MIL-HDBK-347
22 MAY 90

MILITARY HANDBOOK

MISSION-CRITICAL COMPUTER RESOURCES SOFTWARE SUPPORT

AMSC (N/A)

AREA MCCR

FOREWORD

1. This military handbook is approved for use by all Departments and Agencies of the Department of Defense.

2. Beneficial comments (recommendations, additions, deletions) and any pertinent data which may be of use in improving this document should be addressed to: HQ AFSC/ENR, Andrews AFB, Maryland 20334

3. The Department of Defense is committed to maintaining the operational viability of our weapons systems and at the same time keeping support costs under control. Historically, software support has been a major contributor to overall life cycle costs. This standardization document, short titled "The Software Support Handbook", is an attempt to improve this situation through standardization of related concepts and management procedures. The Software Support Handbook was developed by the Department of Defense with the assistance of the military departments, federal agencies, and industry. It is a cornerstone document, covering pre- and post-deployment software support operations. In addition to a standard definition and detailed description of software support and the post-deployment software support process, this handbook provides guidance in the areas of post-deployment software support and transition planning, post-deployment software support resource analysis, software support activity resource requirements planning, and post-deployment software support concept alternatives.

4. Each Service or agency needs to consult applicable software standards and directives for current DoD policy and reporting formats.

CONTENTS

CONTENTS

CONTENTS

1. SCOPE

1.1 Scope. This handbook presents software support concepts, procedures, and guidance to all managers responsible for Mission-Critical Computer Resources (MCCR) development or support.

1.2 Applicability. This handbook covers software support activities and requirements throughout the system life cycle and should be selectively applied within the defense community to achieve consistency with individual Service and program requirements. It is intended to be used by Department of Defense (DoD) and commercial agencies involved in any software support activity.

2. APPLICABLE DOCUMENTS

2.1 Government Documents.

2.1.1 Specifications, standards, and handbooks. The following specifications, standards, and handbooks form a part of this document to the extent specified herein. Unless otherwise specified, the issues of these documents are those listed in the current issue of the Department of Defense Index of Specifications and Standards.

STANDARDS

MILITARY

MIL-STD-480	-	Configuration Control, Engineering Changes, Deviations, and Waivers
MIL-STD-483	-	Configuration Management Practices for Systems, Equipment, Munitions, and Computer Programs
MIL-STD-490	-	Specification Practices
MIL-STD-881	-	Work Breakdown Structures for Defense Material Items
MIL-STD-1521	-	Technical Reviews and Audits for Systems, Equipments, and Computer Software
DoD-STD-2167	-	Defense System Software Development
DoD-STD-2168	-	Defense System Software Quality Program
DI-E-7142	-	Software Support Transition Plan

(Unless otherwise indicated, copies of federal and military specifications, standards, and handbooks are available from the Naval Publications and Forms Center, (ATTN:NPODS), 5801 Tabor Avenue, Philadelphia, PA 19120-5099.)

2.1.2 Other Government documents, drawings, and publications. The following other Government documents, drawings, and publications form a part of this document to the extent specified herein.

DoD Directive 5000.1 - Major and Non-Major Defense Acquisition Programs

DoD Directive 5000.3 - Test and Evaluation

DoD Directive 5000.29 - Management of Computer Resources in Major and Non-Major Defense Systems

DoD Directive 5000.39 - Acquisition and Management of Integrated Logistics Support for Systems and Equipment

DoD Instruction 5000.2 - Defense Acquisition Program Procedures

(Copies of DoD directives are available from the Commanding Officer, Naval Publications and Forms Center, 5801 Tabor Avenue, Philadelphia, PA 19120-5099.)

3. DEFINITIONS

3.1 Acronyms used in this handbook. The acronyms used in this handbook are defined as follows:

a. ABL - Allocated Baseline

b. CCB - Configuration Control Board

c. CI - Configuration Item

d. CM - Configuration Management

e. CRISD - Computer Resources Integrated Support Document

f. CRLCMP - Computer Resources Life Cycle Management Plan

g. CSC - Computer Software Component

h. CSCI - Computer Software Configuration Item

i. CSU - Computer Software Unit

j. DoD - Department of Defense

k. ECP - Engineering Change Proposal

l. FBL - Functional Baseline

m. FCA - Functional Configuration Audit

n. FQT - Formal Qualification Test

o. ILS - Integrated Logistics Support

p. ILSP - Integrated Logistics Support Plan

q. IRS - Interface Requirements Specification

r. IV&V - Independent Verification and Validation

s. LCC - Life Cycle Cost

t. LSA - Logistics Support Analysis

u. MCCR - Mission-Critical Computer Resources

v. OT&E - Operational Test and Evaluation

w. PBL - Product Baseline

x. PCA - Physical Configuration Audit

y. PDSS - Post-Deployment Software Support

z. SCN - Specification Change Notice

aa. SDD - Software Design Document

bb. SDF - Software Development File

cc. SEE - Software Engineering Environment

dd. SLCC - Software Life Cycle Cost

ee. SLCSC - Software Life Cycle Support Cost

ff. SPS - Software Product Specification

gg. SRS - Software Requirements Specification

hh. SSA - Software Support Activity

ii. STD - Software Test Description

jj. STE - Software Test Environment

kk. STP - Software Test Plan

ll. STR - Software Test Report

3.2 Baseline. MIL-STD-480B: "A configuration identification document or a set of such documents formally designated by the Government at a specific time during a CI's life cycle. Baselines, plus approved changes from those baselines, constitute the current approved configuration identification. For configuration management purposes there are three baselines, which are established sequentially, as follows:

a. Functional baseline (FBL). The initially approved documentation describing a system's or item's functional characteristics and the verification required to demonstrate the achievement of those specified functional characteristics.

b. Allocated baseline (ABL). The initially approved documentation describing an item's functional and interface

characteristics that are allocated from those of a higher level CI, interface requirements with interfacing configuration items, additional design constraints and the verification required to demonstrate the achievement of those specified functional and interface characteristics.

c. Product baseline (PBL). The initially approved documentation describing all of the necessary functional and physical characteristics of the CI, any required joint and combined operations interoperability characteristics of a CI (including a comprehensive summary of the other service(s) and allied interfacing CI's or systems and equipments), and the selected functional and physical characteristics designated for production acceptance testing and tests necessary for support of the CI."

3.3 Computer resources. DoD-STD-2167A: "The totality of computer hardware, software, personnel, documentation, supplies, and services applied to a given effort."

3.4 Computer software (or software). DoD-STD-2167A: "A combination of associated computer instructions and computer data definitions required to enable the computer hardware to perform computational or control functions."

3.5 Computer software configuration item (CSCI). DoD-STD-2167A: "A configuration item for computer software." MIL-STD-480B: "Configuration item (CI). An aggregation of hardware, firmware, software, or any of its discrete portions, which satisfies an end use function and is designated for configuration management...."

3.6 Computer software documentation. DoD-STD-2167A: "Technical data or information, including computer listings and printouts, which documents the requirements, design, or details of computer software, explains the capabilities and limitations of the software, or provides operating instructions for using or supporting computer software during the software's operational life."

3.7 Configuration. MIL-STD-480B: "The functional and physical characteristics of hardware, firmware, software or a combination thereof as set forth in technical documentation and achieved in a product." The physical characteristic of software is the machine language image and is described in the Software Product Specification (SPS). Any change to the SPS, which includes source code listings and identification of the compiler/assembler used to translate the source code, is a change to the product

configuration identification and to the physical characteristic of the CSCI.

3.8 CSCI development process. As described in DoD-STD-2167A, the CSCI development process consists of six activities: software requirements analysis, preliminary design, detailed design, coding and computer software unit (CSU) testing, computer software component integration and testing, and CSCI testing.

3.9 Firmware. DoD-STD-2167A: "The combination of a hardware device and computer instructions or computer data that reside as read-only software on the hardware device. The software cannot be readily modified under program control."

3.10 Independent verification and validation (IV&V). DoD-STD-2167A: "Verification and validation performed by a contractor or Government agency that is not responsible for developing the product or performing the activity being evaluated. IV&V is an activity that is conducted separately from the software development activities governed by this standard [i.e., DoD-STD-2167A]."

