

AD-A123 840

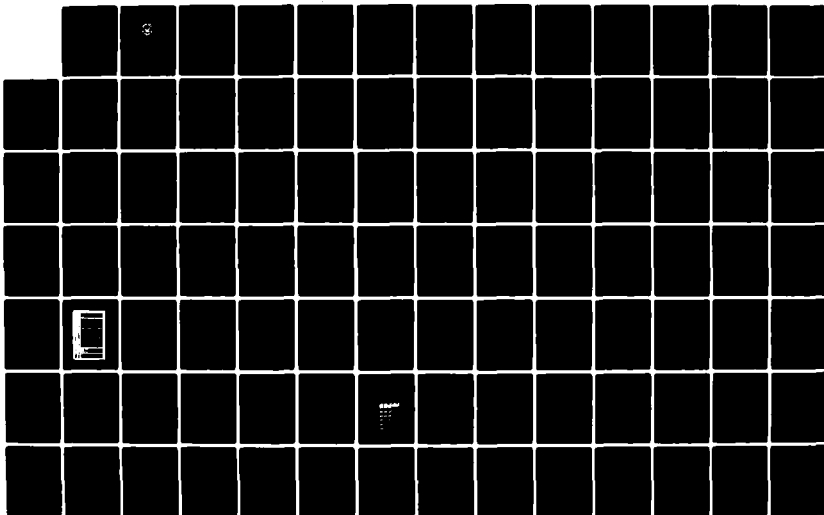
SOFTWARE QUALITY ASSURANCE AND THE FLEET MATERIAL
SUPPORT ENVIRONMENT(U) NAVAL POSTGRADUATE SCHOOL
MONTEREY CA J H HARRINGTON JUN 82

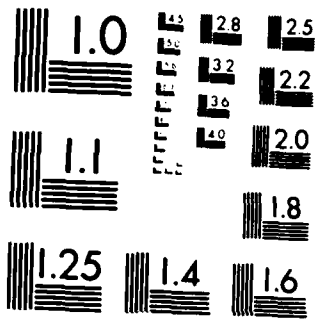
1/2

UNCLASSIFIED

F/G 14/4

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

2

NAVAL POSTGRADUATE SCHOOL Monterey, California

ADA 123840



THESIS

SOFTWARE QUALITY ASSURANCE AND THE FLEET
MATERIAL SUPPORT ENVIRONMENT

by

Joseph H. Harrington, Jr.

June 1982

Thesis Advisor:

Norm R. Lyons

Approved for public release; distribution unlimited

DTIC FILE COPY

DTIC
SEARCHED
SERIALIZED
INDEXED

83 01 27 04 2^A

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A123840	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Software Quality Assurance and the Fleet Material Support Environment		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis June 1982
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Joseph H. Harrington, Jr.		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE June 1982
		13. NUMBER OF PAGES 129
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for Public Release; Distribution Unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Software Quality Assurance Software Quality Control		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This thesis investigates the trends of thought and actual practices of commercial computer companies in the area of software quality assurance. This is done to see if any of these practices could be utilized in the Fleet Material Support Office (FMSO) environment. This was accomplished by personal interview of software quality assurance personnel in a few randomly selected computer companies and comparing their quality assurance programs to that of FMSO. A		

The following companies were selected:

1. International Business Machines (IBM) Corporation,
2. TRW Incorporated;
3. Hewlett Packard Company,
4. Amdahl Corporation;
5. Software Research Associates (SRA).

Results indicate that the greatest differences between the commercial world and the FMSO environment are in management's view of what role or function a quality assurance group should take, staff as compared to line, and this group's interface with the software design and development personnel.



Approved for public release: distribution unlimited

Software Quality Assurance and the Fleet Material Support
Environment

by

Joseph H. Harrington, Jr.
Lieutenant, United States Navy
B.S., United States Naval Academy, 1977

Submitted in partial fulfillment of the
requirements for the degree of

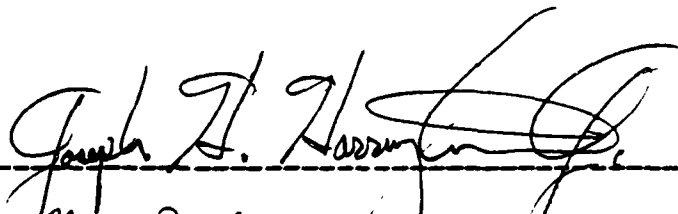
MASTER OF SCIENCE IN MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL

June 1982

Author



Approved by:



Thesis Advisor



Second Reader



Chairman, Department of Administrative Sciences



Dean of Information and Policy Sciences

ABSTRACT

This thesis investigates the trends of thought and actual practices of commercial computer companies in the area of software quality assurance. This is done to see if any of these practices could be utilized in the Fleet Material Support Office (FMSO) environment. This was accomplished by personal interview of software quality assurance personnel in a few randomly selected computer companies and comparing their quality assurance programs to that of FMSO. The following companies were selected:

1. International Business Machines (IBM) Corporation,
2. TRW Incorporated,
3. Hewlett Packard Company,
4. Andahl Corporation,
5. Software Research Associates (SRA).

Results indicate that the greatest differences between the commercial world and the FMSO environment are in management's view of what role or function a quality assurance group should take, staff as compared to line, and this group's interface with the software design and development personnel.

TABLE OF CONTENTS

I.	INTRODUCTION	9
	A. THE PROBLEM	9
	B. PURPOSE	10
	C. METHODOLOGY	11
	D. STRUCTURE OF THE THESIS	12
II.	SURVEY OF LITERATURE	13
	A. DEFINITIONS	13
	B. QUALITY FACTORS AND CRITERIA	17
	C. GENERAL RESPONSIBILITIES AND ORGANIZATION	17
	D. SUMMARY	25
III.	FLEET MATERIAL SUPPORT OFFICE	27
	A. HISTORY	27
	B. ORGANIZATIONAL STRUCTURE	28
	C. THE CDA	31
	D. SYSTEM DESIGN AND QUALITY ASSURANCE PROCESS	41
	E. CONCLUSION	50
IV.	INTERVIEWS	51
	A. HEWLETT PACKARD	51
	1. Organization	53
	2. Quality Assurance and Designer Interface	57
	3. Operations	60
	B. TRW	61
	1. Organization	61
	2. Management and Software Areas of the Project	66
	C. IBM	72
	1. Organization	72
	2. Quality Assurance and Design Interface	75
	3. Operations	79
	D. AMDAHL	79
	1. Organization	80

2. Development Interface	80
3. Operations	80
V. ANALYSIS CONCLUSION AND RECOMMENDATIONS	83
A. QUESTIONS FROM INTERVIEW:	83
B. SUMMARY CONCLUSIONS	90
C. RECOMMENDATIONS	91
APPENDIX A: FMSO SYSTEM DEVELOPMENT PROCESS	94
APPENDIX B: SOFTWARE RESEARCH ASSOCIATES (SRA)	100
LIST OF REFERENCES	126
INITIAL DISTRIBUTION LIST	129

LIST OF FIGURES

2.1	Software Development Process	15
2.2	The Price of Procrastination in Error Detection	16
2.3	Definition of Software Quality Factors	18
2.4	Allocation of Software Quality Factors to Product Activity	19
2.5	Relationship of Criteria to Software Quality Factors	20
2.6	Criteria Definitions for Software Quality Factors	22
3.1	Index of FMSO CDA Systems Users	29
3.2	FMSO Organization	30
3.3	Code 92 Organization	32
3.4	Code 94 Organization	36
3.5	Code 95 Organization	37
3.6	Code 96 Organization	39
3.7	Code 97 Organization	40
3.8	Code 98 Organization	42
3.9	Quality Assurance Checklist	44
4.1	Hewlett Packard Organization	54
4.2	Hewlett Packard Data Systems Division	55
4.3	Hewlett Packard Products Assurance Group	56
4.4	Hewlett Packard Software Development Cycle	58
4.5	TRW Defense Systems Group	62
4.6	TRW Products Assurance Department	63
4.7	TRW Product Assurance Standardized Project WBS	64
4.8	TRW Quality Assurance Standardized Project WBS	65
4.9	TRW Quality Assurance Project WBS - Management Detail	67
4.10	TRW Quality Assurance Project WBS - Software Detail	70

4.11	IBM Organization	73
4.12	IBM General Products Division	74
4.13	IBM Software Area of Development	74
4.14	IBM Software Development Process	76
4.15	Amdahl Software Department	81

I. INTRODUCTION

A. THE PROBLEM

Software quality assurance is the planned and systematic actions required to provide adequate confidence that software products conform to standards established by the company developing the product and the contractual requirements provided by the customer [Ref. 1]. This phenomenon crosses all customer boundaries: commercial, industrial, military, other governments; and crosses different application types: operating systems, information systems, process control, command and control, communication, business systems, etc [Ref. 2].

In the United States Navy (USN), the office in charge of design, development and life cycle maintenance of the supply system computer network is the Fleet Material Support Office (FMSO) in Mechanicsburg, Pennsylvania. On 29 October 1981, FMSO's Commanding Officer established a quality program task group which consisted of Automatic Data Processing (ADP) technical personnel from each of its Central Design Agency (CDA) departments and supporting departments. Its purpose was to consider quality in a broad sense as it related to the ADP system development process and to outline a general plan for a viable and continuing quality program. The group's main objectives were to provide recommendations that would improve the quality of FMSO products, account for this quality process and sustain it throughout the product's life cycle. The conclusions of the task group were:

1. Quality improvement was possible in the FMSO environment.

2. Quality accountability was required and was becoming increasingly important. Correctly performed, measurement would result in an effective and accountable quality process.

3. The ability to sustain acceptable levels of quality in an environment of changing technology can be accommodated through the iterative accounting and analysis of productivity and inventory characteristics. [Ref. 3]

During this same time period, two other special projects were receiving major attention by FMSO. One is the Resolicitation project which identifies the computer requirements at the two Inventory Control Points (ICP's) in the 1980's and beyond, taking into account both the near saturation of the present Univac 494 computers at the ICP's and their obsolescence. The other project, called "Resystemization," is also a massive undertaking as it will eventually result in new software or computer programs for the ICP's. Talks between the author and FMSO's Commanding Officer indicate that this project [Ref. 4] gives FMSO more incentive to take a serious look at its quality assurance program.

B. PURPOSE

The purpose of this thesis is to investigate the methods used by large commercial computer companies in the area of software quality assurance. The primary objective is to see if any of these practices can be utilized in FMSO's environment.

C. METHODOLOGY

The procedure used to accomplish the thesis objective was to interview personnel from the various computer companies. The following companies' personnel were interviewed:

1. International Business Machines (IBM) Corporation,
2. TRW Incorporated,
3. Hewlett Packard Company,
4. Amdahl Corporation,
5. Software Research Associates (SRA).

These companies were chosen because they are located near the Naval Postgraduate School in Monterey, CA and they give a broad view of the computer software industry. The following questions were asked at the interview:

1. Where does the quality assurance group fit into the organization?
2. What type of authority/power does the quality assurance group have over the software product?
3. What qualifications do the people in the quality assurance group have?
4. How does the quality assurance group interface with the design/development group?
5. What tools, methodologies, or techniques does the quality assurance group use to do their job?
6. Are historical records kept of problems with software products after their release and who in the company's organization keeps them?
7. Who handles problems with software after release, and how are such problems handled?
8. If a brand new product is designed, who in the company's organization trains the customer on this product?

The data was then compared with existing practices at FMSO and conclusions and recommendations were rendered.

D. STRUCTURE OF THE THESIS

Chapter I, the introduction to the thesis, presents the thesis objective and methodology. Chapter II presents a general literature review of the problem of quality assurance and the factors that are taken into consideration when defining it. Chapter III addresses the FMS environment and its process of quality control. Chapter IV presents the interviews conducted with the personnel of the five computer organizations as to their software quality assurance organizations and how they work. The final chapter offers a summary of these interviews and provides recommendations on how these ideas might be applied at FMSO.

II. SURVEY OF LITERATURE

Chapter II deals with the problem of software quality assurance. After a computer search to find current literature on this subject, it was discovered that all authors of these writings failed to agree on the definition of pertinent terms. In order to define the terms relevant to this thesis, the author presents the following definitions.

A. DEFINITIONS

Software is a set of coded instructions which are supplied to and operate with the computer hardware to cause the hardware to perform the functions defined in the instructions. [Ref. 5]

A system, as defined by the Fleet Material Support Office, is an organized set of Automatic Data Processing (ADP) hardware, environmental and application software, and documented procedures designed to automate the basic management and operating processes for a customer site or group of customer sites with common mission responsibilities [Ref. 6]. "Documented procedures," as used above, refers to the applicable ADP-related and non-ADP-related procedures established to support the hardware and software aspects of the system, e.g. the computer operation manual and the users manual [Ref. 7].

Quality assurance of hardware has been successfully accomplished for many years, but there are major differences between hardware and software:

1. Software development specifications are usually not as specific as those for hardware. Precise sounding

terms with unspecified definitions such as "optimum" or "99.9 percent reliable" are used which are potential seeds of dissension or lawsuits once the software is produced.

2. Software product (built-to) specifications are usually less rigorous.
3. The software development process is also the production process because there are no bread boards, brass boards, phototypes or pre-production models to use.
4. The production of software (code) is neither a fully constrained nor a uniquely defined process.
5. The software product itself (code) is essentially an intangible substance with form, content, and functions manifested via images.
5. Software problem fixes always result in a product configuration change. [Ref. 8]

A basic software development process is shown in Figure 2.1. Corporate analyses of life-cycle cost have shown that the cost of maintenance and redesign exceed the cost of initial development and that the cost of fixing errors after the software is operational is up to 30 times greater than for correcting errors during system testing. Figure 2.2 shows a summary of experience at International Business Machines, General Telecommunications Equipment (GTE), and TRW on the relative cost of correcting software errors as a function of the phase in which they are corrected. Figure 2.2 suggests that it pays to invest in one-man hour searching for errors during the early stages of development than to spend 100-man hours correcting errors after the system is in operation. [Ref. 9]

SOFTWARE DEVELOPMENT PROCESS

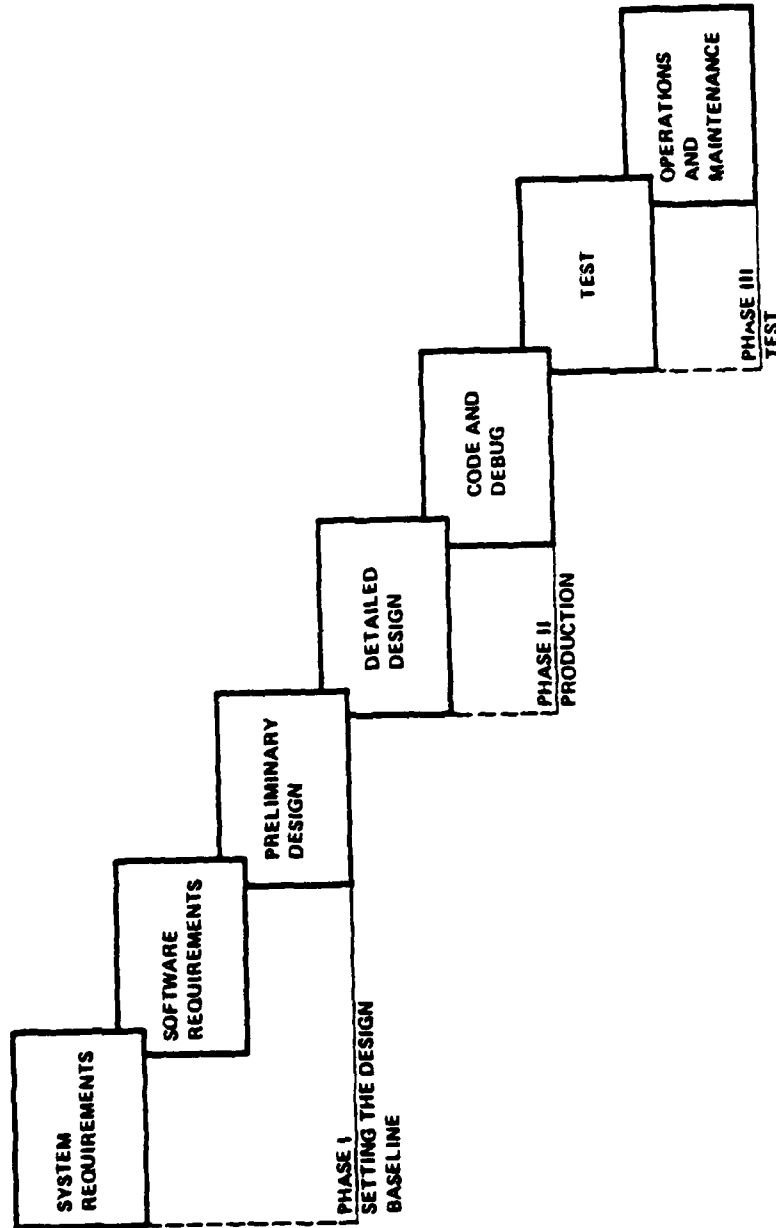


FIGURE 2.1 Software Development Process
SOURCE: Mr. Kenehan's Presentation at Internal Meeting in TRW,
December, 1981

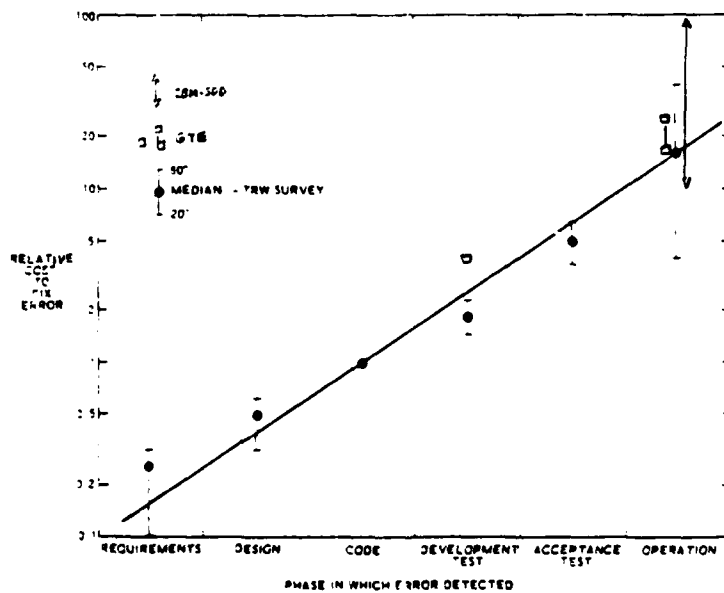


FIGURE 2.2 The Price of Procrastination in Error Detection
 SOURCE: Dr. Barry W. Boehm's Article on Software Engineering,
 1 June 1981

B. QUALITY FACTORS AND CRITERIA

Specific factors contribute to the quality of software. Eleven of these factors are defined in Figure 2.3. The rationale [Ref. 10] behind the choice of these is one of utility; each factor identified could be applied to a production environment. The interaction of support groups within an operational environment involves three distinct activities: product operation, product revision, and product transition. Figure 2.4 shows a conceptual scheme with these three activities and some related questions which involve the quality factors [Ref. 11].

These quality factors can be further broken down into criteria which could be used for other purposes. First, a set of criteria for each factor further defines the factor. Second, the criteria which affect more than one factor help describe the relationships between the factors, and the criteria establish a working hierarchical framework for factors in software quality. These criteria are defined and their relations to factors are shown in Figures 2.5 and 2.6. Lastly, with the use [Ref. 12] of these factors and their criteria, a possible numerical value may be added to help forecast the quality of the software during its development cycle. This is the goal of software metrics, a tool used by some companies for this purpose [Ref. 13].

C. GENERAL RESPONSIBILITIES AND ORGANIZATION

Companies are finding that it is advantageous, from both product quality and cost-effectiveness standpoints, to have an explicit quality assurance activity on their software projects [Ref. 14]. The tasks of this activity are usually tailored to the project and depend on size and scope. This approach has proved effective in ensuring that the project

<u>CORRECTNESS</u>	Extent to which a program satisfies its specifications and fulfills the user's mission objectives.
<u>RELIABILITY</u>	Extent to which a program can be expected to perform its intended function with required precision.
<u>EFFICIENCY</u>	The amount of computing resources and code required by a program to perform a function.
<u>INTEGRITY</u>	Extent to which access to software or data by unauthorized persons can be controlled.
<u>USABILITY</u>	Effort required to learn, operate, prepare input, and interpret output of a program.
<u>MAINTAINABILITY</u>	Effort required to locate and fix an error in an operational program.
<u>TESTABILITY</u>	Effort required to test a program to insure it performs its intended function.
<u>FLEXIBILITY</u>	Effort required to modify an operational program.
<u>PORTABILITY</u>	Effort required to transfer a program from one hardware configuration and/or software system environment to another.
<u>REUSABILITY</u>	Extent to which a program can be used in other applications related to the packaging and scope of the functions that programs perform.
<u>INTEROPERABILITY</u>	Effort required to couple one system with another.

FIGURE 2.3 Definition of Software Quality Factors

SOURCE: Macabe's Book on Software Quality Assurance - A Survey

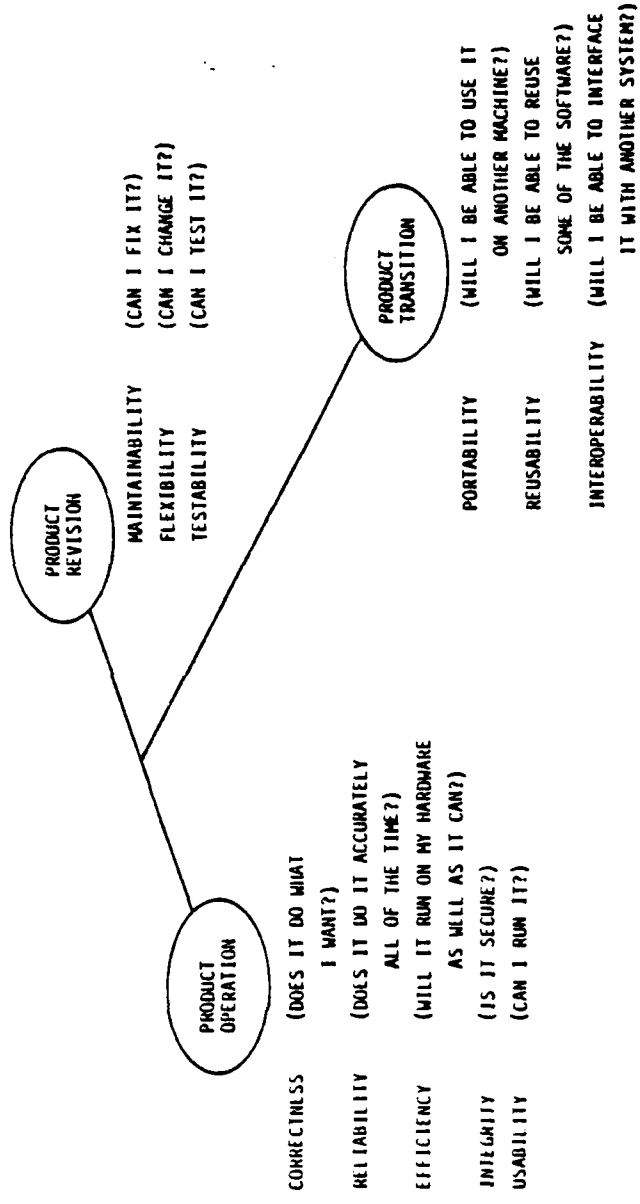


FIGURE 2.4 Allocation of Software Quality Factors to Product Activity
 SOURCE: Macabe's Book on Software Quality Assurance - A Survey

