AD-A042 776 DEFENSE SYSTEMS MANAGEMENT COLL FORT BELVOIR VA F/G 9/2
EMBEDDED COMPUTER SYSTEM SOFTWARE MANAGEMENT AND SUPPORT AFTER --ETC(U)
MAY 77 A E PATTERSON

UNCLASSIFIED NL

1 OF 1

ADA042 776



AD-A042 776

1

DEFENSE SYSTEMS MANAGEMENT COLLEGE



94

PROGRAM MANAGEMENT COURSE
INDIVIDUAL STUDY PROGRAM

EMBEDDED COMPUTER SYSTEM SOFTWARE
MANAGEMENT AND SUPPORT
AFTER SYSTEM DEPLOYMENT

STUDY PROJECT REPORT
PMC 77-1

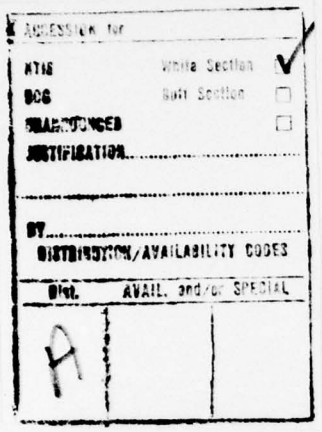
ALTON E. PATTERSON
GS-13 DAFC

D D C
RECEIVED
AUG 12 1977
D

FORT BELVOIR, VIRGINIA 22060

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

EMBEDDED COMPUTER SYSTEM SOFTWARE MANAGEMENT AND SUPPORT AFTER SYSTEM DEPLOYMENT



Study Project Report

Individual Study Program

Defense Systems Management College

Program Management Course

Class 77-1

by

Alton E. Patterson
GS-13 DAFC

May 11, 1977

Study Project Advisor
Lt Col Arcieri

This study project report represents the views, conclusions, and recommendations of the author and does not necessarily reflect the official opinion of the Defense Systems Management College or the Department of Defense.

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* EMBEDDED COMPUTER SYSTEM SOFTWARE MANAGEMENT AND SUPPORT AFTER SYSTEM DEPLOYMENT		5. TYPE OF REPORT & PERIOD COVERED Study Project Report 77-1 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) ALTON E. PATTERSON		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS DEFENSE SYSTEMS MANAGEMENT COLLEGE FT. BELVOIR, VA 22060		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS DEFENSE SYSTEMS MANAGEMENT COLLEGE FT. BELVOIR, VA 22060		12. REPORT DATE 77-1 13. NUMBER OF PAGES 94
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)*

UNLIMITED

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

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

18. SUPPLEMENTARY NOTES

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

SEE ATTACHED SHEET

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

SEE ATTACHED SHEET

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

DEFENSE SYSTEMS MANAGEMENT COLLEGE

STUDY TITLE: EMBEDDED COMPUTER SYSTEM SOFTWARE MANAGEMENT AND SUPPORT AFTER SYSTEM DEPLOYMENT

STUDY PROJECT GOALS:

1. To identify embedded computer system software support problems and address significant areas of support.

2. To present and discuss a management and support approach for embedded computer system software which has been implemented for aircraft computer software systems.

STUDY REPORT ABSTRACT:

This report addresses embedded computer system software management and support after system deployment. It covers the essential aspects of planning a support capability and discusses maintenance and modification requirements, the change process, software configuration management, testing, support organization considerations, and support resources to include personnel, equipment, software, documentation and facilities.

SUBJECT DESCRIPTORS:

COMPUTER SOFTWARE
CONFIGURATION MANAGEMENT
EMBEDDED COMPUTER SYSTEMS
F-111 AIRCRAFT OPERATIONAL FLIGHT PROGRAMS
MANAGEMENT ENGINEERING
MANAGEMENT METHODS
MANAGEMENT PLANNING
SOFTWARE
TEST & EVALUATION

NAME, RANK, SERVICE	CLASS	DATE
Alton E. Patterson, GS-13, DAFC	PMC 77-1	11 May 1977

EXECUTIVE SUMMARY

Over the past decade there have been an ever-increasing number of embedded computer systems (ECS's) integral to DoD systems. ECS's are characterized as highly accurate, reliable and programmable devices which attribute greatly to system performance, capability and flexibility. The programmable feature of an ECS is derived through its software computer programs. This software, because of the relative ease with which it can be changed, provides tremendous potential for maintaining system performance and capability current with continuing changes in requirements.

ECS software can be changed to correct errors and deficiencies, add new capabilities and enhancements, and compensate for changes and/or degradation in system equipment. These types of change requirements continue throughout the life of the weapon system.

This report addresses a systematic approach for implementing ECS software changes after system deployment.

It concludes that software provides tremendous flexibility in responding to system problems and requirements over the life of the system provided an efficient, effective and responsive ECS management and support system is established which (1) is planned and fully coordinated with the weapon system integrated logistics support plan; (2) includes a staff of highly qualified and experienced personnel and a maintenance and modification laboratory equipped with general purpose computing equipment, dynamic simulation, data acquisition and system equipment mock-ups; and (3) employs a systematic method

for developing changes which includes strong program management, configuration management, test management and system engineering.

The report recommends (1) that greater emphasis be placed on educating management on the requirements and benefits of ECS software support; and (2) that stronger and more substantive management policies and support be provided in acquiring resources, in particular personnel, and in implementing ECS software support plans.

TABLE OF CONTENTS

		Page
EXECUTIVE SUMMARY		ii
LIST OF FIGURES		1
SECTION		
I.	INTRODUCTION	2
	Problem	2
	Purpose	3
	Background	3
II.	PLANNING	8
	Support Requirements	9
	Alternatives	11
	Procedures	13
III.	CHANGE PROCESS	16
	Software Changes	16
	Change Sequence	19
	Block Change Concept	22
	Change Development Cycle	23
IV.	CONFIGURATION MANAGEMENT	28
	Configuration Identification	29
	Configuration Control	32
	Configuration Status Accounting	37
	Automated Documentation Generation	39
V.	TESTING	43
	Test Objectives	47
	Test Planning and Procedures	49
	Test Methods	50
VI.	RESOURCES	55
	Personnel	58
	Equipment	58
	Software	59
	Documentation	60
	Facilities	63

TABLE OF CONTENTS CONT'D.

		Page
VII.	SUPPORT ORGANIZATION	65
VIII.	SUMMARY, CONCLUSIONS AND RECOMMENDATIONS	68
	Summary	68
	Conclusions	68
	Recommendations	69
APPENDIX A:	Computer Program Change Request Form	70
APPENDIX B:	Summary Descriptions of Resources	71
	Personnel	72
	Equipment	76
	Software	80
BIBLIOGRAPHY		87

LIST OF FIGURES

Figure		Page
I-1	F-111 MODIFICATIONS	5
II-1	PRE PMRT MILESTONES	15
III-1	OFP CHANGE "WIND VECTOR FIX"	17
III-2	MECHANIZATION/INTEGRATION "WIND VECTOR FIX"	18
III-3	ECS SOFTWARE CHANGE SEQUENCE	20
III-4	ECS SOFTWARE CHANGE DEVELOPMENT CYCLE	24
IV-1	ECS SOFTWARE CHANGE CONTROL	34
IV-2	AUTOMATED DOCUMENTATION GENERATION, F-111 OFP's	41
V-1	F-111 AVIONICS INTEGRATION SUPPORT FACILITY	44
V-2	SOFTWARE TEST	45
V-3	COMPUTER SOFTWARE RELIABILITY	46
V-4	ECS SOFTWARE VERIFICATION/VALIDATION/CERTIFICATION MODEL	53
V-5	ECS SOFTWARE TESTING	54
VII-1	ECS SOFTWARE SUPPORT ORGANIZATION	66
B-1	F-111 OFP DYNAMIC SIMULATOR	78
B-2	F-111 OFP's	81
B-3	SUPPORT SOFTWARE SYSTEM F-14 WEAPON CONTROL COMPUTER PROGRAM	86

SECTION I

INTRODUCTION

PROBLEM:

Embedded Computer System (ECS)[1] software is receiving high level Department of Defense (DoD) interest and attention because of the many problems experienced and high dollar expenditures incurred during their development and acquisition. (8:2-6)[2] To help alleviate these problems, new directives and regulations have been promulgated which establish policy and offer guidance in this area, e.g., Department of Defense Directive 5000.29, Management of Computer Resources in Major Defense Systems, dated 26 April 1976 and Air Force Regulation 800-14 Volume I, Management of Computer Resources in Systems, and Volume II, Acquisition and Support Procedures for Computer Resources in Systems, dated 26 September 1975. (1), (7) As a result, better planned and executed software development programs are in effect, as evidenced by the Air Force F-16, B-1 and AWACS systems.

However, a grey area still remains in regards to efficient and effective management and support of ECS software after deployment. Greater interest, attention, guidance and visibility must be focused on this phase of the life cycle if a viable solution is to be achieved for each of the ECS software systems entering the DoD inventory.

1. "An ECS is integral to an electronic or electromechanical system (for example, combat weapon system, tactical system, aircraft, ship, missile, spacecraft, command, control and communication systems) from a design, procurement and operation viewpoint". (8:3)

2. This notation is used throughout the report for sources of quotations and references. The first number is the source listed in the Bibliography. The second is the page(s) in the reference.

PURPOSE:

The purpose of this report is to provide visibility and offer guidance for the management and support of deployed ECS software. Attention is focused on planning, changes and modifications[1], configuration management, testing, support resources, and organizational structure as they apply to software.

To accomplish this, the author has researched the limited documentation covering the area, and has drawn heavily on his seven years of experience in pioneering ECS software management and support while working for the Air Force Logistics Command on the F-111 aircraft system.

BACKGROUND:

Over the past decade, there have been an ever-increasing number of ECS's integral to DoD systems. ECS's are characterized as programmable devices exhibiting high speed, accuracy and reliability in performing computations, making logical decisions, establishing priorities, selecting alternatives and exercising control. These features have contributed to the achievement of DoD systems with greater performance, capability and flexibility.

---

1. The term "modification" as used in this report means software changes which are visible to the weapon system, i.e., effect performance, capability, operation, function, mode, etc., while "change" is used to mean software changes which correct deficiencies, optimize computer memory and timing, change logic, coding, etc. Modification is also used synonymously with software block change[2]. There is a fine line between change and modification and at times they are used interchangeably.

2. A software block change is a collection of software changes (i.e., changes with no hardware impacts) which are concurrently developed and integrated into the baseline computer program as a unit change.

The programmable feature of most ECS's is derived through its software computer programs. The software provides a tremendous potential for maintaining weapon system[1] performance and capability current with the continuing changes in requirements. This concurrency has far exceeded the capability of hardware changes in terms of responsiveness and cost, as illustrated by Figure I-1. The data is from the F-111 program and shows cost and time for implementing comparable capabilities, (additional off-set aim points and updated weapon ballistics) through hardware on F-111 A/E aircraft and through software on the F-111 D/F aircraft. The differences are quite dramatic and had the software changes not been part of a normal software annual update, i.e., block change, the differences would have been even more dramatic, as these software changes could be implemented in a matter of weeks.

The fundamental difference between software and hardware that permits software modifications to be implemented much faster and cheaper is that software does not go through a production phase and requires no modification kits. With the exception of documentation, the cost and time for a software change is primarily consumed in developing and testing the prototype which when complete can be immediately sent to the field for operational use.

Changes and modifications that can be implemented through software cover a wide range, from correcting a deficiency, to adding a new weapon

1. The term "weapon system" is used in this report for clarity in lieu of "DoD system" or "system". It is fully recognized that ECS applies equally to other types of DoD systems, as previously defined.

HARDWARE VS SOFTWARE MODIFICATIONS (EQUIVALENT CAPABILITY)

CAPABILITY	HARDWARE	SOFTWARE
ADD 2 OAP[1]	MOAP[2] ECP F-111A/E $5.278M/42 MO	4 OAP CHANGE F-11 $100K/16 MO
BALLISTICS UPDATE (10 WEAPONS)	ECP 2884 F-111A/E $1.05M/36 MO	BALLISTICS UPDATE D-17 $20K/10 MO
BALLISTICS UPDATE (10 WEAPONS)	FLEXIBLE BCU[3] F-111A/E $8M/78 MO	BALLISTICS UPDATE OFP[4] $20K/ 15 MO

1. OFF-SET AIM POINTS
2. MULTI OFF-SET AIM POINTS
3. BALLISTICS COMPUTER UNIT
4. OPERATIONAL FLIGHT PROGRAM

F-111 MODIFICATIONS

Figure I-1

system capability. There are a continuing number of requirements that dictate software changes during the operational phase of the weapon system life cycle. These requirements can be categorized as:

- Addition and/or change in system capability
- Deletion, addition and/or change in operational modes
- Changes in operational functions
- Changes in weapons and/or weapon ballistics
- Changes and/or replacement of system equipment
- Addition of new equipment
- Correction of deficiencies
- Corrections to programming problems
- Corrections to operations and/or operational procedures
- Corrections to equipment problems
- Compensation for equipment degradation
- Optimization of computer memory utilization
- Optimization of computer time utilization

After weapon system deployment, it is the responsibility of the DoD supporting agency to see that responsive action is taken to investigate ECS software change requirements and to implement those that are valid. Each Service is taking steps to provide the required support. The trend is to develop organic capabilities, i.e., DoD in-house efforts, to perform this maintenance[1] function. This trend is particularly evident for aircraft software where organic support has been implemented for the F-111,

1. For the purpose of this report, maintenance includes all changes, modifications, restructuring or recoding of the ECS software for whatever reason, and is used synonymously with ECS software support. It also includes other ancillary and attendant tasks performed on the software.

A-7 and F-14 aircraft. The Air Force has also planned organic software support for the F-4, F-15, F-16, B-1, AWACS and for most new electronic and communication systems entering the inventory. Results of software support studies on these systems have shown that organic software support is (1) the most efficient, effective and responsive to user requirements, (2) provides the best guarantee of uninterrupted support over the life cycle of the weapon system, and (3) provides DoD with the best means of positive control over system performance and capabilities.

SECTION II

PLANNING

Planning for ECS software management and support must start early in weapon system acquisition development, where it is closely coordinated and integrated with other weapon system logistic and support plans, and continues throughout the weapon system life cycle. This Section addresses the planning phase essential for determining and implementing efficient and cost effective ECS software management and support approaches. It looks at ECS software support requirements and alternatives, and reviews formal planning procedures.

The first step in planning a support strategy is to prepare an explicit problem statement which includes requirements; viable alternatives for satisfying requirements (including criteria for evaluation); constraints; and assumptions. For ECS software, support requirements and alternatives are generally functions of a generic class of software, e.g., aircraft operational, shipboard control, electronic warfare, ground communication, command and control, etc., and in some cases they are functions of a particular ECS software system. Each generic class of software has its own peculiar characteristics, hardware/software interfaces, system configuration, deployment environment, user/supporter relationships, and policies, practices and procedures governing management and support. Each of these must be analyzed and investigated closely in determining support requirements and in determining the best alternative for satisfying these requirements. Support alternatives are also dependent on, and constrained by, the availability and location of certain essential support resources. These resources are weapon system equipment, (i.e., the ECS computers,

and the sensors, displays, controls, etc., which interface with the ECS computers); and ECS engineers.

Weapon system equipment is essential for software validation, i.e., system integration, system performance tests, compatibility tests, etc. It is normally required on a dedicated basis and is configured into a system test bed. The location and availability of this equipment is dependent on the type of weapon system, its cost, deployment environment, the number of systems procurred, and its maintenance concept, e.g., on-site, depot, contract, etc. Site alternatives for ECS software support, i.e., maintenance, will typically be constrained to the locations where this equipment is available.

The ECS engineers provide the technical expertise required for total system (hardware/software) performance, integration, compatibility and configuration and are typically contractors and/or part of the weapon system support agency engineering staff. Their availability tends to be limited to particular locations and this can have an impact on and/or constrain the choice of the support location and/or support agency.

SUPPORT REQUIREMENTS:

ECS software support requirements can be divided into two categories: (1) ECS software change and modification requirements, and (2) routine software maintenance requirements.

Software change and modification requirements are determined in terms of change frequency and change response. The frequency is the expected number of change requirements per unit of time (month, year, etc.) and the response is the expected time for transforming requirements into released changes. Determining frequency and response is not simple and can

widely vary with the category of changes (reference Section I). However, if the changes are divided into two classes: (1) software only changes (no system hardware impacts) and (2) software changes with system hardware impacts or caused by hardware changes, then an average mean frequency and average mean response for each of these classes can be determined with reasonable confidence.

Routine software maintenance requirements are determined in terms of the amount of activity associated with software optimization, problem investigation, studies, performance analysis and response to day-to-day "what if situations".

The aggregate of these requirements will normally dictate a "sustained level of effort support[1] over the operational phase of the system life cycle. This has proven to be the case for support of the FB-111A, F-111D/F and A-7D/E aircraft Operational Flight Programs (OFP's)[2]. In the case of the F-111 OFP's, the software change activity averages about 27 changes per OFP update, i.e., block change. These updates run continuously for each aircraft configuration and take 15 - 18 months for development and implementation. This, coupled with the routine maintenance support requirements, has resulted in approximately a 60 man year sustained level of effort.

