


LOAN DOCUMENT

PHOTOGRAPH THIS SHEET

AD-A276 362


DTIC ACCESSION NUMBER

LEVEL

INVENTORY

①

WL-TR-94-1008
DOCUMENT IDENTIFICATION
Feb 94

PROPERTY OF THE GOVERNMENT
NOT TO BE DISTRIBUTED

DISTRIBUTION STATEMENT

ACCESSION FOR:	
NIJ	GRAB <input checked="" type="checkbox"/>
DTIC	TRAC <input type="checkbox"/>
UNANNOUNCED <input type="checkbox"/>	
JUSTIFICATION	
BY	
DISTRIBUTION/	
AVAILABILITY CODES	
DISTRIBUTION	AVAILABILITY AND/OR SPECIAL
A-1	

DISTRIBUTION STAMP


DTIC
MAR 1 1994
C D

DATE ACCESSIONED

DATE RETURNED

94 2 24 1 6

DATE RECEIVED IN DTIC

94-06193


REGISTERED OR CERTIFIED NUMBER

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-FDAC

H
A
N
D
L
E

W
I
T
H

C
A
R
E

WL-TR-94-1008

GETTING A HANDLE ON DESIGNING FOR
AVIONICS SOFTWARE SUPPORTABILITY AND
MAINTAINABILITY



CHARLES P. SATTERTHWAITE

FEBRUARY 1994

FINAL REPORT FOR 05/18/92-05/22/92

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

AVIONICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT PATTERSON AFB OH 45433-7409

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

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

Charles P. Patton

Project Engineer
WL/AAAF-3

Robert L. Harris

Chief, WL/AAAF-3

Anna M. Harris

Chief, WL/AAAF

If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify WL/AAAF, WPAFB, OH 45433-6543 to help us maintain a current mailing list.

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

FEB 1994 FINAL
GETTING A HANDLE ON DESIGNING FOR
AVIONICS SOFTWARE SUPPORTABILITY AND
MAINTAINABILITY

05/18/92--05/22/92

C
PE
PR 9994
TA 00
WU 00

CHARLES P. SATTERTHWAITE

AVIONICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT PATTERSON AFB OH 45433-7409

AVIONICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT PATTERSON AFB OH 45433-7409

WL-TR-94-1008

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS
UNLIMITED.

THE ENGINEERING DESIGN ISSUES OF SUPPORTABILITY AND MAINTAINABILITY ARE EXPANDED UPON SO THAT INTERESTED INDIVIDUALS CAN UNDERSTAND (1) HOW THESE ISSUES IMPACT A MODERN WEAPON SYSTEM, (2) WHAT IMPACT THE LACK OF CONCERN FOR THESE ISSUES IS HAVING ON TODAY'S WEAPON SYSTEMS, AND (3) HOW THESE ISSUES MIGHT BE BETTER CONSIDERED AND IMPLEMENTED IN FUTURE WEAPON SYSTEM DESIGNS.



DEPARTMENT OF THE AIR FORCE

WRIGHT LABORATORY (AFSC)
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6543

20 JAN 1992

REPLY TO AAA
ATTN OF

SUBJECT: Request for Public Release Approval (AFR/ASDR 190-1)

WL/DOOS
ASD/PA
TO
IN TURN

1. Please review the attached material for public release approval. The following information is provided in support of this request:

- a. Type of information (abstract, journal article, technical paper, film/script, technical report, etc): Technical paper
b. Title: Getting a Handle On Designing For Avionics Software Supportability and Maintainability
c. Author(s) name, title, and organization: Charles P. Satterthwaite, Engineer, WL/AAAF-3, WPAFB OH 45433-6543
d. Contract #/company name: N/A
e. Name/location of publisher: N/A
f. To be presented at (give sponsoring organization or technical society, location (city, state), and exact date of presentation): NAECON '92, Dayton OH 18-22 May 1992
g. Deadline for submittal (if other than presentation date): 30 January 1992

2. The reverse side of this letter has been completed/signed by requester.

3. The information contained in this material is unclassified, technically accurate, non-proprietary, and considered suitable for public release. It contains no computer software, owned or developed by or for the government. Export restrictions (ie, MCIL, Munitions List (ITAR), and CCL) and current AF/DOD policy have been considered prior to requesting public release approval.