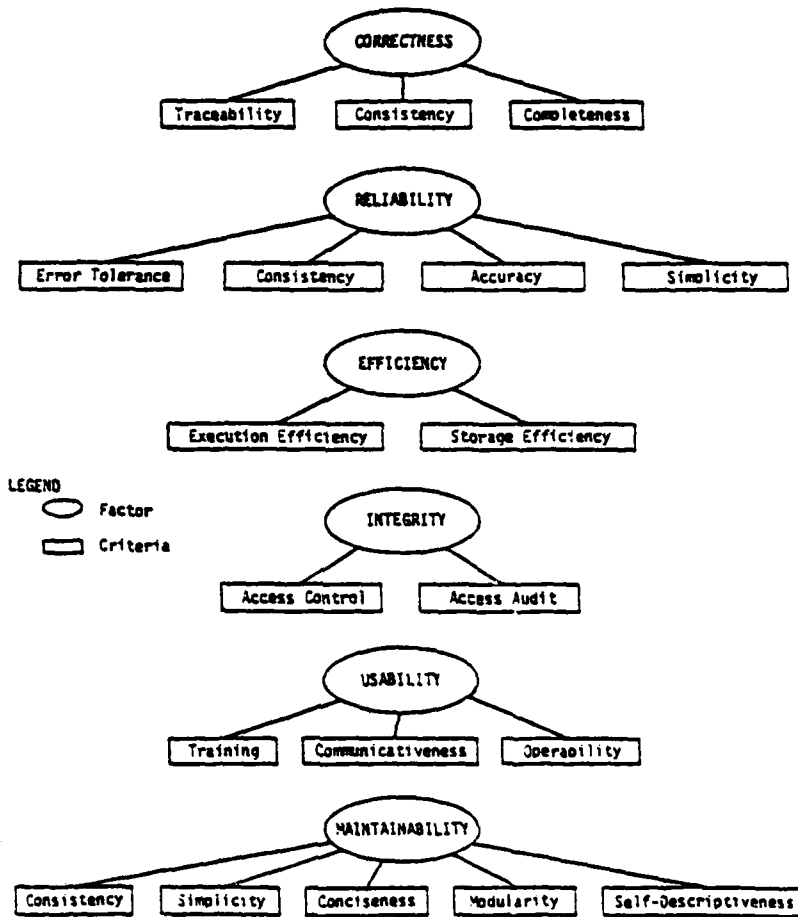


FIGURE 2.5 Relationship of Criteria to Software Quality Factors

SOURCE: Macabe's Book on Software Quality Assurance - A Survey

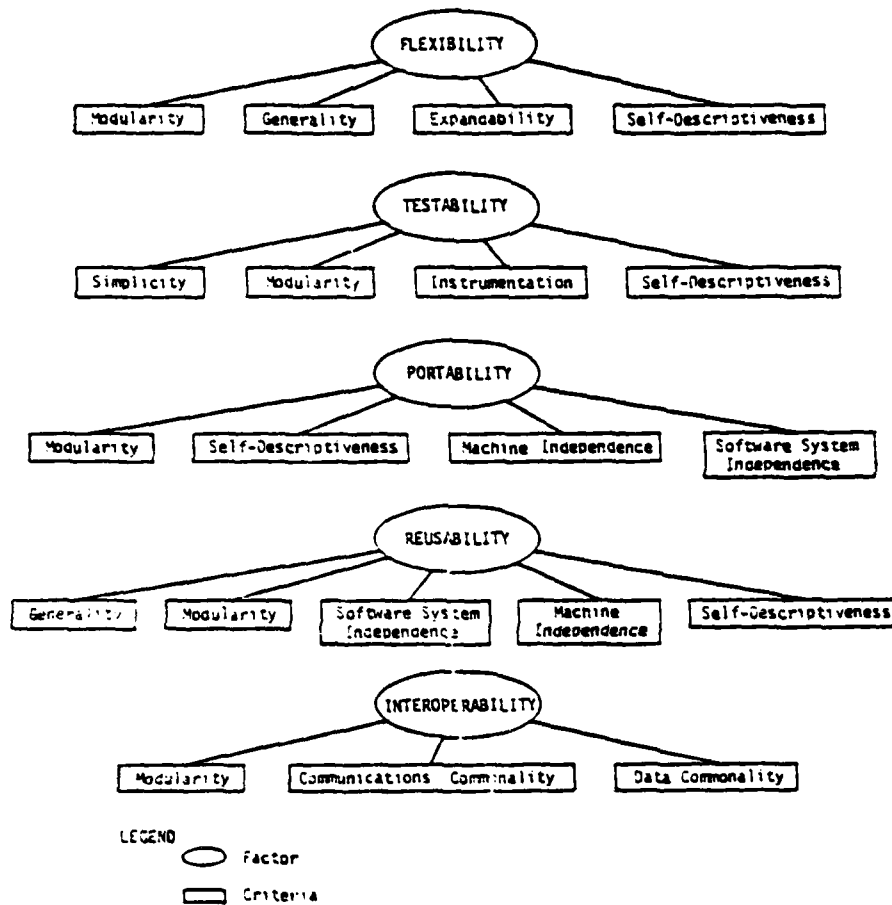


FIGURE 2.5 Relationship of Criteria to Software Quality Factors (Contd.)

SOURCE: Macabe's Book on Software Quality Assurance - A Survey

CRITERION	DEFINITION	RELATED FACTORS
TRACEABILITY	Those attributes of the software that provide a thread from the requirements to the implementation with respect to the specific development and operational environment.	Correctness
COMPLETENESS	Those attributes of the software that provide full implementation of the functions required.	Correctness
CONSISTENCY	Those attributes of the software that provide uniform design and implementation techniques and notation.	Correctness Reliability Maintainability
ACCURACY	Those attributes of the software that provide the required precision in calculations and outputs.	Reliability
ERROR TOLERANCE	Those attributes of the software that provide continuity of operation under nonnominal conditions.	Reliability
SIMPLICITY	Those attributes of the software that provide implementation of functions in the most understandable manner. (Usually avoidance of practices which increase complexity.)	Reliability Maintainability Testability
MODULARITY	Those attributes of the software that provide a structure of highly independent modules.	Maintainability Flexibility Testability Portability Reusability Interoperability
GENERALITY	Those attributes of the software that provide breadth to the functions performed.	Flexibility Reusability
EXPANDABILITY	Those attributes of the software that provide for expansion of data storage requirements or computational functions.	Flexibility
INSTRUMENTATION	Those attributes of the software that provide for the measurement of usage or identification of errors.	Testability
SELF-DESCRIPTIVENESS	Those attributes of the software that provide explanation of the implementation of a function.	Flexibility Maintainability Testability Portability Reusability

FIGURE 2.6 Criteria Definitions for Software Quality Factors
SOURCE: Macabe's Book on Software Quality Assurance - A Survey

CRITERION	DEFINITION	RELATED FACTORS
EXECUTION EFFICIENCY	Those attributes of the software that provide for minimum processing time.	Efficiency
STORAGE EFFICIENCY	Those attributes of the software that provide for minimum storage requirements during operation.	Efficiency
ACCESS CONTROL	Those attributes of the software that provide for control of the access of software and data.	Integrity
ACCESS AUDIT	Those attributes of the software that provide for an audit of the access of software and data.	Integrity
OPERABILITY	Those attributes of the software that determine operation and procedures concerned with the operation of the software.	Usability
TRAINING	Those attributes of the software that provide transition from current operation or initial familiarization.	Usability
COMMUNICATIVENESS	Those attributes of the software that provide useful inputs and outputs which can be assimilated.	Usability
SOFTWARE SYSTEM INDEPENDENCE	Those attributes of the software that determine its dependency on the software environment (operating systems, utilities, input/output routines, etc.)	Portability Reusability
MACHINE INDEPENDENCE	Those attributes of the software that determine its dependency on the hardware system.	Portability Reusability
COMMUNICATIONS COMMONALITY	Those attributes of the software that provide the use of standard protocols and interface routines.	Interoperability
DATA COMMONALITY	Those attributes of the software that provide the use of standard data representations.	Interoperability
CONCISENESS	Those attributes of the software that provide for implementation of a function with a minimum amount of code.	Maintainability

FIGURE 2.6 Criteria Definitions for Software Quality Factors (Contd.)

SOURCE: Macabe's Book on Software Quality Assurance - A Survey

is responsive to the quality requirements of the customer and to the particular system application. The general responsibilities of such an activity include:

1. Planning
 - a) Preparation of the quality assurance plan stating duties, responsibilities, and schedule.
 - b) Project and customer interfaces.
 - c) Resource management.
 - d) Subcontractor/supplier management.
2. Policy, Practice and Procedure Development
 - a) Preparation of standards manuals for all phases of the software production, including requirements design, coding, and test tailored to specific project requirements.
 - b) Problem reporting and analyses.
3. Software Quality Assurance Aids Development
 - a) Adaptation of existing tools or methods.
 - b) Development of manual and automated procedures.
 - c) Keeping abreast of new and "state of the art" aids.
4. Audits
 - a) Review of project procedures and documentation for compliance to standards.
 - b) Participation in interim reviews.
 - c) Participation in customer audits of the project.
 - d) Quality assurance inspections.
5. Test Surveillance
 - a) Participation in the testing phase.
 - b) Reporting of software problems.
 - c) Analysis of error causes and assurance of corrective action.
6. Records Retention
 - a) Quality assurance records management.

b) Retention of problem reports, test cases, test data, logs of quality assurance reviews.

c) Insure proper documentation.

7. Physical Media Control

a) Inspection of disk, tapes, cards, and other program-retaining media - verification at all times of physical transmittal or retention.

b) Protection from mishandling or altering by environment. [Ref. 15]

The classical quality assurance group role or interface with the development cycle usually comes at the end of the development cycle when testing starts. Their job is to dissect the problem, find errors, test for the environment in which the software product is to be used in and notify the developers of faults. This sometimes produces an adversary relationship between the groups, destroying any cooperation or aid one might give the other. The autonomy of the quality assurance group is also imperative for achieving any type of success. [Ref. 15]

In software production environments today, the quality assurance group's intention is to work with the development side of the house throughout the development cycle. They view themselves as a tool or aid to the management of the development process, informing the manager of problems they see as a hinderance to the schedule or quality of the product under development. The autonomy of this group is still important. [Ref. 17]

D. SUMMARY

This chapter has listed the questions which must be answered about the software product before the duties of the quality assurance group can be delimited. Along with these questions, the exact role of the quality assurance group and

its interfaces with the development group may be viewed differently, depending on the character of the organization itself. The following chapters define the purpose and environments of the quality assurance organizations under consideration and explain their process of quality assurance.

III. FLEET MATERIAL SUPPORT OFFICE

The purpose of this chapter is to describe the FMSO environment and the process of software quality assurance in this organization. The following references were used:

1. Fleet Material Support Office Organizational Manual
2. Fleet Material Support Office Central Design Agency (CDA) Management Handbook, 1 January 1981,
3. Fleet Material Support Office Internal Instruction 5230.20A CDA Development Handbook, 1 December 1979,
4. Fleet Material Support Office Internal Instruction 5230.12 Quality Assurance Program, 17 May 1978,
5. Fleet Material Support Office Quality Program Task Group Report, 31 January 1982,
6. The Navy Supply Corps Newsletter, January 1982, Special Issue "Celebrating FMSO's 20th Anniversary."

A. HISTORY

Established in 1962, FMSO was originally chartered to provide central management for the retail portion of the Navy Stock Fund (NSF). It was also used to obtain and stock supplies from other services. It also catalogued data for supply system performance analysis and evaluation. Originally this organization consisted of five officers and 56 civilian employees, but today it has grown to more than 33 military personnel and over 1,300 civilians.

The main reason for the organization's growth has been its increase in responsibilities. The first addition occurred in 1965 when the Central Design Agency function was incorporated into its mission areas. This function involves the design, development and life cycle maintenance of

programs used in computer systems. This initial designation was limited to computer systems used in supply and financial operations at various field activities.

In 1973, FMSO's direct relationship with the fleet was increased with the assignment of the 3M program. This function was reassigned to the Navy maintenance support office in 1978. In 1977, two additional increases in FMSO's mission area occurred. The financial systems role was significantly expanded with the assignment of CDA responsibility for financial systems utilized by headquarters activities in Washington, D.C., such as the Naval Material Command and various systems commands. The other expansion was the result of FMSO's designation as the CDA for the Trident Logistics Data System, which added submarine intermediate level maintenance to FMSO's CDA mission. The most recent addition to their mission area occurred in 1978 with the responsibility assignment of the prototype development for the Naval Aviation Logistics Command Management Information System (NALCOMIS).

Approximately 80% of FMSO's work force is engaged in CDA activities. A significant portion of that effort is expended in four Uniform Automatic Data Processing Systems (UADPS): the Uniform ADP System for Inventory Control Points (UICP), the Uniform ADP System for Stock Points (UADPS-SP), the Level II/III system, and the Disk Oriented Supply System (DOSS). A list of the user sites for each system appears in Figure 3.1.

B. ORGANIZATIONAL STRUCTURE

Figure 3.2 shows the organizational structure of FMSO. Two departments carry out all of the staff functions such as resource management, operation and maintenance Navy budgeting, planning/administration, production support, project

UADPS-SP	UADPS-ICP	TOPS:	LEVEL II
ASO	ASO	ASO	NS ADAK
CHAVRES	CMAL	CBC POINT BEUCHE	NS KEFLAVIC
MCAS CHERRY POINT	CNET	FC CLEVELAND	NS KEY WEST
MCAS EL TORO	FHSO	NAD CRANE	NS ROTA
MCAS YUMA	NAVAIR	NAS BRUNSWICK	NAF NAPLES
NAS ALAMEDA	NAVELEX	NAS CORPUS CHRISTI	NAF SIGONELLA
NAS HARBERS POINT	NAVSEA	NAS DALLAS	NAS BERHUDA
NAS CECEL FIELD	NAVSEC	NAS JACKSONVILLE	NAS BRUNSWICK
NAS CORPUS CHRISTI	SPCC	NAS KEY WEST	NAS KADENA
NAS GLENVIEW		NAS LEHORE	NAS OCEANA
NAS JACKSONVILLE	MISII	NAS MEMPHIS	NAS ROOSEVELT ROADS
NAS LEHORE		NAS MERIDIAN	
NAS MIRAMAR	NAVTELCO	NAS NEW ORLEANS	HQ FINANCIAL
NAS MOFFETT FIELD	NALCOMIS:	NAS PATUXENT RIVER	ROPEERS
NAS NEW ORLEANS	ACTIVITY TYPES:	NAS PENSACOLA	CNM
NAS NORFOLK	CV	NAS SCOTIA	CNO
NAS NORTH ISLAND	LPH/LHA	NAS SO WEYMOUTH	FAADCLANT/PAC
NAS PENSACOLA	NAG	NAS WASHINGTON	JCMFO
NAS WHIDREY ISLAND	MCAS	NAS WASHINGTON	NAVAIR
NATC PATUXENT RIVER	NOTE:	NSC CHARLESTON	NAVCOMPT
NAVBASE BANGOR	TOTAL OF 94	NSC NORFOLK	NAVELEX
NAVBASE PEARL HARBOR	ACTIVITIES TO	NSC OAKLAND	NAVEAC
MPFC PHILADELPHIA	BE IMPLEMENTED	NSC SAN DIEGO	NAVINTCOM
NSC CHARLESTON		PGS HONOLULU	NAVPRO (7)
NSC NORFOLK	NATDS	SPCC	NAVSEA
NSC OAKLAND	NAVMO	SUBASE NEW LONDON	NAVSUP
NSC PEARL HARBOR		TC GREAT LAKES	NAVTELCOM
NSC PUGET SOUND	NAVAIDS:	TC ORLANDO	NREC (2)
NSC SAN DIEGO		USNA ANNAPOLIS	SSPO
NSD GUAM	NSC CHARLESTON		SUPSHIP (9)
NSD SUBIC BAY	NSC NORFOLK	RH6E	
NSY PHILADELPHIA	NSC OAKLAND	ASO	FADPS
PHOLANT	NSC PEARL HARBOR	NSC NORFOLK	NAS JACKSONVILLE
PROFAC	NSC PUGET SOUND	NSC OAKLAND	NAS OCEANA
PTTC POINT HURD	NSC SAN DIEGO	PHOLANT	NAS NORTH ISLAND
SPCC	TRIDENT LBS		COMNAVAIRANT
SUBASE NEW LONDON			COMNAVAIRANT
SWPAC SILVERDALE	SPCC		
	TRF		

FIGURE 3.1 Index of FMSO CDA Systems Users

SOURCE: CDA Management Handbook

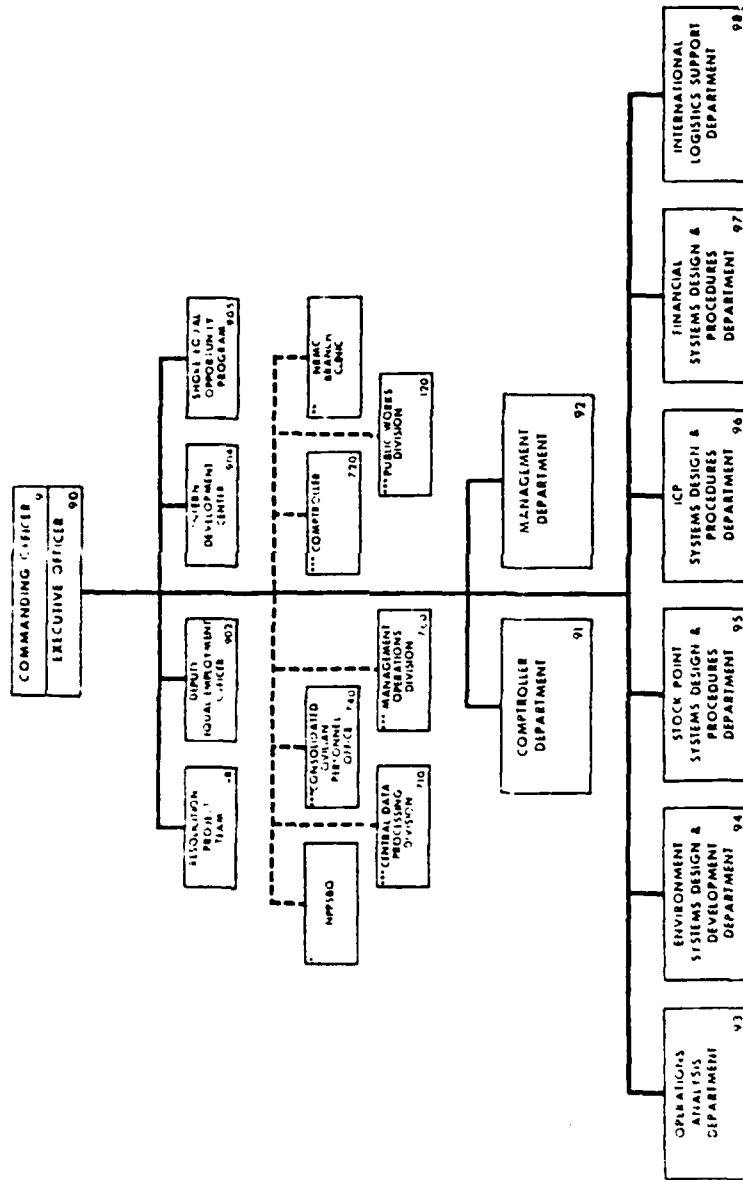


FIGURE 3.2 FMSO Organization
SOURCE: FMSO Organization Manual

control, standards development, data base administration, training and ADP operations support. These are the Comptroller Department (Code 91) and the Management Department (Code 92). The software quality control branch is in the planning division of Code 92. (Figure 3.3) The Comptroller Department also performs external missions including stock fund budgeting and direct fleet support functions.

The Operations Analysis Department (Code 93) is the Naval Supply Systems Command's (NAVSJP) principal agent for conducting analysis in logistics management. This department is made up of operations research analysts and mathematicians who use various mathematical, statistical and economic analysis techniques to study and improve the procurement, financial and inventory management functions throughout the United States Navy. These services are also provided for all NAVSUP activities, the fleet, Chief of Naval Operations and Chief of Naval Material offices, other systems commands and various project managers.

C. THE CDA

"A central design agency is defined as a single organization which designs, develops, implements and maintains automated data processing systems in support of multiple operating sites." [Ref. 19] The five FMSO CDA production departments (Code 94 through 98) are the line organizations which are directly responsible for the development and maintenance of standard ADP systems. The personnel in these departments are functional systems designers, computer systems analysts, computer specialists, and computer programmers. Their work, development and documentation of these programs, is the major product of the CDA.

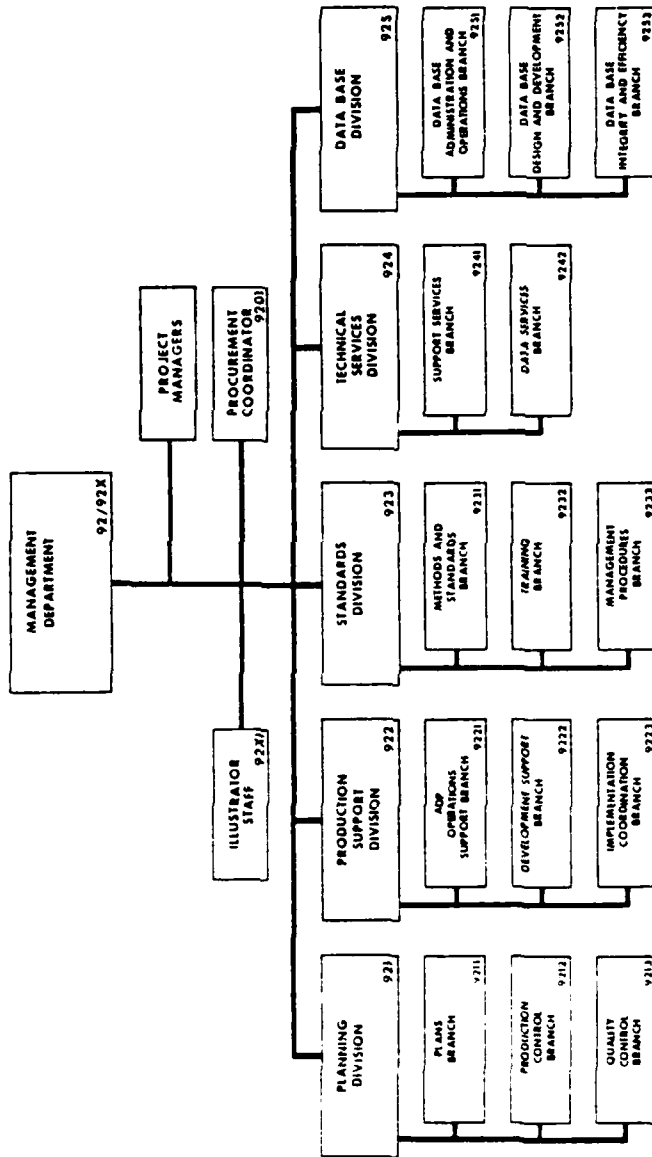


FIGURE 3.4 Code 92 Organization
SOURCE: FMSO Organization Manual

Three basic principles necessitate the existence of this type of production organization and directly impinge on its effectiveness.

1. There must be a potential group of customer sites which perform a mission of functional similarity and operate with business volume of a magnitude sufficient to justify acquisition and operation of automated systems.

2. The functional similarity of the individual sites within the customer group must be complete enough to permit a degree of system standardization by which the single product of the design agency can adequately support the needs of multiple users, thus the cost of system development and maintenance can be defrayed by the benefits obtained by the many users. At the same time, a marked degree of standardization and improved management is obtained.

3. The concentration of system design and development talent in a CDA affords opportunities for single operating sites to obtain development of systems that they could not afford to develop themselves.

The objectives of a CDA is as follows:

- To initiate ADP developmental action on projects which have undergone cost benefit analysis and were determined to have a high ratio of benefit to cost.
- To insure continued compatibility of all systems with approved military standardization programs and existing supply and financial management policy.
- To optimize responsiveness to logistic managers in the fleet and shore establishments in the development and maintenance of assigned systems. Optimum responsiveness is the timely production of accurate reports and analysis documents required to improve the effectiveness of supply, financial and maintenance functions.
- To emphasize user site resource savings in staffing, ADP hardware, plant equipment and inventory investments in the development and maintenance of assigned systems.
- To involve user sites in the identification of automation opportunities, definition of requirements and economics, prioritization of workload, and support of systems prototyping and implementation.