3.11 Mission-Critical Computer Resources (MCCR). Title 10, United States Code, Section 2315 (Public Law 97-86) provides legal requirements for the acquisition of computer resources within DoD. For the purposes of this document MCCR is defined as follows:

A. Mission-Critical Computer Resources are elements of computer hardware, software, or services whose function, operation or use:

1. involves intelligence activities;

2. involves cryptological activities related to national security;

3. involves the command and control of military forces;

4. involves equipment which is an integral part of a weapon or weapons system; or

5. is critical to the direct fulfillment of military or intelligence missions, provided that it does not include automatic data processing equipment used for routine administrative and business applications such as payroll, finance, logistics, and personnel management.

B. Computer resources whose function, purpose, or use are critical to the direct fulfillment of military or intelligence missions include, but not be limited to:

1. Warning, surveillance, reconnaissance, and electronic warfare systems;

2. Mission-support systems deployed in combat environments;

3. Classified systems and programs;

4. Strategic and tactical military communications systems;

5. Satellite systems supporting strategic or tactical military missions;

6. Environment monitoring and prediction systems directly supporting military missions (e.g. weather, and oceanographic);

7. Locating, positioning, mapping, charting and geodesy systems directly supporting military missions;

8. Maintenance systems which provide direct support to weapon systems and software support facilities;

9. Systems used internally within the Department of Defense for classified analyses, research and development in direct support of military or intelligence missions.

C. Computer resources used primarily for routine administrative and business applications such as payroll, finance, logistics, and personnel management shall not be considered military or intelligence mission-critical.

3.12 Operational version. A software version which has been approved for release to the using community.

3.13 Post-deployment software support (PDSS). Those software support activities that occur during the full-rate production and initial deployment and operations support phases of the acquisition process.

3.14 PDSS environment. The PDSS environment includes the software engineering environment, software test environment, and other equipment, material and documentation, including data rights, necessary to provide PDSS for a designated MCCR in accordance with the integrated logistics support concept.

3.15 Pre-deployment software support. Those software support activities that occur before initial deployment and operations support.

3.16 Release. DoD-STD-2167A: "A configuration management action whereby a particular version of software is made available for a specific purpose (e.g., released for test)."

3.17 Software change categories. All software changes can be placed into one of the following categories:

3.17.1 Software correction. A software correction is a software change implemented to correct a fault in the software.

3.17.2 Software enhancement. A software enhancement is any software change which is not a software correction. There are two types of software enhancements:

3.17.2.1 Adaptive. Adaptive enhancements are software changes necessary to accommodate change(s) in the operational or hardware environment. Software adaptations include changes to implement new system interface requirements, new system requirements (i.e., changes to the system requirements specification), or new hardware requirements.

3.17.2.2 Perfective. Perfective enhancements are software changes which improve software performance (e.g., man-machine interface enhancements), maintainability or other software attributes. A perfective change does not implement a new system requirement.

3.18 Software development process. As described in DoD-STD-2167A, the software development process includes the following major activities: (Note: The CSCI development process is a subset of the software development process.)

a. System Requirements Analysis/Design
b. Software Requirements Analysis
c. Preliminary Design
d. Detailed Design
e. Coding and CSU Testing
f. Computer Software Component (CSC) Integration and Testing

g. CSCI Testing
h. System Integration and Testing

3.19 Software engineering environment (SEE). DOD-STD-2167A: "The set of automated tools, firmware devices, and hardware necessary to perform the software engineering effort. The automated tools may include but are not limited to compilers, assemblers, linkers, loaders, operating system, debuggers, simulators, emulators, test tools, documentation tools, and data base management system(s)."

3.20 Software support. DoD-STD-2167A: "The sum of all activities that take place to ensure that implemented and fielded software continues to fully support the operational mission of the software." Software support includes pre-deployment software support and post-deployment software support.

3.21 Software Support Activity (SSA). The DoD or military Service organization responsible for the software support of designated MCCR.

3.22 Software test environment (STE). DoD-STD-2167A: "A set of automated tools, firmware devices, and hardware necessary to test software. The automated tools may include but are not limited to test tools such as simulation software, code analyzers, etc. and may also include those tools used in the software engineering environment."

3.23 Software transfer. That point during the system acquisition process when the SSA assumes responsibility for PDSS. Software transfer is the final step in software transition.

3.24 Software transition. A controlled and coordinated sequence of actions wherein software development passes from the organization performing initial software development to the organization performing PDSS. Software transition also involves the SSA moving from pre- to post-deployment software support activities.

3.25 Validation. DoD-STD-2167A: "The process of evaluating software to determine compliance with specified requirements."

3.26 Verification. DoD-STD-2167A: "The process of evaluating the products of a given software development activity to determine correctness and consistency with respect to the products and standards provided as input to that activity."

3.27 Version. DoD-STD-2167A: "An identified and documented body of software. Modifications to a version of software (resulting

in a new version) require configuration management actions by either the contractor, the contracting agency, or both." During PDSS the SSA is responsible for maintaining the configuration identification of all versions.

4. SOFTWARE SUPPORT CONCEPTS

4.1 Purpose. The purpose of this section is to introduce the principles of software support, to discuss the evolution of software support, to discuss the relationship between integrated logistics support (ILS) and software support, to describe the roles of the Program Manager and the SSA, and to examine the SSA charter.

4.2 Introduction. Software support includes all activities necessary to ensure implemented and fielded software continues to fully support its operational mission. Implicit is the notion that software support activities occur throughout the acquisition process. For discussion and planning purposes, software support can be conveniently separated into two stages, pre-deployment software support and post-deployment software support. Generally, pre-deployment software support activities occur during concept exploration, demonstration and validation, and the full scale development phases of the acquisition process, while PDSS usually coincides with the production/deployment and support phases. Although the SSA is a key participant throughout the acquisition process, it is important to recognize that the activities and responsibilities of the SSA vary considerably between the pre- and post-deployment software support stages.

4.3 Software support principles. It is important to establish the principles upon which this document is based. They are:

a. Software support activities occur throughout the system acquisition process.

b. The evolution of software is a continuum of successive software development cycles beginning with initial software development and continuing with one or more follow-on software development cycles.

c. Software should continue to evolve as long as essential requirements or cost-effective benefits can be derived through software change.

d. The SSA is the Government's agency responsible for providing PDSS requirements to the Program Manager and for cost effective PDSS operations following transition.

e. Critical MCCR activities are performed by Government organizations to ensure

accountability, responsiveness, control, continuity and compliance. Critical activities include the control and identification of Government controlled software baselines; defining and evaluating software quality requirements; system level testing; managing and planning for PDSS.

f. Direct communication between the using community and the SSA is important to help identify and isolate software problems, and to foster attitudes of support and cooperation.

4.4 Evolution of software support. Software support, and hence the role of the SSA, evolves as acquisition progresses. The SSA's capability to conduct PDSS and the eventual cost of PDSS closely depend on what occurs, or what fails to occur, during the pre-deployment stage. In general, during the pre-deployment software support stage the focus of software support should be to ensure software supportability and to plan for cost effective PDSS. During the PDSS stage, the focus of software support should be to ensure continued supportability, to manage PDSS, and to conduct efficient PDSS operations.

4.4.1 Software support and the acquisition process. In accordance with DoD Directive 5000.1, the acquisition process is divided into five phases: concept exploration/definition; concept demonstration/validation; full-scale development; full-rate production and initial deployment; and operations support. The chronological relationship between the acquisition process and the evolution of software support can vary. Figure 1 identifies typical ILS and software support activities within the context of the acquisition process. Early in the acquisition process, program direction defines the responsibilities of the Program Manager and other acquisition participants. (Note: The term "Program Manager", as defined in DoD Directive 5000.1, is used throughout this handbook to designate the individual vested with the authority and responsibility for program direction and execution.) Working within the Planning, Programming and Budgeting System, the Program Manager provides direction and funds for development, production, and support of the system.

