ENSURING SOFTWARE SUPPORTABILITY DURING ACQUISITION:
AN AIR FORCE CASE STUDY

Captain Anthony C. Johndro, USAF
and
Daniel V. Ferens

Air Force Institute of Technology
Wright-Patterson AFB, Ohio 45433

ABSTRACT

Active consideration of software support during acquisition, or development, is tantamount to successful software support. A key objective of the Air Force's new Integrated Weapon System Management (IWSM) philosophy is to ensure software supportability by making the same organization responsible for both software development and software support.

A thesis effort performed at the Air Force Institute of Technology (AFIT) in 1993 assessed whether IWSM could meet this objective and what practices would result in improved software supportability. A literature review and survey of several programs designated as IWSM pilot programs concluded that the IWSM concept is indeed superior for software supportability, and that several practices used by the organizations surveyed can enhance supportability. Furthermore, the surveys and literature review suggested several additional practices that can further enhance the effectiveness of IWSM. A product of the AFIT study is a set of guidelines for ensuring supportability of software for IWSM systems. Although these guidelines were written specifically for IWSM, they will help all software managers more effectively consider software support during acquisition.

CURRENT STATE OF SOFTWARE SUPPORT

In 1992, United States companies spent over $30 billion supporting, or maintaining, software. This represents 60 to 80 percent of each company's software budget. It is estimated that this percentage will grow to 90 percent by 1995 (Sherer, 1992:70). The situation is no better for the Department of Defense (DoD). Figure 1 shows the past and predicted future DoD expenses for embedded software (DSMC, 1992:4-4). If 70 percent of these costs are for software support, DoD spent about $21.8 billion for software support in 1993.

Figure 1: DoD Software Expenses

One reason that software support costs are high is that software support is sometimes misunderstood. Some managers wrongly equate software support to mere "maintenance", or correcting software errors. However, error correction only accounts for about 17 percent of software support costs (Glass and Noiseux, 1981). Most software support activities may be categorized as either "adaptive" support which involves responding to changing data or processing requirements, or "perfective" support which involves enhancements to improve features such as performance, reliability and maintainability, or efficiency. Software maintainability (or better, supportability) is a characteristic of software which reflects the degree of effort required to...
perform the following tasks: correction of errors, addition of features, deletion of capabilities, and adaptation or modification (Sunday, 1989:50). Also, support procedures, as shown in Figure 2 (MIL-HDBK-347, 1990), can be quite elaborate, and reiterate developmental activities. Perhaps a good synonym for software support is "redevelopment" (Ferens, 1990:16-2).

Figure 2: Software Support Process

Perhaps the major cause of the high cost of software support, however, is that support usually has not been considered adequately during development (Ferens, 1990:18-2). Software developers are concerned with building new products, while supporters, or maintainers, are concerned with keeping software programs functional and current until they are no longer used. If the support organization is separate from the development organization, the supporters have probably had little interaction with the developers (Arnold, 1987:24). Consequently, the software developer is often more concerned with meeting contractual obligations for delivering a completed software product than with providing supportable software (Arnold, 1987:27). Software supporters, however, need a product which can be easily modified.

THE IWSM PROCESS

Until recently, the Air Force's organization exacerbate the problems discussed above. Software development was the responsibility of a System Program Office assigned to Air Force Systems Command (AFSC), while post-deployment software support (PDSS) was usually the responsibility of an Air Force Logistics Command (AFLC) organization. Unless the AFLC organization worked closely with the AFSC Program Office and could influence them to address PDSS, software support concerns were not sufficiently addressed during development.

Recently, however, AFSC and AFLC were combined into a single Command, Air Force Materiel Command (AFMC), and the IWSM philosophy was implemented (AFMCP 800-60, 1993). IWSM is "the AFMC management philosophy for acquiring, evolving, and sustaining products. It empowers a single manager with authority over the widest range of decisions and to satisfy customer requirements throughout the life cycle of a product" (AFMCP 800-60, 1993:2). This single manager, the System Program Director (SPD), is assigned to oversee a program during its entire life cycle. The SPD manages two groups, the Development System Manager (DSM) organization during development, and the Support System Manager (SSM) organization during PDSS. The DSM and SSM organizations, in turn, manage elements of their systems through Integrated Product Teams (IPTs), which are multidisciplinary teams responsible for developing or supporting their assigned system elements. The ultimate goal of IWSM is to provide a better product to the customer.

Under IWSM, the SPD now has an inherent interest in PDSS issues since he or she is now responsible for software support as well as software development. Because up to 80 percent of the software effort, time, and budget are spent in PDSS, the SPD is motivated to pay particular attention to techniques which improve software supportability (Edelstein and Mamone, 1992:82). Planning for PDSS is now a vital activity for the SPD during development.

