

AD-A191 874

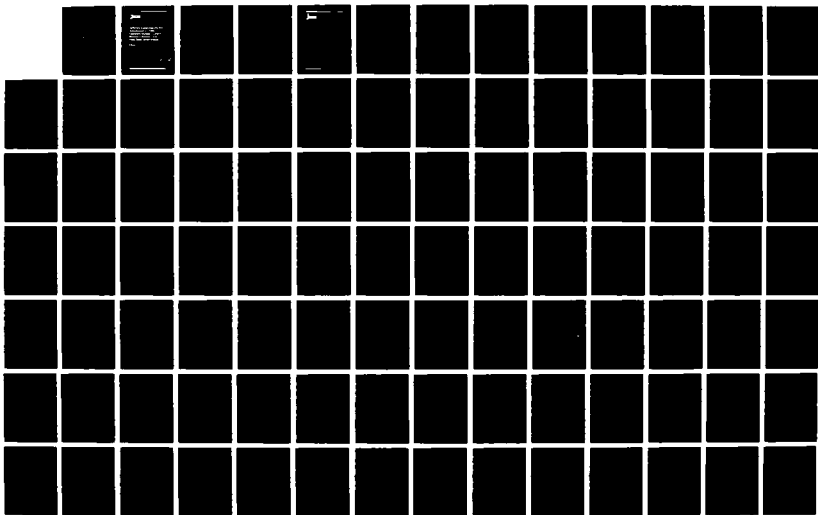
SOFTWARE SUPPORTABILITY RISK ASSESSMENT IN OT&E  
(OPERATIONAL TEST AND EVA.) (U) BDA CORP ALBUQUERQUE NM  
W HOESSEL ET AL. 28 SEP 84 BDA/A-84-322-TR  
F29601-80-C-0035

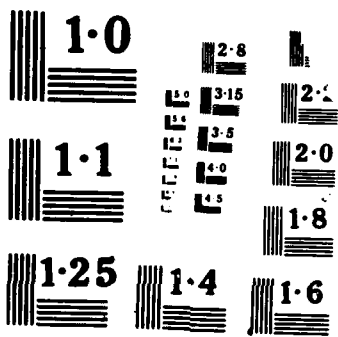
1/4

UNCLASSIFIED

F/G 12/5

NL





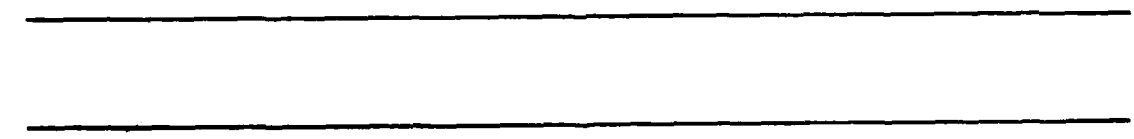
AD-A191 874



1801 RANDOLPH ROAD, S.E. · ALBUQUERQUE, NEW MEXICO 87106 · (505) 848-5000 TWX 910-989-0619

# Software Supportability Risk Assessment in OT&E: Literature Review, Current Research Review, and Data Base Assemblage

## FINAL



This document has been approved for public release and sale; its distribution is unlimited.

DISTRIBUTION: UNLIMITED

DTIC  
ELECTE  
S FEB 17 1988 D  
E

SEPTEMBER 28, 1984

BDM/A-84-322-TR



68 2 09 126

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

AD-A191874

## REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION <b>Unclassified</b>		1b. RESTRICTIVE MARKINGS <b>None</b>	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE		Unlimited	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) <b>BDM/A-84-322-TR</b>		5. MONITORING ORGANIZATION REPORT NUMBER(S) <b>Air Force Operational Test and Evaluation Center/RMC</b>	
6a. NAME OF PERFORMING ORGANIZATION <b>The BDM Corporation</b>	6b. OFFICE SYMBOL <i>(If applicable)</i>	7a. NAME OF MONITORING ORGANIZATION	
6c. ADDRESS (City, State and ZIP Code) <b>1801 Randolph Rd SE Albuquerque, NM 87106</b>		7b. ADDRESS (City, State and ZIP Code) <b>Kirtland Air Force Base, NM 87117</b>	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION <b>Same as 7a</b>	8b. OFFICE SYMBOL <i>(If applicable)</i>	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER <b>F29601-80-C-0035/SS304</b>	
8c. ADDRESS (City, State and ZIP Code) <b>Same as 7b</b>		10. SOURCE OF FUNDING NOS.	
		PROGRAM ELEMENT NO.	PROJECT NO.
		TASK NO.	WORK UNIT NO.
11. TITLE (Include Security Classification) <b>Software Supportability Risk Assessment in OT&amp;E:</b>			
12. PERSONAL AUTHOR(S) <b>W. Hoessel, W. Huebner, D. Peercy, G. Richardson</b>			
13a. TYPE OF REPORT <b>Technical</b>	13b. TIME COVERED FROM <b>4/16/84</b> TO <b>9/28/84</b>	14. DATE OF REPORT (Yr., Mo., Day) <b>September 28, 1984</b>	15. PAGE COUNT <b>300</b>
16. SUPPLEMENTARY NOTATION			
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB. GR.	
19. ABSTRACT (Continue on reverse if necessary and identify by block number) <p>&gt; Assessing the software supportability risk of Air Force acquired systems is necessary to enable various decision makers to properly plan for system deployment. Risk assessment (RA) is required throughout the system acquisition life cycle. Since the perspective of OT&amp;E is focused upon the overall system mission, including supportability, methods are required which provide software testers with areas which require testing emphasis and which provide decision makers with an assessment for software and software support risk for production decisions. Due to the complexity of these requirements, it is necessary to determine the feasibility of developing and implementing a risk assessment model of software supportability with the proper system mission perspective to ultimately assist the top level decision maker.</p> <p>This report contains the results of a literature review and current research review to determine the level of effort and usefulness of developing and implementing a risk assessment model for software supportability (RAMSS) in OT&amp;E. This (SEE NEXT PAGE) ←</p>			
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT. <input checked="" type="checkbox"/> DTIC USERS <input type="checkbox"/>		21. ABSTRACT SECURITY CLASSIFICATION <b>Unclassified</b>	
22a. NAME OF RESPONSIBLE INDIVIDUAL <b>Major Gary R. Horlbeck</b>	22b. TELEPHONE NUMBER <i>(Include Area Code)</i> <b>505-846-1254</b>	22c. OFFICE SYMBOL <b>LG5T</b>	

DD FORM 1473, 83 APR

EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED  
SECURITY CLASSIFICATION OF THIS PAGE



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

Item 11: (cont'd)

Literature Review, Current Research Review, and Data Base Assemblage (FINAL)

Item 19: (cont'd)

document makes no judgement on the information gathered, but instead serves as the first step in the development of an RAMSS model.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE



1801 RANDOLPH ROAD, S.E. • ALBUQUERQUE, NEW MEXICO 87106 • (505) 848-5000

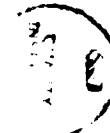
SOFTWARE SUPPORTABILITY RISK ASSESSMENT IN OT&E  
LITERATURE REVIEW, CURRENT RESEARCH REVIEW,  
AND DATA BASE ASSEMBLAGE  
(FINAL)

September 28, 1984

BDM/A-84-322-TR

DISTRIBUTION: UNLIMITED

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



## FOREWORD

This technical report, BDM/A-84-322-TR, is submitted by The BDM Corporation, 1801 Randolph Road, S.E., Albuquerque, New Mexico, 87106, to the Air Force Operational Test and Evaluation Center, Kirtland Air Force Base, New Mexico, 87117. This report is in compliance with CDRL item A008, Contract F29601-80-C-0035, and fulfills the requirements of paragraph 7.3 of Subtask Statement 304/00, titled "Software Risk Assessment in OT&E," as amended by Subtask Statement 304/01, /02, and /03.

This report was the result of effort by Mr. William Hoessel, Mr. Walter Huebner, Jr. (Task Leader), Dr. David Peercy, and Dr. G. Don Richardson of The BDM Corporation. The primary Subtask Statement Project Officer was Maj. Gary R. Horlbeck (AFOTEC/LG5T); the alternate Subtask Statement Project Officer was Mr. Jim Baca (AFOTEC/LG5).

Reviewed by:



for

Fred A. Ragland  
Program Manager

## PREFACE

The use of the term "ADP" in this document is not meant to imply any particular functional category or system. In particular, the term is meant to encompass at least the four categories outlined in AFR 800-14: Category A--ADP resources in combat weapon systems and specially designed equipment; Category B--ADP resources in other systems developed under AFR 800-2; Category C--ADP resources in systems developed, acquired, and managed by AFR 80-2, AFR 65-2, AFR 71-11, and AFR 100-2; and Category D--ADP resources in general purpose ADPS developed, acquired, and managed by the 300-series regulations and manuals. Primary application of risk assessment tools and methodologies will be to mission-critical ADP systems covered by categories A and B in accordance with AFR 800-14.

## TABLE OF CONTENTS

<u>Section</u>	<u>Page</u>
I INTRODUCTION	I-1
1.1 BACKGROUND	I-1
1.2 STUDY OBJECTIVE	I-2
1.3 STUDY APPROACH	I-3
1.4 REPORT ORGANIZATION	I-4
II FACT-FINDING VISITS/CONFERENCES	II-1
2.1 INTRODUCTION	II-1
2.2 RA TOPICS ADDRESSED ON FACT FINDING VISITS/ TELEPHONE CONTACTS	II-1
2.3 SUMMARY OF FACT-FINDING VISITS	II-1
2.4 ATTENDANCE AT CONFERENCES/SEMINARS	II-1
2.5 SUMMARY OF TELEPHONE/OTHER CONTACTS	II-5
2.6 RESEARCH REVIEWS	II-5
III LITERATURE AND RESEARCH REVIEW/KEY DOCUMENTS	III-1
3.1 INTRODUCTION	III-1
3.2 RA DATA BASE SOURCES	III-1
3.3 RA DATA AND TEXT BASES ASSEMBLAGE	III-2
IV SOFTWARE SUPPORTABILITY, RISK METHODS, AND EVALUATION MEASURES	IV-1
4.0 INTRODUCTION	IV-1
4.1 PROBLEM STATEMENT	IV-1
4.1.1 General Problem Discussion	IV-1
4.1.2 Software Supportability Issues	IV-3
4.1.3 Risk Assessment Issues	IV-7
4.1.4 Literature Survey and Research Review Summary	IV-8
4.2 SOFTWARE SUPPORTABILITY	IV-10
4.2.1 Definitions	IV-10
4.2.2 Conceptual Framework	IV-11
4.2.3 Evaluation Factors	IV-14
4.3 RISK ASSESSMENT	IV-13
4.3.1 The Theoretical Foundation	IV-13
4.3.2 Subjective Methodologies, Techniques	IV-23
4.3.3 Objective Methodologies, Techniques	IV-26

## TABLE OF CONTENTS (Concluded)

<u>Section</u>		<u>Page</u>
	4.4 APPLICATION OF RISK ASSESSMENT TO SOFTWARE SUPPORTABILITY	IV-29
V	REFERENCES	V-1
APPENDICES		
A	ACRONYMS	A-1
B	GLOSSARY OF TERMS	B-1
C	TRIP/CONFERENCE/CONTACT REPORTS	C-1
D	CONTACTS/KNOWLEDGEABLE PERSONS	D-1
E	AUTHOR BIBLIOGRAPHIC INDEX	E-1
F	TITLE BIBLIOGRAPHIC INDEX	F-1
G	DATE BIBLIOGRAPHIC INDEX	G-1
H	BIBLIOGRAPHY	H-1

LIST OF TABLES

<u>Figure</u>		<u>Page</u>
2-1	Topic List for Visits/Telephone Contacts	II-2
2-2	Fact-Finding Visits	II-3
2-3	Conferences/Seminars	II-6
2-4	Telephone Contacts/Other Visits	II-7
2-5	Research Reviews	II-8
3-1	RA Data Base Summary	III-2
3-2	List of Key Documents	III-4

## LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
4.1-1	AFOTEC OT&E Assessment: Systems and Constraints	IV-2
4.1-2	Software Maintenance Problem Factors	IV-5
4.2-1	Software Maintenance Definitions	IV-12
4.2-2	Software Maintenance Activities	IV-13
4.2-3	Software Supportability Risk Framework	IV-13
4.2-4	Some Potential Software Supportability Factors	IV-15
4.2-5	One Possible Software Software Supportability Evaluation Factor Tree (Support Activity Viewpoint)	IV-16
4.2-6	One Possible Software Supportability Factor Tree (Decision Maker Risk Viewpoint)	IV-17
4.3-1	Risk Representation	IV-18
4.3-2	Sample Discrete Probability Density Function	IV-19
4.3-3	Sample Continuous Probability Density Function	IV-20
4.3-4	Sample Risk Probability Graph for Zero Uncertainty	IV-20
4.3-5	Sample of Baseline for Risk Probability	IV-21
4.3-6	Samples of Baselines for Probability Density Functions	IV-22
4.3-7	Samples of Risk Using Probability Density Functions	IV-22
4.4-1	Elements of Risk Assessment Model	IV-30
4.4-2	Evaluation Model Framework	IV-31



**Section I**  
**Introduction**

## SECTION I INTRODUCTION

### 1.1 BACKGROUND.

The Air Force Operational Test and Evaluation Center (AFOTEC) has the responsibility for conducting operational test and evaluation (OT&E) of assets entering the Air Force inventory. AFOTEC has developed and implemented various software OT&E methodologies. These methods have matured and have become the Air Force standard for evaluating software supportability. Each of these developed methods evaluates specific characteristics of the supportability aspects of delivered software and software support resources. These stand-alone evaluations provide AFOTEC with information to identify particular software supportability deficiencies, but do not identify overall risk associated with contractor or military ownership and organic maintenance of contractor-delivered software.

Assessing the software supportability risk of Air Force acquired systems is necessary to enable various decision makers to properly plan for system deployment. Risk assessment (RA) is required throughout the system acquisition life cycle. The perspective of OT&E is focused upon the overall system mission operation, including support. Methods are needed to provide software testers with areas which require testing emphasis, and decision makers with an assessment of the software supportability risk.

Software support for major weapon systems is becoming a major system cost factor. Major weapon systems are using more sophisticated computer systems and the support costs required for embedded software is projected to increase. Furthermore, since most enhancements to the system are dependent on software modifications, the timeliness of such software support is critical to system operational availability and effectiveness. Because of this criticality of the software support function to overall system mission operational capability, it is desired that top decision makers be aware of the risk associated with the software supportability

of a system at the conclusion of OT&E. In order to determine this risk during OT&E, AFOTEC needs to develop and implement a risk assessment model of software supportability with the proper system mission perspective to ultimately assist the top level decision maker. Due to the complexity of this requirement, it is first necessary to determine the feasibility of developing and implementing such a model.

AFOTEC produced a concept proposal (reference 5.12) for computer resources risk assessment during operational test and evaluation. This effort integrates an approach, appropriate models, and subjective and quantitative software operational and supportability measures into a management-oriented assessment of user and supporter risk. This initial involvement with the application of risk assessment to software supportability provided AFOTEC with justification to support a study of the feasibility of developing and implementing a risk assessment model for software supportability (RAMSS). The AFOTEC Subtask 304 (reference 5.0) is the statement of this feasibility study's objectives and required reports. This report documents one part of this study.

## 1.2 STUDY OBJECTIVE.

The overall objective of this task study, as stated in Subtask Statement (SS) 304/00, is to perform a feasibility study to determine the level of effort and usefulness of developing and implementing a risk assessment model for software supportability (RAMSS). This report documents the first part of the effort: to "review defense and technical literature and current research concerning methods of software supportability testing and risk assessment applicable to an OT&E environment" (reference 5.0).

The emphasis for this first part of the task was placed upon:

- a) Identifying and collecting information
  - 1) Literature search and review
  - 2) Fact-finding visits/conferences
  - 3) Contact with risk assessment/software experts

- b) Assembling risk assessment data base
  - 1) Glossary of terms
  - 2) Annotated bibliography
  - 3) Key documents
  - 4) Experts/knowledgeable contacts list
  - 5) Current research list.

### 1.3 STUDY APPROACH.

A three-step study approach was adopted in SS 304/00. The steps were:

- a) Conduct a literature search and research review.
- b) Analyze the literature and research information to determine the feasibility of developing and implementing a RAMSS to be applied to military systems during AFOTEC-conducted OT&E.
- c) Identify and analyze candidate measures of supportability risk for use in developing a feasible RAMSS.

The first step results are presented in this report.

The literature search and review required identification of key documents published by governmental agencies and civilian agencies. Literature searches of the Defense Technical Information Center (DTIC), National Technical Information Service (NTIS), and Rome Air Development Center (RADC) data bases were conducted. A search and review of National Bureau of Standards (NBS) publications was done. Key documents from these searches were identified and ordered for inclusion in the RA data base. Several documents from another AFOTEC subtask on Computer System Security were identified. Researching the available RA technology also involved contact with a number of agencies, and identification of and discussion with RA research and evaluation personnel. The basic form and content of this data base of RA information is described in this report and was augmented and updated as necessary to keep the data base current throughout this study.

## 1.4 REPORT ORGANIZATION.

The remainder of this report is organized into five sections plus a set of appendices that include the detailed information concerning the activities described in paragraph 1.3. Report sections satisfy the following objectives:

- a) Section II summarizes information obtained from points of contact, fact-finding visits, and other visits/conferences.
- b) Section III discusses data base sources and assemblage, and presents key documents obtained in the literature search, particularly those concerning: DoD and government regulations; approaches to risk assessment (such as formal models); and evaluation/verification techniques for determining specific risk assessment measures as applicable to software support.
- c) Section IV describes a top-level view of elements of risk assessment from the viewpoint of decision makers and support personnel required to assess the mission needs of a system.
- d) Section V lists the documents whose contents have been referenced in this report.
- e) Appendix A lists acronyms used in this report.
- f) Appendix B is a glossary of terms (sources of the terms and descriptions are listed).
- g) Appendix C contains copies of all trip reports and contact summaries.
- h) Appendix D lists RA contacts (name, organization, address, and phone number); plus responsibilities, title and areas of RA expertise/knowledge as available.
- i) Appendix E lists alphabetically the authors in the RA bibliography along with an index of item references.
- j) Appendix F is a title index to the RA bibliography.
- k) Appendix G is a date bibliography index.

- 1) Appendix H contains the RA bibliography. It provides title, date, source, author, abstract, and review comment (where applicable) for each entry.

**Section II**  
**Fact-Finding Visits/Conferences**

SECTION II  
FACT-FINDING VISITS/CONFERENCES

2.1 INTRODUCTION.

Several visits to agencies or persons involved with some aspect of RA were anticipated as part of the information gathering activities. Most visits have been via telephone or other project trips. Those specific travel visits which have been conducted as well as research personnel contacted are discussed in this section.

2.2 RA TOPICS ADDRESSED ON FACT-FINDING VISITS/TELEPHONE CONTACTS.

Table 2-1 shows the general topic list of the visits/telephone contacts, which was tailored to the activities and scope of RA involvement by each agency or person contacted.

2.3 SUMMARY OF FACT-FINDING VISITS.

There have not been any fact-finding visits during the contract period through September 15, 1984, although personnel have obtained some information concerning risk assessment research and documentation as part of non-project related trips. These documents and contacts are indicated in table 2-2. Details of trips are contained in trip reports, copies of which are in appendix C. Table 2-2 is a listing of all the agencies visited, date(s) of visit, purpose of visit, and summary of results.

2.4 ATTENDANCE AT CONFERENCES/SEMINARS.

There has been only one conference/seminar attended during the contract period through September 15, 1984, although personnel have obtained some information concerning risk assessment research and documentation as part of non-project related conference attendance. These



Table 2-1.

## Topic List for Visits/Telephone Contacts

- (1) Organization, key personnel and charter relative to RA; relationships to DOD/USAF/Other organizations.
- (2) Guidance, plans, and methodology for risk assessment of software and/or software supportability.
- (3) Threats and vulnerabilities related to software supportability risk, including: hardware; software; operational and support procedures/controls; physical environment; and personnel.
- (4) Mechanisms (means, techniques) of attaining risk factor measurements, evaluating risk factors, and reporting risk assessment results.
- (5) Data on software risk assessment projects.
- (6) RA requirements, policy, design, implementation, verification and validation, and major trends.
- (7) RA terminology and definitions.
- (8) Formal models related to RA.
- (9) RA program initiatives.
- (10) References and documentation related to the above topics (1-9).
- (11) Current research, i.e., not formally documented, related to the above topics (1-9).
- (12) Risk assessment and software support experts/knowledgeable personnel who should be considered for contact/inputs under any of the above topics (1-11).

## Table 2-2.

## Fact-Finding Visits

<u>AGENCY VISITED</u>	<u>DATE</u>
AFCSP0: Gunter AFS, AL	1/26/84
<p>PURPOSE: Discuss the role of AFCSP0 in the Air Force computer security program, topics in computer system security, key personnel, and available documentation relevant to AFOTEC CSS OT&amp;E.</p> <p>SUMMARY OF RESULTS: Valuable information and insight was gained on the AF computer security program and AFCSP0. Key documents were obtained, including the ADPSEC Guideline series, AFCSP0 Charter, an evaluation (circa 1980) of the AF ADP security program, a sample ADP security plan, OMB circular A-71 TM No. 1, interim policy guidance, survey of ADPSEC and assistance requirements. Security issues were discussed. Good rapport was established with AFCSP0. Several contacts were identified. Computer security risk assessment was considered to be a very important part of the security evaluation process.</p> <p>COMMENT: See technical report BDM/A-84-108-TR as part of AFOTEC subtask 294 on Computer System Security tasks for further details.</p>	
MITRE Corp.: Bedford, MA	2/14/84
<p>PURPOSE: Discuss MITRE Corporation activities, research efforts, and documentation relevant to CSS.</p> <p>SUMMARY OF RESULTS: Maureen Cheheyl and her group personnel were helpful and discussed three research projects: the Practical Verification System (PVS), the Automated Threat Analysis Methodology (ATAM), and an "integrity lock" concept for data base security. The ATAM project, with eventual prototype, development and expected applications in quantification of AFR 205-16 risk analysis, was of highest interest. The MITRE CSS bibliography was obtained for review and ordering of documents through AFOTEC.</p> <p>COMMENT: See technical report BDM/A-84-108-TR as part of AFOTEC subtask 294 on Computer System Security tasks for further details.</p>	
NSA/DODCSC: Ft. Meade, MD	2/15/84
<p>PURPOSE: Discuss NSA/DODCSC organization activities, research efforts, and documentation relevant to CSS.</p>	

## Table 2-2.

## Fact-Finding Visits (Concluded)

SUMMARY OF RESULTS: Research efforts reviewed included the pending correlation of environments to the "Orange Book" (mapping of risk range to security levels), and an Orange Book for networks. Documents identified included an NSA Computer Threat Briefing. The long conversation with Col. Roger Schell was especially valuable.

COMMENT: See technical report BDM/A-84-108-TR as part of AFOTEC subtask 294 on Computer System Security tasks for further details.

MITRE Corp: McLean, VA 5/10/84

PURPOSE: Discuss MITRE Corporation activities, research efforts, and documentation relevant to CSS and WIS.

SUMMARY OF RESULTS: Valuable information on current WIS security status was obtained. WIS Configuration Management Requirements, Certification and Accreditation Plan, Security Evolution, Security Testing, and Clandestine Vulnerability Analysis were discussed. MITRE will be updating the WIS Accreditation Planning Model and the JCS PUB 22. The security evolution master plan has been considerably updated and needs to be obtained from the WIS JPMO. A new WIS Security Certification Working Group charter is being circulated. Two NBS documents which contain information on CSS measurement risk assessment, tools and techniques are: "Software Validation, Verification, and Testing Technique and Tool Reference Guide" and "Technology Assessment: Methods of Measuring the Level of Computer Security."

documents and contacts are indicated as part of the telephone and other contact summary discussions. Table 2-3 is a listing of the agency visited, date(s) of visit, purpose of visit, and summary of results.

Details are provided in the conference reports (appendix C).

#### 2.5 SUMMARY OF TELEPHONE/OTHER CONTACTS.

Table 2-4 provides a listing of persons/agencies contacted either directly or indirectly as part of this literature search and research review effort. Also included is the date of contact, and a summary statement of purpose/results of contact.

#### 2.6 RESEARCH REVIEWS.

Table 2-5 summarizes research reviews afforded by the persons/agencies contacted. Details of the reviews are included in trip or conference reports (appendix C).

Table 2-3.

## Conferences/Seminars

DESCRIPTIONDATE

STARS Measurement DIDS  
Workshop

1/14/84  
through  
1/15/84

PURPOSE

Attend Software Technology for Adaptable, Reliable Systems (STARS) workshop to review draft Data Item Descriptions (DIDS) for software life cycle measurement. Chair the session on development and operational environment DIDS. Determine applicability of proposed environment characteristics to risk assessment of software supportability.

SUMMARY OF RESULTS

The current DIDS characteristics for the support environment and software products are not oriented toward addressing the risk assessment issues identification by AFOTEC. However, the possibility of future DIO development incorporating such information may now be more likely due to the efforts of this workshop. This rework of the measurement DIDS should be carefully followed by AFOTEC to assure such information is valuable to AFOTEC.

Table 2-4.

## Telephone Contacts/Other Visits

<u>PERSON/AGENCY CONTACTED</u>	<u>DATE</u>	<u>PURPOSE/RESULTS</u>
Dr. Victor Basili University of Maryland	5/15/84	Review SEL/NASA research
Mr. John Musa Bell Laboratories	5/16/84	Review reliability applications
Dr. William Riddle Software Design and Analysis, Inc.	5/18/84	Review SAB report and software environments
Dr. Barry Boehm TRW	5/18/84	Review SAB report/TRW RA activity
Dr. Allen Stubberud Air Force Chief of Staff	5/29/84	Review SAB report/AF RA activity
Dr. Nancy Leveson University of California, Irvine	5/31/84	Review software safety applications
Mr. Jim McCall SAI	6/1/84	Review software quality metrics
Mr. Gerald Fisher AF/SASF	6/18/84	Review AF/SA technical note/SASF RA activity
Mr. William Rowe Institute of Risk Analysis American University	6/19/84	Review current research/Book-- An Anatomy of Risk
Mr. Mark van den Broek Ford Aerospace Corp.	6/19/84	Review SAB report/AFLC RA activity
Dr. Dixie B. Baker The Aerospace Corporation	7/10/84	Discuss risk analysis as applied to the Consolidated Space Operations Center (CSOC)
Dr. Richard DeMillo Ms. Ronnie Martin Georgia Institute of Technology	7/20/84	Discuss current research in software risk assessment being conducted at Georgia Tech

Table 2-5.  
Research Reviews

- o MITRE (see table 2-2)
  - ATAM: Automated Threat Assessment Methodology (Sept. 83 Draft)
- o DODCSC (see table 2-2)
  - Environments Paper: Corr. of Env. to Orange Book (pending)
  - Orange Book for Networks: Current
- o ROWE (see table 2-4)
  - CSS Risk Assessment Methodology
  - Automated Assessment Tools
- o McCALL (see table 2-4)
  - Integrated Software Management System (ISMS) Tool Set
  - IV&V Software Quality Measures
- o GEORGIA TECH (see table 2-4)
  - A Risk Model for Software Testing

**Section III**  
**Literature and Research Review /**  
**Key Documents**



### SECTION III LITERATURE AND RESEARCH REVIEW/KEY DOCUMENTS

#### 3.1 INTRODUCTION

The literature search, fact-finding visits, conferences and conversations with other Risk Assessment (RA) and Software Supportability researchers provided a list of valuable documents. From that large list of documents, a selected number were obtained for further review, abstracting and commenting. In some cases, the documents were received in microfiche form, since the receipt time for microfiche was 3-10 days as opposed to 6-10 weeks for paper copies. Of those documents reviewed, there were many which were considered key because of their direct relevance to risk assessment, provisions, testing, and/or technology; because of their potential impact on risk assessment software supportability; because of the basic foundation of their information to software supportability; or some combination of these.

#### 3.2 RA DATA BASE SOURCES.

Sources of data included the Defense Technical Information Center (DTIC); the Rome Air Development Center (RADC); National Technical Information Service (NTIS); RA experts and knowledgeable personnel contacted by telephone, on fact-finding trips and at conferences; and, references in key documents. Documents were ordered by BDM, obtained by BDM personnel during fact-finding trips, or obtained by AFOTEC for BDM.

The selection of documents for ordering was based on the need for adequate coverage of risk assessment, provisions, testing, and technology without recourse to "blanket" ordering which would have flooded the system and inhibited identification, review, and assessment of key documents. The data base was "living," in the sense that additional documents were accessed and/or incorporated as the project progressed, as appropriate. The bibliography contained in this final report identifies

all documents which were received and judged applicable by BDM during this project.

In order to summarize this information in a form which approximately defines the magnitude of the RA data bases, the major sources of documents, and the document counts (identified, ordered and received as of September 15, 1984) are given in table 3-1. For purposes of counting "received" documents, one count was given to each document regardless of the number of volumes. This partially accounts for the difference between documents ordered and received.

Table 3-1.

## RA Data Base Summary

<u>Source of Data</u>	<u>Quantity of Documents Identified</u>	<u>Quantity of Documents Ordered</u>	<u>Quantity of Documents Received</u>
DTIC (1970-1984)	450	5	3
NTIS (1964-1984)	3000	53	38
RADC	3200	21	9
CSS TASK	16	16	15
AFOTEC	13	13	7
OTHER/IN HOUSE	<u>76</u>	<u>76</u>	<u>76</u>
TOTALS	6755	134	148

## 3.3 RA DATA AND TEXT BASES ASSEMBLAGE.

BDM analysts reviewed the documents received from DTIC, NTIS, RADC, places visited, and other sources. Bibliographic information for all received documents was added to the bibliographic data base and each document was screened for further review, abstracting, and commenting. Many of the most important documents (most of the directives and regulations, for example) had no abstract; BDM analysts provided abstracts in these cases.

Appendices E, F, and G provide author, title, and date indices, respectively, to the annotated bibliographic data and text bases combined listing in appendix H (arranged by index key, which corresponds approximately to the order of document identification). The annotations include abstract and/or comment where the document reviewed was considered a key item. A preliminary list of the key documents (fewer than 1/3 of the documents received were considered "key") is provided below (table 3-2). The table is organized alphabetically.

For this report, the data base listings and indices were compiled from information gathered and input up to September 15, 1984.

Table 3-2.

## List of Key Documents

AFR 205-16, "Automatic Data Processing (ADP) Security Policy, Procedures, and Responsibilities," Attachment 5: Guidance for Performing Risk Analysis, 1 Aug 84.

AFOTEC, AFOTECP 800-2 Volumes 1 through 5, Software OT&E Guidelines.

Air Force, "Management of Operational Test and Evaluation," AFM 55-43 Vol. I, Jun 79.

Air Force, "Managing the USAF Automated Data Processing Program (Data Automation)," AFR 300-2, May 80.

Air Force, "Test and Evaluation," AFR 80-14, Sep 80.

Atzinger, E. M. and W. J. Brooks, (eds.), "A Compendium on Risk Analysis Techniques," Aberdeen Proving Ground: U.S. Army Material Systems Analysis Agency, 1972.

Boehm, B., J. Brown, and M. Lipow, "Quantitative Evaluation of Software Quality," Proceedings 2nd International Conference on Software Engineering, San Francisco, CA: 1976, pp. 592-605.

Booch, G., Software Engineering with Ada, Reading, MA: Benjamin/Cummings, 1983.

Crouch, E. A. C. and P. Wilson, Risk/Benefit Analysis, Cambridge, MA: Ballinger, 1982.

DoD, "Test and Evaluation," DoDD 5000.3, Dec 79.

Efron, B., The Jackknife Bootstrap, and Other Resampling Plans, Philadelphia: Society for Industrial and Applied Mathematics, 1982.

Fisher, G. and Lt. Col. E. Gay, "An Approach to Risk Analysis: A Process View," AF/SA Technical Note, Jun 81.

Fisk, F. and W. Murch, "A Proposal for Computer Resources Risk Assessment During Operational Test and Evaluation," AFOTEC Draft Report, 3 Oct 83.

GAO Report, "Federal Agencies Maintenance of Computer Programs: Expensive and Undermanaged," AFMD-81-25, Feb 81.

Howden, W., "Contemporary Software Development Environments," Communications of the ACM, 25(1982), 5, pp. 313-329.

Lathrop, F., "Alternative Methods for Risk Analysis: A Feasibility Study," Air Force Computer Security Program Office, 1 Sep 81.

LeBlanc, R. and J. Goda, "Ada and Software Development Support: A New Concept in Language Design," Computer, 15(1982), 5, pp. 75-82.

Lientz, B. and E. Swanson, "Problems in Application Software Maintenance," Communications of the ACM, 24(1981), 11, pp. 763-769.

Lientz, B. and E. Swanson, Software Maintenance Management, Reading, MA: Addison-Wesley, 1980.

McCall, J. and M. Matsumoto, "Software Quality Measurement Manual," RADC-TR-80-109, Vol II (of two), Apr 80.

Megill, R. E., An Introduction to Risk Analysis, Tulsa: Petroleum Publishing, 1977.

NBS, "Guidelines for Automatic Data Processing: Physical Security and Risk Management," FIPS PUB 31, National Bureau of Standards, Jun 74.

NBS, "Guideline for Automatic Data Processing Risk Analysis," FIPS PUB 65, National Bureau of Standards, Aug 79.

Neugent, W., "Technology Assessment: Methods for Measuring the Level of Computer Security," Section 4.2: Risk Assessment Methodologies, National Bureau of Standards, Draft, Sep 81.

OPNAVINST 5239.1A, "Department of the Navy Automatic Data Processing Security Program," Appendix E: Risk Assessment Methodology, 3 Aug 82.

Parikh, G., Techniques of Program and System Maintenance, Cambridge, MA: Winthrop, 1982.

Peercy, D., "A Framework for Software Maintenance Management Measures," Proceedings of the Seventeenth Annual Hawaii International Conference on System Sciences, Jan 84.

Peercy, D. and G. Swinson, "A Software Support Facility Evaluation Methodology," Proceedings of Symposium on Application and Assessment of Automated Tools for Software Development, Nov 83.

Rescher, N., Risk, Washington, D.C.: University Press of America, 1983.

Rowe, W. D., An Anatomy of Risk, New York: J. Wiley and Sons, 1977.

Thayer, R., A. Pyster, and R. Wood, "Validating Solutions to Major Problems in Software Engineering Project Management," Computer 15(1982), 8, pp. 65-77.

USAF Scientific Advisory Board, "The High Cost and Risk of Mission-Critical Software," USAF SAB Ad Hoc Committee, Dec 83.

Worm, G. H., "Applied Risk Analysis with Dependence Among Cost Components," Clemson University, Department of Industrial Management, 1981.

**Section IV**  
**Methods and Evaluation Measures**

## SECTION IV

## SOFTWARE SUPPORTABILITY, RISK METHODS, AND EVALUATION MEASURES

## 4.0 INTRODUCTION.

This section contains some concepts from the literature concerning software supportability, risk, and evaluation measures. First, some general problems of conducting software supportability risk assessment are described. Next some of the basic elements of software supportability are identified and a possible conceptual framework described for further analysis. Then, some of the generic risk assessment elements are described, including an overview of the theoretical foundation of risk and some subjective and objective methodologies/techniques. Lastly, the application of risk assessment to software supportability is described within some of the current AFOTEC capabilities and constraints.

This section is intended to be illustrative of some of the aspects of risk assessment and software supportability which will be considered in greater breadth and depth during the analysis phase of this task. It is not meant to indicate any particular constraint in the direction that the analysis effort might take.

## 4.1 PROBLEM STATEMENT.

4.1.1 General Problem Discussion.

Software supportability encompasses the personnel, resources, and procedures necessary to assure that software can be installed, operated, and modified to meet user requirements within acceptable limits. The structured OT&E of software and software support resources by the Air Force is a relatively new effort (less than 5 years, see reference 5.1). The wide range of weapon systems containing software, the criticality of those systems to national defense, and the ever present problem of limited OT&E resources set the broad boundaries of the general risk assessment problem (see figure 4.1-1). The difference can be rather



SYSTEMS EVALUATED

- (1) C<sup>3</sup>/ADP
- (2) SPACE/MISSILE
- (3) AVIONICS/EW
- (4) ATE/SIMULATORS

EVALUATION CONSTRAINTS

- (1) RESOURCE LIMITATIONS
  - PERSONNEL
  - TIME
  - DATA COLLECTION (AVAILABILITY AND ACCURACY)
- (2) VARIABLE ENVIRONMENT
  - COMPUTER
  - SOFTWARE
  - DEVELOPMENT
  - TESTING/TEST COVERAGE SCENARIO
- (3) EVALUATION REPEATABILITY AND UNDERSTANDABILITY
  - EVALUATOR EXPERIENCE
  - EVALUATION RELIABILITY
  - DEPTH OF EVALUATION MCEs
- (4) INTERNAL CHARTER
  - RESTRICTS CERTAIN OVERLAP AREAS (R&D)
  - EARLY LIFE CYCLE INVOLVEMENT NOT WELL DEFINED

Figure 4.1-1. AFOTEC OT&E Assessment: Systems and Constraints

significant between the required objectives of software supportability OT&E risk assessment, and the capability of AFOTEC and other designated resources to accomplish a timely assessment of adequate depth and understanding to assist the appropriate decision makers. Therein lies the general problem statement: Is it feasible for AFOTEC with their limited resources to assess the risk of software supportability across the wide range of systems entering the Air Force inventory such that the assessment:

- a) has a technical depth and result format appropriate to adequately assist decision makers;
- b) integrates at least the current AFOTEC evaluation methodologies;
- c) has enough accuracy and repeatability to warrant confidence in its results;
- d) is based upon a sound theoretical software and risk assessment foundation; and
- e) allows for determination of what acceptable level of risk means depending upon the identity of the risk agent and the software supportability requirements?

#### 4.1.2 Software Supportability Issues.

In order for risk assessment to be applied in the software supportability context, it is necessary to understand the elements of software, its support environment, and what software maintenance activity is required.

Software maintenance (see 5.13) is both a phase in the software life cycle as well as all those actions taken during that phase which result in any change to the software. In addition, the early decisions concerning software requirements, quality, development environment, configuration management, and delivery mold the software maintenance process. The nature of software is to encourage change. Each step in the evolution may require integration of new requirements and design.

One of the major problems (see 5.14) with software maintenance is the diversity of software product and environment "forms" that any given organization must support. Software source may be written in several different languages (even for one application system). The target operational system may have several different processors. The development environment and configuration management vary greatly across applications and are frequently not deliverable during the scope of OT&E to the target maintenance organization, which is usually tasked with supporting several applications. Even when there is some early planning for software maintenance to ease such transition diversity, the "styles" of software structure and programming tend to vary within and across application systems. The DoD concept (see 5.15, 5.16) of one language (Ada) and a reasonably uniform support (development and maintenance) environment (APSE) may help lessen the diversity of future weapon systems. Howden's (see 5.17) four levels of support environment might help management identify and control the extent of the diversity.

Lientz (see 5.18, 5.19) et. al., have investigated some of the problems in application software maintenance through the survey process and statistical factor analysis. The five principal problem factors and their primary item components (out of twenty-six) are illustrated in figure 4.1-2. These problem factors were derived from a survey of over 450 data processing managers. System reliability and machine requirements are characteristics of the software maintenance environment. Programmer-effectiveness is related to characteristics of both software maintenance environment and software maintenance management. User knowledge is an interface issue among user, development, operational, and maintenance organizations, and is normally a management level concern. The single most important item component problem identified in this survey was user demands for enhancements and extensions. This may indicate a lack of user involvement in determining the original software requirements, but more and more it probably indicates good software whose use is being expanded. Management normally controls the extent of user involvement in the development and maintenance process. Software maintenance management has been identified as a major problem by the GAO

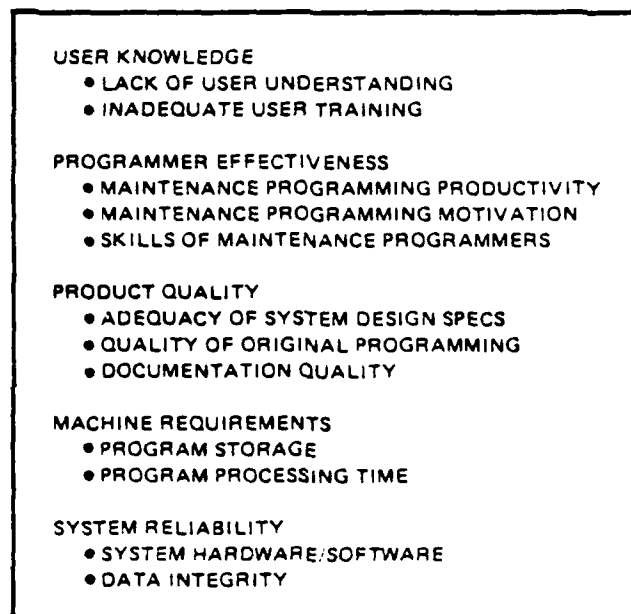


Figure 4.1-2. Software Maintenance Problem Factors

report (5.20) and several contributors to Parikh's book (5.21). Thayer (5.22), et. al., specifically identify and describe survey results on twenty software development management problems, including planning for and controlling maintainability.

Review (5.14) of the literature reveals that most of the identified software maintenance problems and solutions are "perceived." That is, identification of these problems and solutions is based upon sound logical principles, but is not correlated directly to software maintenance actions. A major deficiency in the current research is this lack of an adequate data base of software maintenance activity so that objective and subjective measures can be correlated with actual measures of software maintenance actions.

The measures of software supportability are determined from the characteristics of the identified elements and actual software support activity (e.g., the measures of resources consumed during software maintenance). These measures must be reasonably accurate, easy to collect, and based upon a viable software supportability conceptual framework (or model). The scale of measurement must be consistent across the characteristics.

The model/conceptual framework of the software and its support environment, which represent the characteristics to be evaluated as part of the risk assessment process, must be simple, yet have reasonable fidelity. The framework should allow for evaluations to be conducted under varying resource constraints and test objectives (e.g., at high level or more detailed level characteristics).

The outcome of a software supportability risk assessment should be representable in a form which pinpoints high risk drivers as well as the associated detailed risk assessment and evaluation information which determines why those drivers are a high risk. It is useful if such information can be organized so that succeedinglly greater detail can be derived depending upon the decision maker requirements.

As an example, it should be possible to determine the overall level of the supportability risk for a delivered software system. If needed, it should also be possible to determine what level of risk is associated with the delivered software products and the software support environment. It may be necessary to pinpoint risk to greater levels of depth in some cases; for example, to the level of identifying which software modules are the high risk drivers or whether the support environment personnel, support systems, and/or facilities are the high risk drivers. And, it should be possible to obtain risk assessment across groups of quality characteristics. For example, it may be that evaluation information indicates the software is very reliable, but is not easily modified or able to be ported to a different environment. If the user requirements during deployment of the system are likely to include any major modifications or a conversion to a new hardware system, then the risk assessment should be capable of appropriately identifying these software support risk drivers.

Risk assessment of software supportability also must be sensitive to the risk agent. The risk agent may be the developer, system user, the supporter, the evaluator, or even an indirect agent such as the general public. The perspective may vary a great deal from one agent to the next. Generally, all agents have some involvement, and if anyone has too much software support risk, even if it is only "perceived", then the other agent's risk is affected in a "real" way.

The bottom line to the decision maker concerning any risk assessment will be whether the associated software supportability risk is acceptable as it relates to system performance and support resource cost.

#### 4.1.3 Risk Assessment Issues.

The discipline of risk assessment/analysis has the normal problems of consistent terminology. Even the use of the term risk, not to mention risk assessment and risk analysis has only an occasional contextual agreement among users of the term.

Risk assessment discipline is also characterized by its own unique limitations. The application of very successful methods to risk assessment of nuclear waste disposal, or alcohol-related automobile accidents may be inappropriate for application to software supportability. Yet, the conceptual framework of successful risk assessment approaches should form the basis for any risk assessment of software supportability. The literature search and research review has indicated very little activity in the application of risk assessment to software, and none directly to software supportability, other than the proposed Fisk/Murch model (reference 5.12) or the Georgia Tech Model (reference 5.31).

For any specific application discipline there are always measurement problems. Who evaluates risk, why, and with which biases are a concern. The meaning of value and utility, and cost-benefit analysis from each risk agent's perspective must be considered. The scales of measurement, goals, referent baselines and required measurement confidence must be carefully considered. Sensitivity relationships between risk metrics and risk agent acceptance levels under varying environment and measurement conditions must be understood and easily determined for maximum risk assessment effectiveness. For any given application discipline, the hierarchical model of application factors and characteristics will dictate which risk assessment methodologies, techniques, and tools are applicable.

Thus, although there are models of risk assessment for some areas, a complete risk assessment model for software supportability does not exist. Such a model would have to be developed and implemented based upon guiding principles and theory from both areas of risk assessment and software supportability.

#### 4.1.4 Literature Survey and Research Review Summary.

Now that the literature search and research review is complete, there seems to be a reasonable recurring theme. Risk assessment is being done, some standards exist; very little is being done in software, and more should be done.

In some particular instances guidelines and standards exist relative to certain areas for ADP systems (e.g., references 5.2 through 5.5). And, there has been some research into application of risk analysis/assessment to software, in particular software security (e.g., references 5.6 and 5.7), software reliability (e.g., reference 5.8), software safety (e.g., reference 5.9) and software testing (e.g., reference 5.31). However, other studies (e.g., reference 5.10) have indicated more emphasis in risk assessment is needed for software and particular Post Deployment Software Support (PDSS), including an Air Force policy on software risk management. According to reference 5.10, "software for weapons systems... represents the highest risk in systems development."

The technical note from AF/SA (reference 5.11) represents an attempt to generate interest within Air Force in pursuing a more detailed research program in risk assessment. However, in talking with the author of reference 5.11 as well as several other Air Force personnel (see appendix C), there does not appear to be much if any current Air Force emphasis or activity in risk assessment of software, much less software supportability. The reference 5.12 is a high-level introduction into some of the issues of software supportability risk assessment. The Booz Allen AFRAMP effort (see reference 5.6) represents an aborted Air Force effort to develop a comprehensive security risk analysis management program.

Most of the software experts contacted (see appendix C) knew of no current research in software supportability risk assessment, although Dr. William Rowe who is primarily a risk analyst is involved in developing a methodology and assessment tools for computer system security risk assessment. His approach is apparently very detailed and is being adapted from a proprietary generic approach to risk analysis already successfully applied to other areas (e.g., criminal justice, chemical hazards, nuclear waste disposal). Although it was not available for study, the approach may be applicable to software supportability.



## 4.2 SOFTWARE SUPPORTABILITY.

This section considers some of the major elements of software supportability as contained in the literature or as derived from research review or personal contacts. Although some effort at organizing the information into a coherent presentation is made, no detailed analysis of the information is appropriate for this task report. An effort is made to show the structure of software supportability from which risk assessment can be discussed and more detailed analysis conducted.

### 4.2.1 Definitions.

There are no standard definitions for software supportability. The following definitions are supplied by AFOTEC, other definitions can be found in the glossary, appendix B.

- a) Software: A set of computer programs, procedures, and associated documentation concerned with the operation of a data processing system.
- b) Software Support Facility (SSF): The facility which houses and provides services for the support systems and personnel required to maintain the software for a specific ECS.
- c) Software Supportability: A measurement of the adequacy of personnel, resources, and procedures to facilitate:
  - 1) modifying and installing software;
  - 2) establishing an operational software baseline;
  - 3) meeting user requirements.
- d) Software Maintainability: A measure of the ease with which software can be maintained, i.e., errors can be corrected; system capabilities can be added or enhanced through software changes; features can be deleted from software; or modifications can be made to the software in order to have the system remain compatible with hardware changes.

"Supported" in this context thus implies all that accompanies "right of ownership" due to Program Management Responsibility Transfer (PMRT), including installation, modification, configuration management, and distribution.

The primary source for software supportability related definitions in the context of OT&E is reference 5.1. The literature has many publications on software maintenance and the definitions are essentially consistent with AFOTEC use. A variation of some of the AFOTEC definitions from reference 5.13 is shown in figures 4.2-1 and 4.2-2. These tie together the terminology most commonly used to define and describe software maintenance actions in the much quoted reference 5.14.