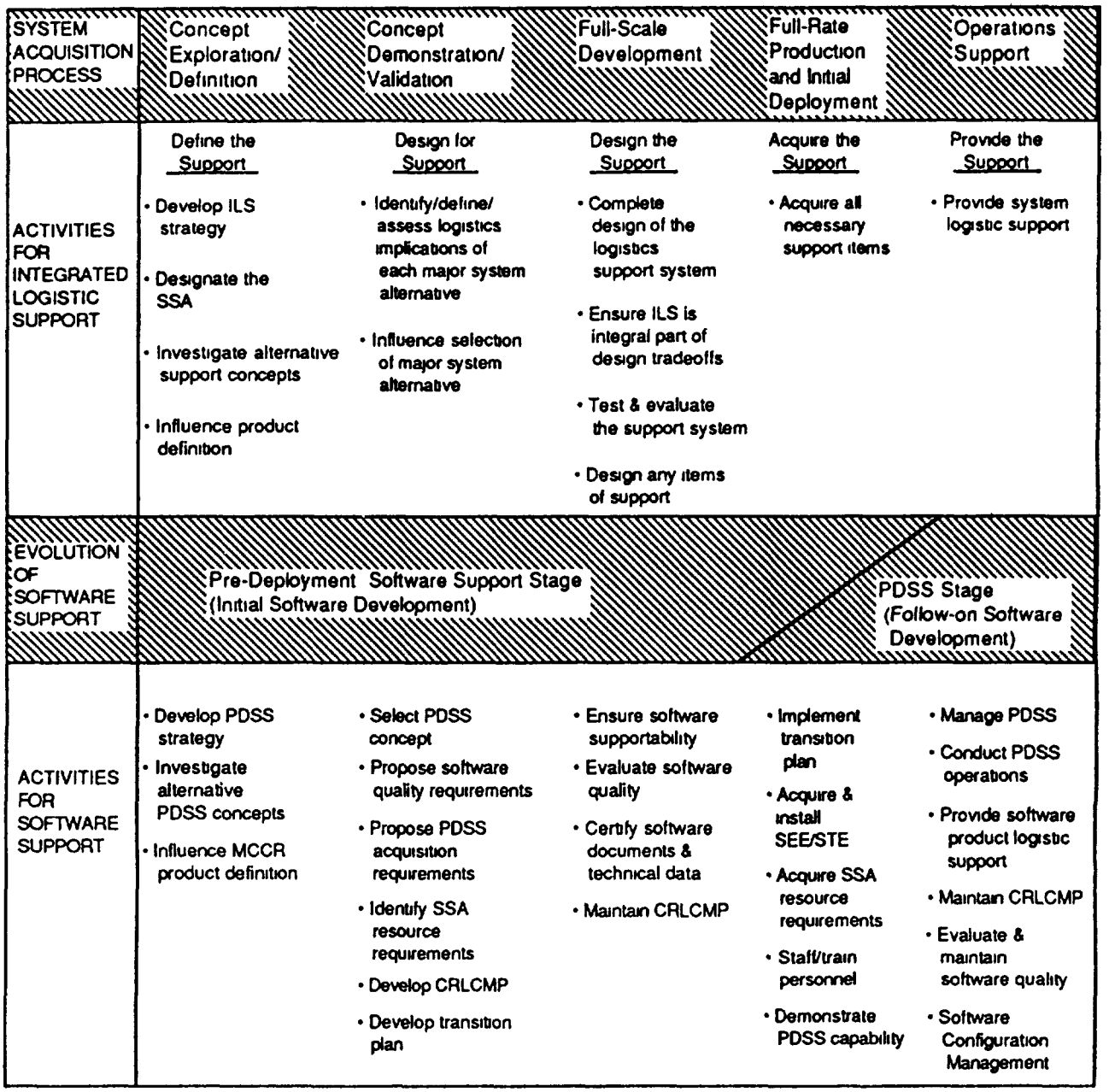

FIGURE 1. Software support activities and the acquisition process.

4.4.2 Integrated logistics support. Computer resources support is one of the ten integrated logistics support elements identified in DoD Directive 5000.39. Integrated logistics

support planning is based on the approved system support concept. The Integrated Logistics Support Plan (ILSP), which is the product of the logistics support analysis (LSA), reflects the system support concept. The ILSP should also incorporate requirements for each of the ILS elements identified in figure 2. Computer resources support embraces support requirements for facilities, hardware, software, documentation, and personnel. The PDSS concept, PDSS acquisition requirements, SSA resource requirements, and cost estimates are essential components of ILS planning. Consequently, the SSA plays a crucial role in integrated logistics support planning.

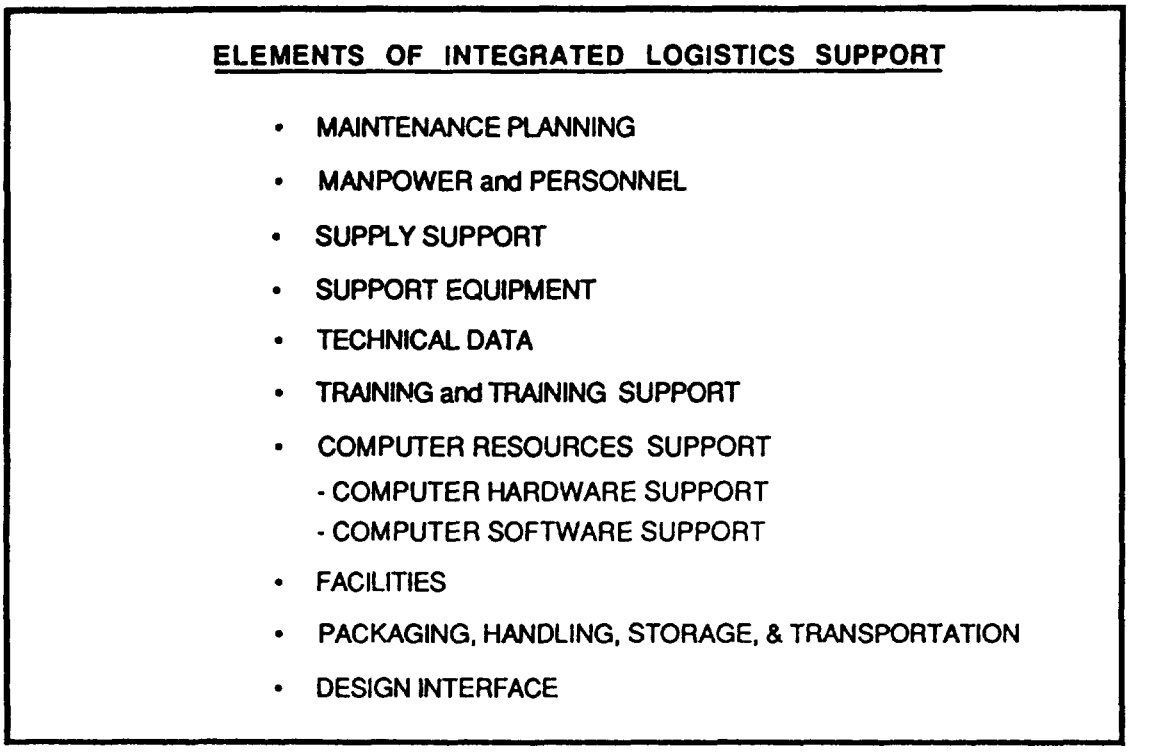

FIGURE 2. Elements of integrated logistics support.

4.5 Program Manager's role. During acquisition, the Program Manager is responsible for acquiring a supportable system which satisfies the fundamental needs of the using and support communities. This encompasses a broad range of responsibilities to include:

a. Program direction and system concept definitions,

b. Budget and schedule planning,

c. Requirements validation,

d. Specification verification,

e. Quality assurance,

f. Configuration management,

g. Preparing acquisition plans, requests for proposal, statements of work, contract data requirements lists,

h. Software IV&V,

i. Security requirements,

j. Testing and evaluation, and

k. System support, ILS analysis and planning.

System support, an explicit responsibility of the Program Manager, involves establishing and managing an adequately funded ILS program. This entails incorporating PDSS acquisition requirements in the software development contract, early identification of SSA resource requirements, acquiring necessary resources, and finally, providing support. The Program Manager is often assisted in these duties by an ILS Manager, who oversees ILS planning. Support plans and requirements should be documented in the ILSP.