THE IWSM PDSS STUDY

The objective of this study, conducted as an AFIT thesis effort (Butts and Johndro, 1993), was to determine how SSM organizations should plan for software supportability and involve software support personnel during development to improve software supportability for weapon systems and Command, Control, Computer, Communication, and Intelligence (C4I) systems developed under IWSM. To help meet this objective, a questionnaire was sent to fifteen of the twenty-one weapon system and C4I programs designated as IWSM pilot programs. The pilot programs were in two categories: operational
systems and systems still under development. The questionnaire was divided into five parts: IWSM program management structure, PDSS planning efforts, how PDSS plans are (or were) implemented during software development, transition from software development to PDSS, and operational software support.

Of the fifteen programs surveyed, eight programs responded. These programs included seven weapon system programs and one C4I program. The seven weapon system programs included two space programs, two electronic programs, and three aircraft programs. Figure 3 shows the percentages for each class of programs. Also, of the eight programs responding, four were operational and four were still under development. The results of the five part questionnaire are now discussed.

Figure 3: Responding Programs

Program Organizations Under IWSM

Figure 4 shows that, of the forty IPTs identified by the eight responding programs, twenty-nine, or 72 percent, have software as part of their IPT efforts. The eleven IPTs which don't have software involvement are in support areas such as site activation, contracting, and system testing. Therefore, most IPTs directly managing acquisition or PDSS for weapon and C4I systems can be expected to manage software.

Another noteworthy aspect of IWSM programs is the degree of coordination among IPTs. A coordinated team effort can result in better planning and conflict resolution for resources, budgets, schedules, and requirements. The coordinated team approach also facilitates necessary ongoing communication among the various IPTs. Figure 5 shows that five of

Figure 4: IPTs With Software Effort

the eight responding programs currently employ a coordinated management approach. Of the three programs that do not currently use a coordinated approach, one manager believed that, under IWSM, more coordination will take place in the future. It is also interesting to note that six of the eight responding programs include both development and PDSS personnel in their Computer Resource Working Group (CRWG), which is established early in a program to insure PDSS issues are addressed during the entire life cycle of a program.

Figure 5: Degree of Managerial Coordination

Planning for PDSS

All but one of the eight programs surveyed has a Computer Resources Life Cycle Management Plan (CRLCMP) to address PDSS planning. The CRLCMP, normally prepared
by the CRWG, addresses PDSS planning, including such items as number and type of personnel, equipment, and environment required for PDSS. The other program used a system maintenance plan for PDSS, such as an Integrated Logistics Support Plan. Three programs had both a system plan and a CRLCMP, but most software information was confined to the CRLCMP. The system maintenance plans usually were sparse in software information.

Including software product assurance personnel in PDSS planning, such as those specializing in software configuration management (SCM) and software quality assurance (SQA), can be highly beneficial. According to Dean, the bottom line purpose of SCM is to ensure continuing logistics supportability of systems (Dean, 1979:48), and similar statements can be made for SQA. Figure 6 shows that seven of the eight responding programs use SCM personnel in PDSS planning, and three programs used SQA personnel. Only one program did not use software product assurance personnel in PDSS planning.

Implementing PDSS Planning During Development

Aggressive participation by SSM software support personnel in contract monitoring activities, such as reviews and testing, can promote software supportability. Personnel can not only better assess contractor progress in the area of supportability, but can also learn more about the software they will later support. In six of the eight programs surveyed, SSM personnel participated in formal reviews and audits. SSM personnel in five programs were also able to perform code inspections, but did not often perform these inspections rigorously.

Figure 7 shows the level of participation in software testing by SSM personnel for the eight programs surveyed. In six of the eight programs, SSM personnel witnessed at least some of the tests performed by the contractor, and SSM personnel performed some hands-on testing in four of the programs. In two of the programs SSM personnel actually performed supportability testing.

Transitions From Software Development to PDSS

The information here applies only to the four programs surveyed which are currently in PDSS. One of the programs was divided into two separate programs, which represent different configurations of a system where PDSS plans differed: Figure 8 shows that, of the (now) five programs surveyed, three reported a routine, or easy transition, while the other two reported that transition was relatively difficult. Personnel in all three programs for which transition was routine received training, while training was not received for one of the two difficult programs. This may indicate that training can facilitate transitioning from development to support; however, it must also be noted that the three routine programs are entirely supported by the developing contractor, while both difficult programs have mixed government and contractor support. Furthermore, one of the difficult program's SPD is managing several dozen development and PDSS efforts concurrently, which complicates any transition.
Operational Software Support
Figure 9 shows that, for the five programs shown in Figure 8 above, three programs experienced routine PDSS, while the other two experienced difficult PDSS. Not surprisingly, the three programs for which PDSS is routine are the same three programs for which transition was routine as shown in Figure 8 above. Again the ease of PDSS may be correlated with training, but is also likely to be due to total contractor support. A computer resources manager for one of the difficult programs wrote an operating instruction in an attempt to minimize transition and PDSS difficulty for future programs. While the effects of this operating instruction are unknown at this time, it appears to be a step in the right direction.