---

1. "Sustained level of effort support" means a constant staff of personnel (Government, contractors, or mix) employed for an indefinite period of time (generally life of weapon system) to perform software maintenance (exclusive of major modifications having hardware impacts).

2. OFP's are the software resident in the aircraft avionics computers which integrate the on-board sensors, control systems, and displays; perform computations and exercise control; and provide operator cues for automatic and/or crew aided navigation, bombing and fire control.

ALTERNATIVES:

Alternatives for ECS software management and support generally consist of some combination of arrangements between the weapon system support agency, user and contractor with the maintenance site being either the depot, a user site, or contractors facility. For example, one combination might be a situation where the support agency has management responsibility, the contractor performs the maintenance and the maintenance facility is located at the user site. Alternatives are normally constrained, to some degree, by Service and Command regulations and policies delineating management and support responsibilities. Key criteria used for evaluating alternatives consist of cost, availability of resources, support effectiveness, ECS software control and risk associated with a continuous support posture. An attempt should be made to rank the criteria in order of importance. However, this is difficult since the criteria are not independent. For example, cost is highly dependent on availability of resources.

Very little latitude exists for evaluating alternatives for assigning management responsibility to ECS software. Generally, it is assigned to the organization having management responsibility for the weapon system and/or subsystem containing the ECS. For example, in the Air Force, aircraft OFP's are managed by the aircraft system manager, and the avionics subsystem ECS software is managed by the respective subsystem manager.

There is much more latitude for evaluating alternatives for the assignment of responsibility for ECS software support, i.e., maintenance. However, there are still certain constraints. For example, maintenance for command and control software and software relating to mission requirements is traditionally assigned to the user. The evaluation of alternatives

for ECS software maintenance is one of the more difficult and emotional planning tasks. With the increasing number of ECS's entering the inventory, software maintenance is now looked upon as a very attractive and prestigious workload with everyone wanting a piece of the action. The evaluation outcome tends to be a function of who performs it. Therefore, an evaluation performed by a team made up of all involved agencies generally produces a result with the greatest credence. It must be emphasized that each system will have different circumstances and must be evaluated accordingly.

Site selection for a location to perform software maintenance is normally constrained to sites which have weapon system equipment available that can be used in a system test bed configuration. This constraint is weapon system dependent. For example, a one-of-a-kind system like the FPS-85 satellite tracking radar is only available in a system configuration at the user site. This makes that site the only practical choice, as the site for software maintenance. In this situation, software maintenance requirements must be integrated with operational requirements and use. The alternative is to develop software changes at a remote location and bring them to the user site for validation and system test. However, this has not proven cost effective because of the iterative nature of software development and test.

In contrast, depots for fighter aircraft have proven to be the best site location for OFP software maintenance support. The system and system equipment are more readily available at the depot, along with the required technical expertise and other support resources. Also, if the aircraft are deployed at more than one site or used by more than one Command or Service, the depot becomes the logical central support point. Further,

fighter aircraft are configured in such a manner that field software maintenance would be extremely difficult. The equipment is densely packaged, making on-board maintenance impossible, and because of the field maintenance concept for the electronics equipment, other resources such as a laboratory equipment mock-up are not available in the field.

PROCEDURES:

Formal procedures such as AFR 800-14 have been instituted to assist personnel in the difficult task of planning ECS software support. AFR 800-14, Volume II, requires that a computer resources integrated support plan (CRISP) be prepared for each ECS software system entering the inventory. Early in system full scale engineering development, a computer resources working group (CRWG) is established for the purpose of preparing the CRISP and overseeing its implementation, and has responsibility for determining ECS software support requirements and alternatives, and for coordinating the transition of the ECS software from the developing to the supporting and using agencies. The CRWG is made up of personnel from the developing, supporting, training, and using agencies and also has representation from the developing contractors. (1:3-4,5)

The CRISP is the key planning document for ECS software management and support. It is prepared early in full scale engineering development, becomes part of the overall weapon system integrated logistics support plan (ILSP), and remains updated throughout the system life cycle. The CRISP covers all management and support aspects essential to the maintenance and modification of ECS software after deployment, including requirements for transfer and turnover of the ECS software from the developing to the supporting and using agencies. (1:3-4,5)

Based on expected ECS software support requirements and the selected course of action for implementation, the CRISP details and schedules all activities, functions, and resources required for development, implementation and operation of an ECS software management and support capability. Included in the CRISP are: (1) organizational responsibilities, relationships, and interfaces; (2) site location for maintenance; (3) configuration management plan; (4) modification and change plan; (5) test plan; (6) funding plan; (7) hardware/software integration plan; and (8) the resource allocation plan, which includes requirements and sources for personnel and training, facilities, laboratory and weapon system equipment, computers, software, and documentation. (1:3-4,5), (23)

The CRISP also integrates the ECS software transition with the weapon system program management responsibility transfer (PMRT) by closely interfacing with the PMRT plan. This plan establishes a time phased schedule of actions and events necessary for accomplishing an orderly and timely transfer of weapon system program management responsibility from the developing agency to the supporting agency on a specific date. Figure II-1 shows the pre PMRT milestones as they relate to the CRISP. (23)

In summary, early, thorough and fully coordinated planning should result in the selection of the most cost effective and efficient approach for the management and support of ECS software and should result in a smooth transition of the ECS software from the developing to the supporting and using agencies with a support capability in place and operational at the time of transfer.



△ SM ASSIGNMENT

△RFP PUBLICATION

△SOURCE SELECTION EVALUATION

△COMPUTER RESOURCE WORKING GROUP (CRWG) ESTABLISHED

△COMPUTER RESOURCE INTEGRATED SUPPORT PLAN (CRISP) PUBLISHED

△CRISP INTEGRATED INTO PROGRAM MANAGEMENT PLAN (PMP)
INTEGRATED LOGISTICS SUPPORT PLAN (ILSP)
PROGRAM MANAGEMENT RESPONSIBILITY TRANSFER (PMRT) PLAN

△DSARC 3

△PMRT PLAN IMPLEMENTATION

△FIRST SYSTEM IOC

△PMRTD

PRE PMRT MILESTONES

Figure II-1 (23:15-8)

SECTION III

CHANGE PROCESS

Software provides the weapon system with tremendous flexibility because of the relative ease with which it can be changed. However, the success in achieving weapon system flexibility through software changes is highly dependent on the ability to make timely, efficient, and effective changes. This requires an orderly, systematic change process, combined with considerable resource commitments. This section addresses the technical change process. It examines a typical ECS software change and discusses the change sequence, block change concept and development cycle. Subsequent sections expand on configuration management, testing, resources and the support organization.

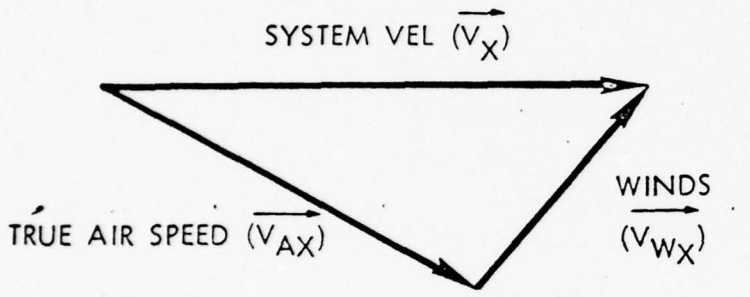
SOFTWARE CHANGES:

As noted in Section I, changes to ECS software are required for a number of reasons and will continue over the life of the weapon system. As the weapon system matures, the reduced number of changes required to correct errors and deficiencies is off-set by the increased number of changes required to implement new capabilities, enhancements, etc. Most changes are highly complex, as depicted in Figure III-1 and III-2. Illustrated is the "Wind Vector Fix" change which has been implemented in F-111 OFP's, and can be considered a typical ECS software change. Figure III-1 summarizes the change requirement in terms of operational enhancements, i.e., greater navigation and weapon delivery accuracy in the degraded operational mode and easier operator use. Figure III-2 illustrates change mechanization and avionics functional interfaces

PURPOSE

- IMPROVE PROBABILITY OF MISSION SUCCESS IN DEGRADED MODE
- BACK UP MEANS TO CALCULATE MORE ACCURATELY KEY WEAPON DELIVERY AND NAVIGATION PARAMETERS WHEN INS IS DOWN
  - SYSTEM VELOCITY
  - WINDS
- EASIER OPERATIONAL USE
  - RADAR CURSOR LAYOUT
  - DESIGNATE

GEOMETRY



$$\overrightarrow{V_X} = \overrightarrow{V_{AX}} + \overrightarrow{V_{WX}}$$

RADAR DISPLAY

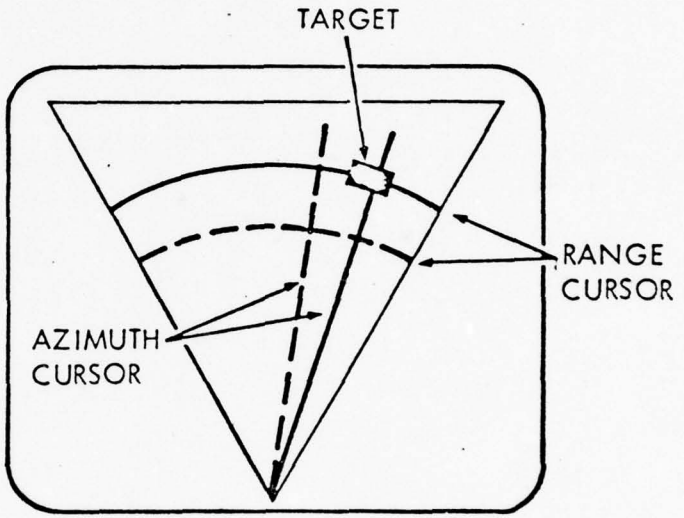
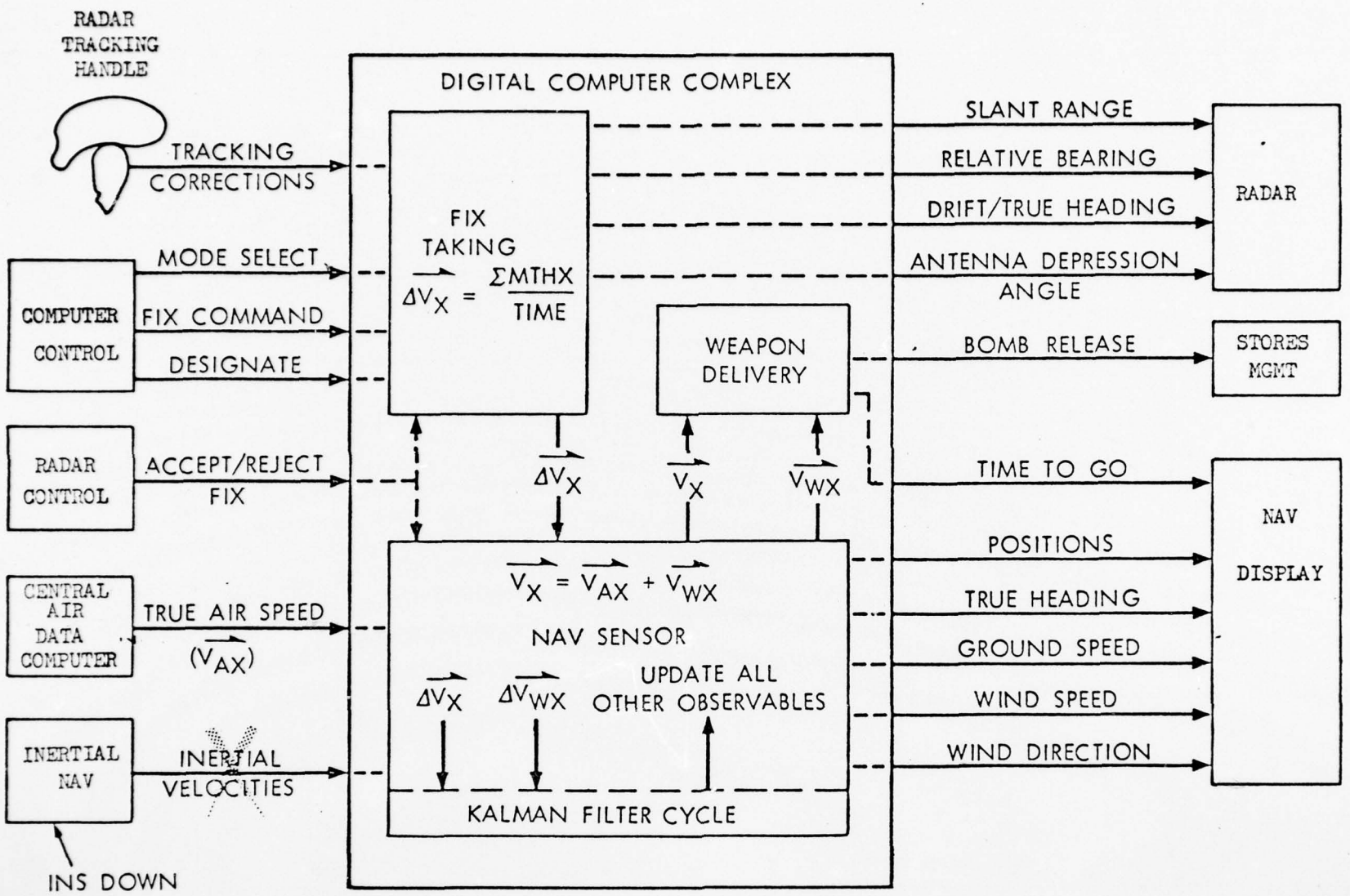


Figure III-1 (17:14-4) OFP CHANGE "WIND VECTOR FIX"



MECHANIZATION/INTEGRATION "WIND VECTOR FIX"

Figure III-2 (17:14-4)

which have been translated from the change requirement.

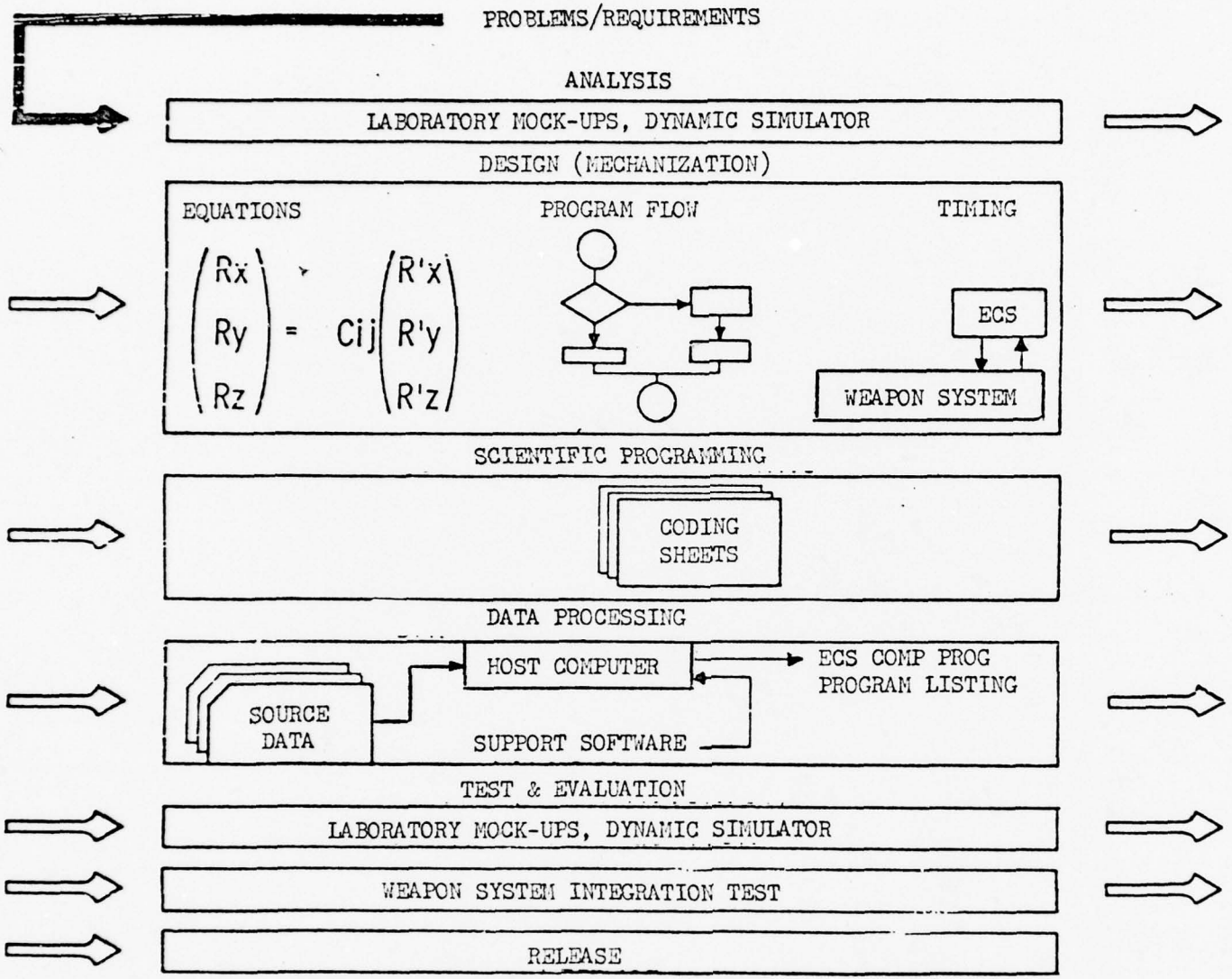
To effectively make changes of this type requires personnel who have intimate knowledge of the weapon system, computer interfaces, avionics, and ECS software. Further, they must have adequate tools to analyze, process and test software changes. These resources are discussed in detail in Section VI and Appendix B. (17)

CHANGE SEQUENCE:

The software change sequence is a process of analyzing, designing, programming, processing, debugging, integrating, testing, evaluating, and implementing changes to ECS computer programs. This process is illustrated in Figure III-3 and although not shown, is highly iterative. It begins with the analysis of the change requirement to determine validity, feasibility, risk and scope. The analysis includes investigation of:

- Operational, performance, functional and interface requirements
- Computer time, core and word length requirements
- Interfaces and weapon system impacts
- Test provisions and data
- Documentation requirements
- Integration and acceptance testing
- Time, cost and resource requirements
- Alternatives
- Trade-off considerations

This analysis is usually performed in the laboratory, using a dynamic simulator and system mock-up (reference Section VI and Appendix B). Outputs of this phase are a change requirements document and statement



ECS SOFTWARE CHANGE SEQUENCE

Figure III-3

of work. Requirements are specified in terms of operational performance and capability.