#### 4.2.2 Conceptual Framework.

A framework (see figure 4.2-3) for integrating the aspects of software product and software support facility evaluations already being conducted by AFOTEC (see reference 5.1) into a software supportability evaluation framework has been proposed in reference 5.13. This framework might form the foundation for the risk determination phase of an overall risk assessment methodology (see section 4.4). Within this framework, measures for support cost, impact of support residual risk upon system performance, various software product quality factors, and support maintenance activity can be defined and evaluation results used as part of the risk evaluation phase of an overall risk assessment methodology. The output of this risk evaluation phase would be the results of the software supportability risk assessment process.

Although a more detailed analysis of the feasible risk assessment methodologies may discover some conflicts, this conceptual framework appears to integrate some of the major elements of AFOTEC evaluation and risk assessment without committing too early to the implementation details of how the evaluation or risk assessment is actually conducted. The analysis phase will consider the feasibility of this framework as well as other identified techniques in greater detail.

<p><b>SOFTWARE:</b> THE PROGRAMS WHICH EXECUTE IN A COMPUTER. THE DATA INPUT, OUTPUT, CONTROLS UPON WHICH PROGRAM EXECUTION DEPENDS AND THE DOCUMENTATION WHICH DESCRIBES IN A TEXTUAL MEDIUM DEVELOPMENT AND MAINTENANCE OF THE PROGRAMS</p>
<p><b>SOFTWARE FAILURE:</b> ANY DEPARTURE OF PROGRAM OUTPUT FROM PROGRAM REQUIREMENTS AS THE PROGRAM IS EXECUTED.</p>
<p><b>SOFTWARE FAULT:</b> THE PRESENCE OR ABSENCE OF THAT PART OF A SOFTWARE PRODUCT WHICH CAN RESULT IN SOFTWARE FAILURE.</p>
<p><b>SOFTWARE ERROR:</b> THE HUMAN DECISION (INADVERTENT OR BY DESIGN) WHICH RESULTS IN THE INCLUSION OF A FAULT IN A SOFTWARE PRODUCT.</p>
<p><b>SOFTWARE MAINTENANCE:</b> THOSE ACTIONS REQUIRED FOR</p> <ol style="list-style-type: none"><li>(1) CORRECTION. REMOVAL CORRECTION OF SOFTWARE FAULTS</li><li>(2) ENHANCEMENT. ADDITION/DELETION OF FEATURES FROM THE SOFTWARE</li><li>(3) CONVERSION. MODIFICATION OF THE SOFTWARE BECAUSE OF ENVIRONMENT (DATA HARDWARE) CHANGES</li></ol>
<p><b>SOFTWARE MAINTAINABILITY:</b> A QUALITY OF SOFTWARE WHICH REFLECTS THE EFFORT REQUIRED TO PERFORM SOFTWARE MAINTENANCE ACTIONS.</p>
<p><b>SOFTWARE RELIABILITY:</b> A QUALITY OF SOFTWARE WHICH REFLECTS THE PROBABILITY OF FAILURE. FREE OPERATION OF A SOFTWARE COMPONENT OR SYSTEM IN A SPECIFIED ENVIRONMENT FOR A SPECIFIED TIME.</p>
<p><b>SOFTWARE PORTABILITY:</b> A QUALITY OF SOFTWARE WHICH REFLECTS THE EFFORT REQUIRED TO TRANSFER THE SOFTWARE FROM ONE ENVIRONMENT (HARDWARE AND SYSTEM SOFTWARE) TO ANOTHER</p>
<p><b>SOFTWARE MAINTENANCE ENVIRONMENT:</b> AN INTEGRATION OF PERSONNEL SUPPORT SYSTEMS AND PHYSICAL FACILITIES FOR THE PURPOSE OF MAINTAINING SOFTWARE PRODUCTS.</p>
<p><b>SOFTWARE MAINTENANCE MEASURES:</b> MEASURES OF SOFTWARE MAINTAINABILITY, SOFTWARE MAINTENANCE ENVIRONMENT CAPABILITIES TO SUPPORT MAINTENANCE ACTIVITIES, AND SOFTWARE MAINTENANCE ACTIVITY.</p>
<p><b>SOFTWARE MAINTENANCE MANAGEMENT:</b> THE POLICY, PROCEDURES AND GUIDELINES APPLIED IN A SOFTWARE MAINTENANCE ENVIRONMENT TO THE SOFTWARE MAINTENANCE ACTIVITIES. ALSO, THOSE PERSONNEL WITH SOFTWARE MAINTENANCE MANAGEMENT RESPONSIBILITIES.</p>

Figure 4.2-1. Software Maintenance Definitions

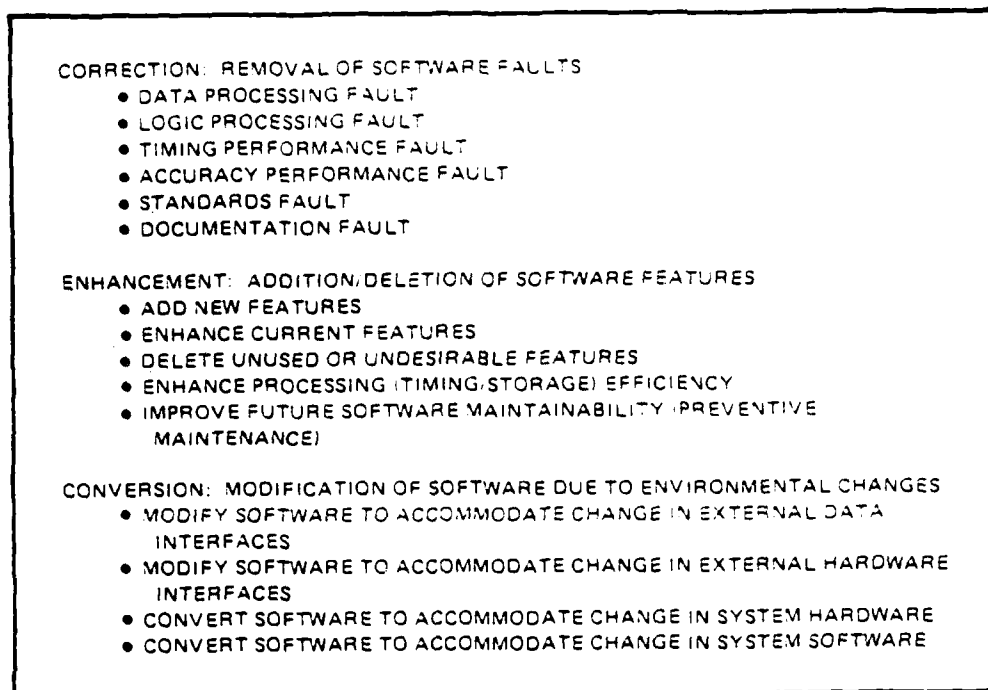


Figure 4.2-2. Software Maintenance Activities

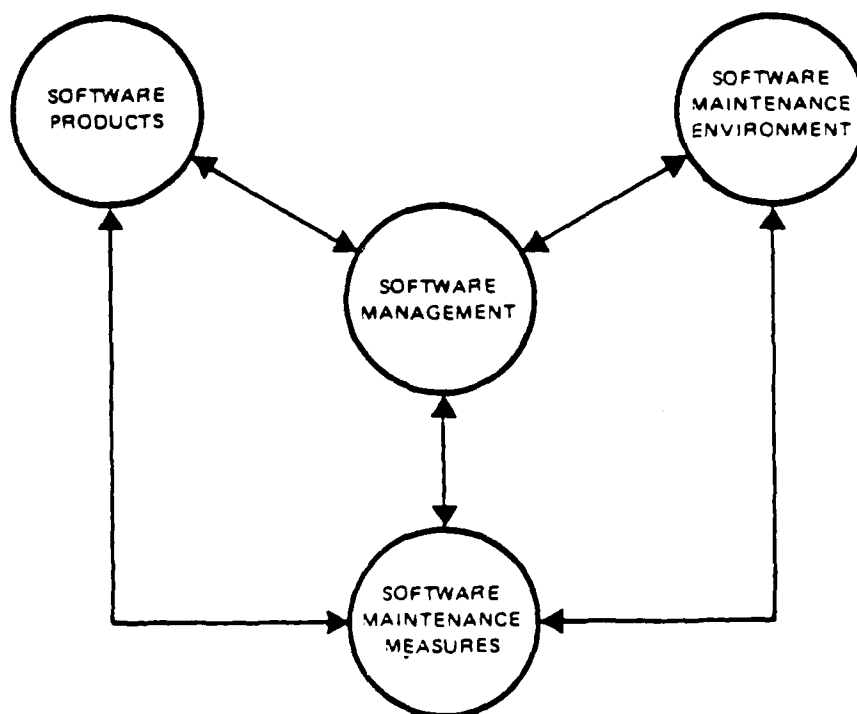


Figure 4.2-3. Software Supportability Risk Framework

#### 4.2.3 Evaluation Factors

There are many possible factors which could be evaluated and which might affect software supportability risk. And, the organizational representation of such factors can be represented in various forms, each of which might have special importance depending upon the evaluation objectives. The key is to determine a broad enough set of evaluation factors to supply appropriate fidelity and which are capable of being described by characteristics to a variable depth of detail depending upon evaluation objectives and constraints.

The framework described in reference 5.13 is suggestive of the need to measure factors of software quality and support environment capabilities, and compare these factor measures against predicted or required maintenance support activity. The comparison would provide a basis for risk assessment measures which could be derived depending upon the risk agent. The software support management is an organizational function to make certain the information upon which repeated risk assessment decisions can be made is available throughout the software support phase.

Throughout this literature search and research review task, possible factors have been identified by AFOTEC, the task team, and the literature. Typical factors and various organizational schemes have been described in references 5.13, 5.14, 5.23, 5.24 and others.

A list of some potential factors with no particular organization is shown in figure 4.2-4. A couple of draft attempts to show organization of some of the factors are illustrated in figures 4.2-5 and 4.2-6. Such factors and organization are only meant to be illustrative of the general process which are described in more precise detail as part of the analysis phase of this project. References 5.13, 5.14, 5.23, and 5.24 have described such general approaches from a software viewpoint. References 5.2 through 5.7, 5.11, and 5.25 provide such factors and organization for risk assessment.

MODULARITY	SYSTEM MISSION (Priority)	METRICS
DESCRIPTIVENESS	SYSTEM DEFINITION	HARDWARE STABILITY
INSTRUMENTATION	FLEXIBILITY	CONFIDENCE
CONSISTENCY	LEVEL OF EVALUATION	RISK ASSESSMENT PROCESS
SIMPLICITY	SOFTWARE CHANGE RATE	PROGRAMMING LANGUAGES
EXPANDABILITY	RESOURCES/PLANNING	NUMBER OF LANGUAGES
MAINTAINABILITY	PRODUCTS	PRODUCTIVITY
RELIABILITY	PERSONNEL (Types)	OPERATIONAL CONSTRAINTS (Speed, Accuracy, etc.)
MATURITY	COST	ESTIMATION TECHNIQUES
TEST COVERAGE	SYSTEMS	SUPPORTER EXPERIENCE
INTEROPERABILITY	FACILITIES	OVERSIGHT MANAGEMENT
HUMAN FACTORS	CONTRACTOR VS. GOVERNMENT	DOCUMENTATION
SECURITY	AVAILABILITY OF TOOLS	EXTENT OF (I)V&V
PORTABILITY	CONFIGURATION MANAGEMENT (Control)	SCHEDULE (TIME)
SYSTEM ARCHITECTURE (Design)	COMPLEXITY	PERSONNEL STABILITY
QUALITY	TOTAL SYSTEM SIZE	SYSTEM CRITICALITY

Figure 4.2-4. Some Potential Software Supportability Factors

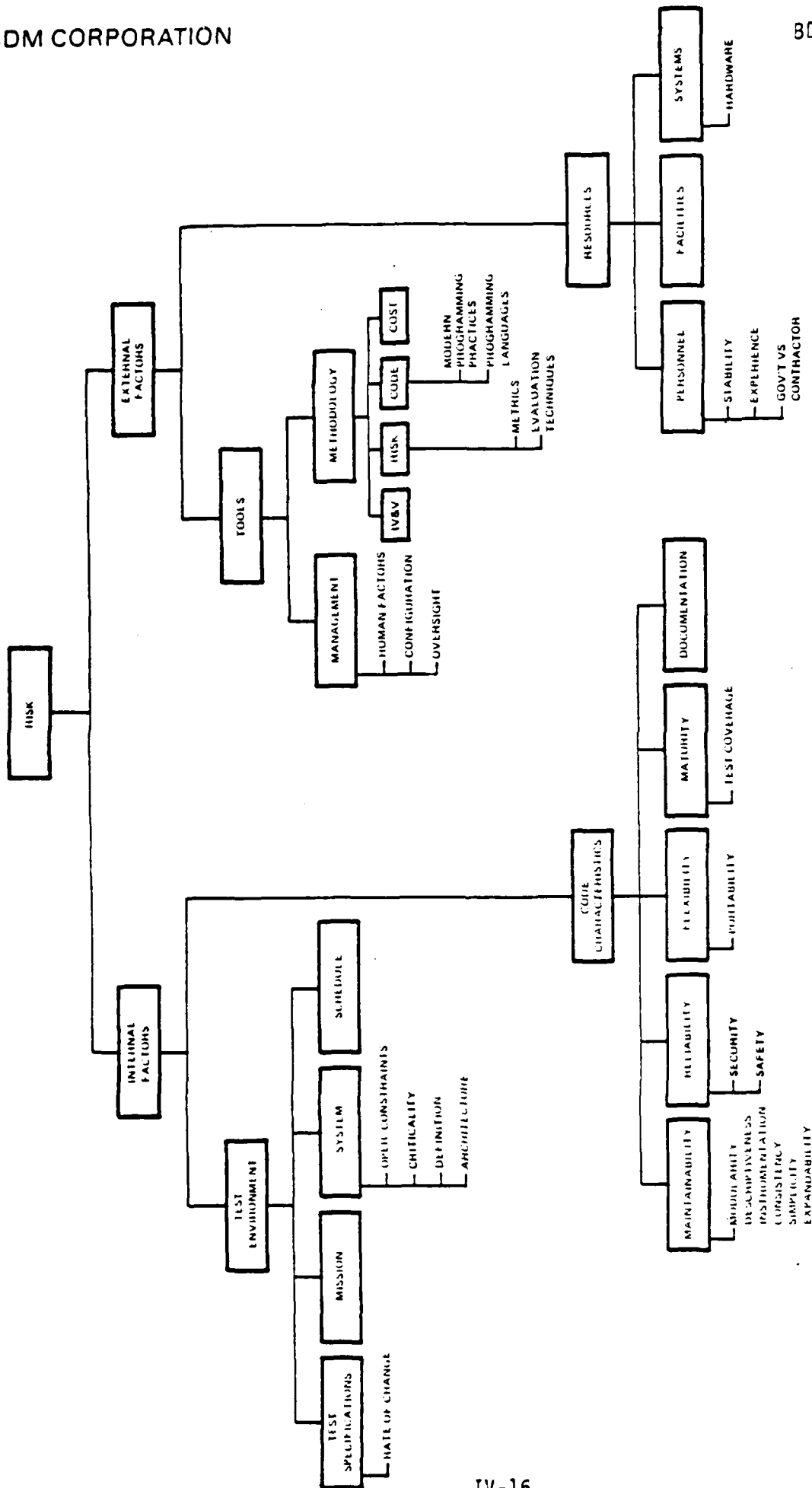


Figure 4.2-5. One Possible Software Supportability Tree (Decision Maker Risk Viewpoint)

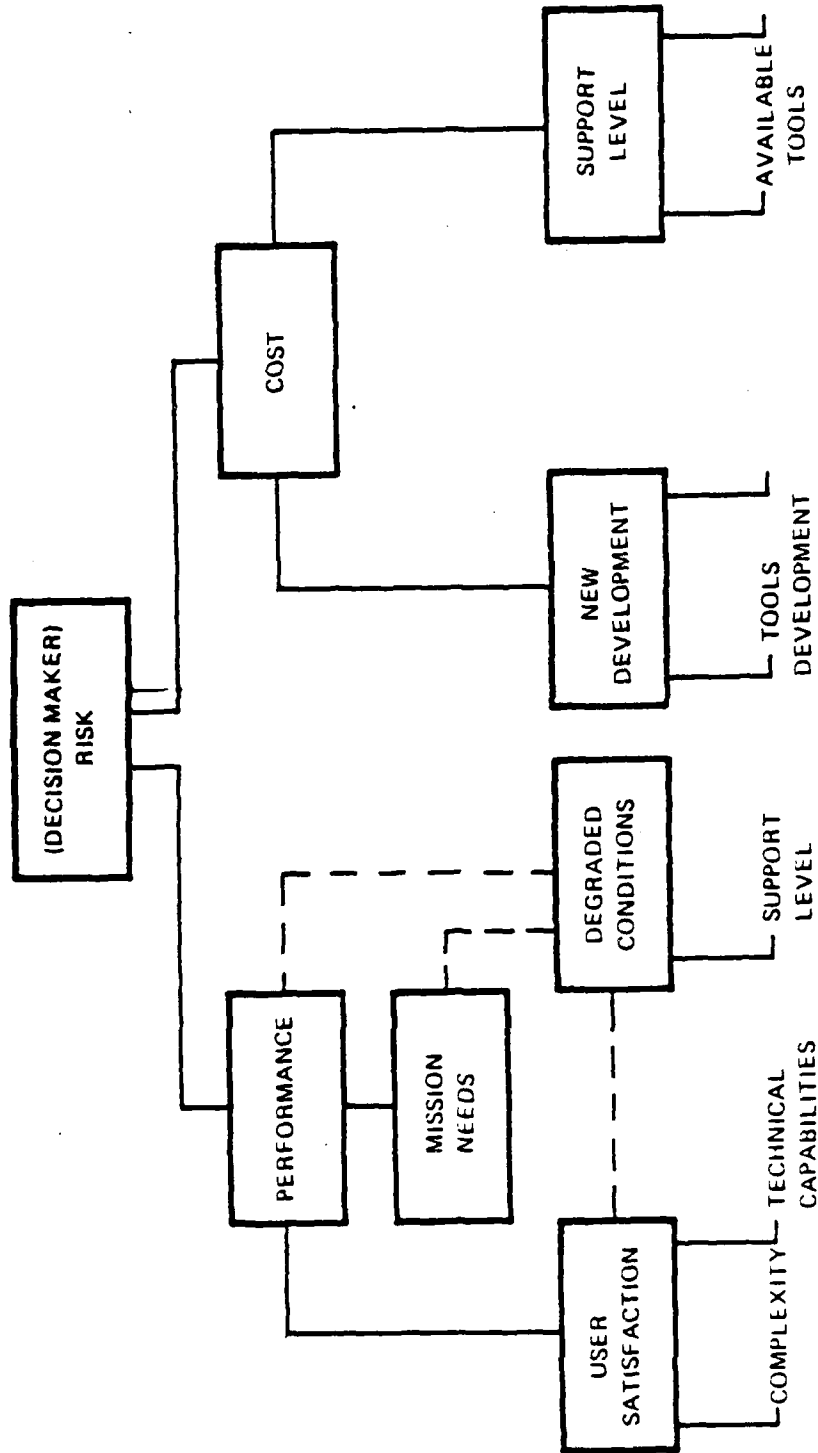


Figure 4.2-6. One Possible Software Supportability Factor Tree (Decision Maker Risk Viewpoint)



### 4.3 RISK ASSESSMENT.

This section summarizes information obtained from the risk literature reviewed and cited in the references and bibliography. Risk assessment has both theoretical and practical aspects to it. First, the theoretical foundations of risk will be discussed. What is the definition of risk? How is risk expressed? Next, those methodologies used to assess risk will be addressed. Finally, risk assessment is discussed as it applies to software supportability in an OT&E environment.

#### 4.3.1 The Theoretical Foundation.

Risk is defined as "a possible negative outcome" (Reference 5.30) or as "the realization of unwanted, negative consequences of an event" (reference 5.25). These definitions imply that the concept of risk is two-dimensional; i.e., risk consists of two parts. One part of risk is the negative outcome or the unwanted consequence. The second part of risk is the probability or potential of the negative outcome's occurrence. These two parts can be conveniently thought of and represented as two orthogonal scales as shown in figure 4.3-1.

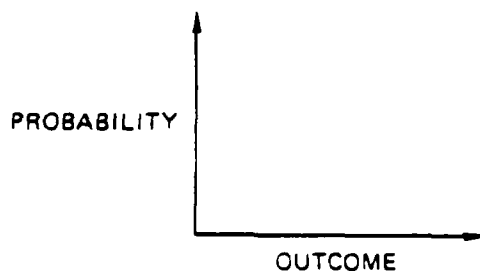


Figure 4.3-1. Risk Representation

Probability, the vertical scale in figure 4.3-1, is measured in conventional statistical terms. That is, the measure of probability ranges from 0 percent (no chance of occurrence) to 100 percent (absolute certainty of occurrence). A probability value is associated with each outcome. Outcome can be measured by a number of ways and depends on the problem context in which the risk assessment is being made. In the case of software supportability, outcome may be specified by either a cost, schedule, or performance variable. For example, consider that to support a given software package it is estimated that there is a 30 percent chance that supportability will require 50,000 dollars, a 50 percent chance that 100,000 dollars will be needed for supportability, or a 20 percent likelihood that 150,000 dollars will be required. This case is depicted in figure 4.3-2.

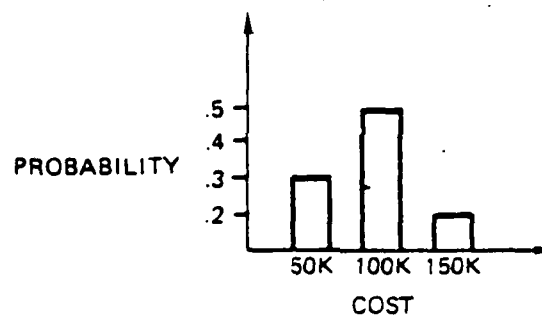


Figure 4.3-2. Sample Discrete Probability Density Function

When outcomes are assigned probabilities so that the probabilities add up to 100 percent, then a probability density function is established. The probability density function is a fundamental concept to risk assessment and its estimation is the basis for risk determination. Probability density functions may be discrete (as in the case of figure 4.3-2) or continuous. For continuous probability density functions, the probability of occurrence for some interval of outcomes is that area under the density function that is cut off by the outcome interval. For example, in figure 4.3-3, the probability of an outcome greater than a and less than b is the area under the curve between a and b.

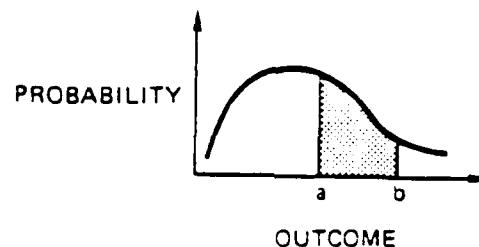


Figure 4.3-3. Sample Continuous Probability Density Function

Implicit to the definition of risk is the notion of uncertainty. If there is no uncertainty, there is no risk relative to the uncertainty. Risk analysts do not discuss situations with certain outcomes. Risk analysis specifically "attempt(s) to quantify uncertainty" (reference 5.26). And, it is the probability density function that is the vehicle for the expression of uncertainty in quantitative terms. From the example depicted as figure 4.3-2, it is uncertain as to the cost of supporting a given software package. The uncertainty is expressed by explicitly stating that more than one cost outcome has a potential for occurrence. In other words, the cost of software supportability is not certain. Conversely, if it is certain that software supportability will require 100,000 dollars, as shown by figure 4.3-4, then there is no uncertainty in the risk.

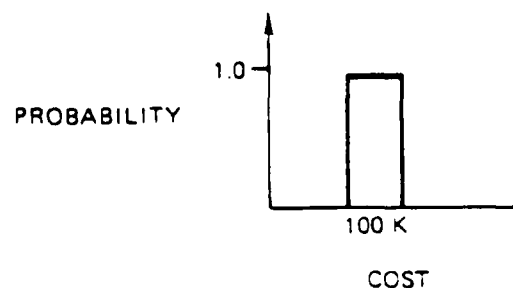


Figure 4.3-4. Sample Risk Probability Graph for Zero Uncertainty

Still to be considered in the definition of risk is the negative aspect, or magnitude, of the outcome. The concept of negative outcome or consequence can only be evaluated with respect to some baseline level. This baseline level is some value of the outcome which usually represents the available resources. For instance, if we are allocated 120,000 dollars for supportability and our estimation of outcomes are those shown in figure 4.3-2, then the negative outcome is that area of the probability density function that exceeds the baseline value. This relationship is shown in figure 4.3-5.

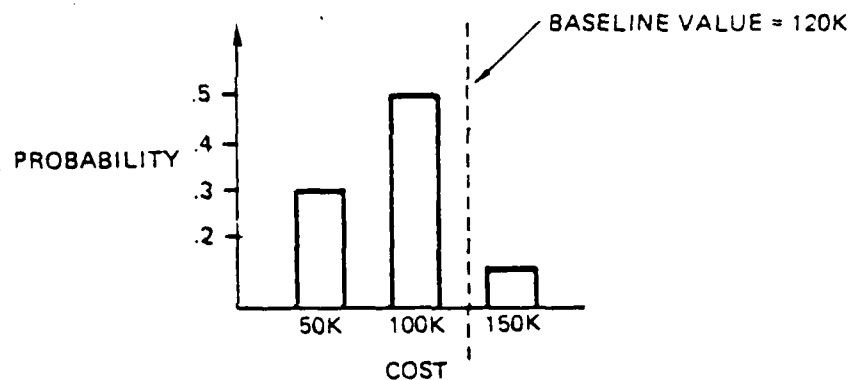


Figure 4.3-5. Sample of Baseline for Risk Probability

Given the conceptualization of risk put forth so far, then qualitative assessments of risk can be directly related to the area of the probability density function that exceeds the baseline value. Now terms such as "high" or "low" risk can be explicitly defined mathematically. As an example, high risk may be defined as a situation where 40 percent or more of the probability density function exceeds the baseline value (see figure 4.3-6a). Low risk may be the case where 10 percent or less of the probability density function exceeds the baseline outcome (see figure 4.3-6b).



Figure 4.3-6. Samples of Baselines for Probability Density Functions

Risk assessment must go further than simply considering the probability component of risk. The severity of the outcome has to be accounted for. To illustrate this idea, consider figures 4.3-7a and 4.3-7b. In both figure 4.3-7a and figure 4.3-7b, 30 percent of the probability density function exceeds the baseline value.

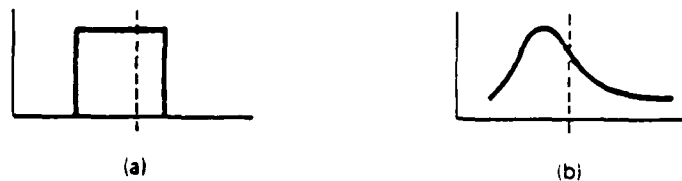


Figure 4.3-7. Samples of Risk Using Probability Density Functions

However, it is apparent that figure 4.3-7b represents the riskier situation since the possible outcomes are more severe. Thus, risk is some combination of probability and severity.

The key to risk assessment is the estimation of the probability density function. In other words, some estimate must be made of the outcomes (e.g., costs) and the probability of each outcomes' occurrence

(perhaps dependent upon risk agent). It is this step in which the risk analyst must find a methodology which best conforms to the theoretical framework of risk just laid out. This step is usually an arduous task. Data sources for most risk assessments are quite limited. Thus, a risk assessment methodology is used that is practical, implementable, and can yield some evaluation of risk, however partial the analysis. Not every risk assessment methodology, however, explicitly or implicitly attempts to estimate a probability density function.

#### 4.3.2 Subjective Methodologies, Techniques.

Risk assessment methodologies usually rely on either objectively-derived data or subjectively-derived data. First, let's consider methodologies using subjective data. Several methodologies exist in the literature for arriving at an estimated probability density function based on subjective judgments. These methods include: choice-between-gambles technique, batteries, a modified Churchman-Ackoff technique, modified Delphi technique, Bayesian estimates, and estimates of the moments of the distribution via direct questioning. A short overview of each of these methods is given below (see references 5.27, 5.28 for further details). Several other risk assessment methodologies exist that are based on subjective data. However, none of these other methods attempt to estimate a probability density function. These methods include checklists, qualitative surveys, rating scale surveys, and so on. In essence, these methods attempt to yield a "gut feel" of risk as opposed to an explicit statement of risk by a probability density function.

##### 4.3.2.1 Choice-Between-Gambles Technique for Deriving Probability Density Functions.

This method employs betting-type or gambling situations to elicit inferred probability density function from the expert. The expert

proceeds to reveal indifference probabilities between a hypothetical gamble and a real-life gamble involving a fixed level of the variable of interest. By varying probabilities in the hypothetical gamble and the level of the variable of interest, a subjective probability distribution is obtained.

#### 4.3.2.2 Choice-Between Gambles Technique for Deriving Cumulative Distribution Functions.

A cumulative distribution function of subjective probabilities is derived based on the expert's revealed indifference characteristic values. These values result from a hypothetical gamble versus real-world-gamble (i.e., involving the variable of interest) betting situation for a fixed level of probability. Each successive decision stage of the procedure reveals a characteristic value within a specified interval of values which divides the interval into equally probable subintervals. Relating each specified value directly to a cumulative probability of occurrence, a distribution function is obtained.

#### 4.3.2.3 Standard Lottery.

A probability density function is derived for the component characteristic variable of interest. Probabilities are inferred based on a selected number of hypothetical lottery tickets chosen from a lot of fixed size. The number of tickets chosen by the experts for each defined level of the component characteristic directly infers his subjective feeling for the probability of realization of that characteristic value.

#### 4.3.2.4 Modified Churchman-Ackoff Technique.

No indifference assessments or betting decisions are required in this technique. Instead, the expert is asked to make relative probability-of-occurrence-type judgments (i.e., greater than, equal to,

and less than) between various sets of possible characteristic probabilities. Then, he is asked to make numerical relative probability judgments between values on the ordinal scale desired in the previous decision stage. The resulting relative probability scale is directly converted algebraically into a probability density function.

#### 4.3.2.5 Modified Delphi Technique.

Group (i.e., at least 3 experts) subjective probability distributions, as opposed to individual probability distributions, are desired. Employing the Modified Delphi Technique, individual probability responses are elicited, reasons stated regarding such judgments are made, and all information is fed back to all respondents in an iterative procedure. A group probability response for all characteristic values is ultimately defined by averaging.

The techniques developed in this section for eliciting subjective probabilities involve asking the expert:

- a) to make choices between different betting situations,
- b) state preferences between combinations of component characteristic values; or
- c) evaluate responses in a group decision-making situation.

The resulting probability distributions are in the form of a probability density function.

#### 4.3.2.6 Bayesian Analysis.

The Bayesian analysis approach holds that it is possible, at any time, to express one's state of knowledge (e.g., about risk) in the form of a probability density function. As additional experimental evidence becomes available, then Bayes' theorem is used to combine this new evidence with the previous probability density function in order to obtain a new posterior probability distribution. The new distribution represents the updated state of knowledge.



#### 4.3.2.7 Estimates Via Direct Questioning.

Probability density functions can be obtained directly from subjective estimates. In essence, all that is usually asked for is three values describing the nature of your variable of interest. These three values (e.g., for cost) are usually:

- a minimum value of cost
- a most likely value of cost
- a maximum value of cost

Once these values are elicited, then a statistical assumption is made about the functional form of the distribution to be used. Given the estimated values and an assumed functional form, the probability density function can be completely defined.

#### 4.3.3 Objective Methodologies, Techniques.

The probability density function can also be estimated from objective data. Parametric models are used for risk assessment where objective data is available. Where extensive objective data bases exist, accurate risk models have been developed. The insurance industry immediately comes to mind. With a great deal of accuracy, the auto insurance business can tell me the probability that I will have an accident of varying degrees of severity.

##### 4.3.3.1 Concept of Parametric Model.

Any parametric model used for risk assessment will be an abstraction of reality, by definition. The model will be a way of summarizing, representing, and expressing in a formal way the complex relationships and interrelationships of the software supportability problem. Thus, it is realized that the model will not account for every detail affecting risk. Any evaluation of risk must be accompanied by a caveat on what is included or excluded in the model. Given that a model is an abstraction,

then the first objective is to identify the main drivers of risk. In other words, those components that account for the most variation in the uncertainty in cost, scheduling, or performance will be considered first in the model development.

A parametric model for risk assessment will not simply estimate some definitive quantity of risk such as the exact support costs for a given software package. Instead, the model must provide in some way a set of probabilities. That is, some measure of the variation must be at least appended to the expected value of the risk measure (e.g., cost). The model must incorporate some notion of the statistical uncertainty of the supportability expense. In this way the model touches base with the theoretical basis of risk. Some estimate of a probability density function must be predicted, however crude.

#### 4.3.3.2 Risk Drivers.

First, the risk assessment model will be a fairly simplistic and parsimonious one. Perhaps only a dozen major risk factors will be modeled to predict the cost, schedule, or performance measures of supportability. Factors such as maintainability and reliability have received considerable attention in terms of attempting to model these concepts. This previous research may be relied upon for our model development. Pre-existing parametric-type relationships can be directly incorporated into our model (given an understanding of their applicability). More often than not, however, well defined pieces of our model will not exist. For this scenario, the structural relationships of the model must first be determined. For instance, the cost of supportability may be an inverse function of the amount of code documentation. In some cases, the driving risk factor may not easily be measured by some metric. Second, a proxy variable or a set of proxies will be used. Where data exists matching the structural model, then parametric relations can be developed via regression techniques. Jackknife or bootstrap methods can be used to incorporate uncertainty into the model (see reference 5.28).

Where data is sparse or nonexistent, then equations can be developed that are heuristics or "rules of thumb". As an example, higher level computer languages are easier to modify than assembly language codes. This concept may be incorporated into a model as a multiplying factor of sorts. The heuristics can be developed by analogy, from concepts published in the literature, from intuition, or from some reasonable method of obtaining subjective estimates.

Technical issues of the modeling task are also apparent. Of critical importance is the way in which the components of the model are combined together. Specifically, if the model estimates probability density functions of cost for only two risk drivers, say maintenance requirements and code characteristics, then it is problematic in combining the estimates into a total estimate. The interdependence among risk components causes mathematical complications in building a total probability distribution of cost. (See reference 5.29 for more details.) Another issue is the distributional form of the probability density function. Where the probability density function is not completely and entirely determined, then some distributional form is assumed. This assumption makes the risk assessment process tractable in that only moments of the distribution need be estimated. From the risk literature reviewed, normal, beta, triangular, Weibul, and Rayleigh distributions have all been considered.

Parametric models are built in a top-down fashion. That is, the major risk drivers are considered first. Only when a simple, basic model is scientifically acceptable does the risk analyst build a more complex model. In model building, the structural relationships between variables are first hypothesized. If sufficient data exists, then the relationships can be made mathematically explicit by regression techniques. If data is nonexistent, then heuristic relationships between variables may be defined. The main point of departure from most parametric models is that risk deals with uncertainty. Not only must a risk assessment model estimate some definitive value, but the model must estimate a range of values each having an associated probability. Different estimated values

with different probabilities define a probability density function, a fundamental concept of risk assessment. Risk is defined when a baseline value is compared to the density function. That is, risk is defined by those outcomes and their probabilities that are negative consequences with respect to the baseline. With this approach, concepts such as "risky", "not risky", "high risk", "low risk", etc. can be explicitly and precisely defined.

#### 4.4 APPLICATION OF RISK ASSESSMENT TO SOFTWARE SUPPORTABILITY.

The integration of risk assessment and software supportability evaluation methodologies/techniques apparently has not been heavily researched by others, much less applied in any generic way. Only one of the literature references (see reference 5.12) and only one of the experts contacted (see appendix C) indicated any active involvement in risk assessment of software supportability. In addition, there is substantial activity in risk assessment of some aspects of automated systems, in particular software security. Several of the references already cited (e.g., references 5.2 through 5.9) involve some aspect of software risk assessment. Rowe (see reference 5.25 and contact summary in appendix C) is actively involved with application of risk assessment methodology to software and computer systems. There is every reason to believe that it is feasible to integrate current risk assessment methodologies and software supportability evaluation methodologies. The analysis phase of this current contract addresses specific issues of this feasibility (see reference 5.32).

Some of the aspects of software supportability and risk assessment along with problems to be solved as derived from some of the literature have been summarized in earlier sections. Integration of these various aspects will require a careful development of a combined model framework. Elements of such a model (partially derived from reference 5.25) are illustrated in figure 4.4-1.

A generic evaluation model framework is illustrated in figure 4.4-2. None of the frameworks represent the results of detailed analysis, but they are representative of the process.

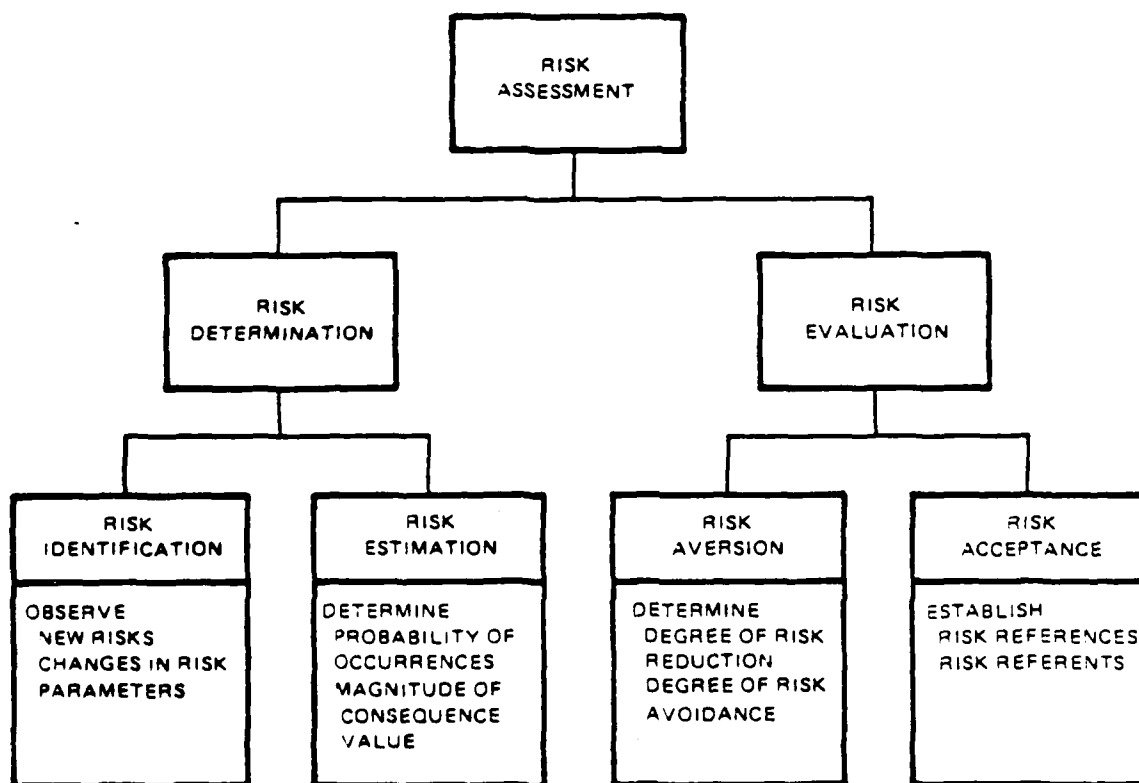
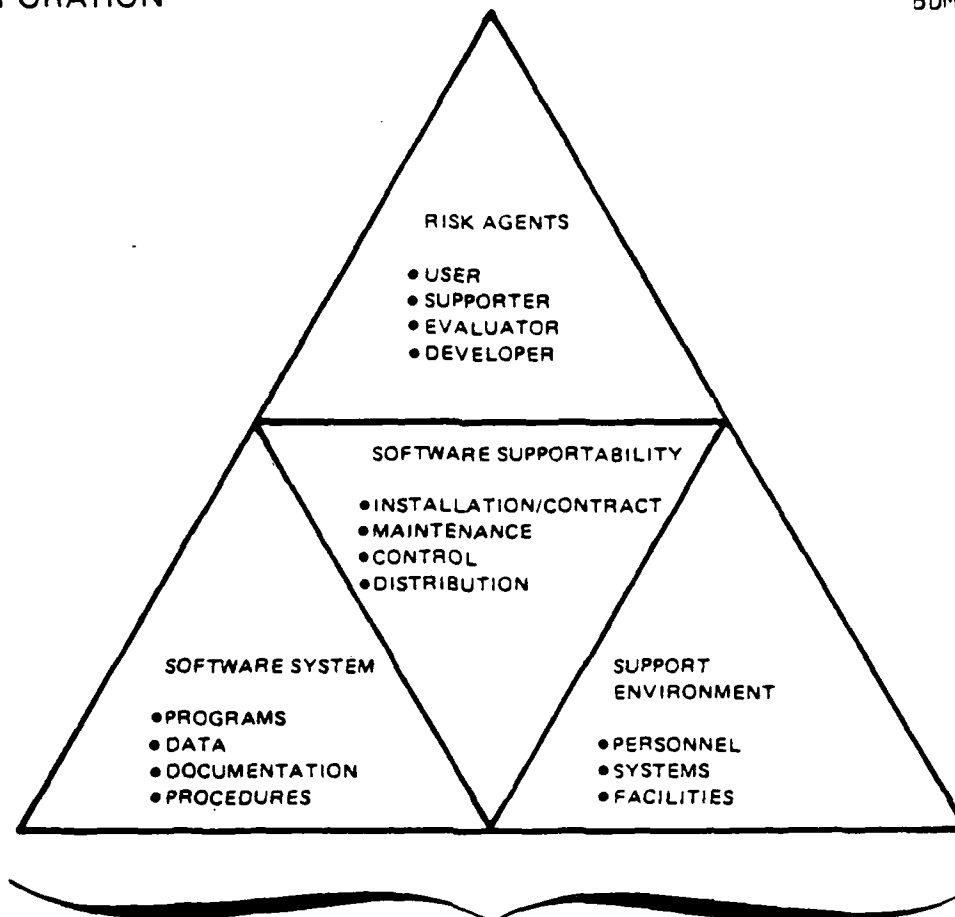


Figure 4.4-1. Elements of Risk Assessment Model

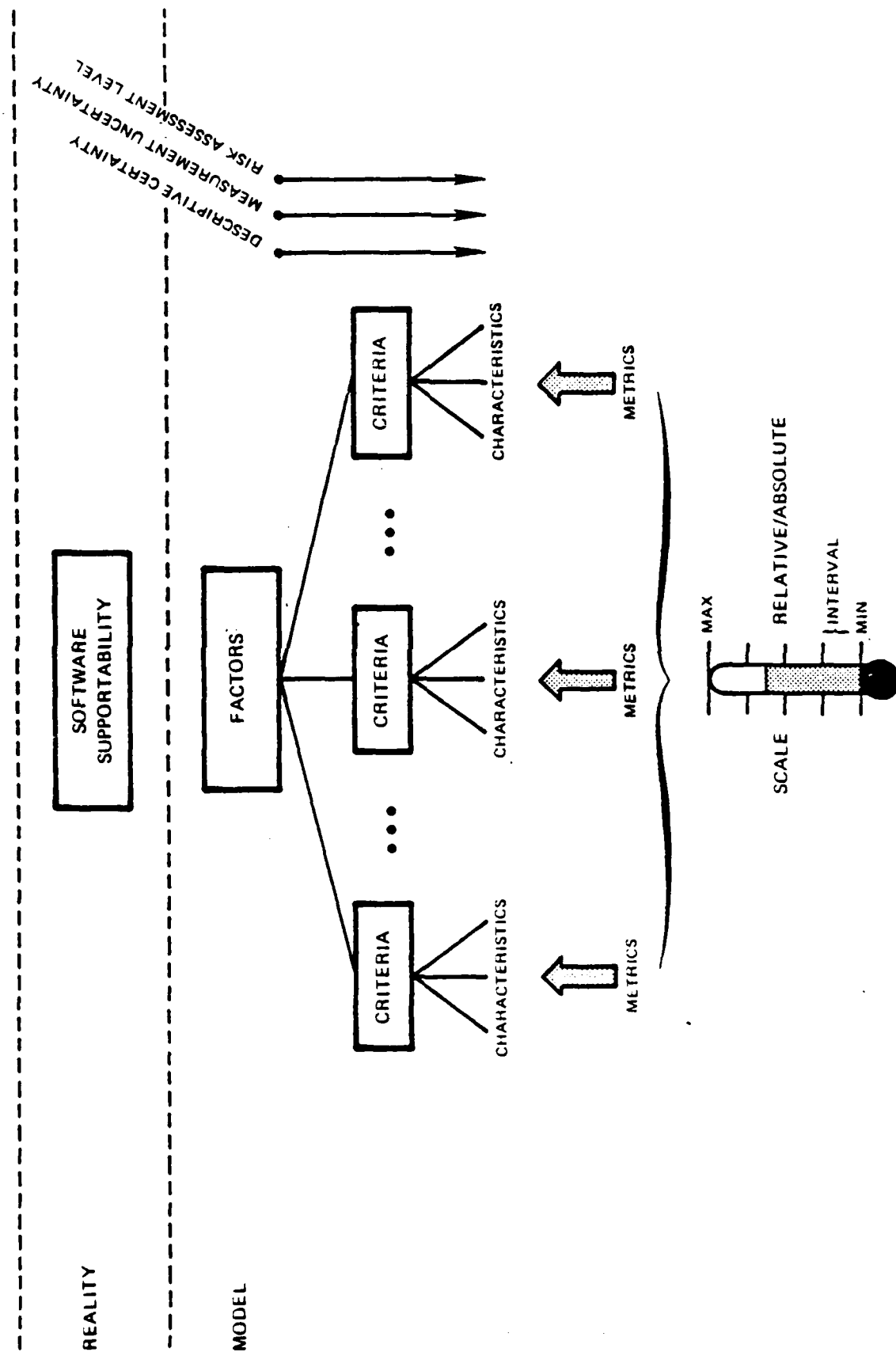


Figure 4.4-2. Evaluation Model Framework

Risk determination is: 1) the identification of software supportability objectives, MOEs, criticality/sensitivity, and application specific risks; and 2) the estimation of the risk event probability of occurrence and the magnitude of the importance a risk agent subjectively attaches to the undesirability of a specific risk consequence. Risk determination involves the application of a theoretical foundation of risk to determine appropriate baseline software supportability risk values and associated software supportability event probability distributions, and uncertainty boundaries on the risk and supportability evaluation measures.

Risk evaluation assimilates the determined risk estimation measures of software supportability and by applying the theoretical foundation of statistical risk assessment determines the degree of risk reduction and avoidance possible by selection of appropriate alternatives. From this foundation, the risk evaluation process establishes risk acceptance levels and identifies residual risk for each risk agent (called risk referents). It is after the risk referents have been established for software supportability that the Air Force decision maker can integrate supportability risk with other system risks and cost-benefit analysis to make ultimate decisions concerning system acceptability and support planning.

**Section V**  
**References**



SECTION V  
REFERENCES

- 5.0 "Software Risk Assessment in OT&E," Final Subtask Statement 304 for AFOTEC Contract F29601-80-C-0035, AFOTEC, Kirtland AFB, NM, Apr 84.
- 5.1 AFOTTECP 800-2 Volumes 1 through 5, Software OT&E Guidelines.
- 5.2 FIPS PUB 31, "Guidelines for ADP Physical Security and Risk Management," National Bureau of Standards, Jun 74.
- 5.3 FIPS PUB 65, "Guidelines for Automatic Data Processing Risk Analysis," National Bureau of Standards, Aug 79.
- 5.4 AFR 205-16, "Automatic Data Processing (ADP) Security Policy, Procedures, and Responsibilities," Attachment 5: Guidance for Performing Risk Analysis, 1 Aug 84.
- 5.5 OPNAVINST 5239.1A, "Department of the Navy Automatic Data Processing Security Program," Appendix E: Risk Assessment Methodology, 3 Aug 82.
- 5.6 Lathrop, F., "Alternative Methods for Risk Analysis: A Feasibility Study," Air Force Computer Security Program Office, 1 Sep 81.
- 5.7 Neugent, W., "Technology Assessment: Methods for Measuring the Level of Computer Security," Section 4.2: Risk Assessment Methodologies, National Bureau of Standards, Draft, Sep 81.
- 5.8 RADC, "Reliability Model Demonstration Study," RADC-TR-83-207, Volumes I and II, Aug 83.
- 5.9 Directorate of Aerospace Safety, "A Risk Management Guide for Air Force Operations," Air Force Inspection and Safety Center, Norton AFB, CA, 6 Nov 79.
- 5.10 USAF Scientific Advisory Board, "The High Cost and Risk of Mission-Critical Software," USAF SAB Ad Hoc Committee, Dec 83.
- 5.11 Fisher, G. and Lt. Col. E. Gay, "An Approach to Risk Analysis: A Process View," AF/SA Technical Note, Jun 81.
- 5.12 Fisk, F., and W. Murch, "A Proposal for Computer Resources Risk Assessment During Operational Test and Evaluation," AFOTEC Draft Report, 3 Oct 83.