- To develop rigidly-uniform programs with design options, alternatives and modularity which facilitate subsequent policy/procedural changes and accommodate unique customer requirements with due consideration of efficiency/flexibility trade-offs.
- To design and develop ADP systems which will be compatible with the projected role of user sites during future years.
- To participate in the exchange of information with other DDD design agencies and to enhance systems effectiveness and personnel proficiency.
- To identify project resource requirements in the initial planning stage so that sufficient lead time is provided for timely acquisition and development.
- To prepare CDA budgets which reflect sound and integrated production plans; to allocate resources within the CDA in accordance with reconciled budget/production plans.
- To optimize CDA organizational structure, staffing levels, and allocation of personnel resources in order to insure maximum productivity on high priority projects.
- To pursue personnel recruitment and training programs which insure availability of advanced knowledge and skills in logistics, data processing, financial management and related disciplines.
- To enhance CDA productive capability through the use of special tools, including interactive programming, data base management, pre-compilers, and other available techniques.
- To employ the most effective training techniques available in order to implement systems at new user sites and install new applications at existing user sites; to conduct a program of field assistance which assures continued proficiency of user sites in operations supported by CDA systems.
- To utilize standard high-level programming languages to the maximum extent feasible and to use assembly languages only where technical requirements unequivocally dictate." [Ref. 19]

While all of the CDAs are involved in basically the same operation, they are separated into logical functional areas of support. Because of this separation, the CDAs do not all serve the same customers. FMSO as a CDA is divided into the following areas:

Environmental Systems Design and Development Department
(Code 94) (Figure 3.4)

This department is responsible for the design, development, implementation and maintenance of environmental systems software in support of NAVSUP-sponsored ADP systems, including UADPS of stock points, UICP, the trident program, the international logistics program, and programs that are assigned. This department also performs these functions for the systems maintained by the other CDA's. Telecommunications networks sponsored by the Naval Supply Systems Command are another area in which code 94 is responsible for the environmental systems software. This department is made up of 109 computer specialists and 27 people who handle all managerial and clerical activities. Other major projects either designed or supported are:

1. SPLICE - stock point logistics integrated communications environment
2. LDC - logistics data communications
3. OLA - on-line autodin
4. AUTODIN II - automatic digital network

Stock Point Systems Design and Procedures Department (Code 95) (Figure 3.5)

This department's purpose is to develop and maintain the automated systems for Navy stock point support including trident Logistic Data System (LDS), NALCOMIS, Automated Ready Supply Stores System (ARSSS), Tape Oriented Supply System (TOSS), Disk Oriented Supply System (DOSS), Electric Point of Sale, level II, Navy Automated Transportation Documentation System (NAVADS), Navy Automated Transportation Data System (NATDS), Transportation Operational Personal Property Standard System (TOPS), Navy Integrated Storage-Tracking and Retrieval System (NISTARS), Requisition Material Monitoring and Expediting (RMM&E), Closed Loop

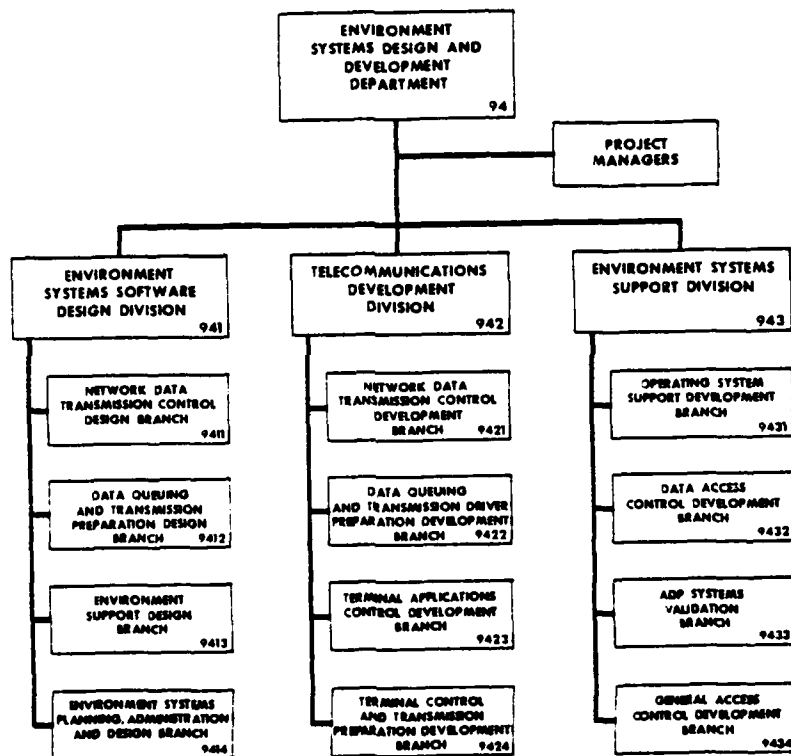


FIGURE 3.4 Code 94 Organization
SOURCE: FMSO Organization Manual

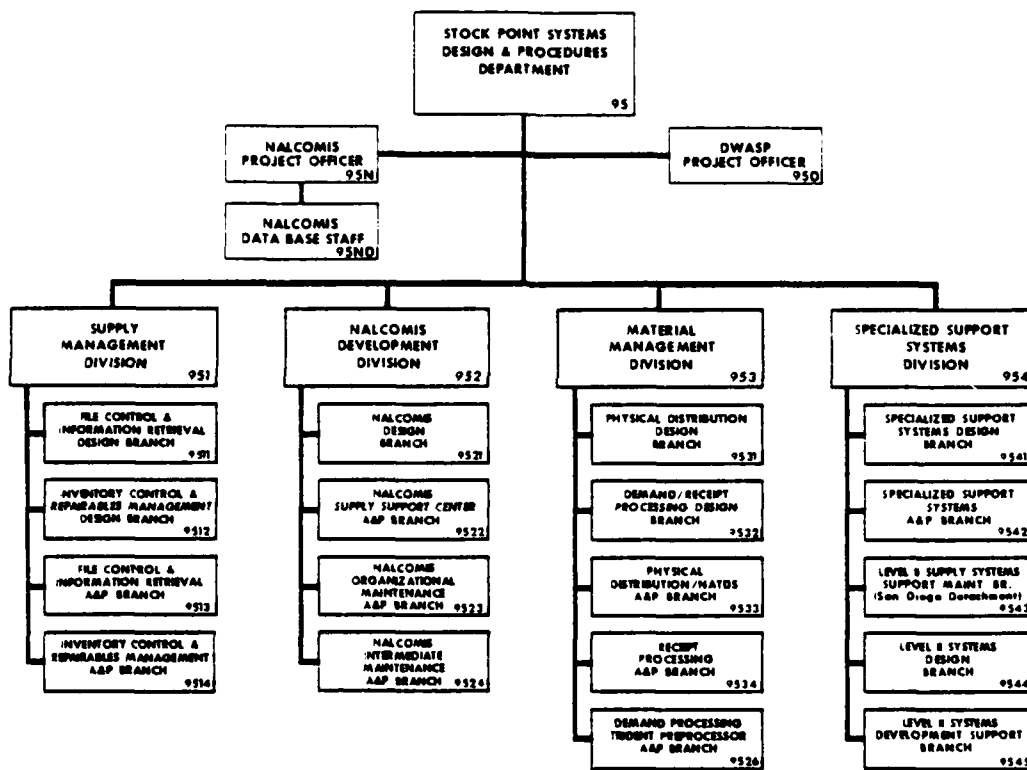


FIGURE 3.5 Code 95 Organization
SOURCE: FMSO Organization Manual

Aeronautical Management Program (CLAMP), and Defense Warehousing and Shipment Process (DWASP). This department also assists customers with the implementation of these ADP systems through development of training documentation, initial training and installation assistance, monitoring of performance under operational conditions and follow-on field assistance. The department is involved with approximately 40 Navy stock points located around the world.

Inventory Control Points Design and Procedures Department (Code 96) (Figure 3.6)

The purpose of this department is to develop and maintain the ICP's UADPS design and work on refinements to these programs to carry out NAVSJP and hardware SYSCOMS inventory control functions. Their principal customers are the two major Navy ICPS: the Ships Parts Control Center (SPCC) and the Aviation Supply Office (ASO). This department also develops and maintains detailed systems design for trident and ship-support functions. It is comprised of approximately 250 people and is a functionally oriented department. The Financial Systems Design and Procedures Department (Code 97) (Figure 3-7)

This organization is responsible for systems design, development implementation and maintenance services for headquarters, Naval material command; Chief of Naval Material designated project management offices; and other participating headquarters commands and offices. It provides service to both of the major customer groups; inventory control points and stock points and other activities under the UICP and UADPS programs in the areas of financial inventory control, stores accounting, disbursing, plant property, payroll and personnel accounting.

The systems designed by this organization supports 91% of the Navy's financial inventory report requirements, 75%

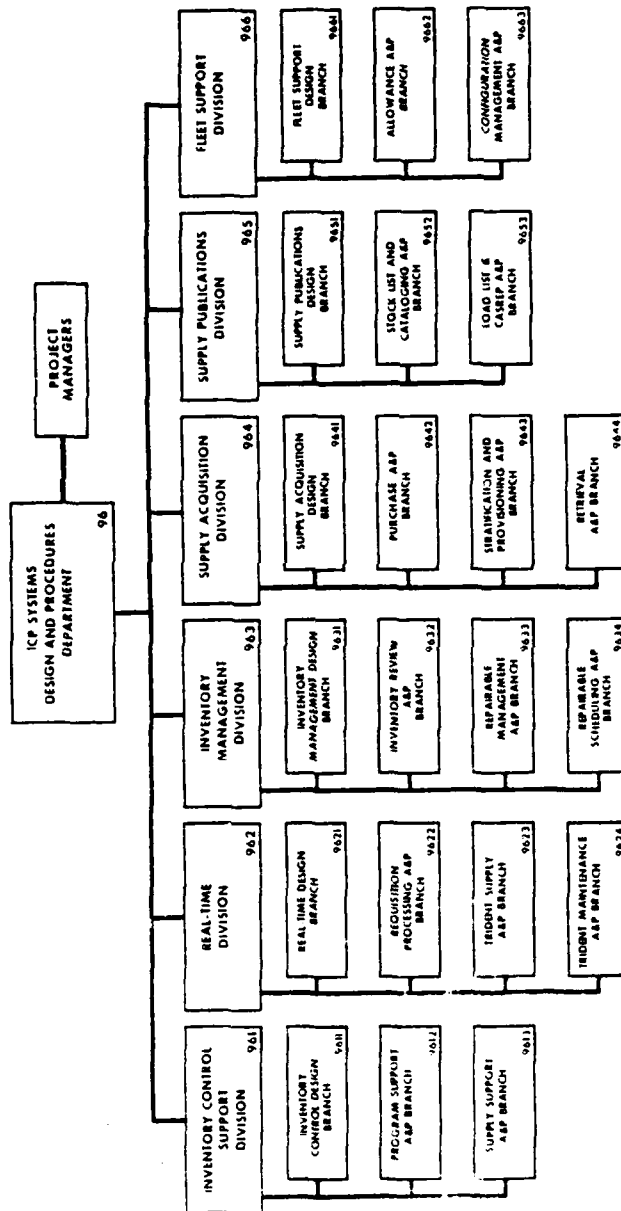


FIGURE 3.6 Code 96 Organization
SOURCE: FMSO Organization Manual

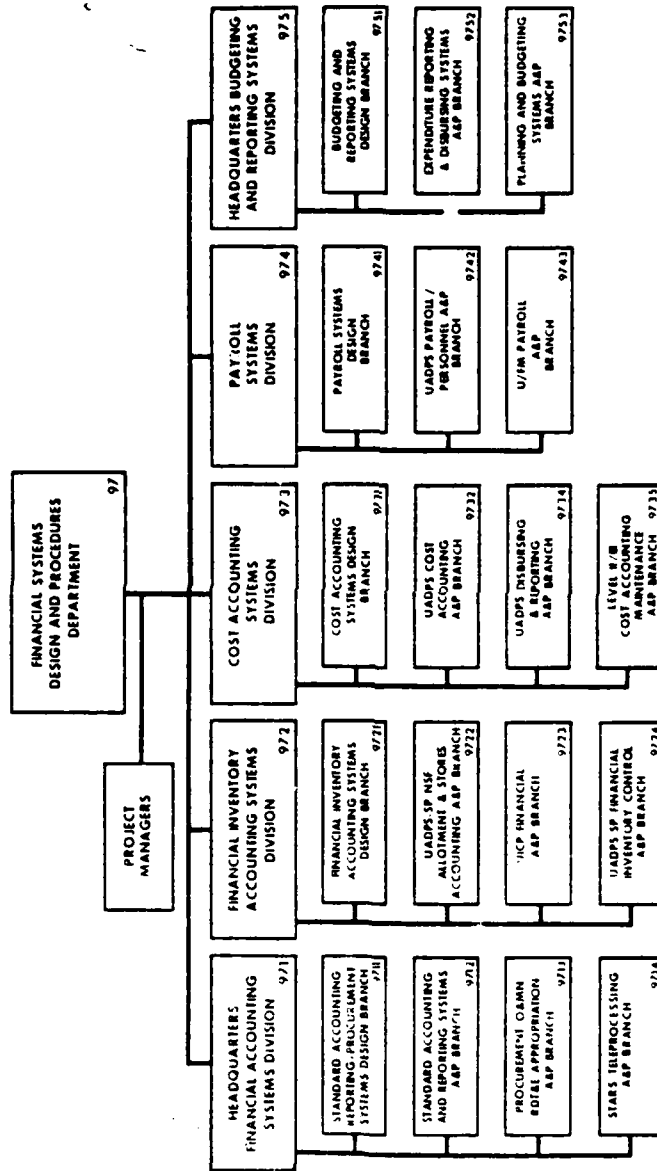


FIGURE 3.7 Code 97 Organization

SOURCE: FMSO Organization

of the current Navy dollar resources under its resource management system, and 63% of 189,000 civilian employee salaries.

Code 97 provides similar services to the Navy regional finance centers and evaluates the performance and develop such projects as the Integrated Disbursing and Accounting (IDA) System, Standard Accounting and Reporting System (STARS) and the Automated Procurement and Accounting Data Entry (APADE) System.

This department consists of three military officers and a civilian complement of 244, covering the full range of financial systems and data processing expertise.

International Logistics Support Department (Code 98) (Figure 3.8)

This department is responsible for the maintenance and continual enhancement of the Management Information System for International Logistics (MISIL). Its principal customer is the Naval International Control Officer (NAVILCO) which utilizes its systems to provide services to numerous allied navies and governments. The department handles complete automation for the Saudi Arabian's Navy supply system and automation of support systems (supply, environmental, personnel and financial) for Kuwait's Navy. It establishes training programs for United States Navy Supply Corps personnel going to Military Assistance Advisory Group (MAAG) duty and develops an advance base supply system for overseas supply depots.

D. SYSTEM DESIGN AND QUALITY ASSURANCE PROCESS

The top down design method is used as the standard approach for new system/program development in the FMSO environment. This approach is also known as stepwise refinement, hierarchical design, levels of abstraction and

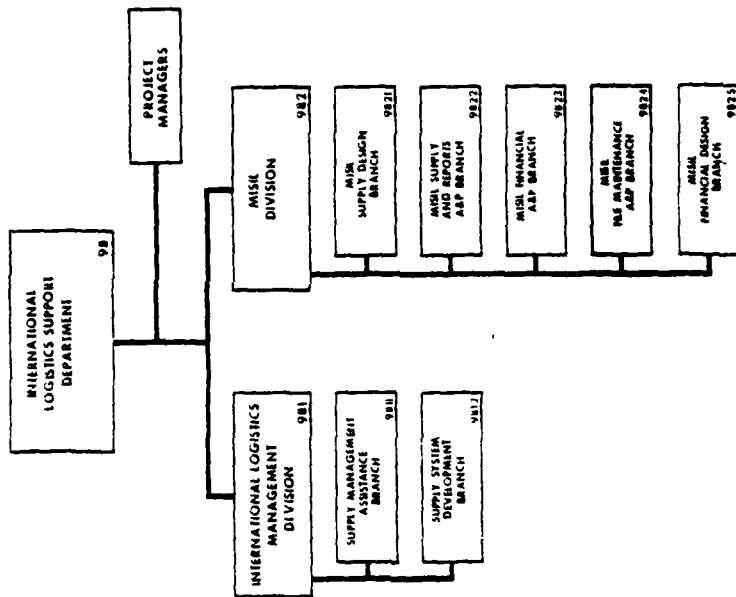


FIGURE 3.8 Code 98 Organization
 SOURCE: FMSO Organization Manual

design by explosion. The method uses a breakdown technique, dividing the main function into smaller subfunctions. The primary function, thought of as the central or driving function, is designed first; then stepwise, this process is continued until the smallest functional unit of the system is specified.

Because of this breakdown, the system can be viewed as modules. Every stage of the system and program yields a visible output. Each subsequent subfunction which is defined becomes a module of code which, when tested, serves to retest and more thoroughly test all higher level modules.

The use of hierarchical charts forces the design of new system/programs in the top down method. This use of visual diagrams shows the major functions and their subfunctions with the emphasis on their subordination and not their logical flow.

FMSO personnel state that the system designers focus on what is required and the systems analysis workers focus on how to achieve it. The system designer, working very closely with the system user, defines what information is required, how it is required, when it is required, and for whom it is required. This helps tremendously in keeping this process of development at minimum cost.

The system development process is delineated in FMSO's CDA Management Handbook. Appendix A, taken from the handbook, shows the process.

During the development process a quality assurance checklist is required. Figure 3.3 is an example of the checklist.

On 31 January 1982, a quality program task group report was published. In this report were the results received from the following: an internal survey taken from the CDAs; an examination of the ADP development model and the CDA

QUALITY ASSURANCE CHECKLIST

Program/version _____	Date _____	
<u>ELEMENT</u>	<u>SIGNATURE</u>	<u>DATE</u>
1. <u>Scope of Release:</u>		
a. New program/complete rewrite	<input type="checkbox"/>	
b. Major modification	<input type="checkbox"/>	
c. Moderate revision	<input type="checkbox"/>	
d. Minor adjustment	<input type="checkbox"/>	
	_____	_____
2. <u>Criticality of Release:</u>		
a. Mandatory (HQ. directed)	<input type="checkbox"/>	
b. PTR response	<input type="checkbox"/>	
c. Solves serious program deficiencies	<input type="checkbox"/>	
d. Highly desirable enhancement	<input type="checkbox"/>	
e. Routine release	<input type="checkbox"/>	
	_____	_____
3. <u>Urgency of Implementation:</u>		
a. Immediate	<input type="checkbox"/>	
b. No later than _____	<input type="checkbox"/>	
c. Optional	<input type="checkbox"/>	
	_____	_____
4. <u>Level of Testing:</u>		
a. Local FMSO testing with simulated test data	<input type="checkbox"/>	
b. Service tested at _____	<input type="checkbox"/>	
c. Prototyped/Op Reviewed at _____	<input type="checkbox"/>	
d. Tested by FMSO with live data files/transactions from _____	<input type="checkbox"/>	
	_____	_____

FIGURE 3.9 Quality Assurance Checklist
SOURCE: FMSO Quality Assurance Program

<u>ELEMENT</u>	<u>SIGNATURE</u>	<u>DATE</u>
5. Meets standards of hardware utilization	_____	_____
6. Availability of proper hardware verified at user sites	_____	_____
7. Impact on hardware capacity assessed and verified as available at user sites	_____	_____
8. Release will lengthen real time responses by _____	_____	_____
9. <u>Documentation meets standards of NAVSUP Pub. 506 (Rev. April 1976)</u>		
a. Functional Description	_____	_____
b. System/Subsystem Specification	_____	_____
c. Program Specification	_____	_____
d. Computer Operation Manual	_____	_____
e. Program Maintenance Manual	_____	_____
f. Test and Implementation Plan	_____	_____
g. User's Manual	_____	_____
h. Data Requirements Document	_____	_____
i. Data Base Specification	_____	_____
j. Change Transmittal Notice	_____	_____
k. Test Analysis Report	_____	_____
l. Project Manual	_____	_____
m. Technical Report	_____	_____
n. Technical Note	_____	_____
10. System/Subsystem Specification was approved by NAVSUP	_____	_____

2

FIGURE 3 9 Quality Assurance Checklist (Contd.)
SOURCE: FMSO Quality Assurance Program

<u>ELEMENT</u>	<u>SIGNATURE</u>	<u>DATE</u>
11. Satisfies System/Subsystem Specification as Approved	_____	_____
12. Satisfies Program Specification	_____	_____
13. File Integration/Integrity Verified	_____	_____
14. System Integration/Integrity Verified	_____	_____
15. Tested in (Simulated/ Production) Environment	_____	_____
16. Test Data Base Updated To Ensure Adequate "Real World" Cases	_____	_____
17. Program Restart Capability Verified	_____	_____
18. Program Interfaces with Software:		
a. Currently Implemented	_____	_____
b. New Software Package Required	_____	_____
c. Scheduled for Release	_____	_____
19. (Software) Release is Upward Compatible with Prior Releases	_____	_____
20. Programs have been developed, analyzed, coded and reviewed at critical steps utilizing the FMSO standard Improved Programming Techniques, as described in FMSOINTINST _____.	_____	_____
21. User Training Has Been Provided/ Is Not Required	_____	_____
a. Type Training Provided	_____	_____
b. Date Training Completed	_____	_____

3

FIGURE 3.9 Quality Assurance Checklist (Contd.)
SOURCE: FMSO Quality Assurance Program

22. Standard data element names have been used throughout the program coding.

23. Remarks/qualification/explanation: _____

24. Element certification responsibilities: see item 24, enclosure (4) for individual element certification responsibility.

25. QUALITY ASSURANCE CHECKLIST CERTIFICATION.

Each of the above quality assurance checkpoints has been verified/ validated by myself or by persons under my supervision. The responses given are true and correct to the best of my professional knowledge. I understand that individual quality assurance level is a significant factor in each annual performance rating. I certify that this program release has met all FMSO quality assurance tests and standards and is ready for release to customer activities.

_____ Branch Head	_____ Date
_____ Division Head	_____ Date
_____ Department Head	_____ Date

4

FIGURE 3.9 Quality Assurance Checklist (Contd.)
SOURCE: FMSO Quality Assurance Program

handbooks; a review of the functional operations of the quality control organization, research and review of the technical libraries and publications dealing with software quality assurance programs, and an external survey questionnaire directed to the FMSO-systems customer community.

The report stated that the following factors in the FMSO environment prejudice quality in varying degrees:

1. Mandated, multiple and dissimilar hardware configurations.
2. Unrealistic/inflexible/mandate project completion dates.
3. Ill defined or undocumented requirements.
4. Inadequate test facilities.
5. Funding (budget/travel) constraints.
6. Project prioritization process.
7. Diversity of customer activity in systems/processing requirements.
8. System changes/controls edicted from agency/system command echelons.
9. Federal procurement policies and regulations.

The task group's work experience, a review of industrial literature, and the internal survey revealed that the following specific conditions exist:

1. Poorly Defined Requirements/Specifications
 - a) FMSO design procedures/practices tend to be application-oriented and at the discretion of the developer.
 - b) System design and analysis knowledge is not being shared between or within the CDAs.
 - c) Formal review and walkthroughs are not being carried out properly during system development.
 - d) There is no visible interaction with customers.

- e) System analysts are not always required during unit testing.
 - f) With the exception of the program trouble report, there is no provision for soliciting or consolidating customer feedback information on a recurring basis.
 - g) ADP system developmental information and experience is not formally or consistently shared among developmental organizations.
 - h) A more business-like, comprehensive policy and procedures document is necessary for FMSO/customer relationships.
2. Unrealistic Schedules/Estimated Completion Dates
- a) Mandatory due dates cause abbreviation of quality events.
 - b) Completion date as set by the POA&M is usually "set in stone."
 - c) Project tracking/status reporting and resource accounting are not currently provided on an integrated basis for project management.
 - d) There is limited automated capability in the areas of documentation preparation, storage, assembly, packaging and distribution.
3. Insufficient Testing Time/Test Facilities
- a) Unreliability of hardware (FMSO), basically the test beds, precludes estimating realistic time frames and completion dates.
 - b) There is lack of uniformity in the assignment of specific responsibilities in program/system testing.
 - c) No uniform methods or procedure exist for establishing and maintaining FMSO's test files.

- d) An undisciplined approach to program testing among CDAs is used.
- e) Software engineering is not a distinct function.
- 4. Lack of "State-of-The-Art" Developmental Tools and Aids
- 5. Unnecessary Paperwork and Processes

E. CONCLUSION

As shown in the system development process, Appendix A, the quality assurance branch interfaces with development personnel in tracking of the functional description and system specifications and in checking the product before release for compliance with standards and quality assurance procedures (check list). All tests and project reviews are carried out by the development personnel with the use of the quality assurance check list. The actual duties of the quality assurance branch may be viewed as only administrative in nature. The next chapter shows how other quality assurance groups function in their organizations.

IV. INTERVIEWS

This chapter presents the author-conducted interviews with personnel of the quality assurance groups in the computer organizations addressed in Chapter I. The following questions were asked during the interview:

1. Where does the quality assurance group fit into the organization?
2. What type of authority/power does the quality assurance group have over the software product?
3. What qualifications do the people in the quality assurance group have?
4. How does the quality assurance group interface with the design/development group?
5. What tools, methodologies, or techniques does the quality assurance group use to do their job?
6. Are historical records of problems with software products kept after the products' release, and who in the company's organization keeps them?
7. Who handles problems with software after release, and how are such problems handled?
8. If a brand new product is designed, who in the company's organization trains the customer on this product?

The reader is enjoined to compare the interviews with the discussions in Chapters II and III.

A. HEWLETT PACKARD

The Hewlett Packard Company is a major designer and manufacturer of precision electronic equipment for measurement, analysis and computation. The company makes more than

4,000 products which are sold worldwide and have broad application in the fields of science, engineering, business, industry, medicine and education. Their four main product segments are:

1. Electronic Data Products -- computational products including personal computing devices, desk top computers for engineering and scientific applications, small business computers, and larger computer systems for both business and technical needs. They also offer a large selection of application software and have developed a wide selection of peripheral equipment for use with their computers, including computer terminals, disc memories, printers and plotters.

