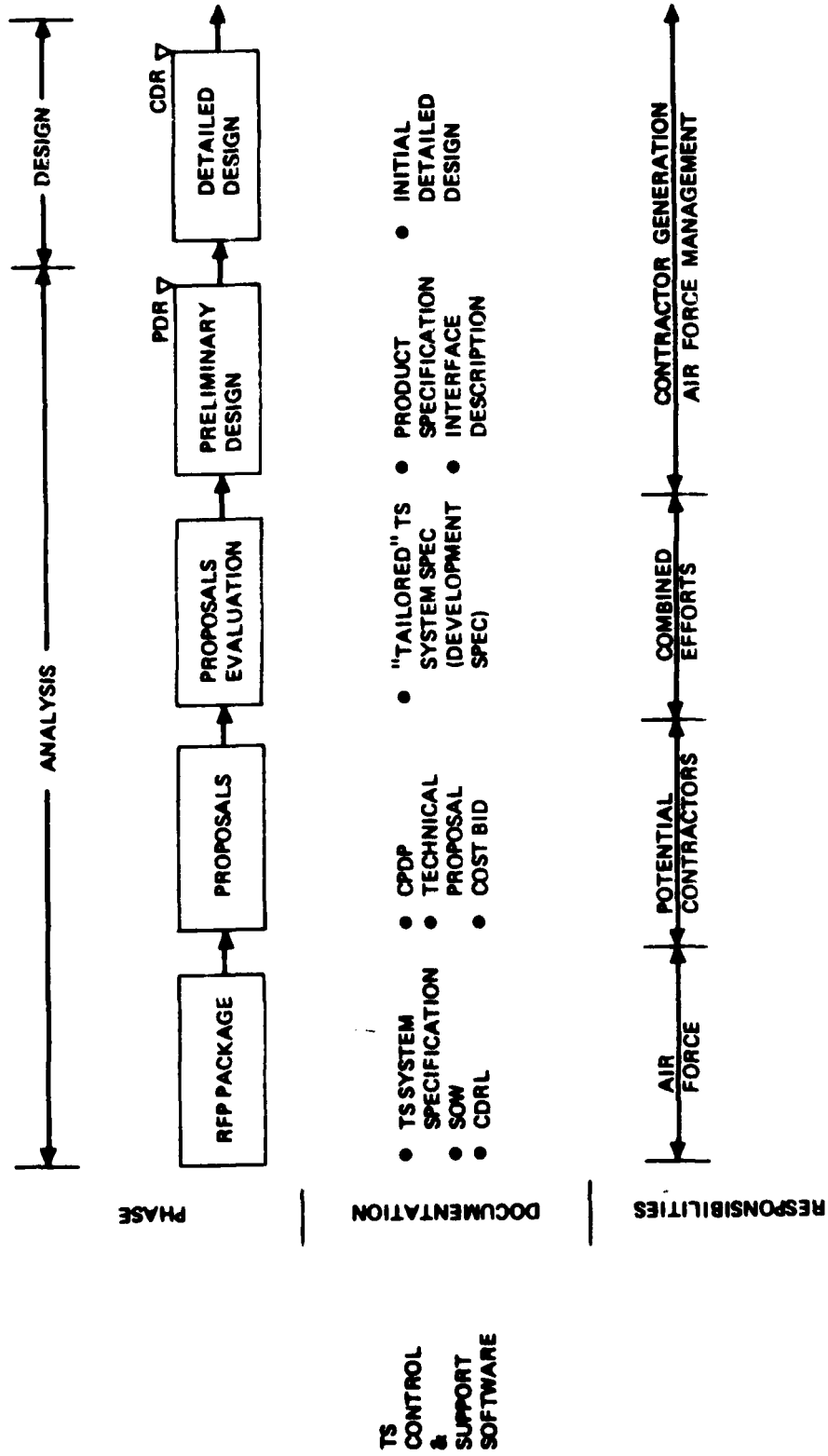


Figure 4.0-1. Trainer Simulator Software Development Process (Sheet 1 of 2)

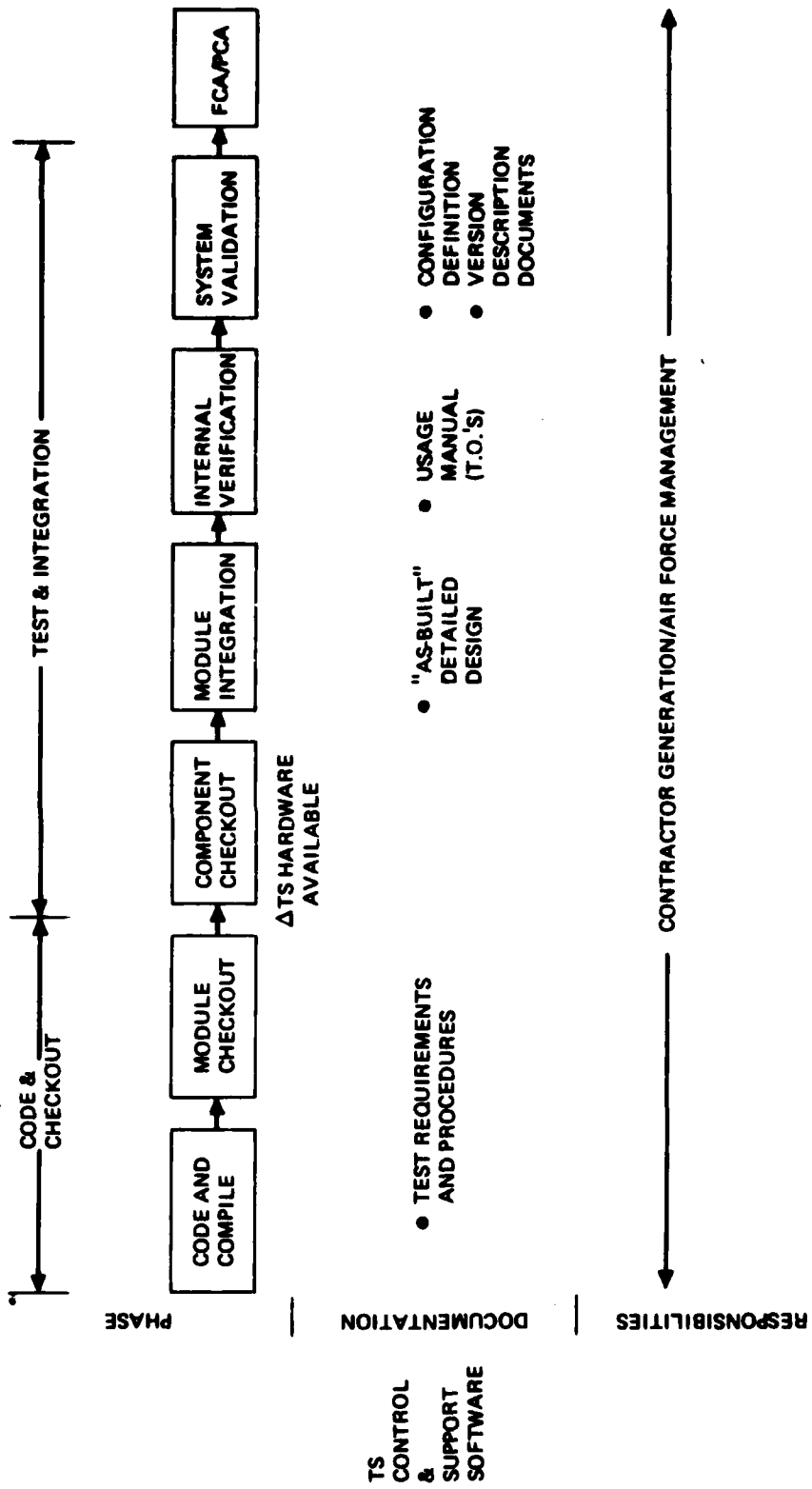


Figure 4.0-1. Trainer Simulator Software Development Process (Sheet 2 of 2)

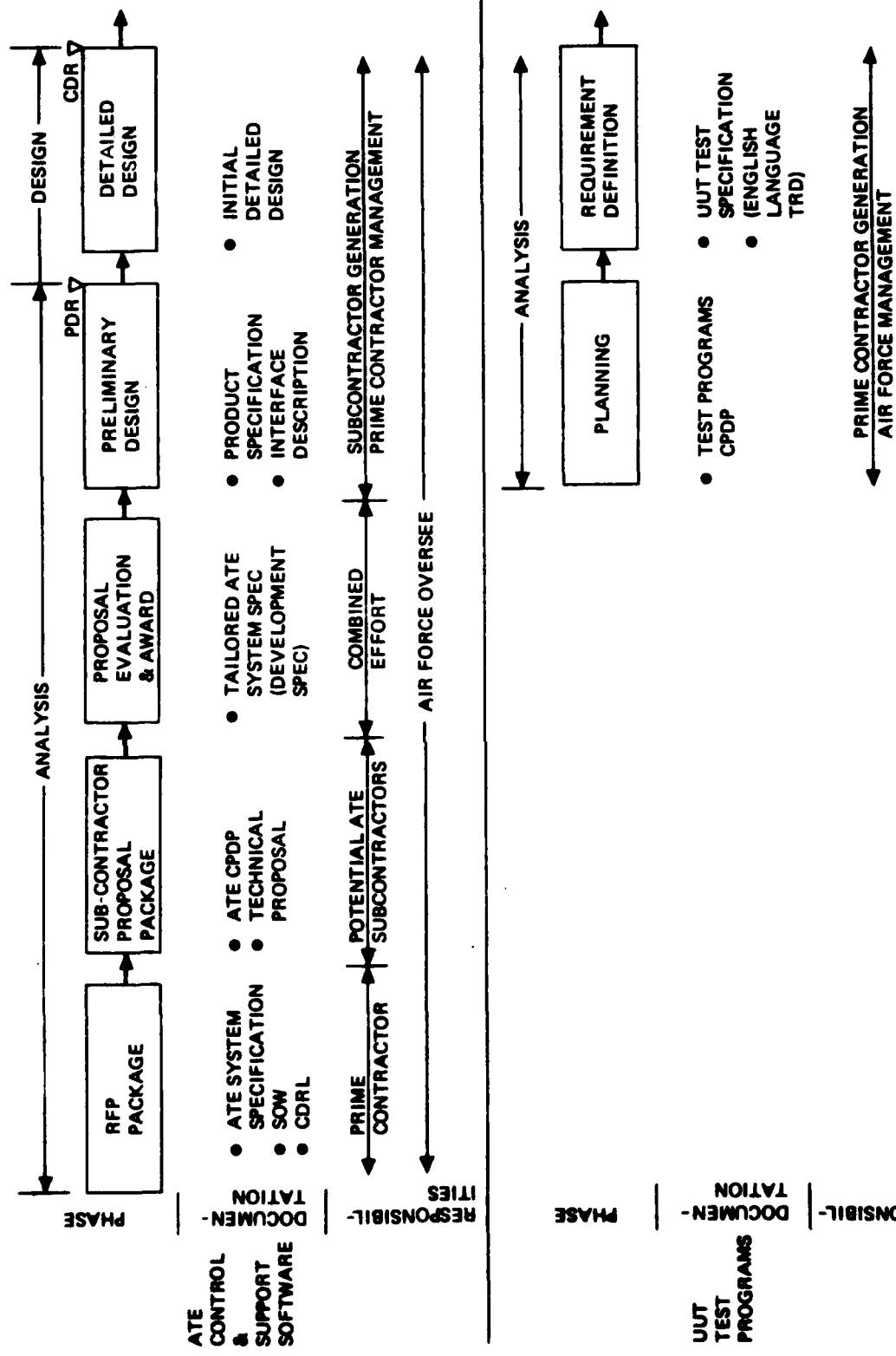


Figure 4.0-2. Automatic Test Equipment Software Development Process (Sheet 1 of 3)

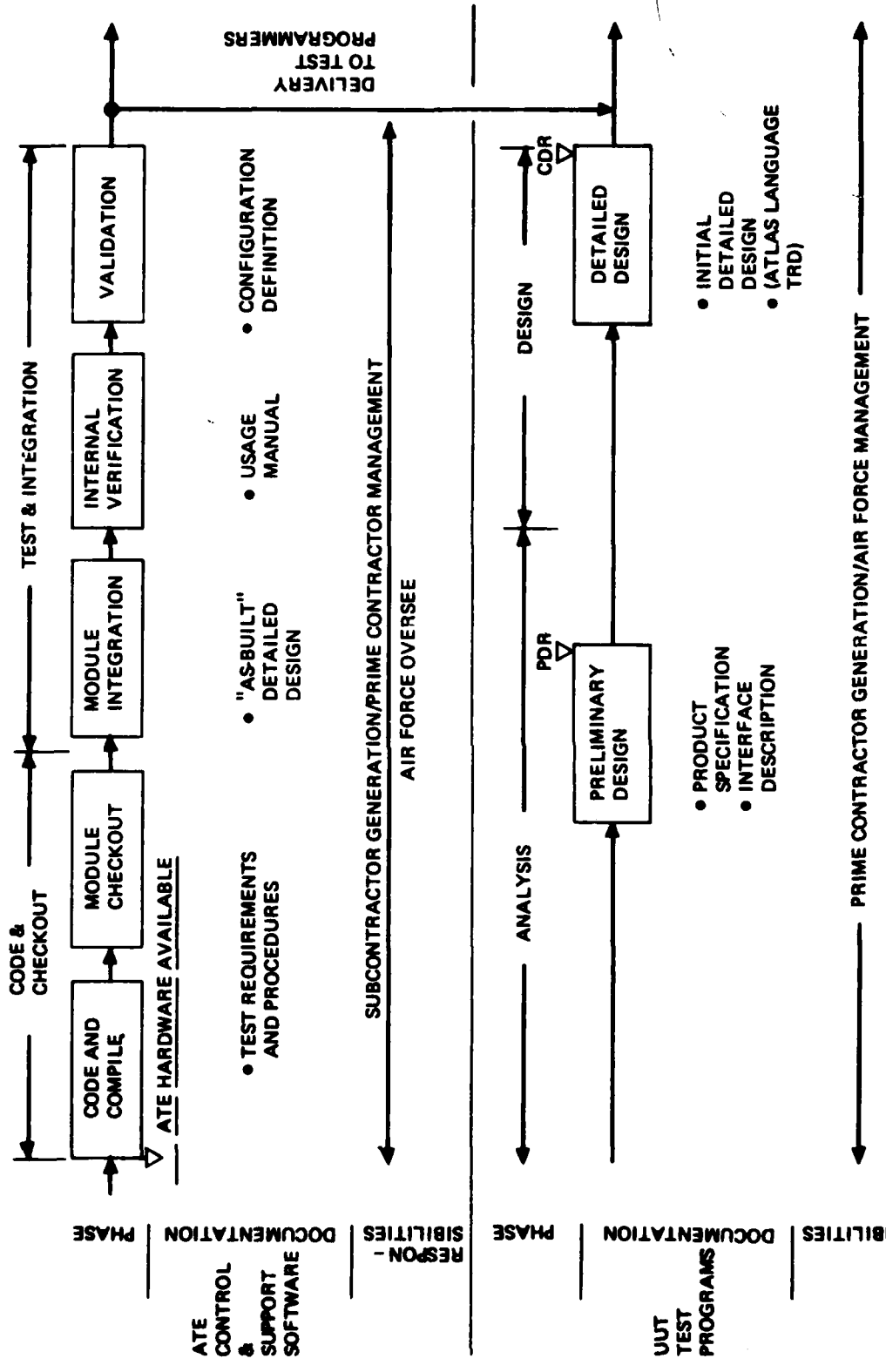


Figure 4.0-2. Automatic Test Equipment Software Development Process (Sheet 2 of 3)

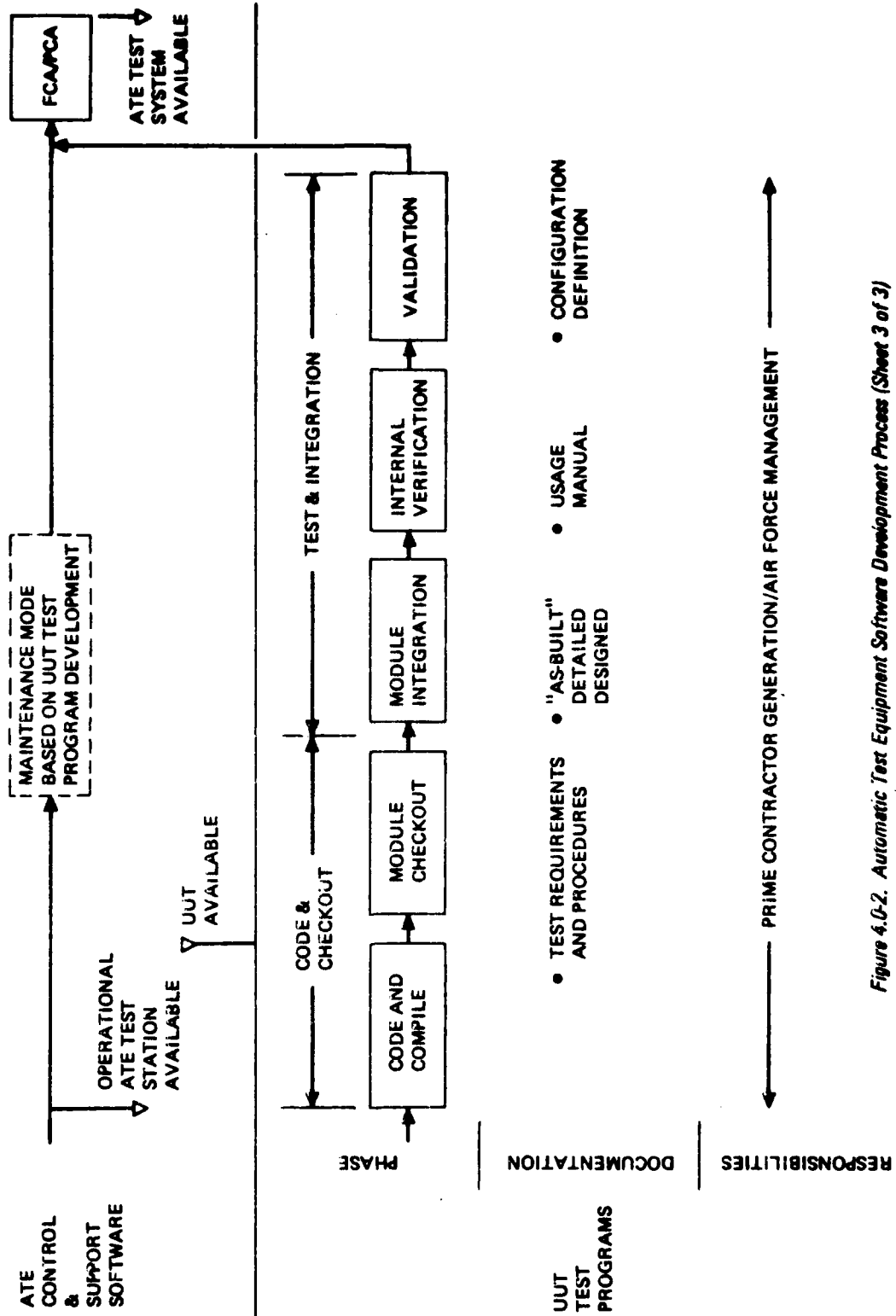


Figure 4.0-2. Automatic Test Equipment Software Development Process (Sheet 3 of 3)

Table 4.0-1. Typical 100 Instruction Software Module Development Manhour Estimation

Major Phase	Percent of Total		Activity	Percent of Phase		Effort (in Manhours) For Nominal Module (100 Instructions) at Nominal Rate (4 Instructions/Manday)		
	Value	Cum		Value	Cum	Value	Activity Cum	Phase Cum
Analysis & Design	40	40	Functional Specification	10	10	8	8	
			Design and Performance Specification	25	31	20.	28	
			Detailed Module Design	50	81	40.	68	
			Test & Development Plan	15	100	12	80	80
Code and Debug	20	60	Code Modules	40	40	16.	16.	
			Test Procedures	20	60	8.	24.	
			Module Debug and Verification	40	100	16.	40.	120
Integrate and Test	40	100	Validation Procedures and Analysis	30	30	24.	24.	
			Test Operations	25	55	20.	44.	
			Problem Resolution	35	90	28.	72.	
			Validation Report	10	100	8.	80.	200

NOTE: Above estimates do not include overhead and support costs such as system engineering, subcontract management, supervision, internal reports, travel or special documentation.