- 5.13 Peercy, D., "A Framework for Software Maintenance Management Measures," Proceedings of the Seventeenth Annual Hawaii International Conference on System Sciences, Jan 84.
- 5.14 Peercy, D., and G. Swinson, "A Software Support Facility Evaluation Methodology," Proceedings of Symposium on Application and Assessment of Automated Tools for Software Development, Nov 83.
- 5.15 Booch, G., Software Engineering with Ada, Reading, MA: Benjamin/Cummings, 1983.
- 5.16 LeBlanc, R., and J. Goda, "Ada and Software Development Support: A New Concept in Language Design," Computer, 15(1982), 5, pp. 75-82.
- 5.17 Howden, W., "Contemporary Software Development Environments," Communications of the ACM, 25(1982), 5, pp. 313-329.
- 5.18 Lientz, B., and E. Swanson, Software Maintenance Management, Reading, MA: Addison-Wesley, 1980.
- 5.19 Lientz, B., and E. Swanson, "Problems in Application Software Maintenance," Communications of the ACM, 24(1981), 11, pp. 763-769.
- 5.20 GAO Report, "Federal Agencies Maintenance of Computer Programs: Expensive and Undermanaged," AFMD-81-25, Feb 81.
- 5.21 Parikh, G., Techniques of Program and System Maintenance, Cambridge, MA: Winthrop, 1982.
- 5.22 Thayer, R., A. Pyster, and R. Wood, "Validating Solutions to Major Problems in Software Engineering Project Management," Computer 15(1982), 3, pp. 65-77.
- 5.23 Boehm, B., J. Brown, and M. Lipow, "Quantitative Evaluation of Software Quality," Proceedings 2nd International Conference on Software Engineering, San Francisco, CA: 1976, pp. 592-605.
- 5.24 McCall, J., and M. Matsumoto, "Software Quality Measurement Manual," RADC-TR-80-109, Vol II (of two), Apr 80.
- 5.25 Rowe, W., An Anatomy of Risk, J. Wiley and Sons, New York, 1977.
- 5.26 Atzinger, E., and Brooks, N. (eds), "A Compendium on Risk Analysis Techniques," Aberdeen Proving Grounds, J.S. Army Materiel Systems Analysis Agency, 1972.
- 5.27 Megill, R., An Introduction to Risk Analysis, Petroleum Publishing, Tulsa, 1977.

- 5.28 Efron, B., The Jackknife, Bootstrap and Other Resampling Plans, Society for Industrial Mathematics, Philadelphia, 1982.
- 5.29 Worm, G.; "Applied Risk Analysis with Dependence Among Cost Components," Clemson University Department of Industrial Management, 1981.
- 5.30 Rescher, N., Risk, Washington, D.C.: University Press of America, 1983.
- 5.31 DeMillo, R., "A Risk Model for Software Testing," Georgia Institute of Technology, Briefing Slides, July 20, 1984.
- 5.32 Huebner, W., D. Percy, G. Richardson, "Software Supportability Risk Assessment in OT&E: An Evaluation of Risk Methodologies," BDM/A-84-496-TR, Aug 1984.

**Appendix A**  
**Acronyms**

APPENDIX A  
ACRONYMS

ACM	Association for Computing Machinery
ADP	Automatic Data Processing
AFCSP0	Air Force Computer Security Program Office
AFLC	Air Force Logistics Command
AFOTEC	Air Force Operational Test and Evaluation Center
AFR	Air Force Regulation
AFRAMP	Air Force Risk Analysis Management Program
AFSAB	Air Force Scientific Advisory Board
AF/SA	Air Force Studies and Analysis
AF/SASF	Air Force Studies and Analysis Strategic Force
AF/SATF	Air Force Studies and Analysis Tactical Force
APSE	Ada Programming Support Environment
ATAM	Automated Threat Analysis Methodology
ATE	Automatic Test Equipment
CRISP	Computer Resources Integrated Support Plan
CSOC	Consolidated Space Operations Center
CSPO	Computer Security Program Office
CSS	Computer System Security
C <sup>3</sup> I	Command, Control, Communications and Intelligence
DACS	Data and Analysis Center for Software
DID	Data Item Description
DPI	Data Processing Installation
DoD	Department of Defense
DTIC	Defense Technical Information Center
ECS	Embedded Computer System
EW	Electronic Warfare
FIPS PUBs	Federal Information Processing Standards Organization (of the National Bureau of Standards) Publications (Series)
GAO	Government Accounting Office
HANDSE	"Handbook for the Deputy for Software Evaluation" (AFOTEC Publication)

IV&V	Independent Verification and Validation
MOE	Measure of Effectiveness
NBS	National Bureau of Standards
NTIS	National Technical Information Service
O&M	Operation and Maintenance
OT&E	Operational Test and Evaluation
PDSS	Post Deployment Software Support
PMRT	Program Management Responsibility Transfer
RA	Risk Assessment
RADC	Rome Air Development Center
RAMP	Risk Analysis and Management Plan
RAMSS	Risk Assessment Model for Software Supportability
SAB	See AFSAB
SEL	Software Engineering Laboratory
SSF	Software Support Facility
STARS	Software Technology for Adaptable and Reliable Systems
T&E	Test and Evaluation
TEMP	Test and Evaluation Master Plan
WIS	WWMCCS Information System

**Appendix B**  
**Glossary of Terms**

APPENDIX B  
GLOSSARY OF TERMS

B.1 INTRODUCTION.

The glossary of terms for the Analysis of Software Supportability Risk Assessment models is relevant to the entire subtask environment and content of all the subtask reports.

Some terms have more than one description; when this is the case, the descriptions either:

- a) Are significantly different between sources (though the effective meaning may be not much different).
- b) Are used differently (different users or technical language).
- c) May be found within the context of a different source.
- d) Have real differences in meaning.

Both DoD and non-DoD (e.g., FIPS PUBs, NBS Special Publications) sources are used. The non-DoD sources and terms are not mandated for our use, but are rather included for breadth of understanding, for those relevant terms commonly used within the non-DoD governmental and/or private sectors.

The source of each description is indicated by a symbol in parentheses before that source's term description:

```

TERM1
  (SYMBOL1.1)
  Description1.1...
  (SYMBOL1.2)
  Description1.2...
  .
  .
  (SYMBOL1.n)
  Description1.n...
TERM2
  .
  .
TERMN

```



The symbols used and corresponding sources are:

- (AFOTECP1) AFOTECP 800-2, Volume 1, 10 Nov 82, "Software Test Manager's Guide."
- (AFR800-14) Air Force Regulation 800-14, Volume I, "Management of Computer Resources in Systems," 12 Sep 75.
- (AFR300-15) Air Force Regulation 300-15, "Automated Data System Project Management," Jan 78.
- (AFOTECP5) AFOTECP 800-2, Volume 5, 25 Jul 83, "Software Support Facility Evaluation--User's Guide."
- (ROWE) Rowe, William, An Anatomy of Risk, John Wiley, 1977.
- (LATHROP) Lathrop, Frank, "Alternative Methods for Risk Analysis: A Feasibility Study," Air Force Computer Security Program Office, 1 Sep 81.
- (AFR205X) Air Force Regulation 205-16, "Automatic Data Processing (ADP) Security Policy, Procedures and Responsibilities," 1 Aug 84.
- (AFOTECP3) AFOTECP 800-2, Volume III, 1 Jan 84, "Software Maintainability Evaluator's Guide."
- (CURRENT) Current document definition.

8.2 GLOSSARY OF TERMS FOR THE ANALYSIS FOR DETERMINING FEASIBILITY OF DEVELOPING AND IMPLEMENTING A RISK ASSESSMENT MODEL FOR SOFTWARE SUPPORTABILITY.

Accuracy

(ROWE)

The quality of being free from error. The degree of accuracy is a measure of the uncertainty in identifying the true measure of a quantity at the level of precision of the scale used for the quantity.

Algorithm

(AFOTECP3)

A prescribed set of well-defined rules or processes for the solution of a problem in a finite number of steps.

Allocated Baseline

(AFR300-15)

The initial approved allocated configuration identification established at end of the definition phase.

Alternative

(ROWE)

One member of a set of options associated with a decision, the decision being limited to a choice of one and only one.

Application Functions

(AFOTECP3)

Any functions which provide specific operational (mission) computations.

Application Software

(AFOTECP5)

The software written by software support personnel, or purchased from a contractor, used directly in supporting ECSs. It is normally used for simulation, testing, and ECS code development.

Application Software (functional)

(AFR205C)

Those routines and programs designed by or for automatic data processing system users and customers to complete specific, mission-oriented tasks, jobs, or functions, using available automated data processing equipment and basic software. Application Software may be either general purpose packages, such as demand deposit

accounting, payroll, machine tool control, etc., or specific application programs tailored to complete a single or limited number of user functions (for example, base level personnel, depot maintenance, aircraft, missile or satellite tracking, command and control, etc.). Except for general purpose packages that are acquired directly from software vendors or from the original equipment manufacturers, this type of software is generally developed by the user, either with in-house resources or through contract services.

#### Approval to Operate

(AFR205X)

Represents concurrence by the designated approving authority (DAA) that a satisfactory level of security (that is, minimum requirements are met and an acceptable level of risk exists) has been provided, and authorizes the operation of an automated data processing system (ADPS) or network at an automatic data processing facility (ADPF). Approval results from an analysis of the ADPF, ADPS, and automatic data system (ADS) certifications and the operational environment of the automatic data processing (ADP) entity by the DAA.

#### Attributes

(AFOTECP3)

Type, units, range, description, etc., as appropriate.

#### Automated Decisionmaking System

(AFR205X)

Those computer applications which issue checks, requisition supplies, or perform similar functions based on programmed criteria, with little human intervention.

#### Automated Software Development Tool

(AFOTECP5)

A component of System Software that assists in the design, implementation, documentation, and verification of ECS software.

#### Automatic Data Processing Facility (ADPF)

(AFR205X)

The physical resources, including structures or parts of structures, which house and support data processing capabilities. For each computer facility designated as a data processing installation (DPI, reference AFR 300-6), the ADPF is the DPI. For small computers, stand-alone systems, and word processing equipment, the ADPF is the physical area in which the computer is used.

### Automatic Data Processing Resources

(AFR205X)

The totality of automatic data processing equipment, software, data, computer time, computer programs, automatic data processing (ADP) contractual services, ADP personnel, and supplies.

### Automatic Data Processing Security

(AFR205X)

Includes all hardware and software functions, characteristics, and features; operational procedures, accountability procedures, and access controls at all automatic data processing facilities (including those housing mainframes, terminals, minicomputers, or microcomputers); the management constraints, the physical environment, control of compromising emissions (TEMPEST); and personnel and communications security needed to provide an acceptable level of protection for hardware; software; and sensitive or critical data, material, or process, classified or otherwise, in the system.

### Automatic Data Processing Security Plan

(AFR205X)

The overall plan for providing security throughout the life cycle of automated project or program, automated data processing system, or facility. The plan documents the operational requirements, security environment, hardware and software configurations and interfaces; all security procedures, measures, and features; and, for automatic data processing facilities, the contingency plans for continued support in case of a local disaster. The plan represents the baseline for the risk analysis.

### Availability

(AFR800-14)

A measure of the degree to which an item is in the operable and committable state at the start of the mission, when the mission is called for at an unknown (random) point in time. (MIL-STD-721)

(AFOTEC P5)

The probability that a system is operating satisfactorily at any point in time when used under stated conditions.

### Axiology

(ROWE)

The study of the nature of types and criteria of values and of value judgements, especially in ethics.

### Backup System

(AFOTECP5)

An additional computer system which is available to perform the functions of a support system that fails to operate.

### Baseline

(AFR300-15)

A configuration identification document or set of such documents formally designated and fixed at a specific time during a CPCI's life cycle. Baselines, plus approved changes to those baselines constitute the current configuration identification.

(ROWE)

A known reference used as a guide for further development activities.

### Baseline Change Request (BCR)

(ARF300-15)

A request for alteration in the configuration of a computer program configuration item that is delivered or under development, after formal establishment of its configuration identification.

### Baseline Profile

(CURRENT)

The set of 27 pairs of numbers (or any subset) determined by specifying the (time to complete request, number of requests per unit time) pair for each maintenance request category.

### Basic Software (nonfunctional)

(AFR205X)

Those routines and programs designed to extend or facilitate the use of particular automatic data processing (ADP) equipment. As a rule, the ADP vendor provides this software which is usually essential for the system operation. Examples of basic software are executive and operating systems, diagnostic programs, compilers, assemblers, utility routines such as sort-merge and input or output conversion routines, file management programs, and data management programs. Data management programs are commonly linked to or under the control of the executive or operating system programs.

### Bayesian Statistics

(ROWE)

"Bayes Rule" (Thomas Bayes, a nineteenth century English mathematician and clergyman) states that the probability that both of two events will occur is the probability of the first multiplied by the probability that if the first has occurred, the second will also

occur. Bayesian statistics is a way of making quantity of information substitute for quality of information. There are two kinds of probability: the classical type derived from empirical information, and subjective probability. Bayesian statistics is based on these "subjective probabilities." It involves the joint probability of A and B. The probability of the second event occurring if the first has occurred is called the conditional probability of the second, given the first. Stated another way, the probability of any event  $P(A)$  is always positive but never greater than 1. Symbolically,  $0 < P(A) \leq 1$ . If  $P(A) = 0$ , the occurrence of the event B is considered impossible. If  $P(A) = 1$ , the occurrence of the event B is considered to occur with  $P(B)$ .

#### Behavior

(ROWE)

The observable manifestations of performance.

#### Benefit

(ROWE)

- a) An axiological concept representing anything received that causes a net improvement to accrue to the recipient.
- b) A result of a specific action that constitutes an increase in the production possibilities or welfare level of society.

#### Benefit-Cost Ratio

(ROWE)

The ratio of total social benefit to total social costs related to a specific activity.

#### Capability

(ROWE)

A measure of the degree to which a system is able to satisfy its performance objectives.

#### Cardinal (interval) Scale

(ROWE)

A continuous scale between two end points, neither of which is necessarily fixed.

#### Central Processing Unit (CPU)

(AFOTECPS)

A part of a computer system that performs all calculations and controls what is done by all other parts of the system (called peripherals) and may include central memory and input/output interfaces.

## Certification

(AFR205X)

A statement by the appropriate manager (automated data processing system (ADPS), automatic data system (ADS), or automatic data processing facility (ADPF)) of the extent to which the security measures in the system or facility meet specifications. Certification is based upon the results of the risk analysis performed and does not necessarily imply a guarantee that the "system" described is nonpenetrable. It is an input to the security approval process.

## Complexity Level

(CURRENT)

The general degree of difficulty to complete a maintenance request: high, medium, low.

## Computer Abuse

(AFR205X)

Willful or negligent unauthorized activity affecting the availability, confidentiality, or integrity of automatic data processing resources. Computer abuse includes fraud, embezzlement, theft, malicious damage, unauthorized use, denial of service, and misappropriation. Level of computer abuse are:

- (1) Minor Abuse. Acts which represent management problems, such as the printing of calendars or the running of games, which do not impact system availability for authorized applications.
- (2) Major Abuse. Unauthorized use (possibly criminal), denial of service, and multiple instances of minor abuses, including waste.
- (3) Criminal Act. Fraud, embezzlement, theft, malicious damage, misappropriation, conflict of interest, and unauthorized accesses to classified data.

## Computer Program

(AFR800-14)

A series of instructions or statements in a form acceptable to an electronic computer, designed to cause the computer to execute an operation or operations.

## Computer Program Configuration Item (CPCI)

(AFR300-15)

An ADS or portion of an ADS that is designated for configuration management.

### Computer Resources

(AFR800-14)

The totality of computer equipment, computer programs, associated documentation, contractual services, personnel and supplies.

### Computer System

(AFOTECP5)

Both the Hardware and the System Software.

### Configuration Audit

(AFR300-15)

A process to verify conformance to specifications and standards.

### Configuration Control

(AFR300-15)

The systematic evaluation, coordination, approval or disapproval, and implementation of approved changes in the configuration of a CPCI after formal establishment of its configuration identification.

### Configuration Control Board (CCB)

(AFR300-15)

A board composed of representatives from program/project office and using/supporting organizations.

### Configuration Identification

(AFR300-15)

The currently approved technical description of the CPCI or ADS.

### Configuration Item (CI)

(AFR300-15)

An item of ADPE that is designated for configuration management.

(AFR900-14)

An aggregation of equipment software, or any of its discrete portions, which satisfies an end use function and is designated by the Government for configuration management. CIs may vary widely in complexity, size and type, from an aircraft or electronic system to a test meter or round of ammunition. During development and initial production, CIs are only those specification items that are referenced directly in a contract or an equivalent in-house agreement. During the operation and maintenance period, any repairable item designated for separate procurement is a configuration item.

(AFR 65-3)



Configuration Management (CM)

(AFR300-15)

A management discipline that applies technical and administrative direction and surveillance to:

- (1) Identify and document the functional and physical characteristics of a configuration item.
- (2) Control changes to those characteristics.
- (3) Record and report configuration status.

Configuration Management Plan (CMP)

(AFR300-15)

A document which describes project responsibilities and procedures for implementing CM.

Configuration Management System (CMS)

(AFOTECP5)

A system applying technical and administrative direction and surveillance to identify and document the functional and physical characteristics of a configuration item; to control changes to those characteristics and to record and report change processing and implementation status.

Configuration Status Accounting

(AFR300-15)

The recording and reporting of the approved configuration identification, the status of the proposed changes to the approved configuration, and the implementation status of approved changes.

Consequence Value

(ROWE)

The importance a risk agent subjectively attaches to the undesirability of a specific risk consequence.

Consensus

(ROWE)

Group solidarity in sentiment and belief...general agreement.

Contractor

(AFOTECP5)

Person working at a software support facility who is employed by a private company rather than by the Government (as military or civilians). Most often the company will be the one that produced the ECS.

## Controlled Security Mode

(AFR205X)

A mode of operation where internal security controls prevent inadvertent disclosure. Personnel, physical, and administrative controls prevent deliberate, malicious attempts to gain unauthorized access. A system operating in the controlled security mode may service both cleared and uncleared users and, if required, may independently service both secured and unsecured remote terminal areas.

## Cost

(ROWE)

A result of a specific action that constitutes a decrease in the production possibilities or welfare level of society or a net loss.

## Cost-Benefit Analysis

(ROWE)

An attempt to delineate and compare, in terms of society as a whole, the significant effects, both positive and negative, of a specific action. Generally a number of alternative actions are analyzed resulting in the selection of the alternative that provides either the largest benefit-cost ratio (total benefit/total cost) or one with a positive ratio at least. If an alternative results in a net benefit less than zero or a benefit-cost ratio less than 1, it is deemed socially inefficient and is not carried out.

## Cost-Effectiveness Analysis

(ROWE)

A term less specific than cost-benefit analysis, usually meaning the selection of the lowest cost alternative that achieves a predetermined level of benefits. Alternatively, the analysis may determine of the path that yields the largest social benefit for a predetermined specified level of social costs.

## Critical Automatic Data Processing Resources

(AFR205X)

Those resources that must be protected because their compromise, alternation, destruction, loss, or malfunction could significantly jeopardize the accomplishment of an Air Force, Air Force component, or other service mission or the accomplishment of a critical support function.

## Critical Design Review (CDR)

(AFR300-15)

A formal review conducted during the development phase of a project involving logic, and algorithms to coded instruction.

### Critical Issues

(AFOTECP1)

Those aspects of a system's capability, either operational, technical, or other, that must be questioned before a system's overall worth can be estimated and that are of primary importance to the decision authority in reaching a decision to allow the system to advance into the next acquisition phase. (DoD Directive 5000.3).

### Data Base Change Request (DBCR)

(AFR300-15)

A form used to initiate and control data base changes after the data base is placed under configuration control.

### Data Item Description

(AFR800-14)

A form which specifies an item of data required to be furnished by a contractor. This form specifically defines the content, preparation instructions, format and intended use of each data product. (AFR 310-1)

### Decision Analysis

(ROWE)

A methodology of decomposition of the decision-making process into parts, whereby the appropriate data can be associated with the parts, to provide a rational basis for decision making.

### Decision Making

(ROWE)

A dynamic process of interaction, involving information and judgment among participants who determine a particular policy choice. Decision models are either models of the decision-making process itself, or analytical models (e.g., decision trees, decision matrices) used as aids in arriving at the decisions. Decision theories usually are in relation to the process itself.

### Decision Matrices

(ROWE)

Matrices whose elements exhibit quantitative relationships (cardinal or ordinal) among sets of factors coming into play in the decision-making process.

  
Decision Tree

(ROWE)

A device used to portray alternative courses of action and relate them to alternative decisions showing all consequences of the decision. The tree represents alternative courses or series of actions related to a previous decision.


## Decisive Decision Conditions

(ROWE)

Conditions in which the preference between values on a utility scale is clearly discernible because ranges of uncertainty of the two values do not overlap (in the case of uniform distributions of uncertainty) or are below a certain error level (for normal distributions of uncertainty).

## Dedicated Security Model

(AFR205X)

A mode of operation where the automatic data processing system (ADPS), its peripherals and remotes are exclusively used and controlled by specific users or groups of users for processing a particular type and category of classified or otherwise sensitive material. All users of the system have clearances and need-to-know for all material in the ADPS.

## Degree of Uncertainty

(ROWE)

That proportion of information about a total system that is unknown in relation to the total information about the system.

## Delphi Technique

(ROWE)

An iterative method designed to produce a consensus by repeated queries of an individual with feedback of group responses. Members of the group do not interact directly.


## Descriptive Uncertainty

(ROWE)

The absence of information about the completeness of the description of the degrees of freedom of a system.

## Design Problem Report (DPR)

(AFR300-15)

A form used for documenting problems identified during reviews and audits.

Designated Approving Authority

(AFR205X)

An official designated to approve the operation of automatic data processing systems at the automatic data processing facilities under his or her jurisdiction for storage of classified or sensitive unclassified information or for critical processing.

Development Test Plan (DT)

(AFR300-15)

A document which specifies the method and content for development testing from the lowest compilable level up through the complete computer program configuration item. Defines test management, reports, controls, manpower, acceptance criteria, and test procedures.

Development Testing

(AFR300-15)

Testing of computer programs by the development programmers and analysts prior to EST I.

Deviation

(AFR300-15)

A written authorization, granted prior to the development of a CPCI, to depart from a particular performance or design requirement; a specification for a specific number of units; a specific period of time; or established standards.

Documentation

(AFOTECP5)

All of the written work describing operating and maintenance procedures for a system.

Documentation Consistency


(AFOTECP5)

A measure of the consistency in the information provided in support system documentation.

Documentation Descriptiveness

(AFOTECP5)

A measure of the descriptiveness of the information provided in support system documentation.

  
Documentation Modularity

(AFOTTECP5)

A measure of the modular organization of information provided in support system documentation.

## Documentation Simplicity

(AFOTTECP5)

A measure of the ease of use and lack of complexity in the information provided in computer system documentation.


## Economic Assessment

(AFR205X)

A detailed study of security measures, their operational and technical feasibility, and their costs and benefits. Economic assessment is used in planning and selecting security measures.

## Embedded Computer Resources

(AFOTTECP1)

Computer resources incorporated as integral parts of, dedicated to, required for direct support of, or for the upgrading or modification of major or less than major system(s). (Excludes ADP resources as defined and administered under AFR 300 series.) (USAF/RD/LE Policy letter, 13 October 1981).

## Embedded Computer System (ECS)


(AFOTTECP1)

a) A computer that is integral to an electromechanical system and that has the following key attributes:

- (1) Physically incorporated into a large system whose primary function is not data processing.
- (2) Integral to, or supportive of, a larger system from a design, procurement, and operations viewpoint.
- (3) Inputs include target data, environmental data, command and control, etc.
- (4) Outputs include target information, flight information, control signals, etc.

b) In general, an embedded computer system (ECS) is developed, acquired, and operated under decentralized management. (DoD Directives 5000.1, 5000.2).

(AFOTTECP5)

A computer that is integral to an electronic or electromechanical system (e.g., aircraft, missile, spacecraft, communications device) from a design, procurement, and operational viewpoint.

Empirical

(ROWE)

Originating in or based on observation or experience.

Endogeneous Risk Imposition

(ROWE)

Choice of risk exposure is under control of the risk agent alone.

Environment

(AFOTECP5)

The air conditioning, lighting, and safety features that improve working conditions within facilities.

Equitable Risk

(ROWE)

A risk agent receives direct benefits as a result of exposure to a risk, and the knowledge of the risk is not purposely withheld from the risk agent.

Error Processing

(AFOTECP3)

The steps required to set program data and control statements following the detection of an undesirable event.

Estimation

(ROWE)

The assignment of probability measures to a postulated future event.

Estimator Uncertainty

(ROWE)

Uncertainty in measurement resulting from deliberate use of less complex measures such as central value estimates of dispersion and smoothing functions for time-dependent parameters.

Evaluation

(ROWE)

Comparison of performance of an activity with the objectives of the activity and assignment of a success measure to that performance.

Evaluation Criteria

(AFOTECPI)

Standards by which achievement of required operational effectiveness/suitability characteristics or resolution of technical or operational issues may be judged. For full-scale development and beyond, evaluation criteria must include quantitative goals (the desired value) and thresholds (the value beyond which the characteristic is unsatisfactory) whenever possible. (OOB Directive 5000.3).

Event

(ROWE)

A particular point in time associated with the beginning or completion of an activity, and possibly accompanied by a statement of the benefit or result attained or to be attained because of the completion of an activity.

Exogeneous

(ROWE)

External to a system (part of the environment of the system).

Exogeneous Risk Imposition

(ROWE)

Choice of risk exposure is not under control of the risk agent alone

Expandability

(AFOTECPS)

A measure of the ease with which the functional capability of computer hardware or software may be expanded.

Expected Value, Use Of

(ROWE)

Valuation of an uncertain numerical event by weighting all possible events by their probability of occurrence and averaging.

Expert Judgment

(ROWE)

Designating the relevance of opinions of persons well informed in an area for estimates (e.g., forecasts of economic activity).



Exposure (to risk)

(ROWE)

The condition of being vulnerable to some degree to a particular outcome of an activity, if that outcome occurs.

Extrapolation/Projection

(ROWE)

The technique of estimating the future by a continuation of past trends without attempts to understand the underlying phenomena.

Facility

(AFOTECPS)

The physical plant and the services it provides; specific examples are physical space, electrical power, physical and electromagnetic (TEMPEST) security, environmental control, fire safety provisions, and communications availability.

Feasible

(ROWE)

That which is possible to do, realistically.

Feedback

(ROWE)

The return of performance data to a point permitting comparison with objective data, normally for the purpose of improving performance (goal-seeking feedback), but occasionally to modify the objective (goal-changing feedback).

Firmware

(AFOTECPI)

- a) Computer programs and data loaded in a class of memory that cannot be dynamically modified by the computer during processing.
- b) Hardware that contains a computer program and data that cannot be changed in its application environment.

Note 1. The computer programs and data contained in firmware are classified as software; the circuitry containing the computer program and data is classified as hardware. (Data and Analysis Center for Software).

Functional Baseline

(AFR300-15)

The initial approved functional configuration identification.

## Functional Configuration Audit (FCA)

(AFR300-15)

The formal examination of CPCI to verify that the performance specified in the SS has been achieved.

## Hardware

(AFOTECP5)

The CPU and all of the peripheral devices that are the physical components of a computer system.

## Higher Level Programming Languages

(AFR800-14)

Primarily, machine independent programming languages (of a higher order than assembly languages) designed for ease of expression of a class of problems or procedures by humans. These languages are designed for convenience of program specification rather than for easy conversion to machine code instruction. The languages are intended:

- (1) As a means for directly presenting procedures to a computer for which a compiler exists; and
- (2) As a means of communicating such procedures among individuals. (AFR 300-10)

## Independent Verification and Validation (IV&amp;V)

(AFOTECP1)

An independent assessment process structured to ensure that computer programs fulfill the requirements stated in system and subsystem specifications and satisfactorily perform the functions required to meet the user's and supporter's requirements. IV&V consists of three essential elements: independence, verification, and validation:

- (1) Independent. An organization/agency which is separate from the software development activity from a contractual and organizational standpoint.
- (2) Verification. The evaluation to determine whether the products of each step of the computer program development process fulfill all requirements levied by the previous step.
- (3) Validation. The integration, testing, and/or evaluation activities carried out at the system/subsystem level to evaluate the developed computer program against the system specifications and the user's and supporter's requirements. (AFR 88-14)

### Individual Risk Evaluation

(ROWE)

The complex process, conscious or unconscious, whereby an individual accepts a given risk.

### Inequitable Risk

(ROWE)

A risk agent is exposed to a risk and receives no direct benefits from such exposure, or the knowledge of the risk is purposely withheld from him.

### Interdependence

(ROWE)

A property shared by two or more entities whenever the performance of any one affects the performance of some or all the rest.

### Interoperability

(AFOTECPS)

A measure of the degree to which computer hardware or software can interface to and operate with other similar computer hardware or software

### Intrinsic Parameter

(ROWE)

A variable whose measurement is based on the value system of an individual and his perception of these values.

### Laboratory-Integrated Test Facility

(AFOTECPS)

A facility used to integrate and test hardware and software systems, by exercising the operational software on the Target Computer in a simulated operational environment. Includes operator controlled displays and all or most of the actual equipment which tie directly to the target computer. Also, the portion of Support System Facility required to house the laboratory-integrated test facility.

### Loss Function

(ROWE)

A function used in decision theory for evaluating the losses incurred when certain decisions are made under uncertainty. If the loss function is independent of the decision value used, it is frequently called a cost function.

Maintainability

(AFOTECP3)

Those characteristics of software which affect the ability of the software programmer to correct errors, enhance system capabilities through software changes, and modify the software to be compatible with hardware changes.

(AFOTECP5)

The probability that a system out of service for maintenance can be properly repaired and returned to service in a stated elapsed time.

Maintenance Documentation

(AFOTECP5)

The documentation that describes the maintenance of computer system hardware and software.

Maintenance Request Category

(CURRENT)

The identification of a maintenance request by specification of the priority type, maintenance type, and complexity level.

Maintenance Type

(CURRENT)

The type of maintenance actions required to complete a maintenance request: enhancement, conversion, correction.

Measurable

(ROWE)

a) Capable of being sensed, that which is sensed being convertible to an indication; the indication can be logical, axiological, numerical, or probabilistic. If probabilistic, it is empirical and subjective.

b) Comparable to some unit designated as standard.

Measured Risk Level

(ROWE)

The historic, measured, or modeled risk associated with a given activity.

Measurement Uncertainty

(ROWE)

The absence of information about the specific value of a measurable variable.

Methodology

(ROWE)

An open system of procedures.

Model

(ROWE)

An abstraction of reality that is always an approximation to reality.

Module

(AFR300-15)

A program unit that is discrete and identifiable with respect to compiling and combining with other units.

Multilevel Security Mode

(AFR205X)

A mode of operation that provides a capability for various levels and categories or compartments of data to be concurrently stored and processed in an automatic data processing system and permits selective access to such material concurrently by personnel (users) who have differing security clearances and need-to-know. Internal controls, as well as personnel, physical, and administrative controls, separate users and data on the basis of security clearance and need-to-know. The internal security controls must be thoroughly demonstrated to be effective in preventing deliberate malicious attempts to gain unauthorized access to classified information. This mode of operation can accommodate the concurrent processing and storage of two or more levels of classified data, or one or more levels of classified data with unclassified data, depending on the constraints that the DAA places on the system.

Negative Systemic Control

(ROWE)

The absence of a systemic control concept and/or a system whose risk behavior is characterized by an increase in risk overtime.

Nominal Scale (taxonomy)

(ROWE)

A classification of items that can be distinguished from one another by one or more properties.

Objective Function

(ROWE)

A specified mathematical relationship between a dependent variable (e.g., overall measure of benefits) and a set of independent

variables (e.g., individual benefit measures and their relative weights). In choosing among alternatives, the decision maker typically seeks to maximize the (dependent variable of the) objective function.

#### Operating System

(AFR205X)

An integrated collection of service routines for supervising the sequencing and processing of programs by a computer. Operating systems control the allocation of resources to users and their programs and play a central role in operating a computer system. Operating systems may perform input or output, accounting, resource allocation, storage assignment tasks, and other system related functions. (Synonymous with monitor, executive control program, and supervisor.)

#### Operational Effectiveness

(AFOTECPI)

The overall degree of mission accomplishment of a system used by representative personnel in the context of the organization, doctrine, tactics, threat (including countermeasures and nuclear threats), and environment in the planned operational employment of the system. (DoD Directive 5000.3)

#### Operational-Integrated Test Facility

(AFOTECPI5)

A facility used to perform final testing of a field-configured system in an actual or representative operational environment. Also, the portion of Support System Facility required to house the operational-integrated test facility.

#### Operational Suitability

(AFOTECPI)

The degree to which a system can be satisfactorily placed in use, with consideration being given availability, transportability, interoperability, reliability, maintainability, safety, human factors, logistic supportability, and training. (DoD Directive 5000.3)

#### Opinion Survey/Sampling

(ROWE)

Any procedure for obtaining or analyzing the views of any portion of the population of benefit levels expected, their distribution. Typically, scientific sampling is used.

AD-A191 874

SOFTWARE SUPPORTABILITY RISK ASSESSMENT IN OT&E  
(OPERATIONAL TEST AND EVA. (U) BDM CORP ALBUQUERQUE NM  
M HOESSEL ET AL. 28 SEP 84 BDM/A-84-322-TR

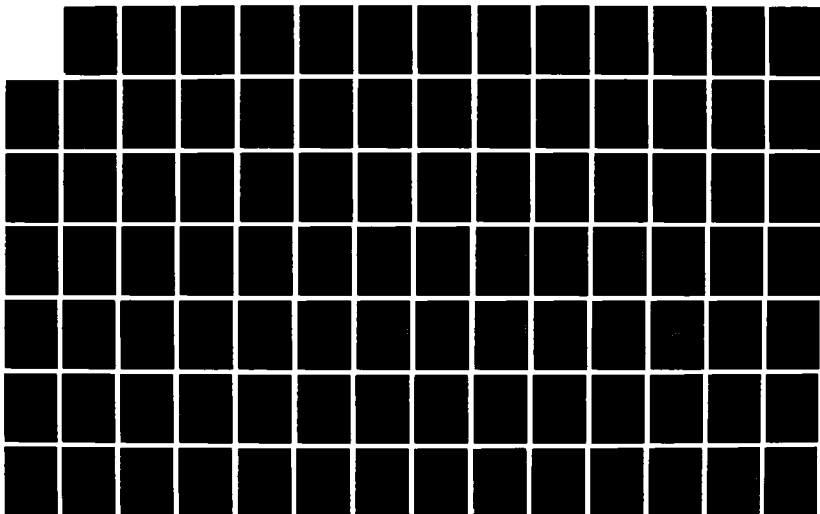
2/4

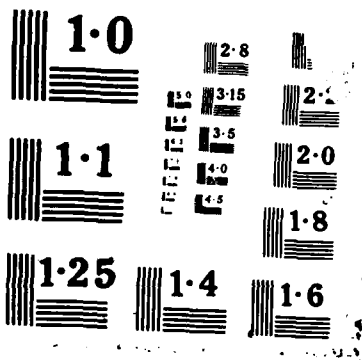
UNCLASSIFIED

F29681-88-C-0035

F/G 12/5

NL







(for a given level of effort) the accuracy and precision of the results obtained.

Opportunity Cost

(ROWE)

The value to society of the next best alternative use of a resource. This is the true economic cost to society of using a resource for a specific purpose or in a specific project.

Ordinal Scale (rank scale)

(ROWE)

An ordering (ranking) of items by the degree to which they satisfy some criterion.

Paradigm

(ROWE)

A structured set of concepts, definitions, classifications, axioms, and assumptions used in providing a conceptual framework for studying a given problem.

Parametric Variation

(ROWE)

A technique for sensitivity analysis of any given model in which the values of parameters that are input to the model's calculation are systematically varied to permit observation of how such variation affects the model's output (especially ranking of alternatives).

Pareto Optimization

(ROWE)

Optimization using a criterion that each person's needs be met as much as possible without diminishing the degree of achievement of any other person.

Peripheral

(AFOTECP5)

A hardware element of a computer system, controlled by the CPU, including Mass Storage device, Paper Tape Reader and punch, card reader and punch, Printer, Plotter, Video Display Terminal, Data Communications, and other similar devices.

Personnel

(AFOTECP5)

A general term for the experience, education, and quantity of people who are assigned to the software support facility either directly or

indirectly maintaining the ECS. It includes Management, Technical, Support, and Contractor resources.

#### Personnel Profile

(AFOTECP5)

The characteristics that describe the experience, education, and quantity of software support facility personnel.

#### Physical Configuration Audit (PCA)

(AFR300-15)

The formal examination of the coded version of a computer program configuration item against its technical documentation.

#### Precision

(ROWE)

The exactness with which a quantity is stated; that is, the number of units into which a measurement scale of that quantity may be meaningfully divided. The number of significant digits is a measure of precision.

#### Predictive Modeling

(ROWE)

Use of any mathematic model that estimates or predicts the value of a dependent variable in terms of component factors specified as independent variables.

#### Preference

(ROWE)

Assignment of rank to items by an agent when the criterion used is utility to the ranking agent.

#### Preliminary Design Review (PDR)

(AFR300-15)

A formal review of the subsystem design approach for a CPCI occurring between the SDR and CDR.

#### Priority Type:

(CURRENT)

The criticality of the maintenance request in order to preserve mission readiness: emergency, urgent, normal.

#### Probability

(ROWE)

A numerical property attached to an activity or event whereby the likelihood of its future occurrence is expressed or clarified.

Probability Distribution

(ROWE)

The representation of a repeatable stochastic process by a function satisfying the axioms of probability theory.

Probability of Occurrence

(ROWE)

The probability that a particular event will occur, or will occur in a given interval.

Probability Threshold

(ROWE)

A probability of occurrence level for a risk below which a risk agent is no longer concerned with the risk and ignores it in practice (Threshold of concern).

Product Baseline

(AFR300-15)

The initial approved product configuration identification.

Product Verification Review (PVR)

(AFR300-15)

A formal review conducted by the developer for each CPCI at the end of the development phase to establish the Product Baseline for that CPCI and to ensure preparation for the Test Phase has been completed.

Program Manager

(AFR800-14)

The generic term used to denote a single Air Force manager (System Program Director, Program/Project Manager, or System/Item Manager) during any specific phase of the acquisition life cycle. (AFR 800-2).

Program Management Directive (PMD)

(AFR800-14)

The official HQ USAF management directive used to provide direction to the implementing and participating commands and satisfy documentation requirements. It will be used during the entire acquisition cycle to state requirements and request studies as well as initiate, approve, change, transition, modify or terminate programs. The content of the PMD, including the required HQ USAF review and approval actions, is tailored to the needs of each individual program. (AFR 300-2)

Program Management Plan (PMP)

(AFR800-14)

The document developed and issued by the Program Manager which shows the integrated time-phased tasks and resources required to complete the task specified in the PMD. The PMP is tailored to the needs of each individual program. (AFR 800-2)

Program Office (PO)

(AFR800-14)

The field office organized by the Program Manager to assist him in accomplishing the program tasks. (AFR 800-2)

Program Support Tools

(AFOTECP3)

General debug aids, test/retest software, trace software/hardware features, use of compiler/link editor, library management/configuration management/text editor/display software tools.

Program Test Plan

(AFOTECP3)

Set of descriptions and procedures for how the program is to be (or can be, or has been) tested.

Programming Conventions

(AFOTECP3)

Standards which are used to develop computer programs. Preface content, variable/module names, source code and embedded comment formats, I/O, error handling, etc.

Propensity for Risk Acceptance

(ROWE)

An individual, subjective trait designating the degree of risk one is willing to subject himself to for a particular purpose.

Quality Assurance (QA)

(AFR300-15)

All actions that are taken to assure that a development organization delivers products that meet performance requirements and adhere to standards and procedures.

Quantification

(ROWE)

The assignment of a number to an entity or a method for determining a number to be assigned to an entity

Recovery

(AFOTECP3)

- The procedures taken to report/correct some program failure (error processing).

Reliability

(ROWE)

The probability that the system will perform its required functions under given conditions for a specified operating time.

Residual Risk

(AFR205X)

That portion of risk which remains after security measures have been applied.

Risk

(AFR205X)

The loss potential which exists as the result of threat/vulnerability pairs. Reducing either the threat or the vulnerability reduces the risk.

(ROWE)

The potential for realization of unwanted, negative consequences of an event.

Risk Acceptance

(ROWE)

Willingness of an individual, group, or society to accept a specific level of risk to obtain some gain or benefit.

Risk Acceptance Function

(ROWE)

A subjective operator relating the levels of probability of occurrence and value of a consequence to a level of risk acceptance.

Risk Acceptance Level

(ROWE)

The acceptable probability of occurrence of a specific consequence value to a given risk agent.

Risk Acceptance Utility Function

(ROWE)

The profile of the acceptability of the probability of occurrence for all consequences involved in a risk situation for a specific risk agent.

Risk Agent

(ROWE)  
See Valuing Agent.

Risk Analysis

(AFR205X)  
A part of risk management that is used to minimize risk by effectively applying security measures commensurate with the relative threats, vulnerabilities, and values of the resources to be protected. (The value of the resources includes impact on the organizations the automatic data processing system supports, and impact of the loss or unauthorized modification of data). Risk analysis may be thought of as consisting of four modules: sensitivity assessment, risk assessment, economic assessment, and security test and evaluation.

Risk Assessment

(AFR205X)  
A detailed study of the vulnerabilities, threats, likelihood, loss or impact, and theoretical effectiveness of security measures. The results of a risk assessment may be used to develop security requirements and specifications.

(ROWE)  
The total process of quantifying a risk and finding an acceptable level of that risk for an individual, group, or society. It involves both risk determination and risk evaluation.

Risk Averse

(ROWE)  
Displaying a propensity against taking risks.

Risk Aversion

(ROWE)  
The act of reducing risk.

Risk Aversive

(ROWE)  
Acting in a manner to reduce risk.

Risk Baseline

(CURRENT)  
The risk probability density function and the associated magnitude of consequence for the potential negative outcomes.

Risk Consequence

(ROWE)

The impact to a risk agent of exposure to a risky event.

Risk Conversion Factor

(ROWE)

A numerical weight allowing one type of risk to be compared to another type.

Risk Determination

(ROWE)

The process of identifying and estimating the magnitude of risk.

Risk Estimation

(ROWE)

The process of quantification of the probabilities and consequence values for an identified risk.

Risk Evaluation

(ROWE)

The complex process of developing acceptable levels of risk to individuals or society.

Risk Evaluator

(ROWE)

A person, group, or institution that seeks to interpret a valuing agent's risk for a particular purpose.

Risk Identification

(ROWE)

The observation and recognition of new risk parameters, or new relationships among existing risk parameters, or perception of a change in the magnitude of existing risk parameters.

Risk Management

(AFR205X)

The total process of identifying, controlling, and minimizing uncertain events. The process of obtaining and maintaining DAA approval is a major element of the risk management program. The process facilitates the management of automatic data processing (ADP) security risks by each level of ADP management throughout the ADP life cycle. The approval process consists of three elements: risk analysis, certification, and approval.

Risk Profile Baseline

(CURRENT)

The measure of information and/or requirements which serve as the zero reference against which negative (and positive) outcomes can be determined.

Risk Proportionality Derating Factor

(ROWE)

Quantifying the degree to which risks become less acceptable as indirect benefits to the risk agent declines.

Risk Proportionality Factor

(ROWE)

That portion of the total societal risk that society will accept for a new technology.

Risk Reduction

(ROWE)

The action of lowering the probability of occurrence and/or the value of a risk consequence, thereby reducing the magnitude of the risk.

Risk Reference

(ROWE)

Some reference, absolute or relative, against which the acceptability of a similar risk may be measured or related; implies some overall value of risk to society.

Risk Referent

(ROWE)

A specific level of risk deemed acceptable by society or a risk evaluator for a specific risk; it is derived from a risk reference.

Risky Shift

(ROWE)

The tendency of certain groups to become more extreme or take riskier positions in their judgments than they would acting as individuals.

Security

(AFOTECPS)

The means to prevent unauthorized access to and compromise of classified information within Facilities.



Security Incident

(AFR205X)

Any act or circumstance that involves classified information in which there is a deviation from the requirements of governing security regulations (for example, compromise, inadvertent disclosures, need-to-know violations, and administrative deviations).

Security Measures

(AFR205X)

Elements of software, hardware, or procedures which are included in the system for the satisfaction of security specifications.

Security Requirements

(AFR205X)

The types and levels of protection necessary for equipment, data, information, applications, and facilities.

Security Specifications

(AFR205X)

Detailed descriptions of the measures required for protection in accordance with security requirements. Applicable requirements from Air Force policies, regulations, and standards are addressed.

Sensitive Automatic Data Processing Resources

(AFR205X)

Those resources that must be protected because their compromise, alteration, destruction or loss will adversely affect the security of classified proprietary, personal, or other data/information which has been restricted by competent authority from general disclosure. This includes information used to manage sensitive resources (for example, high dollar value, munitions, etc.).

Sensitivity Analysis

(ROWE)

A method used to examine the operation of a system by measuring the deviation of its nominal behavior due to perturbations in the performance of its components from their nominal values.

Sensitivity Assessment

(AFR205X)

A detailed study of the sensitivity or criticality of the automatic data processing (ADP) entity. It consists of gathering information about the physical, administrative, and operational environments in which the ADP entity must exist; and provides for preliminary development of security requirements based upon known vulnerabilities and possible threats.

### Significant Modification

(AFR205X)

- Any modification to the ADPF, ADPS, or ADS which impacts the operation of the system or affects the security measures of the system. "Significance" is a subjective term and depends on the environment in which the system operates.

### Simulation

(AFR800-14)

The representation of physical systems or phenomena by computers, models or other equipment.

### Software

(AFOTECP1)

A set of computer programs, procedures, and associated documentation concerned with the operation of a data processing system.

(CURRENT)

The programs which execute in a computer. The data input, output, and controls upon which program execution depends and the documentation which describes, in a textual medium, development and maintenance of the programs.

### Software Bench

(AFOTECP5)

An item used to test software units and integrated software by using a simulation CPU to represent the target computer and exercising the operational software on either the actual target processor or an Instruction Level Emulator.

### Software Error

(CURRENT)

The human decision (inadvertent or by design) which results in the inclusion of a fault in a software product.

### Software Fault

(CURRENT)

The presence or absence of that part of a software product which can result in software failure.

### Software Maintainability

(AFOTECP1)

The ease with which software can be changed in order to:

- (1) Correct errors.
- (2) Add or modify system capabilities through software changes.

- (3) Delete features from programs.
- (4) Modify software to be compatible with hardware changes.

(CURRENT)

A quality of software which reflects the effort required to perform software maintenance actions.

#### Software Maintenance

(CURRENT)

Those actions required for:

- (1) Correction. Removal, correction of software faults
- (2) Enhancement. Addition/deletion of features from the software
- (3) Conversion. Modification of the software because of environment (data hardware) changes.

#### Software Maintenance Environment

(CURRENT)

An integration of personnel support systems and physical facilities for the purpose of maintaining software products.