During the design phase, operational requirements are translated into interface, performance, and functional requirements. These requirements are then translated into a detailed design which reflects program structure, logic, timing, inputs, outputs, equations, and algorithms. The output of this phase is a design specification which reflects design requirements, detailed design, and acceptance and test criteria.

During the programming phase, the detailed design is translated into a source computer program (symbolic code). This translation is performed, taking into account computer timing, memory, word length, accuracy, and subroutine interfaces.

The data processing phase uses a host computer to assemble or compile the source program into an object program, i.e., load program for the ECS computer. In addition, documentation, such as program listings and memory maps are generated. The output of this phase is the ECS computer program.

The debug, integration, test and evaluation phases are performed both in the laboratory and in an operational environment. Computer program debugging is performed using a host computer, dynamic simulator and diagnostic software. Integration, test and evaluation are performed using a dynamic simulator, system mock-up and operational weapon system.

The final phase releases the ECS computer program for field implementation. This is accomplished after the changes have been documented and accepted by the user. (3), (13), (17)

BLOCK CHANGE CONCEPT:

At the rate which ECS software changes are requested, it would be highly impractical to attempt implementation on an individual basis. Most support systems could not be responsive or afford the cost. Major problems would be created in allocating and scheduling resources, controlling configuration and maintaining documentation. Because of these problems, the Sacramento Air Logistics Center created the "block change concept" for processing changes to F-111 OFP's. As stated in Section I, a block change is a collection of software changes, i.e., changes with no hardware impacts, which are concurrently developed and integrated into the baseline computer program as a unit change. Block changes are normally made on a periodic basis with the cycle time negotiated by the user and supporting agency. It depends on change activity, change response requirements and available support resources. For example, block changes are made to F-111 OFP's every 12 to 18 months.

Change requests are analyzed as received, with feasible changes accumulated as block change candidates. Emergency changes are expedited by processing on an individual basis. Block change candidates are reviewed on a scheduled basis by the user and support agency for the purpose of prioritizing and establishing the block change definition. This process is discussed in detail in Section IV under Configuration Control.

The block change concept enhances the efficiency, effectiveness and responsiveness of the change process because changes can be collectively developed, tested and documented. The major pay-off is in the documentation and test areas. (14), (17)

CHANGE DEVELOPMENT CYCLE:

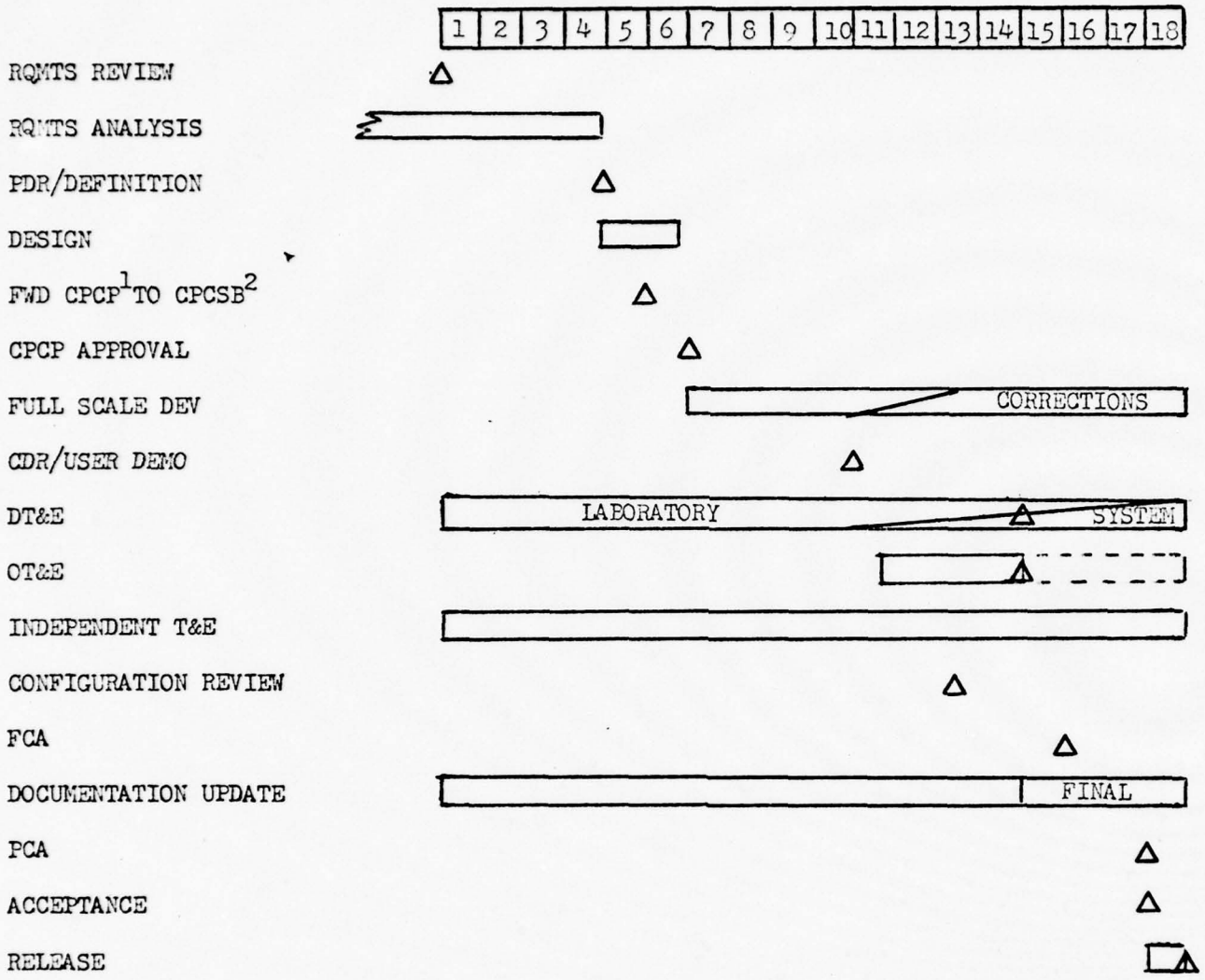
The ECS software change development cycle, as illustrated in Figure III-4 can be viewed as a microcosm of the acquisition development cycle. It provides the basis for an orderly, systematic and well controlled software update program. It consists of a series of phases, each well defined and separated by significant milestones. The cycle was established for development of block changes to OFP's, most notably F-111 OFP's, but is applicable to other ECS software.

As shown, the cycle is designed to allow ample time for testing and documenting changes. These are major time consumers and start at the beginning and continue throughout the cycle with the last third dedicated almost exclusively to these efforts. (Only a relative time scale is shown.) The cycle is also designed to provide maximum interface and communication between the user and developer.

The cycle begins with a joint user, developer meeting to review and prioritize change requirements. Results of the analysis performed on previously received change requests (reference Section IV and Figure IV-1) are reviewed to determine which changes should be included in the block change definition. Trade-offs are made between user priorities, and the scope of each change in order to define a block change which can be completed with available support resources and within the development cycle time. The meeting is concluded with a block change definition.

The review is followed by a continuation of the requirement analysis phase. During this phase, operational requirements are translated into design requirements and a preliminary detailed design. This is followed by a preliminary design review (PDR) and block change definition meeting.

ECS SOFTWARE CHANGE DEVELOPMENT CYCLE



1 Computer program Change Proposal
2 Computer Program Configuration Sub-Board

Figure III-4

Here, the user reviews the requirements document and adequacy of design. The design is reviewed to assure that it will accomplish intended results, that risk has been minimized and that it can be completed on time and with allocated resources. The outputs of this review are the block change requirements document, preliminary design and agreement to proceed into detailed design. Also, during this phase the test plan and documentation requirements are initiated. (Reference Sections IV and V.)

During the design phase, detailed design is completed and the computer program change proposal (CPCP) is prepared and submitted to the computer program configuration sub-board (CPCSB) for approval. (Change Control is covered in Section IV under Configuration Control.) The CPCP describes in detail the block change, total weapon system impact, and cost and schedule for implementation. CPCSB approval constitutes formal "go ahead" to develop the block change and implement, subject to final acceptance of the computer program by the user.

Development commences with approval of the CPCP. Here the cycle deviates slightly from the acquisition development cycle in that a critical design review (CDR) is not held with the user prior to proceeding into design implementation. During development, the block change is coded and assembled into a new engineering baseline program. Each program change and the final assembled engineering program goes through extensive debugging, system integration and test in the laboratory. Also during this phase, the test plan and formal test procedures are finalized and documentation is concurrently updated with change development and closely monitored by configuration management.

At completion of initial development, a CDR and user demonstration is held. During this review, the design is examined for completeness and each change is demonstrated to the user to verify that it performs as expected in a laboratory environment. Also during this review, design specifications, test plans and test procedures are reviewed for adequacy, completeness and acceptability.

Agreement between the user and developer at the CDR that development is complete constitutes "go ahead" into final test and evaluation. During this phase, development testing continues in the laboratory and the developer integrates and conducts developmental tests on an operational system. For example, OFP's are tested using instrumented flight test aircraft. Initial testing of the ECS software integrated with an operational system assures that no major problems or unsafe conditions exist. Once this is accomplished, the ECS software is certified as worthy for operational testing and is released to the user for test purposes. Developmental and operational testing are then conducted concurrently and problems and anomalies fed back to the laboratory for analysis and correction.

Part way through the test program a configuration review is held. The purpose of this review is to assess test results and freeze configuration. At this point, changes experiencing problems which do not have readily apparent solutions that can be implemented and tested with low risk by the scheduled completion of testing are removed from the block change and deferred to the next update. The idea behind this philosophy is to allow sufficient testing of the final configuration and allow required lead time for final documentation.

At the completion of scheduled testing, results are analyzed, evaluated and documented in the final test report. Results are reviewed and the user makes a determination on final acceptance. If accepted, the program, along with required field documentation, is released to the user for operational implementation through the weapon system release process. During this phase, all documentation and testing are completed. Also, the functional and physical configuration audits are performed. These audits are discussed in Section IV under Configuration Status Accounting. (13), (14), (17)

In summary, a methodology for efficient, effective and responsive ECS software change processing was discussed. The methodology develops changes collectively as a block change using a development cycle tailored for ECS software support.

SECTION IV

CONFIGURATION MANAGEMENT

Configuration management is perhaps the least understood, and most difficult to enforce, of all disciplines associated with the management and support of ECS software, yet without question, it is one of the most important. Many technical personnel, and even managers, because of vague understanding, perceive configuration management as a time consuming over-kill control task which obstructs the software change process. Yet, without it, programs invariably get into trouble. For example, programmers tend to code early without adequate definition or design; designs tend to get implemented without adequate documentation; because of the ease with which software can be changed, unauthorized and unapproved changes tend to get implemented; and because of the intangible nature of software (cannot see or touch it), these changes tend to go undetected until they create major problems. These types of problems, coupled with software complexity, flexibility, and continuous state of change, during the operational phase, make it imperative that all personnel working with ECS software understand and employ sound configuration management principles and practices.

This section attempts to put configuration management into proper perspective by: (1) reviewing configuration management principles and practices; and (2) looking at how they apply and can be tailored to ECS software support. In accordance with DoD Directive 5000.29, ECS software will be managed as configuration items (CI's). A CI is defined as "an aggregation of hardware/software or any of its discrete portions, which satisfies an end use function and is designated by the Government for configuration management". (5:2), (7)

Configuration management provides the management tools and procedures essential for systematic ECS software maintenance and modification. It is defined as "a discipline applying technical and administrative direction and surveillance to: (1) identify and document the functional and physical characteristics of a CI; (2) control changes to those characteristics; and (3) record and report change processing and implementation status". (5:2) This definition identifies the three functions of configuration management:

- configuration identification (for software the significance is on identification of functional characteristics, since software has no physical characteristics)
- configuration control
- configuration status accounting

CONFIGURATION IDENTIFICATION:

Configuration identification is defined as "the current approved or conditionally approved technical documentation for a CI, as set forth in specifications, drawings and associated lists, and documents referenced therein", and configuration is defined as "the functional and/or physical characteristics of hardware/software, as set forth in technical documentation and achieved in a product". (5:2) For the purpose of this report, CI's include the ECS computer programs and elements of the ECS software support system (reference Section VI and Appendix B). From a weapon system point of view, CI's are the end item software, i.e., the ECS computer programs. However, for efficient and effective ECS software support, it is essential that all elements of the ECS software support

system, both hardware[1] and software, also be managed as CI's.

The primary purpose for managing the ECS software and support system as CI's is to assure positive control over their configuration. This is required in order to efficiently and effectively track performance, identify deficiencies, and effect changes. Hence, the documentation must be tailored to provide a clear, accurate, thorough, explicit and understandable definition of the software. The definition must define the computer program requirements, functions, performance, inputs, outputs, interfaces, operational characteristics, and structure. Requirements should be stated in terms of operational needs or deficiencies. Functions, performance, inputs, outputs, interfaces and operational characteristics should reflect the translation of requirements into engineering design; and the computer program structure should reflect the detailed software design and include narratives, timing, memory utilization, logic, equations, algorithms, and code. Documentation must also include the criteria for acceptable performance and methods for test and evaluation. Further, documentation must provide for easy traceability, trackability and accountability. The type of documentation that sets forth this information varies widely. For ECS computer programs, it normally is contained in a number of documents, as summarized in Section VI, or it may be contained in a single document. For example, many of the F-111 OFP support software programs are documented with a single combined functional and implementation specification which is embedded in the computer program code and stored on the

1. The emphasis of this Section is on configuration management as it applies to software. It is assumed that the hardware support elements are managed in accordance with standard configuration management principles as they apply to an engineering laboratory environment.

laboratory host computer system. Discretion must be used in tailoring documentation to the specific application. For example, documentation required for an off-the-shelf compiler which is envisioned to change very infrequently will be less than that required for a data reduction program which is envisioned to change occasionally; and documentation for the data reduction program will be much less than that required for an ECS computer program which is envisioned to change frequently. Normally, documentation will consist of a requirements document, design and product specifications, program listings, object and source code, test documents, version description documents; and control and status accounting records, i.e., configuration indexes, change requests, change proposals, and change logs.

It is extremely important to carefully scrutinize and tailor all documentation to fit the specific computer program application and support requirement. Further, every effort should be made to standardize documentation and to develop formats that are adaptable to automated updating, control and status accounting. Documentation tends to be voluminous, extremely expensive and time consuming to maintain, and is worthless if out of date. The weapon system flexibility provided by the ECS software can be lost through indiscriminate documentation requirements. Voluminous documentation cannot possibly be updated at the pace at which software is capable of being changed, causing one of two things to happen: (1) either the software change pace is slowed down, making the support system non-responsive to the user's needs; or (2) the documentation does not get updated and eventually brings the whole operation to a stop. This can be avoided by using prudent judgment in tailoring documentation to the need.