Managers' Recommendations
The questionnaire also asked the managers to make recommendations for improving software supportability for current and future programs. The following are recommendations made by the managers:

1. Try to retain common software configurations among systems.
2. Require the software developer to use modular code.
3. Use standard software libraries to minimize recoding of common algorithms.
4. Institute common coding practices between the development and support organizations.
5. Ensure the required software documentation is deliverable under a development contract, including the documentation produced by subcontractors.
6. Educate contractors as to why the items above are needed by the government.

STUDY CONCLUSIONS
It is difficult to generalize results from a sample of only eight programs, and it is even more difficult to generalize results of only one C4I program to other C4I programs. Nevertheless, certain results of the study can provide useful guidance for weapon system, C4I, and other programs managed both within and outside the government. First, involvement by PDSS personnel throughout the development process is necessary for software supportability. PDSS personnel especially need to be involved in the planning process, where poorly-written CRLCMFs have been correlated with difficulty in PDSS. PDSS personnel should also actively participate in reviews and audits and in testing. Active participation by product assurance personnel such as those involved in SCM and SQA can also enhance supportability. Adequate documentation is necessary for supportability and, although this was not proven by the survey results, adequate training is also needed. Finally, coordinated management among IPTs or other program personnel can also enhance supportability.

As part of the study effort, a draft set of supportability guidelines for IWSM
programs was prepared. The complete
guidelines are documented in the thesis
(Butts and Johndro; 1993, Appendix B).
The table of contents, shown in Table 1,
highlights areas of concern for
considering software supportability
during development.

TABLE 1 Guidelines Contents

1. OVERVIEW
   1.1 Software Supportability
   Background
   1.2 Integrated Weapon System
   Management (IWSM) Background
   1.3 Purpose
   1.4 Overview

2. PLANNING FOR PDSS
   2.1 Introduction
   2.2 Planning for PDSS
      2.2.1 Management and
      Administration
      2.2.2 Software Engineering and
      Test
      2.2.3 Software Configuration
      Management
      2.2.4 Software Generation and
      Distribution
      2.2.5 Technical Documentation
      2.2.6 Deployment and
      Installation
      2.2.7 Quality Assurance
   2.3 PDSS Acquisition Requirements
      2.3.1 Software Environment
      Requirements
      2.3.2 Technical Data
      Requirements
      2.3.3 Software Quality
      Requirements
      2.3.4 Transition Requirements
   2.4 Planning Summary

3. ENSURING SUPPORTABILITY DURING
   DEVELOPMENT
   3.1 Introduction
   3.2 Higher Order Languages
   3.3 Reusable and Commercial Off-
   the-Shelf Software
   3.4 Modularity
   3.5 Reviews and Audits
   3.6 Standards
      3.6.1 Development Standards
      3.6.2 Coding Practices
      3.6.3 Maintainability Checklists
      3.6.4 Comments
   3.7 Code Walkthroughs and
   Inspections
   3.8 Testing
   3.9 Supportability Techniques in
   the Development Cycle
   3.10 Summary

REFERENCES

SUMMARY

Because of the high cost of software and
shrinking budgets, the Air Force and most
other agencies must focus on improving
supportability of the software they
develop or acquire. The Air Force has
taken a major step toward improving
supportability with IWSM concept.
However, IWSM must be implemented
properly to achieve the desired
objectives. The results of the 1993 AFIT
thesis study demonstrated ways in which
supportability can, and should be
considered during development in IWSM and
other programs. Implementing these
guidelines can greatly contribute to the
benefits of supportable software.

REFERENCES

1. AFMCP 800-60, IWSM Guidebook, Dayton,
2. Arnold, Robert S., "Improving
   Software Acquisition to Facilitate
   Software Maintenance", 1987 IEEE
   Conference on Software Maintenance
   Proceedings, Washington, DC, IEEE
3. Butts, Forrest F., III, and Anthony
   C. Johndro, Guidelines for Ensuring
   Software Supportability In Systems
   Developed Under the Integrated Weapon
   System Management Concept (APIT Thesis
   GSS/LAS/93D-1), Dayton, OH, Air Force
   Institute of Technology: 1993.
4. Dean, William A., "Why Worry About
   Software Configuration Management", De-
   fense Systems Management Review, Volume
   2, Number 3, Summer, 1979.
5. Defense Systems Management College,
   Mission Critical Computer Resources Man-
6. Edelstein, D. Vera, and Salvatore
   Mamone, "A Standard For Software Mainte-
7. Ferens, Daniel V., Defense System
   Software Project Management, Dayton, OH,
8. Glass, Robert L., and Ronald A. Nois-
   eux, Software Maintenance Guidebook, En-
9. MIL-HDBK-347, Post-Deployment