2. Electronic Test and Measurement Products -- range from general purpose instruments and systems for electronic test and measurement to specialized instrumentation for computed measurements to components and accessories such as microwave semiconductors, optoelectric displays, bar code readers, and fiber optic systems.

3. Medical Electronic Equipment -- family of more than 300 medical products which are used for diagnosing, monitoring, and treating patients, and for medical information management. This equipment ranges from portable electrocardiographs to powerful computer-aided patient monitoring and patient data management systems.

4. Analytical Instrumentation -- Product family includes gas and liquid chromatographs, mass spectrometers, automatic fluid samplers, analytical laboratory data acquisition systems, and spectrophotometers. This instrumentation is used for research, production, and environmental applications.

1. Organization

Figures 4.1 thru 4.3 show the organizational structure of the Hewlett Packard Company. In the computer area, there is the technical computers group, of which the Data Systems Division is a part. The products or quality assurance organization comes from this division. This organization is not only responsible for software quality assurance but also for hardware quality assurance, production support, product reliability, information systems, quality assurance, production regulation and safety, etc. The software quality assurance engineering group is made up of 14 people who have the education and experience to be program designers and programmers themselves, but their job is strictly quality assurance. Their main purpose is to work along with the product designers from the research and development group, assisting them in designing a quality product. This interface between designers and quality assurance people is not true for all areas of Hewlett Packard production, but the company is moving in that direction.

The quality assurance group does not have absolute authority over the product. Absolute power would mean that if they thought the product was not ready, it would not be released. They state that their real power lies in their reputation and their ability to persuade. If they predict a failure and it occurs, the group's credibility and reputation are enhanced, and the persuasion speaks for itself. The division general manager makes the final decision on whether a product is released, and it is the job of the quality assurance personnel, in competition with design people, to convince him/her that the product is not ready.

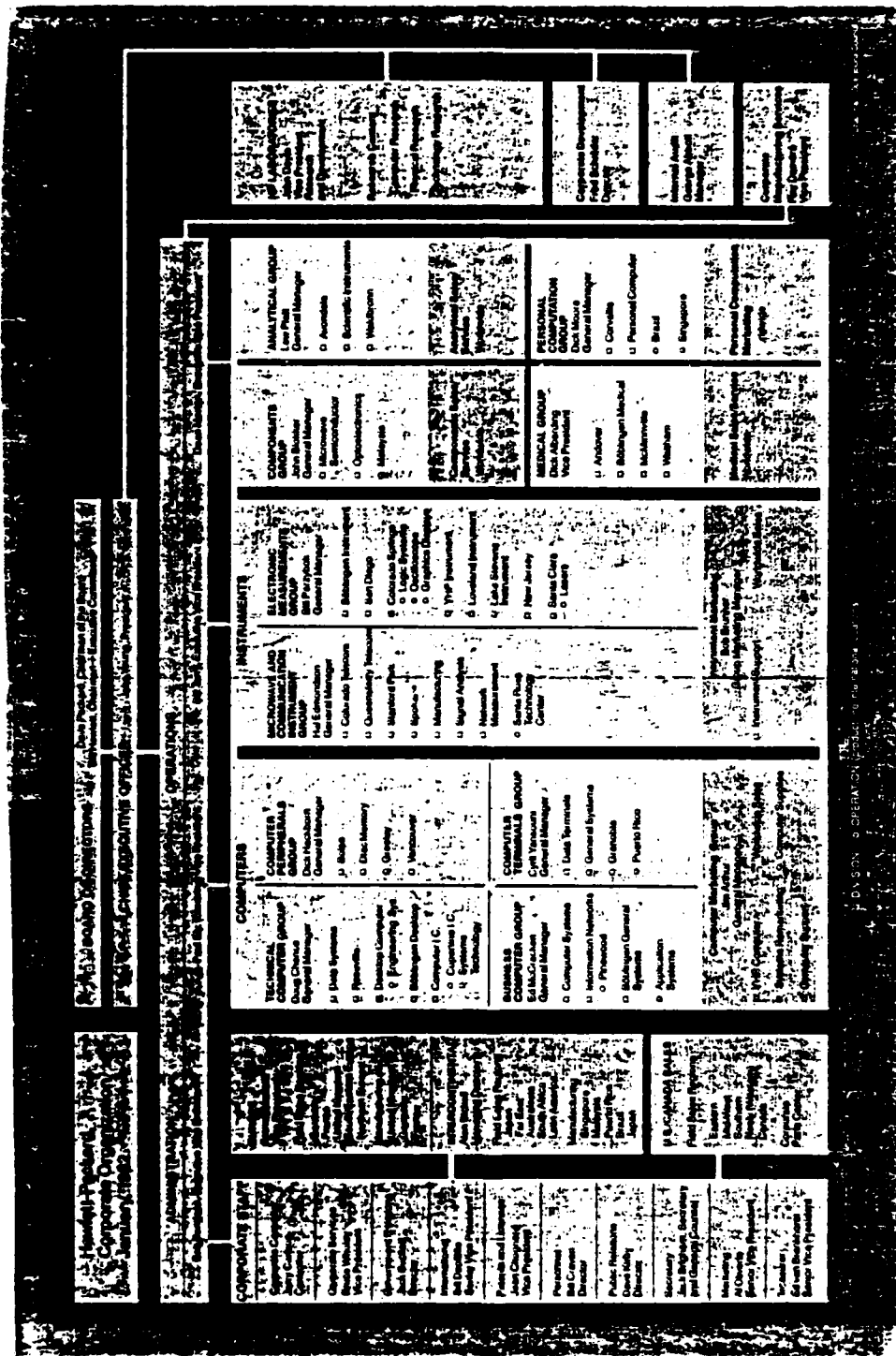


FIGURE 4.1 Hewlett Packard Organization
 SOURCE: Interview with Hewlett Packard Quality Assurance Manager

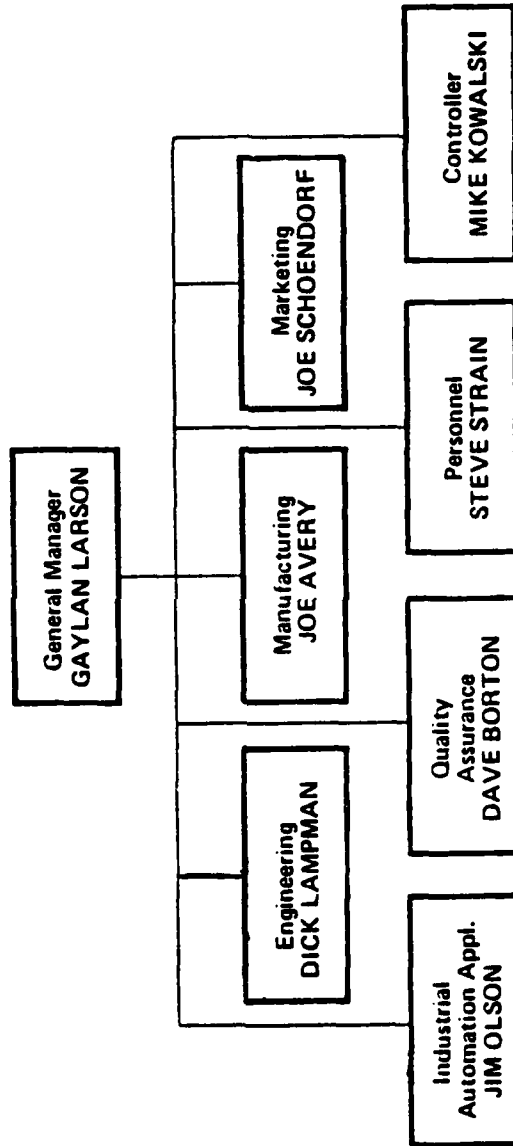


FIGURE 4.2 Hewlett Packard Data Systems Division
SOURCE: Interview with Hewlett Packard Quality Assurance Manager

PRODUCT ASSURANCE MANAGER

DAVE BORTON

MYRNA ALLEN, SECRETARY

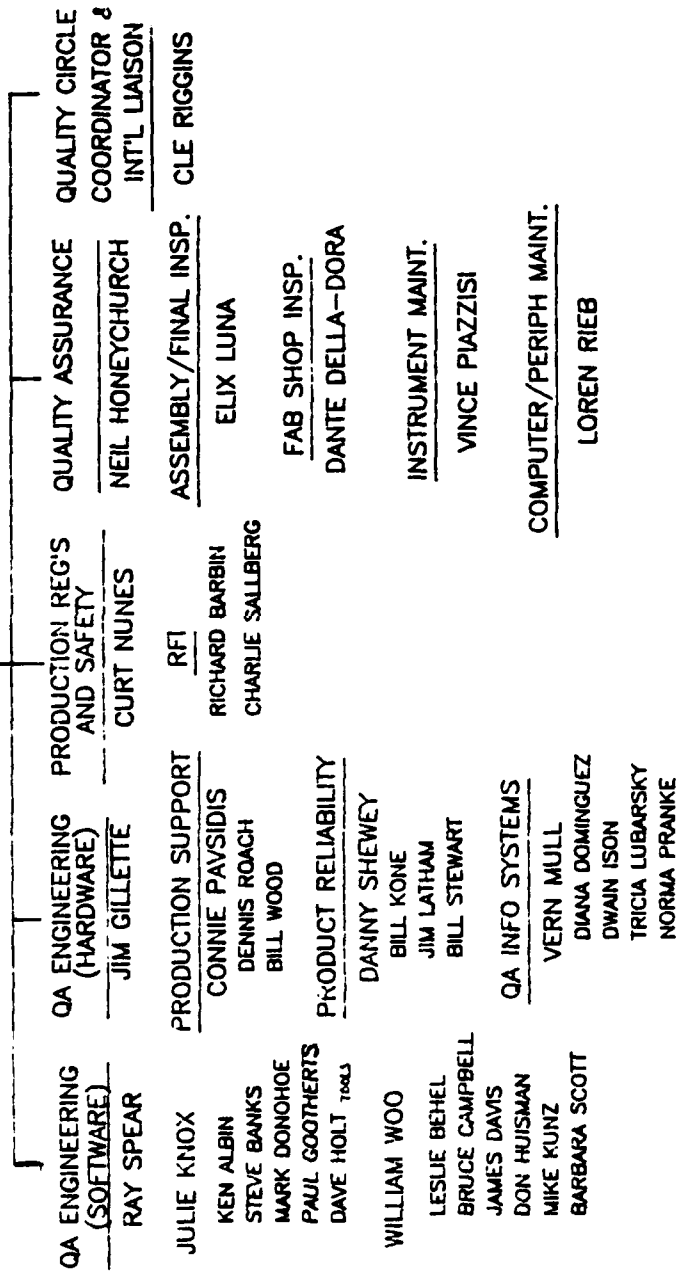


FIGURE 4.3 Hewlett Packard Products Assurance Group
SOURCE: Interview with Hewlett Packard Product Assurance Manager

2. Quality Assurance and Designer Interface

Figure 4.4 shows the development cycle as it is perceived by Hewlett Packard personnel. When the designers from research and development have an idea for a new product, a proposal is sent to management. If permission is given, a product design group is formed consisting of people from marketing, research and development, manufacturing, and quality assurance. When the design is laid out, the quality assurance people ask "What if" questions to ensure all aspects are considered. The company sets no particular specifications to which the designers must adhere, so they have the freedom to be creative. The main languages used by the designers are assembler, Pascal, and FORTRAN because their products tend to be more technical than commercial in nature. They also produce environmental and applications software. One person from quality assurance is assigned to each project.

During the requirement phase of the development process, an investigation has to be completed in order to produce a detailed specification plan and a user interface specification plan. In the external design segment the quality assurance people must produce a quality plan delineating the quality goals or objects of the project and how they are to be measured. This is a problem area for the quality assurance people because if the product is generated at a customer's request, the request is usually not specific or incomplete. It is important that formalized communications be established to eliminate this problem.

In the internal design phase of the development cycle, the internal specifications, top down design, and submodule design take place. The quality assurance personnel set up, monitor, and participate in design reviews and code reviews held during this period. They also produce the functional test plan.

PROPOSAL OR CONCEPT

REQUIREMENTS

1. Investigation
2. External Design

DEVELOPMENT

1. Internal Design
2. Implementation
3. Integration & Test
 - A. Functional
 - B. Systems
4. Quality Certification (Customer Acceptance)
5. Production Certification
6. Manufacturing

OPERATION

Figure 4.4 Hewlett Packard Software Development Cycle
Source: Interview of Hewlett Packard Software Quality Assurance Personnel

During the implementation statement, the quality assurance people set up the systems test plan. Actual testing is not accomplished until the integration and test segments, and it is done on the function and systems levels. Although the functional test plan is produced by the quality assurance people, the actual testing is done by the designers themselves. This level of testing is viewed mainly as a debugging exercise and would be a waste of the quality assurance organization's time and resources if done by them. At the systems level of testing, the plan and tests are done by the quality assurance people. These tests are viewed as a third party auditing inspection of the product. This third party testing is done because Hewlett Packard does not believe that the program designers and analysts can be completely objective about their product. The quality assurance group is also responsible for the packaging of all test plans for reusability. There are no percentages of correctness sought during these testing levels. When this segment is complete, the product is considered 100% correct..

According to the quality assurance people, another problem area is the schedule planning. The designers do not think that problems will occur during this testing phase, so they have to be careful to plan for extra time if problems occur.

After the quality certification segment, which is basically a customer acceptance inspection, and the production certification segment, comes the manufacturing segment. During this segment a pilot run is made on the product to ensure that, if a customer requested the product, all the materials -- the product itself, user manuals and any other items -- are shipped.

3. Operations

Hewlett Packard believes in "cradle to grave" involvement with its software products, which means they do not abandon their customers after sale. All Hewlett Packard software is copyrighted so if there are any problems after it is in operation, the cost to the customer is \$100 per hour for repairs unless the customer has a subscription service. Subscription service entitles the customer to have software repaired, updated, or replaced at a lower fee. This service includes a plan by which, if a program is updated or fixed for any customer, the updated version is sent out to all other customers who have the same program. The decision to use it within the customer's system is left to the customer.

If there is a problem, the customer first notifies the field activity which, if necessary, creates a "work around" program to keep the customer's system operational. From the field activity, the problem is referred to the manufacturer, via support, and eventually to the people in research and development who design the program. They prioritize the problem and place it in their schedule, and it is eventually fixed. No historical records of problems or changes to programs are maintained.

The quality assurance organization keeps abreast of the latest ideas and changes in this field and is constantly striving to improve its program.

Reference

Personal interview with Mr. Raymond L. Spear, software produces assurance manager, at the Hewlett Packard plant, Data Systems Division, Cupertino, California, on 14 April 1982.

B. TRW

TRW is a diversified multinational company which manufactures a wide range of products from components for cars and trucks to defense electronics and space systems. TRW produces transistors, resistors, capacitors, diodes, potentiometers, trimmers, tuning devices, TV convergence yokes, connectors, transformers, printed circuit boards, electric motors, electric data processing terminals, and jet engine parts. Other products include pumps, fluid handling equipment, nuclear reactor components, fasteners, bearings, cutting tools, and hand tools.

This company handles defense systems contracts which include the development of software and the construction of the entire system.

1. Organization

TRW is divided into many groups because of its diversification. One of these groups, the defense systems group, contains the engineering division of which the products assurance organization is a part. (Figure 4.5) This level is made up of managers who are assigned to the different projects in assistant project manager capacity. This department is not just concerned with software product assurance, but also with hardware and system engineering and design (SEAD) product assurance. (Figure 4.6)

Figure 4.7 shows the standard work breakdown structure for any product in the defense systems group as it is concerned with product assurance. The assistant project manager heads up a staff of personnel who work in the areas of quality assurance, configuration management, reliability, and safety.

Figure 4.8 shows the standard work breakdown structure for the quality assurance area of the project which is

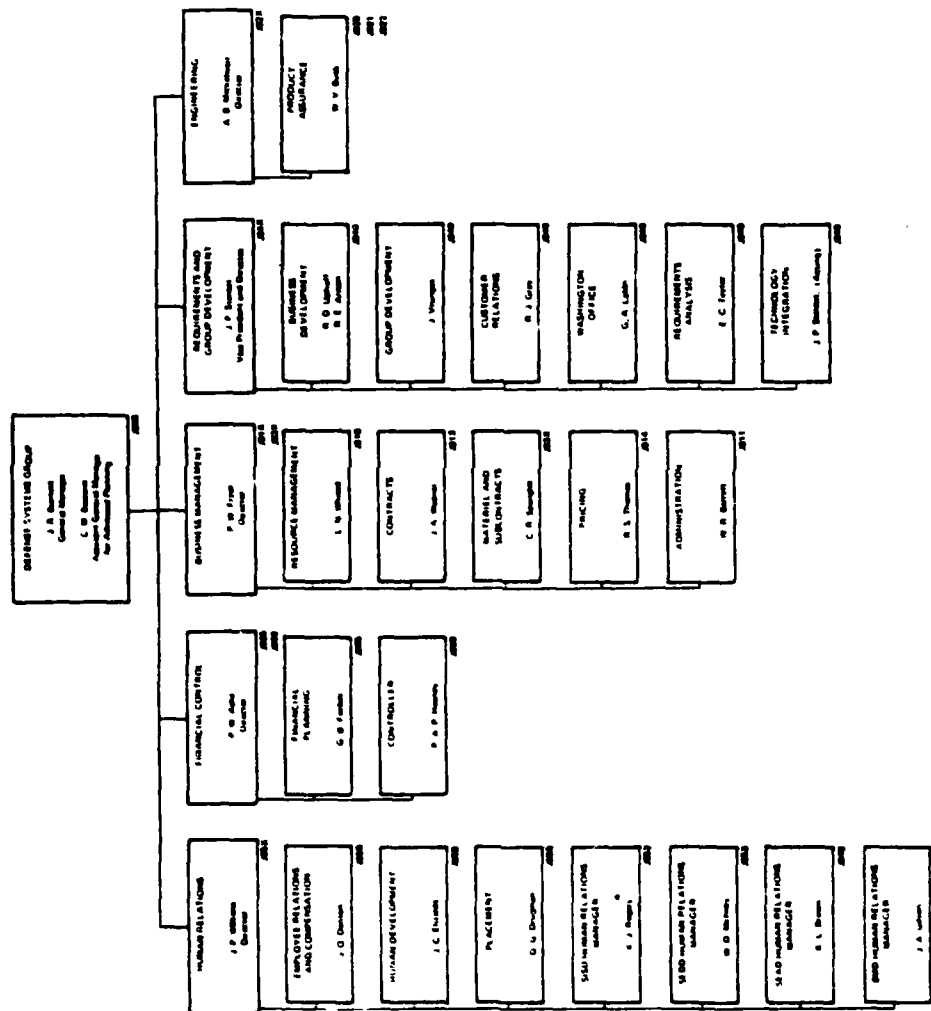


FIGURE 4.5 TRW Defense Systems Group
SOURCE: Interview with TRW Products Assurance Personnel

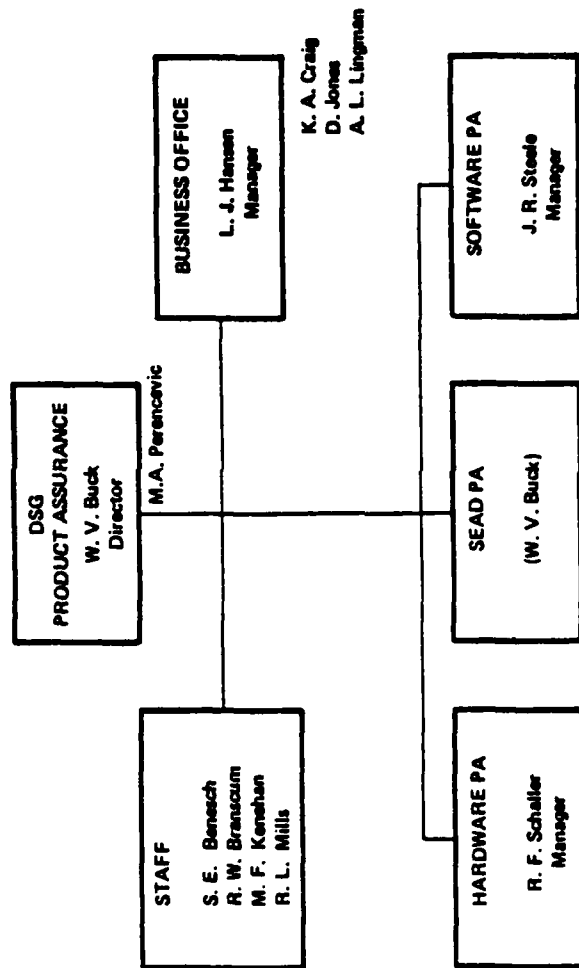


FIGURE 4.6 TRW Products Assurance Department
SOURCE: Interview with TRW Products Assurance Manager

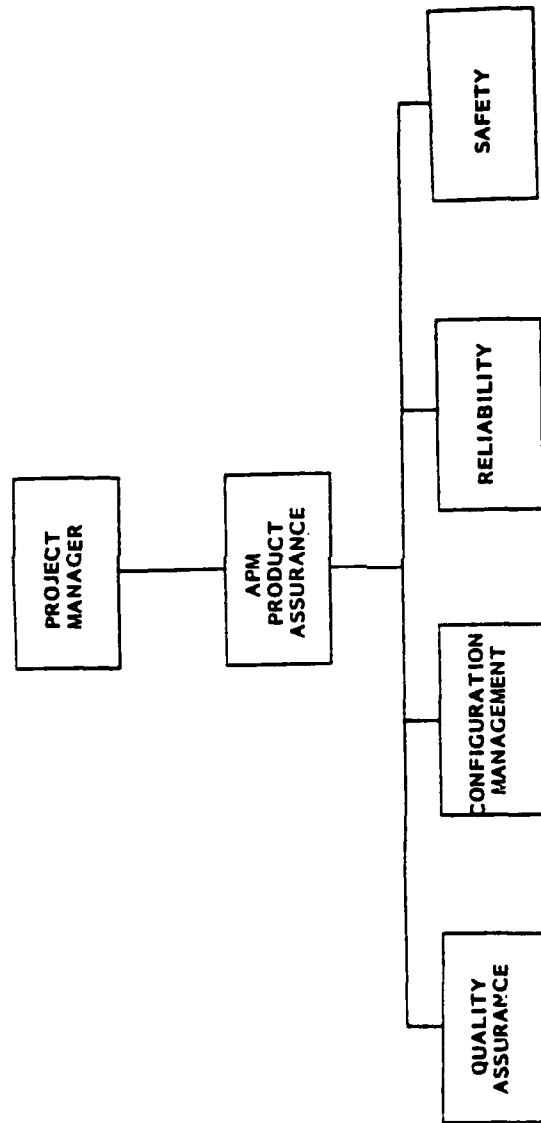


FIGURE 4.7 TRW Product Assurance Standardized Project MBS
SOURCE: TRW Status Report on Standardization of Quality Assurance Functions
Task, 20 April 1982

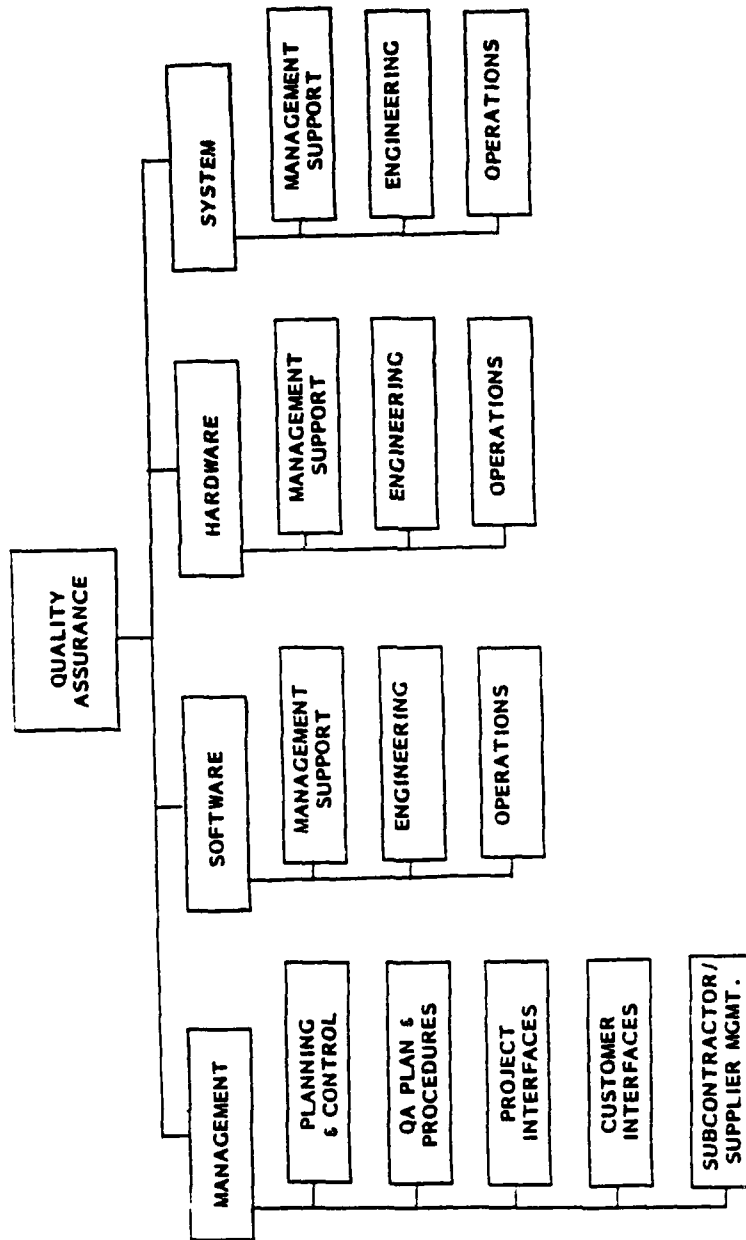


FIGURE 4.8 TRW Quality Assurance Standardized Project WBS
 SOURCE: TRW Status Report on Standardization of Quality Assurance Functions Task,
 20 April 1982

further subdivided into management, software, hardware and system.