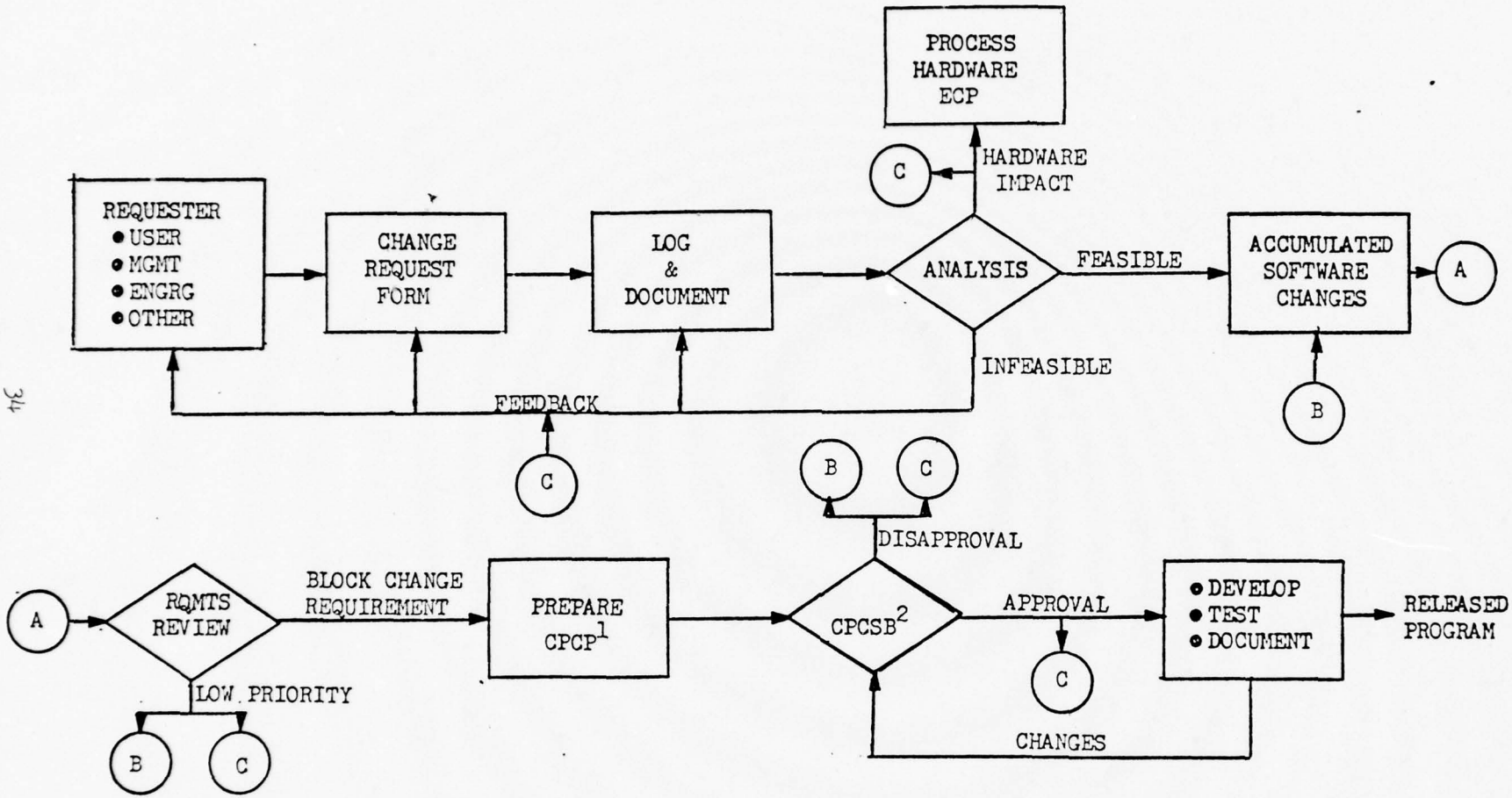
Another important aspect of configuration identification is the number and marking of computer programs and documentation. In order to track computer program baselines and changes, a computer program numbering and marking system must be implemented which is formalized and standardized. (Baseline is defined as "a configuration identification document or a set of such documents formally designated and fixed at a specific time during a CI's life cycle. Baselines, plus approved changes from those baselines, constitute the current configuration identification".) (6:Encl 1,1) The system should identify all computer programs with a distinguishing baseline part number and subsequent changes with version description numbers. Documentation should track with a basic document number and subsequent revision numbers. The system must provide a one-to-one correlation between computer programs and documentation; must provide traceability to the original baseline; and facilitate control and status accounting. (1:6), (5), (6), (10), (12), (17)

CONFIGURATION CONTROL:

Configuration control is defined as "the systematic evaluation, coordination, approval or disapproval, and implementation of all approved changes in the configuration of a CI after formal establishment of its configuration identification". (5:2) Its primary purpose in the ECS software support environment is to control changes to the software. In addition, it controls documentation, controls the software change release, and assures that a systematic development process is followed which results in a released computer program whose configuration is accurately reflected in the documentation. Configuration control is accomplished through a configuration control board (CCB), and a series of reviews and audits.

Figure IV-1 illustrates a software change control process that is suggested for ECS software support. It tracks closely with the guidance promulgated by AFR 800-14 and is similar to the change control process in use for F-111 OFP's and also similar to the one being implemented for F-15, F-16, F-4E, and PAVE TACK OFP's. Basic differences between this process and the standard configuration control process are: (1) a computer program configuration sub-board (CPCSB) subservient to the weapon system CCB approves or disapproves changes; and (2) software changes are not segregated in accordance with the MIL-STD 480 definition of Class I and Class II changes, but are segregated according to whether they are software changes only (no weapon system hardware impacts) and according to whether they can be implemented within the existing sustained level of effort support. All proposed ECS software changes follow this process. An abbreviated form of this process is used for changes to support software.

Change requests are initiated by submitting a Change Request Form of the type shown in Appendix A. This form is routed through appropriate channels and eventually to the organization responsible for ECS software maintenance, where it is logged, documented and analyzed. The requested change can range from a correction to a system anomaly, perceived as a software problem, to the addition of a new capability, perceived as having a software solution. The analysis, as outlined in Section III, investigates all aspects and impacts of the change and scopes the development and implementation effort. Interfaces are one of the most important areas and must be investigated thoroughly, as something inadvertently overlooked can cause serious cost, schedule, and performance problems downstream. For example, a change made to the F-111 OFP's resulted in a compatibility



1 Computer Program Change Proposal

2 Computer Program Configuration Sub-Board

Figure IV-1 ECS SOFTWARE CHANGE CONTROL

problem with the aircraft cockpit panels resulting in a 300 thousand dollar panel mod. The more common interfaces with the ECS computer programs which warrant close scrutiny are:

- User/operator
- ECS computers
- Weapon system equipment
- Trainers and mission simulators
- Weapon system test and support equipment, e.g., automatic test equipment for the computers
- Technical data system
- Documentation
- ECS software support system

The analysis results in one of three decisions: (1) the software change is feasible and has no weapon system hardware impact; (2) the change is infeasible, (e.g., does not provide the results the requestor envisioned, violates weapon system capability or engineering principles, is beyond the computer capability, is beyond the capability of the sustaining level of effort, is excessively risky in terms of cost, schedule and/or performance); and (3) the change impacts weapon system hardware. In this case, the change request is routed to the weapon system manager for processing as a hardware engineering change proposal (ECP) or other appropriate action.

Feasible software changes are accumulated and periodically reviewed by the user and supporting agency to determine whether they should be incorporated into an ECS computer program block change. Those changes which are determined to be of too low a priority to commit resources are deferred. Those which do warrant implementation are defined in a block

change requirements document, and a computer program change proposal (CPCP)[1] is prepared and submitted to the CPCSB, which has the delegated authority to approve or disapprove software changes. The CPCSB is made up of members from both the supporting and using agency and is the formal controlling point for software changes and the integration of those changes with the weapon system. They review the validity of requirements and review proposed solutions to assure that they are feasible, adequately defined, do not represent high risk, are compatible with the weapon system and can be accomplished within existing resources.

Approved changes process through final development, test and documentation which is an iterative process controlled by a series of reviews and audits. Changes that evolve during development and test which impact the approved baseline are processed back through the CPCSB for approval. The final documented ECS computer program, after acceptance by the user, is released for field implementation through a formal and controlled release system. This system assures that the ECS computer program has satisfactorily passed the functional and physical configuration audits and is compatible and complies with the weapon system change release system.

In summary, this change control process: (1) controls the approval and implementation of block changes to ECS computer programs; (2) eliminates software changes from getting bogged down in the full CCB process or from competing with hardware changes for priorities; and (3) permits the change approval/disapproval decision to be made by people who are familiar with the ECS computer programs and support system. This results in a more efficient and responsive software change process. (1:6), (5), (6), (10), (14), (12), (17)

1. A CPCP is an ECP tailored to software application .

Another important function of configuration management is documentation control. This function assures that all documentation used for computer program configuration identification and copies of all computer programs are controlled and stored in a safe and secure area. Usually, this entails establishment of a master library and a working library. The master library retains the master copy, usually on microfilm or microfiche, of all documentation and computer programs. The working library contains copies of current master documentation. These documents are required and used by personnel to support software maintenance and modifications. It is the responsibility of configuration management to: (1) assure that all documentation and computer programs in these libraries are strictly controlled, remain current, and reflect the exact configurations of the ECS software and support systems; (2) control unauthorized changes to documentation and software; and (3) control the use of unauthorized documentation and software in support of the ECC software and support system.

CONFIGURATION STATUS ACCOUNTING:

Configuration Status Accounting is defined as "the recording and reporting of the information that is needed to manage configuration effectively, including a listing of the approved configuration identification, the status of proposed changes to configuration, and implementation status of approved changes". (5:2) Its purpose in the ECS software support environment is to: (1) provide management with feedback information to monitor implementation of software changes; (2) identify configuration management problems in a timely manner; (3) provide historical and current information on the ECS software and support system; (4) monitor and report

on the state of the software and software documentation; and (5) standardize and eliminate redundant documentation. This is accomplished by participation in technical reviews, conducting configuration audits and making informal inspections of software and documentation.

Computer programs and documentation status is maintained in a configuration index and a change status report. The configuration index is a log which provides a running and current status of all computer programs and documentation. The computer program change status report, which is normally a supplement to the configuration index, is a log which provides a historical and current running account on the status of all computer program changes. Normally, there are separate logs for each ECS software configuration and another for the support software. In practice, these logs should be automated to facilitate compiling data. They provide management with up-to-date visibility and status of the software and documentation configuration.

During technical reviews, it is the responsibility of configuration management to: (1) assess the software configuration identification and change control; (2) assure that the computer program design is being adequately documented, in accordance with established configuration management practices; and (3) document and report any deficiencies.

At the end of the software change development cycle, it is the responsibility of configuration management to conduct two audits, the functional configuration audit and the physical configuration audit. These audits must be satisfactorily completed before the software is released for operational use.

The functional configuration audit reviews test results and compares

them against functional performance requirements contained in computer program specifications. It assures that these requirements were used as the criteria against which computer programs were tested and that results reflect that the criteria were satisfied. All discrepancies are reported so that corrective action can be initiated. The test results become a part of the configuration identification and are incorporated into the version description document.

The physical configuration audit reviews the final computer program configuration and compares it against documentation to assure that documentation reflects the as-built computer program. To do this, a listing of the computer program code is generated and documentation is reviewed to assure that it accurately, thoroughly and clearly tracks the computer program from the requirements through the code. Documentation is also reviewed for completeness and conformity to established standards. Discrepancies are reported so that corrective action can be initiated. The documentation becomes the configuration identification for the newly established configuration baseline of the computer program CI. (5), (6), (10), (12), (17)

AUTOMATED DOCUMENTATION GENERATION:

Even with a major effort to tailor software documentation, the amount can still be awesome, requiring considerable resources to maintain, control and update. It is not uncommon to find that 20 - 30 percent of the total time spent in maintaining and modifying software is consumed in documentation tasks. One way to alleviate some of the burden is to automate documentation generation. This, in itself, creates more computer programs and requirements for more documentation, but can have big pay-offs; and might

be the difference between a responsive and non-responsive ECS software support system.

The Sacramento Air Logistics Center, in an effort to keep up with their ECS software change commitment for F-111 OFP's, found it necessary to implement automated documentation generation. The system developed does not generate all documentation, as it is not fully implemented, but greatly reduces the amount of manual tasks that were previously required. Essentially, the system, operating on the laboratory host computer, sorts, manipulates and generates all data pertinent to ECS software change requirements. This data is compiled and used to generate the block change requirements document, design specification, CPCP, test plan and procedures, change narratives, status and code. During development, this data is automatically updated and correlated with the generation of the block change development, engineering and release baselines. This process is illustrated in Figure IV-2. (17:14-8,9)

Requirements for automated documentation should be stated, and spelled out, in the planning document, e.g., CRISP. This will enable the acquisition contractor to develop documentation that is computer generated. It is very difficult, time consuming, and expensive to convert a manual system to an automated system, as is the case for the F-111 programs.

In summary, the benefits derived from sound software configuration management are:

- Well defined and documented ECS software and software support system
- Efficient, effective, and responsive change control

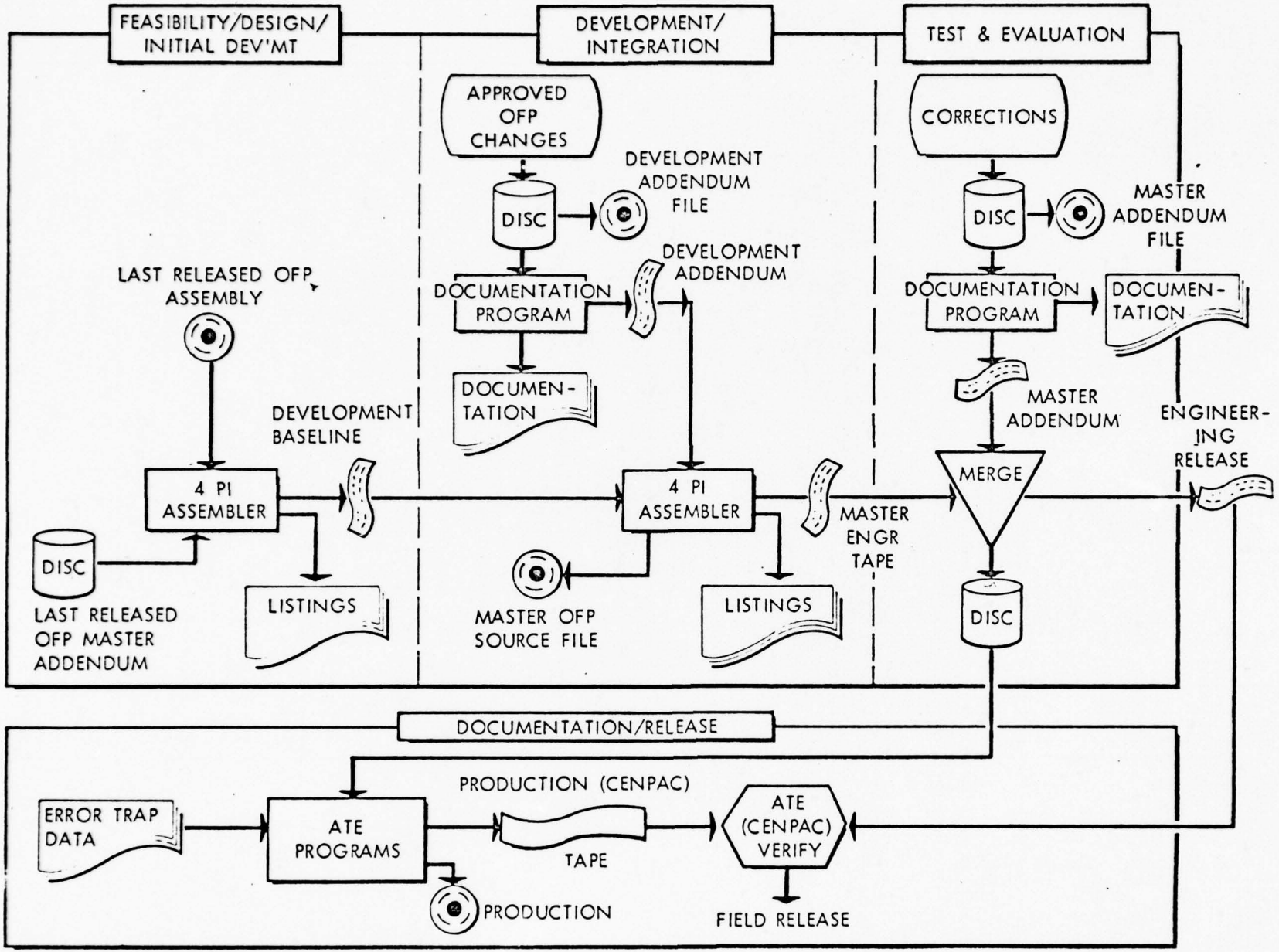


Figure IV-2 (17:14-9) AUTOMATED DOCUMENTATION GENERATION, F-111 OFP's

- Total system traceability
- Software delivered in accordance with users requirements
- Delivered software accurately and completely described by standard documentation
- Documentation tailored to the specific need
- Total software visibility