4.6 SSA's role. The role of the SSA in support of the Program Manager changes as acquisition progresses and software support evolves. For example, during pre-deployment major concerns of the SSA include planning, identifying requirements, software quality, software supportability, and transition. During post-deployment, the SSA's primary concern is implementing approved software changes (i.e., conducting PDSS operations in accordance with the approved PDSS concept) and maintaining the quality of software and technical data at acceptable costs. Specific responsibilities of the SSA should be defined in the SSA's charter.

4.7 SSA's charter. After designation of the SSA, a charter should be created and approved which defines specific responsibilities and authority of the SSA; identifies SSA taskings; and discusses organizational relationships (channels for management, support, and technical direction). Frequently, organizational relationships, responsibilities, lines of authority, and support channels are tailored to conform to a particular PDSS concept. As suggested by figure 3, this can

create a complex, dynamic, and, sometimes, confusing situation. In such environments, the SSA's charter is a vital management document which should be promulgated early and updated as necessary. This is true especially if organizational relationships are not routine (e.g., in a joint Service PDSS program). Appendix A (Sample SSA's Charter) provides a sample charter outline. Generally, the Program Manager directs and funds the SSA to implement program requirements contained in the charter in accordance with each Service's policies and procedures.

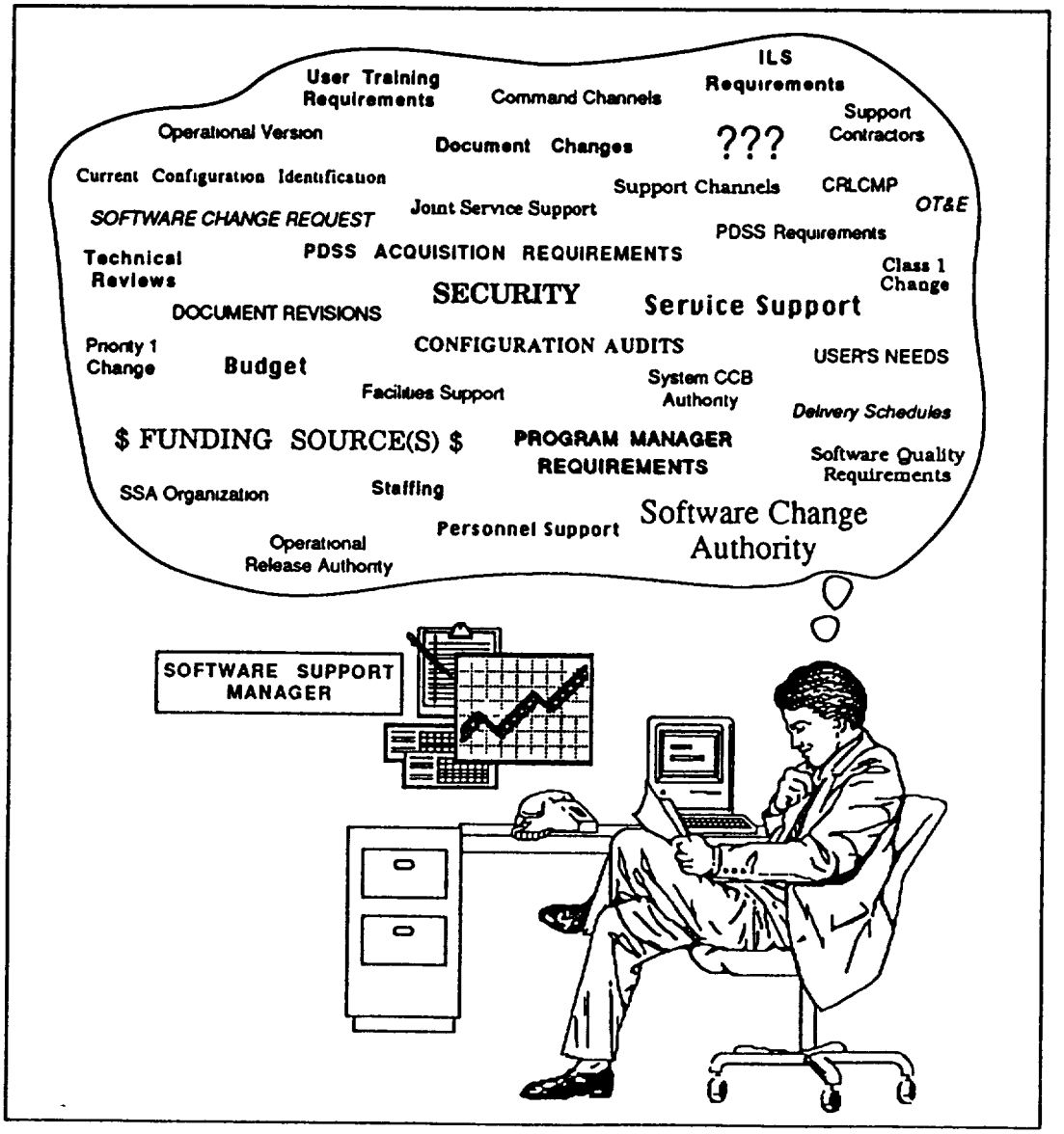

FIGURE 3. The SSA's environment.

5. PRE-DEPLOYMENT SOFTWARE SUPPORT

5.1 Purpose. The purpose of this section is to discuss the pre-deployment software support responsibilities of the Program Manager and the SSA's role in the software acquisition and ILS effort during the pre-deployment stage.

5.2 Introduction. It is in the Program Manager's interest that the SSA participate early in the acquisition process. Software acquisition issues are important components of the Program Manager's responsibilities, and acquisition managers at all levels should understand that PDSS cost, and as a result software life cycle cost, is largely determined during acquisition. In addition to providing cost saving PDSS concept alternatives, the SSA supports the Program Manager by performing or actively participating in many other important activities. The term "developing agency" is used throughout this section to refer to the organization, Government or commercial, performing initial software development.

5.3 Pre-deployment software support activities. Pre-deployment software support activities can be grouped into five general areas:

a. Plan for PDSS,

b. Identify PDSS acquisition requirements,

c. Ensure software supportability,

d. Assure software quality, and

e. Develop and implement transition plan.

5.3.1 Plan for PDSS. One of the most important ILS activities to reduce software life cycle cost and manage the risk associated with PDSS is planning. Early designation and participation by the SSA is vital for effective PDSS planning. All PDSS plans should be reviewed periodically to ensure they remain current and that assumptions are still valid. PDSS planning, which includes PDSS concept and SSA resource requirements planning, should be reflected in the Computer Resources Life Cycle Management Plan (CRLCMP) and the ILSP. PDSS requirements also form an integral aspect of system supportability test requirements that are documented in the Test and Evaluation Master Plan.

5.3.1.1 PDSS concept planning. The PDSS concept is the first step in the PDSS planning sequence and the basis for all subsequent PDSS planning. The PDSS concept, which is derived

from the system support concept, describes how and to what extent the software will be supported during deployment. If PDSS planning begins early, the set of feasible PDSS concept alternatives can be large. If PDSS planning begins late, the set of feasible alternatives can be very limited. As a general rule, the later PDSS concept planning begins, the more constrained the environment and more costly the implementation. In the absence of an early PDSS concept, decisions are made which limit future choices and affect future PDSS costs. Therefore, it is important that concept planning begin early. The PDSS concept should become part of the ILS concept. Since all subsequent PDSS planning is based on the PDSS concept, it should remain relatively stable. Later in the acquisition process, changes to the approved PDSS concept can have costly consequences. Appendix B (The PDSS Concept) provides a more detailed discussion of PDSS concept planning.

5.3.1.2 SSA resource requirements planning. The purpose of SSA resource requirements planning is to identify the resources required to implement the approved PDSS concept. In order to accurately project requirements, planners must consider the combined needs of all PDSS activities to be performed. These activities should be identified and described as part of the PDSS concept. Appendix C (PDSS Resource Analysis) provides a more detailed discussion of SSA resource requirements planning and analysis.