FOR THE COMMANDER

[Signature]
(Division level/signature/title)
USAF
Higher echelon signature (if applicable)

ASD/PA APPROVAL

CLEARED FOR PUBLIC RELEASE
DEPARTMENT OF THE AIR FORCE
AERONAUTICAL SYSTEMS DIVISION (AFSC)
OFFICE OF PUBLIC AFFAIRS

[Signature] 27 JAN 1992
SECURITY REVIEW OFFICER (SRO)

ADDITIONAL INFORMATION

1. The following statement is applicable (check one):

All references are unclassified/unlimited and are available to the public.

references # _____ are subject to distribution limitation. No limited information from these references is included in the document for which clearance is being requested.

No references are contained in attached material.

Charles P. Satterthwaite 10JAN92
(Signature of requester) (date)

2. The technology contained in the material proposed for release:

a. Is (1) being applied, (2) nearing application (maturing), (3) _____ years or more away from application (select one or more as appropriate).

b. Resulted from technical efforts funded under Program Element 6.2 (identify PE, i.e., 6.1, 6.2, etc.).

c. Is the latest state-of the art: Yes No

d. Has previously been released to the public: Yes No

e. Has the following intended application: To educate avionics software practitioners as to the importance of designing supportable and maintainable parameters into new weapon systems applications.

3. Other applicable comments or rationale justify clearance for public release:

The opportunity to elevate Avionics Software Engineering Issues to an interactive avionics audience.

4. Government point of contact (name, symbol, and telephone):
Charles P. Satterthwaite, WL/AAAF-3, (513)255-3947

5. SDI involved: _____ YES NO
(If yes, explain) _____

6. Contract:

a. Contains a DD Form 254 yes no

b. Refers to a security classification guide yes no

GETTING A HANDLE ON DESIGNING FOR AVIONICS SOFTWARE SUPPORTABILITY AND MAINTAINABILITY

Charles P. Satterthwaite
Avionics Logistics Branch
Wright Laboratory
Wright Patterson AFB OH 45433-6543

ABSTRACT

The engineering design issues of supportability and maintainability are expanded upon so that interested individuals can understand (1) how these issues impact a modern weapon system, (2) what impact the lack of concern for these issues is having on today's weapon systems, and (3) how these issues might be better considered and implemented in future weapon system designs.

I. INTRODUCTION

The planners of future applications for avionics software are frequently tasked with the issues of supportability and maintainability of this software. These issues are often treated as unique design issues which await break-through technologies for their implementation. Oftentimes they are shoved aside (or deprioritized) in preference of more concrete and understandable concepts. As a result, poor specifications for supporting and maintaining systems are generated, which are increasingly costly in terms of the weapon system's life-cycle.

This paper addresses those people involved in the acquisition, logistics, and use of avionics software by showing cases of avionics software supportability and maintainability and how to go about designing it. Section I is an introduction of the material in this paper. Section II defines supportability and maintainability. Section III gives an idealistic example of a strongly designed supportable and maintainable futuristic fighter. Section IV reviews the state of supportability and maintainability today. Section V examines the new field of Software Engineering and how it should be focused to impact designing for supportability and maintainability. Section VI reviews the engineering design process and how it can accommodate supportability and maintainability. Sections VII and VIII apply the engineering design process to software scenarios and show where supportability and maintainability should be considered in their designs. Section IX gives a brief summary of this paper.

II. SUPPORTABILITY AND MAINTAINABILITY DEFINED

Supportability (in the context of Avionics Software) is the provision of sophisticated equipment, special facilities, and detailed documentation in order to perform maintainability. Maintainability is the probability that a failed computer software configuration item (CSCI) can be repaired in a specified amount of time using a specified set of resources [4].

III. AN IDEALISTIC EXAMPLE OF SUPPORTABLE AND MAINTAINABLE AVIONICS SOFTWARE

Let's project ourselves into the year 2020 and pretend we have the management responsibility for the nation's latest ALL-ENGAGEMENT Fighter. This remarkable aircraft can fly in space or in close ground support, it is stealthy, and it has a new impulse propulsion system which greatly reduces its fuel requirements. It also has a fully integrated open system architecture enabled by 50 million lines of source code.

More remarkable than this system's hardware features and capabilities is the built-in design for system supportability and maintainability. The Fighter is accentuated with redundant, reconfigurable, and fault-tolerant software which provides a full mission software report following each sortie. The Fighter is also arranged so that it can be configured as a simulator with the capability of supporting any level of testing, including sub-system testing and full-up integrated testing (air or ground). Every level of this Fighter's life-cycle is linked by a network of interconnected media which enables precise and traceable documentation as well as on-line communication. The significance of this is that the software maintainer, the acquisition specialist, and the pilot have a perfect flow of information in which they can report problems, specify requirements, and train each other through a common media.