Since interaction with computer hardware, system tests, documentation, and operational equipment exist in various development phases, a major question can arise relative to exactly what constitutes software costing alone. This question is addressed in the following paragraphs of this section. However, each organization distributes cost allocations according to its own accounting methods, therefore no attempt is made to specify a preferred cost allocation method.

Numerous attempts have been made to provide formulas for software estimating, but often when an estimate is produced it is based on past experiences of the individuals involved and upon the specific systems with which they were knowledgeable. The estimates, thereby, are a judgemental delta from previous known or existing software package components. Both potential cost models and methods are presented in Section 3.

4.1 ANALYSIS

ATE and TS software development is initiated during a weapon system's development phase. The present Air Force procurement method for TS typically consists of receipt of a Required Operational Capability (ROC) document, development of a RFP for industry bid, and subsequent contractor selection. This process is described in detail in Section 3 of the Requirement Specification guidebook and is accomplished by the Simulator System Program Office (SPO) on a competitive bid process rather than purchasing directly from the Weapon System prime contractor. ATE, however, is normally procured via the Weapon System Contractor by means of a contract supplement (modification) or as a separate procurement by the support equipment SPO. Section 4 of the Requirement Specification guidebook provides a detailed description of the process involved in generation of the ATE software requirements.

Costs associated with the analysis phase are borne by the procuring agency until a contractor or sub-contractor is given a procurement contract and preliminary design is undertaken. The contractor's cost of preparing a proposal in response to the RFP, and his subsequent cost of involvement in fact finding (source survey) and negotiations is also at the contractor's expense. This process is considered new business risk capital with the TS and ATE suppliers allocating a portion of their profit return from existing business for this purpose.

Estimates that are accomplished for initial pre-RFP release are usually required for life cycle cost studies and to indicate the magnitude of cost estimates anticipated in response to an RFP. These estimates are accomplished similar to the methods used in preparing a firm cost estimate: the judgement and experience of the estimator based on past program development. It is impossible to estimate, with any high degree of accuracy during the first phase of the development (requirements definition, proposal evaluation and the preliminary design phase), what the costs and schedule for a project will be. Only the most experienced analysts, from either industry or the Air Force, should make estimates at this early stage. One approach is having several persons make estimates and then computing an average estimate.

Prior to preliminary design, the estimate is based on the recommended approach; and at the end of preliminary design it is based on the recommended solution for that approach. Contractors are required to provide firm cost bids in RFP replies and have the experienced personnel required to accomplish detailed software cost estimates prior to the preliminary design task.

The tasks for initial estimating of the total software costs are to (1) list the phases of development, (2) identify the major resources required in each phase, and (3) estimate the approximate cost of those resources (this is the top-down-

estimating process of paragraph 3.2.1.1). Table 4.0-2 provides an illustration of the gross level of detail of an initial estimate.

Contractor's responses to the RFP will provide cost estimates based on articles of the SOW. From the SOW, the contractor identifies the software modules necessary to fulfill the requirements and arranges the necessary tasks into a WBS. An example of the software modules constituting a typical ATE system is shown in paragraph 4.1 of the Requirements Specification guidebook. Paragraph 3.1 of that same guidebook provides a software breakdown description for a TS system. Each task identified in the WBS is then estimated and collected to give an approximation of its total cost. The key to this effort is to obtain a previously completed similar system to use for the data base. From this data base comparison software modules are located to whatever extent possible. For those modules where a good analogy is made, a ratio between module size and complexity is made and the costing data is proportioned according to the ratios to obtain the required approximation. For those modules where a poorer match exists, the cost approximation must be made via a differences analysis where the similar portions are handled by ratio and the different portions are approximated by extrapolation. A discussion of the various methods employed is provided in paragraph 3.2 of this guidebook.

Figure 4.1-1 is an example of a typical contractor's engineering estimate form that is utilized in estimating software tasks. From the manpower estimates generated on these type forms, a contractor's finance group converts the manhours into dollars, adds the appropriate overhead, support costs, inflation escalators and anticipated profit to produce a firm bid. The estimator generates supplementary task definition, estimate rationale, and calculation sheets as steps toward generation of the manpower estimates. These detailed back-up sheets are normally available to Air Force engineering personnel who are members of the

evaluation team. A review of this data, possible during the fact-finding or source survey visits, can give an excellent insight into a contractor's grasp of the understanding of requirement fulfillment. The data contained therein also provides insight to those areas where possible discussion is necessary during contract negotiations, both towards reducing costs to the Air Force and to insure sufficient and correct effort is being allocated to each task. A specific area that should be reviewed carefully are the tasks which are modifications to existing software packages. Since both ATE and TS contractors bid evolutionary rather than revolutionary systems, many of the software modules will be "very nearly off-the-shelf" particularly for ATE systems. An examination of the task associated with developing modules which meet requirements, provides an excellent insight into the capabilities of the candidate contractor.

Planning and proposal package generation is accomplished on a combined cost and schedule basis. Since initial system requirements determine major task completion milestones, software completion dates will normally be dictated based on hardware availability dates. The software tasks, once identified at a high level, are broken down into the appropriate time periods to provide a per fiscal year estimate of software costs. A cost related situation that has occurred in past programs is the modification of TS and ATE schedules and/or concepts to accommodate budget constraints. Since both TS and ATE are not mandatory for most weapon systems prototype development, a natural tendency exists to cut costs by delaying development of the TS and ATE systems. Although it normally can be shown that very significant life cycle savings are realized from TS and ATE, the pay-back associated with these systems is not realized until the system becomes operational, a number of years removed from RFP package generation. As with most purchases in both the business and government world, priorities must be made when funding constraints appear; therefore, making ATE and TS a lower

Table 4.0-2. Initial Software Estimate Example

PHASE	RESOURCE	UNITS	COST	ELAPSED TIME
Requirements Definition & Proposal Evaluation	AF Personnel	2500MH	\$100,000	8 months
	Support Personnel	1500MH	60,000	
	Travel & Expense		20,000	
			<u>\$180,000</u>	
Preliminary Design	AF Personnel	400MH	\$ 16,000	4 months
	Support Personnel	400MH	16,000	
	Contractor Personnel	3500MH	140,000	
	Contractor's Support	1000MH	40,000	
	Computer Facility	20Hrs.	2,000	
	Travel & Expense		2,000	
	Other		5,000	
		<u>\$221,000</u>		
Detail Design	AF Personnel	600MH	\$ 24,000	6 months
	Support Personnel	600MH	24,000	
	Contractor Personnel	6000MH	240,000	
	Contractor Support	2000MH	80,000	
	Computer Facility	40Hrs.	4,000	
	Travel & Expenses		2,000	
	Other		10,000	
		<u>\$384,000</u>		
Code and Checkout	AF Personnel	600MH	\$ 24,000	6 months
	Support Personnel	300MH	12,000	
	Contractor Personnel	4000MH	160,000	
	Contractor Support	1000MH	40,000	
	Computer Facility	500Hrs.	50,000	
	Travel & Expense		2,000	
	Other		5,000	
		<u>\$293,000</u>		
Test & Integration	AF Personnel	1000MH	\$ 40,000	14 months
	Support Personnel	200MH	8,000	
	Contractor Personnel	10,000MH	400,000	
	Contractor Support	2,000MH	80,000	
	Test Facility	1,000Hrs.	100,000	
	Hardware Support		100,000	
	Travel & Expense		5,000	
	Other		15,000	
		<u>\$784,000</u>		
TOTAL PROJECT ESTIMATE			<u>\$1,826,000</u>	<u>38 months</u>

NEW BUSINESS ENGINEERING ESTIMATE FORM

Submitted by _____
 BFE _____
 Organizational _____

Prepared by _____
 Approved by _____
 Date _____

ETC. _____
 E.O. _____

Months After Go-Ahead: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

Year: 19 20 21 22 23 24

Month: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

Day: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

TOTAL

MANPOWER SPREAD

In the spaces below indicate the manmonth (in months) or the manhour requirements by months for the required phases or tasks, and by organization. One manmonth equals 193 man hours.

PROF. ENGR. B 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 16.0 17.0 18.0 19.0 20.0 21.0 22.0 23.0 24.0

SUPERVISION 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 16.0 17.0 18.0 19.0 20.0 21.0 22.0 23.0 24.0

SR. CP. ENGR. A 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 16.0 17.0 18.0 19.0 20.0 21.0 22.0 23.0 24.0

SR. CP. ENGR. B 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 16.0 17.0 18.0 19.0 20.0 21.0 22.0 23.0 24.0

OFFICE EXEMPT 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 16.0 17.0 18.0 19.0 20.0 21.0 22.0 23.0 24.0

TOTAL

LABOR GRADE BREAKDOWN

In the spaces below indicate the manmonth or the manhour (in weeks) requirements for the specified items by labor grade. For general office (technical or non-technical) indicate grade.

PROFESSIONAL OR TECHNICAL 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 16.0 17.0 18.0 19.0 20.0 21.0 22.0 23.0 24.0

GENERAL OFFICE GRADE 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 16.0 17.0 18.0 19.0 20.0 21.0 22.0 23.0 24.0

TOTAL

TASK OR PHASE BY ORGANIZATION

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

TOTAL

COMMENTS AND NOTES

RETURN TO NEW BUSINESS ESTIMATOR:

TEL. EXT.

OTHER REQUIREMENTS

Computer Type _____ Hour _____

Consultants Yes _____ No _____

Developmental Shop Yes _____ No _____

Document Number _____

Material (Eng. Est.) _____

Overtime Yes _____ No _____

Subcontract Yes _____ No _____

TRAVEL REQUIREMENTS

TRIP # FOR TRIP

REASON FOR TRIP

YEAR

TRIP # FOR TRIP

REASON FOR TRIP

YEAR

TRIP # FOR TRIP

REASON FOR TRIP

YEAR

TRIP # FOR TRIP

REASON FOR TRIP

YEAR

TRIP # FOR TRIP

REASON FOR TRIP

YEAR

TRIP # FOR TRIP

REASON FOR TRIP

YEAR

TRIP # FOR TRIP

REASON FOR TRIP

YEAR

TRIP # FOR TRIP

REASON FOR TRIP

YEAR

TRIP # FOR TRIP

REASON FOR TRIP

YEAR

TRIP # FOR TRIP

Figure 4.1-1 Typical Contractor's Engineering Estimating Form

priority for immediate need. Requirement modification occurs most easily early in a program. Major modifications after RFP release can cause a certain level of doubt as to future business in the anticipated contract. Since a potential contract recipient will generally spend company funds on a proposal in a direct relation to the contract's worth, a major RFP modification to cut costs will cause (1) contractor funds to be split between the proposal and the modified proposal, and (2) reduction of company funds due to belief that a contract potential is reduced. Changes after a contract is let are negotiated and the costs passed directly along to the Air Force.

4.2 SOFTWARE DESIGN

The detailed design phase for software consists of translating the Preliminary Design Review (PDR) approved functional flow diagrams, into detailed logic decision diagrams as depicted in paragraph 4.7 of the Requirements Specifications guidebook. This level of detailing allows a much greater depth of software estimating. At the Critical Design Review (CDR), which is the culmination of the detailed design phase, updated estimates should be available based on the exact number of software modules required (and detailed in the logic decision diagrams) and the number and complexity of instructions or lines of code in each module.

TS operational software; which consists of simulated functional models, instructor-interaction components, and executive control routines, will in most cases be specifically developed for the simulator under construction. The cost of developing each line will have to be borne. However, similarities to existing simulator software may allow an improved development efficiency. Present estimates of total software development costs vary greatly with manhour rates which recently range from \$16.25 per man-hour (AFR 173-10 base level labor rate and probably applicable only to the most rudimentary development tasks) to

\$90 per man-hour for development of special purpose software instrument drivers by highly skilled software experts. A usable average of industry rates that presently (FY78) applies to software development is \$25 to \$35 per manhour which includes a company's overhead. These values are presented only as a point of information as almost all engineering contracts with cost will be in the form of manhours, the dollar rate per manhour being a function that is handled within the finance organizations of both the Air Force and industry.

The primary purpose for this engineering exclusion from actual dollar costs is that numerous non-technical respects, such as corporate profit considerations, industrial labor contracts, and competitive pricing, enter into the actual rate a company must charge for its services; thereby, making finance a field of special expertise much the same as engineering is field of technical expertise. One area in which engineering must utilize dollar costs is in life cycle trade studies as described in paragraph 3.1 of this guidebook and identified in section 4.1 of the Computer Program Maintenance guidebook. For these purposes a standard guidebook average rate as indicated above is used. An example of how converting to dollars from manhours becomes complicated is do to inflation considerations. i.e., a manhour this year equals a manhour hour next year, whereas a dollar this year is less than a dollar next year (in purchasing power). Since software development is basically a manual task, the productivity does not vary greatly from year to year; however, the dollar inflation rate can be assumed to be from 5% to 10% annually. The compound interest formula that converts this year's cost into a future year's cost is:

$$n^{\text{th}} \text{ year cost} = (1+i)^n \cdot (\text{base year cost})$$

where n is defined as the number of whole years from base year and i is defined as the annual inflation rate expressed as a decimal (7%=0.07).

The inflation rate presently assumed by Air Force Logistics Command is 5% per annum. The present inflation rate of the Consumer Price Index of the Bureau of Labor Statistics is approximately 8% per annum, and the discount rate (equivalent to inflation rate) of DODI 7041.3 is 10% per annum.

TS support software, which consists of library, utility and maintenance routines, most often are standardized type software packages. Thereby, they require much less actual development time for the number of instructions involved. The same standardization exists for both ATE operational and ATE support software. An ATE supplier will produce only one basic ATE family, i.e., second generation "rack and stack" or third generation computer generated signal synthesis and measurement. From his library of software routines the ATE contractor will add, modify or expand existing software to provide the specific signal generation, signal measurement, and signal routing and control required.

The primary cost function of the Air Force during the detailed design phase is to monitor and review the software design cost and schedule performance. This function culminates at the CDR, whose purpose is to formally verify that the design phase has been completed, to insure compliance with the requirements, assure the technical merit of the design, review plans for the following phases, and to review the cost and schedule performance of the design effort. It is the contractor's responsibility to provide the necessary review materials to show:

- a. original cost estimates
- b. actual cost to date including any deviations
- c. future cost estimates corrected relative to past cost history
- d. identification of risk items that could impact cost and potential work-

arounds or corrective actions that may be necessary

- e. explanation of the anticipated cost reporting system and its implementation

- f. the plan for notification and handling of any major cost deviations

Before approval to proceed is given, the contractor is contractually obligated to respond to all pertinent questions unanswered at the CDR meetings.

4.3 CODING AND CHECKOUT

The coding and checkout phase consists of converting the approved CDR design into operational code. The primary Air Force cost function during this and subsequent periods is that of cost tracking. Cost report data (described in section 6 of this guidebook) are prepared by finance groups. Engineering becomes involved primarily when substantial cost deviations occur due to unanticipated technical problems or required manpower efforts well beyond the CDR contractual estimates. Correction of these problems may require an engineering evaluation to help establish whether the deviations were due to poor contractor performance or to unforeseeable factors. This determination of cause and subsequent review of proposed corrective action will be utilized to assist in providing corrective funding normally via an Engineering Change Proposal to the contract. Performance variations, either Air Force desired or contractor requested, can also cause a technical evaluation of the manpower level of effort, as was accomplished during the Analysis phase.

A common estimate for software development is the generation of 4 to 5 instructions per day. It must be observed; however, that this is not the rate at which the actual code is written during this phase, but rather an estimate to obtain the total development manhours for all phases of the software program generation. Although much of the TS sup-

port software and the ATE control and support software need not be newly generated because of its past development, there may still be a substantial cost associated with modifications and integration. Also, any product previously developed by a private contractor, has a monetary value and is sellable. Therefore, these software modules may be considered proprietary and must (1) be purchased at full price from the contractor, (2) purchased at a reduced price with a restrictive usage condition so that it does not become public property (thereby eliminating its future marketability), or (3) be given to the Air Force to gain a competitive edge in contract obtainment. Many present contracts for weapon systems contain provisions for Air Force ownership of flight characteristics data and mission operational software for subsequent use in training simulators.