When working on military contracts, the company must follow specifications required for contract award. One of these specifications is MIL-S-52779A "Software Quality Assurance Program Requirements" of 1 August 1979. This document states the requirement for the establishment and implementation of a software quality assurance program. It is hoped that this program could be tailored, economically planned and developed in conjunction with the contractors programs of this type. The contractor is required to document this program in the form of a software quality assurance plan which meets its specifications. This plan has to identify organizational responsibility and authority for its execution and make timely provisions for special needs (controls, tools, facilities, skills, etc.). Because this is part of the contract, it is considered to give the products assurance organization its authority over the project.

2. Management and Software Areas of the Project

The standard duties expected to be performed by the personnel in the management area of the project are as follows: (Figure 4.9)

a. Planning and Control

(1) To provide direction and participate in the generation of quality assurance input into the project implementation plan, project schedules, documentation plans and other similar documents.

(2) To define the quality assurance tasks and assign the appropriate personnel. To monitor their performance and prepare status reports.

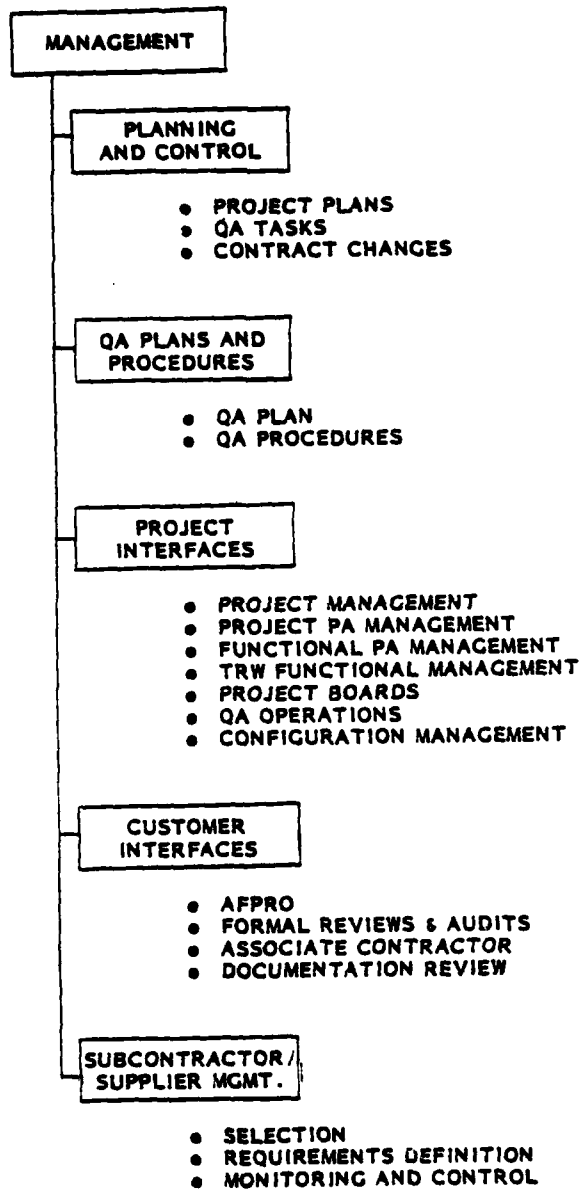


FIGURE 4.9 TRW Quality Assurance Project WBS - Management Detail
 SOURCE: TRW Status Report on Standardization of Quality Assurance Functions Task, 20 April 1982

(3) To monitor all actions in conjunction with contract and engineering changes.

b. Quality Assurance Plans and Procedures

They are required to direct the generation of the quality assurance plan which follows the controlling government specification MIL-S-52779A, to review, maintain, and update it throughout the project's life. This plan is required to address:

(1) Tools, techniques, methodologies and records to be employed in the performance of the work to support the quality assurance objectives.

(2) Procedures by which design documentation is reviewed to evaluate design logic, fulfillment of requirements, completeness, and compliance with specified standards.

(3) Contractor's procedures for formally approving or certifying the description, authorization and completion of work performed under contract.

(4) Documentation of standards, programming conventions and practices to be used for all software.

(5) Documentation of the contractor's procedures and controls for handling of source code and object code and related data in their various forms and versions.

(6) Documentation of contractor's procedures for preparation and execution of reviews and audits necessary in establishing traceability of initial contract requirements.

c. Project Interfaces

The management detail addresses the interfaces between project manager, assistant project manager, sub project managers and others in conjunction with the project. They attend the staff meetings and respond to action items.

d. Customer Interfaces

The management detail works with the customer representative offices, hosts their visits and formal reviews and take care of documentation to and from the customer.

e. Subcontractor and Supplier Management

Figure 4.10 delineates the duties of the personnel in the software area of the project. The three groupings are:

- (1) Management Support -- carries out duties in support of the management section of the project.
- (2) Engineering
 - (a) Identify and define the quality standards and procedures that will be followed during the design, development, programming, testing and documentation stages.
 - (b) Identify software tools and special methodologies that would be used in performance of quality assurance task. Establish procedures for their use and ensure their use during the project.
 - (c) Participate in definition and implementation of a software problem reporting, analysis, correction and control system.
 - (d) Participate in formal reviews, project boards and customer boards.
 - (e) Maintain records and files of documentation review for adherence to standards.
- (3) Operations

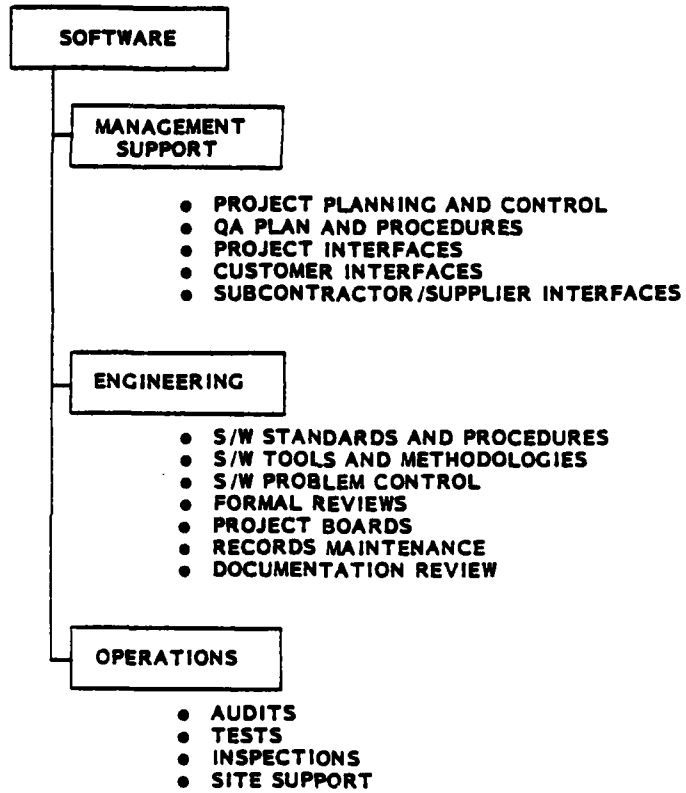


FIGURE 4.10 TRW Quality Assurance Project WBS - Software Detail
 SOURCE: TRW Status Report on Standardization of Quality Assurance
 Functions Task, 20 April 1982

- (a) Perform audits on project activities.
- (b) Participate in each level of software testing as designed by the quality assurance plan and perform surveillance activities.
- (c) Perform visual inspections of all software products purchased with hardware from supplier.
- (d) Perform quality assurance function at each site and remote site for testing.

If, during any documentation audit, a discrepancy is found, the discrepancy is documented and is taken first to the responsible designer. If, in a certain amount of time, the error is not corrected, the problem is taken to the next level in the project organization. The problem will travel up the organization until the discrepancy is corrected even if it means going outside the project's environment.

Approximately 2 to 5.5% of the entire project's funds is charged to quality product assurance, but it is the opinion of the managers of quality assurance in the TRW company that the cost of quality assurance is zero.

Once a product has been accepted by the customer, with the signing of defense form DD250 Material Inspection and Receiving Report, the legal obligation of TRW is ended. If any problems arise after release, the customer pays to have more work done.

Reference

Personal interview with Mr. William V. Buck, Product Assurance Manager; Mr. Samuel E. Benesch, Department Manager Product Assurance; and Mr. Martin F. Kenehan, Senior Staff Engineer of the Defense and Space Group of TRW, Redondo Beach, California on 7 May 1982.

C. IBM

1. Organization

Figure 4.11 shows the structure of the IBM organization as of March 1982. It shows that, under the staff level, the company is divided into two major areas, marketing and service and manufacturing and development. Under these areas, the grouping of divisions start in which, under the information systems and technology group, the general products division exists.

The general products division, with its headquarters located in San Jose, California, is responsible for the development of all hardware and software products at IBM. It has two development laboratories, one located in Santa Teresa, California and the other in Tucson, Arizona. (Figure 4.12)

The general products division is headed by a president with a vice-president in charge of each operational department including: hardware, software, manufacturing, financing, support and products assurance. Heading each development laboratory is a center manager with functional managers in charge of each department below him. Within each of the development centers, a functional manager in charge of products or quality assurance.

The quality assurance department within this organization is completely independent of other departments. The software products developed in these laboratories lie within the environment or operational tool area (Figure 4.13) and they are produced in all of the major programming languages. The quality assurance group does have authority over products that are new and are about to be announced and over products that are being shipped to customers. If this group does not agree that a product is ready, it is not released.

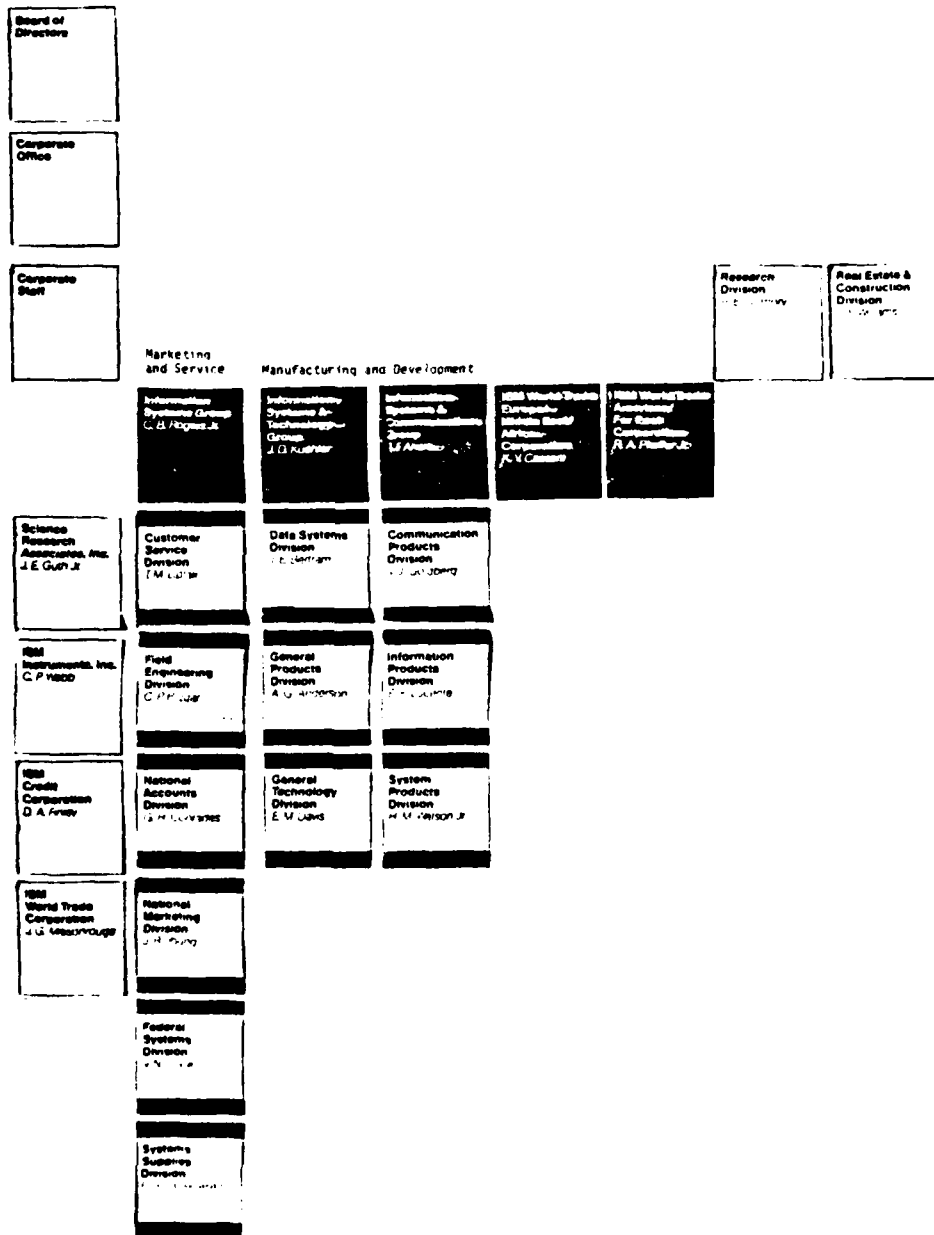


FIGURE 4.11 IBM Organization

SOURCE: Interview with IBM Products Assurance Personnel

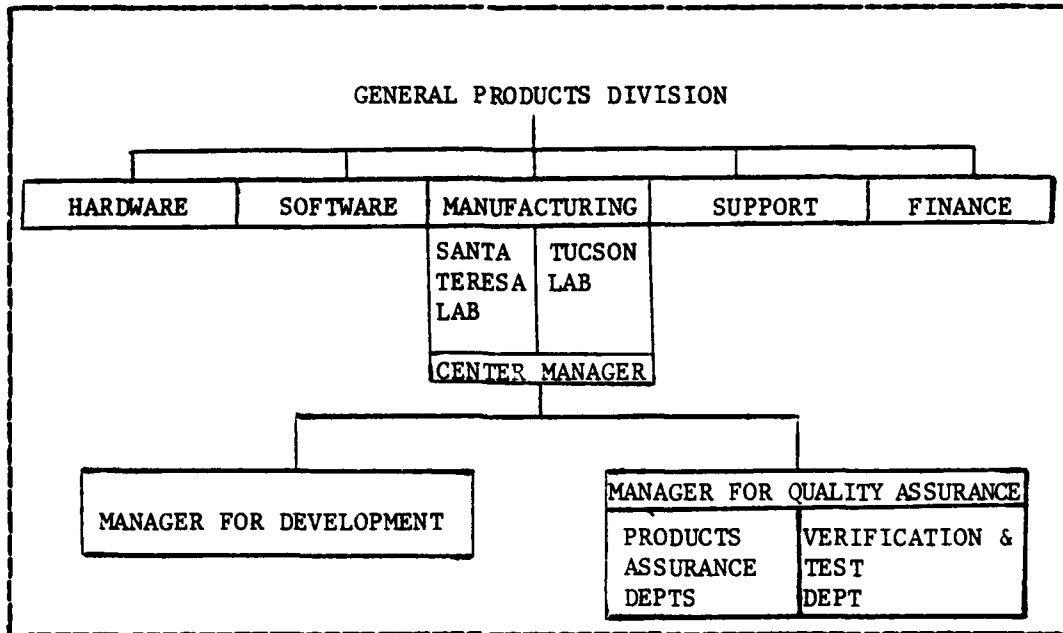


Figure 4.12 IBM General Products Division

Source: Interview with IBM Products Assurance Personnel

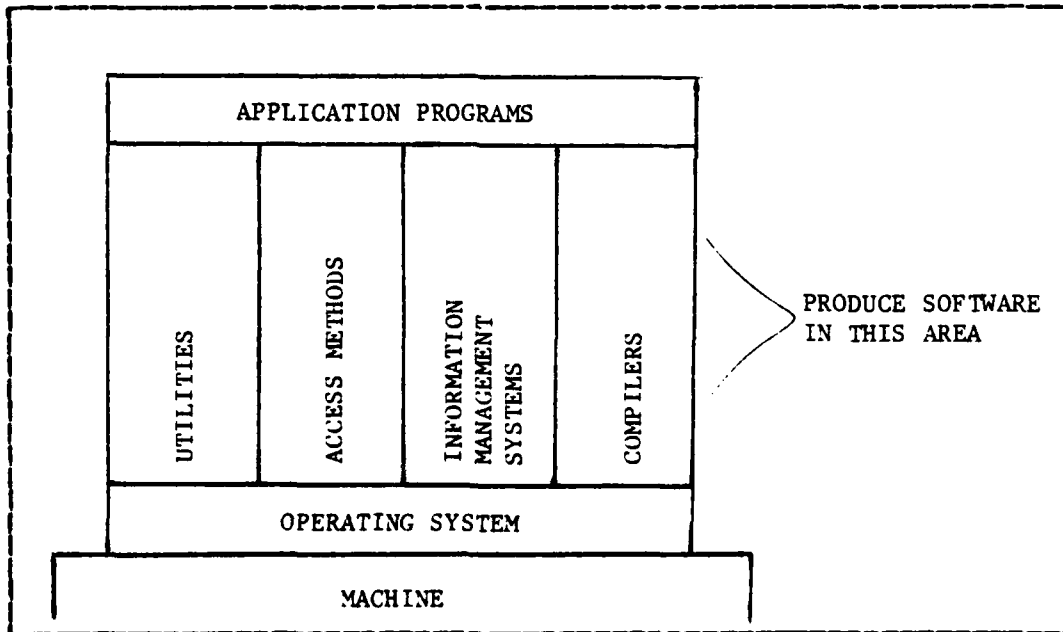


Figure 4.13 IBM Software Area of Development

Source: Interview with IBM Products Assurance Personnel

The decision for product release is not driven by any other factors.

The quality assurance department is divided into three divisions, two of which are products assurance, and the other is verification and testing. Every software product developed is divided between the two product assurance divisions. The number of people assigned is a function of the project's size and their schedule depends on that of the developers. At the end of the development cycle, all products go through the verification and testing division.

2. Quality Assurance and Design Interface

The quality assurance group interfaces with the program developers throughout the entire development cycle. (Figure 4.14) The people within this group have no prerequisite skill requirement and most have varied backgrounds ranging from programming expertise to marketing skills. To do their job, they depend mainly on their experience and gut feelings. It is not considered necessary for them to have a programming or computer engineering background because it is very rare that they have to inspect the actual code itself. Within each development department are performance groups who examine the code and test it periodically throughout the development cycle.

The managers of the development groups depend on the people from products assurance for their objectivity and do not view them as a resource tool. These products assurance people contribute to the product in the following ways:

a. Planning

Before any work can be started, a project plan has to be put together in which the programmers have to claim which development style out of a possible three will

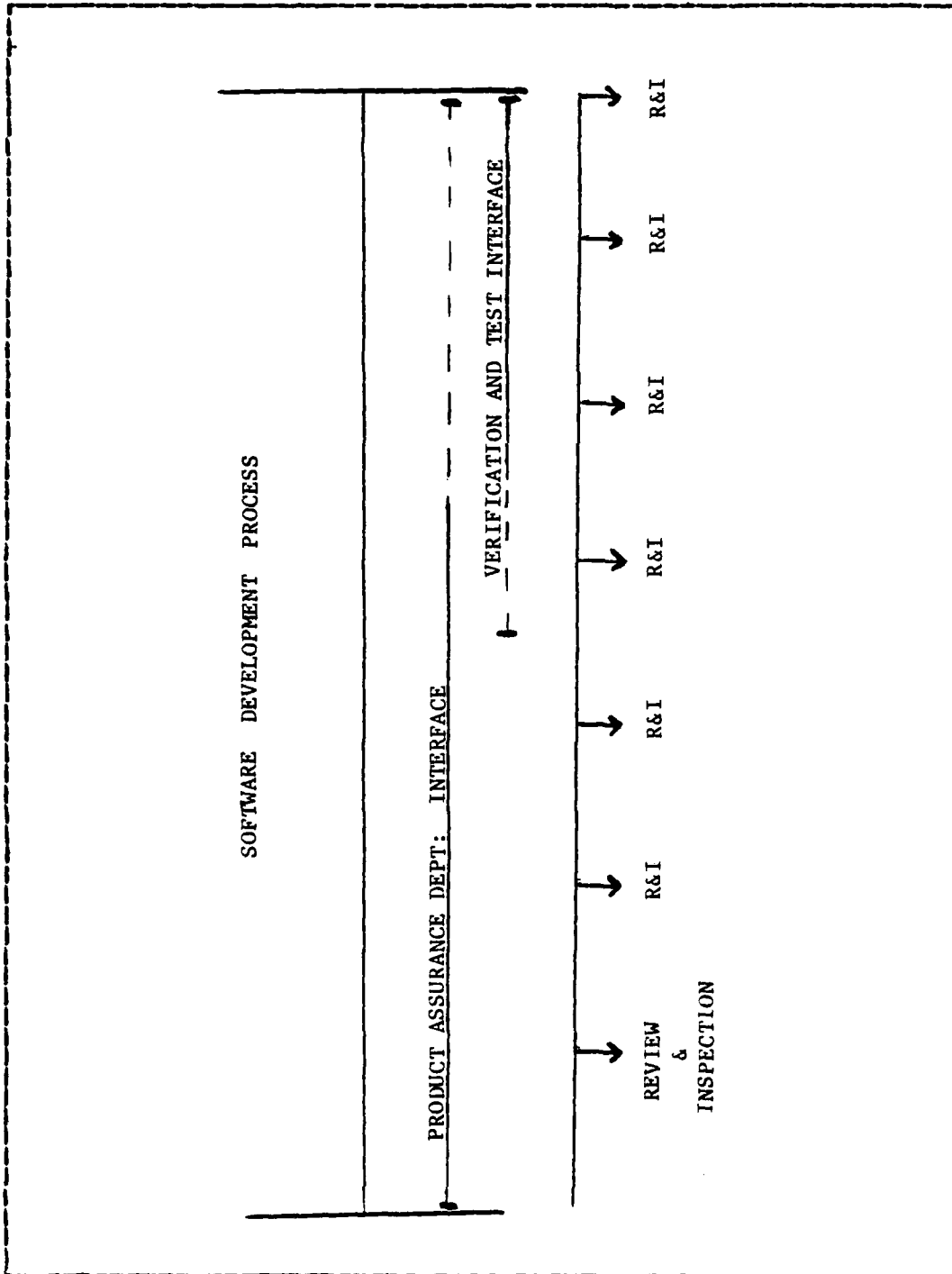


FIGURE 4.14 IBM Software Development Process
 SOURCE: Interview with IBM Products Assurance Personnel

be used for this project. This plan is named the Comprehensive Evaluation Plan (CEP) which also takes into account the quality assurance procedures, use of resources, and the project's schedule. It is considered the main planning document and has to be approved by the products assurance division before the project is started.

b. Early Warnings

If at any time during the development cycle, the quality assurance inspector sees anything which might keep the program development group from keeping schedule, they notify the project manager.

c. Value Added

If, during the process, the quality assurance people feel that something could be added to the software to enhance or improve it, they inform the development group.

d. Education

The education of the programmers on possible development tools, whether developed in house or externally, is carried out by this organization.

IBM sets standards requirements that have to be built into the products, but there is flexibility in their use because it is left to the discretion of the programmer.

The verification and testing people carry out their functional testing at the end of the development process, performing basically user oriented tests. Their main objective is to debug these products of any user oriented problems.