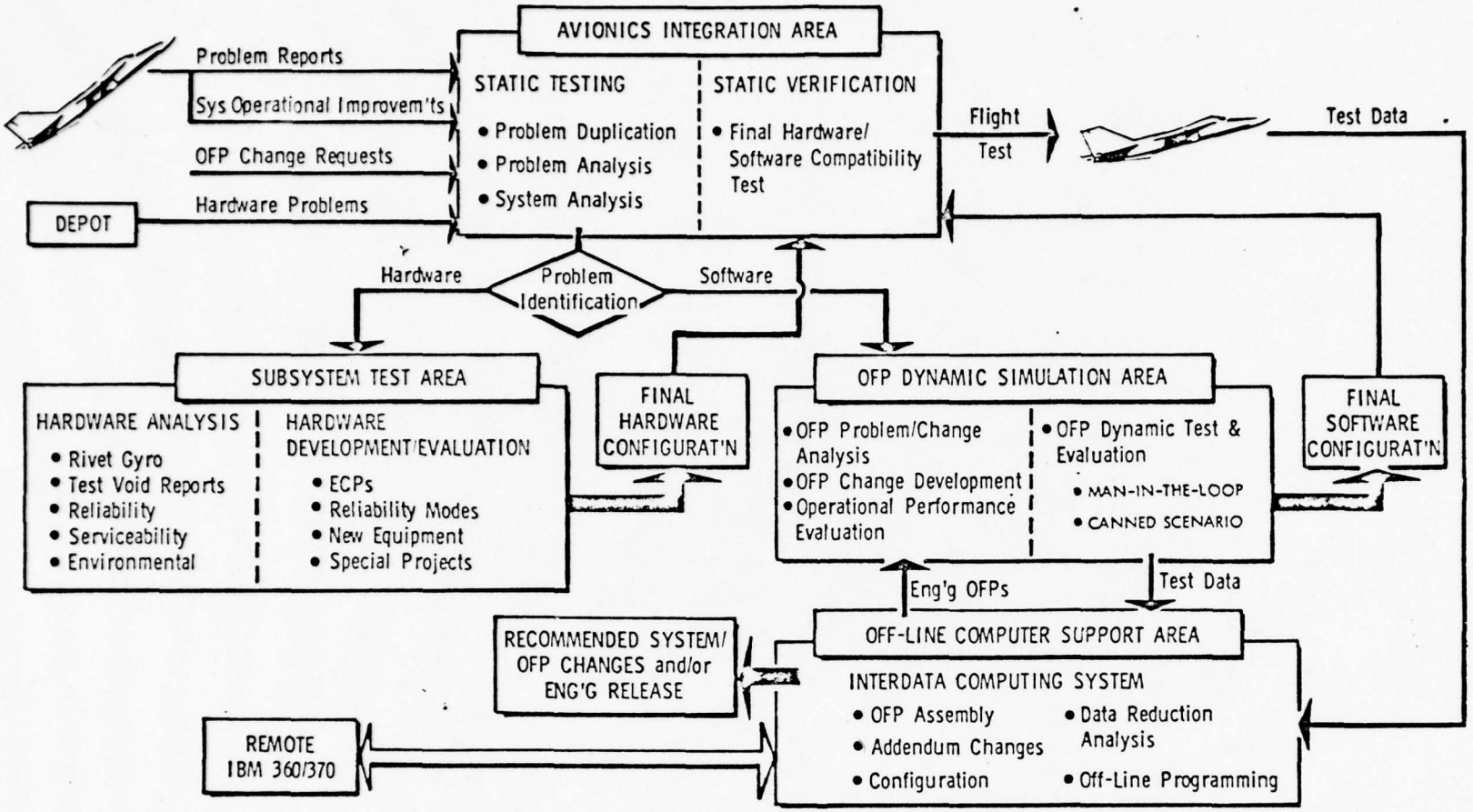
SECTION V

TESTING

Testing is perhaps the most important aspect of the software development process, as it ultimately demonstrates and verifies the success or failure of the development program. Testing is, however, expensive. Historical data compiled on several major development programs indicates that 45-50 percent of development efforts are expended in some form of testing. (2:52) In addition to the testing effort, major costs are also invested in test resources. For example, to provide on-going engineering support for the F-111 OFP's, and for avionics system integration, a 20 million dollar investment in laboratory resources has been made, and two F-111 aircraft have been permanently assigned to this effort. The overriding requirement for these resources is OFP testing. These resources are illustrated in Figure V-1. (15)

Figure V-2 illustrates the challenge of software testing. It can be concluded from this figure that it is literally impossible to prove, through testing or by any other means, that even the simplest computer program or program change is completely free of error. In fact, the Dijkstra maxim states "Testing shows the presence, not the absence of bugs". (3:7) Testing, therefore, finds errors and the more testing that is accomplished, the higher the probability that the software is free of errors and satisfies performance, i.e., the greater the software reliability[1]. This is illustrated in Figure V-3. However, prudent judgment must be exercised in determining the amount of testing by comparing cost expended to benefit gained.

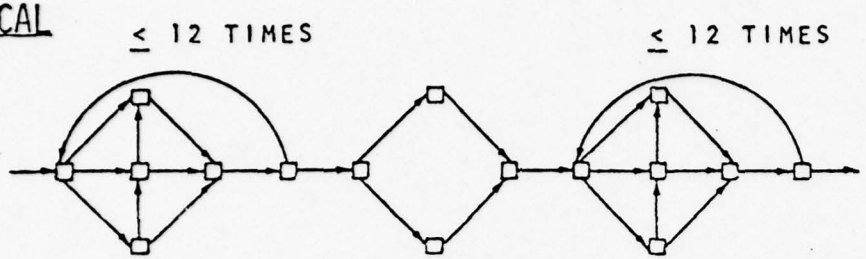
1. Software reliability is generally stated as the probability that a computer program will perform its intended function for a stated period of time in a specified environment.



F-111 AVIONICS INTEGRATION SUPPORT FACILITY

Figure V-1 (15:245)

LOGICAL



- 10^{20} DIFFERENT PATHS
- 4000 YEARS @ 1 LOGICAL PATH PER NANOSECOND

TITAN III MISSILE AND GUIDANCE S/W (6)

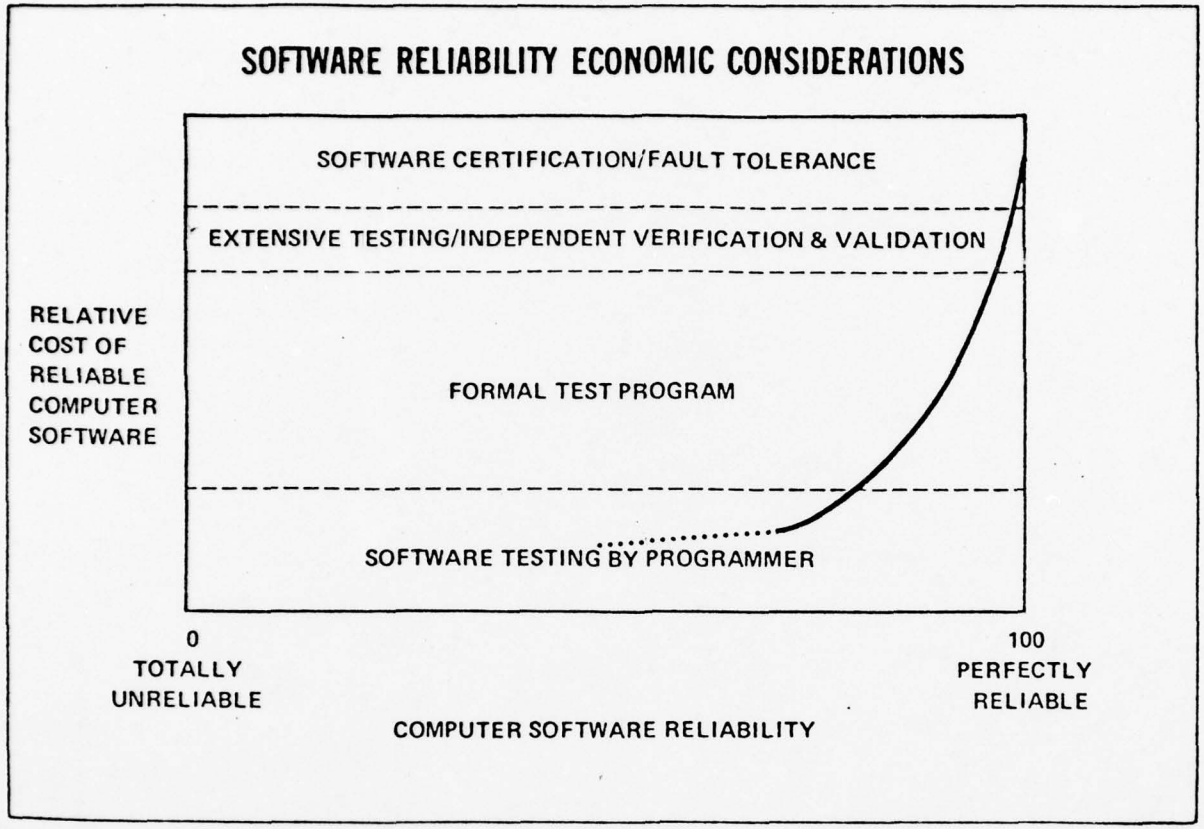
- 60,000 YEARS TO CHECK ALL POSSIBLE PATHS

POINT:

- TESTING IS MOST EXPENSIVE PART OF DEVELOPMENT
- TESTING IS AT BEST "SELECTIVE"

SOFTWARE TEST

Figure V-2 (10:1-8)



COMPUTER SOFTWARE RELIABILITY

Figure V-3 (11:17)

In spite of this rather bleak picture, software testing can be cost effective in producing software with a high degree of confidence in its reliability. To accomplish this requires detailed planning, explicit definition of test objectives and a systematic approach in implementing objectives. This section addresses test objectives, planning, procedures, and methods as they apply to ECS software support.

TEST OBJECTIVES:

Testing can be separated into two categories, development test and evaluation (DT&E) and operational test and evaluation (OT&E).

DT&E, as it applies to ECS software changes, starts during the initial analysis of change requirements and continues throughout the development cycle. DT&E is accomplished by the agency having responsibility for developing the ECS software changes and is monitored by the user. DT&E objectives are: (1) to evaluate the technology, design and engineering of software changes; (2) to demonstrate that risks to the weapon system, if any, have been eliminated; (3) to demonstrate that software changes meet the specification requirements; and (4) to demonstrate that the development process is complete. (4)

OT&E starts when the software changes are turned over to the user for test and evaluation and is completed prior to the user acceptance. For example, in the case of F-111 OFP's, OT&E commences shortly after the start of DT&E flight testing and is conducted in parallel with it, as shown in Section III, Figure 4. OT&E is performed by the user and its objective is to demonstrate that the software changes perform their intended functions in a user mission environment. (4)

Since it is more effective to test software for the presence of errors,

then test objectives should be directed at the most probable error sources. Lt. Col. Manley, in his article "Embedded Computer System Software Reliability" (11:14) summarizes a number of these error sources. They are divided into two general categories, design and coding errors and externally caused failures.

Design and coding errors are those errors caused by personnel during development. They include:

"Syntax Errors: The syntax of a computer language is the precise definition of words, statements and combinations thereof that a compiler (or language translator) will accept. Syntax errors usually result from incorrectly matching programming languages with compilers. Fortunately, most syntax errors can be detected by good compilers and correction is not difficult.

Semantics Problems: In translating user requirements into programming languages and programming languages into machine code, misinterpretations can occur. This is one of the more difficult sources of software errors to detect and correct.

Logic Errors: Caused during design and coding, these errors result in computer program logic deficiencies, such as the creation of continuous loops and impossible states that end in deadlocks.

Algorithmic Errors: Many software failures are caused by mathematical errors such as incorrect scaling of variables. Iterative rounding errors can reduce the precision of variables below acceptable limits. Division by zero, multiplication of excessively large numbers, and similar mathematical operations can cause overflow problems and subsequent incorrect computer outputs". (11:14)

Externally caused failures are external environment influences which

can cause the software to fail even though it is logically correct. They include:

"Computer Hardware Failures: Certain failure modes of the computer itself are a source of software malfunctions not easily traceable to the hardware. Examples are: imperfections on magnetic surfaces, random power fluctuations and the malfunction of single-core elements, or flip-flops.

Interactions With Other System Components: Software logic that requires continuous interaction with other system components, such as servo-mechanisms, inertial navigators, and radar sets, can fail due to the transient malfunctions in the electromechanical components. Since software operates on well-defined computer data inputs, modifications to those inputs can cause abnormal logical operations.

Incorrect Human Inputs: Human inputs through button pushing or keying can generate erroneous computer data inputs". (11:14)

Environmental Changes: An embedded computer system can be subjected to temperature, pressure, humidity and electro-magnetic radiation conditions which can cause hardware malfunctions and subsequent software failures.

Interface Incompatibilities: Changes and/or degradation in hardware performance can result in changes in computer interface timing and signal characteristics resulting in compatibility problems which can cause software failures.

TEST PLANNING AND PROCEDURES:

The best of intentions generally never get implemented without adequate planning--so goes it with software testing. Managers typically look

very optimistically at test objectives and tend to gloss over required resources for implementation, resulting in gross underestimates of the test program. The solution to this lies in early detailed planning.

The software test plan must be prepared early in change development and is updated as more detailed information becomes available. It describes in detail test objectives, course of action for accomplishing objectives, and defines criteria and requirements against which the software will be tested to demonstrate that it satisfies objectives. The plan describes the software to be tested and defines responsible test agencies and their roles; test locations; test environment; schedule; limitations and/or constraints; data acquisition, reduction and analysis requirements; test evaluation criteria; and resource requirements, such as facilities, laboratory and weapon system equipment, support software and personnel. (20)

Test procedures implement objectives and requirements of the test plan. They are prepared for each level of test; describe the unit under test; outline the method and control of test; prescribe methods for recording, evaluating and reporting test results; and specify test resources.

The test report documents and presents test results, conclusions, and recommendations. It summarizes test results in terms of satisfying test objectives; and discusses test anomalies and problems. It, along with data recorded in test procedures, provides the basis for the functional configuration audit and acceptance and release of software changes for operational implementation. (21)

TEST METHODS:

The most widely used test method for ECS software is what is commonly

termed "verification, validation and certification". Essentially, it is a systematic methodology employed to assure that each phase of the development process is sequentially tested in accordance with specified procedures and against stated criteria. It is an iterative, regressive process which starts by verifying ECS software change requirements and is completed with certification of the software in a mission environment.

Verification, validation and certification are either implemented by the organization developing the change or by an independent group or agency. The advantage of using an independent source is an unbiased assessment. Further, they are more prone to look for failure modes in lieu of verifying acceptable performance. The main disadvantage is resources. There obviously is a certain amount of redundancy which generates a requirement for additional people and a queuing of resources. The personnel requirement can be appreciable since these people must be on board during the complete development cycle. For example, to implement independent verification and validation for the F-111 OFP's is requiring approximately a 15 percent increase in support personnel. (18)

Verification is iterative and assures that each step in the development process is correct. In a step-by-step sequence through each phase of development, it checks documentation and tests outputs against input criteria. It assures that system requirements have been properly translated into ECS software requirements and that ECS software specifications have been correctly derived from these requirements. It assures that the program is free from coding or structural errors and that the algorithms and equations used to implement requirements are operationally correct and produce expected results. Verification is normally done in the laboratory

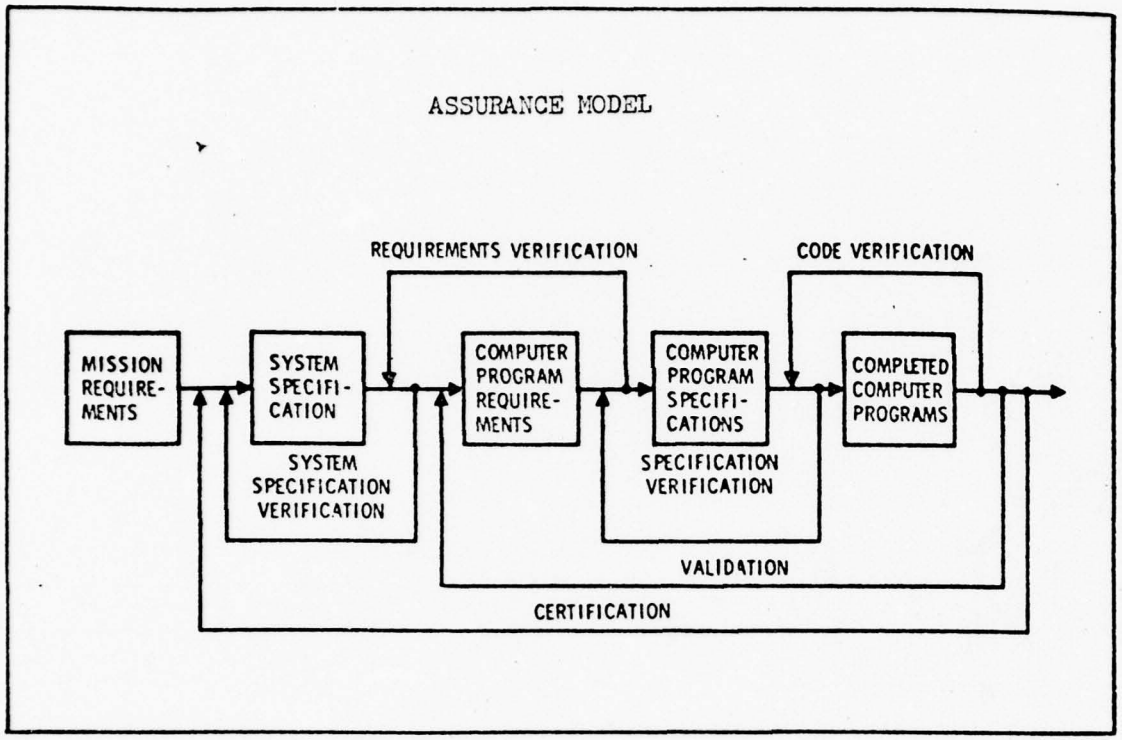
using the host computer system, dynamic simulator and verification software tools. (These resources are discussed in Section VI and Appendix B.) (9)

Validation tests the ECS computer program while performing in the ECS computer. It validates performance against the system specification. For example, in the case of F-111 OFP's, tests are conducted in the F-111 Avionics Integration Support Facility, using the dynamic simulator and system mock-ups. On the dynamic simulator, output parameters such as range to target, destination coordinates and present position are verified against simulated input scenarios; and on the mock-ups, system interfaces, compatibility, signal characteristics, and timing are validated. (9)

Certification tests the ECS computer program in its actual mission environment, assuring that mission requirements, system effectiveness, operational availability, dependability and capability are satisfied. Certification is always performed with actual mission equipment which is modified only to the extent necessary for data acquisition. For example, aircraft OFP's are always certified using operational aircraft modified only to accommodate instrumentation and data acquisition systems. (9)

Figure V-4 illustrates the verification, validation, certification process, and Figure V-5 correlates the process with the software change development cycle and test resources.