At the beginning of the coding and check-out phase of ATE operational and support software, the ATE test software interface is explicitly defined. This allows the Analysis phase of the ATE test software to proceed. The ATE test software is unique in that it consists of a series of independent application programs normally developed by someone other than the ATE station contractor. (i.e.: UUT vendor or weapon system prime contractor). They must each therefore, be contracted for in a manner identical to contracting for other software. The cost estimating methods in this guidebook apply equally well to the ATE test software. Most ATE software is manually developed in the four phase sequence described in this section. One notable exception exists; this being the computer generation of ATE test programs for digital circuit cards (only). This process, known as Automatic Test Program Generation (ATPG), reduces, via automation, the effort in the detailed design and coding and checkout phases. Because these phases are automated, they reduce substantially the number of programmer manhours development required for a price of only a relatively small number of computer hours. An example of the

cost relationship between a programmer generated test and an ATPG test is depicted in Figure 4.3-1. The example chosen is based on Omnicomp estimates and is for a digital circuit card of average complexity of about 50 integrated circuits, none of which are Large Scale Integrated Circuits (LSI) such as memories or microprocessors. It is to be observed that as the level of fault isolation capability increases, the cost for both ATPG and programmer-generated ATE test software increases drastically. This phenomenon exists in all ATE test software, therefore prompting a tradeoff between the level of fault isolation capability and development cost. This trade is essentially a cost trade because as the fault isolation capability and development cost goes up, the time to pinpoint a malfunction and its associated cost goes down. A compromise that is often found on Line Replaceable Unit (LRU, black box level) testing, is a requirement such as a 95% of all faults to be isolated to within three circuit cards, 90% of all faults to be isolated within two circuit cards, and 80% of all faults to be a single circuit card.

For large scale ATE the cost of the test software usually is by far the predominate software development cost. This is true because an individual test program must be generated for each UUT. In addition, few newly developed test programs can rely on existing software modules but rather must be newly developed in their entirety. Table 4.3-1 presents a list of ATE LRU test software development cost, as presented in the Avionics Intermediate Support for Advanced Medium STOL Transport report (Technical Memorandum ENEG-TM-77-1). These data show the estimated level of costs that are anticipated based on F-15 and F-111 programs (which contain units that are functionally similar).

The Table 4.3-1 estimated costs were obtained by analogy to existing test package costs (F-15 and F-111). As the test package development progresses and the new LRU's design becomes well defined, more detailed estimates are periodically

PERCENT OF TEST
COMPREHENSIVENESS
VS. APPROXIMATE
SOFTWARE GENERATION COST
FOR AVERAGE COMPLEXITY
DIGITAL CIRCUIT CARD
(PROGRAMMER GENERATED
ATE TEST SOFTWARE AND
AUTOMATIC TEST PROGRAM
GENERATION)

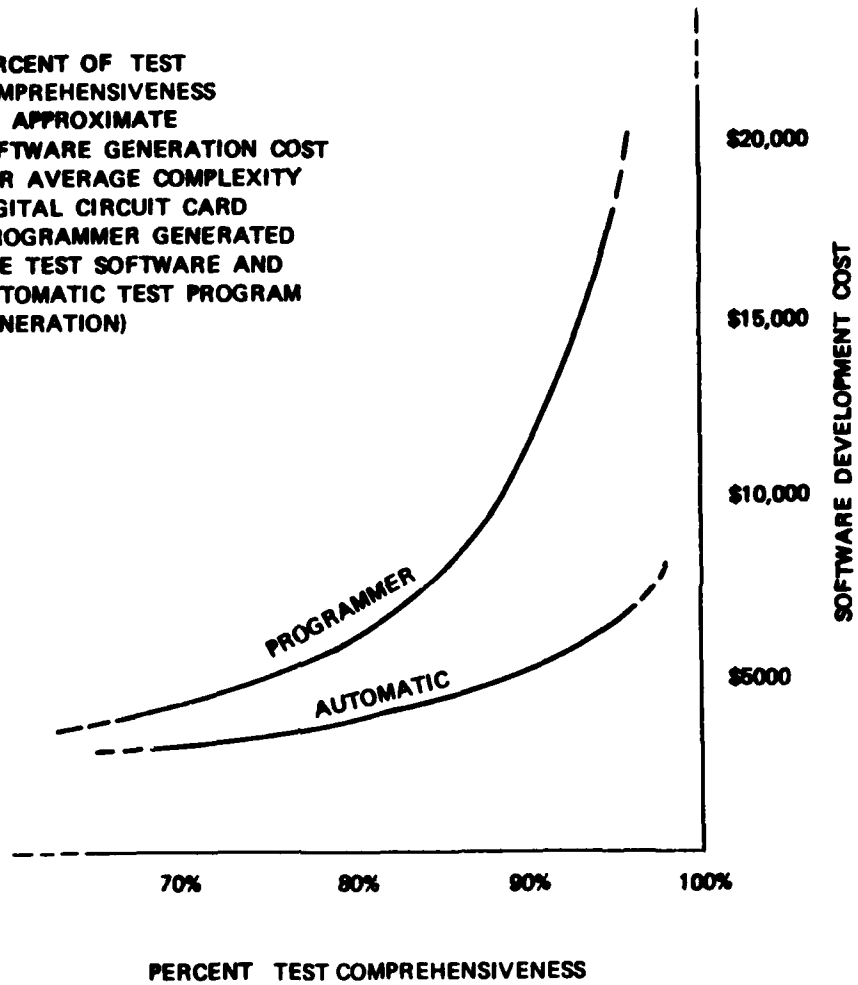


Figure 4.3-1 Digital Circuit Card Test Software Generation Cost

Table 4.3-1. ATE LRU Test Software Development Costs

LRU Complexity Level	Typical LRUs	Estimated ATE Test Software Development Cost
Very Simple	APN-194 Radar Altimeter Antenna (AS-2038) APN-169 Station Keeping Radar Amplifier (AM-6308)	\$ 23K
Simple	AIC-18 Interphone Amplifier (AM-1963) AIC-13 Public Address Control (C-1614)	\$ 139K
Average	ARC-123 HR Radio Amplifier (AM-4573) ARC-164 UHF Radio Receiver/Transmitter (RT-1145)	\$ 556K
Complex	APN-194 Radar Altimeter Receiver/Transmitter (RT-1042) APX-101 IFF Receiver/Transmitter (RT-1063)	\$1160K
Very Complex	APN-169 Station Keeping Radar Coders/Decoder (KY-567) AVQ-30 Weather Radar Receiver/Transmitter (RT)	\$2780K

updated based on LRU interface connector pin count, part complexity (active, passive microelectronics), LRU circuit interface functions, Radio Frequency (RF), power, digital, analog, and the number of test steps as the code and checkout phase for each ATE test program is entered.

4.4 INTEGRATION AND TEST

The integration phase involves, (1) costs to accomplish program de-bug to the point of functional operation of all software components and all hardware/software interfaces, (2) generation of documentation to thoroughly describe the as-built program, (3) test accomplishment for system validation, and (4) an operational (users) manual. Completion of the integration and test phase signifies transfer of efforts from an engineering task to that of a quality assurance task for Formal Qualification Review and certification (based on the documentation developed in the integration phase). The cost of both integration and test will include utilization of the entire TS or ATE system and supporting facilities and personnel.

The question of specifically what constitutes software costs is most pronounced in the integration and test phase. In previous phases, software items were those WBS and requirement items that were associated with system control functions, their description (flow diagrams) and generation of a software media (coding). However, during the integration and test phase, this uniqueness is less well defined as this phase is not actually a software phase, but more properly a systems phase. The purpose is to provide system operation by proper interaction between the hardware, software, documentation and man/machine interface.

The personnel who developed the software through the coding and checkout phase will be assisted by hardware designers, test engineers, quality assurance personnel and various support groups. For budgeting purposes many organizations will identify the software tasks

as that manpower effort expended by the personnel attached to the software group who did the initial software development. The question arises however, whether the time expended in TS or ATE system operations should be applied to software development, i.e. is the software being de-bugged or is the system being de-bugged? Normally during TS or ATE operational and support software development, both the hardware and software are new and the integration and test phase is considered basically a system checkout. During ATE test software integration and test, both the UUT and the ATE test station are operational. Since the only hardware then involved in ATE test software is the Interface Test Adapter (ITA), there is a tendency to consider this phase software development.

The primary Air Force cost function in the integration and test phase is to monitor the cost in a manner similar to that during the code and checkout phase. Particular diligence is required in this activity since this is a vital phase, where the systems must satisfy all the functional and operational requirements, and the cost to correct technical or incompatibility problems can quickly exceed cost estimates.

4.5 TOTAL DEVELOPMENT COSTS

There are two activities that occur after software validation: installation, and operation and support (deployment and maintenance). These have not been discussed because once the initial software media and associated documentation have been accepted, subsequent copies are readily and inexpensively produced with little or no engineering required. The primary effort on software that is encountered during these two phases is software maintenance, which is discussed in Section 5 of this guidebook and in the Computer Program Maintenance guidebook.

Throughout the ATE and TS software development cycle, the software is dependent on ATE and TS system development. This means that software costing is not

treated as a separate entity in cost measuring or reporting. Although the WBS will identify the software aspects, the cost accounting system will not, as indicated in paragraph 3.4 of this guidebook, normally make specific reports or consideration for software. This then makes it incumbent upon the individual concerned with the software to be familiar with the entire systems task. This is particularly true during the later phases when software cost considerations are mainly that of cost monitoring.

There are numerous costs associated with software program development that are not immediately considered in the direct line development. The most straightforward is the purchase of off-line computer time or purchase of a dedicated program development computer to handle software simulations, compiling and syntax correction. In TS mission simulators

the operational mission program is utilized; however, modifications will usually have to be made to allow it to operate properly in the TS environment and to interface properly with the TS data simulation programs. In flight TS the reduction, coding and modification of flight data will have to be undertaken to produce representation of the aircraft dynamics characteristics. For ATE test program development, UUTs will have to be leased or purchased to allow checkout of the program. Training programs will have to be setup for the specific programs that are to be utilized. The examples presented here are not intended to be exhaustive, but rather to give an idea of some of the ancillary cost items that will be encountered and should be addressed in the WBS.

Section 5.0 SOFTWARE MAINTENANCE COSTS

The cost of software is often a large component of total TS and life cycle costs. Software maintenance, which results from error removal, operating improvements, and system requirement changes, is any activity which alters previously developed software. This section, which discusses cost aspects of maintaining existing software, is therefore a discussion relative to the cost of changes; and changes are expensive. Most system overrun costs are due to changes that occur for numerous reasons. Both the Air Force and aerospace contractors are prone to some extent of utilizing changes to make up for underestimated costs and to cover poor prior planning. In this respect, the Air Force may fund desired efforts by the change route rather than obtaining appropriation for a new system (which would be much harder to obtain). Similarly an aerospace contractor may marginally bid a TS or ATE contract with the hope or intent of making an overall profit on the subsequent and inevitable changes associated with the development phase. However, the vast majority of changes are a necessary and expected part of developing a new TS or ATE system and should be accepted as a portion of the overall task with appropriate planning and consideration.

The Computer Program Maintenance Guidebook provides a detailed discussion of the aspects of maintenance relative to ATE and TS. That guidebook should be consulted to understand the significance and extensiveness of maintenance as it relates to TS and ATE software. Specific paragraphs of the maintenance guidebook are referenced in this section rather than include excessive duplication.

5.1 SOFTWARE MAINTENANCE FACILITIES AND ACTIVITIES

The cost associated with software maintenance derive from activities similar to development of new programs. This includes planning for change implementation, program writing/rewriting, syntax

correction and simulation running, documentation updating, system level de-bug (integration), and validation and configuration control. These efforts necessitate not only programmers, computing facilities and system components; but also support services such as resource planning, change control, configuration management, document control, and logistics support.

Changes, and thereby maintenance, can be classified as those occurring both prior to Air Force software acceptance and those occurring subsequent to acceptance, i.e. before and after deployment. Those after acceptance can be implemented either by Air Force or by an appropriate contractor. A prime consideration is the Air Force ownership and possession of the TS or ATE software source code. This code, as opposed to the normally obtained object code which only makes sense to a machine, is mandatory to provide the Air Force the flexibility to incorporate changes itself or from a contractor other than the one which originally developed the program. Although there is an additional cost associated with source code purchase, most recent software acquisitions include the source code because of this downstream maintenance flexibility. In the ATE area, essentially all new test software is being developed in ATLAS which is an English language type higher order language. Procurement of ATLAS source code provides a relatively easy data base to incorporate future changes. Paragraph 6.2.3 of the maintenance guidebook provides a description of this organic (Air Force) capability versus contractor support with respect to life cycle cost trades. Paragraph 4.1 of the maintenance guidebook describes the maintenance planning activities including the resources that must be considered.

Changes prior to Air Force acceptance are implemented by the software development contractor by necessity, and are

classified as Class I or Class II. Class I changes are Air Force funded and implemented via an ECP. Class II changes are contractor funded and are usually for error correction. ECPs are handled through a change board which schedules and plans all aspects of the change. Estimates are provided in the same manner as for the proposal prior to ECP approval. The sequence of implementation events is described in section 4 of this guidebook. Because the change is incorporated in sequence with the basic contract development, a high level of detail is required in describing exactly what effort is required and when it will be accomplished relative to the existing schedule. This detail planning can take a substantial amount of time. Therefore, it is not unusual to initiate the first phases via pre-approval implementation as agreed between the Air Force and the contractor.

5.2 CHANGE ESTIMATING

ATE and TS software change estimating is similar to initial software estimating in that all aspects of the task must be broken into their component parts (similar to the WBS) for item by item pricing. The estimating is accomplished in the same manner as in an original contract proposal estimation. Paragraph 3.2 of this guidebook provides estimating methods that are utilized and paragraph 3.1 discusses the various models that are available.

The one main estimating method that is not directly applicable to paragraph 3.2 is that of estimating the level of maintenance effort that will take place after Air Force acceptance (as opposed to individual change estimating). A general "rule of thumb" that most individuals use is that the ongoing level of maintenance, per year, will average between eight and twelve percent of the original total cost of development. A more precise method, based on experience of The Boeing Company and modified for application to ATE and TS systems, is as follows.

5.2.1 Step 1 - Estimating Difficulty of Task

The estimating task should begin with analysis of the software that is to be maintained, and an assessment of the difficulty of doing the work. For this purpose, the generalizations are made:

<u>If the Software Type is:</u>	<u>Then Maintenance will be:</u>
Mathematical Operations	Medium
Non Real Time Input, Output	Easy
File, Data Base Manipulation	Hard
Logic Operations	Medium
ATE Signal Processing	Medium
Real Time or Executive	Hard
Real Time Input/Output	Hard

Corresponding to this assessment, we can assume the following rates of programming productivity:

<u>If Maintenance is:</u>	<u>Then Productivity will be:</u>
Easy	6000 HOL Statements/Man-Year
Medium	3000 HOL Statements/Man-Year
Hard	1200 HOL Statements/Man-Year

5.2.2 Step 2 - Estimating the Change Rate

The next step is to count or estimate the number of statements of each type in the software, and to estimate the fraction of the total that is likely to be changed each year. As a rule, the "easy" routines are most likely to be changed and the "hard" ones least likely. The acquisition engineer then makes an estimate, based on his judgement of the spe-

cific requirements of the job at hand, experience with other support systems, etc., as to the percentage of difficulty each category of software will change. Boeing experience has indicated that ground system software is such that approximately 15% of "easy," 5% of "medium" and 1% of "hard" coding is changed each year.

5.2.3 Step 3 - Estimating the Labor Expenditure

Given the above assessments, you can now calculate the rate at which you will use programmer labor for the maintenance tasks:

Equivalent Heads, =
In Man-Years

$$\frac{\sum (Total\ Statements_k) (change\ Rate_k)}{K (Productivity_k)}$$

where k indexes the categories of "easy," "medium," and "hard."

5.2.4 Step 4 - Estimating Personnel Turnover

It is common practice to employ younger, less-experienced programmers in maintenance assignments, with the result that such activities often involve a high rate of personnel turnover. Because this can have a significant effect on labor costs, turnover rates should be considered.

The industry turnover rate most frequently used is 30%. USAF personnel turnover rates for the operating base, or using command, of the ATE or TS being provided should be used in most cases.

5.2.5 Step 5 - Estimating Labor Cost Rates

The people who will be assigned to tasks as replacements for others who leave normally will have to be trained and, for some period of time, may be less than fully productive.

For the first year, standard labor rates should apply. For all years after the first, the rate should be composed as:

$$Labor = (Standard\ Rate) + ((Turnover\ Rate) * (Training\ Cost))$$

Where "training cost" is an estimate of the cost to bring a new assignee to full productivity.

5.2.6 Step 6 - Estimating Total Labor Costs

The last step is to simply multiply these yearly labor rates by the "equivalent heads" number derived in Step 3, and compute the sum.

5.3 TOTAL MAINTENANCE COSTS

Since maintenance costs can be such a significant part of system costs, designing for maintainable software and including estimates for this function in an original estimate, is an economically wise decision. Paragraph 3.2 of the maintenance guidebook provides an illustration of the relative dramatic increase in maintenance costs as a development program progresses. This increasing cost of maintenance is caused by the fact that as a program progresses, more-and-more code, documentation, and completed software programs will be affected; plus the fact that as software modules become integrated the "domino effect" takes place. This affect is caused by the interactions of software statements, so that when a change is inserted, far reaching affects come into play. Paragraph 3.1.2 of this guidebook discusses software operational and support costs. A common shortcoming in cost projections is the allocation of insufficient budget and resources to account for this maintenance of delivered software. The Computer Resources Integrated Support Plan (CRISP) is intended to fully identify these required maintenance items.

The CRISP identifies organizational relationships and responsibilities for the management and technical support of com-

puter resources, and is prepared with the guidelines specified in AFR 800-14, Volume II. It functions during the full scale development phase to identify computer resources necessary to support computer programs after transfer of program management responsibility and system turnover to the using command. It continues to function after the transfer of program management responsibility and system turnover, as the basic agreement between the supporting and using commands for management and support of computer resources. It is incumbent upon the ATE or TS software manager to insure that all necessary items are identified in the CRISP, and that sufficient esti-

mates have been incorporated into O&S funding to provide for software maintenance.