Besides the product assurance, performance group, and verification and test groups interfaces, there is still another built-in device for insuring quality products.

A Review and Inspection process (RSI) is carried out by the programmers themselves throughout the development cycle. It is carried out either in a formal manner in which a meeting is held with the programmers and a moderator and they discuss the program and its progress in depth, or it can be held on an informal basis with only the programmers' immediate peers present. A representative of the product assurance division is required to attend these meetings.

3. Operations

Once a product has been released, the field engineering division is responsible for remedying any problems experienced by the customers in use of the product. This division is also responsible for maintaining a historical tracking record on problems with the software products once in the field. If a product is to be renewed or enhanced, the products assurance people can request this historical information, but they are not required to keep track of it.

If a completely new product is released by the company for which the users would require training, the responsibility for this training is assumed by the marketing division. Requests for new products are not received directly by the development laboratories, but through the two main IBM user groups, SHARE and GUIDE, which meet twice yearly to discuss problems and possible ideas for new products. The marketing division is also constantly carrying out surveys of customers for new product ideas.

The people of the quality assurance department thought that their main objective was to maintain a wide range perspective of the product development process and never to become overly involved with details.

Reference

Personal interview with Mr. Barron A. McDonald and Mr. Norman Towns of the products assurance group, IBM Development Center, Santa Teresa, California on 21 April 1982.

D. AMDAHL

Amdahl is a high technology company engaged in the state-of-the-art design, development, manufacturing, marketing and maintenance of large mainframe computers, software and communication systems. These products are used by large computer users in the full spectrum of commercial and scientific data processing environments.

The company's central processing unit's design strategy is to focus on the development of efficient design architecture for high performance, dependability, and flexibility for future enhancement of the product.

The company's communication systems division designs and manufactures digital communication networks which allow users to interface with multiple geographically dispersed systems.

Amdahl also offers a number of services to its customers. There are programs for cross training support with specialists in both hardware and software disciplines. There are also expanded educational offerings with tailored training to enhance Amdahl product support.

The company's software development and program enhancements ensure compatibility of its hardware products to the most widely used systems, and other software products are aimed at increasing productivity of the user.

1. Organization

The software department is a part of the engineering division at Amdahl. The software quality assurance group is a part of that department and it consists of five people. (Figure 4.15) The main purpose of software in the Amdahl world is for architectural interface of its product with the customer's system. Because of this, the software development group does not have to start with any top down design of its product but to develop complement software in order to tie the hardware products together. The driving force for the development of software in this company is the innovative hardware of its competitors, such as IBM. The authority of this organization depends on its credibility and expertise. The products that they release have proven themselves in the market place.

2. Development Interface

The quality assurance group of Amdahl's main interface with the development group comes at the end of the development cycle during the testing and measuring. They also take part in all technical reviews throughout the new products development. The quality assurance group insures that the program is "packaged correctly" for installation. This means that the software product meets all the standards of their competitor's system.

3. Operations

For new software about to be released by this company, they have what is known as the early support program. The program enables the developers to take the software into the field, test and debug it on the system to which it is to be applied before it is announced.

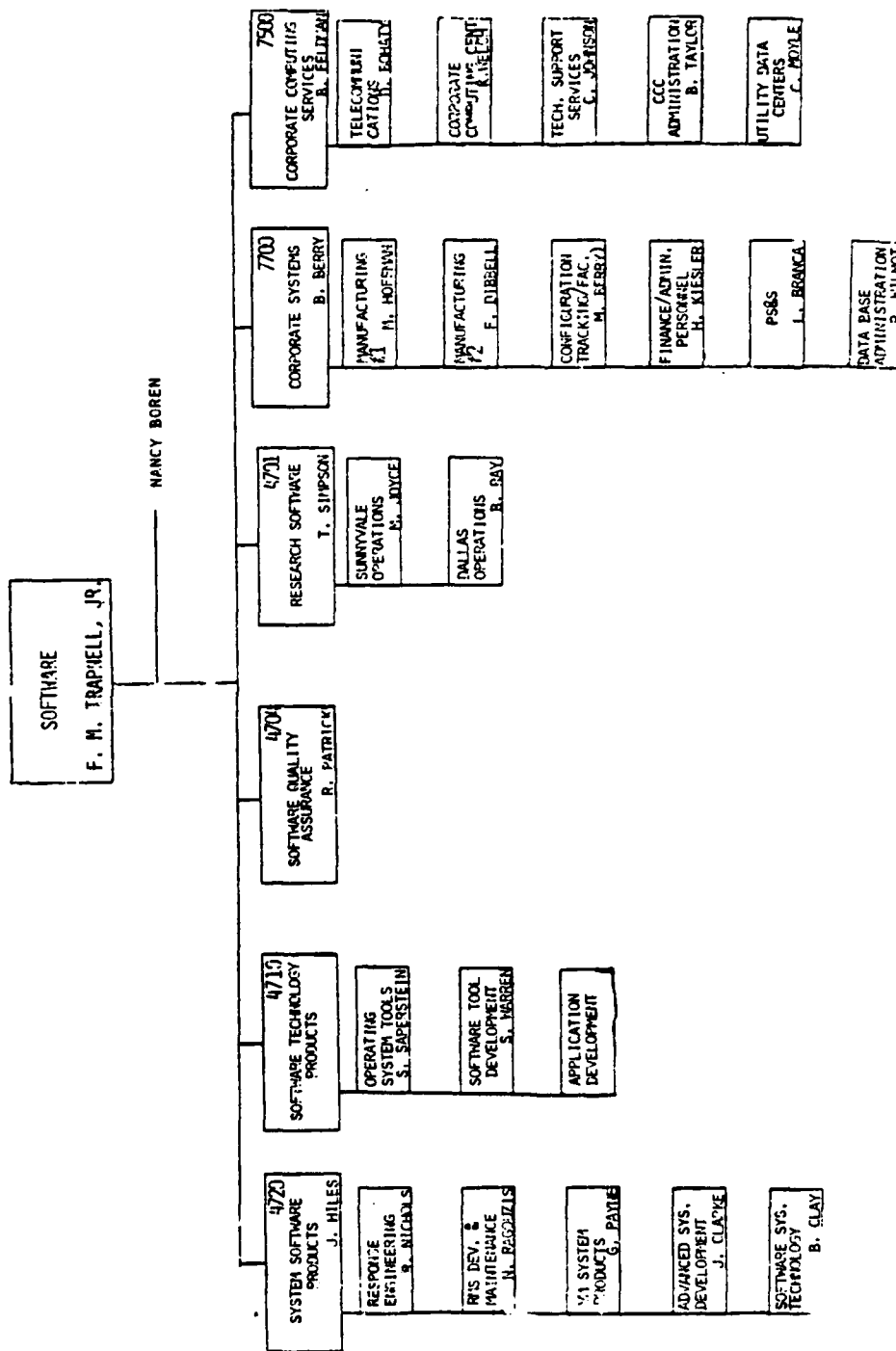


FIGURE 4.15 Amdahl Software Department
SOURCE: Interview with Amdahl Software Quality Assurance Manager

After installation, if there are problems with the software, the field units of Amdahl handle them. There is the Amdahl warning system and maintenance tape, which is maintained by the field units and, if there is a major problem, the software is sent back to the development center for rework.

No training is carried out for the Amdahl products, but there is a tremendous in-house training effort on competitors' equipment.

Reference

Interview with Mr. Richard L. Patrick, Manager, Software Quality Assurance Group at Amdahl's.

V. ANALYSIS CONCLUSION AND RECOMMENDATIONS

This chapter gives the reader an analysis and summary of the interviews with the commercial computer companies and a comparison with the FMSO environment. At the end of the chapter, conclusions and recommendations are given.

A. QUESTIONS FROM INTERVIEW:

1. Where does the quality assurance group fit into the company's organization?

a. Hewlett Packard

The products assurance group is a part of the data systems division and is on the same level as engineering, manufacturing, marketing, and other departments of this division. The products assurance group fits into the company's organization in a line function position.

b. TRW

The products assurance group is a part of the engineering division. This group fits into the company's organization in a staff function.

c. IBM

The products assurance department is a part of the software development center. It is positioned on the same level as the development department of the center, in a line function.

d. Amdahl

The software quality assurance group is a part of the software department. It is positioned on the same level as the research and development groups. The software quality assurance group is in a line function position.

e. FMSO

The quality control branch exists in the management department, Code 92. It is in a staff function.

2. What type of authority/power does the quality assurance group have over their software product?

a. Hewlett Packard

This group's power relies on its ability to persuade management that the product is not ready and its reputation.

b. TRW

The authority of this quality assurance group is given by a contractual requirement, MIL-S-52779A "Software Quality Assurance Program Requirements."

c. IBM

The products assurance group has complete authority over software product. If this group feels that the product is not ready, it is not released.

d. Amdahl

The software quality assurance group's power over the product depends on the group's credibility and expertise.

e. FMSO

The quality control group exercises administrative power over products. It insures that the quality assurance check-off list is properly filled out and that the product meets specifications.

3. What qualifications do the people in the quality assurance group have?

a. Hewlett Packard

Their quality assurance personnel are required to have enough education and experience to be programmers and designers.

b. TRW

No specific qualification required.

c. IBM

No specific qualification required.

d. Amdahl

No specific qualification required.

e. FMSO

Personnel in the quality control branch are expected to have a complete knowledge of the system development process, from all aspects.

4. How does the quality assurance group interface with the design/development group?

a. Hewlett Packard

The quality assurance personnel are a part of the product development group and work with the product designers throughout the development cycle. They are required to produce a quality assurance plan which states the measurements of the quality objectives and to participate in the product testing on both the functional and system levels.

b. TRW

An assistant project manager is assigned to every project, with his own staff, to coordinate and participate in the quality control functions required in the project. They perform audit testing of the product and participate in all technical reviews.

c. IBM

The product assurance people interface with the software development personnel throughout the development cycle. They approve the program development plan and keep management informed of anything that might affect the project's schedule. They do not participate in product testing, but there are two third party groups, the performance group and the verification and test personnel, who carry out this function.

d. Amdahl

The software quality assurance group interfaces with the development personnel at the testing and measurement end of the development cycle. They insure that the product is "packaged correctly" before release. They are required to attend and participate in all technical reviews during the development of the product.

e. FMSO

The quality control branch checks the functional description and system specifications administratively. They insure that the quality control check-off list is filled out properly and participate in product testing on a very infrequent basis.

As shown in the question, all of those interviewed, except TRW and FMSO, had their software quality assurance groups in a line function position in the organization. It should be noted that the products assurance group of TRW was in charge of a line management staff which was assigned to each product to perform in a line function. In FMSO, there is only the staff group.

It is the opinion of the author of this thesis that questions 2, 3, and 4 tie in together. In all the companies interviewed, the quality assurance group is considered and functions as an integral part of the development team. They work with the development personnel throughout the development cycle, relieving any advisory situation.

If the personnel in the quality assurance group do not have the expertise to carry out testing of the product, a third party in the company's organization do. Development personnel cannot be expected to be completely objective about their own product to perform its testing.

Because the quality assurance personnel work alongside the development people and perform some form of audit

function, their opinion has credibility with the development people and management. This has a direct effect on their authority over the product.

In FMSO, the quality control branch does not become an integral part of the development team. They rarely perform any auditing function on the product. The development people in the CDAs carry out all testing. If the quality assurance check-off list is completely filled out, the quality control branch has no real justification for stopping the product's release.

5. What tools, methodologies, or techniques does the quality assurance group use to do their job?

a. Hewlett Packard

No tools, methodologies or techniques were used that were unique to the quality assurance function.

b. TRW

No tools, methodologies or techniques were used that were unique to the quality assurance function.

c. IBM

No tools, methodologies or techniques were used that were unique to the quality assurance function.

d. Amdahl

No tools, methodologies or techniques were used that were unique to the quality assurance function.

e. FMSO

No tools, methodologies or techniques were used that were unique to the quality assurance function.

On this question, none of the companies interviewed stated that they used anything unique to the quality assurance function. The quality assurance personnel were knowledgeable of tools and techniques that could be used by the development programmers which, from their viewpoint, aided in the quality of the software because it helped the

programmers write better programs. These tools and techniques were acquired through the survey of computer science literature or developed within the company and passed on. No company interviewed was willing to share any of these tools with the author of this thesis because their tools were of a proprietary nature.

There are companies that develop tools and provide services which aid in the areas of programming and quality assurance. One such company is Software Research Associates (SRA), headquartered in San Francisco, California. A description of the purpose of this company and its activities is provided in Appendix B.

6. Are historical records kept of problems with software products after their release and who in the company's organization keeps them?

a. Hewlett Packard

No records of this type are being kept at this time.

b. TRW

No records are kept of product problems after release.

c. IBM

Historical records of problems are kept by the field engineering division.

d. Amdahl

A maintenance tape of problems is kept by the field engineering division.

e. FMSO

Records are maintained by the quality control branch through analysis of Program Trouble Reports (PTR).

7. Who handles problems with software after release, and how are such problems handled?

a. Hewlett Packard

Problems are handled by field engineering activities who build "work arounds" for customers if necessary. If there is a critical problem, the software is returned to the development group for repair.

b. TRW

There is no legal obligation on the part of the company to handle problems after a product's release. If a customer desires TRW to fix a problem after product release, the customer will be charged for the services.

c. IBM

All problems are completely handled by the field engineering division. The software is not returned to the development laboratory, no matter how critical.

d. Amdahl

Problems are handled by the field engineering group. If there is a major problem, the software is returned to the development personnel.

e. FMSO

The software is reported to the CDA and repaired.

8. If a brand new product is designed, who in the company's organization trains the customer on this product?

a. Hewlett Packard

Marketing division carries out training.

b. TRW

No training is carried out by the company after product release.

c. IBM

Marketing division carries out training.

d. Amdahl

Marketing division carries out training.

e. FMSO

Field training units go to activities from the CDAs.

A question that might have been asked during these interviews concerned the effectiveness of the company's software quality assurance program. The author did not ask this question because it would be improbable to expect an objective answer. This thesis did not offer a quantitative measure of these groups' performances to make its comparisons. The author's intent was to compare their view of the quality assurance organization's role and how they function.

B. SUMMARY CONCLUSIONS

The purpose of this thesis was to investigate the methods used by large commercial computer companies in the area of software quality assurance. The primary objective was to see if any of these practices could be used in FMSO's environment.

1. The greatest difference between the commercial companies and the FMSO environment was in management's view of what role or function a quality assurance group should take. In the commercial environment, the trend of thought is that the quality assurance role is a line function that could be controlled from a staff position. In FMSO, the quality assurance role is only being fulfilled through a staff position.

2. There was a difference in the way the quality assurance personnel interfaced with the development people. In the commercial companies, the quality assurance personnel became an integral part of the development team, their opinions and actions being a very valuable management device to project managers. In FMSO, the quality control branch from its staff position, does not become a part of the development team, thus creating an adversary environment.

C. RECOMMENDATIONS

1. It is the opinion of the author that FMSO should change the quality control branch's position from a staff to a line function. As shown by the interviews, this is the trend of thought on the position of an organization of this type in a software production environment of today.

2. In the FMSO environment, to convert the quality control branch's position from a staff to a line function, an increase in the branch's size would be necessary.

This could be accomplished in two ways. One way would be to hire more people to increase its size. The other manner would be to take people already in the CDAs and assign them the specific job of quality assurance. The second manner may be more effective because these people would already be acclimated to the FMSO environment and have the knowledge of practices in their own CDA. People of experience and expertise could be chosen and, since already known by the personnel in their development groups, would not be viewed as outsiders. They would be able to either carry out or be in charge of the auditing functions in the software development process. FMSO would not have to change its development process. The staff function or position could still be held in Code 92, but it would be in charge of a line quality assurance organization in the CDAs.

3. The Quality Assurance Checklist could be used as the quality assurance group's work description document. They would be in charge of carrying out the elements of the checklist in a third party auditing function. Because the checklist points out the segments during the development process where surveillance for quality is important and the list covers the entire development process, it would be a very useful guideline.

Looking at the first element of the checklist, the scope of release, a separate checklist should be made up for each of the four levels of projects to cut down on confusion of which elements should be done for which project.

The elements stated in the checklist are also very broad. A more specific description of the tasks that would have to be carried out by the quality assurance personnel should be promulgated. This description of tasks would also have to coincide with the steps of the system development process.

The quality assurance staff function in Code 92 should monitor the projects progress and be involved in it's POA&M phase. They should have final authority over the this milestone plan. They should attend all project internal reviews and participate in, if no more than monitor, all testing.

4. With the quality control branch in its present position, it is the opinion of the author that it is a waste of this organization's time and resources to be involved in the collection and analysis of Program Trouble Reports which record problems after software release. The only organization to which this type of information is important is the organization which developed it and has to fix it. This organization should expend its energy in the maintenance of these types of records, and the quality assurance people should monitor them.

5. An effort should be made by FMSD to maintain records of in-house development tools that could be shared between the CDAs. The assistance of a tool development organization, such as Software Research Associates, could be sought to help them in the areas of program development and software quality control tools.

6. If any justification need be supplied for acquiring resources to accomplish these goals, the requirements

invoked on civilian contractors for a software quality assurance program, MIL-S-52779A, could be given. If the government requires this extensive a program for its civilian contractors, why not require it for itself?

APPENDIX A
FMSO SYSTEM DEVELOPMENT PROCESS

2.3.2 System Development Process (SDP) is the function by which FMSO transforms a Requirements Statement into a documented, functioning set of computer programs and procedures. FMSOINTNOTE 5230 of 21 Nov 1979 established the CDA Development Process Model provided as Figure 2-4. The CDA Development Process Model reflects all of the basic steps appropriate in ensuring that each CDA Tasking received by FMSO is effectively managed and results in a high quality product being released for use by the customer. The model covers all projects, large and small, new development or maintenance. However, it is anticipated that some of the steps in the model may not be applicable to all projects. Therefore, an explicit decision by the appropriate level of management is required in order to exclude process steps determined not applicable on a project.

2.3.2.1 Definitions of Figure 2-4 Symbols

2.3.2.1.1 "△" (Line Management Review and Approval). This responsibility is assigned to FMSO Department line managers that have been tasked with the development of a Project or resolution of a Program Trouble Report (PTR).

2.3.2.1.2 "□" (Top Management Review (Optional)). This responsibility is assigned to a Project Review Board appointed by the Commanding Officer to review designated Command-interest projects. The Commanding Officer will be final approval authority on these projects.

2.3.2.1.3 "⊕" (Management Department (Code 92) Project Tracking). This responsibility is assigned to the Management Department to administratively act as FMSO's front door on all Project and PTR tasking, and to track progress for the Command via the standard FMSO project status tracking reporting system of specific Command-designated projects.

2.3.2.1.4 "○" (Management Department (Code 92) Project Management). This responsibility is assigned to Code 92 for projects that have significant critical interfaces in two or more Departments for which the Command has not specifically designated a Project Manager. Project Managers will be the Command focal point for the project and provide the coordination necessary to ensure that all significant/critical interfaces are resolved.

2.3.2.1.5 "●" (Management Department (Code 92) Quality Control (Q/C)). This responsibility is assigned to the Management Department to assure that all line management tasking has been achieved within FMSO Q/A standards.

AD-A123 840

SOFTWARE QUALITY ASSURANCE AND THE FLEET MATERIAL
SUPPORT ENVIRONMENT(U) NAVAL POSTGRADUATE SCHOOL
MONTEREY CA J H HARRINGTON JUN 82

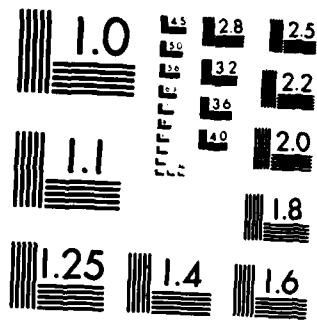
22

UNCLASSIFIED

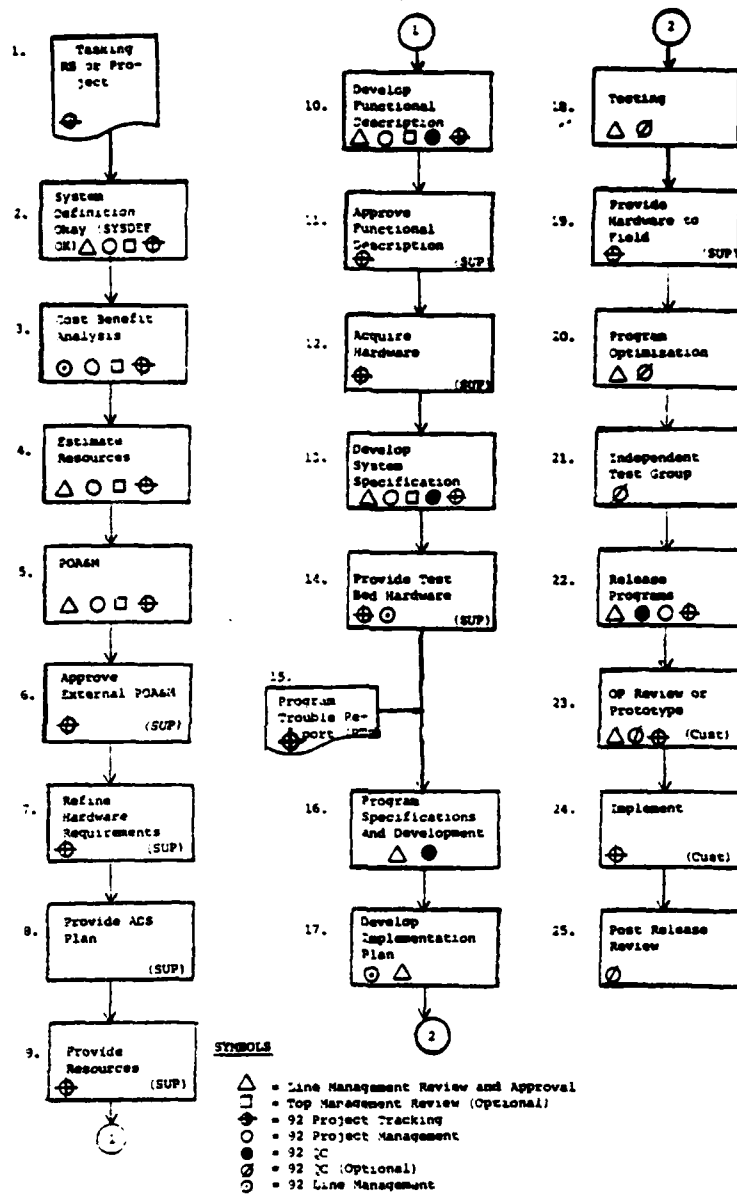
F/G 14/4

NL

END
DATE
FORMED
3 83
DTI



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



2.3.2.1.6 "Ø" (Management Department (Code 92) Quality Control (Q/C Optional)). This responsibility is assigned the Management Department to perform selectively at their discretion on designated development process events.

2.3.2.1.7 "⊙" (Management Department (Code 92) Line Management). This responsibility is assigned to the Management Department to perform line management functions for designated development process events for all projects where applicable.

2.3.2.2 Descriptions of SDP Model Steps

2.3.2.2.1 Tasking Requirements Statement (RS) or Project. The development of a Requirements Statement (formerly entitled the Systems Policy and Concepts Statement) is the responsibility of the system proponent; however, current Command policy is to provide assistance in the preparation of the RS by the system proponent (where warranted and approved by the appropriate Department Director or Project Manager). The RS or project tasking document will be logged in by Code 92 as a Project Tracking function and forwarded to the responsible department(s) for acceptance or rejection.

2.3.2.2.2 System Definition Acceptable (SYSDEF OK). Line management will review the tasking document to ensure that it contains sufficient information from which to develop a functional description, cost benefit analysis, plan of action and milestones (POA&M) (internal or external), resource estimates, and priority acceptability. If sufficient information is not provided, a letter citing tasking deficiencies will be sent by line management or by the Project Manager (if appropriate) to NAVSUP with a copy to Code 92 to stop Project Tracking. Tasking must contain the general definition of the target hardware/software environment to be used or it must be clear that an existing suite of hardware/software is intended. When tasking is acceptable and the project is a new development, is a new Application/Operation, changes disk files or teleprocessing, is estimated to exceed 1,000 manhours of FMSO effort, or may impact system software, a copy of the project will be sent to Code 94 to provide estimated costs or determine that system software is not affected. Code 94 will respond to application Departments within two working days in either case. When tasking is acceptable from all of the above, line management will return a copy of the project to Code 92, with total estimated costs annotated, for a Cost Benefit Analysis.

2.3.2.2.3 Cost Benefit Analysis. Code 92 will develop a Cost Benefit Analysis with the assistance of line management. If not cost beneficial, Code 92 will prepare a letter to NAVSUP rejecting the project, update Project Tracking records, and advise line management and the Project Manager (if appropriate) to stop further effort. CBA may be subsequently iterated at the discretion of Code 92 or line management.