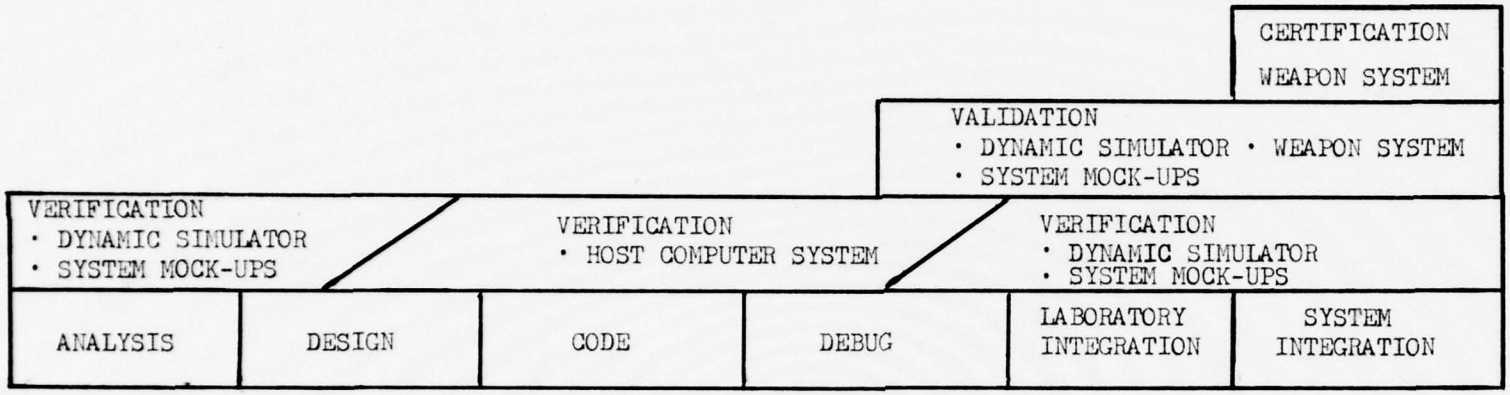
In summary, this Section has highlighted the need for effective testing of ECS software changes and presented a systematic test approach. This approach stressed early and thorough planning, explicit definition of test objectives, detailed test procedures and reports and the verification, validation, certification method of implementation.



ECS SOFTWARE

VERIFICATION/VALIDATION/CERTIFICATION MODEL

Figure V-4 (9:11-6)



ECS SOFTWARE TESTING

Figure V-5

SECTION VI

RESOURCES

As noted in previous sections, the task of supporting ECS software is extensive with software modifications viewed as a continuous series of microcosms of the acquisition development cycle. This section summarizes resources required to provide efficient, effective and responsive ECS software support; and addresses considerations that should be given and trade-offs that should be made in their selection. Resources are expensive and require considerable lead time for acquisition, therefore, discretion must be used in establishing resource requirements and these requirements must be known early. Resources fall into five major categories; personnel, equipment, software, documentation, and facilities. Each of these categories are briefly discussed in this section, with more detailed information provided in Appendix B.

Responsibility for providing resources and determining quantities, time required and eventual location should be spelled out in the planning document. In the case of the Air Force, this document is the CRISP (reference Section II). It should spell out the support approach and extensively detail resource requirements to include funding, acquisition and implementation, allowing sufficient lead time so that all resources are in place and operational at the time of system responsibility transfer from the developing to the supporting agency.

As mentioned, resources are expensive and every effort should be expended to minimize the cost. To this end, several significant things can be done.

First, and perhaps most important, is to finalize the decision on support requirements, agency, and site early in acquisition development. If this is done, the information can be very valuable in directing the contractors acquisition development so that the ECS software and attendant supporting resources are designed with maximum maintainability and transferability in mind. Software can be developed using high order language (HOL), structured programming, modular design and top down programming techniques resulting in easier maintenance and fewer support personnel requirements. Engineering and programming documentation used by the contractor can be prepared in such a way that it is directly transferable to the supporting agency. Support software can be developed so that it is compatible and directly transferable to supporting agencies host computers. Finally, and most important, system integration mock-ups, dynamic simulators, special test equipment, instrumentation, and data acquisition systems used for acquisition development can be designed and structured so that they are easily relocatable to the supporting site after system deployment. This early planning can go a long way in negating the requirement for two sets of resources, one for acquisition development and one for maintenance, i.e., on-going ECS software support. The Air Force, for F-16 OFP's, has worked out this type of an arrangement between the developers (Air Force Systems Command and General Dynamics), and the supporting agency (Ogden Air Logistics Center). One of the problems with this technique is the void in support that can be created during transition.

The transition problem can be completely eliminated by developing the software from inception at the designated supporting site. In this type of arrangement, there is absolutely no duplication of resources.

maximum interchange between the developing and supporting agencies prior to transfer and little risk of support problems during transition and transfer.

Two of the most expensive resources required for ECS software management and support are personnel and weapon system equipment. Cost of support can be significantly reduced if a support site is selected where the bulk of these resources are available without additional acquisition. For example, the F-111 OFP support is located at the Sacramento Air Logistics Center, which is the depot for the F-111 aircraft and its digital avionics. To support F-111 OFP's requires approximately 60 technical personnel, the use of some 7 - 10 million dollars worth of avionics assets configured into laboratory development and test mock-ups, and the use of F-111 aircraft for flight testing OFP changes. The same system equipment is required for engineering support of the F-111 avionics system. Co-location at the Sacramento Air Logistics Center permits the use of a single laboratory (The F-111 Avionics Integration Support Facility, reference Figure V-1) to satisfy both requirements. Co-location further permits the laboratory to borrow avionics assets from the depot, eliminating the need for procurring dedicated resources. Also, co-location at the Sacramento Air Logistics Center with other F-111 aircraft engineering disciplines permits sharing of flight test aircraft. There were no appreciable savings in personnel costs by selecting this site to support F-111 OFP's because at the time there did not exist a pool of software expertise. However, now that this expertise is available, it becomes a factor in evaluating support alternatives for future ECS software workloads.

PERSONNEL:

The task of management and support of ECS software is closely integrated with the weapon system management and is highly technical. It requires management personnel with program management and technical backgrounds, and with weapon system experience and expertise; and technical personnel who are highly skilled in system engineering and software disciplines. These personnel include ECS software managers, ECS computer system engineers, software engineers, technical analysts, scientific programmers, data processing programmers, test engineers, hardware engineers, configuration managers, technicians, technical writers, draftsmen, clerical and supply personnel. The preponderance of personnel required to support ECS software are technical and the number depends of many factors. Some of the more important ones are: software complexity, saturation of the computers, frequency of changes, type of changes, programming language, software structure, quality of documentation, quality of development and test tools, documentation, test and integration requirements, routine maintenance requirements, and the experience level of personnel. All of these factors must be considered when estimating personnel requirements. No attempt is made in this report to tie down numbers, but only to suggest types of personnel and their responsibilities and skills. Descriptions of support personnel are summarized in Appendix B. More detailed personnel descriptions are available in the reference. (18)

EQUIPMENT:

Equipment constitutes the primary ECS software development and test tools. It consists of both laboratory and weapon system resources. The sophistication and quantity of equipment required are highly dependent on

the particular ECS software being supported and support requirements. However, in all cases some type of host computer, dynamic simulator, equipment mock-up and acquisition system will be required, as well as some weapon system equipment.

A host computer system will be required in order to: (1) compile and reassemble source data, (2) generate program code and documentation, (3) operate and control laboratory simulators and mock-ups, (4) acquire, reduce and analyze test data, (5) store, manipulate, and retrieve data and documentation, and (6) execute miscellaneous computer programs.

A dynamic simulator system will be required in order to develop and test ECS computer programs under dynamic conditions in a simulated real world environment.

A weapon system equipment mock-up will be required in order to test and evaluate interfaces between the weapon system equipment and the ECS software which includes tests for timing, compatibility, environmental conditions, RFI/EMI, etc.

A data acquisition system will be required in order to acquire test data, both in the laboratory and during ECS software certification.

Weapon system equipment will be required in order to support ECS software development, and to support laboratory and certification tests.

A more detailed description of this equipment is provided in Appendix B.

SOFTWARE:

Software can be divided into two categories, ECS computer programs and the support software system. ECS computer programs are sometimes referred to as the end item computer programs or operational computer programs. It is the software which resides in the weapon system computers, e.g., OFP's,

and the software which generates the requirement for a management and support system. All other software is considered in the support category. The support software system consists of all attendant computer programs used in the process of effecting changes and modifications to the ECS computer programs. This software can be grouped into programming aids, verification tools, host computer operating systems, utilities, simulation and test bed control systems, application programs, display programs, data acquisition software, configuration management software, and miscellaneous programs. The total number of programs can be large and depends on the degree of laboratory sophistication and automation. For example, there are some 1500 computer programs which make up the support software system for seven F-111 OFP's. Descriptions of support software are summarized in Appendix B.

DOCUMENTATION:[1]

Sufficient documentation is essential for effective configuration management of the ECS software and the software support system. Further, adequate documentation is required to permit ease of software and equipment maintenance and modification by competent technical personnel. Two things must be remembered when determining documentation requirements: (1) it is expensive to procure and maintain; and (2) it is to be used by highly competent technical personnel in a laboratory environment. Therefore, every effort should be made to use the same documentation that was used for acquisition development, and commercial documentation should be acceptable

1. Documentation for the purpose of this report means engineering documentation and does not include documentation which is a part of the Technical Data Systems, e.g., technical orders.

for "off the shelf" laboratory software and equipment. Documentation can be divided into three categories: ECS software, support software and hardware.

ECS Software: Generally, the documentation requirements are fairly straight forward for the ECS computer programs. On this software, the following documents are normally specified: (14:3-4)

- Weapon System Specification
- ECS Computer System Specification
- ECS Software Design Specification
- ECS Software Product Specification
- ECS Software Version Description Document
- ECS Software Configuration Index
- ECS Software Change Status Report
- ECS Software Users Manual
- ECS Software Computer Programming Manual
- ECS Software Test Plans, Procedures and Reports
- ECS Software Program Listings
- ECS Software Source and Object Programs
- Interface Control Documents

It is generally a good practice to acquire as reference material, copies of any additional data or documentation that the contractor used in acquisition development.

Support Software: Documentation requirements are not so straight forward for the support software. First, the support software may or may not be coming in total from the acquisition development contractor and second, even if it were, it (or its documentation) may not have been

specified as deliverable on the acquisition development contract. Further, it is very impractical, both from a cost and maintenance aspect, to require all of the documents listed above, for the support software. The important thing here is to ask for documentation that adequately describes each support program and provides sufficient information so that it can be maintained and changes can be implemented. Generally, the documentation should, as a minimum, provide narrative descriptions, baseline descriptions and logic flow; and describe each program in terms of inputs, outputs, functions, transformations, algorithms, equations, timing, logic, and size. This information is generally contained in functional and implementation specifications, logic flow diagrams, program listings, source and object programs, programmer manuals, and user manuals. This documentation should be acceptable in commerical form.

Hardware: For the weapon system equipment, specifications and drawings are normally delivered on the acquisition contract and it should only be necessary to obtain copies. In addition, user technical data will normally be available.

For other laboratory equipment, commercial documentation should be adequate. The important thing again is to assure that the documentation describes system performance, and provides sufficient detailed description for operation, preventive maintenance, trouble shooting, repair, part replacement and modification. The documentation should include performance specifications, functional descriptions, theory of operations, drawings, part lists, wiring diagrams, user manuals, maintenance procedures, and interface drawings.

FACILITIES:

Several problems generally arise concerning facilities required to house resources previously discussed. The most important is the lead time for procurement if they are not available or require major modification. This becomes a pacing item because of military construction procurement ramifications. Because of this, it is absolutely essential that facilities be identified early to avoid having equipment and no place to operate it. Since most of the equipment involves computers or computer peripherals, power and environmental requirements typically are the same as those for other computer operations, i.e., stringent power, temperature, air circulation, and dust controls. Normally, these facilities also require controlled access. Further, if classified data will be processed, then security requirements will also have to be met. All of these requirements take a great deal of time to implement. Therefore, it is always advantageous to seek out facilities which already meet these requirements. It is also highly desirable to have facilities that have adequate office work and study areas for support personnel. A great deal of efficiency is lost if personnel are not co-located with the laboratory. Floor space requirements will vary with system, but 5,000 - 6,000 square feet is not an unreasonable requirement for the average system.

In summary, this section, in conjunction with Appendix B, has discussed personnel, equipment, software, documentation and facilities essential for ECS software support. Emphasized were resource cost (in particular personnel and weapon system equipment), need for highly

qualified personnel, tailoring of documentation requirements, and the need to identify early in weapon system acquisition development all resource requirements.

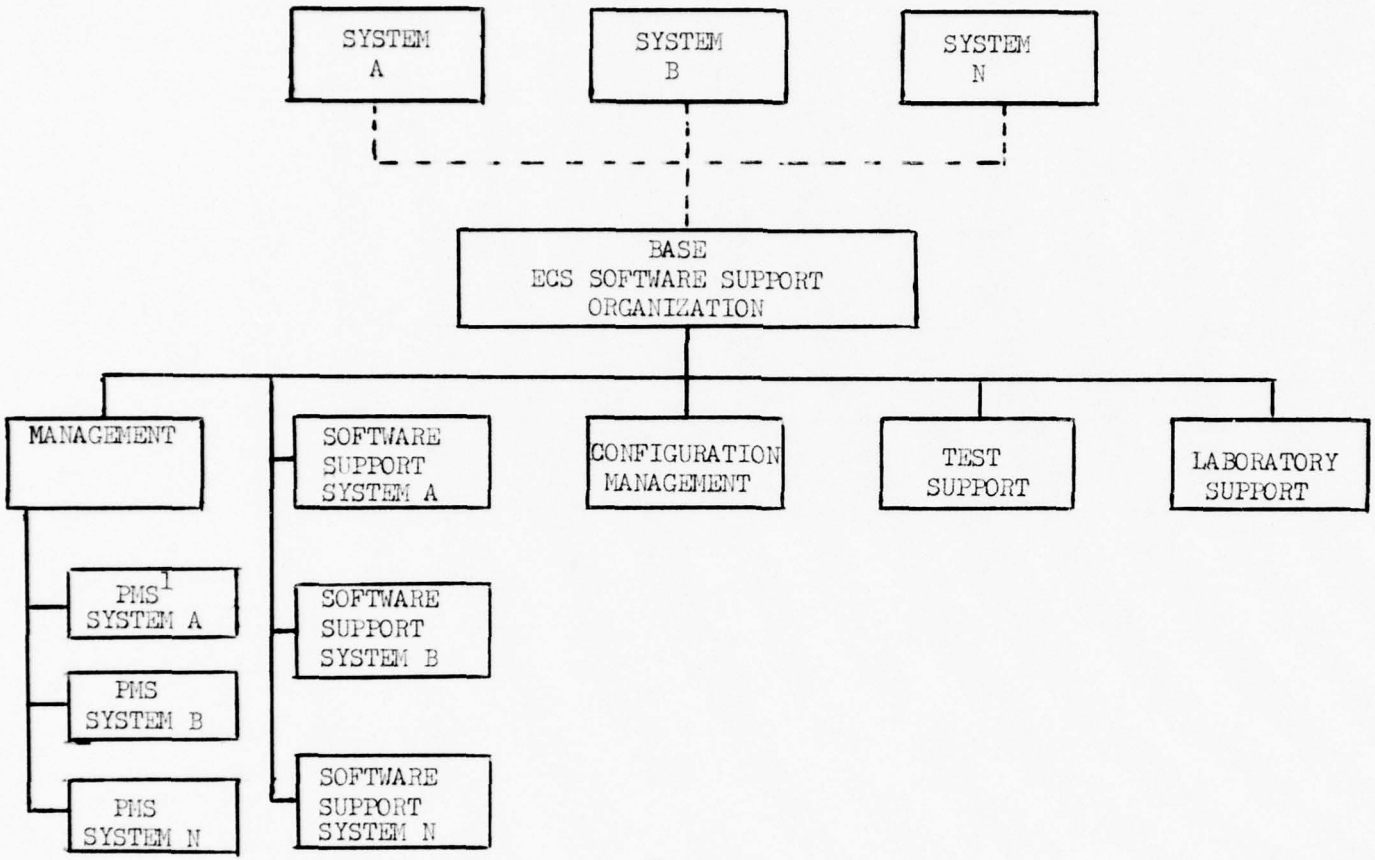
SECTION VII

SUPPORT ORGANIZATION

Support agency policies and regulations normally promulgate organization structure and responsibilities which will vary with agency mission and support role. If ECS software support is one of the agencies assigned responsibilities, then a substantive software organization is essential to effective, efficient and responsive support.