For full cycle maintenance considerations, it must be remembered that a contractor's proposal to a RFP does not include any items for maintenance, (pre-delivery ECPs), and that all funded changes that occur will be in addition to the original contract costs. Since the aggregate ECP costs can exceed the original software cost estimates, each ECP cost and those of projected ECPs must be carefully examined whenever total life cycle costs are being considered.

Section 6.0 BASIC COST REPORTS

Paragraph 3.4 discussed the purpose and the use of cost reports very generally. This section explains the specified finance reports that would be required by the Air Force.

The periodic performance and funding reports are the Cost Performance Report (CPR), consisting of five different formats (see Figures 6.2-1 thru 6.2-5), the Cost/Schedule Status Report (C/SSR) (See Figure 6.2-6), and the Contract Funds Status Report (CFSR) (see Figure 6.2-7). The performance reporting is done either with the CPR or C/SSR which are final outputs generated by the contractor's supporting computer data runs. These reports show program actual progress and expected future program status. Serving as managerial tools for decision making and planning, they indicate areas where further in-depth analysis may be necessary. These reports provide the means to collect summary level cost and schedule performance data. The CFSR is used to provide funding data and is used by both parties to assess funding requirements.

Contractors are encouraged to substitute internal reports for these three reports provided that the data elements and definitions are comparable to the basic reports and that the reports are in forms suitable for management use.

6.1 COST REPORT REQUIREMENTS

The CPR reflects data by the C/SCSC system. It is intended to provide early identification of problems having significant cost/schedule impact, to provide effects of management actions taken to resolve existing problems, and to provide program status information for use in making and validating management decisions.

The CPR will be applied to selected contracts within those programs designated as major defense systems. It will be established as a contractual requirement as stated on the CDRL and DID. The CPR

is applicable to on-going contracts only in those cases where the procuring agencies consider it necessary to support program management needs and DOD requirements for information. Some of the factors which may affect applications to on-going contracts are anticipated time to contract completion, anticipated program deferrals, and the relative importance of subcontracts. The CPR will normally not be required on either firm-fixed-price contracts or those where C/SCSC is not required.

The CPR will normally be required on a monthly basis and submitted to the procuring activity no later than the 25th day following the reporting cut-off date. The level of detail to be reported will normally be limited to level three of the WBS (e.g. Level I = aircraft system; Level II = air vehicle; Level III = airframe; Level IV = wing; Level V = flaps) or higher, except when a problem area is indicated at a lower level of the WBS, in which case more detailed data will be provided until the problem is resolved. In all cases, the CPR is subject to tailoring to require less data via negotiations.

The CPR consists of five formats (see Figure 6.2-1 thru 6.2-5). Format One provides data to measure cost and schedule performance by summary level work breakdown structure elements. Format Two provides a similar measurement by organizational or functional cost categories. Format Three provides the budget baseline plan against which performance is measured. Format Four provides manpower loading forecasts for correlation with the budget plan and cost estimate predictions. Format Five is a narrative report used to explain significant cost and schedule variances and other identified contract problems.

The C/SSR (see Figure 6.2-6) is applicable to contracts of \$2 million or more and of more than one year duration which do not use the CPR. Five areas of C/SSR

are common with the CPR. C/SSR is normally not required on firm-fixed-price contracts. Application of C/SSR to ongoing programs is held to a minimum as well as are the data requirements. C/SSR will be established as a contractual requirement. Data reported will normally be limited to WBS level two and selected level three items. Data will be reported monthly, normally due by the 25th day of the following month.

Application of the C/SSR does not involve certain of the unique requirements or disciplines of the C/SCSC such as use of work packages for determining BCWP unless these methods constitute the contractor's normal way of doing business. Derivation of BCWP is left to the contractor, subject to negotiation and inclusion in the contract. Variance thresholds which, if exceeded, require problem analysis and narrative explanations, will be as specified in the contract or as mutually agreed to by the contracting parties.

The CFSR (See Figure 6.2-7) is normally applicable to all contracts over \$500,000. It is intended to supply funding data that, with other performance measurement inputs, provide DOD information to update and forecast contract fund requirements, to plan and make decisions on funding changes, to develop fund requirements and budget estimates in support of approved programs, and to determine those funds in excess of contract needs that are available for de-obligation. The CFSR may be implemented at a reduced level of reporting for contracts between \$100,000 to \$500,000, for time and material contracts, and for information on limited funding requirements. The CSFR is normally not required on firm-fixed price contracts, on contracts less than \$100,000, on contracts of less than six months expected duration, and on facilities contracts.

6.2 COST REPORT DESCRIPTION

The following figures constitute examples of the cost reporting forms associated with paragraph 6.1.

CONTRACTOR:		COST PERFORMANCE REPORT - WORK BREAKDOWN STRUCTURE				SIGNATURE, TITLE & DATE		FORM APPROVED				
LOCATION:		CONTRACT TYPE / NO.:		PROGRAM NAME NUMBER		REPORT PERIOD:		OMB NUMBER				
NOTE <input type="checkbox"/> PRODUCTION <input type="checkbox"/>		EST COST AUTH. UNPRICED WORK		EST PRICE		SHARE RATIO		EST CEILING				
QUANTITY		EST COST AUTH. UNPRICED WORK		EST PRICE		SHARE RATIO		EST CEILING				
ITEM	CURRENT PERIOD				CUMULATIVE TO DATE				AT COMPLETION			
	BUDGETED COST		ACTUAL COST WORK PERFORMED		BUDGETED COST		ACTUAL COST WORK PERFORMED		VARIANCE			
	Work Scheduled	Work Performed	Cost	(1)	Work Scheduled	Work Performed	Cost	(1)	Work Scheduled	Work Performed		
(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)
<u>WORK BREAKDOWN STRUCTURE</u>												
GER AND ADMIN												
UNDISTRIBUTED BUDGET												
SUBTOTAL												
MANAGEMENT RESERVE												
TOTAL												
RECONCILIATION TO CONTRACT BUDGET BASELINE												
VARIANCE ADJUSTMENT												
TOTAL CONTRACT VARIANCE												

Figure 6.2-1 Cost Performance Report/Work Breakdown Structure

CONTRACTOR		COST PERFORMANCE REPORT - FUNCTIONAL CATEGORIES										FORM APPROVED OMB NUMBER 3300210				
LOCATION: <input type="checkbox"/> ROTEL <input type="checkbox"/> PRODUCTION <input type="checkbox"/>		CONTRACT TYPE / NO.:				PROGRAM NAME NUMBER:				REPORT PERIOD:						
ORGANIZATIONAL OR FUNCTIONAL CATEGORY	CURRENT PERIOD					CUMULATIVE TO DATE					AT COMPLETION					
	BUDGETED COST		ACTUAL COST		VARIANCE		BUDGETED COST		ACTUAL COST		VARIANCE		LATEST REVISED ESTIMATE	VARIANCE		
	Work Scheduled	(2)	Work Performed	(3)	Cost	(4)	Work Scheduled	(7)	Work Performed	(8)	Cost	(10)			(11)	(13)
(1)																
GEN AND ADMIN																

Figure 6.2.2 Cost Performance Report/Functional Categories

CONTRACTOR:		COST PERFORMANCE REPORT - BASELINE						FORM APPROVED
LOCATION:		CONTRACT TYPE / NO.:		PROGRAM NAME NUMBER		REPORT PERIOD:		OMB NUMBER
NOTICE <input type="checkbox"/>		PRODUCTION <input type="checkbox"/>						22R0700
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
ORIGINAL CONTRACT TARGET COST	NEGOTIATED CONTRACT CHARGES	CURRENT TARGET COST (1) + (2)	ESTIMATED COST OF AUTHORIZED UNPRICED WORK	CONTRACT BUDGET BASELINE (1) + (4)	TOTAL ALLOCATED BUDGET	DIFFERENCE (3) - (6) (SEE PAGE 3)		
BUDGETED COST FOR WORK SCHEDULED FROM CUMULATIVE								
ITEM	BCBS CUM TO DATE	3/2 MONTH FOR(LAST)						TOTAL BUDGET
		(1)	(2)	(3)	(4)	(5)	(6)	
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
PM BASELINE (BEGINNING OF PERIOD)								
*LIST BASELINE CHARGES AUTHORIZED DURING REPORT PERIOD								
GEN AND ADMIN								
UNDISTRIBUTED BUDGET								
PM BASELINE (END OF PERIOD)								
MANAGEMENT RESERVE								
TOTAL								

Figure 6.2.3 Cost Performance Report/Baseline

CONTRACTOR:		COST PERFORMANCE REPORT - MANPOWER LOADING										FORM APPROVED					
LOCATION:		CONTRACT TYPE / NO.:				PROGRAM NAME NUMBER:				REPORT PERIOD:		OSR NUMBER					
NOTE <input type="checkbox"/>		PRODUCTION <input type="checkbox"/>		FORECAST (NON-CUMULATIVE)										AT			
ORGANIZATIONAL OR FUNCTIONAL CATEGORY		ACTUAL END OF CURRENT PERIOD (CUM)	36 MONTH FORECAST BY MONTH (ENTER NAMES OF MONTHS)										COMPLETION				
		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	
(1)																	

Figure 6.2-4. Cost Performance Report/Manpower Loading

CONTRACTOR		COST PERFORMANCE REPORT - PROBLEM ANALYSIS		FORM APPROVED
LOCATION		CONTRACT TYPE / NO.:	PROGRAM NAME / NUMBER	OMB NUMBER
NOTE <input type="checkbox"/>	PRODUCTION <input type="checkbox"/>			2128220
				REPORT PERIOD

EVALUATION

Section 1 - Total Contract: Provide a summary analysis, identifying significant problems affecting performance. Indicate corrective actions required, including Government action where applicable.

Section 2 - Cost and Schedule Variances: Explain all variances which exceed specified variance thresholds. Explanations of variances must clearly identify the nature of the problem, the reasons for cost or schedule variance, impact on the immediate task, impact on the total program, and the corrective action taken. Cost variances should identify amounts attributable to rate changes separately from amounts applicable to manhours.

Within this section, the following specific variances must be explained:

- a. Schedule variances (Budgeted Cost for Work Scheduled vs Budgeted Cost for Work Performed)
- b. Cost variances (Budgeted Cost for Work Performed vs. Actual Cost for Work Performed)
- c. Cost variance at completion (Budgeted at Completion vs. Latest Revised Estimate at Completion)

In addition to the variance explanations above, the following analyses are mandatory:

- a. Identify the effort to which the undistributed budget applies
- b. Identify the amount of management reserve applied during the reporting period, the WBS elements to which applied, and the reasons for application

Section 3 - Baseline: If the difference shown in block (7) on format 3 becomes a negative value or changes in value, provide:

- a. Procuring activity authorization for the baseline change which resulted in negative value
- b. The amount (by WBS element) used to adjust for unfavorable performance incurred prior to the baseline change
- c. The amount (by WBS element) added to budgets previously established for future effort. Explain reasons for the additional budget in the following terms:
 - (1) In-scope engineering changes
 - (2) In-scope support effort changes
 - (3) In-scope schedule changes
 - (4) Economic change
 - (5) Estimating change
 - (6) Unpredictable change
 - (7) Other (specify)
- d. The amount (by WBS element) for added in-scope effort not previously identified or budgeted

Figure 6.2-5 Cost Performance Report/Problem Analysis

CONTRACTOR:		COST/SCHEDULE STATUS REPORT		SIGNATURE, TITLE & DATE		FORM APPROVED		
LOCATION:		CONTRACT TYPE / NO.:	PROGRAM NAME/NUMBER	REPORT PERIOD:		OMB NUMBER	22RO327	
RYSE <input type="checkbox"/>	PRODUCTION <input type="checkbox"/>							
		Contract Data						
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	
ORIGINAL CONTRACT TARGET COST	NEGOTIATED CONTRACT CHANGES	CURRENT TARGET COST (1) + (2)	ESTIMATED COST OF AUTHORIZED, UNPRICED WORK	CONTRACT BUDGET BASELINE (1) + (4)				
Performance Data								
CUMULATIVE TO DATE				AT COMPLETION				
WORK BREAKDOWN STRUCTURE	BUDGETED COST		ACTUAL COST WORK PERFORMED	VARIANCE		BUDGETED	LATEST REVISED ESTIMATE	VARIANCE
	Work Scheduled (2)	Work Performed (3)	(4)	Schedule (5)	Cost (6)	(7)	(8)	(9)
(1)								
GEN AND ADMIN								
UNDISTRIBUTED BUDGET								
MANAGEMENT RESERVE								
TOTAL								

Figure 6.2-6 Cost/Schedule Status Report

CONTRACT FUND STATUS REPORT - Dollars in _____											
CLASSIFICATION	CONTRACT NUMBER	CONTRACT TYPE	CONTRACT REPORT DATE	CONTRACT REPORT DATE	CONTRACT REPORT DATE	CONTRACT REPORT DATE	CONTRACT REPORT DATE	CONTRACT REPORT DATE	CONTRACT REPORT DATE	CONTRACT REPORT DATE	CONTRACT REPORT DATE
1. CONTRACT NUMBER	2. CONTRACT TYPE	3. CONTRACT REPORT DATE	4. CONTRACT REPORT DATE	5. CONTRACT REPORT DATE	6. CONTRACT REPORT DATE	7. CONTRACT REPORT DATE	8. CONTRACT REPORT DATE	9. CONTRACT REPORT DATE	10. CONTRACT REPORT DATE	11. CONTRACT REPORT DATE	12. CONTRACT REPORT DATE
LINE ITEM OR ELEMENT	APPROPRIATION	FUND	ACCOUNT	ACTUAL TO DATE	ACTUAL TO DATE	ACTUAL TO DATE	ACTUAL TO DATE	ACTUAL TO DATE	ACTUAL TO DATE	ACTUAL TO DATE	ACTUAL TO DATE
13. CONTRACT REPORT DATE	14. CONTRACT REPORT DATE	15. CONTRACT REPORT DATE	16. CONTRACT REPORT DATE	17. CONTRACT REPORT DATE	18. CONTRACT REPORT DATE	19. CONTRACT REPORT DATE	20. CONTRACT REPORT DATE	21. CONTRACT REPORT DATE	22. CONTRACT REPORT DATE	23. CONTRACT REPORT DATE	24. CONTRACT REPORT DATE
CONTRACT VALUE APPROXIMATED BY FEE REPORT - ACTUAL OR PROJECTED											
A. UNOBLIGATED CONTRACTS	B. ACCRUED LIABILITIES	C. TOTAL 1324 - 123	D. BALANCE OF BILLS TO BE RECEIVED	E. TOTAL	F. TOTAL	G. TOTAL	H. TOTAL	I. TOTAL	J. TOTAL	K. TOTAL	L. TOTAL
13. CONTRACT REPORT DATE	14. CONTRACT REPORT DATE	15. CONTRACT REPORT DATE	16. CONTRACT REPORT DATE	17. CONTRACT REPORT DATE	18. CONTRACT REPORT DATE	19. CONTRACT REPORT DATE	20. CONTRACT REPORT DATE	21. CONTRACT REPORT DATE	22. CONTRACT REPORT DATE	23. CONTRACT REPORT DATE	24. CONTRACT REPORT DATE

Page _____ of _____ Pages

Figure 6.2-7 Contract Funds Status Report (CFSR)

Section 7.0 BIBLIOGRAPHY

The following documents and reports are applicable to the subject of cost for ATE and TS software:

1. A Provisional Model for Estimating Computer Program Development Costs, Brad C. Frederic (Tecalote Research, Inc.) December 1974
2. A Review of Software Cost Estimation Methods, Judith A. Clapp (The MITRE Corp.)
3. AFAL report F33615-77-C-1252, developed under PE62204F, Project 2003, Task 09, Work Unit 02, June-August 1976
4. AFAL/AAA-3, Modified Wolverton Model
5. AFR173-10
6. AFR 800-6, Program Control-Financial
7. AFR 800-11, Life Cycle Costing
8. AFR 800-14, Volume II
9. AFR 800-17
10. AFSC Electronic System Division Software Workshop Summary Notes, October 1974
11. AFSCO 800-19, Design-to-Cost Guide
12. AFSCP/AFLCP 173-5
13. AFSCR 70-11, Financial Reporting
14. Armed Forces Procurement Regulation (ASPR)
15. Avionics Intermediate Support for Advanced Medium STOL Transport, Technical Memorandum ENEG-TM-77-1, May 1977
16. MIL-STD-1513 (USAF), Trade Studies for the Selection of Avionic Test Support Systems, Criteria for 15 January 1971
17. Handbook of Logic-Circuit Testing, Omnicomp, Inc. 1974
18. Preliminary Study on Estimating the Development Cost of Software, R. H. Darrow, Boeing Co. Document D180-20144-1
19. TRW-SS-72-01, The Cost of Developing Large Scale Software, R. W. Wolverton, March 1971 (IEEE Computer Society R-77-189)
20. VS-88-AQ/M, Document, Operation and Support Cost Trade Studies
21. 1975 Aerospace Corporation report on cost estimating.

Section 8.0 MATRIX: GUIDEBOOK TOPICS VS. GOVERNMENT DOCUMENTATION

The elements in Figure 8.0-1 correspond to the sections in the guidebook wherein the corresponding topic is discussed to the largest extent.

TOPICS	GOVERNMENT PUBLICATION											
	AFR 173-10	AFR 800-14	AFSP/A/FLCP 173-5	ASPR	DDO DIRECTIVE 5000.1	DDO DIRECTIVE 5000.2	DDO DIRECTIVE 7000.1	DDO INSTRUCTIONS 7000.2	DDO INSTRUCTIONS 7041.3	ENEG-TM-77.1	MIL-STD-881A	MIL-STD-1513
PERFORMANCE MEASUREMENT		6.1				3.4.1	3.4.1					
PROGRAM MANAGEMENT AND CONTROL				3.4.1								
EQUIPMENT ACQUISITION					3.4.1							
COMPUTER RESOURCES INTEGRATED SUPPORT PLAN			5.3									
COST/SCHEDULE CONTROL SYSTEMS CRITERIA		3.4.1 3.4.2		3.4.1			3.4					
WORK BREAKDOWN STRUCTURE										3.2.2 3.4.1 3.4.2 4.1		
DOLLAR CONSIDERATIONS								4.2	4.3			
TRADE STUDIES												3.5

Figure 8.0-1. Guidebook Topics Versus Government Documentation

Section 9.0 GLOSSARY OF TERMS

Algorithm - A set of rules or processes for solving a problem in a finite number of steps. This software procedure can be presented to a computer precisely and in a standard form, the computer then taking the algorithm's course of action to solve the problem.