5.3.1.3 Computer Resources Life Cycle Management Plan. The CRLCMP is the primary document for computer resources planning and management throughout the acquisition process. The CRLCMP should complement the ILSP, and be updated as often as necessary to reflect current plans and decisions. In addition to identifying the PDSS concept and SSA resource requirements, the CRLCMP can contain the information listed in Appendix D (CRLCMP Topics). The SSA should actively participate in CRLCMP development, but the Program Manager is usually responsible for obtaining final CRLCMP approval. Appendix D (CRLCMP Topics) contains possible subject areas covered in most CRLCMP's. Appendix D (CRLCMP Topics) is not a statement of DoD policy or format. Consult latest CRLCMP format used by your organization.

5.3.2 Identify PDSS acquisition requirements. PDSS acquisition requirements are contractual requirements imposed upon the software developer which facilitate PDSS. (Notice the distinction between "PDSS acquisition requirements" and "SSA resource requirements". The former refers to contractual requirements, and the latter refers to resource requirements.) The SSA can suggest alternatives or options to the Program Manager that can in some way reduce life cycle cost (LCC), ease transition, or

help ensure software supportability and quality. Often, when such PDSS acquisition requirements are included in acquisition contracts, savings can result or risk can be reduced. Since many SSAs support multiple programs, the SSA can often provide a unique cross-program perspective. In addition, SSA planners are familiar with local standards and resources (current and planned) which might lead to cost saving alternatives by reducing complexity or using assets already owned. Such cost saving opportunities should be considered by the Program Manager. In general, PDSS acquisition requirements should not impede the insertion of new or improved technology. However, technology improvements need to be considered in light of LCC and technical risk. PDSS acquisition requirements should be considered in the areas of software environment (i.e., SEE or STE) requirements, technical data requirements, software quality requirements, evaluation requirements, and transition requirements.

5.3.2.1 Software environment requirements. Specific hardware and software requirements can be contractually specified in order to:

a. Promote a uniform PDSS environment at the SSA (e.g., automated tools, network protocols, document or publication standards, configuration management forms/documents or data elements);

b. Use existing Government resources as components of the SEE/STE during initial software development and PDSS;

c. Provide hardware redundancy (i.e., backup systems) at the SSA;

d. Reduce complexity at the SSA by limiting the number of different software environments and software development methodologies;

e. Reduce training requirements; or

f. Implement a consolidated or centralized PDSS concept.

5.3.2.2 Technical data requirements. The SSA should identify clear and complete requirements for necessary technical data (i.e., computer software documentation). While development cost can be reduced by attempting to minimize data requirements, PDSS risk and cost can be significantly increased by not receiving adequate documentation. On the other hand, since document support is a major (and sometimes overlooked) element of PDSS

cost, the quantity, completeness, correctness and quality of delivered technical data is a vital concern to the SSA and the Program Manager. Therefore, a balance must be struck between the cost of maintaining technical data and the need for a comprehensive description of the software product.

5.3.2.3 Software quality requirements. Software quality requirements, evaluation methodology, and acceptance criteria are important to both the Program Manager and the SSA. The SSA should participate in the program office's efforts: to identify and define software quality requirements; to develop and maintain standard techniques for software quality evaluation; and to establish acceptance criteria.

5.3.2.4 Evaluation requirements. In addition to proposing software quality requirements, the SSA can actively participate in the program office's evaluation of software plans, products, and activities as provided in DoD-STD-2168 (Defense System Software Quality Program). The SSA should participate in the evaluation of the developing agency's plans and procedures for software management, software engineering, software qualification, software configuration management, software corrective action, documentation and media distribution, storage, handling, and delivery. Also during initial software development, the SSA can be the Government's IV&V agent since the SSA is usually not directly involved with software development at this point.

5.3.2.5 Transition requirements. The developing agency plays a major role in the transition effort. If the software is developed under contract, specific transition requirements to be imposed on the developing contractor must be identified in the contract. Transition requirements can include requirements to facilitate hardware installation, network installation, software installation and test, security, hardware maintenance, training, transfer of software licensing agreements, and the transition of software configuration management. Transition from initial software development to follow-on software development or PDSS must be planned, carefully controlled, and coordinated.

5.3.3 Ensure software supportability. Specifying software support requirements alone does not ensure supportability. The SSA has a vested interest in the ultimate supportability of the software product; therefore, during acquisition, it is appropriate for the SSA to actively participate in the evaluation of software product supportability. The supportability question, which is often very subjective, should take into consideration characteristics of the software product, the software environments, and the status of SSA resource requirements.

Software characteristics, particularly correctness, testability, and flexibility, are important concerns since they effect software volatility and the effort required to isolate and correct faults. Depending on contract requirements, opportunities for the SSA to evaluate software supportability include:

a. Software product and activity reviews,

b. Software technical reviews and audits,

c. Software IV&V activities,

d. Formal qualification testing, and

e. Government software acceptance evaluations.

5.3.4 Assure software quality. Specifying software quality and software quality program requirements alone does not assure software quality. After requirements have been identified and specified, the SSA may be actively involved in assuring that software quality and program requirements are achieved and correctly implemented. This can include authenticating specifications, verifying requirements, and evaluating proposed software quality plans, records, and activities.

5.3.5 Develop and implement transition plan. Software transition is critical and should not be left to chance. Too often transition is attempted without benefit of a complete and well developed transition plan that includes a detailed checklist. Appendix E (Software Transition) provides a sample transition checklist. At best, without proper planning, transition is inefficient. At worst, the SSA may not be able to implement the PDSS concept. Software baselines are usually accepted by the Government after successful completion of software development milestones, such as technical reviews or configuration audits. The configuration of each accepted baseline is controlled by a Government configuration control board (CCB). The SSA should be an active member of the CCB during initial software development. Complete transition plans, which should be documented in the CRLCMP, must consider software development transition and SSA transition. If both components of transition are not planned and coordinated with adequate management controls in place, the transition stage can become a period of disjointed actions which result in frustration and little progress.

5.3.5.1 Software development transition. Software development transition includes all activities to transition software

development capabilities from the developing agency to the SSA. It involves the transfer of computer hardware, software, data and experience. If contractor support is anticipated during transition, the acquisition contract should identify these requirements. PDSS acquisition requirements can include training, relocation or installation of Government furnished equipment, preparing configuration status accounting reports or contractor support during production/deployment. The SSA should work closely with the program office to coordinate transition milestones and schedules, and to integrate Computer Resources Integrated Support Document (CRISD), CRLCMP, and transition plan requirements.

5.3.5.2 SSA transition. SSA transition includes all activities necessary for the SSA to implement the PDSS concept once the PDSS environment is in place. While the focus of software development transition is the SEE and STE, the focus of SSA transition is PDSS operations. SSA transition includes activities intended to provide a smooth and effective transition from pre-deployment software support operations to PDSS operations. This is a major undertaking and needs to be carefully planned. During SSA transition the entire operational context of the SSA changes from planning and evaluation activities to full-blown PDSS operations. Important SSA transition activities should include:

a. Training;

b. Staffing;

c. Turnover, installation, checkout, and integration of any hardware or software received from sources other than the developing agency;

d. Implementation of all required PDSS activities and capabilities (e.g., problem replication and fault isolation, corrective action, software generation, integration and test, support systems, document production);

e. Approval and implementation of applicable software management plans such as a configuration management plan and a software quality program plan;

f. Verification that transition milestones have been correctly completed and that all necessary resources are available;

g. Integration of all PDSS activities into a cohesive PDSS process;

h. Reporting software transfer; and

i. Determination that security requirements have been satisfied.

The SSA is now ready to assume responsibility for PDSS.

6. POST-DEPLOYMENT SOFTWARE SUPPORT

6.1 Purpose. The purpose of this section is to describe the PDSS process. The general PDSS process is first introduced to provide a high level perspective of PDSS. Afterwards, a detailed PDSS process model is presented which describes the process in more detail and identifies specific activities often associated with each phase of PDSS.

6.2 General PDSS process. The general PDSS process, shown in figure 4, is an abstraction consisting of four phases: initial analysis, software (CSCI) development, system integration and testing, and product logistics. Every PDSS process should include these four phases.