It is suggested that a software support organization structured and interfaced as illustrated in Figure VII-1 can optimize these support features. It places total software support responsibility and resources under a single manager, which maximizes the proficiency of software personnel and maximizes the sharing of resources. It also minimizes excessive coordination and job redundancy. Further, the organization interfaces at a level where it can be responsive to all system managers and is visible to the user for direct communication and coordination.

The organization is structured so that program management can be applied to software modifications. Under this concept, the management section has responsibility for ECS software management, as well as responsibility for budgeting, financial matters, contracts, training and resource management, and has a program manager assigned to the ECS software for each weapon system. The program manager has complete responsibility for software modifications and plans, schedules, budgets, coordinates and directs all resources used in accomplishing modification tasks. He also has responsibility for the coordination and interface with all organizations and agencies external to the software organization. The respective ECS software support sections have responsibility for technical software support and system integration,



1. PROGRAM MANAGER, SOFTWARE

ECS SOFTWARE SUPPORT ORGANIZATION

Figure VII-1

and provide the personnel required to develop software modifications and changes, staff problems, conduct studies, etc. The configuration management section has responsibility for all software configuration management and formal documentation. The test section has responsibility for all formal and independent testing to include preparation of formal test plans, procedures and reports, and test implementation. The laboratory support section has responsibility for all laboratory support resources (both hardware and software) and responsibility for providing laboratory support services for all ECS software systems.

SECTION VIII

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

SUMMARY:

In summary, this report has addressed and discussed management and support of ECS software after weapon system deployment. It included support planning, requirements and alternatives with emphasis placed on tailoring support to generic software classes. The change process was discussed with emphasis on the block change concept and development cycle. Software configuration management was discussed with emphasis on the requirement to manage software as configuration items; the need to tailor configuration management principles to support system requirements, (in particular documentation); change control; the computer program configuration sub-board; status accounting and audits; and automated documentation generation. Software testing was discussed with emphasis on test objectives; software error sources; test plans, procedures, and reports; and the verification, validation, certification method of testing. Support resources were discussed in detail with emphasis on cost, alternatives for minimizing cost, types of personnel skills, and laboratory and weapon system equipment. Finally, a support organization was suggested which places all ECS software support resources at a base under a single software manager.

CONCLUSIONS:

ECS software can provide the weapon system with tremendous flexibility in responding to problems and requirements because of the relative ease at which it can be changed.

However, to provide this capability during the operational phase of the weapon system life cycle requires an efficient, effective and responsive ECS software management and support system and a major resource commitment. In order to accomplish this, the system:

(1) must be planned early in weapon system acquisition development and fully coordinated with the weapon system integrated logistics support plan;

(2) must include a staff of highly qualified and experienced personnel and a maintenance and modification laboratory equipped with a host computer system, system equipment mock-ups; dynamic simulation, data acquisition and weapon system equipment; and

(3) must provide a systematic method for developing ECS software changes which employs strong program management, configuration management, test management and system engineering.

RECOMMENDATIONS:

1. Greater emphasis be placed on educating management on requirements and benefits of ECS software support.

2. Stronger and more substantive management policies and support be provided in acquiring resources, in particular personnel, and implementing ECS software support plans.

COMPUTER PROGRAM CHANGE REQUEST

I.D. NUMBER

1. TITLE

2. DATE

3. COMPUTER PROGRAM IDENTIFICATION:

4. DESCRIPTION/PRESENT OPERATION:

5. DESIRED OPERATION:

6. REASON FOR CHANGE:

7. CHANGE HISTORY/RELATED CHANGES:

8. REQUESTED BY:

NAME ORGN PHONE

9. REQUESTING WING: COORDINATION

NAME PHONE

10. REQUESTING COMMAND: APPROVAL

NAME PHONE

11. SUPPORTING AGENCY: APPROVAL

NAME PHONE

APPENDIX B

SUMMARY DESCRIPTIONS OF RESOURCES

PERSONNEL:

ECS Software Management: These personnel manage the overall ECS software support effort in concert with the weapon system support program. They plan and budget for the development and implementation of supporting resources, forecast personnel requirements, and recruit and train new personnel. They establish ECS software support policy, procedures, and methods; allocate and commit resources; provide administrative and technical guidance to subordinates; and assure that ECS software objectives remain in consonance with the weapon system support objectives. In order to effectively perform assigned tasks, managers must possess a detailed knowledge of the weapon system, ECS software, and ECS software support resources. Further, they must have a thorough knowledge of the support agency and its relationship with other DoD organizations and be capable of effectively communicating and coordinating through the vertical and horizontal management structure.

Embedded Computer System Engineers: These personnel perform the function of program manager for system design and development of large ECS software modifications, i.e., block changes. They establish change requirements through coordination with the users and system manager; direct the preparation of change specifications; and plan, schedule and budget for the complete modification process to include the investigation and analysis of requirements; design, development, code, debug, test, evaluation, and integration of approved changes, and documentation and release of final results. They establish priorities and time phase all tasks and resources required for the program, from ECS software change inception through final completion and release to the user. These personnel also serve as the recognized technical expert for the ECS system and its

interface and integration with the weapon system and all supporting elements. In order to effectively perform these assigned responsibilities, they must possess extensive academic and professional knowledge of scientific and engineering principles. They must have a thorough and detailed knowledge of the ECS software, the ECS computer and the weapon system to include all ECS interfaces and system performance. They must have a detailed knowledge of real-time software operation; laboratory simulation; host computers and system mock-ups; software design; configuration management; documentation; integration testing; evaluation; verification and validation. They must also possess a working knowledge of funding, procurement, and weapon system management; and must be capable of effectively communicating and coordinating through the vertical and horizontal management structure.

Software Engineers: These personnel investigate ECS software requirements and analyze potential solutions considering trade-off analysis involving implementations, cost, algorithm developments, timing requirements, memory size, hardware/software integration, and support equipment. They translate change requirements into engineering specifications; design change mechanization and integration; develop programming code; and debug, test and document results. In order to effectively perform these assigned tasks, they must have a thorough and detailed knowledge of digital systems; ECS software; ECS computers; the weapon system interfaces and performance; real-time programming; laboratory support systems; math modeling; computer architecture and programming languages.

Staff Analysts: These personnel serve as mathematical experts for ECS software and develop numerical and mathematical solutions to complex computer programs operating in real-time. They develop mathematical models

using time and frequency domain transforms and apply optimal estimation theory in the design of changes to Kalman and similar statistical filters. They also analyze and evaluate complex mathematical models used in such functions as navigation, fire control, and weapon delivery; and collect, correlate and develop statistical performance data. In order to effectively perform these tasks, they must possess the same detailed system expertise as the software engineer, as well as being an experienced mathematician.

Scientific Programmers: These personnel maintain the laboratory support software system. They design and develop support software changes in order to maintain compatibility with the ECS software changes, and to satisfy requirements for special test, and acquisition software programs. They prepare software change requirements and specifications, taking into consideration host computer constraints, computer peripheral interfaces and compatibility with all laboratory equipment; and design, code, debug, test, evaluate and document final changes and new programs. In order to effectively perform their assigned tasks, they must possess a detailed knowledge of the support software system; ECS software; host computer systems; simulators; and data acquisition, reduction, and analysis systems. They must also have an intimate knowledge of real-time programming; mathematics; assemblers; compilers; peripheral handlers; acquisition and analysis software; computer architecture; and high order, assembly and machine language programming. They should also be familiar with peripherals, digital hardware interfaces, and hybrid systems.

Programmers: These personnel prepare implementation specifications, design, program, trouble shoot, debug, test and document non-engineering

application software and associated laboratory utility and off-line miscellaneous computer programs. They provide programming support to the scientific programmers, analysts, and engineers; and must be capable of programming in assembly and high order language.

Test and Evaluation Engineers: These personnel serve as the formal test and evaluation team for ECS software modifications. They establish testing requirements; prepare formal test plans; develop test procedures; conduct ECS software verification and validation; and prepare the test reports. In order to effectively perform their assigned tasks, they must possess the same detailed system expertise as the software engineer.

Configuration Management: These personnel develop, establish, implement, and enforce regulations, policies, procedures and directives required for configuration management of the ECS software and of the software support system. They review, control, coordinate, record and account for all software changes. They assure compliance and conformity with configuration management standards and formats and check to assure that all software changes are controlled, documented and compatible with the weapon system and supporting elements. For automated configuration management systems, they prepare and control the software programs which generate documentation. They also maintain and control the computer data base software libraries, and storage area. In order to perform these assigned responsibilities, they must possess detailed knowledge of configuration management principles, and have in-depth knowledge of software and computer systems.

Other Personnel: In addition, there are requirements for support from: (1) hardware engineers for the ECS interfacting subsystems, and for laboratory hardware equipment; (2) electronics technicians for laboratory

equipment maintenance; (3) technical writers to prepare final engineering and technical documentation on released ECS software; and (4) draftsmen, clerical and supply personnel.

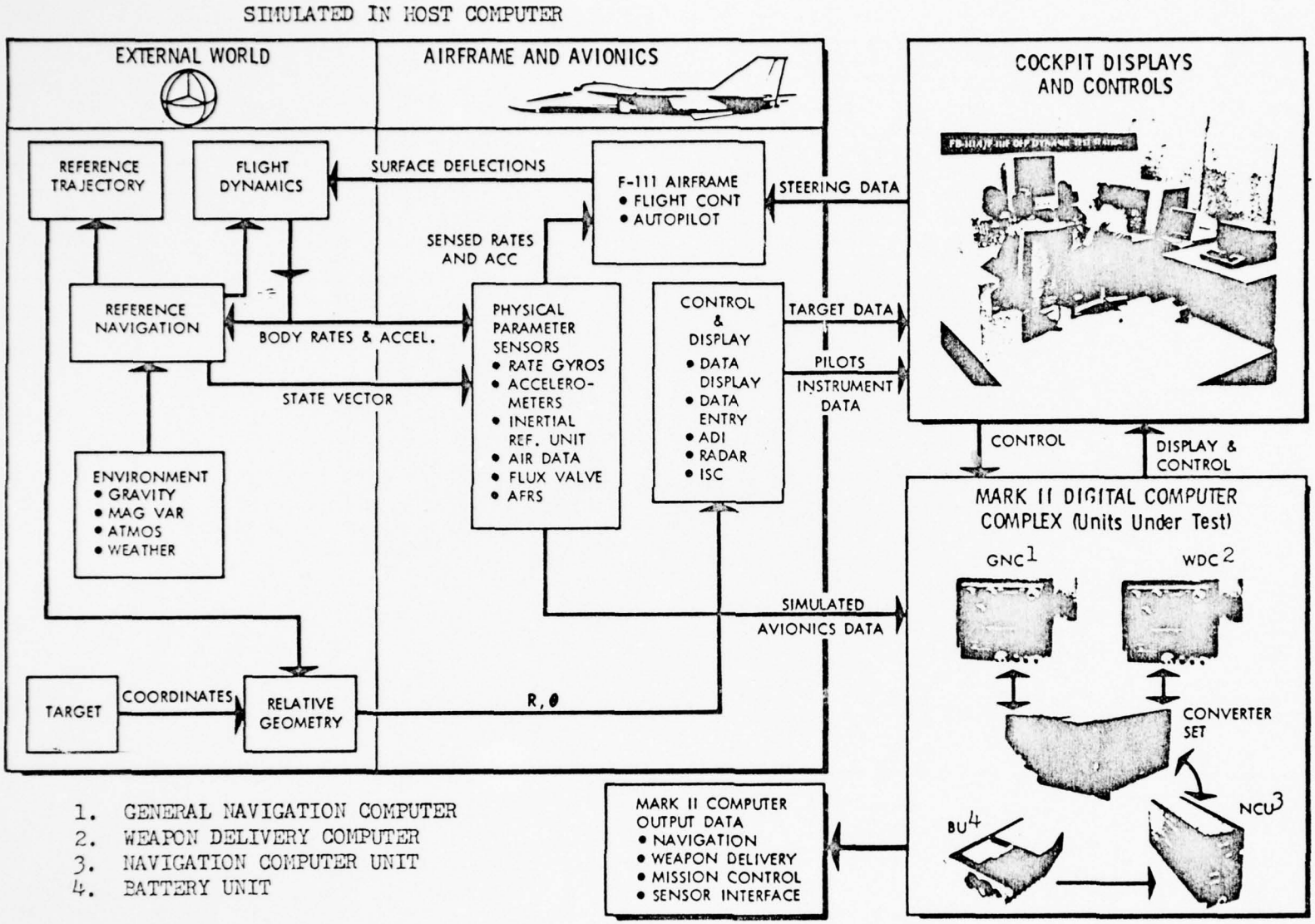
The paramount problem impacting ECS software support is personnel. First, there are no appropriate Civil Service classification series or military career fields for the types of personnel just discussed. This makes it extremely difficult to acquire and retain good people. Second, trained and experienced personnel are in short supply, both in DoD and industry, making it tough to find good people. And third, because of the skill levels required, training is long and expensive.

EQUIPMENT:

Host Computer Systems: These are the general purpose computers and peripherals, i.e., magnetic tape units, discs, CRTs, printers, card readers, paper tape punch, etc., which are used to: (1) execute the support computer programs; (2) operate and control the dynamic simulators, computer controlled mock-ups, and data acquisition systems; and (3) perform various other laboratory functions requiring the use of a computer. Depending on the ECS software support requirements, these computers can range from large systems like an IBM 360/65 to relatively small and inexpensive mini computers like a PDP 11/40. The mini computer approach is becoming extremely popular for several reasons: (1) they are relatively inexpensive compared to the large computer systems; (2) they provide greater flexibility and versatility; (3) they are normally easier to modify or customize for specific laborator operations; (4) they are easier and cheaper to maintain; (5) they take up less floor space; (6) if a failure occurs, only a part of the operation goes down;

and (7) less of a queuing problem exists when operating in real-time, i.e., driving a dynamic simulator. Their disadvantages are: (1) many mini computer manufacturers have not adequately perfected their software; (2) customer support is sometimes less than desirable; and (3) for large ECS computer systems, they may not have the capacity to meet support requirements. (15), (17)

Dynamic Simulators: These are hybrid computer systems which use a host computer to simulate and create a dynamic real world environment in the laboratory, for the ECS software; and is accomplished with the software resident in its ECS computer. These systems are configured in several ways. One configuration uses the host computer to simulate all ECS computer inputs. Another configuration uses some actual weapon system equipment and stimulates this equipment with host computer inputs; and other configurations use some actual weapon system equipment and live inputs. Dynamic simulators have proven to be invaluable development and test tools for ECS software. Basically, they provide a capability to: (1) generate large quantities of data, mission scenarios and test scenarios as inputs to the ECS software under dynamic, repeatable and controlled conditions; (2) provide man-in-the-loop or automated control during test; and (3) acquire performance data on the ECS software under dynamic test conditions. Figure B-1 illustrates the F-111 OFP Dynamic Simulator. It functions as follows: The aircraft computers with their resident OFP's, shown in the block labeled Mark II Digital Computer Complex, receive inputs from the aircraft cockpit mock-up and the host computers, and output control signals and display data to the cockpit. They also output selected acquisition data for post analysis. The host computer generates simulated avionics data going to the aircraft



F-111 OFP DYNAMIC SIMULATOR

Figure B-1 (16:913)

computers; generate stimuli for the avionics housed in the cockpit; and generate control signals, display data, and operator cues going to the cockpit. This is accomplished by using computer program models for the aircraft dynamics, avionics, external world and navigation reference systems. The cockpit is configured very similar to the aircraft cockpit and has actual aircraft controls and displays. This permits realistic operator inputs and control. (16), (14)

System Mock-ups: Mock-ups functionally integrate, in the laboratory, the weapon system equipment which interfaces with the ECS computers in the real world. For an aircraft bomb/navigation system, this equipment might be the sensors, cockpit displays, and control systems. Mock-ups are normally configured to allow easy access to equipment interfaces and ease of instrumentation and monitoring of performance. They are normally tied to the host computer system so that both manual and automated control can be achieved. They are used to test hardware/software interfaces, timing, compatibility, RFI/EMI, and environmental conditions. They are also used to investigate field problems when it has not been determined whether the problem is hardware or software. (15), (14)