Armed Forces Procurement Regulation - An interservice publication which is the basic source-book for the procurement process. The ASPR is the "Bible" for all contracting by the DOD upon which Air Force regulations and manuals interpret and implement the requirements and policy found therein.

Computer Program - A series of instructions or statements in a form acceptable to computer equipment, designed to cause the execution of an operation or series of operations. Computer programs include such items as operating systems, and maintenance/diagnostic programs. They also include applications programs such as payroll, inventory control, operational flight, strategic, tactical automatic test, crew simulator and engineering analysis programs. Computer programs may be either machine dependent or machine independent, and may be general purpose in nature or be designed to satisfy the requirements of a specialized process of a particular user.

Computer Program Development Cycle - The computer program development cycle consists of six phases: analysis, design, coding and checkout, test and integration, installation, and operation and support. The cycle may span more than one system acquisition life cycle phase or may be contained in any one phase. (AFR 800-14, Volume II)

Contract - A legally enforceable agreement between two parties (AF/Contractor, Contractor/sub-contractor) which describes a program for product acquisition. The contract contains the System Specifications, the Statement of Work, the Contract Data Requirements List, and the Work Breakdown Structure.

Contract Funds Status Report - A report, normally applicable to all contracts over \$500,000, to supply the AF with funding data. This report is intended to provide DOD information to update and forecast contract fund requirements, to plan and make decisions on funding changes, to develop fund requirements and budget estimates in support of approved programs, and to determine those funds in excess of contract needs that are available for deobligation.

Cost Performance Report - A five format cost report (Work Breakdown Structure, Functional Categories, Baseline, Manpower Loading, and problem Analysis) generated by the contractor. The Cost Performance Report, which reflects data produced by the C/SCSC system, is intended to provide early identification of problems having significant cost/schedule impact, to provide effects of management actions taken to resolve existing problems, and to provide program status information for use in making and validating management decisions.

Computer Program Configuration Items - A computer program or aggregate of related computer programs designated for configuration management. A CPCI may be a punched deck of cards, paper or magnetic tape or other media containing a sequence of instructions and data in a form suitable for insertion into a digital computer.

Configuration Item - An aggregation which satisfies an end use function and is designated for configuration management.

Configuration Management - A management discipline applying technical and administrative direction and surveillance to:

- a. Identify and document the functional and physical characteristics of a configuration item;
- b. Control changes to those characteristics; and

c. Record and report change processing and implementation status.

Control Software - Software used during execution of a test program which controls the nontesting operations of the ATE. This software is used to execute a test procedure but does not contain any of the stimuli or measurement parameters used in testing a unit under test. Where test software and control software are combined in one inseparable program, that program will be treated as test software. (AFLC 66-37)

Cost/Schedule Status Report - A cost report, applicable to contracts of \$2 million or more and of more than one year duration, which do not use the Cost Performance Report. The Cost/Schedule Status Report provides a means of collecting summary level cost and schedule performance status information on contracts on which Cost/Schedule Control Systems Criteria is not a requirement. The application of the Cost/Schedule Status Report does not involve certain of the unique requirements or disciplines of the Cost/Schedule Control Systems Criteria.

Cost/Schedule Control Systems Criteria - A cost/schedule control system intended to provide an adequate basis for decision making by both contractor management and DOD components. AFSCP/AFLCP 173-5 provides the precise definitions and instructions for the Cost/Schedule Control System Criteria program to be implemented by the contractor. In accordance with the Cost/Schedule Control Systems Criteria, the contractor's internal management control systems must provide data which indicates work progress; which properly relates cost, schedule, and technical accomplishment; which are timely, valid, and auditable; and which supply DOD managers with information at a practical level of summarization.

Data Base - A collection of program code, tables, constants, interface elements and other data essential to the

operation of a computer program or software subsystem.

Estimating Model - A graphical or mathematical representation (of a specific work task) which is utilized to calculate the approximate cost to develop and/or produce a desired product.

Life Cycle Analysis - An analysis of a systems total cost to the government over its full life. It would include the cost of development, production, operation, support, and if applicable, disposal.

Logic Flow - A diagrammatic representation of the logic sequence for a computer program. Logic flows may take the form of the traditional flow charts or in some other form such as a program design language.

Organic - A term used to designate a task performed by the Air Force rather than a contractor.

Program Design Language - An English-like, specially formatted, textual language describing the control structure, and general organization of a computer program. Essential features of a program design language are:

a. It is an English-like representation of a computer that is easy to read and comprehend.

b. It is structured in the sense that it utilizes the structured programming control structures and indentation to show nested logic.

c. It uses full words or phrases rather than the graphic symbols used in flow charts and decision tables.

Quality Assurance - A planned and systematic pattern of all software-related actions necessary to provide adequate confidence that computer program configuration items or products conform to establish software technical requirements and that they achieve satisfactory performance.

Software - A combination of associated computer programs and computer data required to enable the computer equipment to perform computational or control functions.

Source Selection - The process of selecting which among competing contractors shall be awarded a contract. A significant portion of this involves evaluation of proposals to determine the degree to which the government's requirements would be satisfied.

Support Software - Auxiliary software used to aid in preparing, analyzing and maintaining other software. Support software is never used during the execution of a test on a tester, although it may be resident either on-line or off-line. Included are assemblies, compilers, translators, loaders, design aids, test aids, etc. (AFLC 66-37).

System Engineering - The application of scientific and engineering efforts to transform an operational need or statement of deficiency into a description of systems requirements and a preferred system configuration that has been optimized from a life cycle viewpoint. The process has three principal elements: functional analysis, synthesis, and trade studies or cost-effectiveness optimization.

Test Software - Programs which implement documented test requirements. There is a separate test program written for each distinct configuration of unit under test (AFLC 66-37).

Validation - Computer program validation is the test and evaluation of the complete computer program aimed at ensuring compliance with performance and design criteria.

Verification - Computer program verification is the iterative process of continuously determining whether the product of each step of the computer program acquisition process fulfills all requirements levied by the previous step, including those set for quality.

Work Breakdown Structure - A standard method for structuring a program into its various required work tasks. A Work Breakdown Structure is implemented per MIL-STD-881A under the guidance in AFR 800-17. When subdivided as necessary by the contractor to identify tasks associated with a single responsible organization, it provides a basis for contract planning, status determination, and reporting.

Section 10.0 ABBREVIATIONS AND ACRONYMS

ACWP	Actual Cost of Work Performed	CPU	Central Processing Unit
AFAL	Air Force Avionics Lab.	CRISP	Computer Resources Integrated Support Plan
AFLCP	Air Force Logistics Command Pamphlet	CRM	Contract Responsibility Matrix
AFPRO	Air Force Plant Representative Office	CRT	Cathode Ray Tube
AFR	Air Force Regulation	D&D	Design and Development
AFSCP	Air Force Systems Command Pamphlet	DBMS	Data Base Management System
ASPR	Armed Forces Procurement Regulation	DDT&E	Design, Development, Test and Evaluation
ASUPT	Advanced Simulator Undergraduate Pilot Training	DID	Data Item Description
ATE	Automatic Test Equipment	DOD	Department of Defense
ATLAS	Abbreviated Test Language for all Systems	DODI	Department of Defense Instruction
ATPG	Automatic Test Program Generation	ECP	Engineering Change Proposal
BCWP	Budgeted Cost of Work Performed	ESD	Electronic Systems Division
BCWS	Budgeted Cost of Work Scheduled	FACI	First Article Configuration Inspection
C/SCSC	Cost/Schedule Control Systems Criteria	FCA	Functional Configuration Audit
C/SSR	Cost/Schedule Status Report	FY	Fiscal Year
CCP	Contract Change Proposal	GRC	General Research Corporation
CDR	Critical Design Review	HEW	Health, Education and Welfare
CDRL	Contract Data Requirements List	HOL	Higher Order Language
CER	Cost Estimating Relationship	IBM	International Business Machine Co.
CFSR	Contract Funds Status Report	IMS	Integrated Management System
CPR	Cost Performance Report	I/O	Input/Output
		ITA	Interface Test Adapter
		LOE	Level of Effort
		LRU	Line Replaceable Unit

LSI	Large Scale Integrated Circuitry	RSS	Regulations, Specifications and Standards
MEAC	Management Estimate at Completion	SAE	Software Acquisition Engineering
NAA	North American Autonetics	SDC	System Development Corporation
O&M	Operation and Maintenance	SOW	Statement of Work
O&S	Operation and Support	SPO	Systems Program Office
PAR	Problem Analysis Report	STOL	Short Take Off and Landing
PCA	Physical Configuration Audit	T.O.	Technical Order
PDR	Preliminary Design Review	TRD	Test Requirement Document
RCA	Radio Corporation of America	TS	Training Simulator
RD&A	Requirements Definition and Analysis	UUT	Unit Under Test
RDT&E	Research, Development, Test and Evaluation	VAR	Variance Analysis Report
RF	Radio Frequency	VDD	Version Description Document
RFP	Request for Proposal	V&V	Validation and Verification
ROC	Required Operational Capability	VV&C	Verification, Validation and Certification
		WBS	Work Breakdown Structure

Section 11.0 INDEX OF DOCUMENT TOPICS

Air Force Documents	2.0, 3.4.1, 3.5, 4.2, 5.3
Analysis Phase	3.2.2.1, 4.1
ATE Unique	1.3.2, 3.0, 3.4.3, 3.5, 4.0, 4.3
Automatic Test Program Generation	4.3
Changes	3.0, 3.3, 3.4.2.5, 3.6.1, 4.2, 4.3, 4.5, 5.0, 5.1, 5.2, 5.3
Coding and Checkout	3.2.2.4, 4.3
Contractor Cost Data Reporting System	3.2, 3.4.2, 3.4.2.1, 3.4.2.2, 3.4.2.3, 3.4.2.4, 3.4.2.5, 4.0
Cost Accounting	3.4.2.1, 3.4.2.2, 3.4.2.3, 4.5
Cost Estimating	3.0, 3.1, 3.2, 4.3, 4.5, 5.2
Cost Reporting and Analysis	3.4.2.4, 4.2, 4.3, 6.0, 6.1, 6.2
Cost/Schedule Control Systems Criteria	3.4, 3.4.2.4, 3.4.2.5, 3.4.3, 6.1
Detailed Design	3.2.2.3, 4.2
Development Costs	3.0, 3.3, 4.0, 4.3
Dollar Amounts	3.4.1, 4.2, 4.3, 6.1
Estimating Methods	3.2, 4.0, 4.1, 5.2
Estimating Models	3.1.1, 3.1.1.1, 3.1.1.2, 3.1.1.3, 3.1.1.4, 3.1.1.5, 3.1.1.6, 3.1.1.7, 3.1.1.8, 3.1.2, 5.2
Higher Order Language	1.3.1, 3.2.1.7, 3.5, 5.1
Integration and Test	3.2.2.5, 4.4
Life Cycle Costs	3.0, 3.1, 3.3, 3.5, 4.1, 4.5, 5.3
Operational and Support Costs	3.0, 3.1.1.3, 3.1.2, 4.5, 5.0, 5.1, 5.2, 5.3
Preliminary Design	3.2.2.2, 4.1
Software Acquisition Engineering Guidebooks	1.2, 4.0, 4.1, 4.2, 4.5, 5.0

TS Unique

1.3.1, 3.0, 3.4.3, 3.6, 4.0, 4.3

Work Breakdown Structure

3.2.1.1, 3.2.1.5, 3.2.2, 3.4, 3.4.2.1,
3.4.2.2, 3.4.2.3, 4.1, 4.5, 5.2

APPENDIX A: SUMMARY OF EIGHT COST MODELS

Wolverton Model

In November 1973, Ray W. Wolverton of TRW Systems Group presented a paper entitled "The Cost of Developing Large Scale Software." This was not the first attempt at a software cost estimating model, but it has become a standard for software cost estimation. An earlier attempt by System Development Corporation provided an exhaustive quantitative analysis resulting in a data fit for 13 factors based on 169 software projects. The Wolverton model is the most widely used and accepted software cost estimating technique developed thus far. This methodology is applicable to large scale software development programs which utilize a "structured programming" design approach. Structured programming implies modular form.

The basis for the model is a data base containing historical information in the form of cost per instruction. Wolverton assumes that the development cost varies proportionately with the number of instructions. For each identified routine, the procedure combines a user supplied estimate of the number of object instructions, category and relative degree of difficulty, with relationships based on the historical data base, to determine a trial estimate of the total software development cost.

The major pitfall with the Wolverton model lies in the initial estimation of the number of instructions by degree of difficulty and category. Once these estimates are obtained the model is easily applied. Results naturally depend on the accuracy of the initial estimates.

Modified Wolverton Model

AFAL/AAA-3 developed a computerized version of the Wolverton Model for rapid analysis of software development costs.

The only required input to the computer program is the number of instructions by type (control, input, output, algorithm, etc., instructions). The program utilizes ten equations to obtain the cost per instruction for each type. These equations were obtained through regression analysis using prior experience data. Costs associated with the level of effort are computed as follows: (1) total cost of the program is calculated from the number of instructions and cost per instruction by type; (2) analysis is 20% of total cost, design is 18.7%, coding is 21.7%, testing is 28.3% and documentation is 11.3%.

The modified Wolverton Computer Program generates program development costs for "new" and "old" code, for programs ranging in "percent difficulty" from 10-90%. The user, based on subjective decisions relative to these characterizations, selects the appropriate cost figure from the spectrum of data generated.

ESD Model

The summary notes of the October 1974 AFSC Electronic Systems Division (ESD) sponsored software workshop form the basis for what is referred to herein as the "ESD Model".

The primary step in using this model is the determination of the number of delivered executable source instructions; where delivered implies designed, integrated, tested and documented. Source instructions, which for this discussion exclude comment cards, are considered a better estimation factor than the number of object instructions used in the Wolverton and Modified Wolverton models.

Once the number of instructions and the language are known, cost factors are used to arrive at the basic cost figure. Many factors affect the cost estimate, such as whether it is a real-time application program, familiar or unfamiliar program, etc.

The "relation to cost" for several of the factors identified as influencing software cost are listed as "subjective". The size and structure of the data base is an extremely important parameter. Quite naturally, the effect on cost is more for large data file oriented projects but as of yet, no quantitative relationship similar to those developed for cost-per-instruction has been established. The complexity factor has not been defined in a way so as to be reliable in a cost formula. Attempts have been made to correlate costs with such factors as number of interfaces, percentage of branch statements and number of paths through a program, but without any highly reliable correlations. The effect on cost that the development environment has, is the added cost required to adapt software to actual operational conditions, such as different computer configuration and operating procedures. These costs can be quite significant in some instances, but can only be estimated subjectively.

Programmer skill is considered by many experienced estimators to be the most important factor affecting software development costs. Productivity variations of 5:1 between individuals are common. Also, yet to be developed are the quantitative effects on cost of using development techniques such as structured programming, top-down development, chief programmer teams and automated aids. It is generally agreed that systematic approaches to software development are better than disorganized ones. Payoffs for the use of systematic software development techniques are in both development costs and operation and maintenance costs.

To sum up the ESD approach, the basic cost is arrived at by utilizing the number of instructions times the cost per instruction and adding cost for type of program, (unfamiliar, real-time, etc.). Subjective factors are then applied to adjust cost to reflect the development technique, personnel, etc.

The Tecolote Model

The Tecolote Model refers to the basic equations extracted from a report by Brad C. Frederic entitled, "A Provisional Model for Estimating Computer Program Development Costs," December 1974, prepared by Brad C. Frederic of Tecolote Research, Inc., Santa Barbara, California, for the Resource Analysis Branch, Office of the Chief of Naval Operations, Department of the Navy.

The report emphasized the problem of obtaining data to perform statistical analysis and noted that three large software cost data bases had been already compiled at System Development Corporation (SDC), TRW, and North American Autonetics (NAA). There were problems in the data collected by Tecolote (387 separate points from 15 source references) that proved insurmountable. Since the data had to be collected from rather outdated published sources, locating original researchers to interpret the data was not feasible. Therefore, the data base could not be properly qualified or normalized. Hence, Tecolote elected to undertake a small sample approach (5 data points) utilizing only data which were thoroughly understood, and where "the estimating relationships developed would be more in the nature of engineering scaling laws than strictly derived statistical equations."

The Tecolote analysis of software development included the following activities:

- a. Software requirements generation.
- b. Preliminary software design (and release).
- c. Detailed software design (and release).
- d. Code and debug.
- e. Development testing.
- f. Validation testing.

g. Operation demonstration (and handover).

The types of computer architectures which this study included were single CPU, democratic, and autocratic. Single CPU involves a single central processor with storage and peripherals. The democratic architectures consider multiple CPU's operating in parallel with pairwise communication, common storage and peripherals. Autocratic is a combination of single CPU's and democratic subsystems acting in a parallel, under the control of a separate single CPU executive. Mr. Frederic noted that computer system speed and fast storage capacity are the major driver of software requirements. The size of the program in this model is the number of machine language instructions. The size can be input as either the number of operational instructions or the number of delivered instructions.

Aerospace Model

The model referred to here as the "Aerospace Model" was taken from a 1975 Aerospace Corporation report on cost estimating. The data used to develop the cost equations for this model were divided into two groups or types of programming efforts, real-time programs and support programs. Included in the cost data are costs that accrued as a result of problems encountered in developing a large-scale software program. The real-time software program development problem areas identified were:

- a. Limited core storage of computers.
- b. Timing requirements.
- c. Accuracy requirements.
- d. Fixed-point arithmetic.
- e. Changing specifications.
- f. Real-time simulations.

(1) Inability to interface languages.

(2) Non-standardization of computers between machines.

Operational program or support program problem areas identified were:

a. Timing and accuracy problems.

b. Inability to transfer simulation activities of one contractor to another due to language and machine differences.

c. Inadequate and changing specifications.

d. Lack of an organized method of defining endpoints and of various development phases.