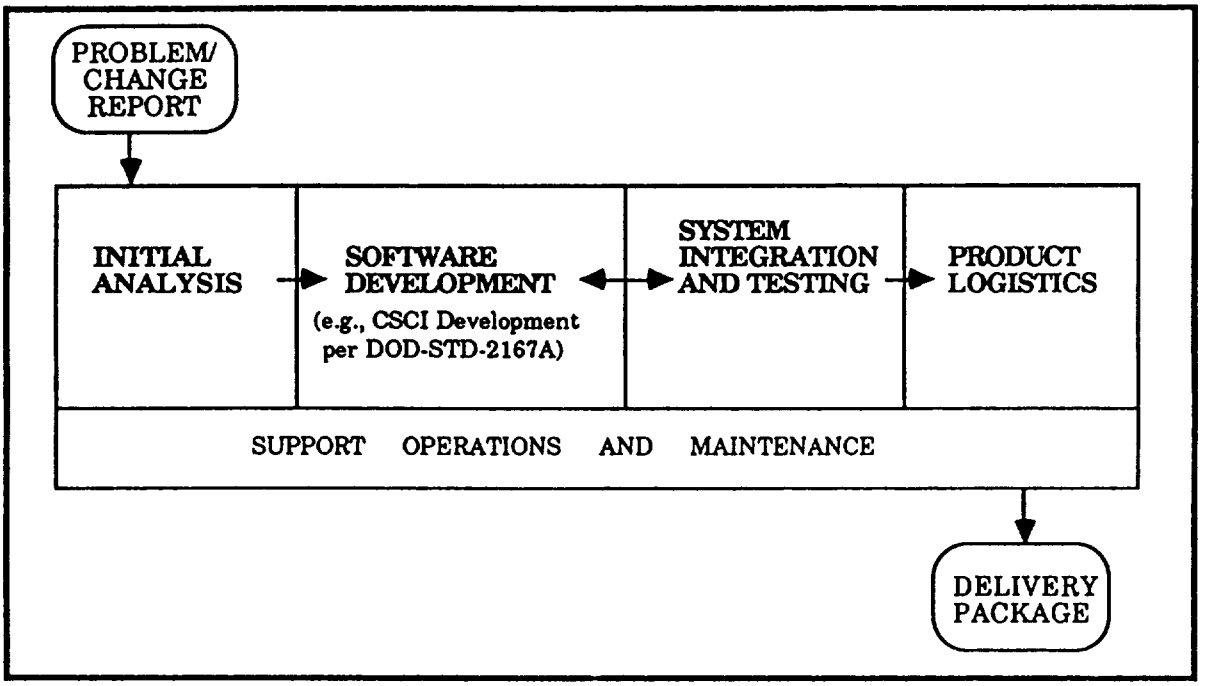

FIGURE 4. The general PDSS process.

6.2.1 Input. The PDSS or software change process is a sub-set of the system change process. Specific system change request procedures, which often initiate PDSS, tend to be Service specific. However, an overview is important to place PDSS in its proper context. The system change process usually begins when an agency or individual perceives a system problem or need for change. The agency or individual can be a system user, support person, CCB, or the program office. Depending on Service requirements, a problem/change report is prepared and submitted

to the logistic support organization, the program management office, or directly to the appropriate SSA. Specific problem/change report forms vary from Service to Service (e.g., software trouble report, problem trouble report, preliminary engineering change proposal), however, the format should be contained in either the Software Development Plan, the Software Configuration Management Plan, or the System Configuration Management Plan, and be readily available to system users. Appropriate data items for a problem/change report can be found in DI-MCCR-80030A. In this handbook, "problem/change report" describes all forms which report software problems or propose software enhancements. A problem/change report is eventually received by the SSA via configuration control channels to initiate the PDSS process.

6.2.2 Initial analysis phase. The implementation decision is the focus of all initial analysis activities and the objective of this phase. Initial analysis of the proposed change should include management, technical, and support impact analyses as required or directed by the CCB. It is not uncommon for the CCB to delegate specific authority to approve certain categories of software change. CCB includes, as appropriate, any lower level control board with delegated authority (e.g., software configuration control board). The SSA should provide the CCB with a detailed description of the problem/change report, the software change classification, the impact analyses, the estimated effort required for implementation, the risks associated with implementation, and the cost to implement. At any point during this phase, the CCB can decide not to implement the proposed change.

6.2.3 Software (CSCI) development phase. During the software development phase, software corrections and enhancements corresponding to approved problem/change reports are isolated, implemented, and tested. The evolving configuration of the CSCI is defined by the developmental configuration. The output of this phase is a CSCI developmental configuration, consisting of a new software version and associated document changes. The new software version, which may be referred to as the test version, is then released for system integration and testing.

6.2.4 System integration and testing phase. The purpose of this phase is to incrementally build the system from its component parts and to logically test the developing system. During system integration and testing, configuration items are integrated and tested to facilitate identification, isolation, and correction of faults. In modern real-time, high-speed, multi-processor systems, software functioning is closely coupled to hardware characteristics. It is important, especially with mission-

critical systems, to validate software on the target hardware. Consequently, system testing is an essential step in PDSS. The double-headed arrow in figure 4 signifies that CSCI development and system integration can be iterative activities. After system testing, the test version may not be approved; it may be approved and released for further development; or it may be approved and released to the using community. If the test version is not approved, the process begins again and the configuration identification is not updated. If the test version is approved and released for further development, the test version becomes the approved version, the configuration identification is updated, and the process begins again. If the test version is approved and released for distribution to the using community, the test version becomes the approved version, the configuration identification is updated, and the product logistics phase begins. In the final case, the output of the system integration and testing phase may be referred to as an operational version.

6.2.5 Product logistics phase. The product logistics phase is the final PDSS phase and involves the logistic support required for the new operational version. Since subsequent changes to the operational version will effect the delivery package, the product logistics phase is entered only after release for distribution to the using community. Principal activities include the production and verification of all delivery package items, delivery, user training, site installation and checkout.

6.2.6 Support operations and maintenance. Support operations and maintenance is not a separate phase of PDSS. Rather, support and maintenance activities must occur throughout the process and resource requirements must be planned. This group of activities is intended to include all PDSS support operations (e.g., computer center operations, tactical system operations, administrative operations, data management and security operations) and maintenance (e.g., facilities, computer equipment, tactical equipment). The agencies responsible for support operations or maintenance activities may be organic or, as in the case of maintenance agreements for commercial computer equipment support, provided by an outside agency.

6.2.7 Output. The final output of the PDSS process is a delivery package in the hands of trained users and a current configuration identification that corresponds with the approved software version.

6.3 Detailed PDSS process model. The detailed PDSS process model expands the general model and views each phase in more detail. Since it portrays specific activities, the detailed process model should be tailored to accommodate Service, program

or PDSS concept requirements. The tailored PDSS process, which becomes part of the PDSS concept, should be developed early in the acquisition process. The activities identified in the tailored PDSS process should relate to elements identified in the work breakdown structure in accordance with MIL-STD-881A. The following paragraphs describe specific activities in each phase of the detailed PDSS process model. To assist the reader, each activity block in figures 5 through 8 refers to the paragraph number where it is discussed. For simplicity, this model traces a single problem/change report through the process with approval at each decision control point. If the problem/change report is not approved for implementation, the originator should be notified and information documented for future use. Also, this example uses DoD-STD-2167A terminology.

6.3.1 Initial analysis phase. See figure 5.

6.3.1.1 Activity: Status accounting.

6.3.1.1.1 Input. Problem/change report

6.3.1.1.2 Description. A problem/change report is received, logged and identified by the SSA. If the problem/change report is a resubmission or if a control number has already been assigned by a higher level configuration control board, a new control number should not be assigned. The rules for numbering problem/change reports should follow the guidelines for engineering change proposal (ECP) numbering contained in MIL-STD-480B. Once the problem/change report is assigned a control number, formal configuration control procedures are in effect. Thereafter, the proposed change is reflected in the change status report (See MIL-STD-483A, Appendix VIII for Change Status Report preparation instructions and status indicators.) until final disposition has been completed and a CCB has directed its deletion. When the problem/change report is dropped from the Change Status Report, the original problem/change report should be archived.