#### Software Maintenance Measures

(CURRENT)

Measures of software maintainability and environment capabilities to support software maintenance activity.

#### Software Management

(CURRENT)

The policy, methodology, procedures, and guidelines applied in a software environment to the software development/maintenance activities. Also, those personnel with software management responsibilities.

#### Software Portability

(CURRENT)

A quality of software which reflects the effort required to transfer the software from one environment (hardware and system software) to another.

#### Software Problem Report (SPR)

(AFR300-15)

A form used to report a suspected or existing discrepancy or deficiency in an existing computer program, its operational documentation, or interfacing hardware.

## Software Reliability

(CURRENT)

A quality of software which reflects the probability of failure free operation of a software component or system in a specified environment for a specified time.

## Software Support Facility (SSF)

(AFOTECPS)

The facility which houses and provides services for the support systems and personnel required to maintain the software for a specific ECS.

## Software Support Facility Manager

(AFOTECPS)

The person in charge of a software support facility.

## Software Supportability

(CURRENT)

A measure of the adequacy of personnel, resources, and procedures to facilitate:

- (1) Modifying and installing software
- (2) Establishing an operational software baseline
- (3) Meeting user requirements.

## Software Supportability Evaluation Metrics

(CURRENT)

The closed-form questionnaire scores for each characteristic and cumulated level in a software supportability evaluation.

## Software Supportability Magnitude of Risk Consequence

(CURRENT)

The level of impact to a software user or supporter as a result of the risk level of a software supportability negative outcome.

## Software Supportability Negative Outcome

(CURRENT)

The final result of a maintenance request as represented by the pair (time to complete request, number of requests per unit time), in which the Baseline SS Profile is not met.

## Software Supportability Risk Agent Acceptance Level

(CURRENT)

The software supportability risk level which is acceptable to a risk agent.

Software Supportability Risk Level

(CURRENT)

The potential for realization of a software supportability negative outcome.

Specification

(AFR300-15)

A document that describes the requirements for the development or acquisition of ADPE and/or software.

Standards

(AFOTECP3)

Procedures, rules, and conventions used for prescribing disciplined program design and implementation.

States of Nature

(ROWE)

A concept from decision theory. In decision making under uncertainty, the outcomes (numerical results) associated with each available alternative are considered to be predictable as a set of  $n$  discrete values depending on conditions beyond the decision maker's control and for which he has no useful estimates of the respective probabilities. The  $n$  sets of conditions under which each one of the outcomes is expected are termed "states of nature."

Stochastic System

(ROWE)

A system whose behavior cannot be exactly predicted.

Structured Value (structured value analysis)

(ROWE)

The resultant value of a particular value set evaluated for a particular data set. This value lies between zero and unity and allows many data sets to be ranked numerically in relation to one another.

Structured Value Analysis

(ROWE)

A multistage procedure for assessing the value of an action, project alternative, and so on, incorporating individual techniques at each stage for computing from quantitative measures of individual components a single figure expressing the overall value. A multistage procedure for assessing the value of an action, project, alternative, and so on, by structuring the complete entity into component elements, to each of which a numeric measure of value (positive or

negative) can be assigned. These are then converted to a common utility scale. Each component is assigned a weight expressing its relative significance in determining overall value of the entity. A single figure of worth or value is then computed from measures and weights of all individual components. The procedure permits considerable flexibility in choice of techniques used to perform each necessary optimal step.

#### Subjective Probabilities

(ROWE)

The assignment of subjective weights to possible outcomes of an uncertain event where weights assigned satisfy axioms of probability theory.

#### Support Personnel

(AFOTTECP5)

A general term for military or DoD civilian personnel whose skills are necessary for the software support facility to function but who do not directly support ECS software maintenance.

#### Support System

(AFOTTECP5)

Any automated system used to change, test, or manage the configuration of ECS software and associated documentation. Includes but is not limited to Host Processor, Software Bench, Laboratory-Integrated Test Facility, Operational-Integrated Test Facility, and Configuration Management System.

#### Support System Facility

(AFOTTECP5)

The facility resources that must be available for the software support resources to accomplish a specific task(s) (see General Facility).

#### Surrogate or Proxy Measures

(ROWE)

The use of a related quantity as a proxy for an unknown or difficult-to-measure value. The relationship may be established by armchair analysis, correlation techniques, scientific studies, or other means.

#### System

(ROWE)

a) A complex entity formed of many, often diverse, parts subject to a common plan or serving a common purpose.

b) A composite of equipment, skills, and techniques capable of performing and/or supporting an operation.

#### System Design Review (SDR)

(AFR300-15)

A formal review of the system design approach for an ADS.

#### System High Security Mode

(AFR205X)

A mode of operation in which all personnel having access to the automatic data processing system (ADPS) have a security clearance, but not a need-to-know, for all material then contained in the system. An ADPS is operating in the system high security mode when the central computer facility and all of its connected peripheral devices and remote terminals are protected according to the requirement for the highest classification of material contained in the system. In this mode, the ADPS design and operation must accordingly provide for some internal control of concurrently available classified material in the system on the basis of need-to-know.

#### System Requirements Review (SRR)

(AFR300-15)

A formal review of the requirements for an ADS.

#### System Software

(AFOTECP5)

All of the software that is part of the software support facility computer system. It is never or seldom accessed directly by software support facility personnel; it controls the processing of application software. It includes the Operating System, Source Code Editor, Language Translator, Link Editor/Loader, Librarian/File Manager, Data Base Manager, and Automated Software Development Tool.

#### System Validation Review (SVR)

(AFR300-15)

A formal review of the results of the Test Phase to ensure that the ADS satisfies the requirements of the SS and FD.

#### Taxonomy

(ROWE)

The identification and definition of properties of elements of the universe; a disaggregation, as contrasted with systematics (which is an aggregation) and as contrasted with morphology (which encompasses both taxonomy and systematics).

## Test Analysis Report (RT)

(AFR300-15)

A document containing the results and analyses of tests executed during the Test Phase.

## Threat

(AFR205X)

The means through which the ability or intent of a threat agent to adversely affect an automatic data processing system, facility, or operation may be manifested. Threats may be categorized and classified as follows:

<u>Categories</u>	<u>Classes</u>	
Human	Intentional	Unintentional
Environmental	Natural	Man-Made

## Threat Agent

(AFR205X)

Those methods and things (for example, fire, natural disaster, etc.) which may be used to exploit a vulnerability in an ADP system, facility, or operation.

## Threshold

(ROWE)

A discontinuous change of state of a parameter as its measure increases. One condition exists below the discontinuity, and a different one above it.

## Time to Complete Maintenance Request (TC)

(CURRENT)

The calendar time from receipt of the maintenance request by the support control group until the request has been denied or the maintenance actions required by request have been accepted as part of an operational system software configured release. (This does not mean the configuration is released or distributed, and this time does not include this additional delay if any.)

## Transfer

(AFR800-14)

That point in time when the designated Supporting Command accepts program management responsibilities from the Implementing Command. This includes logistic support and related engineering and procurement responsibilities. (AFR 800-4)



## Turnover

(AFR800-14)

That point in time when the operating command formally accepts responsibility from the Implementing Command for the operation and maintenance of the system, equipment, or computer program acquired.

(AFR 800-19)

## Uncertainty

(ROWE)

The absence of information; that which is unknown.

## User

(AFR205X)

Any persons (or organizations) having access to an automatic data processing system via communication through a remote device or who is allowed to submit input to the system through other media (for example, tape or card decks). (Does not include those persons or organizations defined as customers.)

## Valuation

(ROWE)

The act of mapping an ordinal scale onto an interval scale (i.e., assigning a numerical measure to each ranked item based on its relative distance from the end points of the interval scale... assigning an interval scale value to a risk consequence.

## Value

(ROWE)

A quality quantified on a scale expressing the satisfaction of man's intrinsic wants and desires.

## Value Function (structured value analysis)

(ROWE)

A function relating points on the parameter measurement scale to the value scale for a particular parameter. These functions may result from explicit information or may be arrived at through value judgment.

## Value Set (structured value analysis)

(ROWE)

A specific set of model parameters made up of terms and factors, expressed in particular measurement scales, value functions, and weights.

Valuing

(ROWE)

The act of assigning a value to a risk consequence.

Valuing Agent

(ROWE)

A person or group of persons who evaluates directly the consequence of a risk to which he is subjected. A risk agent.

Verification/Validation (of computer programs)

(AFR800-14)

The process of determining that the computer program was developed in accordance with the stated specification and satisfactorily performs, in the mission environment, the function(s) for which it was designed.

Vulnerability

(AFR205X)

A weakness in automatic data processing security procedures, administrative controls, internal controls, etc., that could be exploited by a threat to gain unauthorized access to sensitive information (both classified and unclassified) or disrupt critical processing.

Waiver

(AFR300-15)

A written authorization to accept a configuration item or other designated item that has been found to depart from specified requirements, but nevertheless is considered suitable for use as is or after rework by an approved method.

Weight (structured value analysis)

(ROWE)

The relative importance of terms in a model expressed as a decimal fraction; weights for a set of terms add to unity.

Weighting Factor

(ROWE)

A coefficient used to adjust variable accuracy to a subjective evaluation; these factors are usually determined through surveys, Delphi sessions, or other formats of expressing social priorities.

**Appendix C**  
**Trip/Conference Reports**

APPENDIX C  
TRIP/CONFERENCE/CONTACT REPORTS

APPENDIX C  
TRIP REPORTS

The trip reports required as part of any visit or conference are included in this appendix as delivered to AFOTEC. In addition, telephone or other contact summaries not required as a deliverable are also included in this appendix.

<u>Trip Report</u>	<u>Page</u>
STARS Measurement DIDs Workshop	C-3

<u>Contact Summary Report</u>	<u>Page</u>
Dr. Victor Basili	C-9
Mr. John Musa	C-11
Dr. William Riddle	C-13
Dr. Barry Boehm	C-15
Dr. Allen Stubberud	C-17
Dr. Nancy Leveson	C-19
Mr. Jim McCall	C-21
Mr. Gerald Fisher	C-23
Mr. William Rowe	C-25
Mr. Mark van den Broek	C-27
Dr. Dixie Baker	C-28
Dr. Richard DeMillo	C-30

## CONTACT SUMMARY

SS 304

SUBJECT: STARS Measurement DIDS Workshop

DATE OF CONTACT: August 14-15, 1984

PLACE CONTACTED: Rome Air Development Center (RADC)  
Rome, NY

PRINCIPAL CONTACTS: Workshop Membership

PERSON(S) MAKING CONTACT: Dr. David E. Peercy, BDM/A

## PURPOSE:

Attend Software Technology for Adaptable, Reliable Systems (STARS) workshop to review draft Data Item Descriptions (DIDS) for software life cycle measurement. Chair the session on development and operational environment DIDS. Determine applicability of proposed environment characteristics to risk assessment of software supportability.

## BACKGROUND:

The STARS program is a DoD initiative to provide a technology push to improve software, its acquisition, and the environment in which it is developed, maintained, and operated. Cornerstones of this effort include development of the Ada DoD Common Language, and its environment APSE; establishment of the Joint Services Software Engineering Environment (JSSEE) program; and the encompassing DoD-STD-SDS proposed standard for software development. The STARS program has seven major areas of interest: measurement, project management, human resources, systems, application-specific, human engineering, and support systems. The Air Force is the lead agency for the software measurement task area.

The software measurement task area includes activities in five general categories: baseline development, automated data collection, measurement analysis, measurement improvement, and measurement technology transfer.

The proposed Joint Logistics Commanders (JLC) software development standards (DoD-STD-SDS, MIL-STD-1521B, etc.) have been adopted as the

basic standard for the software measurement terminology related to life cycle phases, hierarchical system components, classifications, and system documentation.

The purpose of the STARS Measurement Project is to ensure consistency, completeness, and availability of measurement data needed to support software research under the STARS Program. DIDS will be developed for the collection of data on DoD software acquisition and support programs. DIDS will be designed to collect five major classes of data: software quality, resource, software product, development environment, and operation assessment. The purpose of the subject workshop was a review, by invited participants, of comments on the initial drafts of the proposed DIDS. The initial draft consisted of 19 separate DIDS which were distributed to a closed review group of approximately 214 prior to the workshop. AFOTEC personnel participated as a review group. The initial drafts of the DIDS were developed by Dynamic Research Corporation in subcontract to RADC.

#### DISCUSSION:

##### 1. Overview

The purpose of the subject measurement workshop was to review comments from the initial draft of the proposed set of 19 DIDS, and provide constructive suggestions for improvements of the DIDS. After incorporation of the suggestions, the plan is to reissue the DIDS for a more formal (and lengthy) public review. A brief summary of the workshop results is presented below.

A workshop welcome and introductions were provided by the agenda speakers. Objectives of the measurement thrust, major tasks, and workshop procedures/aims were explained. Of the 214 packets of DIDS mailed, 38 had been returned with comments at the time the conference convened. The workshop was divided into six sessions with each session independently reviewing a logically related set of DIDS. Dr. D. Peercy was chairman of the Session E on development environment (SWOESUM DID) and operational environment (SWOESUM DID). In addition, another group called the "issue group" was formed to consider overall encompassing issues concerning the DIDS.

## 2. General Comments

The issues group presented the following recommendations and comments:

- (1) Limit number of DIDS to six for R&D with one operational DID.
- (2) Scope of DID and funding level from DoD are not compatible, but should be.
- (3) Implementation strategy is absent. There needs to be:
  - (a) Mechanism for automation
  - (b) Funding from DoD
  - (c) Mission critical tailoring.
- (4) Industry and Professional participation needs to be expanded.
- (5) Need mechanism to systematically capture and assess measurement experience and lessons learned.
- (6) The measures, models, and data need to be identified and prioritized. Include list and matrix relationship of models and permit inclusion of current and future models for which DIDS data is supposed to support.
- (7) Timing and frequency should not be embedded in DIDS. Contractual problem? Include in guidebook. (Note: The group consensus was not clear on this issue.)
- (8) Greater focus needs to be placed upon the measurement of software reuse.
- (9) Classified software/data need to be addressed.

## 3. Specific Session Comments

Nearly all sessions were overwhelmed by the amount of information in the DIDS, the lack of organization of the DIDS, the redundancy of information across the DIDS and the presence of unusable information in the DIDS. It was generally difficult to accomplish the specific task--that is, review specific DID comments--because of their overall deficiencies. The number of DIDS should be reduced. This reduction would create a complete reorganization, hopefully based upon life cycle phase, with a more generic approach within the details of each area. Trying to assess the worth of a current comment on a specific detail was felt to be a



waste of time until the reorganization takes place. Thus, most assessment of the review comments was directed toward categorizing the comments and assessing the resulting categories.

Many of the issues for the operational and development environment group (SWOESUM and SWDESUM DIDS) and the maintenance environment group (SWMESUM and SCED) were very similar. There seemed to be very little evidence of information in the DIDS relating to the "integrated environment" concepts being supported through the Ada/APSE and STARS efforts. The environment workshop sessions felt the development, operational and maintenance environments DIDS should be reworked as a single DID with an integrated life cycle environment approach. In fact, these sessions felt there might well be only two operational DIDS, a software life cycle (SWLC) DID and a software evaluation report (SWER) DID.

The "environment" of the DID should be generically treated with sections for environment category (and other identification data), management (procedures, standards, conventions, methodology), personnel (classification, experience), configuration (systems, facilities), and resource measurement (availability, capability, utilization, and shared/dedicated attributes for each resource as measured on a high/medium/low scale). An environment part would be completed for each category (host, software bench, integrated laboratory, operational) as appropriate and useful for each reporting period and as "major" environment changes occur.

It was recommended that a methodology/technique/tool matrix (e.g., similar to NBS taxonomy) be included with each element appropriately labeled. Then, in the configuration identification section, one would list the configuration elements by label with a more precise "name identification" (e.g., VAX/VMS FORTRAN Version 2.0) as optional information.

The maintenance group felt the information required in the SCED was overwhelming (consider the reporting required for 67 CSCIs undergoing over 1,300 changes during a two-week integration period--an actual example from one of the workshop participants). The general process of software error reporting (e.g., as part of configuration management) was not adequately addressed.

The software evaluation report group (SER DIDS) had over 400 comments to absorb, and concentrated primarily on the aspect of providing more informative relationships and explanations in a guidebook which would accompany the SER DID(s). An outline and content summary of the guidebook was discussed briefly. It was recommended that evaluators be independent of developers. Validation of SER data and the use of a prototype process were two areas not adequately addressed. Tailoring of the SER DIDS was not adequately addressed.

The resource group (RESUM, REDET) concentrated on the consistency aspects of size data across CSCIs/CSCs/modules/units and the impact of reporting data below the CSCI level. It was recommended that the RESUM and REDET DIDS be consolidated and that there was redundancy with other DIDS.

The software characteristics group (SWCHRSUM, SWCHRDET) was concerned about the volume of data required, 357 items on characteristics alone. This group recommended completion of the glossary to include more information definitions for system, subsystem, CSC, function, application type, and so forth. The frequency and timing information for these DIDS emphasized development. There should also be some emphasis on the O&M phase. Despite the volume of data, several critical areas (e.g., hardware-to-hardware interfaces) have been ignored.

The software test group (SITSUM, SITDET) recommended a reorganization of the test DIDS more closely following DoD-STD-SDS and the recommendations of the Software Test and Evaluation Project (STEP). In particular, the measurement interval should be fixed, not based upon milestones. Section A should address testing strategies, and methodologies and percentage of test areas generated using each. Section B should address specific tests and test cases. As with most sessions, global issues surrounding the DIDS seemed to dominate everyone's concerns.

#### CONCLUSIONS:

It was apparent from the workshop participant comments (more than from individual reviewer comments) that the current form and content of

the DIDS is unsatisfactory. The DIDS should be consolidated. The information requested should be more systematically and generically structured to clearly cover all acquisition phases. More support information such as purpose, scope, applicability, and so forth should be developed as a foundation for the measurement DIDS development. Much rework of these DIDS is required if the next public review of the DIDS is to elicit positive response.

The current DIDS characteristics for the support environment and software products are not oriented toward addressing the risk assessment issues identification by AFOTEC. However, the possibility of future DID development incorporating such information may now be more likely due to the efforts of this workshop. This rework of the measurement DIDS should be carefully followed by AFOTEC to assure such information is valuable to AFOTEC.

ACTION ITEMS:

- (1) Obtain the latest version of the JLC DoD-STD-SDS and related documents MIL-STD-1521B, MIL-STD-490 (Notice 3), MIL-STD-483 (Notice 3), and MIL-STD-SDS Data Item Description, all dated December 5, 1983.
- (2) Maintain contact with STARS Measurement Project to know disposition of measurement workshop suggestions and to obtain a revised draft of DIDS.

## CONTACT SUMMARY

SS 304

SUBJECT: AFOTEC Software Supportability Risk Assessment

DATE OF CONTACT: 5/15/84

PLACE CONTACTED: University of Maryland  
College Park, MD 20801  
(301) 454-4254 X2002

PRINCIPAL CONTACTS: Dr. Victor Basili

PERSON(S) MAKING CONTACT: Dr. David E. Peercy, BDM/A

## PURPOSE:

Discuss current research tasks in software risk assessment, in particular, any activity being conducted through the Software Engineering Laboratory (SEL) "contract" with NASA Langley. Obtain contacts and document titles related to software risk assessment.

## BACKGROUND:

Dr. Basili is very knowledgeable in the area of software metrics and methodology. He has worked closely with NASA Langley pioneering a Software Engineering Laboratory (SEL) to study in a practical research environment various productivity effects of programming and support environments. His numerous publications in the software engineering area reflect his concern with the practical application and implementation of software methodologies and engineering principles.

## DISCUSSION:

During this telephone contact with Dr. Basili, the primary outcome was the lack of research in applying technical risk assessment and methodology (e.g., sophisticated statistical tests) to the whole field of software. There are several efforts under way to quantify various software characteristics for aspects which might affect software supportability (much as AFOTEC is already conducting), but formalized risk

analysis/assessment is for the most applied only to an individual program, by request, adhoc manner. Dr. Basili postulated the reason was the infancy of the science of software engineering. And, what risk analysis was being done seemed to be tailored toward the software development cycle with the assumption that O&M would take care of itself if the development was done properly. Dr. Basili's recent paper "Monitoring Software Development through Dynamic Variables" in COMPSAC 1983 proceedings may provide some insight into risk drivers.

ACTION ITEMS:

- (1) Obtain copy of referenced paper.

## CONTACT SUMMARY

SS 304

SUBJECT: AFOTEC Software Supportability Risk Assessment

DATE OF CONTACT: 5/16/84

PLACE CONTACTED: Bell Laboratory

Whippany, NJ

(201) 386-2398

PRINCIPAL CONTACTS: Mr. John Musa

PERSON(S) MAKING CONTACT: Dr. David E. Peercy, BDM/A

## PURPOSE:

Discuss current research tasks in software risk assessment, in particular, application of software reliability assessment and its relationship to risk assessment. Obtain contacts and document titles related to software risk assessment.

## BACKGROUND:

John Musa is very knowledgeable in the field of software reliability. He has published many articles in this area and has performed internal R&D work in this area for Bell Laboratory. He has a software reliability model which has been installed by BDM on the AFOTEC IBM 4341.

## DISCUSSION:

During this telephone conversation with Mr. Musa, the primary outcome was that he was not involved with software risk assessment and did not know of any specific projects or personnel in this area. A discussion on the application of his reliability model (and other reliability models) to risk assessment led to the belief that the model(s) could be used as part of a risk assessment. For example, the actual reliability growth curves could be assessed against the extrapolated curves over time to determine the difference between perfect and predicted reliability (assuming maintenance effort continues). Predicted

absolute numbers of faults over unit time could be compared against support and user reliability constraints. The risk associated with these differences and the associated confidence band around the reliability growth curves would be analyzed as part of the total software supportability risk assessment. Although these concepts have not been implemented as far as Mr. Musa knows, it appears reasonably feasible to do so.

ACTION ITEMS:

- (1) Review Musa's reliability model and documentation.

## CONTACT SUMMARY

SS 304

SUBJECT: AFOTEC Software Supportability Risk Assessment

DATE OF CONTACT: 5/18/84

PLACE CONTACTED: Software Design and Analysis, Inc.

Boulder, CO 80303

(303) 499-4733

PRINCIPAL CONTACTS: Dr. William Riddle

PERSON(S) MAKING CONTACT: Dr. David E. Peercy, BDM/A

## PURPOSE:

Discuss current research tasks in software risk assessment, in particular, Dr. Riddle's participation on the USAF Scientific Advisory Board (SAB), an Ad Hoc Committee on "The High Cost and Risk of Mission-Critical Software", DEC 83. Obtain contacts and document titles related to software risk assessment.

## BACKGROUND:

Dr. Riddle has been involved for several years in software methodology research and is an acknowledged expert and consultant on software environments. Dr. Riddle was a member of the referenced committee which produced the recent Air Force study on software risk assessment. Dr. Riddle is a private consultant through his corporation Software Design and Analysis, Inc.

## DISCUSSION:

During this telephone contact with Dr. Riddle, several ideas concerning software risk assessment were discussed, but there were no current research tasks known to him. Dr. Riddle explained the USAF SAB committee report as a compendium of information derived from a series of briefing-meetings. Dr. Barry Boehm was the key committee member for concepts related to software risk management, including the Appendix I, a



proposed addition to AFR 800-14, Vol II, "Chapter II, Risk Management". Dr. Riddle also suggested looking at a recent paper by Dr. Boehm on "Comparing Phased Development Methodology and Prototyping Development Methodology" for some issues in software development risk assessment. This article is in a recent issue of COMPUTER magazine.

ACTION ITEMS:

- (1) Review Dr. Boehm's paper.

## CONTACT SUMMARY

SS 304

SUBJECT: AFOTEC Software Supportability Risk Assessment

DATE OF CONTACT: 5/18/84

PLACE CONTACTED: TRW Systems Engineering  
Defense System Group  
Los Angeles, CA  
(213) 535-2184

PRINCIPAL CONTACTS: Dr. Barry Boehm

PERSON(S) MAKING CONTACT: Dr. David E. Peercy, BDM/A

## PURPOSE:

Discuss current research tasks in software risk assessment, in particular, any activity at TRW and any activity briefed to the USAF Scientific Advisory Board (SAB) an Ad Hoc Committee in "The High Cost and Risk of Mission-Critical Software", DEC83. Obtain contacts and document titles related to software risk assessment.

## BACKGROUND:

Dr. Boehm is a recognized expert on software engineering, and has been responsible for heading TRW research in software productivity and software development research. He is the author of several documents from the TRW Software Engineering Series, including perhaps the first known attempt to build and describe a taxonomy of software quality factors. Dr. Boehm is also the author of the recent book on Software Engineering Economics, which is a practical approach to costing software. Some aspects of software risk are contained in chapter 20 of this book.

## DISCUSSION:

During this telephone conversation Dr. Boehm explained his contribution to the USAF SAB report and his work at TRW. He summarized the SAB report as a top level view of risk assessment issues and primarily a "plea" to do more, with some reasonably common sense suggested actions.

Dr. Boehm's contribution was primarily Appendix I which was a suggested Risk Management chapter addition to AFR 800-14. He felt this additional "policy" was at too high level to be of much use, except possibly for general guidance. According to Dr. Boehm, there is no generic risk assessment methodology research or application at TRW. Individual programs/projects do risk assessment on an ad hoc basis and have the basic goal of helping TRW to minimize software development risk. Dr. Boehm is not aware of any specific efforts for the equivalent O&M related software support risk assessment. Dr. Boehm referenced chapter 20 of his book on Software Engineering Economics as containing some general software risk assessment issues. Dr. Boehm indicated he was receptive to a visit by BDM personnel, but it was agreed that there was not much to talk about at this time.

**ACTION ITEMS:**

- (1) Review risk assessment in chapter 20 of Boehm's book.

## CONTACT SUMMARY

SS 304

SUBJECT: AFOTEC Software Supportability Risk Assessment

DATE OF CONTACT: 5/29/84

PLACE CONTACTED: Pentagon, AFCCN

Washington, DC

(202) 697-7842

PRINCIPAL CONTACTS: Dr. Allen Stubberud

PERSON(S) MAKING CONTACT: Dr. David E. Peercy, BDM/A

## PURPOSE:

Discuss software risk assessment initiatives within the Air Force, in particular the USAF Scientific Advisory Board Report on "The High Cost and Risk of Mission-Critical Software," DEC 83.

## BACKGROUND:

Dr. Stubberud is an Air Force Chief Scientist reporting directly to the Chief of Staff of the Air Force. Dr. Stubberud was a member of the Scientific Advisory Board (SAB) which produced the referenced report. The SAB was an ad hoc Committee (advisory capacity) to the Chief of Staff and Secretary of the Air Force.

## DISCUSSION:

During this telephone conversation with Dr. Stubberud the Air Force software risk assessment initiatives were discussed. The SAB report conclusions were basically that no one is doing software risk assessment/analysis, but that someone should be. In particular, the Air Force should concentrate upon predictability and control, productivity and quality, and post deployment software support. The DoD programs, Ada and STARS, are important for improving the cost-benefit risk to Air Force system acquisition. The DoD VHSIC program also has important implications for software risk assessment.

Dr. Stubberud indicted that Dr. Boehm was the key SAB committee member regarding software risk assessment. In addition, Mark van den Broek (AFLC) of the SAB was technically competent in software risk assessment. Dr. Stubberud felt Hughes Aircraft might be a good source for some application methodologies since Paul Mauro, a member of the SAB, was from Hughes and some of the better committee briefings were by Hughes personnel. In particular, the 12 January 1983 briefing (not listed in the SAB report) was by Hughes personnel and concerned risk assessment techniques by the Hughes Aircraft Flight Dynamics Lab for NASA. This was a report on contract F33615-80-C-3614.

Dr. Stubberud also referenced the AF/SA technical note of 1981. Generally, the conclusion was software risk assessment is not being done and, if required, is based upon rather adhoc and impromptu methods.

**ACTION ITEMS:**

- (1) Contact Mark van den Broek at AFLC.
- (2) Contact ESD, e.g. Col. John Marciariak, RADC.
- (3) Obtain Hughes Aircraft report.

## CONTACT SUMMARY

SS 304

SUBJECT: AFOTEC Software Supportability Risk Assessment

DATE OF CONTACT: 5/31/84

PLACE CONTACTED: University of California at Irvine  
Irvine, CA

(714) 856-5517 (office)/7403 (department)

PRINCIPAL CONTACTS: Dr. Nancy Leveson

PERSON(S) MAKING CONTACT: Dr. David E. Peercy, BDM/A

## PURPOSE:

Discuss software risk assessment research activity, in particular as it relates to software safety. Obtain contacts and document titles related to software risk assessment.

## BACKGROUND:

Dr. Leveson is best known for her research contributions to the field of software safety, a factor in software supportability risk assessment. Several of Dr. Leveson's research activities have at least indirect relevance to software risk assessment. BDM has talked with Dr. Leveson on some software system safety related tasks.

## DISCUSSION:

During this telephone contact Dr. Leveson indicated some of her work in software safety and other researcher's work, such as Bev Littlewood (reliability), were indirectly applicable to software risk assessment. She did not know of any specific software risk assessment efforts currently in progress. Her work with NASA Langley is a research study an automated fault tolerant testing. Some early results indicate that it is a bad assumption to ever assume software faults are zero, even after millions of tests. In some instances short programs (e.g., 4000 source

lines) have had faults surface after one million test cases. More information should be available by mid summer. Dr. Leveson will also send some relevant papers by Bev Littlewood at the end of June.

Two documents were identified relevant to software safety: MIL-STD-882B, a one month old system safety program requirement document, and an Air Force handbook to support 882B by Bruce Bennett for the Norton AFB, CA.

Dr. Leveson was receptive to a possible visit by BDM to further discuss possible software risk assessment research ideas. There are several other professors at Irvine (Peter Freeman, Dick Taylor, Tim Standish) who have an interest in related software assessment methodologies, including Ada support environments.

ACTION ITEMS:

- (1) Obtain MIL-STD-882B
- (2) Obtain AF handbook to support MIL-STD-882B
- (3) Maintain Contact with Dr. Leveson for possible visit during analysis subtask
- (4) Review Bev Littlewood's research papers when sent in late June.

CONTACT SUMMARY  
SS 304

SUBJECT: AFOTEC Software-Supportability Risk Assessment

DATE OF CONTACT: 6/1/84

PLACE CONTACTED: Science Applications, Inc. (SAI)

La Jolla, CA

(619) 454-3811

PRINCIPAL CONTACTS: Mr. Jim McCall

PERSON(S) MAKING CONTACT: Dr. David E. Peercy, BDM/A

PURPOSE:

Discuss current research in software risk assessment, in particular, the possible use of the software metrics developed by Mr. McCall for RADC and applied by SAI for IV&V. Obtain contacts and document titles related to software risk assessment.

BACKGROUND:

Jim McCall is well known for his work in developing the software quality metrics and general framework now being expanded and refined by RADC. In addition, Mr. McCall has been a key investigator for research work as a software maintenance management guidebook for the National Bureau of Standards. Many of these metrics, tools, and guidelines are now part of the IV&V work being performed by SAI.

DISCUSSION:

During this telephone contact, Mr. McCall discussed the current software quality work and its application to a generic tool set, ISMS. ISMS helps management trace the software product metric quality profile across the complete software life cycle. This product is a proprietary SAI tool set, but is being installed as a supported product in government maintenance facilities. Mr. McCall will send some information on ISMS.



Mr. McCall also referenced the current reliability study by RADC which is considering the development and testing reliability profile as a predictive tool for operational effects. Joe Cavano at RADC is the primary contact for this study, and also for the earlier and still expanding work on software quality metrics.

Mr. McCall was not aware of any work being done to apply the software quality metrics as part of a generic software risk assessment methodology. Some work was being done to validate the metrics. Mr. McCall felt the software risk assessment study was a very worthwhile effort and would be interested in the study results.

ACTION ITEMS:

- (1) Review latest RADC reliability and software quality metrics research.
- (2) Review the ISMS tool set information when it arrives.

## CONTACT SUMMARY

SS 304

SUBJECT: AFOTEC Software Supportability Risk Assessment

DATE OF CONTACT: 6/15/84

PLACE CONTACTED: AF/SASF

Washington, D.C.

(202) 697-9890

PRINCIPAL CONTACTS: Mr. Gerald J. Fisher

PERSON(S) MAKING CONTACT: Dr. David E. Peercy, BDM/A

## PURPOSE:

Discuss current AF/SASF software risk assessment research, in particular the ongoing work implied by the AF/SA Technical Note; "An Approach to Risk Analysis: A Process View", June 1981, which Gerald Fisher co-authored.

## BACKGROUND:

Mr. Gerald Fisher is co-author of the referenced AF/SA technical note. He has been involved with the AF Strategic Force studies and analyses for several years.

## DISCUSSION:

During this telephone contact Mr. Fisher indicated the AF/SA technical note was intended to be a basic concept paper for AF/SASF to be followed by a more detailed series of studies and analyses of risk assessment methodologies, techniques and tools leading to more complete AF policy, and guidelines on software risk analysis. However, the concept paper was as far as the effort progressed. As far as Mr. Fisher knows, there is no current research activity within AF/SASF on software risk assessment. He offered to check the tactical force activity and report any research by Tuesday, 19 June 1984. One technique which was mentioned by Mr. Fisher as a possible (but complex) risk assessment

approach was the "Palm" (this may be the PANRISK which is an older version of IST/RAMP). He felt William Rowe, who heads the Institute for Risk Analysis at American University was a good source for current risk analysis activity.

ACTION ITEMS:

- (1) Contact William Rowe at American University
- (2) Follow up any research activity by the AF/SATF.

## CONTACT SUMMARY

SS 304

SUBJECT: AFOTEC Software Supportability Risk Assessment

DATE OF CONTACT: 6/19/84

PLACE CONTACTED: Risk Limited Corporation

Washington, D.C.

(301) 340-7990

PRINCIPAL CONTACTS: Dr. William Rowe

PERSON(S) MAKING CONTACT: Dr. David E. Peercy, BDM/A

## PURPOSE:

Discuss current research in software risk assessment, in particular, the activity of Dr. Rowe and his various risk analysis business enterprises. Obtain contacts and document titles related to software risk assessment.

## BACKGROUND:

Dr. Rowe is the author of the book An Anatomy of Risk, John Wiley & Sons, 1977. He has been an official in a federal regulatory agency, and has been involved for over 15 years with major programs for assessing acceptable levels of risk. His work goes across several technical areas including chemicals, nuclear waste, high radiation, terrorism, and computer security.

## DISCUSSION:

The telephone contact with Dr. Rowe was very informative and resulted in several possible follow-on tasks. Dr. Rowe is a professor at American University in charge of the Institute for Risk Analysis. It is University policy that its programs not be involved in classified work, so separate business enterprises were formed by Dr. Rowe to support classified work as well as other non-academic business ventures. The Risk Analysis Corp. was formed to support private industry work and the

Pure Consultants Corp. was formed to support government work. There are approximately 35 personnel in these organizations.

Currently Dr. Rowe has several research tasks in risk assessment. Unfortunately, most of the work is proprietary and could not be discussed. Dr. Rowe did indicate several areas where he has specific interest and activity: criminal justice, chemical toxics, high level radiation, nuclear waste disposal, and computer security.

Apparently, Dr. Rowe has a reasonably generic approach to risk assessment which can be applied across a broad range of functional areas. In particular, this approach involves the integration of procedures, functional area factors, relevant functional area technology constraints, and process controls into an automated program for risk assessment. Currently this approach is operational for criminal justice risk assessment, including all procedures, controls and a computer program to support decision making, statistical analysis, and bookkeeping. His corporation is currently working on a similar program for computer security risk assessment which is supposed to work for both civilian and military applications. They are 80 percent complete with the procedures and controls, and about 40 percent complete with the computer program support. The approach uses historical/empirical/requirement/heuristic data for inputs and does not rely on relative weighting. It matches the target vulnerabilities vs. threat motivation bridged by the technological feasibility of the threat to cause a risk event. Two types of risk events are considered: accidental or random; and purposeful or non-random.

Dr. Rowe will send brochures on his current risk assessment methodology. If the brochures appear interesting, it would be worthwhile to see if a visit with Dr. Rowe could be arranged.

**ACTION ITEMS:**

- (1) Review brochures when they arrive.
- (2) Pursue the possibility of a follow up visit.

## CONTACT SUMMARY

SS 304

SUBJECT: AFOTEC Software Supportability Risk Assessment

DATE OF CONTACT: 6/19/84

PLACE CONTACTED: Ford Aerospace Corp.

Sacramento, CA

(916) 929-0185

PRINCIPAL CONTACTS: Mark van den Broek

PERSON(S) MAKING CONTACT: Dr. David E. Peercy, BDM/A

## PURPOSE:

Discuss current software risk assessment research, in particular the participation of Mr. van den Broek on the Air Force Scientific Advisory Board ad hoc Committee to study the High Cost and Risk of Mission-Critical Software.

## BACKGROUND:

Mr. van den Broek was division chief of the AFLC LOC/CFE at Wright Patterson AFB. In this capacity he was an Air Force representative on the referenced Scientific Advisory Board (SAB). Currently, Mr. van den Broek is with Ford Aerospace in Sacramento, CA. where he is involved with system engineering and some risk analysis.

## DISCUSSION:

Mr. van den Broek indicated Paul Vican was now the division chief of AFLC LOC/CFE and should be able to report on the risk analysis activity of AFLC. Mark gave a good exposé of the SAB organization and the activities of the one of which he was a member. This SAB had briefings from contractors for a day or two each month for about six months followed by a two week session in Monterey to complete detailed review, analysis and writing of the committee report. Mark was not aware of any risk analysis models of software supportability.

## ACTION ITEMS:

- (1) Contact Paul Vican at AFLC LOC/CFE (513) 257-6751

## CONTACT SUMMARY

SS 304

SUBJECT: AFOTEC Software Supportability Risk Assessment

DATE OF CONTACT: 7/10/84

PLACE CONTACTED: Aerospace Corporation

El Segundo, CA

(213) 648-5834

PRINCIPAL CONTACTS: Dr. Dixie Baker

PERSON(S) MAKING CONTACT: Dr. David E. Peercy, BDM/A

## PURPOSE:

Discuss risk analysis as applied to the Consolidated Space Operations Center (CSOC) project and in particular Dr. Baker's paper on CSOC software risk analysis presented at a recent NSIA conference on Ada.

## BACKGROUND:

Dr. Baker is manager of CSOC segment for Aerospace Corporation. She has responsibility for risk-analysis issues (among others) concerning facility management, security control (physical), the technical data resource center (administrative ADP center), and system security. Dr. Baker presented a paper at a recent NSIA Ada conference on software and risk analysis.

## DISCUSSION:

The telephone conversation with Dr. Baker was very interesting from at least three views: risk analysis, computer security, and Ada. Dr. Baker has primary responsibility for risk analysis on the CSOC segment. Her paper includes a risk analysis matrix concerning the impact of Ada upon application software development. The CSOC development has several subcontractors and each is required to complete the Ada risk matrix if any new software development is required. Depending upon the results of the risk analysis matrix, the subcontractor may be required to use Ada or may receive a waiver.

CSOC has adopted the newly proposed MIL-STD-SDS and the modifications to other existing military standards (e.g., 1521B, 490, 483) as their standard, with the exception that the formal requirements analysis (e.g., using SREM or PSL/PSA) has been modified to an informal level.

Dr. Baker will send a copy of her paper and will maintain contact with us concerning their progress and our progress.

ACTION ITEMS:

- (1) Read Dr. Baker's paper when it arrives.
- (2) Maintain contact with Dr. Baker on the three areas of interest: risk analysis, security, and use of MIL-STD-SDS and Ada.



## CONTACT SUMMARY

SS 304

SUBJECT: AFOTEC Software Supportability Risk Assessment

DATE OF CONTACT: 7/20/84

PLACE CONTACTED: Georgia Institute of Technology Personnel

Meeting Held at BDM/A

1801 Randolph Rd. SE

Albuquerque, NM 87106

PRINCIPAL CONTACTS: Dr. Richard DeMillo Georgia Institute of Technology (GIT)  
Ms. Ronnie Martin Georgia Institute of Technology (GIT)  
Lt. Col. Richard Cline AFOTEC  
Maj. Gary Horlbeck AFOTEC  
Mr. Jim Baca AFOTEC

PERSON(S) MAKING CONTACT: Dr. David E. Peercy, BDM/A

Dr. G. Donald Richardson, BDM/A

## PURPOSE:

Discuss current research in software risk assessment being conducted at Georgia Institute of Technology. Discuss in particular, Dr. DeMillo's risk model for software testing and the Software Test and Evaluation Project (STEP) related work in which Ms. Martin is involved. Discuss current research and objectives of AFOTEC in software supportability risk assessment. Discuss current AFOTEC Subtask 304 objectives.

## BACKGROUND:

GIT personnel had contacted Lt. Col W. Mueller of AFOTEC concerning the work in risk assessment being done at AFOTEC. Col. Mueller felt an exchange of information would be appropriate. During a recent trip to the west coast it was arranged that GIT personnel, AFOTEC personnel, and BDM personnel, as listed above, would meet at BDM/A facilities for such a meeting to exchange information.

## DISCUSSION:

The meeting took place on July 20, 1984 from approximately 12:30 p.m. until 3:00 p.m. Format for the discussions was as follows:

- (1) AFOTEC review of current risk assessment study
- (2) GIT review of current risk assessment research
- (3) BDM review of effort to date on risk assessment task
- (4) Open discussion of issues.

The meeting was profitable in that both GIT and AFOTEC/BDM personnel became familiar with the content and level of detail in the respective risk assessment efforts. It is apparent that both efforts are at the concept level, although GIT effort is probably not as far along as is the AFOTEC effort.

The AFOTEC review was presented by AFOTEC personnel. Basic problems of OT&E supportability risk assessment were presented. A concept briefing on Embedded Computer Resources Risk Assessment was presented. Basic ideas included requirements for:

- (1) Development of a testing concept that provides the user, supporter, and decision makers with a risk assessment of system deployment.
- (2) Development of a risk assessment methodology to provide qualitative and quantitative data on the performance and support of the system which would allow for logical conclusions in risk areas and support for the associated recommendations.
- (3) Development of a test measurement methodology for combining test results into a meaningful metric for the user, supporter, and decision maker.

Some potential hierarchy of assessment factors along the current AFOTEC approach was presented along with some candidate measures of effectiveness/indicators of risk. Objectives of the current AFOTEC risk assessment effort (subtask 304) were also reviewed. These objectives were to:

- (1) Identify candidate OT&E software supportability risk models.
- (2) Identify supporting measures for candidate risk models.

- (3) Identify feasibility/level-of-effort to further develop and implement candidate risk models.

The GIT review primarily focused on briefing slides Dr. DeMillo had prepared summarizing research on "A Risk Model for Software Testing." The major emphasis in this research is to derive a method for determining an optimum software test strategy which would identify critical factors in decisions and reduce the decision risk. A framework for deriving such a method was presented. It is based upon decision theory using a "top down" approach. Some alternative strategies and test policies were presented in example form.

The basic form of a test strategy is to choose a sequence of tests from among a possible set of tests, enumerate the set of possible outcomes from the test (predicted, actual) and, on the basis of the possible pairs (test, outcome) matrix, determine utility functions and risk functions. The goal is to be able to rank possible tester sequences with respect to the utility and risk function values and some optimality criteria. As an example, one test may be high cost and produce high utility and determine risk very well. Another test may be low cost but offer minimal utility, and determines residual risk for only a part of the system. Which test should a tester choose? In constructing a sequence of such tests it may happen that there is some synergy among certain tests when conducted as a segment together (i.e., the sum of the parts is less than the whole). Hopefully, a test strategy would aid determination when such effects occur and the magnitude of the effect.

The BDM review was an informal discussion of the current status of Subtask 304. At the time of this meeting, the draft of the report on literature review, current research review, and data base assemblage had been delivered to AFOTEC. In addition, significant progress had been made toward identification of candidate software supportability risk assessment models. Dates when such reports would be delivered and availability of such reports through AFOTEC or DTIC/NTIS government report distribution services was discussed. A brief background was also presented of previous BDM work for AFOTEC on software maintainability and

software support facility evaluation methodologies, and a current subtask to study computer system security test and evaluation.

The open discussion focused on some aspects of STEP, particularly data collection for software error tracking, and what the issues in software supportability risk assessment (from AFOTEC viewpoint) were. BDM personnel presented some thoughts on the use of a maintenance activity requirements profile dictated by the user to baseline software supportability risk assessments. Such a profile would indicate the required number, type, and complexity of maintenance support requests expected by the user in a given unit of time. A draft guidebook to software T&E specifications for a TEMP should be available from the STEP shortly. There is also a STEP advisory panel on which BDM might want to participate. Mr. Baca of AFOTEC is also familiar with STEP as an AFOTEC focus. The question of AFOTEC/BDM helping sponsor a workshop focusing on risk assessment was posed. It was agreed that such a workshop was needed and should be pursued.

#### CONCLUSIONS:

This was an impromptu and reasonably informal meeting which had some good technical interchange. It was good to learn of Georgia Tech involvement in this area of software risk assessment and all parties agreed to maintain contact and exchange future results.

#### ACTION ITEMS:

- (1) Review "A Risk Model for Software Testing" for possible inclusion in risk assessment task report.

**Appendix D**  
**Contacts/Knowledgeable Persons**

APPENDIX D  
CONTACTS/KNOWLEDGEABLE PERSONS

APPENDIX D  
RISK ASSESSMENT CONTACTS/KNOWLEDGEABLE PERSONS

1.0 INTRODUCTION.

This appendix is a list of points of contact involved with some aspect of Risk Assessment (RA), and who can be generally categorized as experts because of their research or publications, or knowledgeable because of their experience and responsibilities. Each contact included is a prominent author in the field (see the bibliography in appendix H), or has been contacted through visits or conferences, or has been contacted by telephone.

2.0 LIST OF CONTACTS.

The list is given in alphabetical order by name. No list entry is split between pages in order to keep information on each person as readable as possible. Entries include a brief description of responsibilities, title, and areas of expertise/knowledge, where available.

ALPHABETICAL BY INDIVIDUAL'S NAME

Baca, Jim

Alternate Subtask Statement Officer  
AFOTEC/LG5C  
(505) 844-9421

Baker, Dr. Dixie

Space Operations System Division  
The Aerospace Corporation  
El Segundo, CA 90245  
(213) 648-5834

Basili, Dr. Victor

Software Methodology, Metrics  
University of Maryland  
College Park, MD  
(301) 454-4254 X2002

Boehm, Dr. Barry

Software Engineer, Software Economics  
TRW Software Information Systems Division  
Los Angeles, CA  
(213) 535-2184

DeMillo, Dr. Richard

Georgia Institute of Technology  
Atlanta, Georgia 30332  
(404) 894-3130

Fisher, Gerald

AF/SASF  
Washington, D.C.  
(202) 697-9890



Hoessel, William

Subtask Statement 304 Technical Analyst, System Software Cost

BDM/A

(505) 848-5000

Hornbeck, Maj. Gary R.

Subtask Statement 304 Officer

AFOTEC/LG5T

(505) 846-7822

Huebner, Walt

Subtask Statement 304 Task Leader

BDM/A

(505) 848-5000

Leveson, Dr. Nancy

Software Safety

University of California

Irvine, CA

(714) 856-5517

McCall, Jim

Software Quality

Science Applications, Inc.

La Jolla, CA

(619) 456-6220

Musa, John

Software Reliability

Bell Laboratories

Whippany, NJ

(201) 386-2398

Peercy, Dr. David E.

Software Methodology, Security, Maintainability

Subtask Statement 304 Technical Leader

BDM/A

(505) 848-5000

Richardson, Dr. George D.

Statistics, Operations Analyst

Subtask Statement 304 Technical Analyst

BDM/A

(505) 848-5000

Riddle, Dr. William

Software Consultant, Software Development/Support Environments

Software Design & Analysis, Inc.

Boulder, CO

(303) 499-4783

Rowe, Dr. William

Risk Analyst, Risk Assessment Methodology

Risk Limited Corporation

Washington, D.C.