The data base used to develop the cost equation for real-time software program costs, consisted of 13 large-scale programs (primarily airborne and space oriented programs). The cost equation derived from a regression analysis of those 13 data points.

$$\text{Man-months} = 0.057 (\text{Instruction})^{0.94}$$

The sample size for operational support programs consisted of 7 data points (both airborne and ground software programs were in the data base). The resulting equation for support software man-months estimation is:

$$\text{Man-months} = 2.012 (\text{Instruction})^{0.404}$$

The comment about language type mentioned above holds true in this case as well.

Once the number of man-months required for the development is estimated using both equations, a dollar value per man-month is used to derive the total development cost. The estimated cost per man-month would obviously vary with the particular company performing the programming function. For planning purposes, an average of \$5,000 per man-month is used by Aerospace.

GRC Model

The GRC model was taken from a report entitled "Estimation of Computer Requirements and Software Development Costs", March 1974, prepared by M. A. Tabach and M. C. Ditmore of General Research Corporation. The purpose of GRC was to determine a means of quantifying computer software development costs from overall system requirements. GRC had previously developed a procedure for determining the data processing speed and memory required to implement various computer functions from system performance requirements. The report presents a cost estimating relationship (CER) for computer software development which models the effects of the following: (1) program size, (2) computer language, (3) complexity, and (4) hardware constraints. The key conclusion of the report was that the program size, used along with the effects of program complexity, high-level language, and hardware constraints, is a reasonable predictor of software development costs.

Price Software Model

The PRICE Software Model (PRICE S) applies RCA's parametric cost-modeling methods to help managers and analysts estimate costs for computer software development. The key ingredients of this philosophy are:

- a. Interactive operations, with conversational input and output.
- b. A parametric approach, derived from experience and supported by empirical evidence.
- c. Efficient problem description, with a small set of easily comprehended input factors.
- d. Internal self-checking for consistency of input parameters.
- e. Customizing flexibility, so that users may adapt the model to match individual experience and applications.

Configuration and reliability requirements are incorporated, together with technology growth and inflation, to derive representative values for five cost categories in each of three overlapping development phases. The cross-classifications used are:

<u>Cost Categories</u>	<u>Development Phases</u>
* Systems Engineering	* Engineering Design
* Programming	* Implementation
* Configuration Control	* Test and Integration
* Documentation	
* Program Management	

PRICE S also derives typical schedules for the work to be accomplished. The user may accept these schedules or specify alternative schedules of his own. User specified schedules are examined internally, and costs are adjusted to account for apparent schedule accelerations, stretch-outs and phase transition inefficiencies. These adjustments are made with reference to representative resource expenditure profiles, which the user may adjust to fit his needs.

Three modes of operation are available: Normal Operation, ECIRP and GEOSYN. The Normal mode is used to compute costs directly from user inputs. ECIRP, in contrast, enables PRICE S to be run "in reverse" to calculate complexity factors from known project costs. Complexity factors remain constant regardless of the magnitude of the scope of work or performance schedule variations. The factors are used to calibrate the model to reflect experience resources as appropriate to specific classes of computer programs or their application type. Within the model, the complexity factors are adjusted as well as technological time variations. The ECIRP mode preserves all normal PRICE S relationships, and its reverse-operation capability

greatly facilitates user/model calibration. GEOSYN is similar to ECIRP, except that specified costs are used to compute typical program sizes and project schedules. The GEOSYN mode is used to investigate feasibilities and to set goals for design-to-cost efforts.

The underlying principle of PRICE S is that all estimates involve all comparative evaluation of new requirements in light of analogous histories. It has been designed for use by experienced managers and analysts, to assist them in translating experience and judgement into reliable, self-consistent, timely cost estimates. Price methodology provides a convenient way of reducing empirical data to a few principal variables, each of which can be readily adjusted to account for technological and economic differences between individual projects and organizations.

Boeing Model

Boeing Document D180-20144-1, a Preliminary Study on Estimating the Development Costs of Software, by R. H. Darrow, describes a model which is perhaps easier to use than some of the

models listed above, since only one parameter must be estimated. This model was developed empirically by evaluation of a number of complex software systems which Boeing had developed over a period of time. This model is, simply:

$$\text{Software Development Costs} \\ (\text{man-hours}) = KI^2$$

where,

$$0.30 \leq K \leq 0.33$$

with real time systems being near the high end; test software, data processing, etc., software being near the low end.

I = Total number of inputs, plus total number of outputs, (i.e., the total number of manipulated quantities). For example, a program written to calculate:

$$Y = AX^2 + BX + C \quad \text{has 4 inputs (A, B, C} \\ \text{\& X) and one output (Y). For this} \\ \text{example } I = 5.$$

The referenced document describes this method in detail and reports on the experience data used to generate this empirical model.

APPENDIX B: WORK BREAKDOWN STRUCTURE (WBS) TASKS

PHASE I - Requirements Definition and Analysis (RD&A)

This involves definition of what the system (ATE or TS) is to achieve. Results of Requirements Definition and Analysis include:

- Definition of the scope and objectives of the proposed Ground System software
- Feasibility Determination based on system requirements and an approach satisfying those requirements
- Software development plans

General elemental tasks and sub tasks are:

Document User Needs

Review and Document User Needs
Document Organization and Environment
Document Existing Methods and Procedures
Develop a Glossary

Define Project Scope and Objectives

Define System Boundaries
Define System Interfaces
Define Project Scope
Define Project Objectives
Define Technical Objectives

Define System I/O and Data Base Requirements

Identify Output Requirements
Identify Input Requirements
Identify Data Base Retrieval Requirements
Identify Data Base Maintenance and Update Requirements

Define System Processing Requirements Objectives

Define Input and Output Functional Relationships

Define System Processing Requirements Objectives (Cont)

Define Algorithm Derivation Information
State System Boundaries
List Volume and Throughput Expectations
State Qualitative Expectations

Define Developmental Requirements

Define Developmental Constraints
Document Developmental Assumptions
Define Developmental Responsibilities
Define Documentation Requirements
Define Conversion Requirements
Define Portability Requirements
Define Interface(s) With Other Systems
Define Testing and Acceptance Criteria
Define Firm Versus Optional Requirements
Specify System Expansion Requirements
Define Quality Assurance Requirements
Define Developmental Priorities

Define Operational Requirements

Describe User Interface to System Requirements
Define Training Requirements
Define Human Factor Requirements
Define Administrative Constraints
Document Assumptions
Define Computer Constraints (if any)
Define Recovery and Restart Requirements
Define Security Requirements
Define Installation Requirements
Define System Reliability Requirements
Define System Software Maintenance Requirements
Define Priorities Among Operational Requirements

Confirm Requirements

Review Requirements for Completeness
Interpret Requirements
Coordinate With Using Command

Analyze Resources, Constraints,
Assumptions, and Objectives

Evaluate Available Resources
Evaluate Constraints
Examine Resources and Constraints for
Trade-offs
List Hardware Assumptions
List Time Frame and Schedule
Assumptions
List Personnel and Facility
Assumptions
Check Assumptions for Conflicts
Develop Performance Criteria from
Objectives

Analyze System Outputs, Inputs and
Functions

Initiate Data Dictionary
Review the Output Description
Identify Data Elements Needed to
Produce Outputs
Analyze Relationships Among Data
Elements
Identify Input/Output Handling
Functions
Identify Input Validation Functions

Determine System Capability
Requirements and Approaches

Specify General System Capability
Requirements
Determine Approaches to Satisfy
Capability Requirements
List Advantages/Disadvantages for
Each Approach
Prioritize Advantages and
Disadvantages
Indicate Costs/Benefits for Each
Approach
Consider all Approaches

Evaluate and Select System Approach

Determine Parameters to Evaluate
System Approach
Develop Evaluation Criteria
Select Comparative Analysis Method
Compare Each Approach
Evaluate Comparison
Recommend System Approach

Determine Installation and
Conversation Requirements

Determine the Amount of
Non-Machine-Sensible Data
List Differences Between Existing and
New Files
Sample and Analyze Existing Data
Identify Resources Needed for Data
Conversion
Identify Programs/Procedures to be
Converted
Determine Hardware Preparation
Resources
Determine Forms Preparation Resources
Determine Implementation Approach

Prepare Project Proposal

Identify Requirements by Phase
Organize Plan Elements Using Network
Techniques
Identify Critical Milestones
Define Initial Life Cycle Cost
(Development, Etc.)

Conduct Design Review

Review with EDP Management and
Technical Personnel
Review Estimates and Preliminary
Plans
Get Authorization to Proceed

Administer RD&A Phase

Prepare Work Authorizations
Prepare Staff Reports
Prepare Schedule Status Reports
Analyze Resource Consumption Against
Project Plan
Process Change Requests

Plan Preliminary Design Phase

Prepare Phase Task List
Structure Phase Work Flow
Develop Estimates for Tasks
Develop Task Schedules

Plan Remainder of Project

Prepare Task List for Phases 3
Through 6

Plan Remainder of Project (Cont)

Structure Work Flow for Phases 3
Through 6
Develop Estimates
Develop Schedules
Update Project Plan

Setup Phase Review

Prepare Review Material
Distribute Review Material
Arrange for Review Facilities
Make Necessary Travel Arrangements

Initiate Preliminary Design Phase

Review Project Objectives, Goals and
Achievements
Arrange for Staff
Secure Funding Authorization
Acquire Facilities and Materials
Brief Personnel

PHASE 2 - Preliminary Design

This phase involves consideration alternatives, data base definition, development of implementation, test and training plans. Specific candidate tasks are as follows:

Develop Solutions

Develop Computing Solution
Determine Evaluation Approach
Prepare Narrative of Each Alternative
Prepare Cost/Benefit Analysis for
Each Alternative
Prepare Technical Evaluation for Each
Alternative
Select Computing Solution
Prepare Justification of Selection

Define Overall System Environment

Specify Data Communication
Requirements
Specify Data Base Requirements
Determine the Type of Processing
Required
Specify Time Slots in which System
Will Operate

Define Overall System Environment
(Cont)

Specify Protection and Control
Requirements
Establish System Naming Conventions

Describe Subsystems and Interface
Requirements

Identify Possible Subsystem Divisions
Describe Subsystem Divisions and
Elements
List Each Subsystem's Input and
Output Data Items
Extend Data Dictionary to Non-Data
Base Elements
Determine Best Source of Inputs for
Each Subsystem
Develop User Description of External
Data

Prepare Subsystem Preliminary
Component Design

Identify Subfunctions within
Components
Define Input and Output Sets
Define Operational Sequences and
Conditions
Analyze and Verify Decomposition
Document Component Design
Repeat Design Steps for Each
Component
Examine Other Decompositions
Allocate Processes Between Automatic
and Manual/Operation Based on
Selected Criteria

Identify Human Factors in Design

Identify Man/Machine Interface(s)
Identify Problem Areas in Manual
Subsystems
Identify Training Requirements for
Manual Subsystems
Analyze System for Human Utility

Develop Logical Data Base Design

Define Purpose of Data
Define Data Description
Describe Data Content
Refine Data Dictionary

Develop Logical Data Base Design
(Cont)

Identify Data Functional Relationships
Develop Data Security Specifications
Identify Access Technique
Identify Data Volume

Specify Software, Hardware, Data Base Management System (DBMS) and Language(s)

Specify System Software
Specify Hardware Configuration
Specify DBMS Requirements
Specify Programming Language(s) Requirements

Finalize Requirements and Prepare Design Document

Identify New or Modified Information Complete Requirements Definition
Develop Conversion/Installation Plan
Develop Resource Requirements
Develop Test Plan and Training Plan
Refine Cost/Benefit Analysis
Prepare Detail Estimates for Next Phase
Refine Total Project Estimates
Prepare Certification Plan

Conduct Preliminary Design Review

Review Alternatives Examined
Review Design and Data Base Approach
Review Feasibility with Technical Personnel
Review with Customer and Revise as Required
Get Authorization to Proceed

Administer Preliminary Design Phase

Conduct Team Technical Reviews
Conduct Reviews of Technical Process
Release Work Authorizations
Prepare Staff Reports
Prepare Schedule Status Reports
Analyze Resource Consumption Against Project Plan
Process Change Requests

Plan Detail Design Phase

Prepare Phase Task List
Structure Phase Work Flow
Develop Estimates for Tasks
Develop Task Schedule

Plan Remainder of Project

Revise Task List
Refine Phase Work Flow
Refine Task Estimates
Update Task Schedules
Update Project Plan

Setup Phase Review

Prepare Review Material
Distribute Review Materials
Arrange for Review Facilities
Make Necessary Travel Arrangements

Initiate Detail Design Phase

Review Project Objectives, Goals and Achievements
Arrange for Staff
Secure Funding Authorization
Acquire Facilities and Materials
Brief Staff

PHASE 3 - Detailed Design

This provides detailed design of the ground system software including flow charts, acceptance test plans, detailed data descriptions etc. Specific candidate tasks are:

Develop Human Procedures and Interfaces

Analyze the Human Processes to the Detail Level
Document the Human Processes
Develop List of Possible Human System Failures
Develop Failure Corrective Procedures
Design Form Layout and Content
Design Printer Outputs
Design Video Displays

Design Data Base

Group Data Fields
Categorize Data Fields
Develop Record Layouts
Examine Record Layouts for Redundancy
Establish Priorities for Optimization
Perform File Structure Optimization
Specify Access Keys
Specify the User Interface and Procedures

Design System/Subsystem Structure

Identify Subsystem Data Flow
Decompose Data Flow
Document Subsystem Structure

Design Subsystem Security Features

Document and Analyze Possible System Failures
Develop Audit Trail Specifications
State Data Base and Data Reconstruction Requirements
Develop Fallback and Recovery Specifications
Identify Internal Error Checks
Specify the Facility Security Measures to be Used
Specify Hardware Failure Monitoring Programs
Specify Data Privacy Approach

Develop Subsystem Hierarchical Component Design

Identify Subprocesses Within Component
Define Input and Output Sets
Define Execution Sequences, Conditions, and Outputs
Analyze and Verify Component Design
Document Component Design
Document Component Test Plan
Perform Review of Component Design
Repeat Design Steps for Each Component
Perform Component Design Optimization
Define Module Boundaries

Verify System/Subsystem Design

Verify Interface Definitions
Verify Communication Structure and Flow
Examine Module Boundary Definitions
Review System Layout Specifications

Specify Software Utilities and Common Routines

List Utilities Required and Their Specification
Obtain Required Utilities
Examine Design for Common Routines
Document Common Routines in a Catalog

Develop System/Subsystem Test Plan

List the Specific Activities to Test the Subsystem
Specify Expected Results and Evaluation Procedures
Specify Test Configuration
State Volume and Characteristics of Test Data
Identify Testing Tools
Produce Schedule Covering all Test Cycles

Develop Software Construction Plan

Identify Minimum Configuration
Specify Order of Construction
Determine Construction Schedule and Resources
Identify Required Programming Aids
Identify Required Technical/Programming Skills
Update Installation Plan
Refine Resource Requirements

Conduct Detail Design Review

Review Technical Merit of the Design Personnel
Review the Design Cost and Schedule Performance
Review the Construction Plan
Review with Client
Get Authorization to Proceed

Plan Remainder of Project

Complete Task List of Phases 4
Through 6
Complete Work Flow for Phases 4
Through 6
Complete Estimates
Complete Schedule
Complete Project Plan

Setup Phase Review

Prepare Review Material
Distribute Review Material
Arrange for Review Facilities
Make Necessary Travel Arrangements

Initiate Software Construction Phase

Review Project Objectives, Goals, and
Achievements
Arrange for Project Staff
Secure Funding Authorization
Acquire Facilities and Materials
Brief Staff

PHASE 4 - Software Construction

The objective of this phase is to develop and test ground system computer programs and prepare for certification. At the completion of this phase all software modules will have been coded, integrated and tested. Specific tasks are:

Establish Support Libraries

Set Up Developmental Support Library
Set Up System Support Library
Set Up Test Data Support Library

Perform Construction of Components

Translate Component Design to Code
Desk Check Component Source Code
Perform Review of Component Source
Code
Prepare Component Test Data
Add Component Source Code and Test
Data to Library
Conduct Component Test Results
Integrate and Test Component with
System

Perform Construction of Components (Cont)

Document Component Integration Test
Results
Repeat Construction Steps for Each
Component

Perform Final System Test

Perform Data Base Functional Test
Perform System Functional Test
Conduct Capacity Test
Demonstrate Production Reliability
Test Fallback, Recovery, and
Reconstruction
Stress Test the System
Conduct System Performance Test
Analyze and Document Test Results

Complete Program Technical Document

Determine Actual Storage Requirements
Produce Storage Maps
Include Listings and Results
Produce Cross References
Include Performance Measurements
Describe Maintenance Factors

Complete Certification Test Cases

Complete User and Computer Operations Documents

Finalize User Document
Finalize Computer Operations Document
Prepare Installation Document

Develop Training Materials

Identify Training Objectives
Identify Training Media
Prepare Course Objectives
Prepare Lesson Plans
Develop Course Support Materials
Validate Training Materials

Conduct Software Construction Review

Review Project Implementation Plan
Prepare Project Report
Review the Construction Deliverables
Review Design Deviations (if any)
Review Test Results

Conduct Software Construction Review
(Cont)

Review with Client and Revise as
Required
Get Authorization to Proceed

Administer Software Construction
Phase

Conduct Incremental Construction
Reviews
Conduct Reviews of Technical Progress
Release Work Authorizations
Prepare Staff Reports
Prepare Schedule Status Reports
Analyze Resource Consumption Against
Project Plan
Process Change Requests
Resolve Problem Reports

Setup Phase Review

Prepare Review Material
Distribute Review Material
Arrange for Review Facilities
Make Necessary Travel Arrangements

Initiate Software Certification Phase

Review Project Objectives, Goals, and
Achievements
Alert Certification Committee
Arrange for Independent Third Party
Secure Funding Authorization
Acquire Test Facilities and Materials
Brief Project and Test Personnel
Complete Test Site Negotiations

PHASE 5 - Software Validation and
Verification (V&V)

The objectives of this phase are to
demonstrate the entire software system
performance in accordance with its speci-
fications. Specific tasks are:

Perform Validation Testing

Finalize Test Plan and Procedures
Review and Approve Test Plan
Perform Validation Testing

Conduct Validation Test Review

Review Test Results
Identify Test Action Items
Correct Deviations/Comply with Action
Items
Document Response to Test Results

Perform Verification (Qualification)

Finalize Verification Test Plan
Review and Approve Test Plan
Perform Verification (Qualification)
Test

Conduct Verification Test Review

Review Test Results
Identify Test Action Items
Correct Deviations/Comply with Action
Items
Document Response to Test Results

Perform Training Materials V&V

Finalize Test Plan
Review and Approve Test Plan
Perform Training Materials Test

Conduct Training Materials V&V Review

Review Test Results
Identify Test Action Items
Correct Deficiencies/Comply with
Action Items
Document Response to Training
Materials Testing

Finalize Installation/Conversion Plan

Review Project Development and
Problems
Prepare Plan to Handle Problems
Prepare List of Available Resources
Review Change Control Procedure(s)
Revise Schedule of Pre-Conversion
Activities
Revise Schedule of Pre-Installation
Activities
Complete Installation Schedule

Conduct V&V Review (FACI, Etc.)