Upon close examination of the Fighter's software, we discover more amazing design features. First of all, there

is 100% source code compatibility among the 300 embedded computers on the Fighter. These 300 embedded computers are written in Higher Order Language (Ada 2020) and are facilitated by two very compatible compilers. There are 1500 Software Modules which populate the Operational Flight Programs (OFPs) of the 300 embedded computers. Of these 1500 Modules, 40% (600 Modules) are common among the embedded computers and help encompass the Fighter's dynamically reconfigurable resources. Another useful design feature is the emphasis on reusable code. Thirty percent of the 1500 Modules (450 Modules) are static in nature and can be used over and over again between block cycle changes and between other aircraft.

Our final investigation is of the Fighter's Software Support Environment. There are two options available for those concerned with changing the Fighter's 300 OFPs. The first option is to use the Fighter's built-in capability, which is a pull-down avionics engineering workstation. Dynamic source code changes can be made directly on the aircraft, with the aircraft itself acting as the integrator and tester. This accommodates battlefield fixes allowing for quick response times and emergency repairs. The next option is the more formal option which is the Centralized Fleet System Support Facility (CFSSF) which is able to maintain any of the DoD's 3000 OFPs. In both options, the OFP Module Maintainer uses sophisticated engineering workstations which dynamically configure the required resources for a given change requirement; automatically documents any action; creates specialized testing for the changes; inquires the CFSSF archives for documentation and software related to the change requirement and displays this data on-line; and builds loadable OFPs.

Note that the originality of our ALL-ENGAGEMENT Fighter is that it has been emphasized in its design to be supportable and to be maintainable. This has not been the case with today's weapon systems whose supportability and maintainability have been developed in the middle of their life cycle and oftentimes after they have been transitioned to their support organizations.

IV. SUPPORTABLE AND MAINTAINABLE AVIONICS SOFTWARE TODAY

Today's generation of weapon systems uses avionics software which is supported and which is maintained. But is this software supportable? and is it maintainable? The answer to these questions depends on what is considered acceptable by the owner of the weapon system. If the owner finds acceptable 18 month (or longer) block cycle turn-over times and increasingly costly avionics integrated support environments, then the answer to both is yes. But if

the owner faces rapidly changing threat environments and shrinking budget constraints, the answer is clearly no.

Today's typical software support environment is the Avionics Integrated Support Facility (AISF) which is now being called the Centralized Software Support Activity (CSSA). A typical AISF can maintain 5 to 10 OFPs. Each OFP requires specialized support equipment, most of which is incompatible with the other OFPs' support equipment. Each OFP also requires specially trained people who are generally overloaded with the workload for that OFP [6].

The typical avionics software OFP should be reviewed as far as its maintainability. Of the 5 to 10 OFPs which are maintained in an AISF, each has its unique source code (usually assembler language) with unique support utilities such as compilers and linkers. In order to integrate two or more OFPs together, specialized software and hardware has to be developed allowing them to pass information. Other specialized processes include documentation, testing, configuration management, and training. Given our definition of maintainability and today's increasingly constrained AISFs, the probability is low that CSCIs can be repaired in a timely manner.

The situation facing today's avionics software maintenance organizations is that they will manage an increasing number of OFPs (20 and up) with the same or fewer resources. Given today's scenario of an increasing workload and with the same or fewer resources, previous design considerations for supportability and maintainability will be obsolete, if they aren't already [2],[3],[5],[6].

V. APPLYING SOFTWARE ENGINEERING DISCIPLINE TO OBTAIN SUPPORTABLE AND MAINTAINABLE AVIONICS SOFTWARE

As Software Engineering has been evolving, two camps have emerged as the champions for this new discipline. The first camp has been largely responsible for bringing Software Engineering where it is today. This is the Computer Science Camp. The Computer Science Community has effectively responded to the recent challenges of the "Age of Automation" which include an ever increasing appetite for software. The second camp is responsible for software as it is increasingly utilized in systems of every type. This camp is the Traditional Engineering Camp. Unfortunately, these camps have been less than cooperative. The Computer Science Camp desires Software Engineering to be a more creative and flexible discipline while the Traditional Engineering Camp would like to see Software Engineering stress standards which would enable software to be reliable, supportable, and maintainable for its complex systems. Objectively, Software Engineering

should best serve both camps. It should be creative and flexible enough to respond to new challenges while also serving the existing system's requirements. The existing software requirements of today's Avionics Systems demand increasingly complex and repeatable solutions for mature weapon systems platforms. These systems are not easily receptive to new technologies because they have been independently developed with limited concerns to their long term support requirements. To add to the complexity of these systems is their diverse support environment requirements seen in the increasing demand for Avionics Integrated Support Facilities.