(301) 340-7990

Stubberud, Allen

AF Chief Scientist

AF Chief of Staff/AFCCN

(202) 697-7842

**Appendix E**  
**Author Bibliography Index**

APPENDIX E  
AUTHOR BIBLIOGRAPHIC INDEX

<u>AUTHOR</u>	<u>BIBLIOGRAPHIC INDEX</u>
AGGARWAL, K.	0095
AIR FORCE	0089, 0100, 0125, 0150, 0151, 0152, 0153, 0162, 0235, 0236, 0237, 0239
ANGUS, J. E.	0142
APOSTOLAKIS, G.	0013
ARMY	0015
BANNISTER, J. E.	0113
BARBER, D. E.	0227
BAWCUTT, P. A.	0113
BLACK, M. A.	0226
BOEHM, B. W.	0120, 0130
BOLOTSKY, R.	0241
BOOCH, G.	0144
BOWEN, J. B.	0142
BOWEN, T.	0141
BRATMAN, H.	0078
BROWN, J. R.	0130
BUSHKIN, A. A.	0231
BUTLER, M.	0056
CAMPBELL, R. P.	0233
CHELSON, P. O.	0052
CHURCHWELL, J. B.	0046
CONRAD, J.	0113
COPPOLA, A.	0069
COURTNEY, R. H., JR.	0241
CRAGON, H. G.	0134
CROUCH, E. A. C.	0074
DANIELS, B. K.	0093, 0096
DEFENSE SYSTEMS MANAGMENT COLLEGE	0166
DEMILLO, R.	0105, 0170

AUTHORBIBLIOGRAPHIC INDEX

DEPARTMENT OF NAVY	0161
DIRECTORATE OF AEROSPACE SAFETY	0143, 0160
DoD	0109, 0138, 0238
DOWIE, J.	0119
DURALL, L.	0020
EFRON, B.	0111
FACEMIRE, J. L.	0139
FERENS, D. V.	0010
FINFER, M.	0078
FISHER, G. J.	0009
FISK, F.	0114
FOX, V. M.	0081
GANESH, S. L.	0039
GOVERNMENT ACCOUNTING OFFICE	0136
GAY, E. P.	0009
GEPHART, L. S.	0097
GILLIGAN, J.	0135
GLASS, R. L.	0085, 0098
GODA, J.	0145
GOEL, A. L.	0030, 0049, 0110
GOHEEN, S. M.	0232
GROSS, R. N.	0037
GROVE, H. M.	0123
GUBITZ, M	0063
HECHT, H.	0005
HEIDLER, W.	0101
HELLING, W. D.	0223
HERD, J. H.	0157
HOFFMAN, L.	0135, 0158
HOUGHTON, R. C., JR.	0127
HOWDEN, W. E.	0123
HUDSON, D.	0108
HUEBNER, W. F.	0167, 0168

AUTHORBIBLIOGRAPHIC INDEX

IANNINO, A.	0099
IEEE	0124
IKOKU, C. U.	0116
JELINSKI, Z.	0046
JETTE, G. E.	0117
JONES, S. O.	0103
KAFURA, D. G.	0139, 0140
KOCH, H. S.	0066
KRESS, M P.	0079
KUBAT, P.	0066
LATHROP, F. C.	0121
LEBLANC, R.	0145
LEE, J. A. N.	0140
LEFRERE, P.	0119
LEIBOWITZ, S.	0164
LIENTZ, B. P.	0038, 0131, 0146
LINDQUIST, T. E.	0139, 0140
LIPOW, M.	0130
LITTLEWOOD, B.	0033
MARKHAM, D.	0088
MARTIN, J.	0169
MARTIN, R.	0137
MATSUMOTO, M.	0149
McCALL, J.	0083, 0092, 0149
McCLURE, C.	0169
MEGILL, R. E.	0035
MENDIS, K. S.	0062
MEYER, K,	0004
MOHANTY, S. N.	0048
MORANDA, P. B.	0046
MUNERA, H. A.	0115
MURCH, W. G.	0114
MUSA, J. D.	0043, 0099
NATIONAL BUREAU OF STANDARDS	0122, 0137, 0240

AUTHORBIBLIOGRAPHIC INDEX

NEITZEL, L. A.	0158
NEUGENT, W.	0135, 0234
NOISEUX, R. A.	0085
OKUMOTO, K.	0110
ORCEYRE, M. J.	0241
OSBORNE, W.	0137
OSTERWEIL, L.	0084
OTT	0063
OVADIA, F.	0070
PARIKH, G.	0147
PARISEAU, R. J.	0075
PARRATTO, S. L.	0163, 0164
PEERCY, D. E.	0132, 0133, 0154, 0163, 0164, 0167, 0168
POSTAK, J. N.	0157
PRESSMAN, R. S.	0080
PRINGLE, H. G.	0163, 0164
PRITSKER, A. A. B.	0165
PUTNAM, L. H.	0156, 0159
PYSTER, A.	0148
RAMAMOORTHY, C. V.	0039
RCA	0058
REGDEN, C. D.	0165
RESCHER, N.	0071
REYNOLDS, J. H.	0025
RHODE, R. D.	0232
RICHARDSON, G.	0167, 0168
ROWE, W. D.	0006
RUSSELL, W. E.	0157
SANDS, G. A.	0233
SAYWARD, F. G.	0105
SCHACHT, J. M.	0232
SCHNEIDEWIND, N F.	0086

AUTHORBIBLIOGRAPHIC INDEX

SHEPARD, R. F.	0112
SHOUMAN, M. L.	0155
SMITH, M.	0108
SOI, I.	0095
STEWART, K. R.	0157
SUKERT, A.	0069
SWANSON, E. B.	0131, 0146
SWINSON, G. E.	0103, 0132
SYSCON CORPORATION	0011, 0012
SYSTEMS ARCHITECTS	0018
SYSTEM DEVELOPMENT CORPORATION	0229, 0230
THACKER, J.	0070
THAYER, R.	0148
THAYER, T. A.	0094
THIBODEAU, R.	0027
THOMPSON, W. E.	0052
UNKNOWN	0036, 0061
VANDENBERG, S. J.	0142
VEMURI, V.	0028
VESSEY, I.	0129
VORGANG, B. R.	0041
WALKER, M. G.	0042
WALTERS, G. F.	0088, 0092
WATSON, G.	0024
WATSON, S. R.	0053
WEBER, R.	0129
WESSELS, E.	0068
WHITMORE, D. C.	0016
WILBURN, N.	0104
WILEY, J.	0164
WILSON, R.	0074
WITZKE, E.	0164
WOLVERTON, R. W.	0003, 0159



AUTHOR

BIBLIOGRAPHIC INDEX

WOOD, R.	0148
WORM, G. H.	0008
YADIGAROGLU, G.	0115
YAU, S.	0076, 0171
YOUNG, V. I.	0112

**Appendix F**  
**Title Bibliography Index**

APPENDIX F  
TITLE BIBLIOGRAPHIC INDEXBIBLIO-  
GRAPHIC  
INDEX

## TITLE

- 0231 "A Framework for Computer Security" (Revised Edition), Bushkin, A. A., Santa Monica, CA: System Development Corp. AD-A025 356, Jun 75, (M).
- 0133 "A Framework for Software Maintenance Management Measures," Peercy, David E., Proceedings of the Seventeenth Annual Hawaii International Conference on System Sciences, Jan 84 (P).
- 0227 "A Guide for Developing an ADP Security Plan for Navy Finance Center," Barber, D. E., Monterey, CA: Naval Postgraduate School, AD-A127 244, Dec 82, (M).
- 0030 "A Guidebook for Software Reliability Assessment," Goel, A. L., DTIC, AD-A139240 Aug 83, (P).
- 0041 "A Macro Approach to Software Resource Estimation and Life Cycle Control," Vorgang, B. R., M.A. Thesis, Naval Postgraduate School, 1981, (M).
- 0233 "A Modular Approach to Computer Security Risk Management," Campbell, R. P., and G. A. Sands, Montvale, NM: AFIPS NCC, 48 293-303, Jun 79, (P).
- 0115 "A New Methodology to Quantify Risk Perception," Munera, H. A. and G. Yadigaroglu, Nuclear Science and Engineering, Vol 75, 1980, (P).
- 0114 "A Proposal for Computer Resources Risk Assessment During Operational Test and Evaluation," Fisk, F. B. and W. G. Murch, AFOTEC, 3 Oct 1983, (P).
- 0143 "A Risk Management Guide for Air Force Operations," Directorate of Aerospace Safety, Air Force Inspection and Safety Center (AFISC), Norton AFB, CA, 6 Nov 79, (R).
- 0170 "A Risk Model for Software Testing," DeMillo, R., Georgia Institute of Technology, Briefing Slides, 20 July 84, (P).
- 0075 "A Screening Criterion for Delivered Source in Military Software," Pariseau, R. J., Vol. I & II, Warminster PA: Naval Air Development Center, NTIS, 14 Nov 1979, (M).

BIBLIO-  
GRAPHIC  
INDEXTITLE

- 0084 "A Software Lifecycle Methodology and Tool Support," Osterweil, L., Colorado University Department of Computer Science, CU-CS-154-79, NTIS, AD-A075 335/9, Apr 1979, (M).
- 0154 "A Software Maintainability Evaluation Methodology," Peercy, David E., IEEE Transactions on Software Engineering, Vol SE-7, No. 4, July 1981, (M).
- 0132 "A Software Support Facility Evaluation Methodology," Peercy, David E. and Gary E. Swinson, Symposium on Application and Assessment of Automated Tools for Software Development, Nov 83, (P).
- 0139 "A Specification Technique for the Common APSE Interface Set," Lindquist, Timothy E., Jeffrey L. Facemire, and Dennie G. Kafura, Office of Naval Research, 84004-R, Apr 84, (R).
- 0108 "A Survey of Software Validation, Verification, and Testing Standards and Practices at Selected Sites," Smith, M. and D. Hudson, Boeing Computer Services Co., NBSIR82-2482, NTIS, PB82-209172, Apr 1982, (M).
- 0234 "Acceptance Criteria for Computer Security," Neugent, W., Arlington, VA: AFIPS Press, AFIPS NCC 51, Aug 82, (P).
- 0145 "Ada and Software Development Support: A New Concept in Language Design," LeBlanc, R., and J. Goda, Computer, 15, 5, pp. 75-82, 1982 (P).
- 0117 "Addressing Risk and Uncertainty in Cost Estimating," Jette, G. E., Wright-Patterson Air Force Base: Aeronautical Systems Division, 1983, (M).
- 0003 "Airborne Systems Software Acquisition Engineering Guidebook for Software Cost Analysis and Estimating," Wolverton, R. W., Redondo Beach, CA: TRW Defense and Space Systems Group, Sep 1980, (M).
- 0004 "Airborne Systems Software Acquisition Engineering Guidebook for Supportable Airborne Software," Meyer, K., DTIC, 1980, (M).
- 0005 "Allocation of Resources for Software Reliability," Hecht, H., NTIS, 1981, (P).
- 0121 "Alternative Methods for Risk Analysis: A Feasibility Study," Lathrop, Frank C., Air Force Computer Security Program Office, 2 Sep 1981, (R).

BIBLIO-  
GRAPHIC  
INDEXTITLE

- 0006 An Anatomy of Risk, Rowe, W. D., New York: J. Wiley and Sons, 1977, (B).
- 0009 "An Approach to Risk Analysis, A Process Review," Fisher, Gerald J. and Lt. Col. Eugene P. Gay, An AF/SA Technical Note, Jun 81, (P).
- 0035 An Introduction to Risk Analysis, Megill, R. E., Tulsa: Petroleum Publishing Co., 1977, (B).
- 0008 "Applied Risk Analysis with Dependence Among Cost Components," Worm, G. H., Clemson University, Dept. of Industrial Management, 1981, (M).
- 0235 "Automatic Data Processing (ADP) Security Policy Procedures and Responsibilities," Air Force, AFR 205-16, Washington, D.C.: Department of the Air Force, Headquarters, U.S. Air Force, Aug 84, (R).
- 0161 "Automatic Data Processing Security Program," Department of the Navy, OPNAVINST 5239.1A, Office of the Chief of Naval Operations, Washington, D.C., 3 Aug 82, (R).
- 0011 "Avionics Software Support Cost Model," Syscon Corporation, AFWAL-TR-82-1173, 1 Feb 83, (P).
- 0012 "Avionics Software Support Cost Model: User's Manual," Syscon Corporation, AFWAL-TR-83-1071, May 83, (P).
- 0010 "Avionics Software Support Estimating," Ferens, D. V., Wright-Patterson AFB, OH 45433, 1983, (P).
- 0013 "Bayesian Methods in Risk Assessment," Apostolakis, G., Advances in Nuclear Science and Technology, New York: Plenum, 1981, (B).
- 0015 "Compendium on Risk Analysis Techniques," Army, U.S. Army Material Systems Analysis Agency: Aberdeen Proving Ground, MD, 1972, (M).
- 0016 "Computer Program Maintenance," Whitmore, D.C., et al, Boeing Aerospace Co.; NTIS, AD-A083 209/7, Dec 77, (M).
- 0152 "Computer Programming Languages," Air Force, AFR 300-10, Headquarters U.S. Air Force, Washington, D.C., May 1976, (P).
- 0228 "Computer Security for the Computer Systems Manager," Helling, W. D., Monterey, CA: Naval Postgraduate School, AD-A126 768, Dec 82, (M).

BIBLIO-  
GRAPHIC  
INDEXTITLE

- 0164 "Computer System Security (CSS) Literature Review, Current Research Review, and Data Base Assemblage," (INTERIM), Leibowitz, S., S. Parratto, D. Peercy, H. Pringle, J. Wiley, E. Witzke, BDM/A-84-108-TR, The BDM Corporation, May 84, (R).
- 0163 "Computer System Security (CSS) Test and Evaluation (T&E) Life-Cycle Process Definition," (FINAL), Parratto, S. L., D. E. Peercy, H. G. Pringle, BDM/A-84-0320-TR, The BDM Corporation, 31 Aug 84, (R).
- 0018 "Computer Systems Acquisition Metrics," Vols I-II, Systems Architects Inc., DTIC, AD-A120375, May 1982, (M).
- 0241 "Considerations in the Selection of Security Measures for Automatic Data Processing Systems," Orceyre, M. J., R. H. Courtney, Jr., R. Bolotsky, Department of Commerce, National Bureau of Standards, NBS SP 500-33, Jun 1978 (R).
- 0128 "Contemporary Software Development Environments," Howden, William E., Communications of the ACM, Vol 25, 5, May 1982, (P).
- 0230 "Countermeasures," SDC, McLean, VA: System Development Corp., AD-A072 245, Jun 79, (M).
- 0020 "Data Needs for Software Reliability Modeling," Durall, Lorraine, et al, DACS 82 (1793), 1980, (P).
- 0116 "Decision Analysis: How to Make Risk Evaluations," Ikoku, C. U., World Oil, Sep 1980, (P).
- 0123 "DoD Policy for Acquisition of Embedded Computer Resources," Grove, H. Mark, CONCEPTS, The Journal of Defense Systems Acquisition Management, Vol 5, No. 4, Special Issue-Managing Software, Autumn, 1982, (P).
- 0226 "DoD/DON Requirements for Computer Risk Assessments," Black, M. A., et al, Monterey, CA: Naval Postgraduate School, AD-A132 202, Jun 83, (M).
- 0024 "Evaluation of Computer Software in an Operational Environment," Watson, G., Center for Naval Analysis, Alexandria, VA, NTIS, AD-A091 213/9, Aug 80, (M).
- 0025 "Evaluation of Contemporary Software Engineering Techniques for a Large FORTRAN Simulation," Reynolds, John H., DACS 33 (2401), 1980, (P).

BIBLIO-  
GRAPHIC  
INDEXTITLE

- 0156 "Example of an Early Sizing, Cost and Schedule Estimate for an Application Software System," Putnam, L. H., Computer Software and Applications Conference Proceedings, IEEE Computer Society, November 78, (R).
- 0136 "Federal Agencies' Maintenance of Computer Programs: Expensive and Undermanaged," GAO, Reports to the Congress, Government Accounting Office, AFMD-81-25, 26 Feb 81, (R).
- 0028 "Figures of Merit for Software Quality," Vemuri, V., DACS 83 (2598), 1980, (P).
- 0137 "Guidance on Software Maintenance," NBS, Martin R., and W. Osborne, NBS Special Publication 500-106, National Bureau of Standards, Institute for Computer Sciences and Technology, Dec 83, (R).
- 0240 "Guideline for Automatic Data Processing Risk Analysis," NBS, U.S. Department of Commerce National Bureau of Standards, FIPS PUB 65, Aug 79, (P).
- 0122 "Guidelines for Automatic Data Processing Physical Security and Risk Management," National Bureau of Standards, FIPS PUB 31, Jun 74, (R).
- 0033 "How to Measure Software Reliability and How Not To," Littlewood, B., IEEE Transactions on Reliability, Vol R-28, No. 2, NTIS, Jun 1979, (M).
- 0124 "IEEE Software Reliability Guide, Second Draft - M58 (Risk Assessment), M59 (Software Functional Test Coverage Index), M60 (Software Maturity Index)," IEEE Software Working Group P, 8 Mar 1984, (P).
- 0153 "Independent Cost Analysis Program," Air Force, AFR 173-11, Headquarters U.S. Air Force, Washington, D.C., Dec 1980, (P).
- 0158 "Inexact Analysis of Risk," Hoffman, Lance J., L. A. Neitzel, Computer Security Manual, Vol 1, Spring 1981.
- 0150 "Information Processing Standards for Computers (IPSC)", Air Force, AFR 300-16, Headquarters U.S. Air Force, Washington, D.C., Jun 1979 (P).
- 0036 "Instructions for Using Risk Analysis Matrix," Unknown, (P).
- 0165 Introduction to Simulation and SLAM, Pritsker, A. A. B., C. D. Regden, New York: John Wiley, 1979, (B).

BIBLIO-  
GRAPHIC  
INDEXTITLE

- 0160 "Introduction to System Safety for Program Managers," Directorate of Aerospace Safety, Air Force Inspection and Safety Center (AFISC), Norton AFB, CA, 14 July 80, (R).
- 0037 "Issues and Perspectives in the Validation of Tactical Software," Gross, R. N., Naval Ocean Systems Center, NOSC/TD-139, NTIS, AD-A056 061/5ST, 1 Feb 78, (M).
- 0038 "Issues in Software Maintenance and Measurement," Lientz, B., UCLA Graduate School of Management, Los Angeles, NTIS, AD-A098 982/2, May 81, (P).
- 0039 "Issues in Software Reliability," Ramamoorthy, C. V. and S. L. Ganesh, Symposium on Reliability in Distributed Software and Database Systems, 113-116, NTIS, 1981, (M).
- 0236 "Management of Operational Test and Evaluation," Air Force, Washington, D.C.: Department of the Air Force, Headquarters U.S. Air Force, Jun 79, (P).
- 0042 "Managing Software Reliability, The Paradigmatic Approach," Walker, M. G., New York: Elsevier North Holland, NTIS, 1981, (M).
- 0043 "Measuring and Managing Software Reliability," Musa, J. D., IEEE 1983 Phoenix Conference on Computers and Communications, 1983, (P).
- 0171 Methodology for Software Maintenance, Yau, S. S., Rome Air Development Center, Griffis AFB, NY, RADC-TR-82-262, Feb 84, (R).
- 0046 "Metrics of Software Quality," Jelinski, Z., P. B. Moranda, and J. B. Churchwell, NTIS, AD-A093 788, Nov 80, (M).
- 0048 "Models and Measurements for Quality Assessment of Software," Mohanty, Siba N., Computing Surveys, Vol II, No. 3, DACS 82 (1673), Sep 1979, (P).
- 0049 "Models for Hardware-Software System Operational Performance Evaluation," Goel, Amrit L., IEEE Transactions on Reliability, Vol R-30, No. 3, DACS 83 (2606), 1981, (P).
- 0053 "On Risks and Acceptability," Watson, S. R., NTIS, 82-09 07140, 1981, (P).
- 0052 "On the Specification and Testing of Software Reliability," Thompson, E. E. and P. O. Chelson, Proceedings, Annual Reliability and Maintainability Symposium, 1980, (P).



BIBLIO-  
GRAPHIC  
INDEX

## TITLE

- 0162 "OT&E Reporting," Air Force, Air Force Operational Test and Evaluation Center Regulation 55-1(C2), Chapter 6, 15 Mar 84, (R).
- 0056 "Portability and the National Energy Software Center," Butler, M., Argonne National Lab, NTIS, CONF-781052-1, 1978, (M).
- 0113 Practical Risk Management, Bannister, J. E. and P. A. Bawcutt, London: Witherby and Co., 1981, (B).
- 0058 "Price Parametric Cost Models," RCA/Price Systems, (P).
- 0131 "Problems in Application Software Maintenance," Lientz, Bennet P. and E. Burton Swanson, Communications of the ACM, Vol 24, 11, Nov 81, (P).
- 0151 "Procedures for Managing Automated Data Processing Systems Documentation, Development, Acquisition, and Implementation," Air Force, AFR 300-12, Vol I, Headquarters U.S. Air Force, Washington, D.C., Dec 1977, (P).
- 0061 "Proposed Methodology for Treating Hardware/Software Failures During OT&E," Unknown, (P).
- 0062 "Quantifying Software Quality," Mendis, Kenneth S., Quality Progress, DACS 83(2701), May 1982, (P).
- 0063 "Quantifying Software Reliability by a Probabilistic Model," Gubitzi, M. and K. O. Ott, NTIS, 1983 (P).
- 0130 "Quantitative Evaluation of Software Quality," Boehm, B. W., J. R. Brown and M. Lipow, Proceedings 2nd International Conference on Software Engineering, San Francisco, CA, pp. 592-605, 1976, (P).
- 0159 "Quantitative Management: Software Cost Estimating," Putnam, L. H., R. W. Wolverton, Computer Software and Applications Conference Tutorial, IEEE Computer Society, November 77, (R).
- 0112 "Quantitative Techniques for DARPA Program Risk Management," Shepard, R. F. and V. I. Young, Falls Church, VA: Meridian Corporation, 1983, (M).
- 0066 "Quick and Simple Procedures to Assess Software Reliability and Facilitate Project Management," Koch, H. S. and P. Kubat, The Journal of Systems and Software, 1981, (P).

BIBLIO-  
GRAPHIC  
INDEXTITLE

- 0068 "Rating Techniques for Risk Assessment," Wessels, E., NTIS, 81-02 00219, Mar 80, (P).
- 0069 "Reliability and Maintainability Management Manual," Coppola, A. and A. Sukert, Rome Air Development Center, RADC-TR-79-200, Jul 79, (M).
- 0142 "Reliability Model Demonstration Study," RADC, Angus, J. E., J. B. Bowen, S. J. VanDenBerg, Volumes I and II, Rome Air Development Center (COEE), RADC-TR-83-207, August 1983, (M).
- 0070 "Reliability Measurement for Operational Avionics Software," Thacker, J. and F. Ovadia, NTIS, Sep 79, (M).
- 0071 Risk, Rescher, N., Washington, D.C.: University Press of America, 1983, (B).
- 0119 Risk and Chance, Dowie, J. and P. Lefrere (eds.), Milton Keynes, England: The Open University Press, 1980, (B).
- 0229 "Risk Assessment Methodology," SDC, McLean, VA: System Development Corp., AD-A072 249, Jul 79, (M).
- 0166 Risk Assessment Techniques, Defense Systems Management College, Fort Belvoir, Virginia, July 1983, (R).
- 0074 Risk/Benefit Analysis, Crouch, E. A. C. and R. Wilson, Cambridge, MA: Ballinger, 1982 (B).
- 0076 "Self-Metric Software," Yau, S., Vol. I, U. Northwestern, NTIS, AD-A086 290/4, Apr 1980, (M).
- 0118 Society, Technology, and Risk Assessment, Conrad, J. (ed.), New York: Academic Press, 1980, (B).
- 0078 "Software Acquisition Management Guidebook Verification," Bratman, H. and M. Finfer, System Development Corporation, SDC-TM-5772/002/02, NTIS, AD-A048 577/1ST, Aug 1977, (M).
- 0081 Software and Its Development, Fox, V. M., Englewood Cliffs, NJ: Prentice-Hall, 1982, (B).
- 0079 "Software Configuration Management," Kress, M. P., Boeing Aerospace Co., NTIS, AD-A083, 2 Jan 1979, (M).
- 0157 "Software Cost Estimation Study," Herd, J. H., J. N. Postak, W. E. Russell, K. R. Stewart, Rome Air Development Center, Griffis AFB, NY, RADC-TR-77-220, Vols I and II, June 1977, (R).

BIBLIO-  
GRAPHIC  
INDEXTITLE

- 0127 "Software Development Tools: A Profile," Houghton, Raymond C., Jr., Computer, May 83, (P).
- 0080 Software Engineering: A Practitioner's Approach, Pressman, R. S., New York: McGraw-Hill, 1982, (B).
- 0155 Software Engineering, Design, Reliability and Management, Shooman, M. L., New York: McGraw-Hill, 1983, (B).
- 0120 Software Engineering Economics, Boehm, B., Englewood Cliffs, NJ: Prentice-Hall, 1981 (B).
- 0144 Software Engineering With Ada, Booch, G., Reading, MA; Benjamin/Cummings, 1983, (B).
- 0085 Software Maintenance Guidebook, Glass, Robert L. and Ronald A. Noiseux, DACS 83(2948), 1981, (B).
- 0086 "Software Maintenance: Improvement Through Better Development Standards," Schneidewind, N. F., Naval Postgraduate School, NPS-54-82-002, NTIS, AD-A113 257/0, 22 Feb 1982, (M).
- 0146 Software Maintenance Management, Lientz, B., and E. Swanson, Reading, MA: Addison-Wesley, 1980, (B).
- 0169 Software Maintenance: The Problem and Its Solution, Martin, J., C. McClure, London: Prentice-Hall International, Inc., 1983, (B).
- 0088 "Software Metrics Application Techniques," Markham, D., J. McCall and G. Walters, DACS 83(3005), 1981, (P).
- 0089 "Software Operational Test and Evaluation Guidelines," Air Force, Vol. I, 10 Nov 1982, Vol. III, 1 Jan 1984, Vol. V, 25 Jul 1983, AFOTEC, (R).
- 0237 "Software OT&E Guidelines Volume II Handbook for the Deputy for Software Evaluation," Air Force, Kirtland AFB, NM: Air Force Test and Evaluation Center, Sep 81, (P).
- 0141 "Software Quality Measurement for Distributed Systems," RADC, T. Bowen, et al, Rome Air Development Center, Vols I, II, III, Jul 83, (P).
- 0149 "Software Quality Measurement Manual," McCall, J. and M. Matsumoto, RADC-TR-80-109, Vol I and II, Apr 80, (R).

BIBLIO-  
GRAPHIC  
INDEXTITLE

- 0092 "Software Quality Metrics for Life-Cycle Cost-Reduction," Walters, Gene F. and J. A. McCall, IEEE Transactions on Reliability, Vol R-28, No. 3, DACS 82(1679), Aug 1979, (P).
- 0093 "Software Reliability," Daniels, B. K., NTIS, 1983, (P).
- 0094 "Software Reliability: A Study of Large Project Reality," Thayer, T. A., et al, New York: Elsevier North-Holland, NTIS, 1978, (M).
- 0097 "Software Reliability: Determination and Prediction," Gephart, L. S., et al, U. Dayton, Air Force Flight Dynamics Lab, NTIS, AD-A069 976/9ST, Jun 1978, (P).
- 0095 "Software Reliability and Maintainability: A Life-Cycle Cost Viewpoint," Soi, I. and K. Aggarwal, Reliability in Electrical and Electronic Components and Systems, NTIS, 1982, (P).
- 0096 "Software Reliability Assessment," Daniels, B. K., Microprocessors: Safety Implications for Industry, NTIS, 1982, (M).
- 0098 "Software Reliability Guidebook," Glass, R. L., NTIS, 1979, (M).
- 0099 "Software Reliability Modeling--Accounting for Program Size Variation Due to Integration and Design Changes," Musa, J. D. and A. Iannino, NTIS, 1981, (P).
- 0100 "Software Safety Handbook," Air Force, (Draft) HQ AFISC/SESD, Norton AFB, CA, 1984, (P).
- 0167 "Software Supportability Risk Assessment in OT&E: An Evaluation of Risk Assessment Methodologies," (FINAL), Huebner, W., D. Percy, G. Richardson, BDM/A-84-496-TR, The BDM Corporation, 31 Aug 84, (R).
- 0168 "Software Supportability Risk Assessment in OT&E: Measures for a Risk Assessment Model," (FINAL), Huebner, W., D. Percy, G. Richardson, BDM/A-84-565-TR, The BDM Corporation, 28 Sept 84, (R).
- 0138 "Software Technology for Adaptable, Reliable Systems (STARS) Program Strategy," DoD, Department of Defense, 1 Apr 83, (R).
- 0101 "Software Testing Measures," Heidler, W., et al, General Research Corp., NTIS, AD-A118 254, May 1982, (M).

BIBLIO-  
GRAPHIC  
INDEXTITLE

- 0129 "Some Factors Affecting Program Repair Maintenance: An Empirical Study," Vessey, Iris and Ron Weber, Communications of the ACM, Vol 26, 2, Feb 83, (P).
- 0103 "Standard Software Support Facility Evaluation Final Report," Swinson, Gary E. and Stephen O. Jones, BDM/TAC-80-693-TR, 28 Nov 1980, (R).
- 0104 "Standards and Guidelines Applicable to Scientific Software Lifecycle," Wilburn, N., Hanford Engineering Development Lab, HEDL-SA-2553-FP, NTIS, 1981, (M).
- 0105 "Statistical Measures of Software Reliability," DeMillo, R. and F. G. Sayward, Georgia Institute of Technology, GIT-ICS-80, NTIS, AD-A100 662, Oct 80, (M).
- 0109 "System Safety Program Requirements," DoD, MIL-STD-882B, 30 Mar 1984, (R).
- 0147 Techniques of Program and System Maintenance, Parikh, G., Cambridge, MA: Winthrop, 1982, (B).
- 0135 "Technology Assessment: Methods for Measuring the Level of Computer Security," Neugent, William, John Gilligan, and Lance Hoffman, National Bureau of Standards, Institute for Computer Sciences and Technology, Washington, D.C., Sep 81, (R).
- 0239 "Test and Evaluation," Air Force, AFR 80-14, Washington, D.C.: Department of the Air Force, Headquarters U.S. Air Force, Sep 80, (P).
- 0238 "Test and Evaluation," DoD, CDD 5000.3 Washington, D.C.: Department of Defense, Dec 79, (P).
- 0027 "The Feasibility of Obtaining Software Research Data at the U.S. Army Computer Systems Command," Thibodeau, R., General Research Corporation, NTIS, AD-A107 883, 15 Jul 80, (M).
- 0125 "The High Cost and Risk of Mission-Critical Software," USAF Scientific Advisory Board, Ad Hoc Committee Report, Dec 1983, (R).
- 0111 The Jackknife, Bootstrap, and Other Resampling Plans, Efron, B., Philadelphia: Society for Industrial and Applied Mathematics, 1982, (B).
- 0134 "The Myth of the Hardware/Software Cost Ratio," Cragon, Harvey G., Computer, Open Channel, Dec 82, (P).

BIBLIO-  
GRAPHIC  
INDEXTITLE

- 0232 "User Requirements for Computer Security," Schacht, J. M., S. M. Goheen, and R. D. Rhode, Bedford, MA: MITRE Corp., AD-A073 101, May 79, (M).
- 0148 "Validating Solutions to Major Problems in Software Engineering Project Management," Thayer, R., A. Pyster, and R. Wood, Computer, 15, 8, pp. 65-77, Aug 1982, (P).
- 0140 "Validation in Ada Programming Support Environments," Kafura, Dennis, J. A. N. Lee, and Timothy Lindquist, Engineering Psychology Group, Office of Naval Research, Working Paper, NRSRO-101, 7 Jan 83, (R).
- 0110 "When to Stop Testing and Start Using Software," Goel, Amrit L. and Kazuhira Okumoto, DACS 83(2754), 1981, (P).

**Appendix G**  
**Date Bibliography Index**

APPENDIX G  
DATE BIBLIOGRAPHIC INDEX

<u>Index Number</u>	<u>Date</u>	<u>Title</u>
0015	1972	Army, "Compendium on Risk Analysis Techniques," U.S. Army Material Systems Analysis Agency: Aberdeen Proving Ground, MD, (M).
0122	Jun 1974	National Bureau of Standards, "Guidelines for Automatic Data Processing Physical Security and Risk Management," FIPS PUB 31, (R).
0231	Jun 1975	Bushkin, A. A., "A Framework for Computer Security" (Revised Edition), Santa Monica, CA: System Development Corp. AD-A025 356, (M).
0152	May 1976	Air Force, "Computer Programming Languages," AFR 300-10, Headquarters U.S. Air Force, Washington, D.C., (P).
0130	1976	Boehm, B. W., J. R. Brown, and M. Lipow, "Quantitative Evaluation of Software Quality," <u>Proceedings 2nd International Conference on Software Engineering</u> , San Francisco, CA, pp. 592-605, (P).
0157	Jun 1977	Herd, J. H., J. N. Postak, W. E. Russell, K. R. Stewart, "Software Cost Estimation Study," Rome Air Development Center, Griffis AFB, NY, RADC-TR-77-220, Vols. I and II, (R).
0073	Aug 1977	Bratman, H. and M. Finfer, "Software Acquisition Management Guidebook Verification," System Development Corporation, SDC-TM-5772/002/02, NTIS, AD-A048 577/1ST, (M).
0159	Nov 1977	Putnam, L. H., R. W. Wolverton, "Quantitative Management: Software Cost Estimating," Computer Software and Applications Conference Tutorial, IEEE Computer Society, (R).
0016	Dec 1977	Whitmore, D. C., et al, "Computer Program Maintenance," Boeing Aerospace Co., NTIS, AD-A083 209/7, (M).
0151	Dec 1977	Air Force, "Procedures for Managing Automated Data Processing Systems Documentation, Development, Acquisition, and Implementation," AFR 300-12, Vol. I, Headquarters U.S. Air Force, Washington, D.C., (P).



<u>Index Number</u>	<u>Date</u>	<u>Title</u>
0006	1977	Rowe, W. D., <u>An Anatomy of Risk</u> , New York: J. Wiley and Sons, (B).
0035	1977	Megill, R. E., <u>An Introduction to Risk Analysis</u> , Tulsa: Petroleum Publishing Co., (B).
0037	Feb 1978	Gross, R. N., "Issues and Perspectives in the Validation of Tactical Software," Naval Ocean Systems Center, NOSC/TD-139, NTIS, AD-A056 061/5ST, (M).
0241	Jun 1978	Orceyre, M. J., R. H. Courtney, Jr., and R. Bolotsky, "Considerations in the Selection of Security Measures for Automatic Data Processing Systems," Department of Commerce, National Bureau of Standards, NBS SP 500-33, (R).
0097	Jun 1978	Gephart, L. S., et al, "Software Reliability: Determination and Prediction," U. Dayton, Air Force Flight Dynamics Lab, NTIS, AD-A069 976/9ST, (M).
0156	Nov 1978	Putnam, L. H., "Example of an Early Sizing, Cost and Schedule Estimate for an Application Software System," Computer Software and Applications Conference Proceedings, IEEE Computer Society, (R).
0056	1978	Butler, M., "Portability and the National Energy Software Center," Argonne National Lab, NTIS, CONF-781052-1, (M).
0094	1978	Thayer, T. A., et al, "Software Reliability: A Study of Large Project Reality," New York: Elsevier North-Holland, NTIS, (M).
0079	Jan 1979	Kress, M. P., "Software Configuration Management," Boeing Aerospace Co., NTIS, AD-A083, (M).
0084	Apr 1979	Osterweil, L., "A Software Lifecycle Methodology and Tool Support," Colorado University Department of Computer Science, CU-CS-154-79, NTIS, AD-A076 335/9, (M).
0232	May 1979	Schacht, J. M., S. M. Goheen, and R. D. Rhode, "User Requirements for Computer Security," Bedford, MA: MITRE Corp., AD-A073 101, (M).
0233	Jun 1979	Campbell, R. P., and G. A. Sands, "A Modular Approach to Computer Security Risk Management," Montvale, NJ: AFIPS NCC, 48 293-303, (P).

<u>Index Number</u>	<u>Date</u>	<u>Title</u>
0230	Jun - 1979	SDC, "Countermeasures," McLean, VA: System Development Corp., AD-A072 245, (M).
0033	Jun 1979	Littlewood, B., "How to Measure Software Reliability and How Not To," IEEE Transactions on Reliability, Vol. R-28, No. 2, NTIS, (M).
0150	Jun 1979	Air Force, "Information Processing Standards for Computers (IPSC)", AFR 300-16, Headquarters U.S. Air Force, Washington, D.C., (P).
0236	Jun 1979	Air Force, "Management of Operational Test and Evaluation," Washington, D.C.: Department of the Air Force, Headquarters U.S. Air Force (P).
0069	Jul 1979	Coppola, A. and A. Sukert, "Reliability and Maintainability Management Manual," Rome Air Development Center, RADC-TR-79-200, (M).
0229	Jul 1979	SDC, "Risk Assessment Methodology," McLean, VA: System Development Corp., AD-A072 249, (M).
0240	Aug 1979	NBS, "Guideline for Automatic Data Processing Risk Analysis," U.S. Department of Commerce National Bureau of Standards, FIPS PUB 65, (P).
0092	Aug 1979	Walters, Gene F., and J. A. McCall, "Software Quality Metrics for Life-Cycle Cost-Reduction," IEEE Transactions on Reliability, Vol. R-28, No. 3, DACS 82(1679) (P).
0048	Sep 1979	Mohanty, Siba N., "Models and Measurements for Quality Assessment of Software," Computing Surveys, Vol. II, No. 3, DACS 82(1673), (M).
0070	Sep 1979	Thacker, J. and F. Ovidia, "Reliability Measurement for Operational Avionics Software," NTIS, (P).
0143	Nov 1979	Directorate of Aerospace Safety, "A Risk Management Guide for Air Force Operations," Air Force Inspection and Safety Center (AFISC), Norton AF3, CA, (R).
0075	Nov 1979	Pariseau, R. J., "A Screening Criterion for Delivered Source in Military Software," Vol. I and II, Warminster, PA: Naval Air Development Center, NTIS, (M).

<u>Index Number</u>	<u>Date</u>	<u>Title</u>
0238	Dec 1979	DoD, "Test and Evaluation," DODD 5000.3 Washington, D.C.: Department of Defense, (P).
0165	1979	Pritsker, A. A. B. and C. D. Regden, <u>Introduction to Simulation and SLAM</u> , New York: John Wiley, (B).
0098	1979	Glass, R. L., "Software Reliability Guidebook," NTIS, (M).
0068	Mar 1980	Wessels, E., "Rating Techniques for Risk Assessment," NTIS, 81-02 00219, (P).
0076	Apr 1980	Yau, S., "Self-Metric Software," Vol. I, Northwestern Univ., NTIS, AD-A086 290/4, (M).
0149	Apr 1980	McCall, J. and M. Matsumoto, "Software Quality Measurement Manual," RADC-TR-80-109, Vol. I and II, (R).
0160	Jul 1980	Directorate of Aerospace Safety, "Introduction to System Safety for Program Managers," Air Force Inspection and Safety Center (AFISC), Norton AFB, CA, (R).
0027	Jul 1980	Thibodeau, R., "The Feasibility of Obtaining Software Research Data at the U. S. Army Computer Systems Command," General Research Corporation, NTIS, AD-A107-883, (M).
0024	Aug 1980	Watson, G., "Evaluation of Computer Software in an Operational Environment," Center for Naval Analysis, Alexandria, VA, NTIS, AD-A091 213/9, (M).
0003	Sep 1980	Wolverton, R. W., "Airborne Systems Software Acquisition Engineering Guidebook for Software Cost Analysis and Estimating," Redondo Beach, CA: TRW Defense and Space Systems Group, (M).
0116	Sep 1980	Ikoku, C. U., "Decision Analysis: How to Make Risk Evaluations," <u>World Oil</u> , (P).
0239	Sep 1980	Air Force, "Test and Evaluation," AFR 80-14, Washington, D.C.: Department of the Air Force, Headquarters U.S. Air Force, (P).
0105	Oct 1980	DeMillo, R. and F. G. Sayward, "Statistical Measures of Software Reliability," Georgia Institute of Technology, GIT-ICS-80, NTIS, AD-A100 662, (M).

<u>Index Number</u>	<u>Date</u>	<u>Title</u>
0103	Nov - 1980	Swinson, Gary E. and Stephen O. Jones, "Standard Software Support Facility Evaluation Final Report," BDM/TAC-80-693-TR, (R).
0046	Nov 1980	Jelinski, Z., P. B. Moranda, and J. B. Churchwell, "Metrics of Software Quality," NTIS, AD-A093 788, (M).
0153	Dec 1980	Air Force, "Independent Cost Analysis Program," AFR 173-11, Headquarters U.S. Air Force, Washington, D.C., (P).
0004	1980	Meyer, K., "Airborne Systems Software Acquisition Engineering Guidebook for Supportable Airborne Software," DTIC, (M).
0020	1980	Durall, Lorraine; et al, "Data Needs for Software Reliability Modeling," DACS 82 (1793), (P).
0025	1980	Reynolds, John H., "Evaluation of Contemporary Software Engineering Techniques for a Large FORTRAN Simulation," DACS 83 (2401), (P).
0119	1980	Dowie, J. and P. Lefrere (eds.), <u>Risk and Chance</u> , Milton Keynes, England: The Open University Press, (B).
0115	1980	Munera, H. A. and G. Yadigaroglu, "A New Methodology to Quantify Risk Perception," <u>Nuclear Science and Engineering</u> , Vol. 75, (P).
0028	1980	Vemuri, V., "Figures of Merit for Software Quality," DACS 83 (2598), (P).
0052	1980	Thompson, W. E., and P. O. Chelson, "On the Specification and Testing of Software Reliability," Proceedings, Annual Reliability and Maintainability Symposium, (P).
0118	1980	Conrad, J., (ed.), <u>Society, Technology, and Risk Assessment</u> , New York: Academic Press, (B).
0146	1980	Lientz, B., and E. Swanson, <u>Software Maintenance Management</u> , Reading, MA: Addison-Wesley, (B).

<u>Index Number</u>	<u>Date</u>	<u>Title</u>
0136	Feb 1981	GAO, "Federal Agencies' Maintenance of Computer Programs: Expensive and Undermanaged," Reports to the Congress, Government Accounting Office, AFMD-81-25, (R).
0038	May 1981	Lientz, B., "Issues in Software Maintenance and Measurement," UCLA Graduate School of Management, Los Angeles, NTIS, AD-A098 982/2, (M).
0009	Jun 1981	Fisher, Gerald J. and Lt. Col. Eugene P. Gay, "An Approach to Risk Analysis, A Process Review," An AF/SA Technical Note, (P).
0154	Jul 1981	Peercy, David E., "A Software Maintainability Evaluation Methodology," IEEE Transactions on Software Engineering, Vol. SE-7, No. 4, (M).
0121	Sep 1981	Lathrop, Frank C., "Alternative Methods for Risk Analysis: A Feasibility Study," Air Force Computer Security Program Office, (R).
0237	Sep 1981	Air Force, "Software OT&E Guidelines Vol. II Handbook for the Deputy for Software Evaluation," Kirtland AFB, NM: Air Force Test and Evaluation Center, (P).
0135	Sep 1981	Neugent, William, John Gilligan, and Lance Hoffman, "Technology Assessment: Methods for Measuring the Level of Computer Security," National Bureau of Standards, Institute for the Computer Sciences and Technology, Washington, D.C., (R).
0131	Nov 1981	Lientz, Bennet P., and E. Burton Swanson, "Problems in Application Software Maintenance," Communications of the ACM, Vol. 24, 11, (P).
0158	Spring 1981	Hoffman, Lance J., and L. A. Neitzel, "Inexact Analysis of Risk," <u>Computer Security Manual</u> , Vol. 1, (P).
0041	1981	Vorgang, B. R., "A Macro Approach to Software Resource Estimation and Life Cycle Control," M.A. Thesis, Naval Postgraduate School, (M).
0005	1981	Hecht, H., "Allocation of Resources for Software Reliability," NTIS, (P).

<u>Index Number</u>	<u>Date</u>	<u>Title</u>
0008	1981	Worm, G. H., "Applied Risk Analysis with Dependence Among Cost Components," Clemson University, Department of Industrial Management, (M).
0013	1981	Apostolakis, G., "Bayesian Methods in Risk Assessment," <u>Advances in Nuclear Science and Technology</u> , New York: Plenum, (B).
0039	1981	Ramamoorthy, C. V. and S. L. Ganesh, "Issues in Software Reliability," Symposium on Reliability in Distributed Software and Database Systems, 113-116, NTIS, (P).
0042	1981	Walker, M. G., "Managing Software Reliability, The Paradigmatic Approach," New York: Elsevier North Holland, NTIS, (M).
0049	1981	Goel, Amrit L., "Models for Hardware-Software System Operational Performance Evaluation," IEEE Transactions on Reliability, Vol. R-30, No. 3, DACS 83 (2606), (P).
0053	1981	Watson, S. R., "On Risks and Acceptability," NTIS, 82-09 07140, (P).
0113	1981	Bannister, J. E. and P. A. Bawcutt, <u>Practical Risk Management</u> , London: Witherby and Co., (B).
0066	1981	Koch, H. S. and P. Kubat, "Quick and Simple Procedures to Assess Software Reliability and Facilitate Project Management," The Journal of Systems and Software (P).
0120	1981	Boehm, B., <u>Software Engineering Economics</u> , Englewood Cliffs, NJ: Prentice-Hall, (B).
0085	1981	Glass, Robert L. and Ronald A. Noiseux, <u>Software Maintenance Guidebook</u> , DACS 83 (2948) (B).
0088	1981	Markham, D., J. McCall and G. Walters, "Software Metrics Application Techniques," DACS 83(3005) (P).
0099	1981	Musa, J. A. and A. Iannino, "Software Reliability Modeling - Accounting for Program Size Variation Due to Integration and Design Changes," NTIS, (P).

<u>Index Number</u>	<u>Date</u>	<u>Title</u>
0104	1981	Wilburn, N. "Standards and Guidelines Applicable to Scientific Software Lifecycle," Hanford Engineering Development Lab, HEDL-SA-2553-FP, NTIS, (M).
0110	1981	Goel, Amrit L. and Kazuhira Okumoto, "When to Stop Testing and Start Using Software," DACS 83(2754), (M).
0086	Feb 1982	Schneidewind, N. F., "Software Maintenance: Improvement Through Better Development Standards," Naval Postgraduate School, NPS-54-82-002, NTIS, AD-A113 257/0, (M).
0108	Apr 1982	Smith, M. and D. Hudson, "A Survey of Software Validation, Verification, and Testing Standards and Practices at Selected Sites," Boeing Computer Services Co., NBSIR82-2482, NTIS, PB82-209172(M).
0018	May 1982	Systems Architects, "Computer Systems Acquisition Metrics," Vols I-II, Systems Architects Inc., DTIC AD-A120375, (M).
0128	May 1982	Howden, William E., "Contemporary Software Development Environments," Communications of the ACM, Vol. 25, 5, (P).
0062	May 1982	Mendis, Kenneth S., "Quantifying Software Quality," Quality Progress, DACS 83(2701), (P).
0101	May 1982	Heidler, W., et al, "Software Testing Measures," General Research Corp., NTIS, AD-A113 254 (M).
0161	Aug 1982	Department of the Navy, "Automatic Data Processing Security Program," OPNAVINST 5239.1A, Office of the Chief of Naval Operations, Washington, DC, (R).
0234	Aug 1982	Neugent, W., "Acceptance Criteria for Computer Security," Arlington, VA: AFIPS Press, AFIPS NCC 51, (P).
0089	Nov 1982	Air Force, "Software Operational Test and Evaluation Guidelines," Vol. I, AFOTEC, (R).
0227	Dec 1982	Barber, D. E., "A Guide for Developing an ADP Security Plan for Navy Finance Center," Monterey, CA: Naval Postgraduate School, AD-A127 244, (M).