Administer Software V&V Phase

Conduct reviews
Transmit Data Packages
Complete or support Testing Requirements
Release Work Authorizations
Prepare Staff Reports
Prepare Schedule Status Reports
Analyze Resource Consumption Against Project Plan
Implement Personnel Phase-Out Plan

Set Up Phase Review

Prepare Review Materials
Distribute Review Material
Resolve Problem Reports
Arrange for Review Facilities
Make Necessary Travel Arrangements

Initiate Installation Phase

Review Project Objectives, Goals, and Achievements
Arrange for Installation Team
Secure Funding Authorization
Arrange Facilities and Materials
Brief Installation Team

PHASE 6 - Installation/Delivery

This phase provides for delivery and installation of the ground system software at the using command's installation and initiation of user performed maintenance activities. Specific tasks are:

Conduct User Orientation on System

Conduct High-Level Introduction
Discuss Conversion
Discuss Installation
Formalize Communication Procedures
Answer Questions

Train User and Support Functions

Train User
Train Production Operators
Train Maintenance Team

Install System

Allocate Space for New System
Install System
Perform Installation Demonstration Tests

Certify System for Maintenance

Develop Maintenance Plan
Review System Deliverables
Install System Support Libraries

Certify the System for Production

Perform Acceptance Test
Perform Production Cycles

Obtain Final User Acceptance

Review Installation Reports
Review Training Results
Review Production Documentation
Review System Maintenance Procedures
Turn System Over to User

Administer Installation Phase

Conduct Reviews
Conduct Reviews of Installation Progress
Release Work Authorizations
Prepare Staff Reports
Prepare Schedule Status Reports
Analyze Resource Consumption Against Project Plan
Ensure Development of Maintenance Plans
Identify Maintenance Team Personnel

Set Up Phase Review

Prepare Review Material
Distribute Review Material
Arrange for Review Facilities
Make Necessary Travel Arrangements
Arrange for Formal Turnover of Installed System
Finalize Project File

Initiate Maintenance

Finalize Maintenance Plan
Identify Maintenance Personnel
Finalize System Deliverables

APPENDIX C: INTEGRATED MANAGEMENT SYSTEM (IMS)

1.0 CONTRACTOR ORGANIZATION AND WORK BREAKDOWN STRUCTURE (WBS)

Each major contract normally becomes a contractor's uniquely identified program. The program has its own staff whose sole purpose is that program. The program receives direct support from the functional organizations within the contractor's company (engineering, manufacturing, materiel, quality control, finance, facilities, and industrial relations). The program may also receive support from other of the company's divisions and outside subcontractors. The contract, which drives the program, contains the system specifications, the statement of work, the Contract Data Requirements List (CDRL) and the WBS. The WBS is the basis for all IMS data structuring and is the basis planning tool, data base, cost/accumulation/accounting base, and schedule base. It takes the end product and breaks it down into nearly the smallest system components.

From the WBS and the functional organizations (plus subcontractors) come two further breakdowns: the Contract Responsibility Matrix (CRM) and the Cost Account. The CRM (See Figure C-1) assigns each of the WBS levels (e.g. engine fan) to one or more functional organizations/subcontractors, indicating responsibility but rarely a cost account. The cost account (See Figure C-2) is the assignment of lower level WBS elements to responsible lower level functional managers. This is where task definition of WBS, scheduling, budgeting, work authorization, cost accumulation, earned value, and future data are integrated; where variance analysis is performed, and where authority and responsibility for control and corrective action of cost/schedule management exists. Every cost account is subdivided into work packages. Work packages are basic building blocks, the lowest level of work identified, that are normally of rather short duration. Work packages

have work content that is clearly distinguished from all other work packages, have scheduled start and completion milestone dates, and have a budget expressed normally in dollars or manhours. Cost accounts can also be subdivided into level-of-effort (LOE) and apportioned effort. LOE is effort of a general or supportive nature, such as engineering liaison, which does not produce definite end products or results that are discretely measurable. LOE activity is segregated from work package effort, except for minor amounts, to avoid distorting measurable work packages. Apportioned effort is effort, such as quality control, that by itself is not readily divisible into short-span work packages, but which is related in direct proportion to measurable effort, such as manufacturing labor. The calculation of earned value for a cost account and its subdivisions is covered later in the analysis area.

The program's Contracts organization maintains control through the program manager of all work to be done via a work authorization. This authorization gives authority to functional organizations to proceed with work based on USAF authorizing action. Each functional organization then issues its own particular lower-level authorization, after identifying cost account statements of work and their related schedules and budget authorized by the contract.

2.0 SCHEDULING AND BUDGETING

Scheduling, or planning, involving the use of milestones or completed schedule phases, normally consists of at least four levels of schedules. Tiers I and II are program management schedules. Tier I is the master schedule containing only major milestones of the total program. Tier II contains all significant WBS milestones to accomplish each WBS task and aids the program manager in running the program. Tiers III and lower are operating schedules. Tier III is oriented to the functional organization and

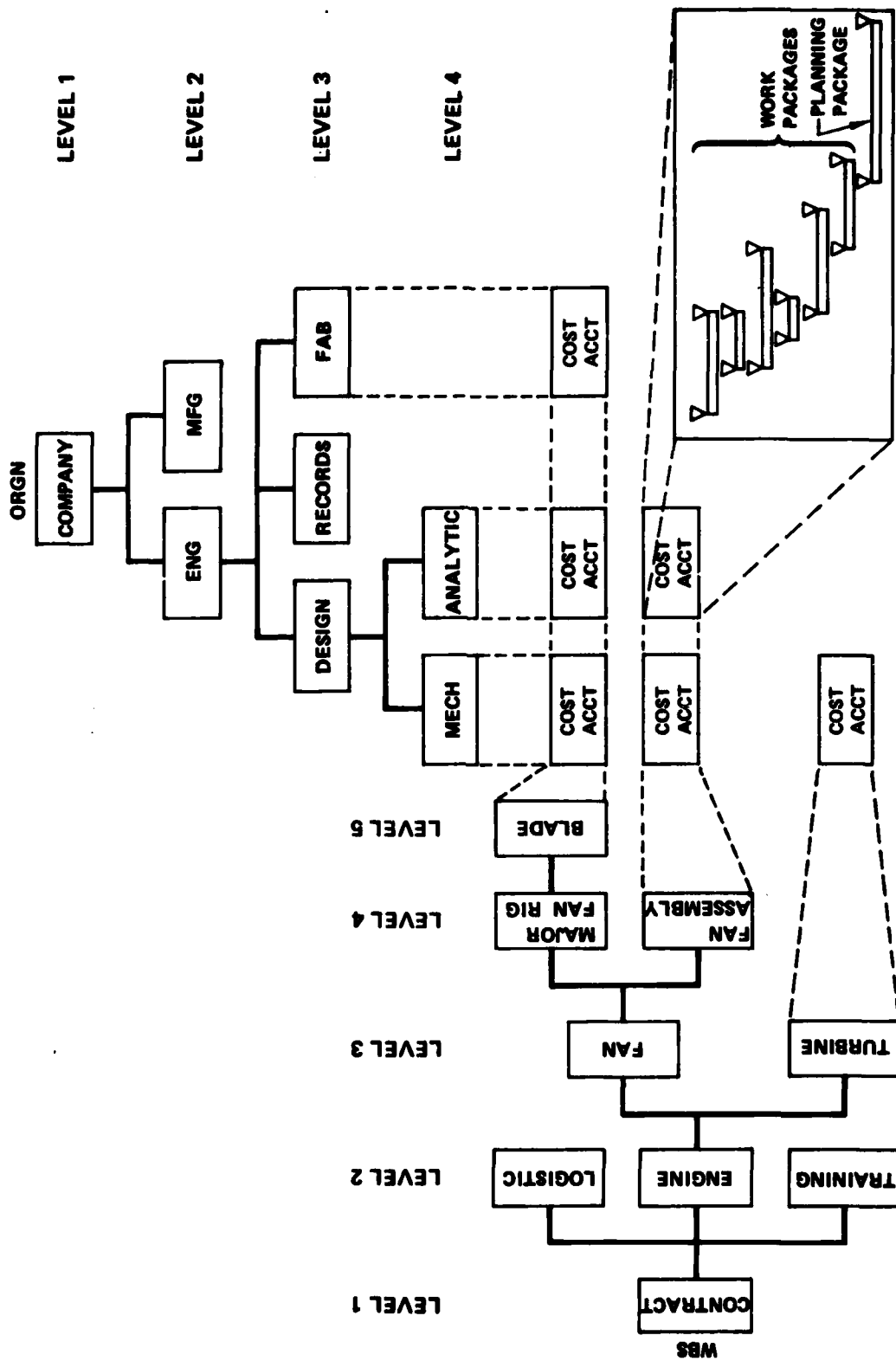


Figure C-2. Cost Account Definition

major WBS elements including major milestones. Tier IV and lower schedules are oriented to cost accounts and the applicable work packages (see Figure C-2). The lower the tier schedule, normally the shorter the length of the work involved.

Budgeting starts with the contract budget base, normally a lump sum value, which is the total authorized contract budget; which also includes overhead, undistributed budget (budget not yet distributed to a cost account), management reserve (budget assigned to a program and/or functional manager for unplanned work contingencies and motivation for underrunning costs), and authorized but not yet negotiated changes. The budget is allocated in accordance with the contract by the program manager, to the functional manager and then to the cost account managers.

Changes to the program contractual schedule and/or budget require the approval of the program manager and/or functional manager and/or the USAF procuring activity. The application of budget values to schedules provides a common measuring tool for both cost and schedule performance. This, in addition to not only the variances from planned cost and schedule but also the revised future data as of the latest monthly figuring, provides the primary outputs of the reports listed in Section 6.

3.0 ACCOUNTING

Accounting in this area is merely a means of identifying and accumulating direct or indirect costs via both the WBS sort (as used by most SPOs) and the contractor's organization sort. The IMS uses work orders that will associate all costs with a particular contract and related WBS items. Therefore the work order allows for sorting and reporting cost data by WBS. The work order is part of the accounting charge number that also identifies organization data which results in also providing functionally oriented reporting.

Costs are accumulated as either direct or indirect. Direct costs can be directly related to work accomplished and/or material used. These costs can be measured at the time they occur. Indirect costs cannot be consistently or economically identified directly with specific contracts. Indirect costs, or overhead burden, are allocated to contracts on an AFPRO negotiated functional rate or percentage basis.

4.0 ANALYSIS

Analysis is the link that makes IMS a management tool rather than just another reporting system. This implies that action must be taken to make analysis work for management. Some USAF officers believe that the contractors' use of a C/SCSC system will find and correct problem areas before they occur. C/SCSC can do this only to the extent that these problem areas are found and analyzed during the course of daily operations. Monthly analysis will document the problem areas quantitatively for use by upper management and government reporting. There can be no perfect total analysis unless computer runs of performance measurement are accomplished daily, which, due to the amount of input/output, is virtually impossible.

The areas of performance measurement, earned value determination, cost/schedule indices calculations, cost/schedule at-completion variance calculation, and variance analysis are major parts of analysis, the crux of the feedback loop for managerial action. Being the crux and also the main ingredient of the reports covered in Section 6, there is no way to cut short the discussion of analysis or the understanding thereof.

Performance measurement consists of the determination of earned value, the computation of the cost and schedule indices and the computation of cost, schedule, and at-completion variances. Performance measurement should be completely objective, and provides the basis for variance analysis, which is

the interpretation of the data for possible initiation of corrective action. Budgeted cost of the work scheduled (BCWS) is the indicator of planned progress, the baseline plan for performance measurement, and the time-phased and work-phased budget, i.e., the budget is phased with the measuring schedule and milestones. Budgeted cost of work performed (BCWP) is the indicator of actual progress, the "earned value", i.e., the BCWS for milestones that have been accomplished or are in the process of being accomplished. The earned value concept is central to the cost/schedule integration process, being the technique of applying BCWS or "budget" to a time-phased work plan or schedule to provide a quantified baseline against which to measure work performance. As work is accomplished the pre-assigned BCWS for that work is considered earned (BCWP).

In IMS at the cost account level, or in certain cases at the lower work package level, there are five techniques for measuring earned value, based on the length of time between milestones. Cost accounts can be subdivided into measured (work packages) work, LOE, and apportioned work. In the case of the variant milestone, apportioned and LOE techniques, value is also earned at the end of each month for completed portions of work in process. Measurable work is measured at the work package level and summarized at the cost account level. LOE and apportioned work is normally measured at the cost account level. Minor amounts of LOE may be mixed with measured work within a cost account, but not within a work package. The method must be determined and documented prior to starting work, and normally cannot change when work commences.

<u>Usual time span</u>	<u>Technique</u>	<u>Calculation Method</u>
1 month or less	100% rule	Earn BCWS when work package is completed.

2-3 months between milestones	Variant Milestones	First month's BCWS earned as work starts. Final month's BCWS earned when work package is completed. Interim month's BCWS earned when milestones are completed, or in isolated months, when meaningful milestones are not feasible, BCWS earned because work package is still in progress.
	50% rule	Earn 50% of work package BCWS when work starts; balance when completed.
No Limitation	Level-of-Effort (LOE)	BCWS earned by passage of time after a certain related task is reported as started.
	Appor-tioned	BCWS earned in direct relationship to earning of related effort on which based.

Figure C-3 graphically portrays the BCWP (earned value) determination process using one of the techniques (the Variant Milestone method). As seen near the bottom of the chart, just above the tabular data, it consists of one cost account made up of a number of completed and near-term work packages and a planning package. The planning package is a future work package not yet scheduled and budgeted in detail. The section just above these indicators shows a typical Tier IV or lower schedule. Each numbered bar is a work package. The milestones

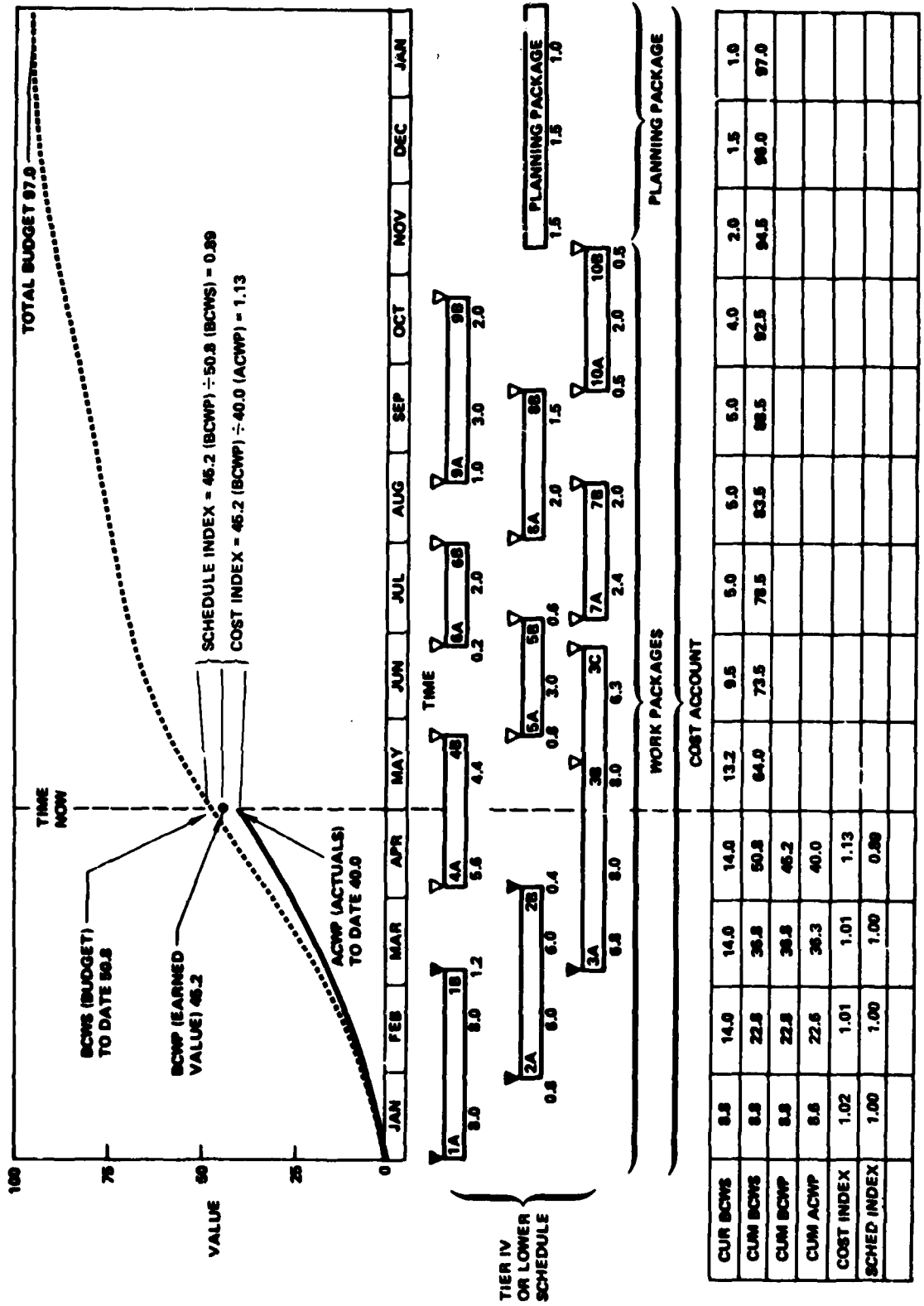


Figure C-3. Cost/Schedule Indices Based on Variant Milestone BCWP Measurement Method

are discretely identified by an alpha designator following the work package number for ease in reference in schedule status reports, etc. The numbers under the work packages are the BCWS by month expressed in manmonths, as planned by cost account managers. The sum of the manmonths for all work packages in work during a given month equals the current BCWS in the tabular data below. This summary BCWS line must equal the cost account budget line. In the upper chart, the total budget is time-phased as shown in the ascending horizontal dotted line.