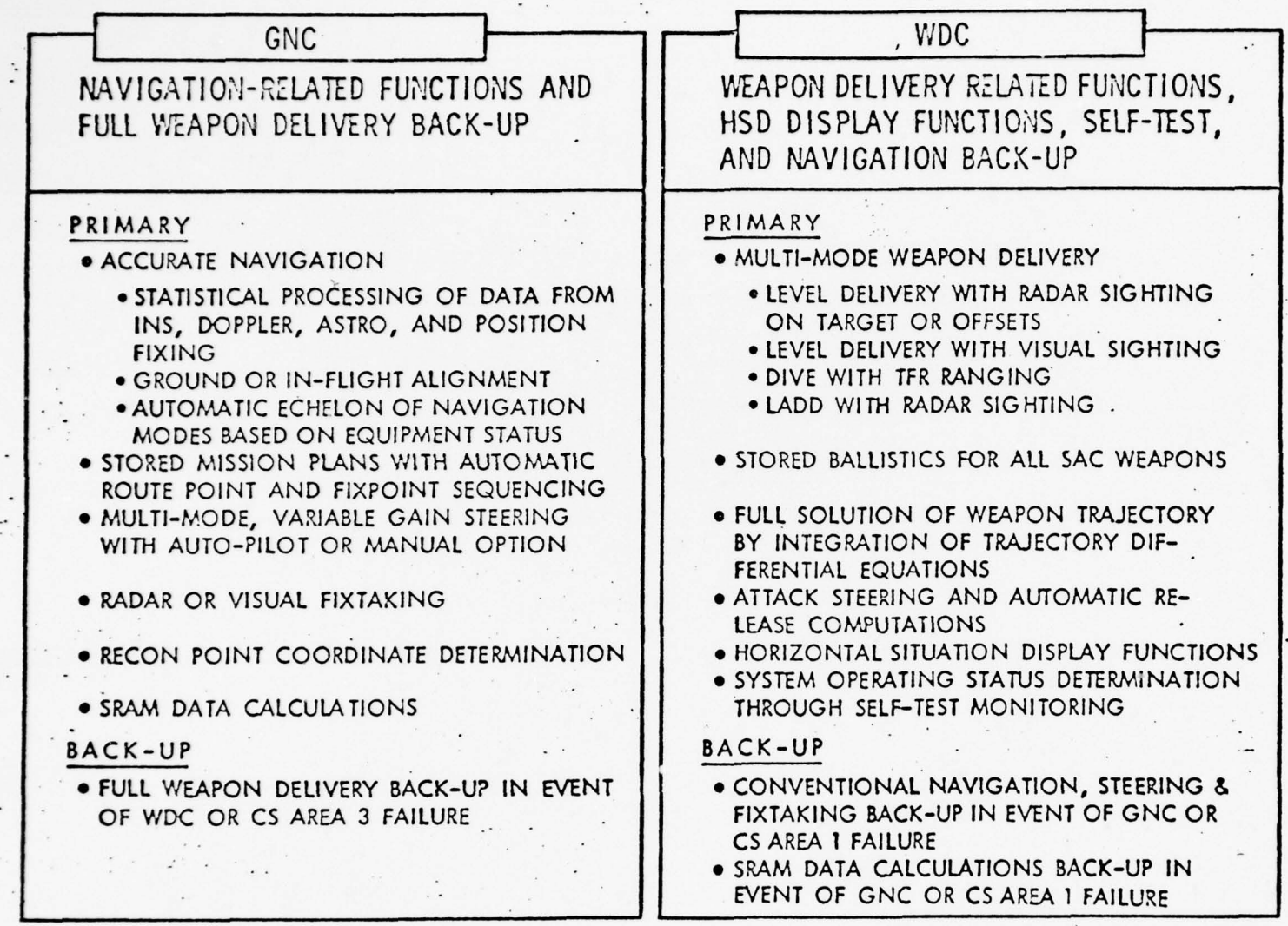
Data Acquisition Systems: These are systems which are used to gather performance data on the ECS computer program during stages of development and testing. They differ from the software tools used for verification, debug and diagnostics in that they actually monitor the ECS software as it performs in the ECS computer. These systems are used in laboratory testing, as well as testing in actual weapon system environments. They generally fall into two categories: (1) systems that monitor the ECS computer at its interface (I/O); and (2) systems that monitor performance

internal to the ECS computer. Instrumentation systems used in flight test are examples of the I/O monitoring system. These are generally limited to validation and certification testing. The latter type is designed for laboratory use. These systems provide total visibility inside the ECS computers during dynamic real-time program execution. They tie to the internal bus structure of the ECS computer and capture data for full, partial, recent history, event and timing traces. These types of systems can be used in all phases of ECS software development, but are extremely valuable for program debug, verification and validation. (16:916)

System Resources: This is the weapon system equipment required in the laboratory to support ECS software change development and test. This equipment is used in the laboratory mock-ups and dynamic simulators. In some cases, a complete system is required for certification testing, e.g., flight test aircraft for OFP's. To assure test results, this equipment should be of production quality and remain in an operational configuration. In most cases, it will represent the largest physical resource investment.

SOFTWARE:

ECS Computer Programs: These are end item computer programs which reside in weapon system operational computers. Figure B-2 illustrates ECS computer programs for the FB-111A aircraft. As seen, ECS computer programs are generally an aggregate of subroutines which perform multi-functions and modes for the weapon system. These programs are subject to continual change and must be delivered in a state and format that provides for ease of maintenance and modification. Changes to these programs can result in changes to support resources.



FB-111A OFP's

Figure B-2

Programming Aids: These are assemblers, compilers, processors, editors, etc., which execute on laboratory host computers and are used by programmers and engineers as development tools. Assemblers, compilers, and processors permit programming changes to be made to the ECS software in assembly and high order language; generate documentation such as memory maps, absolute listings and program listings, and produce ECS computer program source and object code on disc, magnetic tape, paper tape, etc. Editors are programming aids which permit on-line interactive text editing and formatting of computer program changes generally from CRT termintals.

Verification Tools: These are computer programs which execute on laboratory host computers and aid in debugging program code, logic and structure; and are used in program verification. (1:5-4) These programs include:

Comparators: Programs which do an instruction-by-instruction comparison on two versions of the same program. The comparator flags differences, thus indicating where a program has been changed.

Editors: Programs which find and flag coding errors and produce cross-reference listings.

Flowcharters: Programs which operate on a program written in either assembly or higher order language and produce a flowchart of the program. This flowchart can then be compared to the original program flowchart.

Logic/Equation Generators: Programs used to reconstruct arithmetic text and to flowchart assembly language programs.

Pathfinders, Traps and Traces: Programs which analyze possible paths through a given program. This information is useful for planning test cases.

Interpretive Computer Simulations (ICS's): Programs which perform an instruction-level simulation of the ECS computer. The host computer is programmed to act on the ECS computer program in exactly the same way as the ECS computer would act. The host computer will give the same results bit-for-bit as would be produced by the ECS computer under the same conditions and inputs. The advantage of an ICS simulation is greater visibility and diagnostics during program debug.

Host Computer Software: This software consists of: (1) the host computer operating system, i.e., the program which remains resident in the host computer and controls its execution and operation; (2) host computer diagnostics which are used to trouble-shoot the host computer hardware and operating system; (3) peripheral handlers which control the operation of the computer peripherals, i.e., tape decks, discs, CRT's, printers, readers, etc.; and (4) utility programs which interface and link the host computer and operating system with non-standard handlers.

Simulation and Test Bed Control Software: This software executes on the laboratory host computer and consists of real-time and non real-time master executive programs which control the overall operation of dynamic simulators and computer controlled test bed mock-ups. This software also plans, schedules, prioritizes and controls subservient software such as equipment models, application models, displays, programs data acquisition programs, etc. (16), (19)

Application Software: This software executes on the laboratory host computer and is normally a sub-set of the dynamic simulation and computer controlled mock-up software. These computer programs consist of weapon system dynamics; models of equipment sub-systems which interface with or impact on the ECS computer system; external world programs, (e.g., atmosphere, earth, terrain, gravity, reference geometry, targets, reference systems, etc.); and error, noise and bias models. (16), (19)

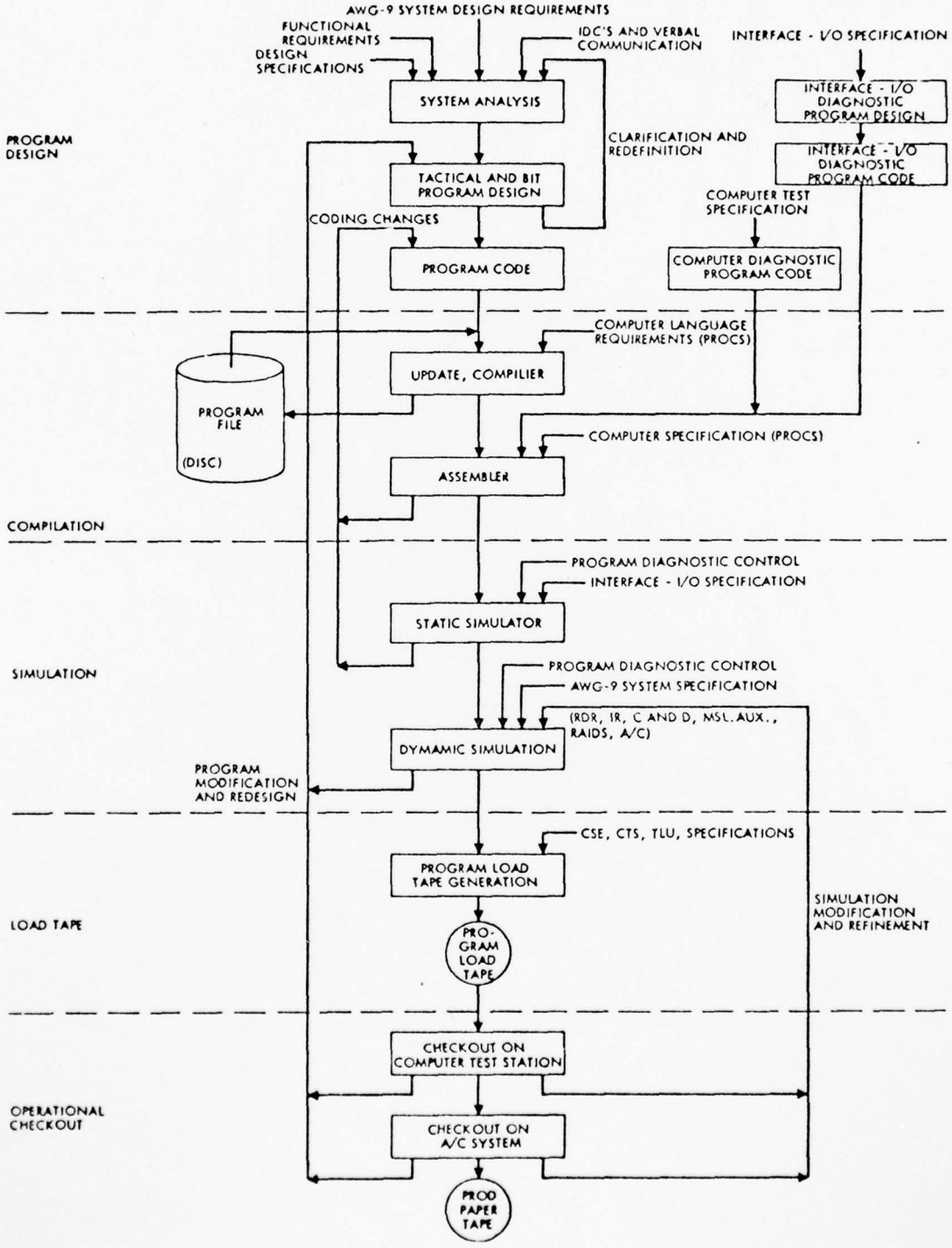
Display Software: This software executes on the laboratory host computer and is used to generate special graphical displays in real and non-real time, (e.g., displays of targeting geometry and simulated position and attitude of the weapon system). Normally, this software displays data both in empirical and graphical formats. (16), (19)

Data Acquisition Software: This software consists of data acquisition, reduction and analysis programs. Acquisition programs control the operation of the instrumentation and data acquisition system, format and temporarily store data. Depending on the acquisition device, this software may reside in the acquisition system, host computer, the ECS computer, or some combination. Data reduction programs execute on the host computer and format, convert, time tag, correlate and produce the required computer outputs. Analysis programs also execute on the host computer and evaluate reduced data against defined criteria and display or print results. (22)

Configuration Management Software: This software executes on the laboratory host computer and consists of: (1) documentation generators which tabulate ECS computer program change requirements and generate specification data, source and object code, narratives, baseline configurations and other configuration data on each ECS computer program change;

(2) documentation processors which define, standardize and format software documentation; and (3) data base control and status accounting computer programs which maintain listings and records of all software and software documentation to include status and configuration identification. (17)

Figure B-3 illustrates the support software system used in development of F-14 aircraft weapon control computer programs. The figure is helpful in illustrating the complexity and inter-relationship of support software for ECS computer programs.



AWG-9 program development process

The support software system is utilized in all five design phases of program development

SUPPORT SOFTWARE SYSTEM
F-14 WEAPON CONTROL COMPUTER PROGRAM

Figure B-3

BIBLIOGRAPHY

1. Air Force Regulation 800-14, Volume I, Management of Computer Resources in Systems, and Volume II, Acquisition and Support Procedures for Computer Resources in Systems, 26 September 1975.

2. Boehm, Barry W., "Software and Its Impact: A Quantitative Assessment", Datamation, May 1973.

3. Clapp, J. A., "Software Engineering: Problems and Future Developments", MITRE Technical Report, Mtr 2791, 2 January 1974.

4. Department of Defense Directive 5000.3, Test and Evaluation, 19 January 1973.

5. Department of Defense Directive 5010.19, Configuration Management, 17 July 1968.

6. Department of Defense Instruction 5010.21, Configuration Management Implementation Guide, 6 August 1968.

7. Department of Defense Directive 5000.29, Management of Computer Resources in Major Defense Systems, 26 April 1976.

8. DeRoza, Barry C., "An Introspective Analysis of DoD Weapon System Software Management", Defense Management Journal, V. II, No. 4, October 1975.

9. Dyke, Frank P., Lt. Col. USAF and Reifer, Donald J., "Computer Program Verification and Validation, A SAMSO Perspective", Conference Proceedings 12-13 May 1976, Managing The Development Of Weapon Systems Software, Maxwell Air Force Base, Alabama.

10. Fagerlin, Mark L. and Nenninger, Paul, Course Handbook, Configuration of Software Programs, School of Engineering and Applied Science, Continuing Engineering Education Programs, The George Washington University, August 1976.

11. Manley, John H., "Embedded Computer System Reliability", Defense Management Journal, V. II, No. 4, October 1975.

12. McCarthy, Rita, "Applying the Technique of Configuration Management to Software", Defense Management Journal, V. II, No. 4, October 1975.

13. Military Standard, MIL-STD-499A (USAF), Engineering Management, 17 July 1969.

14. Ogden Air Logistics Center, F-16 Computer Resources Integrated Support Plan (CRISP), Volume I, Operational Flight Programs, 15 March 1976.

15. Patterson, Alton E. and Algire, Richard G., Capt. USAF, "An Approach to Modern Avionics Integration Support", Proceedings of the IEEE, 1976, National Aerospace and Electronics Conference, NAECON 76, Dayton, Ohio.

16. Patterson, Alton E. and Long, Eugene M., "The F-111 OFP Dynamic Simulation System", Proceedings of the IEEE, 1976, National Aerospace and Electronics Conference, NAECON 76, Dayton, Ohio.

17. Patterson, Alton E., "Operational Flight Programs: Change and Control", Conference Proceedings 12-13 May 1976, Managing the Development of Weapon Systems Software, Maxwell Air Force Base, Alabama.

18. Patterson, Alton E., Sacramento Air Logistics Center, Process Review II, F-111 Software Support, Control #8039, 18 December 1976.

19. Requirements for Sacramento ALC F-111 Operational Flight Program Support Simulation System, General Dynamics/Convair Aerospace Division, GD, FW, Tx, Report FZE-624-SM001, 12 July 1974.

20. Sacramento Air Logistics Center, MME, FB-14, Project Test Plan FB-111A Operational Flight Program, Report #76-609, 25 March 1976.

21. Sacramento Air Logistics Center, MME, Engineering Flight Test Evaluation F-111F Operational Flight Programs, (GNC/WDC), OFP 511/611 (Issue F-11) Report #76-613, 9 September 1976.

22. Sacramento Air Logistics Center, MME, Preliminary Investigation of MME AISF Integration and Data Reduction Plans and Requirement For Support Of F-111F/PAVE TACK Flight Testing at SM/ALC, Engineering Report NR. 77-600, 6 October 1976.

23. Thornell, D. V., "Preparation for Software Management Responsibility Transfer", Conference Proceedings, 12-13 May 1976, Managing The Development of Weapon System Software, Maxwell Air Force Base, Albama.

END 9-77
DDC