<u>Index Number</u>	<u>Date</u>	<u>Title</u>
0228	Dec 1982	Helling, W. D., "Computer Security for the Computer Systems Manager," Monterey, CA: Naval Postgraduate School, AD-A126 768, (M).
0134	Dec 1982	Cragon, Harvey G., "The Myth of the Hardware/Software Cost Ratio," <u>Computer, Open Channel</u> , (P).
0123	Autumn 1982	Grove, H. Mark, "DoD Policy for Acquisition of Embedded Computer Resources," <u>CONCEPTS, The Journal of Defense Systems Acquisition Management</u> , Vol. 5, No. 4, Special Issue-Managing Software, (P).
0145	1982	LeBlanc, R., and J. Goda, "Ada and Software Development Support: A New Concept in Language Design," <u>Computer</u> , 15, 5, pp. 75-82, (P).
0074	1982	Crouch, E. A. C., and R. Wilson, <u>Risk/Benefit Analysis</u> , Cambridge, MA: Ballinger, (B).
0081	1982	Fox, V. M., <u>Software and Its Development</u> , Englewood Cliffs, NJ: Prentice-Hall, (B).
0080	1982	Pressman, R. S. <u>Software Engineering: A Practitioner's Approach</u> , New York: McGraw-Hill, (B).
0095	1982	Soi, I. and K. Aggarwal, "Software Reliability and Maintainability: A Life-Cycle Cost Viewpoint," Reliability in Electrical and Electronic Components and Systems, NTIS, (P).
0096	1982	Daniels, B. K., "Software Reliability Assessment," Microprocessors: Safety Implications for Industry, NTIS, (P).
0147	1982	Parikh, G., <u>Techniques of Program and System Maintenance</u> , Cambridge, MA: Winthrop, (B).
0111	1983	Efron, B., <u>The Jackknife, Bootstrap, and Other Resampling Plans</u> , Philadelphia: Society for Industrial and Applied Mathematics, (B).
0148	1982	Thayer, R., A. Pyster, and R. Wood, "Validating Solutions to Major Problems in Software Engineering Project Management," <u>Computer</u> , 15, 8, pp. 65-77, (P).
0140	Jan 1983	Kafura, Dennis, J. A. N. Lee, and Timothy Lindquist, "Validation in Ada Programming Support Environments," Engineering Psychology Group, Office of Naval Research, Working Paper, NRSRO-101, (R).



# THE BDM CORPORATION

<u>Index Number</u>	<u>Date</u>	<u>Title</u>
0011	Feb 1983	Syscon Corporation, "Avionics Software Support Cost Model," AFWAL-TR-82-1173, (P).
0129	Feb 1983	Vessey, Iris and Ron Weber, "Some Factors Affecting Program Repair Maintenance: An Empirical Study," Communications of the ACM, Vol. 26, (P).
0138	Apr 1983	DoD, "Software Technology for Adaptable, Reliable Systems (STARS) Program Strategy," Department of Defense, (R).
0012	May 1983	Syscon Corporation, "Avionics Software Support Cost Model: User's Manual," AFWAL-TR-83-1071, (P).
0127	May 1983	Houghton, Raymond C. Jr., "Software Development Tools: A Profile," Computer, (P).
0226	Jun 1983	Black, M. A., et al, "DoD/DON Requirements for Computer Risk Assessments," Monterey, CA: Naval Postgraduate School, AD-A132 202, (M).
0089	Jul 1983	Air Force, "Software Operational Test and Evaluation Guidelines," Vol. V, AFOTEC, (R).
0166	Jul 1983	Defense Systems Management College, <u>Risk Assessment Techniques</u> , Fort Belvoir, Virginia, (R).
0141	Jul 1983	RADC, Bowen, T., et al, "Software Quality Measurement for Distributed Systems," Rome Air Development Center, Volumes I, II, III, (P).
0030	Aug 1983	Goel, A. L., "A Guidebook for Software Reliability Assessment," DTIC, AD-A139240, (P).
0142	Aug 1983	RADC, Angus, J. E., J. B. Bowen, S. J. VanDenBerg, "Reliability Model Demonstration Study," Volumes I and II, Rome Air Development Center (COEE), RADC-TR-83-207, (M).
0114	Oct 1983	Fisk, F. B. and W. G. Murch, "A Proposal for Computer Resources Risk Assessment During Operational Test and Evaluation," AFOTEC, (P).

<u>Index Number</u>	<u>Date</u>	<u>Title</u>
0132	Nov 1983	Peercy, David E. and Gary E. Swinson, "A Software Support Facility Evaluation Methodology," Symposium on Application and Assessment of Automated Tools for Software Development, (P).
0137	Dec 1983	NBS, Martin R., and W. Osborne, "Guidance on Software Maintenance," NBS Special Publication 500-106, National Bureau of Standards, Institute for Computer Sciences and Technology, (R).
0125	Dec 1983	USAF Scientific Advisory Board, "The High Cost and Risk of Mission-Critical Software," Ad Hoc Committee Report, (R).
0117	1983	Jette, G. E., "Addressing Risk and Uncertainty in Cost Estimating," Wright-Patterson Air Force Base: Aeronautical Systems Division, (M).
0010	1983	Ferens, D. K., "Avionics Software Support Estimating," Wright-Patterson AFB, OH 45433, (P).
0043	1983	Musa, J. D., "Measuring and Managing Software Reliability," IEEE 1983 Phoenix Conference on Computers and Communications, (P).
0063	1983	Gubitz, M. and K. O. Ott, "Quantifying Software Reliability by a Probabilistic Model," NTIS, (P).
0112	1983	Shepard, R. F. and V. I. Young, "Quantitative Techniques for DARPA Program Risk Management," Falls Church, VA: Meridian Corporation, (M).
0071	1983	Rescher, N., <u>Risk</u> , Washington, D.C.: University Press of America (B).
0155	1983	Shooman, M. L., <u>Software Engineering, Design, Reliability, and Management</u> , New York: McGraw-Hill, (B).
0144	1983	Booch, G., <u>Software Engineering With Ada</u> , Reading, MA: Benjamin/Cummings, (B).
0169	1983	Martin, J. and C. McClure, <u>Software Maintenance: The Problem and Its Solution</u> , London: Prentice-Hall International Inc., (B).
0093	1983	Daniels, B. K., "Software Reliability," NTIS (P).

<u>Index Number</u>	<u>Date</u>	<u>Title</u>
0089	Jan 1984	Air Force, "Software Operational Test and Evaluation Guidelines," Vol III, AFOTEC, (R).
0133	Jan 1984	Peercy, David E., "A Framework for Software Maintenance Management Measures," Proceedings of the Seventeenth Annual Hawaii International Conference on System Sciences, (P).
0171	Feb 1984	Yau, S. S., <u>Methodology for Software Maintenance</u> , Rome Air Development Center, Griffis AFB, NY, RADC-TR-83-262, (R).
0124	Mar 1984	IEEE, IEEE Software Working Group P, "IEEE Software Reliability Guide, Second Draft - M58 (Risk Assessment), M59 (Software Functional Test Coverage Index), M60 (Software Maturity Index)," (P).
0162	Mar 1984	Air Force, "OT&E Reporting," Air Force Operational Test and Evaluation Center Regulation 55-1(C2), Chapter 6, (R).
0109	Mar 1984	DoD, "Systems Safety Program Requirements," MIL-STD-882B, (R).
0139	Apr 1984	Lindquist, Timothy E., Jeffery L. Facemire, and Dennie G. Kafura, "A Specification Technique for the Common APSE Interface Set," Office of Naval Research, 84004-R, (R).
0164	May 1984	Leibowitz, S., S. Parratto, D. Peercy, H. Pringle, J. Wiley, and E. Witzke, "Computer System Security (CSS) Literature Review, Current Research Review, and Data Base Assemblage," (INTERIM), BDM/A-84-108-TR, The BDM Corporation, (R).
0170	Jul 1984	DeMillo, R., "A Risk Model for Software Testing," Georgia Institute of Technology, Briefing Slides, (P).
0163	Aug 1984	Parratto, S. L., D. E. Peercy, and H. G. Pringle, "Computer System Security (CSS) Test and Evaluation (T&E) Life-Cycle Process Definition," (FINAL), BDM/A-84-0320-TR, The BDM Corporation, (R).
0167	Aug 1984	Huebner, W., D. Peercy and G. Richardson, "Software Supportability Risk Assessment in OT&E: An Evaluation of Risk Assessment Methodologies," (FINAL), BDM/A-84-496-TR, The BDM Corporation, (R).

<u>Index Number</u>	<u>Date</u>	<u>Title</u>
0235	Aug - 1984	Air Force, "Automatic Data Processing (ADP) Security Policy Procedures and Responsibilities," AFR 205-16, Washington, D.C.: Department of the Air Force, Headquarters, U.S. Air Force, (R).
0168	Sep 1984	Huebner, W., D. Peercy, and G. Richardson, "Software Supportability Risk Assessment in OT&E: Measures for a Risk Assessment Model," (FINAL), BDM/A-84-565-TR, The BDM Corporation, (R).
0100	1984	Air Force, "Software Safety Handbook," (Draft) HQ AFISC/SESD, Norton AFB, CA, (P).
0036	None	Unknown, "Instructions for Using Risk Analysis Matrix," (P).
0058	None	RCA, "Price Parametric Cost Models," RCA/Price Systems, (P).
0061	None	Unknown, "Proposed Methodology for Treating Hardware/Software Failures During OT&E," (P).

**Appendix H**  
**Abstract and Comments Bibliography**

APPENDIX H  
BIBLIOGRAPHY

The bibliographic entries in this appendix are ordered by index number with each entry starting a new page. Index numbers are in order, but are not consecutive because: 1) Bibliographic data were available (from document order lists) and entered into the bibliographic data base before all the abstracts and comments could be written and entered into the corresponding abstract/comment text base, the offset in entry schedules producing a noncorresponding offset in indexing as the one entry process caught up with the other; 2) Functional duplicates (e.g., older editions and slightly altered republications of documents) were deleted, along with their index numbers; and 3) Analysis of some of the documents received revealed that they were not germane to risk assessment (T&E particularly), and were thus deleted, with their index numbers, from the bibliography data base file and abstract/comment text files.

Each entry follows the following format:

BIBLIOGRAPHY INDEX NUMBER  
[AUTHOR(s)], [TITLE], [PUBLISHER, or SOURCE],  
DOCUMENT REFERENCE NUMBER,  
DATE OF PUBLICATION, MEDIA CODE\*

ABSTRACT

COMMENT

(comment when present)

\*Media Codes

R (Bound Report)  
B (Book)  
M (Microfiche)  
P (Loose Paper)

#0003

Wolverton, R. W., "Airborne Systems Software Acquisition Engineering Guidebook for Software Cost Analysis and Estimating," Redondo Beach, CA: TRW Defense and Space Systems Group, Sep 1980, (M).

ABSTRACT:

This guidebook assists Air Force Program Office engineering and management personnel in costing embedded software for avionics applications. A methodology for cost reporting and avoiding the "90 percent complete" syndrome is presented. An annotated bibliography gives the author's personal view of source material relevant to avionics software costing using modern programming practices.

#0004

Meyer, K., "Airborne Systems Software Acquisition Engineering Guidebook for Supportable Airborne Software," DTIC, 1980, (M).

ABSTRACT:

This report is one of a series of guidebooks whose purpose is to assist Air Force Program Office and engineering personnel in the acquisition and engineering of airborne systems software. This guidebook addresses topics relevant to software supportability. It provides guidance for preparation of the Computer Resources Integrated Support Plan (CRISP) and discusses the acquisition of supportable airborne software through review of the development effort.







1.0



1.1



1.25



1.4



1.6



2.0



2.25



2.5



2.8



3.15



3.5



4.0



4.5



5.0



5.6



6.3

#0005

Hecht, H., "Allocation of Resources for Software Reliability," NTIS, 1981, (P).

## ABSTRACT:

Because software accounts for a steadily increasing proportion of the total cost of major projects, and because special efforts to enhance software reliability are a significant contributor to these costs, techniques for a rational allocation of economic resources for software reliability are urgently required. The paper finds that the benefits of current software reliability practices are difficult to quantify. Evaluation by means of execution time based measures of software reliability holds considerable promise. An example of the use of such data for optimal allocation of resources is presented.

#0006

Rowe, W. D., An Anatomy of Risk. New York: J. Wiley and Sons, 1977, (3).

## ABSTRACT:

The purpose of An Anatomy of Risk is to investigate the complexity of the risk concept, to provide dimensions and definitions that encompass and describe the subject of risk, and to address a variety of methods for dealing with the analysis of risk. The book is basically divided into five sections. The first section discusses the nature of risk by giving definitions, evaluation considerations and methods, and examples of decisions. The second section presents factors involved in risk valuation and evaluation. Section three discusses the general problems in both assessing and measuring (quantifying) risk assessment. Section four evaluates societal preferences for risk assessment. Finally, section five provides insight into methodological approaches to risk assessment and how to implement a formal assessment of risks.

#0008

Worm, G. H., "Applied Risk Analysis with Dependence Among Cost Components," Clemson University, Dept. of Industrial Management, 1981, (M).

## ABSTRACT:

The assessment of uncertainties in component costs, a method of combining these uncertainties for determining the total cost uncertainty, and a method of presentation for risk analysis results are discussed in this paper. An extension of the method of statistical risk analysis which uses the Weibul distribution and the method of moments is developed for incorporating covariance between component costs. A computer program is given for implementing the mathematics.

## COMMENT:

This paper is especially useful because it addresses the critical technical issue of combining separately estimated cost components into an overall, total estimate. Specifically, if a cost model estimates probability density functions of cost for two risk drivers, say maintenance requirements and code characteristics, then it is problematic in combining the two estimates into a total estimate. The interdependence among risk components causes mathematical complications in building a total probability density function of cost. The author uses a Weibul distribution (or double exponential distribution) because its properties allow for manipulation when there are covariances among differing components.

#0009

Fisher, Gerald J. and Lt. Col. Eugene P. Gay, "An Approach to Risk Analysis, A Process Review," An AF/SA Technical Note, Jun 81, (P).

## ABSTRACT:

From an academic perspective, risk analysis is a reasonably well defined process. The application of risk analysis to a real-world problem however, is a difficult task with few well-defined approaches. The practical application of risk analysis is hindered by the lack of an adequate framework with which to approach the problem. Without such a systematic approach, it is difficult to provide useful risk information to a decision maker.

The question of risk, and more fundamentally the uncertainties of future events, needs to be examined to identify the potential competitive and inherent risks associated with alternative military force postures. Having a basic understanding of these uncertainties and their consequences is most important in the decision process.

This paper is intended to aid analysts to understand and structure risk problems. It is not meant to be an academic exercise on the statistics of risk, but rather a practical "handbook" which may be used to view a risk problem as a sequence of steps in a process of problem solving.

## COMMENT:

This short note provides an Air Force concept on the generic flow and structure for laying out the risk problem solution. It identifies five basic steps: state problem, establish alternatives, determine risk factors, evaluate risk, and develop a risk-analysis report profile. It is valuable as a high-level summary of the basic risk analysis steps.

#0010

Ferens, D. V., "Avionics Software Support Estimating," Wright-Patterson AFB, OH 45433, 1983, (P).

## ABSTRACT:

Software support costs comprise an increasingly significant portion of avionics system life cycle costs. Estimating these costs has always been difficult, especially during the conceptual, or early design phase of a software program. Under contract to SYSCON Corporation, the Avionics Laboratory has recently acquired the Avionics Software Support Cost Model (ASSCM) to help the laboratory to analyze software support costs. ASSCM is the only software support cost model which is both based on historical Air Force Logistics Command software support cost data and easy to use during the conceptual phase of a software program. This paper discusses important aspects of ASSCM, including a summary of model inputs, outputs, and internal algorithms, and an illustration of how ASSCM can be used for programs outside of the Air Force avionics environment.

#0011

Syscon Corporation, "Avionics Software Support Cost Model," AFWAL-TR-82-1173, 1 Feb 83, (P).

ABSTRACT:

This report describes the work performed to develop the Avionics Software Support Cost Model (ASSCM). ASSCM is an interactive model which projects annual software support costs of various proposed avionics software configurations during the early design phase of system development. It bases cost projections on a unique algorithm designed to use as much historical data as possible. The algorithm also relies on subjective information obtained from a large group of individuals familiar with support software and its costs.



#0012

Syscon Corporation, "Avionics Software Support Cost Model: User's Manual," AFWAL-TR-83-1071, May. 83, (P).

ABSTRACT:

This manual describes the procedures for running the Avionics Software Support Cost Model (ASSCM) on a computer. This manual is geared toward use on the VAX and CYBER 175 computers at Wright-Patterson AFB, Ohio. However, the general procedures should be useful for any computer on which ASSCM may be hosted.

#0013

Apostolakis, G., "Bayesian Methods in Risk Assessment," Advances in Nuclear Science and Technology, New York: Plenum, 1981, (8).

## ABSTRACT:

Bayesian methods provide a logical framework for risk analysis. By making the use of judgment visible and explicit, we hope they can contribute to the decision-making and consensus-building process for which the risk analysis is performed in the first place. The reason that the word "hope" is used, is that the theory of probability (as well as Decision Theory) are tools for a single analyst and not for groups of analysts. However, the chances that coherent assessors will reach a common decision are much higher than when the assessors are not coherent.

#0015

Army, "Compendium on Risk Analysis Techniques," U.S. Army Material Systems Analysis Agency: Aberdeen Proving Ground, MD, 1972, (M).

## ABSTRACT:

The evolution of risk analysis in the material acquisition process is traced from the Secretary Packard memorandum to current AMC guidance. Risk analysis is defined and many of the existing techniques are described in light of this definition with respect to their specific role in program management and systems analysis. In particular, techniques using subjective judgement data are explained and critiqued. Several choice-between-gambles techniques, a standard lottery, the modified Churchman-Ackoff technique, the Delphi technique, Monte Carlo methods, network analysis, PERT, RISCA, and Bayesian techniques are discussed.

## COMMENT:

This compendium provides a fairly extensive survey of methods using subjective data bases. The monograph provides a summary of each technique's advantages as well as limitations. Unfortunately, the monograph is now fairly dated.

#0016

Whitmore, D. C., et al, "Computer Program Maintenance," Boeing Aerospace Co.; NTIS, AD-A083 209/7, Dec 77, (M).

## ABSTRACT:

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 the software maintenance life cycle, including maintainability, maintenance tasks and required maintenance resources.

## COMMENT:

This guidebook describes the software maintenance life cycle; maintainability attributes; detailed planning and maintenance tasks; and required resources. Responsibilities of the software acquisition engineer and development contractor are identified. The ground systems under specific consideration are training simulators and automatic test equipment.

#0018

Systems Architects, "Computer Systems Acquisition Metrics," Vols I-II, Systems Architects Inc., DTIC, AD-A120375, May 1982, (M).

## ABSTRACT:

This handbook contains a standard set of procedures to quantitatively specify and measure the quality of a computer software system during its acquisition life cycle. These quantitative measures, or metrics, provide the user with a tool to better assess the system's development and potential performance throughout the acquisition phases.

The metrics are calculated from the answers to questions, called data elements in this handbook, which also serve as a checklist to aid Software Quality Assurance. These metrics are a tool for current Software Quality Assurance practices. They are an added feature to current tools and techniques utilized in Software Quality Assurance practices.

The handbook is tailored specifically to address embedded Command Control and Communications (C3) computer systems. Efforts to apply the procedures to other than C3 systems may require reworking by the user of the materials contained in the handbook.

#0020

Durall, Lorraine; et al, "Data Needs for Software Reliability Modeling," DACS 82 (1793), 1980, (P).

ABSTRACT:

This paper summarizes the results of a study to determine the data requirements for software reliability modeling. The major assumptions of the models are presented along with a brief description of their uses and the data needed to exercise the models. Methodologies for evaluating failure databases are presented including a sample evaluation to determine the adequacy of the data to do comparisons across a wide variety of projects and to determine if the database contains data elements as required by the various models.

#0024

Watson, G., "Evaluation of Computer Software in an Operational Environment," Center for Naval Analysis, Alexandria, VA, NTIS, AD-A091 213/9, Aug 80, (M).

## ABSTRACT:

This paper examines general procedures for testing military realtime operational software from the user's perspective. A summary of industrial software testing is given with an evaluation of its applicability to the military's requirement for operational testing. The operational test environment is examined to determine the extent of verification, validation or certification of computer software that is possible given the constraints of this environment.

#0025

Reynolds, John H., "Evaluation of Contemporary Software Engineering Techniques for a Large FORTRAN Simulation," DACS 83 (2401), 1980, (P).

## ABSTRACT:

Those software-engineering/structured-programming techniques designed to detect software errors early and facilitate coding, validation, and maintenance were applied in developing the Trident Computational Simulation (TRICS) at the Naval Surface Weapons Center (NSWC) in Dahlgren, Virginia. This continuous simulation permits validation of the fire control computations required to determine missile presets prior to launch from a Trident submarine. In addition, it permits dynamic (runtime) connectivity of computational subsets (no core penalty for unused subsets) for the purposes of research and experimentation.

In the past, development of applications models at NSWC has depended upon the skills and intuition of individual programmers applying their favorite and immutable ad hoc methods. On the other hand, TRICS was developed by a team of programmers applying an independent and established methodology. This was supplemented by a stringent set of programming and documentation standards as well as in-house tools for automating and managing frequently occurring programming activities. The end result was a highly flexible and user-oriented simulation.

The tools and techniques used are identified, and an evaluation of their effectiveness is presented by examining error data collected during the development cycle.



#0027

Thibodeau, R., "The Feasibility of Obtaining Software Research Data at the U.S. Army Computer Systems Command," General Research Corporation, NTIS, AD-A107-883, 15 Jul 80, (M).

## ABSTRACT:

It is possible for a relatively small cost in personnel time to obtain data for software engineering and computer science research as a by-product of existing USACSC reporting practices. These data when manipulated by automated systems which already exist can provide many of the data elements describing the computer systems built at the Command, the resources required to complete them, and the development and maintenance environment. These three aspects of the software development process are the principal components of any research data structure.

Software product data, which include measures of size, type, and complexity are best obtained from the programs themselves. This can be accomplished by making copies of released systems. Reliability data can be obtained from a modified Incident Report. In both cases, however, and to obtain data describing the system documentation, it will be necessary to use supplementary data collection instruments.

#0028

Vemuri, V., "Figures of Merit for Software Quality," DACS 83 (2598), 1980, (P).

## ABSTRACT:

Software and its development are complex. The complexity stems from a multiplicity of objectives and attributes that one has to work with during its development. Human comprehension of multiple objectives and attributes can be aided by displaying the relevant data on a two-dimensional plane. Several display techniques, and in particular the so-called snowflakes and Chernoff faces, are discussed and their utility in software research explored. Examples using real and hypothetical data are presented to illustrate the suitability of these pictures.

#0030

Goel, A. L., "A Guidebook for Software Reliability Assessment," DTIC, AD-A139240 Aug 83, (P).

ABSTRACT:

The purpose of this guidebook is to provide state-of-the-art information about the selection and use of existing software reliability models. Towards this objective, we have presented a brief summary of the available models backed by a detailed discussion of most of the models in the appendices.

One of the difficulties in choosing a model is to find a match between the testing environment and a class of models. To help a user in this process, we have presented a detailed discussion of most of the assumptions that characterize the various software reliability models.

The process of developing a model has been explained in detail and illustrated via numerical examples.

#0033

Littlewood, B., "How to Measure Software Reliability and How Not To," IEEE Transactions on Reliability, Vol R-28, No. 2, NTIS, Jun 1979, (M).

## ABSTRACT:

The paper criticizes the underlying assumptions that have been made in much early modeling of computer software reliability. The following suggestions will improve modeling:

- (1) Do not apply hardware techniques to software without thinking carefully. Software differs from hardware in important respects; we ignore these at our peril. In particular--
- (2) Do not use MTF, MTBF for software, unless certain that they exist. Even then, remember that--
- (3) Distributions are always more informative than moments or parameters; so try to avoid commitment to a single measure of reliability. Anyway--
- (4) There are better measures than MTF. Percentiles and failure rates are more intuitively appealing than means.
- (5) Software reliability means operational reliability. Who cares how many bugs are in a program? We should be concerned with their effect on its operation. In fact--
- (6) Bug identification (and elimination) should be separated from reliability measurement, if only to ensure that the measurers do not have a vested interest in getting good results.
- (7) Use a Bayesian approach and do not be afraid to be subjective. All our statements will ultimately be about our beliefs in the quality of programs.
- (8) Do not stop at a reliability analysis; try to model lifetime utility (or cost) of programs.
- (9) Now is the time to devote effort to structural models.
- (10) Structure should be of a kind appropriate to software, e.g., top-down modular.

#0035

Megill, R. E., An Introduction to Risk Analysis, Tulsa: Petroleum Publishing Co., 1977, (B).

## ABSTRACT:

This book is a fundamental treatment of risk analysis as it is applied to the petroleum industry. The first several chapters lay the groundwork for risk analysis. Chapters 1-4 discuss varying statistical distributions. Specifically, histograms, the binomial, normal, and log-normal distributions are addressed with respect to the characteristics and mathematics that describe each. The middle chapters of the book describe the concept of "Gambler's Ruin." This concept explains what is meant by a normal run of bad luck. Next, triangular distributions are illustrated by the author. In the final chapter, a review of the steps of risk analysis are given.

## COMMENT:

The book provides a statistical approach to risk analysis. The various distributions that are used in risk analysis are detailed well. The final chapter is especially helpful as it lists in detail seven fundamentals of risk analysis. In brief, these are:

- Isolate key variables
- Quantify key variables
- Make distributional assumptions
- Understand your model
- Put estimates of probability into key variables before simulation of model
- Search for reality checks
- Express uncertainty as a probability density distribution.

#0036

Unknown, "Instructions for Using Risk Analysis Matrix," (P)

ABSTRACT:

This paper is an example of using a matrix to provide an overall risk potential in the use of the Ada programming language. This matrix is designed to allow one to: (1) estimate a "success probability" for each parameter in the Ada risk analysis, and (2) assign weightings to the parameters consistent with the requirements of the software element being considered. The products of the weights and ratings are then summed to provide an overall rating.

#0037

Gross, R. N., "Issues and Perspectives in the Validation of Tactical Software," Naval Ocean Systems Center, NOSC/TD-139, NTIS, AD-A056 061/5ST, 1 Feb 78, (M).

ABSTRACT:

This report represents some of the results obtained under Project Z291 of the NOSC In-House Research and Development program. The title of the project, "Command Control Distributed System Design and Validation Processes," suggests that the task embraces two separate efforts, and indeed this is the case. This document deals only with validation processes; the results on system design are reported elsewhere.

#0038

Lientz, B. "Issues in Software Maintenance and Measurement," UCLA Graduate School of Management, Los Angeles, NTIS, AD-A098 982/2, May 81, (M).

## ABSTRACT:

Up to a few years ago the area of software maintenance was largely ignored. Interest has increased in the last few years due to several factors. First, the increased burden of maintenance from that of ten years ago has restricted resources available for new development. Second, there has been a growing awareness that considering tools which assist development may have little effect on operational systems. This article discusses these issues and proposes solutions.



#0039

Ramamoorthy, C. V. and S. L. Ganesh, "Issues in Software Reliability," Symposium on Reliability in Distributed Software and Database Systems, 113-116, NTIS, 1981, (P).

## ABSTRACT:

It is important to ensure that computer systems for critical real-time applications are sufficiently reliable. This requirement encompasses the need both to ensure the correctness of the design of the combined hardware-software system before it is put into operation and to secure the system from deterioration in the operational phase as the system is patched and augmented and hardware parts wear out. The complexity of many software systems makes the fulfillment of these requirements onerous. Formal proofs of correctness for software are usually lengthy and not completely convincing. Therefore, testing procedures and reliability models are required. We introduce and classify the models that have been proposed in the literature. We also discuss methods for comparing the adequacy of the testing methods used. The need for research on integrated hardware-software reliability models is discussed. Such models will be required in order to derive good reliability estimates of systems in which redundancy of hardware and software is exploited for fault-tolerance, e.g., distributed systems.

#0041

Vorgang, B. R., "A Macro Approach to Software Resource Estimation and Life Cycle Control," M.A. Thesis, Naval Postgraduate School, 1981, (M).

## ABSTRACT:

Planning and controlling the software development process has shown, in the past, to be an extremely difficult task. The estimation of resource requirements, development costs, risk profiles and project feasibility has often proven to be inaccurate, thus costing the government time and dollars. However, by using obtainable management parameters, and simple engineering and operations research techniques, estimating can be done easily and accurately by taking a macro approach to the estimation problem.

This study will present the background and mathematical basis for a software cost estimation model. In addition, an example of an automated application of the model will be presented and discussed.

#0042

Walker, M. G., "Managing Software Reliability, The Paradigmatic Approach," New York: Elsevier North Holland, NTIS, 1981, (M).

## ABSTRACT:

The purpose of the Paradigmatic Approach is to provide a new image for conceptualizing the software development cycle. It is believed that this new image will endanger methodologies that predictably produce reliable software systems. This text is not a cookbook of techniques. It does not attempt to direct action through prescribing specific behavior patterns. This text does, however, present an integrated image for organizing behavior and an universal metric for evaluating that behavior. The reader will be exposed to a powerful image, a paradigm, which provides an integrated perception of software development. This paradigm will help him to organize and judge technical behavior, in a consistent and productive manner. The consistent behaviors which result from paradigmatic thinking are termed "the paradigmatic approach" and will facilitate the evolution of software management from a craft to an engineering discipline.

#0043

Musa, J. D., "Measuring and Managing Software Reliability," IEEE 1983 Phoenix Conference on Computers and Communications, 1983, (P).

## ABSTRACT:

The quantification of software reliability is needed for the planning and management of projects involving computer programs. This paper summarizes a theory of software reliability based on execution or CPU time, and a concomitant model of the testing and debugging process that permits execution time to be related to calendar time. The estimation of parameters of the model is discussed. Application of the theory is described, using actual data.

#0046

Jelinski Z., P. B. Moranda, and J. B. Churchwell, "Metrics of Software Quality," NTIS, AD-A093 788, Nov 80, (M).

## ABSTRACT:

This report covers the period from 1 June 1977 to 30 October 1980. A major task on this contract was to make a comprehensive review of the literature on software metrics and of quantitative measures of program testing. The original review is contained in the first Interim Report (MDC G7517, dated July 1978); this review has been slightly revised and updated in this report.

In the related topics of Software Reliability, two methods of estimating the residual error content of an entire program on the basis of data obtained in the testing of portions of it have been developed and are detailed here.

#0048

Mohanty, Siba N., "Models and Measurements for Quality Assessment of Software," Computing Surveys, Vol II, No. 3, DACS 82 (1673), Sep 1979, (P).

## ABSTRACT:

Several software quality assessment methods that span the software life cycle are discussed. The quality of a system design can be estimated by measuring the system entropy function or the system work function. The quality improvement due to reconfiguration can be determined by calculating system entropy loading measures. Software science and Zipf's law are shown to be useful for estimating program length and implementation time. Deterministic and statistical methods are presented for predicting the number of errors. Testing theory is useful in planning the program test process; as discussed in this paper, it includes measurement of program structural characteristics to determine test effectiveness and test planning. Statistical models for estimating software reliability are also discussed.

#0049

Goel, Amrit L., "Models for Hardware-Software System Operational Performance Evaluation," IEEE Transactions on Reliability, Vol R-30, No. 3, DACS 83 (2606), 1981, (P).

## ABSTRACT:

Stochastic models for hardware-software systems are developed and used to study their performance as a function of hardware-software failure and maintenance rates. Expressions are derived for the distribution of time to a specified number of software errors, system occupancy probabilities, system reliability, availability, and average availability. The behavior of these measures is investigated via numerical examples.

#0052

Thompson, W. E. and P. O. Chelson, "On the Specification and Testing of Software Reliability," Proceedings, Annual Reliability and Maintainability Symposium, 1980, (P).

## ABSTRACT:

This paper deals with the statistics of estimating the software reliability of complex real-time systems where an electronic digital computer and associated computer programs are essential elements of system design and function. Testing is conducted in the operating environment or a simulated environment related to the operating environment in some known way. The procedure is Bayesian so that improvement of reliability estimation is realized in a formal and convenient way as more and more test data are accumulated. The method provides for estimating: (1) both hardware and software components of total system reliability, and (2) Bayesian interval limits using existing analytic techniques developed by the authors and others. The results apply to measurement and prediction of reliability performance, to acceptance testing, and to contractual definition and implementation of software warranty provisions for embedded computer systems.

The Bayesian method of software-hardware reliability estimation presented here exhibits the following unique features:

- (1) The use of a prior  $p$  on the probability that the software contains errors. This prior is updated as test failure data are accumulated. Only a  $p$  of 1 (software known to contain errors) corresponds to a case already treated in the literature.
- (2) Hardware, software, and unknown/ambiguous source failure data are combined to yield a system reliability estimation.
- (3) A decision-rule treatment is developed for the continuation or termination of testing on the basis of specification of consumer and producer risks and observed test results.



#0053

Watson, S. R., "On Risks and Acceptability," NTIS, 82-09 07140, 1981, (P).

## ABSTRACT:

A very attractive notion is that it should be possible not only to determine how much risk is associated with any particular activity, but also to determine if that risk is acceptable. Stated bodily, this seems an entirely unobjectionable and indeed a very acceptable notion. There is, however, underlying this idea, a mistaken view of risk which we might refer to as the "phlogiston" theory of risk. In this paper, presented at the SRP meeting on Ethical and Legal Aspects of Radiological Protection, the phlogiston theory of risk is described; secondly, it will be argued that it is too simple a theory to be realistic or useful; and thirdly, the management of risk will be placed in a wider decision framework. Acceptability, it will be argued is a highly dependent on context, and it is not possible, therefore, to lay down generally applicable notions of acceptability.

#0056

Butler, M., "Portability and the National Energy Software Center," Argonne National Lab, NTIS, CONF-781052-1, 1978, (M).

## ABSTRACT:

The software portability problem is examined from the viewpoint of experience gained in the operation of a software exchange and information center. First, the factors contributing to the program interchange to date are identified, then major problem areas remaining are noted. The import of the development of programming language and documentation standards is noted, and the program packaging procedures and dissemination practices employed by the Center to facilitate successful software transport are described. Organization, or installation, dependencies of the computing environment, often hidden from the program author, and data interchange complexities are seen as today's primary issues with dedicated processors and network communications offering an alternative solution.

#0053

RCA, "Price Parametric Cost Models," RCA/Price Systems, (P).

## ABSTRACT:

"PRICE" is a family of Cost Estimating Models. The name PRICE is an acronym for "Programmed Review of Information for Costing and Evaluation." The PRICE Model estimates development and production costs for proposed electromechanical devices and systems. PRICE was developed at RCA over the last fifteen years and has been available for general use outside of RCA since August 1975.

"PRICE L," the PRICE Life cycle cost Model, is a supplement to and operates in conjunction with the basic PRICE Model to rapidly estimate support costs for a variety of systems.

"PRICE S," the PRICE Software Model, applies the PRICE parametric modeling methods to the problems of computer software costing. It is designed to cover the complete range of systems and applications programming.

"PRICE SL," the PRICE Software Life Cycle Cost Model, is a supplement to and operates in conjunction with the PRICE S Model to rapidly estimate software support costs.

"PRICE A," the PRICE Activity Distribution Model, is designed to support management planning and budgeting by providing projections of time dependent resource requirements.

All PRICE Models are exercised interactively through commercial time-sharing computer networks. Users attend comprehensive training courses at RCA, after which they operate the models from their own location, under strict computer security procedures.

#0061

Unknown, "Proposed Methodology for Treating Hardware/Software Failures During OT&E," (P).

## ABSTRACT:

This paper claims that a proper test plan must address software failures and specify exactly how they will be treated; ie., included in the mean time between maintenance (MTBM)/mean time between critical failure (MTBCF) calculations or not. The plan must also specify that sufficient data be collected to identify software failures and track them through correction and verification during subsequent testing. Time to failure data must be collected for software failures. The method used to project reliability to maturity should be discussed as well as how software failures will be treated in the projection. Two examples are given.

#0062

Mendis, Kenneth S., "Quantifying Software Quality," Quality Progress, JACS 83(2701), May 1982, (P).

## ABSTRACT:

This paper puts forth a methodology for predicting software quality. The first task in quantifying software errors is to classify the data into their types and distribution of programming errors. This is easily accomplished if the errors are grouped into seven major categories. These categories are:

- requirements design change
- software design error
- keypunch/coding and handling
- secondary fault
- maintenance/operator induced
- documentation error
- other.

A reliability model is then proposed so that predicted and actual errors in software can be compared.

#0063

Gubitz, M. and K. O. Ott, "Quantifying Software Reliability by a Probabilistic Model," NTIS, 1983 (P).

## ABSTRACT:

A method based on isotonic regression analysis of the software failure statistics is presented. The basic information needed for this analysis are the execution times between failures during the test period. The method allows an evaluation of the software reliability in terms of the combined rate of the residual potential failures for which a statistical upper limit is obtained. It also gives indications of the extent to which further testing may be rewarding and a rough estimate of the time needed for further testing in order to achieve some set reliability level.

The Isotonic Regression Analysis (IRA) method has been applied to three examples: the testing of single module, of a system comprised of a number of modules, and of the practical application of a system, in an operational environment. The analysis is completed with null hypothesis tests of the statistical significance of the improvement of the software reliability indicated by the IRA procedure.

#0066

Koch, H. S. and P. Kubat, "Quick and Simple Procedures to Assess Software Reliability and Facilitate Project Management," The Journal of Systems and Software, 1981, (P).

## ABSTRACT:

A software reliability model is considered that is easy to implement, use, and interpret. The model works extremely well in the latter stages of testing. A complete history of failures does not need to be stored in a data base or maintained. This reduces the cost of assessing software reliability. Furthermore, it is possible to use the model to estimate software reliability when failure statistics have not been extensively collected. Various estimation procedures are discussed that can aid in project planning. The use of these estimation procedures is illustrated through two sets of actual failure data.

#0068

Wessels, E., "Rating Techniques for Risk Assessment," NTIS, 81-02 00219, Mar 80, (P).

ABSTRACT:

This paper addresses risk in terms of fire hazard and fire safety. Of particular importance to this paper is the calculation of fire insurance premiums and costs. The paper in general offers little in the way of solving the risk assessment problem for software supportability of AFOTEC.



#0069

Coppola, A. and A. Sukert, "Reliability and Maintainability Management Manual," Rome Air Development Center, RADC-TR-79-200, Jul 79, (M).

ABSTRACT:

This manual provides a guide to Air Force program managers, at all levels, for the planning, organizing, manning, leading and controlling of cost-effective reliability and maintainability programs in all phases of acquisition. It addresses both hardware and software reliability.

#0070

Thacker, J. and F. Ovadia, "Reliability Measurement for Operational Avionics Software," NTIS, Sep 79, (M).

## ABSTRACT:

This study was conducted to determine quantitative measures of reliability for operational software in embedded avionics computer systems. Analysis was carried out on data collected during flight testing and from both static and dynamic simulation testing. Failure rate was found to be a useful statistic for estimating software quality and recognizing reliability trends during the operational phase of software development. The scope of the analysis was limited due to insufficient environment where adequate maintenance and service records for avionics systems are kept.

#0071

Rescher, N., Risk, Washington, D.C.: University Press of America, 1983, (B).

## ABSTRACT:

Rescher presents the topic of risk from a philosophical point of view. This perspective enables the author to develop a very fundamental definition of risk. Risk is defined as the chancing of a negative outcome. Further, risk is broken down into two major parts: a negative outcome and the chance of the outcome's realization. The negative outcome further is broken down into the components of character, extent, and timing. Character describes the actual nature of the negative outcome such as negative performance or cost overruns. Extent has two additional subdivisions: severity and distribution. Severity asks the question of how much while distribution asks who's affected and who's involved. Timing deals with the questions of how often and what's the duration. The author proposes that risk description--characterizing the nature, intensity, diffusion, and probability of risks--is a factual, scientific exercise involving matters of observation, theorizing, and inductive extrapolation from experience. On the other hand, risk assessment is a matter of the appraisal and measurement of the negative outcomes. Assessment involves evaluative questions such as how serious and how significant.

## COMMENT:

The strength of this book is its fundamental view of risk. The detailed definition of risk provides a good working framework for risk description and risk assessment.

#0074

Crouch, E. A. C. and R. Wilson, Risk/Benefit Analysis, Cambridge, MA: Ballinger, 1982, (8).

## ABSTRACT:

Although the book is titled Risk/Benefit Analysis, the major emphasis of this book is on the more restricted field of risk assessment. The word "analysis" is used to describe the whole process of considering risks, including the making of decisions. A risk assessor is a person who organizes data in such a way that others can make decisions more reliable.

The major topics of the book are: perspectives on risk, the meaning of risk, the estimation of risk, the perception of risk, the comparison of risks and benefits, managing and reducing risks, and several useful case studies of risk analysis. Chapter 3 on the estimation of risk is particularly useful. In general, this chapter describes how the risk analyst decides which measures of risk to use and within which boundaries to use them.

## COMMENT:

This book provides many salient points on risk assessment. In particular, the general method of parametric modeling is discussed thoroughly. Topics such as the use of proxy variables and other modeling concerns are addressed. The recency of the book also adds to its importance.

#0075

Pariseau, R. J., "A Screening Criterion for Delivered Source in Military Software," Vol. I & II, Warminster PA: Naval Air Development Center, NTIS, 14 Nov 1979, (M).

ABSTRACT:

The goal of this study is to identify measurable characteristics of the program source code that indicate the likelihood of future changes to the program modules. These changes include both repair of software errors and improvement in software performance.

Source code data and module change data were analyzed to correlate the source code characteristics with the number of changes made to the modules.

#0076

Yau, S., "Self-Metric Software," Vol. I, U. Northwestern, NTIS, AD-A086 290/4, Apr 1980, (M).

## ABSTRACT:

This report documents the research performed under RADC Contract F30602-76-CO-0397 by Northwestern University in the area of developing effective techniques for large-scale software maintenance, including those for the design, implementation, validation, and evaluation of reliable and maintainable software systems with a high degree of automation. During this contract period, research in the areas of ripple effect analysis, testing during software maintenance, specification for program modifications, quality factors for software maintainability, and dynamic monitoring of program behavior was conducted. In this report, the software maintenance process is first described. The research results which have been presented in previous papers and interim technical reports are summarized, and unfinished work is presented.

#0078

Bratman, H. and M. Finfer, "Software Acquisition Management Guidebook Verification," System Development Corporation, SDC-TM-5772/002/02, NTIS, AD-A048 577/1ST, Aug 1977, (M).

## ABSTRACT:

This report is one of a series of software acquisition management guidelines which provide information and guidance for ESD program office personnel who are charged with planning and managing the acquisition of command, control, and communications system software procured under Air Force 800 series regulations and related software acquisition management concepts. It provides a review of the software verification practices and procedures employed by industry and set forth in relevant DoD and Air Force regulations, specifications, and standards. It specifically: defines verification; describes the software related planning, system engineering, and testing activities, carried out by the Program Office and the contractor, which lead to Computer Program Configuration Item (CPCI) verification; and references specific software techniques and tools required to CPCI verification.

#0079

Kress, M. P., "Software Configuration Management," Boeing Aerospace Co., NTIS, AD-A083, 2 Jan 1979, (M).

ABSTRACT:

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 provides guidance in the preparation, imposition and enforcement of software configuration management requirements and recommended procedures.



#0080

Pressman, R. S., Software Engineering: A Practitioner's Approach, New York: McGraw-Hill, 1982, (B).

## ABSTRACT:

The contents of this book closely parallel the software life cycle. Early chapters present the planning phase, emphasizing system definition (computer systems engineering), software planning, and software requirements analysis. Specific techniques for software costs and schedule estimation should be of particular interest to project managers as well as to technical practitioners and students.

In subsequent chapters, emphasis shifts to the software development phase. The fundamental principles of software design are introduced. In addition, descriptions of two important classes of software design methodology are presented in detail. A variety of software tools are discussed. Comparisons among techniques and among tools are provided to assist the practitioner and student alike. Coding style is also stressed in the context of the software engineering process.

The concluding chapters deal with software testing techniques, reliability, and software maintenance. Software engineering steps associated with testing are described and specific techniques for software testing are presented. The current status of software reliability prediction is discussed and an overview of reliability models and program correctness approaches is presented. The concluding chapter considers both management and technical aspects of software maintenance.

#0081

Fox, V. M., Software and Its Development, Englewood Cliffs, NJ: Prentice-Hall, 1982, (B).

## ABSTRACT:

This book is about software, about the development of software, and primarily about the development of large scale software.

The first part of the text is devoted to setting the stage for ideas on software development. In the first part, the author gives definitions, sets meanings, and makes distinctions. The bulk of the remaining text is on the development process. Specific topics discussed include: program attributes, requirements definition, conflicting requirements of multiple users, product versus project requirements, the parts and process of design, levels of design, construction, verification and testing, documentation, and traceability.

#0084

Osterweil, L., "A Software Lifecycle Methodology and Tool Support," Colorado University Department of Computer Science, CU-CS-154-79, NTIS, AD-A076 335/9, Apr 1979, (M).

## ABSTRACT:

This paper describes a system of techniques and tools for aiding in the development and maintenance of software. Improved verification techniques are applied throughout the entire process and management visibility is greatly enhanced. The paper discusses the critical need for improving upon past and present methodology. It presents a proposal for a new production methodology, a verification methodology, and the system architecture for a family of support tools.

#0085

Glass, Robert L. and Ronald A. Noiseux, Software Maintenance Guidebook, DACS 83(2948), 1981, (B).

## ABSTRACT:

This book provides information on software maintenance from three points of view: people, technical, and management. Discussed first is the way software maintenance fits into the software life cycle, and a definition of software maintenance and its types. The subject then moves to people--a personality profile of software maintainers, different programming styles, and the goals and priorities of software maintenance. Next the authors discuss the technologies available to the maintainer in terms of tools and techniques which maintainers know or should be aware of. Many examples in the Ada programming language are supplied. Finally the authors discuss how one plans, organizes, and directs software maintenance from a management perspective.

#0086

Schneidewind, N. F., "Software Maintenance: Improvement Through Better Development Standards," Naval Postgraduate School, NPS-54-82-002, NTIS, AD-A113 257/0, 22 Feb 1982, (M).

## ABSTRACT:

Software maintenance is frequently the most expensive phase of the software life cycle. It is also the phase which has received insufficient attention by management and software developers. Software standards have improved the ability of the software community to develop and design software. Unfortunately, most standards do not deal with the maintenance phase in a substantive way. Since maintainability has to be designed into the software and cannot be achieved after the software is delivered, it is necessary to have software standards which explicitly incorporate requirements for maintainability. Accordingly, this report suggests design criteria for achieving maintainability and evaluates Weapons Specification WS 8506 and MIL-STD 1679 against these criteria. Using these documents as typical examples of military software standards, recommendations are made for improving the maintainability aspects of software standards.

#0088

Markham, D., J. McCall and G. Walters, "Software Metrics Application Techniques," DACS 83(3005), 1981, (P).

ABSTRACT:

The purpose of this paper is to review current research and application methodologies of software metrics. The intent of this review is to briefly cover the theoretical foundations of metrics, their current modes of application and future plans to use metrics in the software life cycle. This survey is not exhaustive but touches upon recent field experiences with the software metrics technology.

This research and application has been sponsored in part by the Air Force Systems Command Electronic Systems Division, Rome Air Development Center, and the U.S. Army Computer Systems Command Army Institute in Management Information and Computer Science.

#0089

Air Force, "Software Operational Test and Evaluation Guidelines," Vol. I, 10 Nov 1982, Vol. III, 1 Jan 1984, Vol. V, 25 Jul 1983, AFOTEC, (R).

## ABSTRACT:

Volume I, Software Test Manager's Guide


This pamphlet is a guide for Headquarters (HQ) Air Force Operational Test and Evaluation (AFOTEC) software test managers. It documents techniques and information "learned the hard way" but not necessarily passed on to all succeeding software test managers. HQ AFOTEC software test managers should not view this document as a directive, but rather as a source of information about operational test and evaluation (OT&E) of software and as a reference document to be used in planning for OT&E. Although this pamphlet is primarily for HQ AFOTEC Software Evaluation Division personnel, individuals from other organizations will find in it a description of the AFOTEC approach to OT&E of software.

This pamphlet is divided into three chapters.