In this example, you can see that the variant milestone method is a form of exception reporting. In this method only the variant milestones, i.e., those completed ahead or behind schedule, are considered, and the corresponding values are added to or subtracted from the cum BCWS. Milestone values are derived from the BCWS. The BCWS value for a given month and a given work package is divided by the number of measuring milestones within that work package and month to determine the milestone values. For example, in Figure C-3, milestone 4A was still incomplete at "Time Now." Therefore, no BCWP is earned for that work package in April. So the 5.6 manmonths are subtracted from the 50.8 cum BCWS planned for this cost account, giving a BCWP of 45.2 manmonths.

Should there not be a milestone in a given month, e.g., April for work package 3, and all previous milestones are completed, the BCWP is earned since the work is still in progress.

Figure C-3 also portrays the cost/schedule indices calculation process. The ACWP (actuals) data is now required. BCWP (earned value) is useful only when compared with ACWP or BCWS (budget). This is accomplished by means of cost/schedule indices and cost/schedule variances (see Figure C-4) which puts both cost and schedule performance on a mathematical basis, permitting ready analysis, trend evaluation, initiation of remedial action, and assessment of man-

agerial performances. The cost (or value) index is determined by dividing the cum BCWP by the cum ACWP. A cost index greater than 1.0 indicates favorable performance; less than 1.0 unfavorable performance. Since these indices are based on manloaded schedules, they depict more than favorable/unfavorable cost performance. An index greater than 1.0 indicates a favorable cost performance was accompanied by favorable schedule performance. An index less than 1.0 conversely indicates the real cost impact of being behind schedule. The schedule index is determined by dividing the cum BCWP by the cum BCWS. A schedule index greater than 1.0 indicates an ahead of schedule condition; less than 1.0 behind schedule. But it again tells more because the schedule variances are quantified. Cost variance is calculated by subtracting ACWP from BCWP. Schedule variance is calculated by subtracting BCWS from BCWP.

Figure C-4 portrays cost/schedule at-completion variance calculation. Management Estimate at Completion (MEAC), or what was earlier referred to as future data, is now supplied by each cost account manager, taking into account past performance and anticipated actions coordinated through the program and functional managers. (See Figure C-5). The at-completion variance is calculated by subtracting MEAC from the total budget. Whereas the cost/schedule indices are indicators of performance progress, cost/schedule/at-completion variances emphasize the degree of variance; and when the variances exceed predetermined thresholds, they become the basis for variance analysis. The system must not suppress variances. Lack of systems discipline only results in delaying visibility of cost/schedule problems which will ultimately surface in another way. These variances are not always a result of performance. They may result from estimating errors, economic factors, etc.

Analysis is a continual process: daily, weekly, and monthly, as shown in the

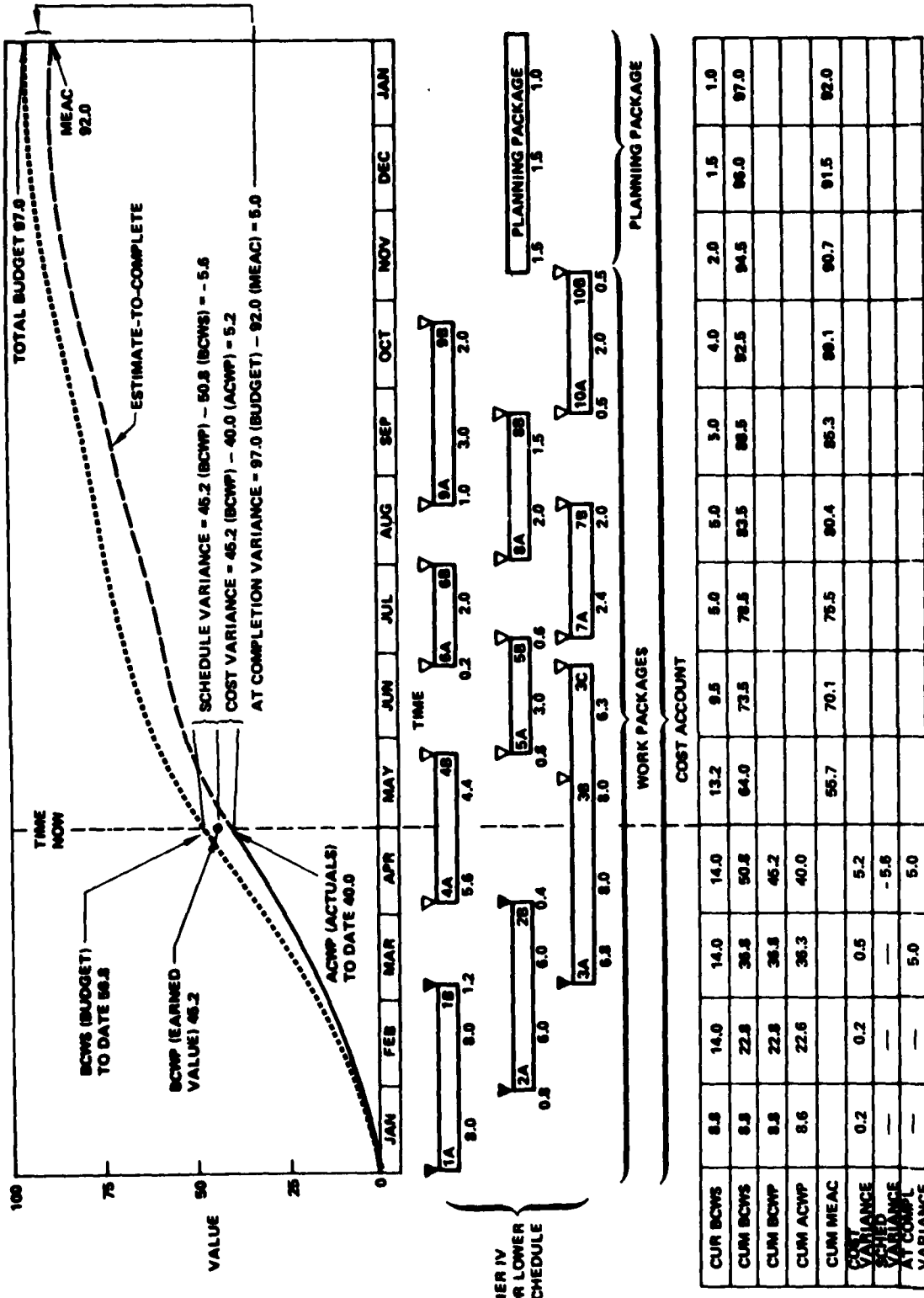


Figure C-4. Cost/Schedule/At-Completion Variances Based on Variant Milestone BCWP Measurement Method

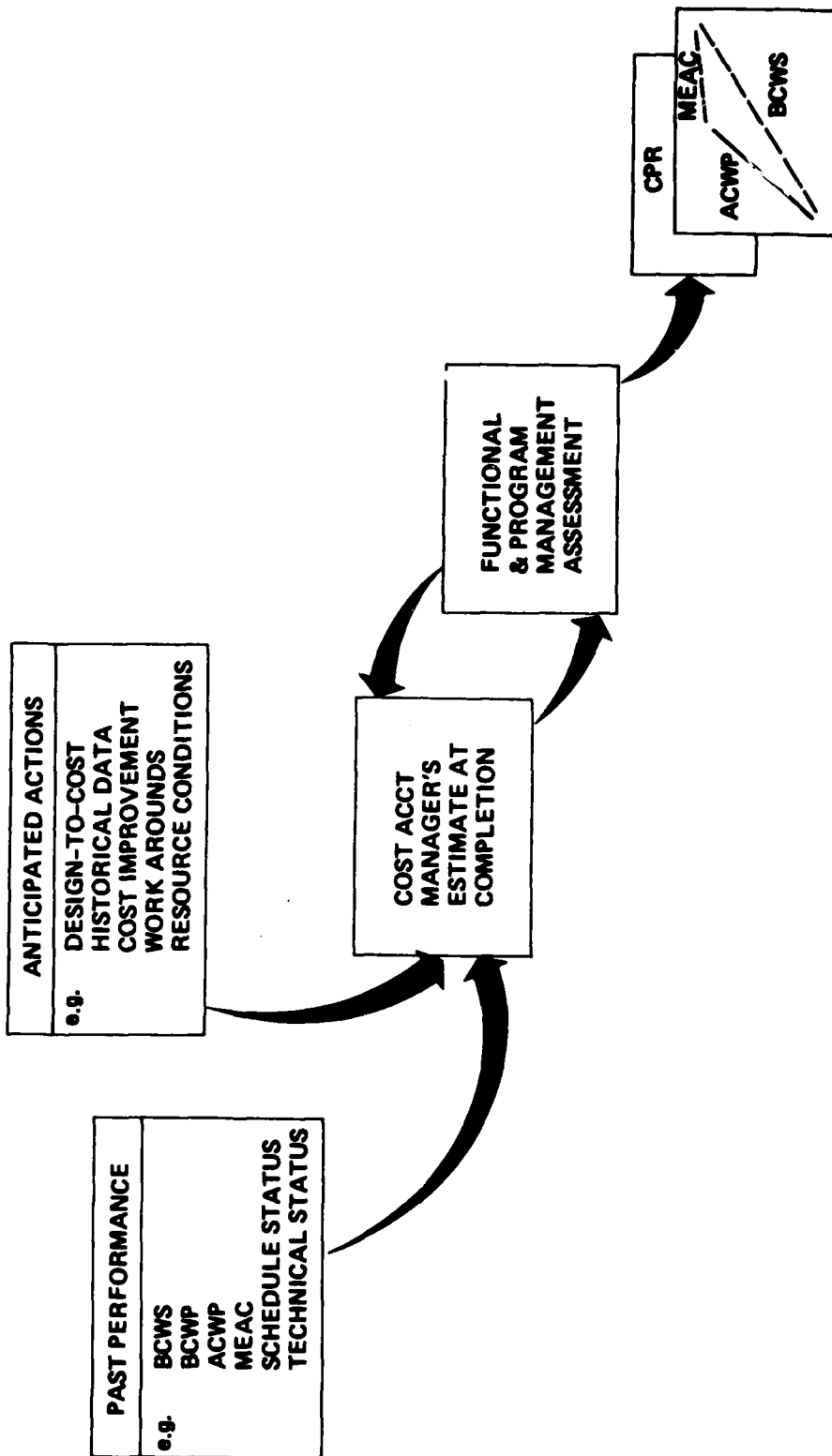


Figure C-5. Management Estimates-At-Completion

chart. Daily, work is performed against a schedule and evaluated. Weekly, cost and schedule reports are evaluated. Monthly, the whole range of performance measurement indicators are available as well as the traditional cost and schedule reports. At each level, corrective action is initiated as required.

Contractual variance analysis (See Figure C-6) thresholds are jointly established by the customer and contractor programs based on the criticality of tasks, customer requirements and familiarity with the nature of the program tasks involved. These thresholds are normally in the form of a percentage and a unit value at the WBS reporting level. For example, the negotiated contract cost variance thresholds could be 10% and \$100K. Thus, if the cum cost variances exceeds both the 10% and \$100K criteria, at the customer specified reporting level, a Problem Analysis Report (PAR) must be submitted with the Cost Performance Report (CPR) as covered in Section 6. The program manager may set internal thresholds at the cost account level at a reduced dollar level, e.g., 10% and \$10K. Cost accounts with cum variances exceeding this threshold would require the preparation of a Variance Analysis Report (VAR). The internal VARs for a given WBS become the basis for the PAR, if required.

In most cases, problems causing significant variances are already known to program management from other sources, and corrective action may already have been initiated. The VAR must, however, depict the projected cost impact of the problem. Large numbers of small unfavorable variances could add up to a major problem, requiring top level management attention. A review of variance analyses, and particularly trends, provides the manager a basis for making decisions, and initiating corrective action.

5.0 REVISIONS/ACCESS TO DATA

Contract revisions are two types: contractual and noncontractual changes. The program contracts organization acts with the program manager in handling the contractual changes. This type of change is often an Engineering Change Proposal which can impact the contract and many aspects of the IMS. The contractor, on receiving or initiating a change proposal, conducts a program change board review for consolidating the technical and cost proposals. After receipt of AF approval for the change, a new or revised work authorization is released accompanied by interim budgets and schedules. Firm budgets and schedules are released subsequent to AF/contractor negotiations of the change. Noncontractual changes may be due to cost, schedule, or technical problems that would directly impact the final product. The contractor here must keep good documentation to show that the change is needed for schedule and/or cost effectiveness. The key is to keep the customer informed and get his approval for the change, if it is needed.

The DOD contracting officer or his appointed representative according to C/SCSC criteria, must be afforded access to cost/schedule data. This data includes the supporting data to the reports in Section 6. This supporting data should be requested only on an exception basis to isolate specific problems.

AD-A083 208

BOEING AEROSPACE CO SEATTLE WA F/S 9/2
SOFTWARE COST MEASURING AND REPORTING, ONE OF THE SOFTWARE ACQU--ETC(U)
JAN 79 D L BOWIE, R B DOGGETT, D FOWLER F33657-76-C-0723
UNCLASSIFIED D180-22813-1 ASD-TR-78-44 NL

2-2

4

10

10

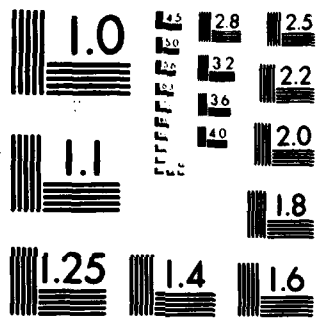
END

DATE

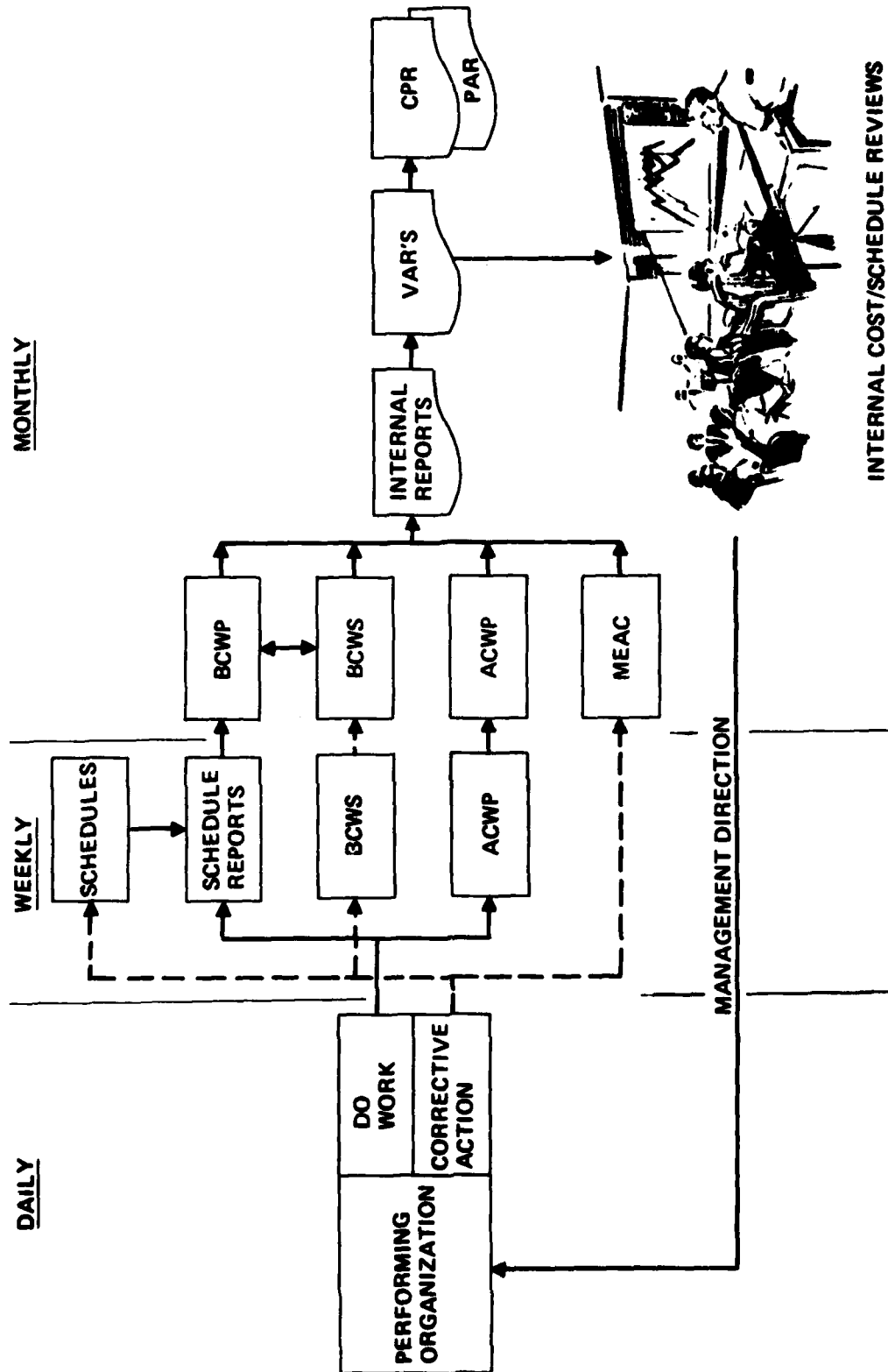
FILED

5 80

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A



INTERNAL COST/SCHEDULE REVIEWS

Figure C-6. Variance Analysis