2.3.2.2.4 Estimate Resources. Line management, including Code 94 if involved, will develop initial resource estimates and determine priority acceptability/required to perform the tasking. Resources include personnel, test bed and operational hardware, software, travel and overtime requirements. If there is a shortfall, line management or the Project Manager (if appropriate) will prepare correspondence (including an impact statement) to NAVSUP requesting additional resources or a change in priority. A copy of the letter will be forwarded to Code 92 for Project Tracking.

2.3.2.2.5 POA&M. Line management, including Code 94 if involved or the Project Manager (if appropriate), will develop internal and external POA&Ms for CO-designated projects as discussed in paragraph 4.1.5.4.2. Examples of POA&Ms are provided in Appendices 4.1-A-1 and 4.1-A-2. A copy of the POA&Ms will be retained by Code 92 for Project Tracking. The CBA, resource estimates, and (for CO-designated projects) POA&M will normally be done concurrently and included in a letter to NAVSUP including a commitment date for FMSO to complete the Functional Description (FD). In addition, FMSO line management or the Project Manager (if appropriate) will update external POA&Ms monthly for submission to NAVSUP. NOTE: A senior executive Project Review Board (PRB) has been established to execute FMSOINST 5200.7B. Line management will, on Commanding Officer-designated projects, provide or present to the PRB a System Definition Review in accordance with FMSOINST 5200.7B. When this is approved by the PRB and subsequently by the Commanding Officer, line management will prepare a letter for the Commanding Officer's signature to NAVSUP stating the official FMSO position).

2.3.2.2.6 Approve POA&M. Code 92 will monitor this event as a project tracking responsibility. When the approved POA&M is received from NAVSUP, the next three steps (i.e., refine hardware requirements, provide ADS plan, provide resources) will be initiated concurrently.

2.3.2.2.7 Refine Hardware Requirements. If required, NAVSUP will refine the hardware requirements at a level adequate for inclusion in an ADS plan. Code 92 will monitor this event for progress as a Project Tracking task.

2.3.2.2.8 Provide ADS Plan. If required, NAVSUP will develop or update an ADS plan and process it up the chain of command for approval. Although it is recognized that further FMSO development of the tasking should wait for ADS plan approval, this has proven to be impractical.

2.3.2.2.9 Provide Resources. If required, NAVSUP will provide resources and/or priorities necessary to execute the POA&M. Code 92 will monitor this event for progress as a Project Tracking task.

2.3.2.2.10 Develop Functional Description (FD). Line management will develop the Functional Description (FD) and submit to NAVSUP for approval, including refined estimates of resources per paragraph 2.3.2.2.7, above, with a copy to Code 92 for Project Tracking, Quality Control, and compliance with standards. Upon completion of the FD, line management or the Project Manager (if appropriate) will conduct a System Design Review. On Commanding Officer-designated projects, the review will be provided or presented to the PRB in accordance with FMSOINST 5200.7B. Code 92 will provide or present an updated CBA as appropriate. When approved by the PRB and subsequently by the Commanding Officer, line management or the Project Manager (if appropriate) will prepare a letter to NAVSUP, for Commanding Officer signature, including an updated POA&M with a commitment date for FMSO to complete the System Specifications (SS).

2.3.2.2.11 Approve Functional Description. NAVSUP will review the FD and approve, approve with qualifications, or disapprove. This is the critical path to the development of the System Specification. NAVSUP will update resource requirements as required. Code 92 will monitor this event for progress as a Project Tracking task, if required.

2.3.2.2.12 Acquire Hardware. FMSO assists by estimating capacity needed for a representative site. NAVSUP coordinates with other NAVMAT or Fleet claimants, performs data call to all affected activities, and determines system-wide requirements. NAVSUP, directly or by notification to other claimants, initiates acquisition. Code 92 will monitor this event for progress as a Project Tracking task, if required.

2.3.2.2.13 Develop System Specifications (SS). Line management will develop the SS for release to customers with a copy to Code 92 for Project Tracking (if required), Quality Control, and standards review. In addition, at the completion of the SS, line management or the Project Manager (if appropriate) will, on CO-designated projects, provide or present to the PRB a Computer System Analysis Review in accordance with FMSOINST 5200.7B. In addition, Code 92 will provide or present an updated CBA if appropriate. When approved by the PRB and subsequently by the Commanding Officer, line management or the Project Manager (if appropriate) will prepare a letter to NAVSUP for Commanding Officer signature, including an updated POA&M with a commitment date for FMSO to make the program release.

2.3.2.2.14 Provide Test Bed Hardware. NAVSUP provides hardware and system software (if any) needed for program development and testing. Code 92 will coordinate or arrange the installation. Since this is the critical path to process event 2.3.2.2.16, program development can begin but not be completed if test bed augmentations or acquisitions are needed but not provided. Code 92 will monitor this process event on projects where test bed hardware/software is required as part of their Project Tracking function.

2.3.2.2.15 Program Trouble Report (PTR). PTRs will be received by Code 92, logged for PTR monitoring as part of their Project Tracking function, and forwarded to the responsible department for resolution. PTRs may affect any development process step in this model, and are discussed in detail in paragraph 4.2.5.

2.3.2.2.16 Program Development. Line management will develop Program Specifications (PSs), develop programs, perform unit testing, develop Program Maintenance Manuals (MMs), Users Manuals (UMs), and Computer Operation Manuals (OMs). PSs, UMs, and OMs will be released by line management to customers. Code 92 will provide administrative documentation release services including review of the documentation for completeness and compliance with documentation and system development process standards.

2.3.2.2.17 Develop Implementation Plan. The customer is responsible for the formulation of a systematic implementation plan based upon individual customer requirements. However, FMSO must assist the customer on some projects by developing a proposed plan and negotiating the issuance of a plan by the customer. Negotiations on the implementation plan will be performed by Code 92 as a line management function for designated projects, with assistance and review/approval by line management in affected departments. Implementation plans required on projects not designated for Code 92 development will be developed by the appropriate department line management.

2.3.2.2.18 Testing. Test Plans will be developed and string tests and/or system tests will be performed by line management. Code 92 will selectively review test plans and test requests for compliance with Quality Assurance standards and procedures.

2.3.2.2.19 Provide Hardware to Field Activities. NAVSUP and other claimants will provide required hardware capacity, if any, for field activity implementation. If required, Code 92 will monitor this event for progress as a Project Tracking function.

2.3.2.2.20 Program Optimization. Line management is routinely responsible for program optimization. Code 92 will select programs for review and processing through available optimization tools, and provide any solutions developed to line management by formal memo with logic changes specified. Line management will schedule and modify the programs in accordance with the solution provided or resolve with Code 92.

2.3.2.2.21 Independent Test Group. An independent test group will be established in Code 92. For Code 92-selected projects, entire release packages will be Quality Controlled for compliance with standards and procedures, clarity and ease of implementation. Also, all output products for the selected projects will be reviewed for quality. In instances where this effort will be accomplished prior to program release, line management will be advised during initial POASM development for inclusion in estimates. Recommendations for changes or corrections will be made to line management. Line management will make the changes or corrections in accordance with the Code 92 recommendations or resolve with Code 92.

2.3.2.2.22 Release Programs. Line management will release programs for Operational Review, Prototype or Implementation when all Q/A functions have been satisfied. When released for prototype, line management may withhold program releases to other customers for implementation pending successful prototype. Program Trouble Reports (PTRs) or Flash notification will normally be forwarded by a prototype activity to FMSO. Code 92 will provide administrative release services in accordance with current procedures, coordinating the release of environmental and application software and coordinating resolution of hardware and software interface requirements. In addition, Code 92 will review program releases for completeness, clarity and compliance with documentation and system development process standards as a Code 92 Q/C function. If required, Code 92 will monitor this event for Project Management or Project Tracking.

2.3.2.2.23 OP Review or Prototype. This is the responsibility of the customer and the primary participating responsibility of line management. When this occurs, Code 92 will participate at their option as a Code 92 Q/C function. If required, Code 92 will monitor this function for Project Tracking.

2.3.2.2.24 Implementation. Implementation is a customer responsibility with support provided by FMSO. Support will be provided by line management and/or Code 92 in accordance with the implementation plan. If required, Code 92 will selectively monitor this event for Project Tracking.

2.3.2.2.25 Post Release Review. As a Quality Control function, post implementation visits will be made to selected sites by Code 92, at their option, to determine whether the FMSO program release satisfied the tasking and whether the activity is using it properly. Feedback will be provided to line management. An attempt will be made to verify that the expected benefits were achieved.

APPENDIX B
SOFTWARE RESEARCH ASSOCIATES (SRA)

Software Research Associates

ABOUT SOFTWARE RESEARCH...

Software Research Associates (SRA) is an advanced technology research and engineering firm involved in software science, software engineering, software quality assurance, and software maintenance. The main activities of the Company are education, research and development, consulting, software tool design and production, and allied technical services. The Company has offices in San Francisco (headquarters) and Los Angeles, California.

Professional Development Technology Seminars...

The Company offers series of Professional Development Seminars on a periodic basis publicly, and on an in-house basis as well. SRA seminars are distinguished by their dedication to presentation of state-of-the-art software engineering techniques. Seminar offerings currently include: Software Quality Assurance, Applied Verification Techniques, Advanced Software Validation Techniques, Automated Software Engineering Tool Technology, and Software Maintenance Technology.

Research and Development...

Company researchers track the latest technical developments in a range of areas, including software production, software testing, and software maintenance, as well as other areas of software science and engineering. Typical Company research projects have included work in such areas as: techniques for validation of software engineering, systematic automation of the maintenance function, and general methodologies for comprehensive software testing and analysis.

Consulting and Technical Services...

Consulting for Company clients has ranged from evaluation of advanced computer architectures to the design of state-of-the-art software quality assurance organizations. The Company's approach to consulting emphasizes complete technical disclosure so that client organizations can make enlightened choices between technical alternatives. The Company also provides specialized technical services using advanced software engineering tools. Such services include software quality assurance, software testing, and software maintenance support.

Publications...

The Company publishes a quarterly newsletter, "Testing Techniques", that is distributed without charge to qualified technologists throughout the world. The new newsletter, "Quality Management Monthly", is focused on applying quality management techniques throughout the software life cycle. The Company also publishes (in printed and machinable form) the "Software Engineering Automated Tools Index" that describes some 500+ software support tools.

Software Engineering Tools...

The Company provides software production, testing and quality assurance, and maintenance tools for a variety of computer systems. The SETRAN system of structured programming preprocessing provides advanced control structures, automated program documentation, and automatic instrumentation. The TCAT system for software system test coverage analysis provides a quantitative base for quality assurance testing of COBOL programs. The ITB interactive software analysis and testing facility employs advanced analysis concepts for support of interactive software quality assurance. The ISUS system for semantic update and maintenance of software systems represents an advance in the state of the art in software configuration management and production control.

Revised: December 1981

P.O. Box 2432 • San Francisco • California 94126 • Telephone (415) 957-1441 • Telex No. 340-201

APPLIED VERIFICATION TECHNOLOGY SEMINAR

Quality in a software system is a function of logical integrity of every part of the system and of the system as a whole. Verification (or "proof") techniques are used to help establish the needed levels of integrity.

This new SRA Software Technology Seminar describes applications of the "proof of correctness" methods to software system quality control. In the correctness proving approach conjectures are formulated which express correctness with respect to specifications. The conjectures are generated by combining assertions about the program behavior with information from the program source text. These conjectures are then proved using information about the "meaning" of the programming and specification languages, mathematical logic, algebraic manipulation, and mechanical theorem proving. The methodology that surrounds the AFFIRM system will be described in detail.

This seminar is intended both for individuals in R&D positions and software engineering personnel working on highly reliable computing systems. A brief outline of the main topics in the seminar is:

PHILOSOPHY AND MOTIVATION: What is Verification?; Programs as Mathematical Objects; Unification of Verification and Design.

THEORETICAL FOUNDATIONS: Inductive Methods for Programs and Data; Proof Rules for Simple Control Structures; Axiomatic Specifications for Data Structures; State Transition Systems; Foundations of the AFFIRM Approach; Styles of Mathematical Proofs.

VERIFICATION METHODS: Inductive Assertions; Recursive Functions And Their Proofs; Proofs of Data Structure Properties; State Transition Proofs.

TECHNOLOGICAL SUPPORT: Verification Conjecture Generators; Formula Simplifiers, Rewrite Rules; Interactive Mechanical Theorem Provers; The AFFIRM Approach.

SURVEY OF APPLICATIONS OF VERIFICATION: Security Kernels; Distributed File Systems; Communication Protocols.

The instructor for this seminar will be DR. SUSAN L. GERHART, Technical Director of Software Research Associates, Los Angeles, California, a post she has held since October 1981. In this capacity she is concerned primarily with the application of verification technology to practical problems of software and system quality engineering.

Dr. Gerhart earned a B.A. from Ohio Wesleyan University, a M.S. from the University of Michigan, and a Ph.D. from Carnegie-Mellon University. After serving on the computer science faculties of the University of Toronto in 1972-73 and Duke University from 1973 to 1977, she joined the Program Verification Project at USC Information Sciences Institute. There she participated in the development of the AFFIRM Specification and Verification System, and served as the AFFIRM Project Leader in 1980-81.

For further information about this and other Software Technology Seminars please check the appropriate box on the enclosed Reader Response Form or call the Seminar Manager at Software Research Associates.

Notes: This and other SRA seminars can be presented "in-house" to larger groups of attendees at substantial overall savings and, in most cases, partially tailored to a client's specific needs. Please write for a copy of the SRA Software Technology Seminar Brochure.

Software Research Associates
P.O. Box 2432
San Francisco, CA 94126

Phone: (415) 957-1441 — Telex: 340-235

SOFTWARE ENGINEERING AUTOMATED TOOLS INDEX

As part of its continuing research activity in the Automated Software Engineering Tools area, Software Research Associates has assembled a comprehensive index of detailed data about a wide variety of software engineering support tools.

Available March 1982, this Software Engineering Automated Tools Index will provide detailed information on approximately 500 different software engineering tools.

Tools described in the Software Engineering Automated Tools Index fall into these major categories:

- o Software Requirements/Specification Tools
- o Software Design Tools
- o Software Implementation (Programming) Tools
- o Software Quality Assurance Tools
- o Software Maintenance Tools
- o Software Project Management Tools
- o Cross-Environment Tools
- o Miscellaneous Utility Systems

The Index also includes a comprehensive By-Name Index, a By-Category Index, and a complete By-Supplier Index. Available information about obtaining each software system is also included.

The information in the Software Engineering Automated Tools Index has been gathered from a wide range of sources (Government, Industry, and Academia) over the past three years. Each automated tool is described in a single "tool frame" that outlines such critical information as a tool's type and classification category, number of installations and price. Special features and exceptional characteristics, plus details about the needed execution environment. There are over 50 tool categories divided equally among the major system classes mentioned above.

The Software Engineering Automated Tools Index is provided in convenient 3-ring binder format, making it easy to survey the entire field of software engineering support tools, or to focus on just one area. This format makes it easy to incorporate quarterly updates that will be available to current users of the Software Engineering Automated Tools Index. The Two-Volume Tools Index costs are: U.S.A./Canada - \$185.00; Foreign - \$225.00. Costs for the quarterly updates (available on a subscription basis) are: U.S.A./Canada - \$85.00; Foreign - \$115.00.

For more information, or to reserve your copy of the Index, please check the appropriate boxes on the enclosed Reader Response Form.

Note: Machine processible versions of the Software Engineering Automated Tools Index are also available on special license arrangement. Please write SRA for details.

Software Research Associates
P.O. Box 2432
San Francisco, CA 94126

Phone: (415) 957-1441 - Telex: 340-235

ADVANCED SOFTWARE VALIDATION TECHNIQUES

Modern methods of software engineering require use of advanced methods to assure the installed quality of complex and critical software systems.

This seminar addresses major issues facing the Verification and Validation community in such areas as Symbolic Evaluation Methods, Verification Methods, Mutation Analysis, Functional Testing, Data Flow Analysis, and Domain Testing.

Besides describing how these advanced concepts can be used in various ways in Quality Management programs, this seminar provides researchers and appliers of these technologies with detailed information about the payoffs as well as the limitations of each method. For example, should mutation analysis be done on "large" programs? Or, should automated test data generation methods be used in a COBOL oriented environment?

Attendees will learn about state-of-the-art concepts, and will receive a comprehensive set of course notes and, in addition, a set of reprints from the current technical literature.

OUTLINE:SYMBOLIC EXECUTION TECHNIQUES

- Introduction
- Components of a Symbolic Execution System
- Problems in Implementing Symbolic Execution
- Detection of Anomalous Constructs
- Generation of Test Data
- Validation of Program Assertions
- Correspondence Between Programs and Specifications
- Partition Analysis
- Reliability of Symbolic Execution

ADVANCES IN VERIFICATION

- Definitions
- Verification by Case Analysis
- Inductive Assertions
- Proofs with Symbolic Evaluation
- Reasoning from the Structure of Data
- Practical Alternatives

MUTATION ANALYSIS

- Definition
- Testing Computer Programs
- Mutant Operators
- Relation to Other Testing Methods
- Practical Experience
- Systems That Have Been Built
- Relationship to Error Seeding

SURVEY OF PROMISING TECHNIQUES

- Functional Testing
- Data Flow Analysis
- Error Seeding
- Domain Testing Strategy

THE INSTRUCTOR

TIMOTHY BUDD is Assistant Professor of Computer Science at the University of Arizona. Professor Budd's research interests have focused on software engineering, program testing and validation techniques, and high level language implementation issues. He was a member of the research team which developed the Program Mutation Testing method, and has authored several papers on this and other areas of program validation technology.

Professor Budd has the Ph.D. degree in Computer Science from Yale University.

For further information about this and other Software Technology Seminars please contact the Seminar Manager at Software Research Associates...

(415) 957-1441

or write to...

Software Research Associates
P. O. Box 2432
San Francisco, California 94126

SOFTWARE QUALITY ASSURANCE TECHNOLOGY

Developing procedures for assuring that a software system has the best possible chance to operate without encountering "bugs" or "errors" is an activity that has formed a major focus of software engineering technology for nearly a decade. The goal of producing error-free software reliably and efficiently has eluded the best theoretical workers, while procedures for systematically analyzing and testing software through static and dynamic analysis has gained in popularity. Recent developments in software quality assurance make it possible to have a reasonable expectation that software meets minimum standards of testing. This seminar focuses on the concepts, tools and techniques, contemporary results, and prognosis for software quality assurance technology. Besides providing an investigation of state-of-the-art methods of program structure analysis (structured testing), the seminar presents a variety of material that deals with many alternative phases of software quality analysis. Attention is given not only to the theoretical aspects of the subject but also to practical results that can likely be achieved by use of known methods.

Attendees receive an extensive set of notes and a copy of the tutorial text Software Testing and Validation Techniques, by Edward Miller and William E. Howden. Attendees will gain an increased understanding of quality assurance processes and procedures and will learn techniques that can be applied immediately to quality assurance problems.

OUTLINE:INTRODUCTION AND OVERVIEW

- Introduction to Methodology
- History of Testing and QA
- Limits of Technology
- Overview of Methodology
- Theoretical Implications/Limitations

MANAGEMENT ASPECTS

- Organizational Setup
- Psychological Issues
- Level of Independence
- Typical Results of QA
- Case Studies
- Toolset Description
- Guidelines and Limits

CODE INSPECTION AND STATIC ANALYSIS

- Goal of Static Analysis
- Code Inspection Procedures
- Typical Code Inspection Rules
- Role of Static Analyzers
- Case Studies

TEST PLANNING PROCEDURES

Objectives of Test Planning
Role of Coverage Measures
Structure of Programs (Graph Theory)
Pure-Structured Programs' Test Plans
Hierarchical Decomposition Methods
Statistics and Inferences

TEST DATA SELECTION METHODS

Critical Values Identified
Optimum Choice of Specific Values
Theoretical Justifications
Relation to Proof of Correctness
Examples
Guidelines

COVERAGE ANALYSIS

Need for Coverage Measures
C1 Defined and Explained
Ct Defined and Explained
S1 Defined and Explained
Analysis for Ci/Si Evaluation
Basis in Graph Theory

DOCUMENTATION AND RETESTING

Need for Documentation
Data to Keep
Retesting (Regression Testing)
Change Control System
Test Documentation Tools

CASE STUDIES

Role of Interactive Test Support System
Small Example: ADD
Medium Example: KCLASS, LEXICAL
Large Example: FORM
Statistics and Reliability Issues
Recommendations

AGENDA FOR RESEARCHERSTHE INSTRUCTOR

EDWARD F. MILLER, JR., is Technical Director of Software Research Associates, San Francisco, California, a firm devoted to advanced computer technology and software applications. His interests include software engineering management, software testing technology, software maintenance technology, automated tool design and computer architecture.

Dr. Miller was previously Director of the Software Technology Center, Science Applications, Inc., San Francisco, and Director of the Program Validation Project at General Research Corporation, Santa Barbara, California. He received a BSEE at Iowa State University in 1962, an M.S. in Applied Mathematics at the University of Colorado in 1964, and the Ph.D. at the University of Maryland in 1968 where he was an Instructor from 1964 to 1968.

Dr. Miller is a member of the IEEE Computer Society, the ACM, SIAM and several honorary societies. He currently serves on several technical committees and is an Associate Technical Editor of COMPUTER Magazine.

For further information about this and other Software Technology Seminars please contact the Seminar Manager at Software Research Associates...

(415) 957-1441

or write to...

Software Research Associates
P. O. Box 2432
San Francisco, California 94126

AUTOMATED SOFTWARE ENGINEERING TOOLS

The central issue of software engineering lies in the use of automated tools that serve the software engineer by amplifying his capabilities. The software life-cycle can be divided into five phases: Requirements Analysis, Design, Implementation (Programming), Testing (Quality Assurance), and Maintenance. Specialized tools for each area have been found effective in many applications, even while extensive tool-building research and development continues.

Contemporary software engineering tools are exemplified by commercially available tools that capture nearly every essential technical concept in good tool environments. Ranging from single tools that perform one important function (like a source-language instrumentor system) to integrated sets of tools that consolidate a variety of closely related functions, continued software engineering experience dictates the use of good tools — and in some cases the replacement or upgrade of bad tools.

This seminar introduces the concepts of automated tools and how they relate to the software engineering life cycle, based on a state-of-the-art survey of contemporary (commercially or publicly available) software engineering tools. Besides providing an in-depth survey of tools that apply in all five areas, attention is devoted to system production support tools that aid in management of software development projects. Attention is also given to estimating when certain conceptually important tools are expected to be introduced in the market place in the near future.

Attendees receive an extensive set of notes and a copy of the tutorial text Automated Tools for Software Engineering, by Edward Miller. Attendees will gain increased appreciation for good software tool design, an increased understanding of how tools interact, and a good feel for the present state-of-the-art in automated tools.

OUTLINE:PHILOSOPHY OF AUTOMATION

Motivating Forces
General Principles
Overview of Software Engineering Phases
Overview of Tool Role

TOOLS FOR SPECIFICATION/REQUIREMENTS

Analysis Tools
Synthesis Tools
Manual Versus Automated Versus Automatable Methodologies
Contemporary Specifications/Requirements Tools

TOOLS FOR DESIGN

Principles of Design
Modes of Design Assistance
Limitations of Design Assistance
Contemporary Design/Implementation Tools
Interaction Between Tools and the Operating Environment
Recommendations for Purchase/Lease Decisions

TOOLS FOR PROGRAM IMPLEMENTATION

Principles of Programming
Programming Procedures
Debugging Concepts
Contemporary Program Implementation Tools

TOOLS FOR QUALITY ASSURANCE AND TESTING

Principles of Program Testing
Role of Tools in Program Testing
Limitations of Tools Applicable During Testing
Specific Examples of Testing Tools
Recommendations for Purchase/Buy Decision

TOOLS FOR PROGRAM MAINTENANCE

Principles of Software Maintenance
Limitations of Automation for Program Maintenance
Specific Example of Maintenance Tools
Recommendations for Purchase/Buy Decision

THE INSTRUCTOR

EDWARD F. MILLER, JR. is Technical Director of Software Research Associates, San Francisco, California, a firm devoted to advanced computer technology and software applications. His interests include software engineering management, software testing technology, software maintenance technology, automated tool design and computer architecture.

Dr. Miller was previously Director of the Software Technology Center, Science Applications, Inc., San Francisco, and Director of the Program Validation Project at General Research Corporation, Santa Barbara, California. He received a BSEE at Iowa State University in 1962, an M.S. in Applied Mathematics at the University of Colorado in 1964, and the Ph.D. at the University of Maryland in 1968 where he was an Instructor from 1964 to 1968.