- (1) Chapter 1 provides general information on OT&E, AFOTEC organization, and the OT&E process--all with a focus on software evaluation and the software test manager.
- (2) Chapter 2 contains a description of the OT&E environment within which the software test manager must function: directives and regulations.
- (3) Chapter 3 contains general instructions and information on the use of various software evaluation tools available to the software test manager, including the software maintainability evaluation questionnaire, the Software Operation-Machine Interface Questionnaire (SOMIQ), the AFOTEC software support evaluation tool (ASSET), and the event trace monitor. Along with the general instructions, references are given for more detailed information. The chapter also contains lessons learned from the efforts of software test managers on earlier programs.

Volume II, Guide for the Deputy for Software Evaluation

This guide provides general information, software OT&E concerns and techniques, and software evaluation lessons learned. Elements of OT&E for embedded computer systems are provided, including software suitability evaluation. Software effectiveness consideration encompasses software performance, software/operator interface, software maturity evaluation, and embedded computer system peculiar evaluations.


  
Volume III, Software Maintainability-Evaluator's Guide

The purpose of this pamphlet is to provide the software evaluator the information needed to participate in the Air Force Operational Test and Evaluation Center's (AFOTEC's) software maintainability evaluation process. In this pamphlet, "software maintainability" is limited in scope to software design and documentation assessments.

This pamphlet provides the evaluator with:

- (1) A background of the AFOTEC software maintainability evaluation concept.
- (2) A basic understanding of the evaluation procedures.
- (3) Detailed instructions for using AFOTEC's standard software maintainability questionnaires and answer sheets.

In addition, the pamphlet contains the questionnaires and explanatory information on each question.

Volume V, Software Support Facility Evaluation-User's Guide  

This document describes the method and procedures used by AFOTEC for evaluating the software support resources (SSR) for an embedded computer system (ECS). Evaluation of the SSR capabilities provides an assessment of the ECS's supportability. The SSR evaluation is supported by an automated process called the AFOTEC Software Support Evaluation Tool (ASSET).

This guide is divided into the following:

- (1) Chapter 1 provides general information about the evaluation methodology and the responsibilities of the different personnel involved in the evaluation.
- (2) Chapter 2 provides guidance for the Headquarters (HQ) AFOTEC software test manager and OT&E test team deputy for software evaluation in planning and conducting the SSR evaluation.
- (3) Chapter 3 gives guidance for the software evaluation members of the OT&E test team to support the SSR evaluation.



#0092

Walters, Gene F. and J. A. McCall, "Software Quality Metrics for Life-Cycle Cost-Reduction," IEEE Transactions on Reliability, Vol R-28, No. 3, DACS 82(1679), Aug 1979, (P).

## ABSTRACT:

This paper identifies factors or characteristics of which reliability is one, which comprise the quality of computer software. It then discusses their impact over the life of a software product, and describes a methodology for specifying them quantitatively, including them in system design, and measuring them during development. The methodology is still experimental, but is rapidly evolving toward application to all types of software. This paper emphasizes those factors of software quality which have greatest importance at the later stages of a software product's life.

#0093

Daniels, B. K., "Software Reliability," NTIS, 1983, (P).

## ABSTRACT:

The reliability of computer software has been causing concern for at least 15 years. The achievement of accurate software has been the goal of many workers, who identified the design process as the main source of software faults. This has led to the development of a number of design methodologies which aim to reduce the propagation of design phase errors.

The measurement of software reliability has also received considerable attention. A number of stochastic models have been developed and tested against observed software system failure data. A small number of models are being used to monitor the reliability performance of software systems as they progress through the various phases of the software life cycle.

This paper reviews reliability analysis techniques developed for hardware dominated systems. The inputs to a software reliability analysis are considered and progress in developing a methodology to assess computer system software is described.

#0094

Thayer, T. A., et al, "Software Reliability: A Study of Large Project Reality," New York: Elsevier North-Holland, NTIS, 1978, (M).

## ABSTRACT:

This document is the final technical report for the Software Reliability Study, performed by TRW for the Rome Air Development Center. It presents results of a study of data, principally error data, collected from four software development projects. These data were analyzed to determine what might be learned about various types of errors in the software; the effectiveness of the development and test strategies in preventing and detecting errors, respectively; and the reliability of the software itself.

This report also provides guidelines for data collection and analysis on other projects: data that are generally available, how project data were collected in this study, and some observed realities concerning the data collection and analysis processes.

Finally, the most recent work on TRW's Mathematical Theory of Software Reliability (MTR), the Nelson model, is presented. This is complemented by a survey of software reliability models currently available in the software community.

#0095

Soi, I. and K. Aggarwal, "Software Reliability and Maintainability: A Life-Cycle Cost Viewpoint," Reliability in Electrical and Electronic Components and Systems, NTIS, 1982, (P).

## ABSTRACT:

The dynamic and ever-changing characteristics of software requirements make life-cycle costs for today's software very expensive. The costs of post-operational maintenance and modification often exceeds the original development cost. Though easily maintainable software cannot be built in a natural manner, yet, much can be done well within the state-of-the-art to accommodate significant life-cycle cost savings provided that the issues are well understood and required time and effort (money) is spent during the software development phase. This paper examines the subject of software reliability and maintainability from a global perspective as it pertains to the production of a large-scale, real-time system.

#0096

Daniels, B. K., "Software Reliability Assessment," Microprocessors: Safety Implications for Industry, NTIS, 1982, (P).

## ABSTRACT:

The reliability of computer software has been causing concern for at least 15 years. The achievement of accurate software has been the goal of many workers, who identified the design process as the main source of software faults. This has led to the development of a number of design methodologies which aim to reduce the propagation of design phase errors.

The measurement of software reliability has also received considerable attention. A number of stochastic models have been developed and tested against observed software system failure data. A small number of models are being used to monitor the reliability performance of software systems as they progress through the various phases of the software life cycle.

The paper reviews reliability techniques developed for hardware dominated systems. The inputs to a software reliability analysis are considered and progress in developing a methodology to assess computer systems software is described.

Keywords: software reliability, reliability assessment methodology, reliability prediction, variability of reliability predictions.

#0097

Gephart, L. S., et al, "Software Reliability: Determination and Prediction," U. Dayton, Air Force Flight Dynamics Lab, NTIS, AD-A069 976/9ST, Jun 1978, (M).

## ABSTRACT:

This study gives a comprehensive review of software reliability determination and prediction techniques and models. Each technique and model is discussed and evaluated as to its applicability to the software in a real-time, automatic digital flight control system. A total of seven techniques, nine empirical models, and fifteen analytical models are studied. Whenever possible, the techniques and models have been applied to real software error data. The report is divided into three sections. Section I discusses software reliability in general and then focuses on each of the techniques and models individually. It provides a preliminary evaluation of each model and partitions out four of the most promising approaches, which are then analyzed more thoroughly. Section II addresses the absolute necessity of gathering well documented software error data as well as the problems associated with its collection. It also provides references for a number of software error data sets. Section III includes conclusions relative to the most attractive models, recommendations for the collection of software error data, and suggestions for future study.

#0098

Glass, R. L., "Software Reliability Guidebook," NTIS, 1979, (M).

## ABSTRACT:

This guidebook is intended to be useful for all application areas and sizes of software projects. Special emphasis is placed on the problems of large projects, such as those of military/space applications and massive interrelated data bases.

Chapter 1 discusses the concept of software reliability. Included are the definitions of reliability, verification and validation, certification, inspection, and so on. Chapter 2 focuses on the role of reliability in software development. Chapters 3 and 4 report on reliability tools and techniques. Chapter 5 makes recommendations of how software can be made more reliable. Finally, several case histories of actual software projects are given in the concluding chapter.

#0099

Musa, J. D. and A. Iannino, "Software Reliability Modeling - Accounting for Program Size Variation Due to Integration and Design Changes," NTIS, 1981, (P).

## ABSTRACT:

Estimation of software reliability quantities has traditionally been based on stable programs; i.e., programs that are completely integrated and are not undergoing design changes. Also, it is ordinarily assumed that all code is being executed at one time or another and that test or operational results are being completely inspected for failures. This paper describes a method for relaxing the foregoing conditions by adjusting the lengths of the intervals between failures experienced as compensation. The resulting set of failure intervals represents the set that would have occurred for a completely inspected program that was at all times in its final configuration. The failure intervals are then processed as they would be for a stable program. The approach is developed for the execution time theory of software reliability, but the concepts could be applied to many other models as well. Many definitions are given to describe program size variation and associated phenomena. Attention is focused on the special case of sequential integration and pure growth. The adjustment method is described and its benefits in improving the estimation of quantities of interest to the software manager are illustrated.



#0100

Air Force, "Software Safety Handbook," (Draft) HQ AFISC/SESD, Norton AFB, CA, 1984, (P).

## ABSTRACT:

The primary purpose of this handbook is to document Air Force technical knowledge of techniques and methodologies that can be used to support acquisition programs which involve computer/embedded computer systems. It is intended to aid the engineering design development of "safe" systems which utilize software and supplement the MIL-STD-882B software hazard analysis task.

This document is intended for use primarily by DoD program managers and technical specialists in the area of safety and software engineering. It is intended to serve as a companion document to MIL-STD-882 and to act as a guide in accomplishing the software safety task.

Specific information includes definitions, rationale for software safety programs, specific requirements necessary to design safety into software systems, software safety analysis philosophy and techniques, and a software system safety checklist.

#0101

Heidler, W., et al, "Software Testing Measures," General Research Corp., NTIS, AD-A118 254, May 1982, (M).

## ABSTRACT:

This report examines the current state of development of automated software testing techniques. The report identifies and describes techniques that are useful for detecting errors in software. It also examines techniques for proving the correctness of programs, for debugging (locating and correcting errors), and for producing documentation automatically. The techniques are evaluated in the areas of effectiveness, reliability, cost, and ease of use--criteria for each of these categories was developed as a part of the study effort. Profiles are presented for five major categories of test techniques--each profile describes in detail the capabilities of a technique, the automated tools that support it, the types of errors that it can detect, its degree of dependence on user skill and judgment, its applicability to various types of software, and its costs in terms of analysis time and computer resources. Important features and shortcomings of the techniques are discussed. The appendices to the report include: a set of guidelines for testing software, a survey of available automated tools which support the techniques, an automated bibliography of testing, and a description and results of an experiment with assertion testing.

#0103

Swinson, Gary E. and Stephen O. Jones, "Standard Software Support Facility Evaluation Final Report," BDM/TAC-80-693-TR, 28 Nov 1980, (R).

## ABSTRACT:

The Air Force Operational Test and Evaluation Center (AFOTEC) has the responsibility for performing operational test and evaluation (OT&E) of assets entering the Air Force inventory. One category of assets for which evaluation is required is software support facilities (SSFs) which provide operational software maintenance services for fielded embedded computer systems (ECS). SSFs are expected to provide the resources necessary to implement required maintenance actions. These resources may be categorized as facilities (i.e., the physical plant and the services it provides), support systems (hardware and software), and personnel. The specific resources employed across SSFs differ widely, particularly due to the variety of systems being supported. Because a standardized approach to evaluating these resources does not exist, a methodology is needed which will allow SSF adequacy to be measured consistently against predetermined criteria.

## This report:

- (1) Presents the results of the research effort to characterize SSFs in terms of similarities and differences in the resources they apply to software maintenance.
- (2) Characterizes the SSF evaluation process and suggests an approach for "standardized" and "consistent" implementation of the process across SSFs.
- (3) Identifies preferred SSF evaluation tools and techniques and recommended means of implementation.
- (4) Lists the documents referenced in this report.
- (5) Presents a comprehensive bibliography of literature dealing with software maintenance, SSFs, and the evaluation of software support activities.
- (6) Provides a comprehensive glossary of acronyms and key terms related to standard SSF evaluation.
- (7) Provides selected source material on SSFs gleaned from the visits reported.

#0104

Wilburn, N "Standards and Guidelines Applicable to Scientific Software Lifecycle," Hanford Engineering Development Lab, HEDL-SA-2553-FP, NTIS, 1981, (M).

ABSTRACT:

A survey of 99 standards and guidelines is given as to their applicability in the development of scientific software. The coverage by the standard or guidelines of the four aspects (performance, documentation, verification, management) of each of the six phases of the software life cycle (requirements, design, implementation, testing, operation, maintenance) is identified.

#0105

DeMillo, R. and F. G. Sayward, "Statistical Measures of Software Reliability," Georgia Institute of Technology, GIT-ICS-80, NTIS, AD-A100 662, Oct 80, (M).

## ABSTRACT:

Estimating program reliability presents many of the same problems as measuring software performance and cost: the central technical issue concerns the existence of an independent objective scale upon which may be based a qualitative judgement of the ability of a given program to function as intended in a specified environment over a specified time interval. Several scales have already been proposed. For example, a program may be judged reliable if it has been formally proved correct (1), if it has been run against a valid and reliable test data set (2), or if it has been developed according to a special discipline (3). While these concepts may have independent interest, they fail to capture the most significant aspect of reliability estimation as it applies to software: most software is unreliable by these standards, but the degree of unreliability is not quantified. A useful program which has not been proved correct is unreliable, but so is, say, the null program (unless by some perversity of specification the null program satisfies the designer); an operationally meaningful scale of reliability should distinguish these extremes.

#0108

Smith, M. and D. Hudson, "A Survey of Software Validation, Verification, and Testing Standards and Practices at Selected Sites," Boeing Computer Services Co., NBSIR82-2482, NTIS, PB82-209172, Apr 1982, (M).

## ABSTRACT:

A survey of software validation, verification and testing (V,V&T) practices at five governmental and five commercial sites was performed. The survey collected information describing each site environment, software development and maintenance practices, the V,V&T techniques and tools employed, and standards and/or procedures guiding the activities at each site. This report summarizes the information obtained and presents observations about current operations with respect to software development, maintenance, and V,V&T. It also includes reports discussing each of the sites surveyed, and the survey instruments used.

#0109

DoD, "System Safety Program Requirements," MIL-STD-882B, 30 Mar 1984, (R).

## ABSTRACT:

This standard provides uniform requirements for developing and implementing a system safety program of sufficient comprehensiveness to identify the hazards of a system and to impose design requirements and management controls to prevent mishaps by eliminating hazards or reducing the associated risk to a level acceptable to the managing activity (MA). The term "managing activity" usually refers to the Government procuring activity, but may include prime or associate contractors or subcontractors who wish to impose system safety tasks on their suppliers.

## COMMENT:

This standard applies to DoD systems and facilities including test, maintenance and support, and training equipment. It applies to all activities of the system life cycle, e.g., research, design, technology development, test and evaluation, production, construction, operation and support, modification and disposal. The requirements will also be applied to DoD in-house programs.

#0110

Goel, Amrit L. and Kazuhira Okumoto, "When to Stop Testing and Start Using Software," DACS 83(2754), 1981, (P).

## ABSTRACT:

During the last decade, numerous studies have been undertaken to quantify the failure process of large scale software systems. An important objective of these studies is to predict software performance and use the information for decision making. An important decision of practical concern is the determination of the amount of time that should be spent in testing. This decision, of course, will depend on the model used for describing the failure phenomenon and the criterion used for determining system readiness.

In this paper the authors present a cost model based on the time-dependent fault detection rate mode of Goel and Okumoto and describe a policy that yields the optimal value of test time  $T$ .

A brief overview of the failure model is given in Section 2. The cost model and the optimal policies are described in Section 3. The results are illustrated via numerical examples in Section 4.



#0111

Efron, B. The Jackknife, Bootstrap, and Other Resampling Plans, Philadelphia: Society for Industrial and Applied Mathematics, 1982, (B).

## ABSTRACT:

This book is a collection of ideas concerning the nonparametric estimation of bias, variance, and more general measures of error. The book proceeds historically, beginning with the Quenouille-Tukey jackknife. Nonetheless, some material has been deliberately omitted from this short book. This includes most of the detailed work on the jackknife, especially the asymptotic theory. Next, the bootstrap method is discussed; both parametric and nonparametric versions are presented. It is shown by the author that the bootstrap underlies the jackknife method and other resampling plans.

#0112

Shepard, R. F. and V. I. Young, "Quantitative Techniques for DARPA Program Risk Management", Falls Church, VA: Meridian Corporation, 1983, (M).

## ABSTRACT:

This paper puts forth a newly developed approach to risk assessment which draws upon numerous statistical and empirical techniques to evaluate contractor performance. The approach focuses on the prediction of costs in the short run and the indication of risk over longer time horizons. The method employs a quadratic curve-fitting algorithm to estimate short term cost fluctuations, and it uses theoretical and empirical cost models both to estimate the cost at completion and as well as to gauge the reasonableness of the expenditures to date. The approach consists of a series of risk assessment indicators which collectively address the potential for short-, mid-, and long-term cost growth. The risk assessment indicators used are:

- a cost performance analysis model,
- a curve fitting algorithm,
- Rayleigh analysis,
- a beta distribution model,
- a parametric milestone analysis, and
- Bayesian analysis.

## COMMENT:

This paper presents several state-of-the-art methods for contract cost analysis at DARPA. The methods are oriented toward the needs of senior level decision-makers who must evaluate in the aggregate the requirements for contingency reserves and who have ultimate responsibility for the successful completion of programs within established cost, schedule, and technical constraints.

#0113

Bannister, J. E. and P. A. Bawcutt, Practical Risk Management, London: Witherby and Co., 1981, (B).

## ABSTRACT:

This book on risk management evolved from a series of techniques borrowed from other disciplines for handling the increasing uncertainties of commercial, industrial, and political life. Risk management has been applied in these situations to reduce substantially the cost of on-going regular loss.

The risk management ideas presented in this book focus on recognizing future uncertainty, thinking through its possible manifestation and effects, and devising plans to reduce the impact of risk on individuals or organizations. Risk management includes assessing the range of possible variation and making sure that provision has been made to handle fluctuation by insurance and other means. The book stresses that the starting point for risk management should be a simple assessment of the problem. Over complexity can make the problem worse. Considerable detail should only be attempted when the broad risk situation is clearly understood and the overall objectives defined.

## COMMENT:

This book provides a fairly comprehensive view of risk management. First, risk management is defined. Then the book goes through the topics of risk identification, risk measurement, risk control, risk financing, and risk management control. The overall flavor of the book is a financial one.

#0114

Fisk, F. B. and W. G. Murch., "A Proposal for Computer Resources Risk Assessment During Operational Test and Evaluation," AFOTEC, 3 Oct 1983, (P).

## ABSTRACT:

The application of risk analysis and reporting of real-world problems is a difficult task with few well-defined approaches. This paper describes the approach, models, and analytical framework under development for combining both subjective and quantitative software operational and supportability measures into a management-oriented assessment of consumer risk. Preliminary results from applying risk assessment to computer resource evaluations are provided to demonstrate its application.

## COMMENT:

This paper provides a framework for evaluation and reporting software user and supporter risks associated with acceptance of computer resources and software. Current AFOTEC evaluation methodologies are used to illustrate the risk assessment approach. An inference correlation matrix of user/supporter risk references and consequence values determine coupling of the various risk factors.

#0115

Munera, H. A. and G. Yadigaroglu, "A New Methodology to Quantify Risk Perception," Nuclear Science and Engineering, Vol 75, 1980, (P).

## ABSTRACT:

A novel approach for establishing acceptability of risk is presented and illustrated by an application to the case of light water nuclear reactors. The methodology is a utility based approach. Specifically, the main advantage of the method is that it decouples consideration of the utility of consequences from an individual's attitude toward uncertainty. Thus, individual preference or aversion of a certain consequence is quantified by a preference index under certainty that can be assessed by presenting deterministic choices to the particular decision maker. His/her attitude toward uncertainty is taken separately into consideration through the use of two risk parameters that quantify his behavior with respect to random events. In other words, an individual's attitude toward a certain consequence, such as loss of life, is described by a preference index under certainty, separately from his attitude toward uncertainty. Another advantage of the method is that the method takes into consideration the shape of the probability density function over consequences, instead of simply using the expected value of the distribution.

## COMMENT:

The value of the paper is that it emphasizes the use of the entire probability density function as opposed to simply using expected values. The method proposed also attempts to integrate utility theory. Considerable work in uncertainty exists in the utility literature, and this is a good effort at combining risk assessment and utility concepts. However, the mathematics of the proposed methodology is quite complex and is applicable to past decisions rather than future ones.

#0116

Ikoku, C. U., "Decision Analysis: How to Make Risk Evaluations," World Oil, Sep 1980, (P).

## ABSTRACT:

This paper discusses the use of the expected monetary value and decision tree techniques for determining the degree of uncertainty associated with a petroleum investment. The paper states that the expected value concept is the cornerstone of decision analysis. Virtually all formal strategies for decision making under uncertainty rest on the expected value concept. This decision analysis process consists of:

- defining the possible outcomes,
- evaluating the profit or loss of each outcome,
- determining or estimating the probability of occurrence of each outcome, and
- computing the expected value.

## COMMENT:

This is a short paper aimed at the oil business. It is useful, nonetheless. The approach is essentially a statistical one. However, the methodology does not go past providing purely a point estimate of the outcomes. That is, no notion of variance or uncertainty is attached to the expected values.

#0117

Jette, G. E., "Addressing Risk and Uncertainty in Cost Estimating," Wright-Patterson Air Force Base: Aeronautical Systems Division, 1983, (M).

## ABSTRACT:

The purpose of this paper is to present some ideas on risk and uncertainty as they apply to cost estimates for weapon systems in various stages of acquisition. The Aeronautical Systems Division has developed, adopted, or refined a number of approaches to address risk in cost estimating. These techniques that have been incorporated into cost estimating are as follows:

- learning curve adjustments,
- technology indexing,
- engineering change order model,
- proposal analysis,
- range of estimates,
- confidence indexes, and
- risk/uncertainty adjustments.

The paper gives about a one page explanation of each of these techniques.

## COMMENT:

The paper represents the latest thinking on risk/uncertainty estimation by the Air Force Aeronautical Systems Division. Many of the techniques discussed fit well within a parametric framework of modeling costs. Thus, the ideas may be of use in the software supportability context.

#0118

Conrad, J. (ed.), Society, Technology, and Risk Assessment, New York York: Academic Press, 1980, (8).

## ABSTRACT:

This book is a collection of articles delivered at an international workshop held by the German Federal Ministry for Research and Technology. The workshop brought together experts from science, industry, and technology to discuss and elaborate the field of risk assessment. The main topics of the workshop and the book are:

- theoretical approaches and methods and their scope and limitations,
- why and how risk assessment has developed,
- the role, function, and practical applications of risk assessment, and
- problems concerning political decision-making.

The book is broken down into three parts. Part one of the book deals with the theoretical approaches and methodological problems of risk assessment. Part two is concerned with risk assessment from the viewpoint of sociology and philosophy of science. Part three addresses the societal and political context of risk assessment. And, part four of the book is an overall perspective of the relationship between society, technology, and risk assessment.

## COMMENT:

Several of the papers in this book are quite useful. The papers in the first section, especially the one by W. D. Rowe, provide a good fundamental basis to the risk assessment problem.



#0119

Dowie, J. and P. Lefrere (eds.), Risk and Chance, Milton Keynes, England: The Open University Press, 1980, (8).

## ABSTRACT:

This book is a collection of readings used within interdisciplinary courses on the theme of risk at the University of Kent and The Open University. The book aims to present a number of different approaches and styles of argument concerning risk and chance. The papers come from the area of psychology, philosophy, sociology, politics, economics, and mathematics. Topics of the book include: game theory, risk and human behavior, the psychology of chance, randomness, risk assessment, hazardous waste, risk and health concerns, and environmental risk.

## COMMENT:

In general, the book is not an exceptionally useful one. The diversity of the papers is its major weakness. No persistent theme on risk ties the papers together.

The best paper is the one by Otway and Pahner on risk assessment. Their paper describes some fundamental concepts such as the different levels of risk, the general structure of risk assessment, risk estimation, and risk evaluation.

#0120

Boehm, B., Software Engineering Economics, Englewood Cliffs, NJ: Prentice-Hall, 1981, (8).

## ABSTRACT:

The majority of Boehm's book describes Constructive Cost Model (COCOMO). COCOMO is a hierarchical cost estimation model consisting of three parts: basic, intermediate, and detailed levels. The basic level of COCOMO estimates the cost and scheduling (timing and staffing) of a software project based solely as a function of the number of delivered lines of source code. Estimates from Basic COCOMO are rough, early stage estimates that are within a factor of 2 of the actual costs 60 percent of the time. Intermediate COCOMO provides estimates based on source code and major software cost drivers such as product attributes, computer attributes, personnel attributes, and project attributes. Each cost driver determines a multiplying factor which estimates the effect of the attribute on software development effort. The two primary limitations of intermediate COCOMO are: 1) the estimated development effort by phase of the software project may be inaccurate, and 2) it is cumbersome to use when there are many components of a large software project. Detailed COCOMO elaborates the intermediate version, overcomes the problems of the intermediate version, and provides more accurate estimates. The Detailed COCOMO model includes phase-sensitive effort multipliers for each cost driver. These multipliers are used to determine the amount of effort required to complete each phase of the software project.

## COMMENT:

The COCOMO model is a state-of-the-art software cost model that was developed from a large data base and the expertise/experience of the author. The factors identified that drive the cost model should strongly correlate with those factors affecting software supportability.

#0121

Lathrop, Frank C., "Alternative Methods for Risk Analysis: A Feasibility Study," Air Force Computer Security Program Office, 2 Sep 1981, (R).

## ABSTRACT:

The Air Force Computer Security Program Office (AFCSPPO) is the Air Force Office of Primary Responsibility (OPR) for technical implementation of HQ USAF-developed Automated Data Processing System (ADPS) security policy. In this capacity, the AFCSPPO has designed a nine (9) element ADP security program aimed at protecting the availability, integrity, and confidentiality of those ADPSs that are under the auspice of the Director of Computer Resources, HQ USAF/ACD. One element of this nine element program is the Risk Management System (RMS) for Air Force computer systems. This paper addresses the theoretical and practical difficulties associated with risk management system implementation.

To depict the function of the Risk Management System, an RMS model has been created and is presented for reader examination and comprehension. As an aid in understanding, and to maintain contextual relevancy for this model, the reader is first exposed to specific requirements mandated by principal Federal agencies, and is further acquainted with trial-and-error efforts to field an Air Force risk management program. The reader is then informed of more recent developments and innovations within the arena of risk management, before being introduced to the RMS model.

Once the RMS model has been presented, two risk management alternatives (a qualitative alternative, and an automated quantitative alternative) are examined for their potential to satisfy the requirements of the RMS. This is done by selecting two existing risk analysis methodologies that are representative of the qualitative method and the automated quantitative method, respectively, and discussing the features of each.

The study culminates with AFCSPPO prognostications on the future development of risk management, and alternatives for managing risk in the interim period.

## COMMENT:

This paper presents an excellent foundation for a generic risk management system. Although computer security is the focus, software supportability is certainly applicable to similar techniques. Also, the historical description of computer security risk assessment is excellent for its "lessons learned" information.

#0122

National Bureau of Standards, "Guidelines for Automatic Data Processing Physical Security and Risk Management," FIPS PUB 31, Jun 74, (R).

ABSTRACT:

This publication provides guidelines to be used by Federal organizations in structuring physical security programs for their ADP facilities. It treats security analysis, natural disasters, supporting utilities, system reliability, procedural measures and controls, off-site facilities, contingency plans, security awareness and security audit. It contains statistics and information relevant to physical security of computer data and facilities and references many applicable publications for a more exhaustive treatment of specific subjects.

#0123

Grove, H. Mark, "DoD Policy for Acquisition of Embedded Computer Resources," CONCEPTS, The Journal of Defense Systems Acquisition Management, Vol 5, No 4, Special Issue-Managing Software, Autumn 1982,(P).

## ABSTRACT:

This paper present a thorough description of the acquisition process of embedded computer resources, the major management issues involved, some of the resource allocation problems, and a solution or two. There is a reasonable emphasis on the importance of the software support environment, both during system development and system deployment.

## COMMENT:

Major system policy initiatives (e.g., 5000.29) and technology initiatives such as Ada, STARS, Military Computer Family, standard instructor set architecture are briefly discussed. All of these are believed to reduce the risk of software support (corrections, enhancements, conversions) during system deployment. This paper is a very thorough, yet understandable expose of the subject area. However, it is also along the lines of reasonable traditional thought, e.g., software cost is by far the system driver, or will be; software cost will soon reach 85 percent of system cost. (See the reference abstract on the myth of the hardware/software cost ratio by Harvey Cragon of Texas Instruments.)

#0124

IEEE, IEEE Software Working Group P, "IEEE Software Reliability Guide, Second Draft - M58 (Risk Assessment), M59 (Software Functional Test Coverage Index), M60 (Software Maturity Index)," 8 Mar 1984, (P).

## ABSTRACT:

M58 (Risk Assessment) This section discusses a measure which is used to quantify the User and Supporter risk of ownership, based on the results of acceptance evaluations oriented towards measureable or subjectively evaluated User and Supporter issues. Implementation of the methodology will accommodate either quantified or subjective results, and result in individual User and Supporter (consumer) risks, or an overall composite risk. Although the primitives described here are designed for operational test and evaluation, the Risk Assessment implementation is applicable during any phase of a software life cycle by the identification of issues and subsequent selection or design of corresponding primitives and metrics.

M59. (Software Functional Test Coverage Index). This section discusses a measure which is used to quantify a software test coverage index for a software delivery. The primitives counted may either be functions or modules. The operational User is most familiar with the system functional requirements and will report system problems in terms of functional requirements rather than module test requirements. It is the task of the evaluator to obtain or develop the functional requirements and associated module cross reference table.

M60 (Software Maturity Index) This section discusses a measure which is used to quantify a software maturity index for a software delivery, based on the functions (modules) that include changes and additions from the previous delivery. The primitives counted may either be functions or modules. The operational User is most familiar with the system functional requirements and will report system problems in terms of functional requirements.

#0125

USAF Scientific Advisory Board, "The High Cost and Risk of Mission-Critical Software," Ad Hoc Committee Report, Dec 1983, (R).

## ABSTRACT:

The USAF recognizes the criticality of the high cost of software as it moves to ever increasing reliance on digital electronics in future weapons systems. Software has long been a significant cost factor and will increasingly impact the cost, availability, lead-time, utility and survivability of these future systems. This was confirmed during 1982 by the AFSAB study on advanced electronics which concluded that software was the critical success issue and which was the progenitor for this study.

Software is still an emerging technology; it has ever increasing demands placed upon it, and its role and its advantages over equivalent hardware are well established. Software, however, is becoming an increasingly larger factor in total system acquisition costs and schedules. The need is to identify specific steps to improve productivity, improve reliability and avoid software-related delays and cost growth in the acquisition of new software-intensive systems. It is also significant that the software life cycle costs allocation is 40 percent for development and 60 percent for support after fielding.

The rapidly increasing growth in the need for new software also increases the demand for improved productivity in this highly labor-intensive system component. Productivity is only one facet to the solution of problems relating to software, however. Problems will remain until there is better cost predictability and schedule control and higher confidence through increased reliability.

Consequently, to respond to the AFSAB objective of studying Air Force opportunities to deal with the high cost and risk of software for mission critical systems, the study group considered its major focal points to be the issues of:

- o Predictability and control
- o Productivity and quality
- o Post deployment software support.

Key to the conduct and goals of the study were that it would result in specific solutions or corrective programs the Air Force could implement directly from this study.

Each of the three issue areas are discussed in detail in sections of the report, together with specific recommendations for each. In addition to the specific findings of each of the three subcommittees, there were three recurring themes: management, organization and personnel, that top-level Air Force management must take action on, and make commitments to, if the Air Force is to realize the full effect of the proposed detailed measures. These are discussed with specific recommendations.

There are two ongoing DoD programs which are important steps in advancing the technology, including productivity of software development and support. These programs, Ada and STARS, offer direct benefits to the Air Force and are essential for software to keep pace with the requirements of new weapons systems. A third DoD program, VHSIC, has the potential for significant advance in warfighting capability. However, it may be limited by its critical dependence on software technology.

COMMENT:

This study is a broad overview of why there is risk associated with software, where emphasis should be, and that some immediate action on the recommendations should be taken. The major recommendations from this study are:

- (1) Establish a focused, high-priority career path for software and computer system personnel.
- (2) Create a plan to evolve to a DCS-level manager of USAF information resources, including mission-critical and embedded computers and software.
- (3) Establish a software engineering and computer system technology and support center to collect and focus Air Force resources on software issues.



#0127

Houghton, Raymond C. Jr., "Software Development Tools: A Profile,"  
Computer, May 83, (P).

## ABSTRACT:

Conclusions reached at the IEEE Test and Documentation Workshop indicated a need for a public information exchange on software development tools. Two reasons were cited: the first is a general lack of information about the tools available, their capabilities, and where they can be obtained; the second is a lack of awareness of current tool development, which leads to duplication of effort. At the workshop, the National Bureau of Standards' Institute for Computer Science and Technology, or NBS/ICST, agreed to initiate the collection of information about software tools in the hope of alleviating some of these problems.

This article reports the results of this collection effort by analyzing the information obtained. Various categorizations of the tools are presented, with classes listed by their characteristics. The lists incorporate percentage summaries based on the total number of tools for which information is available.

## COMMENT:

This paper is an important source for summary information on software tools, identification and classification. Other sources of information are also identified. It might be useful to use the same taxonomy approach to more clearly identify software support tools and their capabilities as part of the SSF evaluation process.

#0128

Howden, William E., "Contemporary Software Development Environments," Communications of the ACM, Vol 25, 5, May 1982, (P).

## ABSTRACT:

There are a wide variety of software development tools and methods currently available or which could be built using current research and technology. These tools and methods can be organized into four software development environments, ranging in complexity from a simple environment containing few automated tools or expensive methods to a complete one including many automated tools and built around a software engineering database. The environments were designed by considering the life-cycle products generated during two classes of software development projects. Relative cost figures for the environments are offered and related issue, such as standardization, effectiveness, and impact, then addressed.

## COMMENT:

This paper presents a practical classification scheme of software support environments in which increasingly more complex and capable support system resource requirements are identified. This could be useful in a risk assessment approach where risk of alternative environments (I, II, III, IV) could be assessed using the defined characteristics in each class as a descriptive checklist rather than actually evaluating each characteristic.

#0129

Vessey, Iris and Ron Weber, "Some Factors Affecting Program Repair Maintenance: An Empirical Study," *Communications of the ACM*, Vol 26, 2 Feb 83 (P).

## ABSTRACT:

The focus of recent research has been structured programming. Previously the concerns were modular programming methodologies, use of decision tables, test data generators, automatic flowcharters, etc. To date the research on methods to improve program quality and lower program development, implementation, and maintenance costs has been primarily theoretical.

Most of the developed theories have been normative, that is, they stated what should be done to improve the quality of programs and the programming process. Unfortunately, these theories have rarely been subjected to empirical testing, and so their value remains unknown. They provide the zealots with opportunities to market a rash of seminars and courses and to flood the literature with papers advocating the new technologies. When the theories are subjected to testing, what little evidence has been obtained sometimes suggests that the claimed benefits, in fact, may not exist.

This paper describes three empirical studies of factors purported to affect the extent of repair maintenance carried out on programs. By repair maintenance we mean maintenance needed to correct logic errors discovered in a program after it has been released into production. These logic errors arise because program specifications are implemented incorrectly when the program is first written, or as the consequence of maintenance carried out incorrectly after the initial production release. We distinguish repair maintenance from adaptive maintenance and productivity maintenance. Adaptive maintenance permits a program to evolve to better meet user needs. Productivity (perfective) maintenance seeks to improve the efficiency with which a program consumes resources.

The paper proceeds as follows. First, we articulate the hypotheses tested in the studies and briefly discuss the theoretical, empirical, and popular bases that exist in support of these hypotheses. Second, we discuss the data collected and the results obtained in an Australian study. Third, we discuss the data collected and the results obtained in two U.S. studies. Fourth, we examine the implications of the results. Finally, we present our conclusions and identify several directions for further research.

## COMMENT:

This paper has significance to risk assessment since the factors of software supportability, upon which risk assessment determination and

evaluation is based, must be these drivers which affect the extent of software maintenance (including repair). If the drivers are incorrectly selected, then the effectiveness of risk assessment is affected.

Some results from this paper include:

- (1) Our first conclusion from the results is that repair maintenance does not seem to constitute a very important activity.
- (2) In two of the three organizations studied, we found support for Boehm's hypothesis that the likelihood of a successful first run after only a minor modification is small.
- (3) Found little difference between the repair maintenance rates for moderately complex programs. The factor is statistically significant because the repair maintenance rate for easy programs differs from the repair maintenance rate for moderately complex or complex programs. In fact, the estimate of the repair maintenance rate for moderately complex programs is slightly higher than the rate for complex programs.
- (4) Only weak support exists for programming style having an effect on the repair maintenance rate.
- (5) We found no support for the hypothesis that the number of production runs between repairs increases after each repair.

#0130

Boehm, B. W., J. R. Brown and M. Lipow, "Quantitative Evaluation of Software Quality," Proceedings 2nd International Conference on Software Engineering, San Francisco, CA, pp. 592-605, 1976, (P).

## ABSTRACT:

The study reported in this paper establishes a conceptual framework and some key initial results in the analysis of the characteristics of software quality. Its main results and conclusions are:

- (1) Explicit attention to characteristics of software quality can lead to significant savings in software life-cycle costs.
- (2) The current software state-of-the-art imposes specific limitations on our ability to automatically and quantitatively evaluate the quality of software.
- (3) A definitive hierarchy of well-defined, well-differentiated characteristics of software quality is developed. Its higher-level structure reflects the actual uses to which software quality evaluation would be put; its lower-level characteristics are closely correlated with actual software metric evaluations which can be performed.
- (4) A large number of software quality-evaluation metrics have been defined, classified, and evaluated with respect to their potential benefits, quantifiability, and ease of automation.
- (5) Particular software life-cycle activities have been identified which have significant leverage on software quality.

Most importantly, we believe that the study reported in this paper provides for the first time a clear, well-defined framework for assessing the often slippery issues associated with software quality, via the consistent and mutually supportive sets of definitions, distinctions, guidelines, and experiences cited. This framework is certainly not complete, but it has been brought to a point sufficient to serve as a viable basis for future refinements and extensions.

## COMMENT:

This paper was one of the first recorded descriptions of a hierarchy of software quality factors and the systematic process by which one can evaluate software quality. It was a foundation paper for the development of some of the AFOTEC OT&E evaluation of software quality characteristics.

#0131

Lientz, Bennet P. and E. Burton Swanson, "Problems in Application Software Maintenance," Communications of the ACM, Vol 24, 11, Nov 81, (P).

## ABSTRACT:

The problems of application software maintenance in 487 data processing organizations were surveyed. Factor analysis resulted in the identification of six problem factors: user knowledge, programmer effectiveness, product quality, programmer time availability, machine requirements, and system reliability. User knowledge accounted for about 60 percent of the common problem variance, providing new evidence of the importance of the user relationship for system success or failure. Problems of programmer effectiveness and product quality were greater for older and larger systems and where more effort was spent in corrective maintenance. Larger scale data processing environments were significantly associated with greater problems of programmer effectiveness but with no other problem factor. Product quality was seen as a lesser problem when certain productivity techniques were used in development.

## COMMENT:

This paper is a much-quoted source for software maintenance problems. The application organizations surveyed were primarily ADP shops rather than military support facilities. Still many of the issues surfaced probably are to some degree also issues for military software support.

#0132

Peercy, David E. and Gary E. Swinson, "A Software Support Facility Evaluation Methodology," Symposium on Application and Assessment of Automated Tools for Software Development, Nov 83, (P).

## ABSTRACT:

The Air Force Operational Test and Evaluation Center has been supporting the development over the past 5 years of a comprehensive methodology and tool set for the evaluation of software and its support environment for maintenance characteristics. The support environment is called a Software Support Facility. This paper describes the methodology developed by The BDM Corporation to evaluate a planned or existing software support facility for its capability to support the software maintenance actions required for a given Embedded Computer System. Elements of the evaluation methodology include a generic resource framework within which requirements can be specified, and a set of systematic evaluation procedures for performing the actual evaluation. A software support evaluation tool was developed to automate a major portion of the operational evaluation process.

## COMMENT:

This paper describes the AFOTEC software support facility evaluation methodology as it existed in the 1983 time period. Any changes have been made as reflected in the AFOTEC 800-2, Volume 5, "Software Support Facility Evaluation - User's Guide."

#0133

Peercy, David E., "A Framework for Software Maintenance Management Measures," Proceedings of the Seventeenth Annual Hawaii International Conference on System Sciences, Jan 84, (P).

## ABSTRACT:

Some of the important issues and problems of software maintenance management are discussed within the context of a proposed software maintenance framework. This framework consists of four elements: software products, software maintenance environment, software maintenance management, and software maintenance measures. Emphasis is upon the need for a data base of accurate measures to support the management decision process. The measures are used to determine which characteristics, techniques, tools, and requirements have the most effect on maintenance resource requirements and allocations. Elements of software product quality, software maintenance environments, and software maintenance activity are briefly discussed.

## COMMENT:

This paper presents a possible evaluation framework for risk assessment of software supportability. Elements of software supportability are introduced. Emphasis is upon management measures.



#0134

Cragon, Harvey G., "The Myth of the Hardware/Software Cost Ratio," Computer, Open Channel, Dec 82, (P).

## ABSTRACT:

This short note discusses one of the "folk laws" of the computer industry that surfaces from time to time: "the cost of user's programming represents approximately 70% of cost, with hardware accounting for the remaining 30%." The law further states that by the end of this decade, software will be 85 percent of total cost. The ratio varies depending on the author. For example, a 70:30 percent ratio is sometimes quoted as is 80:20 percent. Nevertheless, the general thrust of this "law" is that today, software costs are two to four times the cost of hardware.

## COMMENT:

The origin of the famous hardware/software cost trend curve is a paper by B. Boehm which reports the results of an Air Force Study, "Information Processing/Data Automation Implications of Air Force Command Control Requirements in the 1980's," 1973. Boehm projected the current (1972) 3:1 ratio of software: hardware cost would be 9:1 in 1985. As the author points out, a recent (1982) Air Force paper on a proposed DoD software technology program contains a chart of proposed software; hardware costs for DoD embedded systems showing a ratio closer to 2.2:1. This ratio for recurring cost, large volume cases is more in the order of 4 to 5% software, 35 to 40% hardware, and the rest staff and overhead expense.

The problem is not that the earlier projections were incorrect (at the time), but that the tendency is to still use such predictions (now) when clearly they are not valid. Part of risk analysis is an economic resource evaluation, which must carefully avoid folk lore and myths as much as possible. The author is not suggesting that an improvement in software development cost isn't needed. He merely wants to call attention to a myth that permeates our industry. Belief in this myth obscures the very real cost problem in software development and maintenance by creating a meaningless ratio that gives a false understanding of the situation. The cost of software is high, but less than the cost of hardware. Any other interpretation of the available data is invalid.

#0135

Neugent, William, John Gilligan, and Lance Hoffman, "Technology Assessment: Methods for Measuring the Level of Computer Security," National Bureau of Standards, Institute for Computer Sciences and Technology, Washington, D.C., Sep 81, (R).

## ABSTRACT:

This contractor report from System Development Corporation is the result of an effort initiated in early 1980. It is the first phase of a project at the Institute for Computer Sciences and Technology (ICST) to produce a guidance document in the area of computer security certification. At the outset it seemed very clear that such a certification would heavily depend upon a technical evaluation of some kind and that an investigation should be made of current evaluation methodologies. This report comprehensively reviews a large number of the evaluation methods in use today and discusses their major characteristics and differences. It should prove very helpful to those organizations engaged in selecting computer security evaluation methods and should be considered a foundation document for sound security certifications and risk assessment.

This report will be the basis for a National Bureau of Standards Special Publication. It is being released at this time in this form to make the information available sooner than would otherwise be possible. We at ICST hope that interested readers will send us constructive comments on this document so that the final publication will be as useful and accurate as possible.

## COMMENT:

This document is an excellent source for life cycle measurement policy methodology, techniques and tools. A good discussion is included of the various risk assessment/analysis methods such as FIPS PUB 63, AFRAMP, SDC Navy RAM and RAMP. This is definitely a key document for understanding computer system security concepts. The report was produced for the National Bureau of Standards (NBS) in conjunction with the NBS Security and Risk Management Standards Program. The intent of the report is to provide a comprehensive assessment of the state of the art and to provide a suitable basis for subsequent, more focused efforts to produce a Federal Information Processing Standards (FIPS) guideline on computer security certification. This guideline, on computer security certification, FIPS PUB 102 has been released (27 Sep 83).



1·0

1·1

1·25

2·8  
3·15  
3·5  
4·0  
4·5

1·4

2·5

2·0

1·8

1·6

#0135

GAO, "Federal Agencies' Maintenance of Computer Programs: Expensive and Undermanaged," Reports to the Congress, Government Accounting Office, AFMD-81-25, 26 Feb 81, (R).

## ABSTRACT:

Federal agencies spend millions of dollars annually on computer software (program) maintenance but little is done to manage it.

GAO studied 15 Federal computer sites in detail, and received completed questionnaires from hundreds of others. All reported large maintenance efforts but few had good records and very few managed software maintenance as a function.

Improvements can and should be made both in reducing maintenance on existing software and in constructing new software to reduce its eventual maintenance costs.

The National Bureau of Standards should issue a standard definition and specific technical guidelines for software maintenance. Heads of Federal agencies should require their automatic data processing managers to manage software maintenance as a discrete function.

## COMMENT:

This report was the impetus behind the current (1984) NBS effort to produce software maintenance management guidelines such as the NBS Special Publication 500-106, "Guidance on Software Maintenance," Dec 83.

#0137

NBS, Martin R., and W. Osborne, "Guidance on Software Maintenance," NBS Special Publication 500-106, National Bureau of Standards, Institute for Computer Sciences and Technology, Dec 83, (R).

## ABSTRACT:

This report addresses issues and problems of software maintenance and suggests actions and procedures which can help software maintenance organizations meet the growing demands of maintaining existing systems. The report establishes a working definition for software maintenance and presents an overview of current problems and issues in that area. Tools and techniques that may be used to improve the control of software maintenance activities and the productivity of a software maintenance organization are discussed. Emphasis is placed on the need for strong, effective technical management control of the software maintenance process.

## COMMENT:

This report is the first of a series of reports which will address software maintenance. This report is primarily an overview of some of the issues.

#0138

DoD, "Software Technology for Adaptable, Reliable Systems (STARS) Program Strategy," Department of Defense, 1 Apr 83, (R).

## ABSTRACT:

This document proposes a strategy for the Software Technology for Adaptable, Reliable Systems (STARS) program to improve our ability to exploit the advantages of computer technology. The original version was prepared at the direction of Dr. Edith Martin, Deputy Under Secretary of Defense for Research and Engineering (Research and Advanced Technology) and published 1 October 1982. This revised expanded version was produced by the STARS Joint Task Force based on Service and Agency comments on the earlier version and a variety of public comment, including those growing out of discussions at a public workshop. Details of the STARS Joint Task Force activities are summarized in the STARS Joint Task Force Report.

The STARS Program Strategy contains several levels of detail. The Executive Summary provides an overview of STARS. The body develops the rationale and guiding principles, explaining the motivation for the goal, supporting objectives, implementation approach, and organizational mechanisms. Supporting documents provide additional detail. The appendices to the 1 October 1982 Strategy for a DoD Software Initiative provide supporting detail of an historic nature and remain unchanged. STARS Functional Task Area Strategies detail the tasks, ordered according to the eight categories outlined in section 4; which could lead to successful improvement. The STARS Implementation Approach provides details of the initial implementation planning and forms the basis for a program plan. The A Candidate Strategy for the Software Engineering Institute provides details for further planning of the institute.

## COMMENT:

This document document describes a management strategy and an initial approach for a DoD-wide Software Technology for Adaptable, Reliable Systems (STARS) program to improve our ability to exploit the advantages of computer technology through software. The program will improve the state of practice in the acquisition, management, development, and support of computer software for military systems. It establishes overall objectives, provides an approach for achieving the objectives, and identifies the management structure necessary to develop a program plan. Since this approach will require cooperation among DoD elements, industry, and academia, it must be refined continually through extensive coordination within DoD and the computing community. The STARS program could have far reaching effects on how software is supported and the associated risks of that support since development of common productivity tools for a support environment is one of the major goals.