VI. THE ENGINEERING DESIGN PROCESS

The design process is a systematic flow of developing an item of equipment that meets a predetermined product specification. The basic concept for any design process is a controlled approach to translating the product specification into a working system that fulfills the need of the user. While products of the design process can be anything imaginable, the basic design process, in theory, should be identical. The design process is a series of coordinated activities that form the basis for all activities necessary to produce the desired item. The design process must integrate the efforts of all engineering disciplines to ensure that the user's needs and requirements are met by the final product design [4].

In order for supportability and maintainability to be included in the design process, they must be carefully and completely specified along with the other constraints of the system.

VII. APPLYING THE PROCESS TO A SIMPLE SOFTWARE PROBLEM

Let's consider the case where a simple software program is the product of the design process. The series of coordinated activities are those specified in MIL-STD-2167A [1] which include system requirements analysis and design, software requirements analysis, preliminary design, detailed design, coding and unit test, component integration and testing, item testing, and system integration and testing. Let's say this software runs a digital clock and displays the clock's output. In application of our design process, the system requirements analysis and design tells us we need an automated time system. The software requirements analysis tells us we need software to run a digital clock. The preliminary design lays out the functionality of software, its testing needs, and the components it will require. The detailed design specifies the software to the lowest level and plans for low level testing as well. Component integration and

testing checks our low level design. Item testing checks out our complete software design. Finally we try our software out in its digital clock by performing a system integration and test by installing and running it [7]. Where in this process should we specify the desired level of supportability and maintainability? Since we want our software to meet a predetermined level of both supportability and maintainability, we must clearly lay out our requirements in the system requirements analysis and design.

VIII. APPLYING THE PROCESS TO A COMPLEX AVIONICS PROBLEM

Taking the same digital clock software, we will apply some requirements which will make it much more complicated. These include: (1) The software will be the basic timing logic for a modern fighter systems fire control computer operational flight program. (2) The software is to be modular and generic so that it can be used in multiple applications including other weapon systems. (3) The software will be field changeable in 24 hours or less. (4) The support environment for changing and testing this software will be totally commercial off-the-shelf (COTS) equipment. It would be foolish to handle these requirements anywhere but up front in our design process. It has been proven [7] that software support and maintenance account for 70% and more of the overall system life-cycle costs.

IX. SUMMARY

Increasingly, the cost and time spent in developing and operating weapon systems is found in those systems' software. The majority of the software cost and time spent is in its maintenance and support. Without carefully specifying the requirements for supportability and maintainability during system requirements analysis and design, the ability and affordability of operating a weapon system becomes prohibitive. The engineering design process already exists. This process, when properly specified, will accommodate building supportability and maintainability as weapon system requirements.

REFERENCES

- [1] DOD-STD-2167A, Military Standard Defense System Software Development, 29 February 1988.
- [2] Harris,R.L., "Laboratory Concepts in Avionics Software", IEEE/AIAA/NASA 9th Digital Avionics Systems Conference Proceedings, 15-18 October 1990, Virginia Beach VA.
- [3] Harris,R.L.,Jackson,O., "Software Engineering Tools For Avionics Embedded Computer Resources", IEEE/AIAA National Aerospace Electronics Conference Proceedings, 21-25 May 1990, Dayton OH.
- [4] Jones,J.V., *Engineering Design Reliability, Maintainability, Testability*, TAB Books Inc., Blue Ridge Summit, PA, 1988.
- [5] Morris,D.M., "Avionics Operational Flight Program Software Supportability", IEEE/AIAA/NASA 9th Digital Avionics Systems Conference Proceedings, 15-18 October 1990, Virginia Beach VA.
- [6] Satterthwaite,C.P., *Maintaining An Operational Flight Program (OFP)*, Air Force Technical Memorandum WL-TM-91-123, Wright-Patterson AFB, OH, 1991.
- [7] Computer Resource Acquisition Course, "Course Materials Volume I and II", System Acquisition School, 6575TH School Squadron, Air Force Systems Command, Brooks AFB, TX, December 1991.