Dr. Miller is a member of the IEEE Computer Society, the ACM, SIAM and several honorary societies. He currently serves on several technical committees and is Associate Technical Editor of COMPUTER Magazine.

For further information about this and other Software Technology Seminars please contact the Seminar Manager at Software Research Associates...

(415) 957-1441

or write to...

Software Research Associates
P. O. Box 2432
San Francisco, California 94126

USER INTERFACE DESIGN PSYCHOLOGY

User Interface Design, as a topic in its own right, has recently become the focus of significant design efforts. As the price/performance curve of hardware continues to show a decrease by a factor of 100 each 10 years, increasing emphasis can (in fact, must) be put on supporting user interactions. As a result, there is increased recognition in the computer industry of the essential importance of terms like "ease-of-learning" and "ease of use."

This seminar covers the application of selected information from the psychology of learning and of vision and time perception to the design of user/computer interfaces.

Detailed Case Studies of commercial systems will be presented. Video taped demonstrations of these and some experimental systems will provide an awareness and some evaluation of the multitude of interaction techniques, approaches and devices that are now available.

OUTLINE:INTRODUCTION

Evolution of User I/F Technology
Anatomy of the Seminar
User I/F Dimensions
Information Processing Model
Futuristic User I/F Demo

LEARNING THEORIES

Sequential/Parallel Acquisition
Linguistic/Spatial Materials
Physiological Basis for Thinking Styles

CASE STUDY 1

Graphics Editor Workstation
Structural Model Generation Application
Tablet/Menu Interaction
Goals/Constraints/Rationale

HUMAN MEMORY CHARACTERISTICS

Short-Term/Long-Term Memory
Recall Versus Recognition
Spatial/Linguistic Coding
Role of Information Organization

VISUAL PERCEPTION OVERVIEW

Light/Space/Color/Time Sensitivities
Visual Organization
Display Symbols

CASE STUDY 2

Graphics Editor Workstation
Color Charts and Graphs
Mouse/Menu Interaction
Goals/Constraints/Rationale

STRESS IN USER/COMPUTER INTERACTION

Causes of Stress
What Can Be Done to Reduce
Examples in Computer Systems

INTERACTIVITY AND THE PERCEPTION OF TIME

User's Time Versus the Wall Clock
Two Interaction Models
Case Study of a Database Interaction

CASE STUDY 3 AND 4

Desktop Computer Line Editor Study
Application S/W Study
Operating System Interaction Demonstration

TEXT EDITOR DEMONSTRATIONS

Line/Character/Screen Oriented
Keyboard/Mouse/Tablet Devices
Ease-of-Learning Versus Ease-of-Use
Command Invocation Methods

FUTURE CONSIDERATIONS

Spatial Interfaces
Voice Interfaces
Major Issues in the Field

THE INSTRUCTOR

DR. JACK GRIMES received his Ph.D from Iowa State University in Electrical Engineering and Computer Science, his M.S. in Psychology and is currently a doctoral student in Applied Cognitive Psychology at the University of Oregon. Since 1971, he has been employed at Tektronix, Inc., in Beaverton, Oregon, where he is currently a manager of advanced development for desktop computers.

Dr. Grimes' research interests have recently focused on understanding the nature of user-computer interaction from the user's perspective. Previously, he worked in the areas of computer architecture, silicon technology and programming systems.

Dr. Grimes was a participant in the China Technology Exchange Program in 1979, gave presentations at the Computer Architecture Workshop sponsored by Nixdorf in 1976 in West Germany, and participated in the 2nd USA-Japan Conference held in Tokyo in 1975. Dr. Grimes has previously given a shorter version of this seminar at SIGGRAPH '80 and '81, the Sixth West Coast Computer Faire and internally at Tektronix.

For further information about this and other Software Technology Seminars
please contact the Seminar Manager at Software Research Associates...
(415) 957-1441

or write to...

Software Research Associates
P. O. Box 2432
San Francisco, California 94126

SOFTWARE MAINTENANCE TECHNOLOGY

Software maintenance can often require 50% to 80% of the overall costs associated with a software system's life cycle. Most of the activities of software maintenance involve detailed recordkeeping, incremental change to the software system, and analysis of the impact of changes.

Current technology for software maintenance is in its infancy. Technical methods for analysis of complex and sophisticated computer programs can migrate from the research and development arena into practice only if care is taken in choosing the "right" algorithms and the "appropriate" methods of controlling change. This seminar focuses on methods for handling software maintenance problems that are highly analytical in nature, but which can have immediate practical benefit. Besides investigating various aspects of the maintenance problem, the seminar presents methods of measuring and managing a variety of software maintenance scenarios.

Attendees will receive a comprehensive annotated bibliography of current literature pertaining to software maintenance technology, an extensive set of notes (including case studies of typical maintenance situations), and reprints from the current technical literature.

OUTLINE:INTRODUCTION AND OVERVIEW

- Importance of Maintenance
- Purposes of Maintenance
- Principles of Maintenance

PROBLEMS OF MAINTENANCE

- User Knowledge
- Programmer Effectiveness, Availability
- System Quality
- Machine Requirements
- Environment Reliability

PROGRAMMING ISSUES

- Types of Changes and Related Problems
- Maintenance Scenarios
- Review Procedures, Documentation Methods
- Development Practices to Ease Maintenance Problems

METRICS AND TESTING DURING MAINTENANCE

- Maintenance Metrics
- Functional Testing
- Coverage Testing

SOFTWARE SYSTEM MANAGEMENT TECHNOLOGY

- Configuration Control
- Test Libraries
- Error/Change Tracking

MAINTENANCE AIDS AND TOOLS

- Software Tools
- Methodologies

MANAGEMENT ISSUES

Scheduling for Maintenance
Programmer Motivation
Manpower Management

SUMMARY AND RECOMMENDATIONS

Overall Maintenance Plans
Researchers' Agenda
Bibliography

THE INSTRUCTOR

EDWARD F. MILLER, JR., is Technical Director of Software Research Associates, San Francisco, California, a firm devoted to advanced computer technology and software applications. His interests include software engineering management, software testing technology, software maintenance technology, automated tool design and computer architecture.

Dr. Miller was previously Director of the Software Technology Center, Science Applications, Inc., San Francisco, and Director of the Program Validation Project at General Research Corporation, Santa Barbara, California. He received a BSEE at Iowa State University in 1962, an M.S. in Applied Mathematics at the University of Colorado in 1964, and the Ph.D. at the University of Maryland in 1968 where he was an Instructor from 1964 to 1968.

Dr. Miller is a member of the IEEE Computer Society, the ACM, SIAM and several honorary societies. He currently serves on numerous technical committees and is Technical editor of *COMPUTER Magazine*.

For further information about this and other Software Technology Seminars please contact the Seminar Manager at Software Research Associates...

(415) 957-1441

or write to...

Software Research Associates
P. O. Box 2432
San Francisco, California 94126

Software Research Associates

NAME...

Interactive Test Bed (ITB) for SRTRAN

PURPOSE...

Basic support of software quality assurance through systematic testing, by assisting the user in achieving high values of CI coverage. Assistance is provided by allowing the user to alter global data and analyzing the coverage of subsequent executions. Capability to process Standard SRTRAN programs.

SYNOPSIS...

Basic capability for analyzing coverage results of executions in an interactive fashion. Also provided is ability to alter data to program so as to alter program flow.

Version currently available only for Data General AOS environment.

DESCRIPTION...

A free-standing pre-processor and testing aid for interactive analysis of coverage and execution results of SRTRAN programs and subprograms.

The system consists of a SRTRAN instrumentor, a preprocessor which analyzes the data space of the program, and an interactive program which is linked to the specified test object. The preprocessor automatically generates subroutines which are used by the testbed specifically for the given test object.

Coverage and execution results are reported when the user asks for that information.

SPECIAL FEATURES...

The ITB system automatically generates the code it needs to successfully test the test object. There exist macros which allow the user to set up an ITB in a few instructions.

A trace feature is included which allows the user to follow execution of the test object in a segment by segment trace. This may be turned on or off at will.

Commands entered interactively are automatically stored away so as to give the user a complete record of his session on disk. Also available is the ability to use this 'ghosting' of previous sessions to be the input file to another test bed session.

The entire data space can be saved at any time during a test bed session for the user to re-use later in the same session.

P.O. Box 2432 • San Francisco • California 94126 • Telephone (415) 957-1441 • Telex No. 340-235

DOCUMENTATION...

ITB comes with a Reference Manual.

SRA provides substantial related documentation on Software Quality Assurance and Software Maintenance.

AVAILABILITY...

The ITB system is currently only implemented on a Data General AOS environment.

REQUIREMENTS...

The system requires the presence of a FORTRAN compiler and an SRTRAN preprocessor.

CONTACT...

Mr. Thomas E. Mapp
Member of Technical Staff
Software Research Associates
P. O. Box 2432
San Francisco, CA 94126 USA

(415) 957-1441

Updated: March 1981

NAME...

COBOL Test Coverage Analysis Tool (TCAT/COBOL)

PURPOSE...

TCAT provides basic support of Software Quality Assurance through systematic testing by measuring the C1 and P1 coverage values for series of tests (C1 is the percentage of logical segments exercised and P1 is the percentage of paragraphs exercised).

SYNOPSIS...

TCAT provides a basic free-standing capability for automatic instrumentation of programs to analyze and report C1 and P1 coverage levels. TCAT processes ANS Standard COBOL programs, plus local machine dialect features depending on the system version and host.

Versions of TCAT are available for IBM, Univac, ACOS (Japan only), DEC VAX/VMS, Data General MV/8000, and ONYX C3002 (RM-COBOL/Unix) computer environments.

DESCRIPTION...

TCAT is a free-standing pre-processor/post-processor system for batch oriented analysis of testing effectiveness of COBOL programs.

The COBOL Test Coverage Analysis Tool consists of: (1) a comprehensive COBOL automatic instrumentor, INSTRU, (2) a set of run-time routines that are loaded and executed with the instrumented COBOL programs, called RUNTIME, and, (3) a standardized testing coverage analysis package called COVER.

The pre-processing stage produces a **Reference Listing**, used to identify the logical segments and paragraphs within the candidate COBOL program, and the post-execution stage of TCAT activity produces two forms of output: the **Coverage Report** and the **Not Hit** report. These show the percentage of coverage attained by test(s) expressed in the C1 and P1 measures. In addition, the post-processing system generates a **Histogram Report** that shows the proportion of times each segment and paragraph is executed.

Coverage values attained by tests of the COBOL program are reported on a per-test, per-test-group, or an all-test cumulative basis.

Coverage reporting normally is defaulted to a predefined set of commonly used formats, but can be put completely under user control.

SPECIAL FEATURES...

The TCAT system can handle cumulative multi-run tests by storing standard coverage history records. Special blocking is used to reduce the size of the intermediate trace file. The level of system overhead with this method of intermediate file storage is reasonably low.

The TCAT system can handle multiple entry COBOL source modules as well as COBOL modules with multiple names.

The **Reference Listing** produced by the pre-processor is specially annotated to show complete details of each logical segment in the program. The listing identifies the sense of each logical predicate outcome in the COBOL logic, and provides statistics about the COBOL program that are useful for test module size comparisons and test difficulty estimation.

Other features include run-time settable option settings.

DOCUMENTATION...

TCAT is supplied with a comprehensive **Introduction and User's Guide** plus special installation support information as appropriate.

Software Research Associates provides substantial related documentation on **Software Quality Assurance** and **Software Maintenance** in the form of one-day and two-day **Professional Development Seminars** that can be made available for presentation upon request.

AVAILABILITY...

The COBOL TCAT system is available on a single-user binary license agreement for a variety of computer systems (see above).

Full documentation, installation-dependent information, and subscription-type maintenance and upgrade service is also provided with the basic license agreement. Maintenance and upgrade service after the first year's use is also available.

SYSTEM REQUIREMENTS...

The TCAT system requires the presence of both a COBOL and a FORTRAN compiler. (The post-processing phase of TCAT is implemented in a portable subset of FORTRAN.) In addition, during execution of instrumented programs the TCAT system requires the use of one serial file.

CONTACT...

Christopher Walker
Software Research Associates
P. O. Box 2432
San Francisco, CA 94126 USA

Phone: (415) 957-1441 — Telex: 340-235

NAME...

Extended BASIC Validation Test Suite

PURPOSE...

Validation of BASIC interpreters/compiler which contain extensions similar to those found in the DEC BASIC-PLUS language.

DESCRIPTION...

The Extended BASIC Validation Test Suite is designed to validate the syntactic compatibility of a BASIC interpreter/compiler with the DEC BASIC-PLUS language .

The test suite consists of over 200 test programs from the NBS Minimal BASIC Test Suite plus an additional 150 test programs which test the Extended BASIC language features of DEC BASIC-PLUS. The test programs cover standard capabilities, end cases, and exceptions for the language features.

The extensions to the DEC BASIC-PLUS language include such features as matrix functions, block I/O, control flow statements (WHILE, REPEAT, etc), string functions, and logical operators. All test groups are shown below.

The output from the tests are fully machine processible, thereby facilitating later regression testing.

Software Research Associates can offer either a complete testing service for a client's BASIC interpreter/compiler or the source code only for the Extended BASIC Test Programs.

AVAILABILITY...

The Extended BASIC Validation Test Suite is currently available for DEC BASIC-PLUS compatible implementations of BASIC. A future implementation will be compatible with DG AOS/VS BASIC. SRA can also tailor a system to a client's specific language requirements.

The DEC version of the Extended BASIC Test Suite is priced at \$3200 for a single-user, single-site restricted source license.

CONTACT...

Mr. Mark Opperman
Software Research Associates
P. O. Box 2432
San Francisco, CA 94126

(415) 957-1441

Extended BASIC Validation Test Suite Groups

Group	Language Feature	Number of Programs
1	Simple Printing of string constants	1
2	END and STOP	4
3	PRINTing and simple assignment (LET)	9
4	Control Statements and REM	7
5	Variables	2
6	Numeric Constants, Variables	20
7	FOR-NEXT	12
8	Arrays	29
9	Control Statements	7
10	READ, DATA and RESTORE	15
11	INPUT	7
12	Implementation-supplied Functions	37
13	User-defined Functions	13
14	Numeric Expressions	21
15	Miscellaneous Checks	24
1-15	Minimal BASIC Tests (Subtotal)	208

16	Variables	7
17	Arithmetic Operators	2
18	Logical Operators	6
19	String Operators	6
20	Matrix Operators	7
21	Mathematical Functions	11
22	Print Functions	2
23	String Functions	34
24	System Functions	3
25	Matrix Functions	7
26	Input/Output Functions	4
27	Extended Statements	42
28	Matrix Statements	5
29	Statement Modifiers	7
30	Block I/O Statements	6
31	Miscellaneous Features	4
32	Immediate Mode	1
16-32	Extended BASIC Tests (Subtotal)	154
1-32	Extended BASIC Test Suite (Total)	362

SOFTWARE ENGINEERING AUTOMATED TOOLS INDEX

PROSPECTUS

SOFTWARE ENGINEERING AUTOMATED TOOLS INDEX -- PROSPECTUS

TN-875/1

November 1981

© Copyright 1981 by Software Research Associates

ALL RIGHTS RESERVED. No part of this document may be reproduced in any form, by photostat, microfilm, retrieval system, or by any other means now known or hereafter invented, without written permission from Software Research Associates.

Software Research Associates
P. O. Box 2432
San Francisco, CA 94126 USA

Phone: (415) 957-1441 -- Telex: 340-235

Software Research Associates

San Francisco, California

The Software Engineering Automated Tools Index ("TOOLS INDEX") describes some 600 automated tools that are available from commercial, governmental, industrial, and other sources in the United States and elsewhere in the world. All tools are categorized and cross-referenced in detail.

1.0 CONTENTS

Following is the structural contents of the TOOLS INDEX:

Table of Contents

- 1.0 Introduction
 - 1.1 Organization of TOOLS INDEX
 - 1.2 Contents of Tools Data Frames
 - 1.3 Cross-Reference Listings
 - 1.4 Updates and Corrections
 - 1.5 Sources of Information
- 2.0 Tool Categories Listing
- 3.0 Tool Name Cross-Reference Listing
- 4.0 Tool Category Cross-Reference Listing
- 5.0 Tool Supplier Cross-Reference Listing
- 6.0 Supplier Address Listing
- 7.0 SOFTWARE ENGINEERING AUTOMATED TOOLS INDEX DATA FRAMES (A-Z)
- 8.0 References and Bibliography

2.0 AUTOMATED TOOL CATEGORIES

The TOOLS INDEX is categorized based on each Tool's role in the software life cycle. The Tools are classified according to a scheme that provides a special "category number" for each major class of Tool.

Following are the major categories used by the TOOLS INDEX (Reference attached detailed listing - "Automated Tool Categories"):

- Requirement/Specification Tools
- Software Design Tools
- Software Implementation Tools
- Software Testing Tools
- Software Maintenance Tools
- Software Project Management Tools
- Language and Language Processing Systems
- Utility Packages
- Miscellaneous Support Tools
- Research and Development Systems (Future Prototypes)

3.0 AUTOMATED TOOL CROSS-REFERENCE LISTINGS

The TOOLS INDEX provides a series of cross-reference listings to assist in locating specific tool data.

3.1 Tool Name Listing

Contains a three-field columnized description:

<u>Tool Name</u>	<u>Category Number</u>	<u>Supplier Name</u>
------------------	------------------------	----------------------

Listing is alphabetical by Tool Name.

3.2 Tool Category Listing

Contains a three-field columnized description:

<u>Category Number</u>	<u>Tool Name</u>	<u>Supplier Name</u>
------------------------	------------------	----------------------

Listing is in numeric sequence by Category Number.

3.3 Tool Supplier Listing

Contains a three-field columnized description:

<u>Supplier Name</u>	<u>Tool Name</u>	<u>Category Number</u>
----------------------	------------------	------------------------

Listing is alphabetical by Supplier Name.

3.4 Tool Supplier Address Listing

Is an alphabetical listing, by Supplier Name, with addresses and telephone numbers.

4.0 AUTOMATED TOOL DATA

Tools are described on single "Frames" and organized alphabetically by Tool name. (Reference attached complete Frame, Figure 4.1, and actual sample, Figure 4-2.)

The "Frame" contains a set of fields that describe various features of a particular Tool:

FIGURE 4-1: Contents of Automated Tool "Frame"Name

Short name of tool (phrase describing tool use).

Category

Tool's numeric category (determined from "Automated Tools Categories" listing - assigned by SRA).

Description

Short (one paragraph) description of what the tool is and what the tool does.

Number of Installations

Number of Installations.

Cost

The cost for the system (including all options and variations).

Configuration

The configuration on which the tool operates.

Contact

Company name and mailing address to contact about this tool.

Telephone

Telephone number of person to contact about this tool.

Notes

Special notes about the technical capabilities and features of this particular tool.

References

Any technical references that describe how this tool operates, its effectiveness, or its application (using standard bibliographic citation format).

Source

The source of the information in the above (may be altered by SRA).

Updated

SRA date of latest revision/update of this block of information.

FIGURE 4-2
Sample Completed Tool "Frame"

NAME...

SRTRAN 1 (Baseline)

CATEGORY...

3.4 (Structured Programming Preprocessors)

DESCRIPTION...

Structured Programming Preprocessor for FORTRAN systems.

NUMBER OF INSTALLATIONS...

Approximately 15.

COST...

\$750 for perpetual single-user binary license.

CONFIGURATION...

Portable to most FORTRAN environments. SRTRAN has been successfully installed on IBM, Univac, Data General, DEC, and CDC computer systems.

CONTACT...

Software Research Associates
P. O. Box 2432
San Francisco, CA 94126

PHONE...

(415) 957-1441

NOTES...

This is SRA's own structured programming preprocessor. This "baseline" system includes the standard set of Structured Programming constructs such as IF...ELSE...ELSE IF...END IF, CASE OF...CASE...CASE ELSE...END CASE, WHILE...END WHILE, REPEAT...END, etc. In addition, SRTRAN produces automatically indented, annotated listings of the source programs it processes.

SRTRAN is documented in an extensive User's Manual.

UPDATED...

1 October 1981

5.0 TOOLS INDEX UPDATES/CORRECTIONS

The TOOLS INDEX updates/corrections/deletions will be forwarded to subscribers on a quarterly basis. SRA is continually modifying its computerized TOOLS INDEX files in order to reflect the most current information available.

6.0 SUBSCRIPTION RATES

The TOOLS INDEX, Volumes I & II, will be available January 1982. An Order Form is enclosed. Subscriptions for quarterly TOOLS INDEX updates will be available on a subscription basis only at the rates quoted below.

<u>TOOLS INDEX</u> 2-Volume Set		<u>QUARTERLY UPDATES</u>	
U.S.A./Canada	\$185.00	U.S.A./Canada	\$ 85.00/Yr.
Foreign	\$225.00	Foreign	\$115.00/Yr.

IMPORTANT INFORMATION

U.S.A./Canada orders shipped 4th class book rate. Overseas mail shipped via sea mail (10-12 week delivery).

For priority shipping to U.S.A./Canada, or airmail service (2 week delivery) to foreign countries, please add the following charges:

Tools Index:	U.S.A./Canada	\$10.00/Set
	Foreign	\$50.00/Set
Subscription	U.S.A./Canada	\$10.00/Order/Yr.
Updates	Foreign	\$25.00/Order/Yr.

Tools Index price and quarterly subscription rates are subject to change without notice.

Foreign checks must be in U.S. Dollars drawn on a U.S. bank.

5.1 Computerized TOOLS INDEX

Computer readable versions of the TOOLS INDEX are available on special request.

For further information or ordering details, please contact:

Ms. Terryl Ostmo
Software Research Associates
P.O. Box 2432
San Francisco, California 94126

Telephone: (415) 957-1441 -- Telex: 340-235

LIST OF REFERENCES

1. Kenehan, M.F. Software Product Assurance in the Defense Systems Group - An Overview, presentation at internal TRW meeting, December 1981.
2. American Federation of Information Processing Societies, Inc. National Computer Conference Proceedings, Software Quality by K. F. Fisher, p. 755, December, 1980.
3. Fleet Material Support Office Report 9213. FMSO Quality Program Task Group Report by L.C. Harp, pp. 2-3, 8 February 1982.
4. Whittaker, J.B. "Introduction to FMSO", The Navy Supply Corps Newsletter, V. 45, p. 3, January 1982.
5. Kenehan, p. 3.
6. Fleet Material Support Office, CDA Management Handbook, pp. 2-4, 1 January 1981.
7. Ibid.
8. Kenehan, p. 4.
9. Boehm, B. W. "Software Engineering", IEEE Transactions on Computers, V. C-25, pp. 1226-1228, December 1976.
10. Macabe, J. Software Quality Assurance: A Survey, Macabe Press, 1978, p. 9.
11. Ibid., p. 8.
12. Ibid., pp. 12-15.
13. Dynamics Research Corporation, Military Standard for Software Engineering Software Quality Measurement by G. A. Bourdon, pp. 2/15-17, 1 June 1981.
14. Boehm, B. W. and Others, Characteristics of Software Quality, p. 28, North-Holland Publishing Company, 1978.
15. Ibid., pp. 28-29.

16. Stamm, S. L. "Assuring Quality Quality Assurance", Datamation, V. 27, p. 195, March 1981.
17. Ibid.
18. CDA Management Handbook, p. 2/1.
19. Ibid. pp. 2/2-3.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2. Defense Logistics Studies Information Exchange U.S. Army Logistics Management Center Fort Lee, Virginia 23801	1
3. Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
4. Department Chairman, Code 54 Department of Administrative Science Naval Postgraduate School Monterey, California 93940	1
5. Prof N. G. Lyons, Code 54Lb Department of Administrative Science Naval Postgraduate School Monterey, California 93940	1
6. Cdr. R. A. Bobulinski Newbury Avonlake, Ohio 31629	6
7. Richard L. Patrick Amdahl Corporation 1250 East Arques Avenue P.O. Box 470 Sunnyvale, California 94086	2
8. Joan B. Birkhead Jordan Quadrangle, Pine 164 Stanford University Stanford, California 94305	1
9. Barron A. McDonald IBM Corporation 555 Bailey Ave. P.O. Box 50020 San Jose, California 95150	2
10. Edward Miller Software Research Associates P.O. Box 2432 San Francisco, California 94125	2
11. Lt. Joseph H. Harrington Jr. 2597 Lee Roy Drive Virginia Beach, Virginia 23456	1
12. Professor J. Creighton, Code 54CF Naval Postgraduate School Monterey, California 93940	1
13. Capt. Charles Smith Navy Fleet Material Support Office Code 92 Box 2010 Mechanicsburg, Pa. 17055	6

14. Mr. Raymond Spear
Data System Division - Hewlett Packard
11000 Wolfe Rd.
Cupertino, Ca. 95014

1

LMEI
-8