#0139

Lindquist, Timothy E., Jeffrey L. Facemire, and Dennie G. Kafura, "A Specification Technique for the Common APSE Interface Set," Office of Naval Research, 84004-R, Apr 84, (R).

## ABSTRACT:

This report demonstrates an approach to specifying kernel Ada support environment interface components. The objectives are to provide a mechanism which allows building a complete enough specification for validation, an understandable specification, and one that is relatively easy to construct. In meeting these objectives, an Abstract Machine approach has been modified and applied to functional description of kernel operations. After motivating and explaining the approach, the paper exemplifies its utility.

Interactions among kernel operations and pragmatic implementation limits, which are other needed parts of a specification, are also discussed.



#0140

Kafura, Dennis, J. A. N. Lee, and Timothy Lindquist, "Validation in Ada Programming Support Environments," Engineering Psychology Group, Office of Naval Research, Working Paper, NRSRO-101, 7 Jan 83, (R).

## ABSTRACT:

To this date validation has been applied in only two areas, in the validation of programs and the validation of compilers and then not to any degree which can truly be classified as more than "empirical." This study was established to investigate the steps which would be needed to extend those previous experiences into the realm of programming environments and in particular the environments being proposed for use in the Ada program. A model of such environments already exists but is found to be lacking in essential detail necessary for an implementation to prescribe a model by which validation can be specified. This report does not itself provide any details of specific validation procedures or mechanisms, but rather investigates the processes for Ada Programming Support Environment (APSE) implementation in terms of the Ada Programming Language, and uses those specifications to suggest a mechanism for validation suite development.

Further, in order to accomplish these goals it is suggested that the conceptual model of the "STONEMAN" document be extended to express the wider computing environments in which the APSE would reside. This extended model would also provide a fundamental basis for the design of Ada systems which respond to the need to provide networking, distributed processing and security enclaves.

#0141

RADC, Bowen, T., et. al, "Software Quality Measurement for Distributed Systems," Rome Air Development Center, Volumes I, II, III, Jul 83, (P).

**ABSTRACT:**

This document is the final technical report (CDRL A003) for the Quality Metrics for Distributed Systems contract, number F30602-80-C-0330. The contract was performed for Rome Air Development Center (RADC) to provide methodology and technical guidance on software quality metrics to Air Force Software acquisitions managers.

This report consists of three volumes as follows:

- (1) Volume I - Software Quality Measurement for Distributed Systems - Final Report.
- (2) Volume II - Guidebook for Software Quality Measurement.
- (3) Volume III - Distributed Computing Systems: Impact on Software Quality.

The objective of this contract was to conduct exploratory development of techniques to measure system quality with a perspective on both software and hardware from a life cycle viewpoint. The effort was expected to develop and validate metrics for software quality on networked computers and distributed systems; i.e., systems whose functions may be tightly distributed over microprocessors or specialized devices such as data base machines. At the same time, the effects hardware has on software was to be studied, as well as the trade-offs between hardware, firmware, and software. The results of this research are reported in Volume I.

Volume II describes the application of quality metrics to distributed systems and provides guidance for AF acquisition managers. The guidebook provides guidance for specifying and measuring the desired level of quality in a software product.

Volume III describes a qualitative study of distributed system characteristics, reasons for selection, design strategies, topologies, scenarios, and trade-offs. These analyses led to the changes in the framework shown in Volume I, and to the validation of models.

#0142

RADC, Angus, J. E., J. B. Bowen, S. J. VanDenBerg, "Reliability Model Demonstration Study," Volumes I and II, Rome Air Development Center (COEE), RADC-TR-83-207, August 1983, (M).

## ABSTRACT:

This report contains the results of a study to determine the use and applicability to Air Force software acquisition managers of six quantitative software reliability models to a major command, control, communications, and intelligence (C<sup>3</sup>I) system. The scope of the study included the collection of software error data from an ongoing C<sup>3</sup>I project, fitting six software reliability models to the data, analyzing the predictions provided by the models, and developing conclusions, recommendations, and guidelines for software acquisition managers pertaining to the use and applicability of the models.

#0143

Directorate of Aerospace Safety, "A Risk Management Guide for Air Force Operations," Air Force Inspection and Safety Center (AFISC), Norton AFB, CA, 6 Nov 79, (R).

## ABSTRACT:

This guide has been prepared to provide AFISC personnel and members of Air Force major commands with suggested techniques for assessing risk and acting to minimize this risk. The major thrust of this document is aimed at risk analysis of operational missions. Although primary emphasis is on air operations, the approach, the bulk of which is described in chapters 3 and 4, can be used to structure a thought process for managing risk associated with virtually any type of Air Force operation or function. This approach is particularly applicable to risk determination before the operation first takes place. Major elements which comprise the mission are identified. Procedures for performing quantitative or qualitative risk assessments are suggested and cost-benefit considerations are discussed.

Issues and problems an operational commander often faces in carrying out his function of risk management are raised. Unfortunately, a search of the literature reveals no publications that guide Air Force operations and support units on how to approach a risk analysis. This guide can be used by commanders and their staffs responsible for the operation and support of deployed weapon systems. Hopefully, it will be a first step in helping the decision makers to understand the risks involved in certain operations or maintenance activities. It is not intended to be a cure-all for all hazardous activities, but rather a method upon which the major commands or field units can build their risk assessments.

#0144

Boocn, G., Software Engineering With Ada, Reading, MA; Benjamin/Cummings, 1983, (B).

## ABSTRACT:

This book captures much of the software engineering aspect of Ada. It offers a consistent approach to design and offers advice for the development of an appropriate style.

This book is not just another introduction to Ada. It has been written to satisfy the following three specific goals:

- o To provide an intensive study of Ada's features.
- o To motivate and give examples of good Ada design and programming style.
- o To introduce an object-oriented design methodology that exploits the power of Ada and, in addition, helps us manage the complexity of large software solutions.

In short, this book not only describes the details of Ada programming but also suggests ways in which to best apply the features of the language in the creation of software systems.

The book is divided into eight packages, each of which contains three chapters that are logically related. The first package begins with a look at the Ada problem domain. It includes an examination of Ada's development history in order to provide a perspective on some of the features of the language.

In the second package, a number of modern software development principles are examined and the object-oriented design methodology is introduced.

In the third through seventh packages, a detailed presentation of Ada as an embodiment of these methodologies is provided, built around five complete design examples. Each problem is increasingly more complex, and together they require the application of almost every Ada feature. In addition, these problems provide a vehicle for demonstrating the object-oriented design methodology, along with a programming style that emphasizes understandability. In the chapters between these five large examples, a detailed discussion of Ada's constructs is presented. The book concludes with the eighth package, which examines the Ada Programming Support Environment, plus the application of Ada across the software life cycle.

#0145

LeBlanc, R., and J. Goda, "Ada and Software Development Support: A New Concept in Language Design," Computer, 15, 5, pp. 75-82, 1982, (P).

## ABSTRACT:

What Ada does is include support for the development of modular program structure and for the definition of types and operations, allowing a programmer to effectively "extend" the language. Typically, the implementation of a large software system is accomplished through the use of a programming language plus some application-oriented extensions. In most languages, however, procedures are the only available extension capability. But a language such as Ada, which provides support for more comprehensive extensions, allows greater support for software development.

Like most programming languages, Ada can be used most effectively when a programmer allows the language features to influence his or her programming style. In this article, the authors have attempted to illustrate the use of an "Ada style." One of the most important aspects of this style is the development of generalized packages through the systematic use of generics.

#0146

Lientz, B., and E. Swanson, Software Maintenance Management, Reading, MA: Addison-Wesley, 1980, (B).

## ABSTRACT:

This book presents the results of a study of computer application software maintenance in 487 data processing organizations. These applications are mostly of the business type.

Much has been written about the life cycle of computer application software. Within this context, attention has traditionally been focused on the design and development of new software. The maintenance and enhancement of existing software has received relatively little attention. However, there is increasing recognition that maintenance constitutes a persistent and significant burden. The purpose of the study reported here is to contribute to the understanding of maintenance in order that it may ultimately be better managed.

This study reports research results. While not a "how-to-do-it" cookbook, it is intended to be readable and usable by practicing data processing managers and professionals. For this reason, the book has been organized and presented to maximize efficient access to the research findings for those with a minimal level of background and/or interest in research methods.

#0147

Parikh, G., Techniques of Program and System Maintenance, Cambridge, MA: Winthrop, 1982, (B).

## ABSTRACT:

The main purpose of this book is to present programming as well as managerial techniques of software maintenance gleaned from the vast computing literature. The book is a compilation of important and useful material on software maintenance, published in the computer periodicals, conference proceedings reports, books, as well as some original material.

The book is divided into five sections. Though some chapters cover several topics, this broad classification will guide the reader in his study.

The first section introduces the problem of maintenance and provides some perspective. The second section covers "how to" aspects for a maintenance programmer. Techniques for managing maintenance are presented in the third section. The application and impact of structured technologies on maintenance are described in section four. Section five, an extension of section four, indicates possible future developments in this vital area. It includes a chapter related to "structuring engine," a software package that automatically transforms an unstructured program into a structured program.

The book contains an extensive, annotated bibliography listing works on software maintenance, as well as publications in related areas such as software testing and debugging, software tools, and structured technologies. A comprehensive index is also included.



#0148

Thayer, R., A. Pyster, and R. Wood, "Validating Solutions to Major Problems in Software Engineering Project Management," Computer, 15, 8, pp. 65-77, August 1982, (P).

## ABSTRACT:

In August of 1980, the authors wrote an article in which they hypothesized 20 major software engineering project management problems. (To avoid later confusion, they define a "software engineering project" as a software development task that has a prescribed starting point, a specific budget and resources, established responsibilities, and a completion schedule.) They also conducted an opinion survey on a sample of the data processing industry to verify these hypothesized SEPM issues. Their sample consisted primarily of senior computer scientists, authors and lecturers on software engineering and project management, software development project managers, and highly visible individuals who, because of their position in industry, government, and universities, influence the opinion of the computing community.

This article reports the results of the survey. Basically, the conclusions reached were that some techniques showed high correlation, however, many relationships were not clearly causal. The article recommended further research in the area.

#0149

McCall, J. and M. Matsumoto, "Software Quality Measurement Manual," RADC-TR-80-109, Vol I and II, Apr 80, (R).

## ABSTRACT:

Software metrics (or measurements) which predict software quality have been refined and enhanced. Metrics were classified as anomaly-detecting metrics which identify deficiencies in documentation or source code, predictive metrics which measure the logic of the design and implementation, and acceptance metrics which are applied to the end product to assess compliance with requirements.

A Software Quality Measurement Manual was produced which contained procedures and guidelines for assisting software system developers in setting quality goals, applying metrics and making quality assessments.

The purpose of this research was to refine and enhance the software quality measurement process that was originally documented in RADC TR-77-369. The work covered by this effort is contained in two volumes. The first volume includes extensions to the concepts of software quality measurement, analysis of metric applications and validation of metrics for the quality factors portability and maintainability. Appendix B of Volume I documents all the changes that have been made to the software quality metrics based on the experiences of this research study.

The second volume of this report, A Software Quality Measurement Manual, is oriented toward the quality assurance process and identifies how to set quality goals, how and when to apply software metrics and how to make a quality assessment.

#0150

Air Force, "Information Processing Standards for Computers (IPSC)", AFR 300-16, Headquarters U.S. Air Force, Washington, D.C., Jun 1979 (P).

## ABSTRACT:

This regulation provides policies and procedures for developing and implementing standards developed under the IPSC program and gives the basis for formal Air Force support of the program. It implements the Department of Defense (DoD) IPSC program for the Air Force and applies to all Air Force activities that use, or plan to use, Automated Data Processing Systems (ADPSs).

The Air Force Director of Computer Resources was made responsible for the DoD IPSC program in 1965. Responsibility includes developing, coordinating, and approving automated data processing (ADP) standards DoD-wide. Concurrent with this delegation, the Air Force IPSC program was established to manage Air Force participation in the DoD program. Both programs are administered under the DoD policies set for the defense standardization program and the procedures in DoD Manual 4120.3-M, available through normal Air Force distribution channels.

## COMMENT:

The risk of software development may be a function of the applicability or adherence to adopted Air Force software development standards. This regulation lists the appropriate approved National Standards, Federal Standards, and DoD Standards for software development.

#0151

Air Force, "Procedures for Managing Automated Data Processing Systems Documentation, Development, Acquisition, and Implementation," AFR 300-12, Vol I, Headquarters U.S. Air Force, Washington, D.C., Dec 1977, (P).

## ABSTRACT:

This regulation establishes procedures to manage the Air Force Automated Data Processing Systems (ADPS). It must be used with AFR 300-2, AFR 300-6, and other 300-series Air Force directives. It applies to all Air Force activities that plan, design, develop, authorize, select, acquire, maintain, and manage an ADPS or its components. This volume establishes the procedures for documentation, development, acquisition and implementation of Air Force ADPS or ADPS elements.

## COMMENT:

The assessment of software development risk is partially a function of the extent to which effective management policies are or can be applied to the development effort. This regulation specifies milestone reporting procedures, configuration management procedures, and various reviews and audits that are expected to occur during the life cycle of software development.

#0152

Air Force, "Computer Programming Languages," AFR 300-10, Headquarters U.S. Air Force, Washington, D.C., May 1976, (P).

## ABSTRACT:

This regulation prescribes policy for using computer programming languages, and for specifying procurement and testing requirements for computer programming language compilers.

Implementation of this policy provides Air Force computer programming language standards to enable commanders and their staffs to improve interchangeability and upward compatibility of computer programs within and among Air Force systems; reduce programming and reprogramming costs; reduce conversion efforts during transition from one computer to another; minimize requirements for retraining of computer programmers; and ensure that standard computer programming language compilers acquired from vendors comply with the Air Force standard specifications.

## COMMENT:

The risk of software development may be a function of the programming language selected and its adherence to AF standards. The provisions of this regulation potentially impact the risk to the extent that the regulation is enforced and adhered to. Of particular concern is that this regulation does not address the Ada Programming language.

#0153

Air Force, "Independent Cost Analysis Program," AFR 173-11, Headquarters U.S. Air Force, Washington, D.C., Dec 1980, (P).

## ABSTRACT:

This regulation establishes the Independent Cost Analysis Program (ICAP), prescribes policies, assigns responsibilities, and defines procedures for preparation, review, documentation, and presentation of studies conducted as part of the ICAP program. It outlines the Air Force Cost Analysis Improvement Group (AFCAIG) support provided to the Air Force System Acquisition Review Council (AFSRC) and Defense System Acquisition Review Council (DSARC). It applies to all major commands (MAJCOMs) and separate operating agencies (SOAs).

The Independent Cost Analysis Program (ICAP) consists of three types of cost analysis studies:

- (1) An Independent Cost Analysis (ICA). This analysis will be prepared on all major weapon system programs subject to DSARC/AFSARC Milestone I, II, and III reviews and as otherwise directed.
- (2) An Independent Sufficiency Review (ISR). This review is required on weapon system programs subject to DSARC/AFSARC Program Reviews or special reviews and as otherwise directed.
- (3) An Independent Cost Study (ICS). This will be prepared as a special study when requested to support the DSARC/AFSARC decision process and as otherwise directed. The ICS is the current designation for the former Independent Cost Estimate (ICE).

## COMMENT:

Cost is a major element of risk for major weapon system programs. This regulation stipulates procedures for estimating cost risk during the ICA. Specifically, paragraph 7 directs the ICA team to examine and address AFOTEC as a data source, and for "cost elements with a high degree of uncertainty, the ICA will provide sensitivity analysis using frequency distribution or ranges of cost."

#0154

Peercy, David E., "A Software Maintainability Evaluation Methodology," IEEE Transactions on Software Engineering, Vol SE-7, No. 4, July 1981, (M).

## ABSTRACT:

This paper describes a conceptual framework of software maintainability and an implemented procedure for evaluating a program's documentation and source code for maintainability characteristics. The evaluation procedure includes use of closed-form questionnaires completed by a group of evaluators. Statistical analysis techniques for validating the evaluation procedure are described. Some preliminary results from the use of this methodology by the Air Force Operational Test and Evaluation Center are presented. Areas of future research are discussed.

## COMMENT:

This paper describes the AFOTEC software maintainability evaluation methodology as it existed in the 1981 time period. Any changes made have been reflected in the AFOTEC 800-2, Volume 3, "Software Maintainability - Evaluator's Guide."

#0155

Shooman, M. L., Software Engineering, Design, Reliability, and Management, New York: McGraw-Hill, 1983, (B).

## ABSTRACT:

This book presents software engineering methodologies for the development of quality, cost-effective, schedule-meeting software. This book is divided into six lengthy chapters. Chapter 1 addresses the reader who has little previous software development experience. The focus of this chapter is on the source of software costs. Chapter 2 deals with modern software-design methods such as modularity, structured programming, top-down design, and defensive programming. Chapter 3 develops complexity measures related to development cost and the number of program errors. Chapter 4 treats testing as the prime method of revealing and pinpointing residual program errors, which must be reduced in number to improve the software. Chapter 5 explains reliability concepts and develops models for predicting and measuring software errors, reliability, and availability. Chapter 6 deals with the basic principles of software management.



#0156

Putnam, L. H., "Example of an Early Sizing, Cost and Schedule Estimate for an Application- Software System," Computer Software and Application Conference Proceedings, IEEE Computer Society, November 78, (R).

## ABSTRACT:

Software development has been characterized by severe cost overruns, schedule slippages and an inability to size, cost and determine the development time early in the feasibility and functional design phases when investment decision must be made. Managers want answers to the following questions: Can I do it? How much will it cost? How long will it take? How many people? What's the risk? What's the trade-off? This portion of the paper shows how to size the project in source statements ( $S_s$ ), how to relate the size to management parameters (life cycle effort ( $K$ ) and development time ( $t_d$ )) and the state-of-technology ( $C_k$ ) being applied to the problem through the software equation,  $S_s = C_k K^{1/3} t_d^{4/3}$ . The software equation is then solved using a constraint relationship  $K = |\nabla D| t_d^3$ , where  $|\nabla D|$  is the magnitude of the difficulty gradient empirically found to be related to system development characteristics measuring the degree of concurrency of major task accomplishment. Monte Carlo simulation is used to generate statistics on variability of the effort and development time. The standard deviations are used to make risk profiles. Finally, having the effort and development time parameters, the Rayleigh/Norden equation is used to generate the manpower and cash flow rate at any point in the life cycle. The results obtained demonstrate that engineering quality quantitative answers to the management questions can be obtained in time for effective management decision making.

#0157

Herd, J. H., J. N. Postak, W. E. Russell, K. R. Stewart, "Software Cost Estimation Study," Rome Air Development Center, Griffis AFB, NY, RADC-TR-77-220, Vols I and II, June 1977, (R).

## ABSTRACT:

The study identified factors that have an adverse effect on software cost estimates, determined their impact on software cost estimates, discussed methods for controlling the effect of these factors, and developed an overall methodology for estimating the costs of software development. In addition to a generalized model for estimating software development costs, separate models have been generated for estimating the development cost of command and control, scientific, utility, and business software.

The final technical report of the software cost estimation study consists of two volumes. Volume I contains the analytical study results, and Volume II is a management guide presenting a time phased overall methodology for estimating software development costs.

#0158

Hoffman, Lance J., L. A. Neitzel, "Inexact Analysis of Risk," Computer Security Manual, Vol 1, Spring 1981, (P).

## ABSTRACT:

Risk analysis often involves situations where little data are known on which to base estimates and where variances may be hard to find. Nevertheless, risk analysis has traditionally used numerical estimates and probability theory. An alternative approach presented here discusses risks in linguistic rather than numerical terms. An underlying calculus (which may, but need not, be based on the theory of fuzzy sets) can be used to calculate risks of subsystems. The relative chance of component failure, the severity of loss caused by such a failure, and the reliability of these estimates are each specified in linguistic terms. This paper suggests algorithms to combine these estimates and produce risk indicators. An example is given.

#0159

Putnam, L. H., R. W. Wolverton, "Quantitative Management: Software Cost Estimating," Computer Software and Applications Conference Tutorial, IEEE Computer Society, November 77, (R).

## ABSTRACT:

Two different perspectives are presented.

- (1) That of the government (or customer) going to an industrial contract software house to build an application system from a set of functional requirements and specifications that has been put together internally or by a separate project. The government organization needs to know the manpower, life cycle effort, development cost, development time and critical milestone events so that they can prepare their economic analysis to justify funding of the project. The government then is really interested in early macro-scopic estimators that will predict the overall system behavior in terms of the management parameters, manpower, cost and time. The government is also interested in systems that will provide management control and minimum cost throughout the system's operational life. A rationale and methodology to analyze software projects from this viewpoint are presented in the first two lectures.
- (2) That of the industrial contract software house charged with building a system for a government customer. The prior information needs of the system builder are different from the customer. The industrial organization needs the macro-scopic management parameters for costing at proposal time, but they also need far more detail relating to the phasing and work breakdown structure so that the various organizational entities (plus equipment and facilities) that will have to do the work can be allocated and scheduled. Cost control by work centers is important. Micro-scopic behavior is necessary to monitor progress at the project manager level so that day-to-day and week-by-week control can be exercised. Accordingly, lectures 3 and 4 deal with the philosophy, quantitative techniques and management methods to deal with software system building from the industrial builders viewpoint.

#0160

Directorate of Aerospace Safety, "Introduction to System Safety for Program Managers," Air Force Inspection and Safety Center (AFISC), Norton AFB, CA, 14 July 80, (R).

## ABSTRACT:

This guide was prepared to introduce program managers to system safety, its application and its importance during all phases of a system's life cycle. The guide is designed to outline the objectives of a system safety engineering program and provide management guidelines for their accomplishment. It is expected to provide only the essence of system safety in the briefest possible manner. The referenced sources/documents should be consulted for detail orientation and training.

#0161

Department of the Navy, "Automatic Data Processing Security Program," OPNAVINST 5239.1A, Office of the Chief of Naval Operations, Washington, D.C., 3 Aug 82, (R).

## ABSTRACT:

This instruction establishes the Department of the Navy (DON) Automatic Data Processing (ADP) Security Program. The DON ADP Security Manual, enclosure (2) of this instruction, consolidates all pertinent ADP security information on policies, procedures, and responsibilities for establishing and maintaining ADP security programs at all levels within the DON.

In implementing an activity ADP security program, one of the biggest obstacles facing the commanding officer is developing a command awareness of ADP security. The scope of ADP security covers more than just the traditional bounds of security classified information. It must safeguard Privacy Act data, sensitive financial information, For Official Use Only--indeed, all data and the ability to process data. The nature of the media--magnetic tape, disk packs, microfiche--allows a physical concentration of data. The number of users is large and constantly growing. There is a proliferation of peripheral terminals, networks, and systems. It is no longer simply a matter of card decks and batch processing at a few sites; it includes timesharing, word processors, and users, data, and programs of all different levels of classification.

How can an activity develop a program to tackle a problem of this magnitude? The DON approach is to analyze the problem and find solutions through a Risk Assessment. This involves systematically studying assets, their weaknesses and strengths, and possible threats; determining the probability of a successful attack occurring and the dollar value of its impact; and conducting a cost/benefit analysis of possible countermeasures to achieve an optimum level of security. The effectiveness of the countermeasures is evaluated through a security test and evaluation. A contingency plan formalizes procedures for continuity of ADP operations.

## COMMENT:

Reference Appendix E: Risk Assessment Methodology.

#0162

Air Force, "OT&E Reporting," Air Force Operational Test and Evaluation Center Regulation 55-1(C2), Chapter 6, 15 Mar 84, (R).

ABSTRACT:

This chapter outlines responsibilities and procedures for reporting (written and oral) an AFOTEC-conducted OT&E and for terminating AFOTEC involvement. The principal ways of reporting are activity (status) reports, execution briefings, interim and/or quick-look reports, final report briefings, final reports, lessons learned reports, and inputs to congressional data sheets. Report content, guidance for the report writer and/or briefer, typical report formats, and termination guidance are provided.

#0163

Parratto, S. L., D. E. Peercy, H. G. Pringle, "Computer System Security (CSS) Test and Evaluation (T&E) Life-Cycle Process Definition," (FINAL), BDM/A-84-0320-TR, The BDM Corporation, 31 Aug 84, (R).

## ABSTRACT:

The computer system security (CSS) test and evaluation (T&E) life cycle process, whether applied for acquisitions with embedded computer systems under AFR 800-series or for automated data system acquisitions under AFR 300-series, features:

- (1) Early, i.e., concept phase, conduct of CSS risk analysis, to enable definition of CSS residual risk which is the top level measure of effectiveness upon which the Designated Approving Authority (DAA) decision rests.
- (2) Re-iteration of portions of the CSS risk analysis as needed, due to changes in CSS provisions or concepts, or to data and findings from CSS T&E.
- (3) Earliest feasible and continued involvement of DAA(s) or their representatives.
- (4) Effective incorporation of CSS T&E within the established framework of T&E planning, documentation, and conduct, while accommodating CSS unique considerations and requirements.

Applicable CSS methodologies, techniques, and tools (MTT) to support the defined CSS T&E process are discussed. A framework of major CSS elements is presented within the three categories: administration, systems, and facilities. This framework includes CSS provisions for management, personnel, procedures, trusted computer systems, trusted communications systems, operations, emanations, physical facilities, environment, and contingency plans.

The methodologies, techniques and tools include the CSS risk analysis, Automated Threat Assessment Methodology (ATAM), IST/RAMP, fuzzy risk analysis, manual calculation of CSS risk, accreditation planning models, penetration testing, formal verification, evaluation criteria for computer systems (ORANGEBOOK) and for communications systems (GREENBOOK-DRAFT), application doctrine, software requirements engineering methodology (SREM), simulated emergency conditions, pass/fail criteria for CSS T&E plans, internal (program) testing, measures of coverage, software quality metrics, checklists and guidelines, COMSEC monitoring, TEMPEST, OPSEC survey or appraisal, audits, formal reviews and audits in the acquisition process, and performance/throughput testing as in acceptance testing.



The methodologies, techniques, and tools are related to the CSS T&E life cycle process by identification and discussion of their uses or roles in CSS T&E, where each is used in the process, the adequacy of each in defined use or role, and how they complement one another.

Future, planned research impacting the CSS T&E life cycle process is described, and additional needed research areas are identified.

#0164

Leibowitz, S., S. Parratto, D. Peercy, H. Pringle, J. Wiley, E. Witzke, "Computer System Security (CSS) Literature Review, Current Research Review, and Data Base Assemblage," (INTERIM), BDM/A-84-108-TR, The BDM Corporation, May 84, (R).

## ABSTRACT:

The literature search and review requested identification of key documents published by governmental agencies, civilian agencies, and specifically the WIS project. Literature searches of the Defense Technical Information Center (DTIC) and DIALOG data bases were conducted. A search and review of National Bureau of Standards (NBS) publications was done. Key documents from these searches were identified and ordered for inclusion in the CSS data base. A CSS documents list was received from the Aerospace Corporation library in late April, 1984. The final report bibliography will include any additional documents selected from that list. Researching the available CSS technology also involved fact-finding visits to a number of agencies, and identification of and discussions with CSS research and evaluation personnel. The basic form and content of this data base of CSS information is described in the sections of this report at a particular point in time, but will be augmented and updated as necessary to keep the data base current throughout this study and any subsequent related study efforts.

#0165

Pritsker, A. A. B.,- C. D. Regden, Introduction to Simulation and SLAM,  
New York: John Wiley, 1979, (B).

## ABSTRACT:

This textbook combines the presentation of a simulation language and the background material required for performing simulation projects. Thus, for the first time, a complete simulation methodology is available in textbook form.

SLAM, a new simulation language for alternative modeling, is described in detail. SLAM is an advanced FORTRAN based language that allows simulation models to be built based on three different world views. It provides network symbols for building graphical models that are easily translated into input statements for direct computer processing. It contains subprograms that support both discrete event and continuous model developments, and specifies the organizational structure for building such models. By combining network, discrete event, and continuous modeling capabilities, SLAM allows the systems analyst to develop models from a process-interaction, next-event, or activity-scanning perspective. The interfaces between the modeling approaches are explicitly defined to allow new conceptual views of systems to be explored.

#0166

Defense Systems Management College, Risk Assessment Techniques, Fort Belvoir, Virginia, July 1983, (R).

## ABSTRACT:

The primary objectives of this handbook are to make the reader aware of the risk assessment techniques being used by Department of Defense organizations, to alert the reader to the advantages and disadvantages of these techniques, and to assist him in applying risk assessment to his acquisition program.

The handbook is intended to be a practical guide and reference for program management personnel--not a textbook dealing with the theories supporting risk analysis, nor a user's manual for applying any particular techniques. Thus, the handbook is organized to address, in summary, the most important questions to program management personnel, i.e., Why do a risk assessment? What techniques are available? How do I select and implement a technique? These questions are answered in the first six chapters. This summary-level material is supported by a series of appendices that provide detailed discussions of the techniques in use, the service regulations pertaining to risk assessments, a glossary of terms, and a structured bibliography.

#0167

Huebner, W., D. Peercy, G. Richardson, "Software Supportability Risk Assessment in OT&E: An Evaluation of Risk Assessment Methodologies," (FINAL), BDM/A-84-496-TR, The BDM Corporation, 31 Aug 84, (R).

## ABSTRACT:

Assessing the software supportability risk of Air Force acquired systems is necessary to enable various decision makers to properly plan for system deployment. Risk assessment (RA) is required throughout the system acquisition life cycle. Since the perspective of OT&E is focused upon the overall system mission, including supportability, methods are required which provide software testers with areas which require testing emphasis and which provide decision makers with assessment of software and software support risk for production decisions. Due to the complexity of these requirements, it is necessary to determine the feasibility of developing and implementing a risk assessment model of software supportability with the proper system mission perspective to ultimately assist the top level decision maker.

This report contains the results of an analysis of literature and current research to determine the level of effort and usefulness of developing and implementing a risk assessment model for software supportability (RAMSS) in OT&E. This document also describes candidate RAMSS methodologies, techniques, and tools.

#0168

Huebner, W., D. Percy, G. Richardson, "Software Supportability Risk Assessment in OT&E: Measures for a Risk Assessment Model," (FINAL), BDM/A-84-565-TR, The BDM Corporation, 28 Sept 84, (R).

## ABSTRACT:

Assessing the software supportability risk of Air Force acquired systems is necessary to enable various decision makers to properly plan for system deployment. Risk assessment (RA) is required throughout the system acquisition life cycle. Since the perspective of OT&E is focused upon the overall system mission, including supportability, methods are required which provide software testers with areas which require testing emphasis and which provide decision makers with an assessment of software and software support risk for production decisions. Due to the complexity of these requirements, it is necessary to determine the feasibility of developing and implementing a risk assessment model of software supportability with the proper system mission perspective to ultimately assist the top level decision maker.

This report contains the results of an analysis of candidate measures of software supportability to determine the level of effort and usefulness of developing and implementing a risk assessment model for software supportability (RAMSS) in OT&E.

The document also describes the model framework and assesses the feasibility of model development and implementation under this framework.

#0169

Martin, J., C. McClure, Software Maintenance: The Problem and Its Solution, London: Prentice-Hall International, Inc., 1983, (B).

## ABSTRACT:

Software maintenance claims an extremely large share of the software dollar and is becoming the most expensive part of the software life cycle. Yet, although there are countless books and courses on systems analysis and design, the very important subject of software maintenance has been almost totally neglected. There is little understanding of what can be done to lessen the crippling maintenance problem.

In fact, much can be done. Widespread use of the techniques described in this book would cut the maintenance costs in most organizations to a fraction of what they are today.

This book deals with the maintenance of computer programming in data processing organizations. The authors describe the software maintenance problem, then discuss such methods as fourth-generation languages, prototyping, preprogrammed application packages, and contracting for maintainable software, as well as other tools, for solving the maintenance problem.

#0170

DeMillo, R., "A Risk Model for Software Testing," Georgia Institute of Technology, Briefing Slides, 20 July 84, (P).

ABSTRACT:

The GIT review primarily focused on briefing slides Dr. DeMillo had prepared summarizing research on "A Risk Model for Software Testing." The major emphasis in this research is to derive a method for determining an optimum software test strategy which would identify critical factors in decisions and reduce the decision risk. A framework for deriving such a method was presented. It is based upon decision theory using a "top down" approach. Some alternative strategies and test policies were presented in example form.



#0171

Yau, S. S., Methodology for Software Maintenance, Rome Air Development Center, Griffis AFB, NY, RADC-TR-83-262, Feb 84, (R).

ABSTRACT:

Improved techniques for specifying and implementing software modifications were developed including logical ripple effect analysis, logical and performance stability measures, and effective testing for software maintenance. An experiment was performed to analyze logical stability measurements.

#0226

Black, M. A., et al, "DoD/DON Requirements for Computer Risk Assessments," Monterey, CA: Naval Postgraduate School, AD-A132 202, Jun 83, (M).

## ABSTRACT:

The current methodology for conducting Computer Risk Assessments within the Department of the Navy is examined by studying the theories and philosophies that have evolved from the perspective of the Federal Government. A review of the Navy's attitude and procedures for contractual assessments is presented, along with a general framework for conducting an assessment of the computer systems at the Naval Postgraduate School. Attention is then focused on the relative merits of automated and manual Risk Assessment methods, followed by an outline of proposed design specifications for a decision support system.

#0227

Barber, D. E., "A-Guide for Developing an ADP Security Plan for Navy Finance Center," Monterey, CA: Naval Postgraduate School, AD-A127 244, Dec 82, (M).

## ABSTRACT

This paper is intended to be used as a guide by personnel at the Navy Finance Center, Cleveland, OH, in developing an Automatic Data Processing (ADP) Security Plan. The importance of the devotion of personnel, time and funds to ADP security planning has been emphasized. Individual chapters have been devoted to the elements that must be considered when developing an ADP security plan. They include risk assessment, physical security, systems security, contingency planning and the managerial procedures necessary for the implementation of an ADP security plan.

#0228

Helling, W. D., "Computer Security for the Computer Systems Manager,"  
Monterey, CA: Naval Postgraduate School, AD-A126 768, Dec 82, (M).

## ABSTRACT:

This thesis is a primer on the subject of computer security. It is written for the use of computer systems managers and addresses basic concepts of computer security and risk analysis. An example of the techniques employed by a typical military data processing center is included in the form of the written results of an actual on-site survey. Computer security is defined in the context of its scope and an analysis is made of those laws and regulations which direct the application of security measures into Automatic Data Processing systems. Finally, a list of some of the major threats to computer security and the countermeasures typically employed to combat those threats is presented.

#0229

SDC, "Risk Assessment Methodology," McLean, VA: System Development Corp., AD-A072 249, Jul 79, (M).

## ABSTRACT:

This report treats risk assessment as an organized examination of events and conditions that could harm a Navy ADP system or facility. A comprehensive risk assessment does the following:

- a) Identifies conditions or potential events that threaten harm to the ADP system or facility, and evaluates the seriousness of these threats.
- b) Identifies and evaluates the properties and importance of all of the resources of the ADP system or facility, i.e., its assets.
- c) Estimates the Annual Loss Expectancy (ALE) of the ADP system or facility from the threats being realized.
- d) Estimates the level of risk to which classified, sensitive, or mission-essential assets are exposed
- e) Identifies the most dangerous or costly weaknesses of the ADP system or facility, and recommends the most cost-effective way to remedy them.

Risk assessment involves detailed examination of the threats to the ADP system or facility; the missions, assets, and procedures of the system or facility; and the operational and security weaknesses of the system or facility. Changes in the mission, configuration, location, or procedures of the system or facility are cause for a review of the existing risk assessment.

#0230

SDC, "Countermeasures," McLean, VA: System Development Corp., AD-A072 245, Jun 79, (M).

ABSTRACT:

This appendix describes countermeasures that will reduce the vulnerability of an ADP facility. The countermeasures described herein are a representative group for improving overall computer security. They are to be used to assist ADP installations in performing a risk assessment.

#0231

Bushkin, A. A., "A Framework for Computer Security" (Revised Edition),  
Santa Monica, CA: System Development Corp. AD-A025 356, Jun 75, (M).

ABSTRACT:

This document presents:

- a) An overview of the computer security problem.
- b) An interrelated set of Axioms and Principles of Computer Security as the beginning of a top-down, structured approach to the computer security problem.
- c) A discussion of the issues involved with using these axioms and principles as the basis for additional research leading to the development of guidelines, standards, and measures in the areas of:
  - 1) System design and implementation
  - 2) Procurement specifications and acceptance criteria
  - 3) Daily operations
  - 4) Assessment of existing system (with a special emphasis on the attainment of an acceptable level of risk).

#0232

Schacht, J. M., S. M. Goheen, and R. D. Rhode, "User Requirements for Computer Security," Bedford, MA: MITRE Corp., AD-A073 101, May 79, (M).

ABSTRACT:

The various approaches to secure computer processing of classified information are summarized and contrasted. Dedicated processing, period processing, jobstream separation, multilevel security, and other approaches are characterized according to cost and risk factors, and data-sharing capabilities.



#0233

Campbell, R. P., and G. A. Sands, "A Modular Approach to Computer Security Risk Management," Montvale, NJ: AFIPS NCC, 48 293-303, Jun 79, (P).

## ABSTRACT:

The Risk Management Model (RMM) presented in this article decomposes into sufficient detail to allow depth of analysis to vary with the specific nature of the problem. The less sensitive operation will require lesser analysis, while the more sensitive will require considerably more extensive analysis. The RMM is composed of eight basic steps--Value Analysis, Threat Identification/Analysis, Vulnerability Analysis, Risk Analysis, Risk Assessment, Management Decision, Control Implementation and Effectiveness Review. Each step is described in detail.

#0234

W. Neugent, "Acceptance Criteria for Computer Security," Arlington, VA: AFIPS Press, AFIPS NCC 51, Aug 82, (P).

## ABSTRACT:

Acceptance criteria define the degree of duality required and identify areas to be examined in evaluating the degree of quality. Three categories of computer security acceptance criteria are proposed: functionality, performance, and development method. Each is further divided into sub-categories. Aids in formulating requirements and criteria are noted, including the use of organizational policies and risk analysis methods. Quantification is shown as a volatile tool, since numbers are often treated as single data points rather than as ranges. A set of principles is presented, to be followed in formulating acceptance criteria. Illustrative principles are as follows:

- a) Get a good start
- b) Make sure everyone understands
- c) Distinguish shall from should
- d) Explain why.

The acceptance determination process is discussed, a key point being that intermediate products must be approved. The value of acceptance criteria is in making the product better and the judgement easier.

## COMMENT:

Mr. Neugent is the author of general computer security papers as well as WIS security-related documents such as "WIS ADP Security Strategy" (Draft). This paper presents a "quality criteria" structure for computer security acceptance which can be patterned along the lines of earlier work by AFOTEC in software supportability evaluation.

#0235

Air Force, "Automatic Data Processing (ADP) Security Policy Procedures and Responsibilities," AFR 205-16, Washington, D.C.: Department of the Air Force, Headquarters, U.S. Air Force, Aug 84, (R).

**ABSTRACT:**

This regulation replaces AFR 300-8. With respect to AFR 300-3, it incorporates additional policy on the protection of sensitive unclassified and critical data and systems; adds security requirements for word processing systems; redefines existing responsibilities for the protection of sensitive unclassified and critical data and systems; adds responsibilities for program or project managers, ADPS manager, ADS managers, systems analysts and programming personnel, and on the control and prevention of computer abuse; updates terminology on the control of compromising emission; incorporates policy on the inclusion of security throughout the ADP life cycle, including concepts, policy and guidance on risk management, certification, and approval; replaces the concept of data processing installation (DPI) by automatic data processing facility (ADPF); updates guidance on declassifying plated wire memory and adds guidance on declassifying new technology memory devices; adds guidance on addressing security in the ADP system life cycle; adds guidance for performing risk analysis; and adds sample letters for the certification and approval process.

**COMMENT:**

Security Test and Evaluation (ST&E), in this regulation, is one of four risk analysis modules. The other modules addressed in the extensive attachment 5, Guidance for Performing Risk Analysis, are Sensitivity and Criticality Assessment, Risk Assessment, and Economic Assessment. This is a key document for CSS, which provides policy, guidelines, procedures, and responsibilities delineations.

#0236

Air Force, "Management of Operational Test and Evaluation," Washington, D.C.: Department of the Air Force, Headquarters U.S. Air Force, Jun 79, (P).

## ABSTRACT:

This manual is designed to explain the operational test and evaluation (OT&E) program, and how it relates to other Air Force and Department of Defense (DoD) activities. It outlines the principles and procedures that will promote consistent OT&E management throughout the Air Force. A method for storing data is described which permits recovery of all data on a track or other size physical record. It establishes guidelines for standardizing the planning, conducting, and reporting of OT&E programs in the Air Force; however, because the scope of these programs varies, judgement must be used in applying these guidelines to each individual program. The major commands may set specific command policies and procedures not only to implement this manual, but to provide for specific procedures and tests outside its scope. This volume is a general explanation of the OT&E process, and it is directed at all levels of management. Individual chapters address OT&E evolution, organization and management, types, objectives, role in requirements and acquisition process, funding, planning and management, test execution, and reporting (deficiencies and test).

## COMMENT:

CSS scope and methodology will be derived in a manner compatible with the general framework and provisions for OT&E such as are provided in this AFM and in AFR 80-14 (separate listing). These references would also be of interest to agencies associated with CSS, but which are relatively unfamiliar with OT&E.

#0237

Air Force, "Software OT&E Guidelines Volume II Handbook for the Deputy for Software Evaluation," Kirtland AFB, NM: Air Force Test and Evaluation Center, Sep 81, (P).

## ABSTRACT:

This handbook provides general information, software OT&E concerns and techniques, and software evaluation lessons learned. Elements of OT&E for embedded computer systems are provided, including software suitability evaluation. Software effectiveness consideration encompasses software performance, software/operator interface, software maturity evaluation, and embedded computer system peculiar evaluations.

## COMMENT:

A similar "handbook" may be appropriate, for Computer System Security (CSS) OT&E Guidelines. This Handbook is a valuable example of such a product tailored to AFOTEC needs. In addition, software effectiveness and suitability shortcoming could impact CSS.

#0238

DoD, "Test and Evaluation," DODD 5000.3 Washington, D.C.: Department of Defense, Dec 79, (P).

## ABSTRACT:

This directive re-issues and establishes policy for the conduct of test and evaluation in the acquisition of defense systems; designates the Director Defense Test and Evaluation (DDTE) as having overall responsibility for test and evaluation matters within the Department of Defense; defines responsibilities of the DDTE, organization of the Joint Chiefs of Staff (OJCS) and DoD Components; and provides guidance for the preparation and submission of Test and Evaluation Master Plans. The provisions of this directive apply to the Military Departments and the Defense Agencies (hereafter referred to as "DoD Components"), the Office of the Secretary of Defense (OSD), the OJCS, and the Unified and Specified Commands.

As used herein, the term "Military Services" refers to the Army, Navy, Air Force, and Marine Corps.

These provisions encompass major defense system acquisition programs, as designated by the Secretary of Defense under DoD Directive 5000.1, and apply to all DoD Components that are responsible for such programs. In addition, the management of system programs not designated as major system acquisitions shall be guided by the principles set forth in this Directive.

The provisions of this Directive apply to the software components of defense systems as well as to hardware components. Quantitative and demonstrable performance objectives and evaluation criteria shall be established for computer software during each system acquisition phase. Testing shall be structured to demonstrate that software has reached a level of maturity appropriate to each phase. Such performance objectives and evaluation criteria shall be established for both full-system and casualty mode operation. For embedded software, performance objectives and evaluation criteria shall be included in the performance objectives and evaluation criteria of the overall system.

Decisions to proceed from one phase of software development to the next will be based on quantitative demonstration of adequate software performance through appropriate T&E. Before release for operational use, software developed for either new or existing systems shall undergo sufficient operational testing as part of the total system to provide a valid estimate of system effectiveness and suitability in the operational environment. Such testing shall include combined hardware/software and interface testing under realistic conditions, using typical operator personnel. The evaluation of test results shall include an assessment of operational performance under other possible conditions which were not employed, but which could occur during operational use.

The OT&E agencies shall participate in the early stages of software planning and development to ensure that adequate consideration is given to the system's operational use and environment, and early development of operational test objectives and evaluation criteria.

COMMENT:

This is the primary DoD directive for test and evaluation, including T&E of hardware and software.

#0239

Air Force, "Test and Evaluation," AFR 80-14, Washington, D.C.:  
Department of the Air Force, Headquarters U.S. Air Force, Sep 80, (P).

## ABSTRACT:

AFR 80-14 outlines policy for test and evaluation (T&E) activities during the development, production, and deployment of defense systems in the Air Force. It assigns T&E responsibilities to the implementing the Air Force Test and Evaluation Center (AFTEC), and the operating and supporting commands. The regulation implements DoDD 5000.3, 26 December, 1979. The applicability of AFR 80-14 extends to new or existing systems. A computer system, subsystem, or component; software computer program configuration item, or a computer program component of a defense system are also under the purview of the regulation. Concepts and general policy guidance topics include, for example: Test and Evaluation Master Plan (TEMP), Documentation Requirements, Management of OT&E, OT&E Objectives, and separate Initial Operational Test and Evaluation (IOT&E). Responsibilities are assigned for HQ USAF, implementing command, OT&E command, AFTEC, MAJCOMs, operating commands, AFLC, ATC, and ESC.

## COMMENT:

This is the prime USAF directive for T&E, including OT&E. Attachment 1 provides DoDD 5000.3, Test and Evaluation, 26 December 1979, with specific guidance for T&E of computer software (see separate listing for DoDD 5000.3).



#0240

NBS, "Guideline for Automatic Data Processing Risk Analysis," U.S. Department of Commerce National Bureau of Standards, FIPS PUB 65, Aug 79, (P).

## ABSTRACT:

This document presents a technique for conducting a risk analysis of an ADP facility and related assets. Risk analysis produces annual loss exposure (ALE) values based on estimated costs and potential losses. The ALE values are fundamental to the cost effective selection of safeguards for the security of the facility. An ADP facility of a hypothetical government agency is used for an example. The characteristics and attributes of a computer system which must be known in order to perform risk analysis are described and an example is given of the process of analyzing some of the assets showing how the risk analysis can be handled. The ALE is the product of estimated impact in dollars (I) and estimated frequency of occurrence per year (F). Indices "i" and "f" are provided in a table, for different orders (i.e., magnitudes) of dollar loss and frequency of occurrence. An alternate formula is:

$$ALE = \frac{10^{(f+i-3)}}{3}$$

using the table of indices. A risk analysis worksheet provides for ALE calculations for three categories: data integrity, data confidentiality, and ADP availability.

## COMMENT:

The document is of interest since it describes the risk analysis procedures and techniques for ADP security in Federal agencies other than those with specific, specialized risk analysis approaches such as those of USAF AFRs 300-8 and 205-16/205-X. (The ALE as described is not sufficient for USAF CSS.)

#0241

Orceyre, M. J., R. H. Courtney, Jr., R. Bolotsky, "Considerations in the Selection of Security Measures for Automatic Data Processing Systems," Department of Commerce, National Bureau of Standards, NBS SP 500-33, Jun 1978 (R).

## ABSTRACT:

This document presents an overview of currently known methods and techniques for securing information processed by computers and transmitted via telecommunication lines. Originally contributed by the authors to the Federal Information Processing Standards Task Group 15 on Computer Systems Security, this revised document is intended as a followup document to Automatic Data Processing Risk Assessment (NBSIR 77-1228). This publication summarizes protective measures which aid in identifying controls already in use and selecting further safeguards to offset existing risks and potential losses identified by a risk analysis. Information in this document was submitted to Federal Information Processing Standards Task Group 15 (Computer Systems Security) as an appendix to a risk analysis document authored by Robert H. Courtney, Jr. The information was considered valuable by the participants as a tutorial on what to consider using for security improvements after risk analysis has been performed. The steps of a computer security program include: perform a security risk analysis; consider all security measures available; select those measures that minimize the risk at a minimum cost; implement those measures that are feasible; evaluate their effectiveness and actual cost; restart the process. Information in this document is intended to outline those security measures which may be selected and used in this process.

## COMMENT:

The content includes sections on authorization, surveillance, identification, cryptography, system integrity, distributed processing and auditing. The document can contribute to knowledge of risk assessment and evaluation evolution.

END  
DATE  
FILMED  
5-88  
DTIC