6.3.1.1.3 Output. Identified problem/change report and updated Change Status Report

6.3.1.2 Activity: Configuration control decision.

6.3.1.2.1 Input. Identified problem/change report

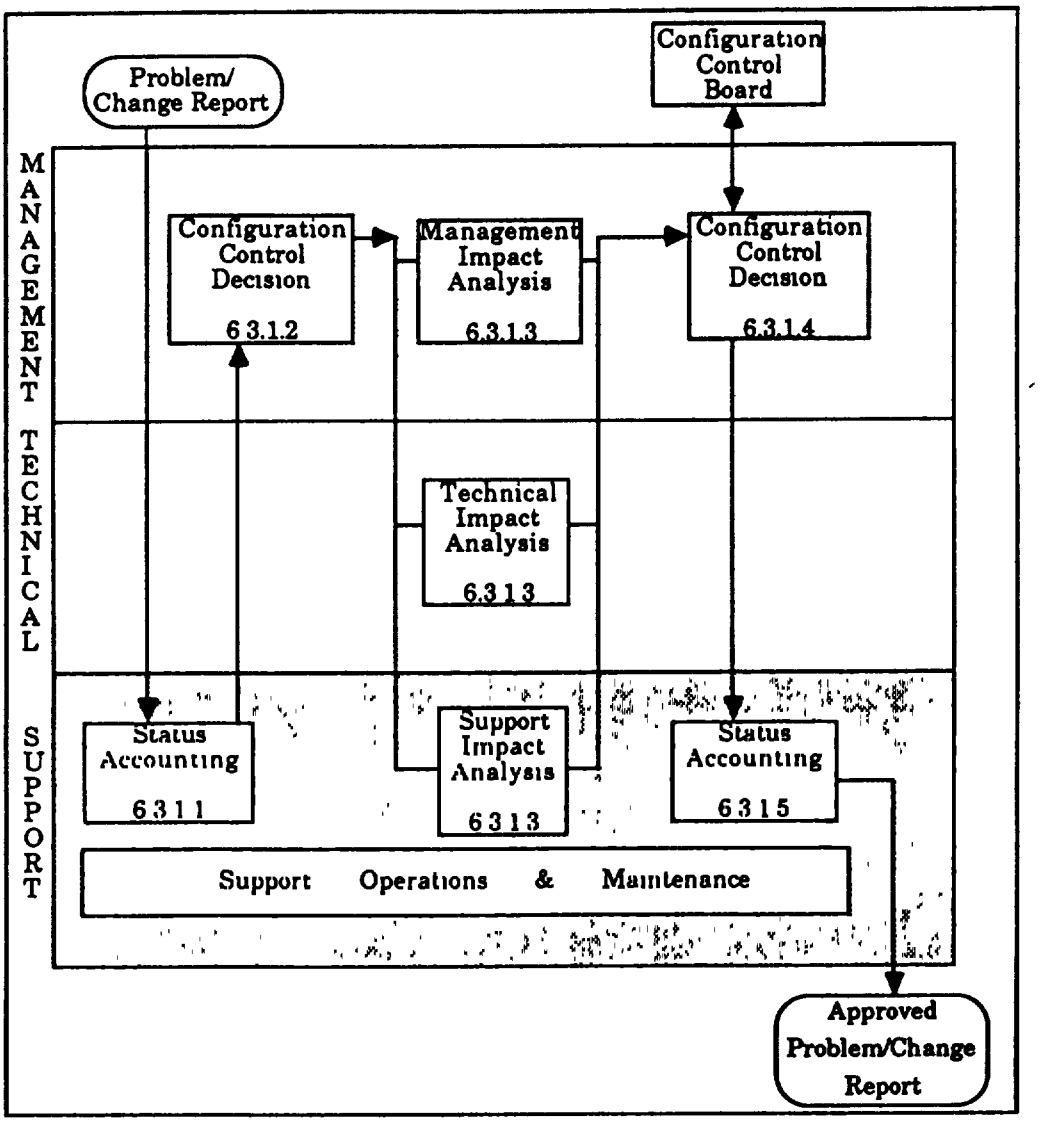

FIGURE 5. Detailed PDSS process model: initial analysis.

6.3.1.2.2 Description. The purpose of the initial configuration control decision is to review the proposed change, verify the classification, and to take appropriate action. The actions available to the CCB are limited by the authority that has been delegated, but appropriate actions may include:

a. forwarding a problem/change report to the CCB with a recommendation,

b. directing implementation of a problem/change report,

c. making a decision not to implement the proposed change,

d. directing that impact analyses be conducted,

e. directing that an ECP be prepared for submission to the CCB, or

f. requesting additional information.

6.3.1.2.3 Output. Impact analyses tasking(s)

6.3.1.3 Activity: Impact analyses.

6.3.1.3.1 Input. Impact analyses taskings

6.3.1.3.2 Description. Management, technical and support impact analyses are important activities since they are the basis for the implementation decision and cost estimates. The following issues should be considered during impact analyses:

a. Management

1. Is the SSA adequately staffed (numbers, disciplines and experience levels) to implement the proposed change?

2. Is the program adequately budgeted to implement the proposed change?

3. Are sufficient resources available and will this proposal affect ongoing or projected projects?

4. What are the operational issues that need to be considered (e.g., anticipated changes to system interface requirements, expected useful life of system, operational priorities, safety, security, impact if not implemented?) Many PDSS concepts provide an advisory group from the using community to consider and make recommendations on operational issues.

5. If the problem correction involves a hardware change, what are the management issues to be resolved or coordinated?

6. What is the estimated management cost to implement the change?

7. What is the impact on existing schedules?

8. What is the level of test and evaluation required?

b. Technical

1. Has the problem been replicated, verified (as a software problem) and isolated?

2. Does the problem correction impact a hardware configuration item?

3. What is the impact to the current operational version(s) and what are the risks associated with the proposed change?

4. What is the estimated level of software engineering effort required to correct the problem or to implement the proposed change?

5. What is the estimated technical cost to implement the change?

6. What is the level of test and evaluation required?

7. What is the category classification of the proposed software changes (i.e., software correction or enhancement)?

c. Support

1. What is the effect on computer center operations?

2. Is the target hardware available for system level testing?

3. What are the effects on facilities, support operations and maintenance?

4. What is the estimated support cost to implement the change?

6.3.1.3.3 Output. Impact analyses and cost estimates

6.3.1.4 Activity: Configuration control decision.

6.3.1.4.1 Input. Impact analyses and cost estimates

6.3.1.4.2 Description. If the proposed change is a class I change, a preliminary or formal ECP is forwarded to the CCB with impact analyses and estimated costs. Cost estimates should be recorded in block 21 of the ECP, DD Form 1692. If the proposed change is a class II change, the implementation decision can usually be made at this point. Often the software configuration control board at the SSA can make the implementation decision for class II changes without referring to a higher level CCB. Appropriate status and priority codes are assigned to the approved problem/change report. Also, early in the PDSS process, planning should begin for production and delivery of delivery packages. Early plans should include proposed milestones, release dates and tentative delivery schedules.

6.3.1.4.3 Output. Approved (for implementation) problem/change report and proposed delivery schedules

6.3.1.5 Activity: Status accounting

6.3.1.5.1 Input. Approved problem/change report

6.3.1.5.2 Description. In accordance with MIL-STD-483A, the Computer Software Configuration Index and the Change Status Report are updated to reflect the approved problem/change report and its current status.

6.3.1.5.3 Output. Approved problem/change report with current Computer Software Configuration Index and Change Status Report

6.3.2 Software (CSCI) development phase. See figure 6. The second phase of the PDSS process, follow-on software (CSCI) development, is the same as initial software development described in DoD-STD-2167A. The software (CSCI) development

phase begins at the highest level activity to be performed, depending on the magnitude of the software change(s) to be implemented. However, once the phase is entered, each subsequent activity is performed in accordance with DoD-STD-2167A. An adaptive software change (i.e., one that is the result of a change in the system requirements specification) usually involves all six activities of software (CSCI) development. However, software corrective or perfective changes often only involve the later activities (e.g., code & CSU testing, CSC integration and testing, CSCI testing). Throughout this phase a draft specification change notice (SCN) is prepared for each specification changed.

6.3.2.1 Activity: Software requirements analysis.

6.3.2.1.1 Input. Approved problem/change report

6.3.2.1.2 Description. The purpose of the software requirements analysis activity is to perform detailed problem analysis which often includes replication/isolation of the problem; a detailed analysis of the software change; and complete definition of any changes to the CSCI engineering requirements. These requirements are documented in the Software Requirements Specification (SRS) and the Interface Requirements Specification (IRS). In addition to initiating software development, an approved problem/change report also keys test personnel to begin identifying and implementing changes necessary to validate new software requirements. Block 31 of the ECP identifies specific test documents and test software changes necessary to validate an implemented ECP. As required, a new set of CSCI qualification requirements is developed and documented in the Software Test Plan (STP).

6.3.2.1.3 Output. If applicable, an updated allocated configuration identification and STP

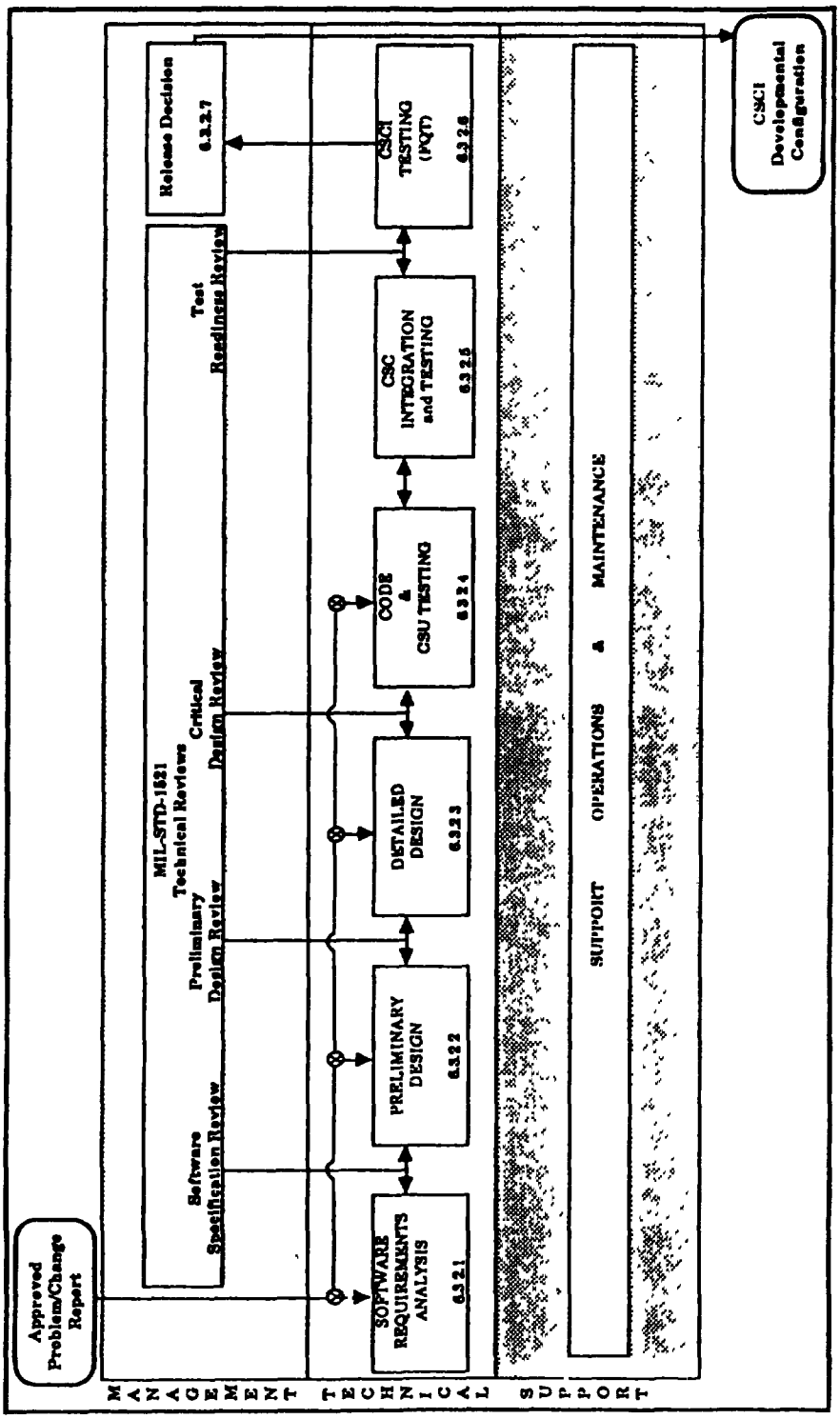

FIGURE 6. Detailed PDSS process model: software (CSCI) development.

6.3.2.2 Activity: Preliminary design.

6.3.2.2.1 Input. Updated allocated configuration identification and STP

6.3.2.2.2 Description. The purpose of the preliminary design activity is to develop a preliminary design for each CSCI based on the current (i.e., revised) SRS and IRS or, in the case where implementation of the proposed change begins with preliminary design, the change proposal. The preliminary design is documented in the Software Design Document (SDD). As required, the STP should be updated to reflect changes made to the SDD.

6.3.2.2.3 Output. New preliminary design as reflected in the SDD and updated STP

6.3.2.3 Activity: Detailed design.

6.3.2.3.1 Input. New preliminary design (SDD) and updated STP

6.3.2.3.2 Description. The purpose of the detailed design activity is to develop a detailed design for each CSCI based on the current preliminary design or, in the case where implementation of the proposed change begins with detailed design, the change proposal. The detailed design is documented in the SDD. As required, the Software Test Description (STD) should be updated to reflect changes made to the SDD and the STP.

6.3.2.3.3 Output. New detailed design and updated STD

6.3.2.4 Activity: Code & computer software unit (CSU) testing.

6.3.2.4.1 Input. New detailed design and updated STD

6.3.2.4.2 Description. The purpose of the code and CSU testing activity is to code and test each CSU ensuring that the algorithms and logic employed by each CSU are correct and that the CSU satisfies specified requirements. CSU Software Development Files (SDF's) should be updated as required including new test requirements, cases, procedures, and results. The new source code listing should be incorporated into the appropriate developmental configuration.

6.3.2.4.3 Output. New developmental configuration and updated CSU SDF's

6.3.2.5 Activity: Computer software component (CSC) integration & testing.

6.3.2.5.1 Input. New developmental configuration, STP, and STD

6.3.2.5.2 Description. The purpose of the CSC integration and testing activity is to ensure that the algorithms and logic employed by each CSC are correct and that the CSC satisfies its specified requirements. Test results should be recorded in the corresponding CSC SDF. CSC's are combined and tested until the CSCI developmental configuration has been generated and tested. The new integrated source code listing should be incorporated into the appropriate developmental configuration.

6.3.2.5.3 Output. New developmental configuration and updated CSC SDF's

6.3.2.6 Activity: CSCI testing.

6.3.2.6.1 Input. New developmental configuration, updated CSC SDF's, and updated STD

6.3.2.6.2 Description. The purpose of the CSCI test, or formal qualification test (FQT), is to determine whether the CSCI complies with allocated requirements (DoD-STD-2167A). FQT is conducted on the CSCI developmental configuration in accordance with updated procedures contained in the STD. A Software Test Report (STR) is prepared which provides a permanent record of formal qualification testing performed on each CSCI.

6.3.2.6.3 Output. Software Test Report

6.3.2.7 Activity: Release decision.

6.3.2.7.1 Input. Software Test Report

6.3.2.7.2 Description. This activity includes the review of the STR, validation and authentication (i.e., ensuring specifications are complete, correct and current) of the CSCI developmental configuration and release of the developmental configuration for system integration and testing. After the developmental configuration is released for system integration and testing it is sometimes referred to as the test version. The earlier software faults are identified, the cheaper they are to correct. Therefore, the CCB should ensure that the test version satisfies requirements before it is released for system integration and testing.

6.3.2.7.3 Output. Authenticated CSCI test version (developmental configuration) released for system integration and testing

6.3.3 System Integration and Testing phase. See figure 7.

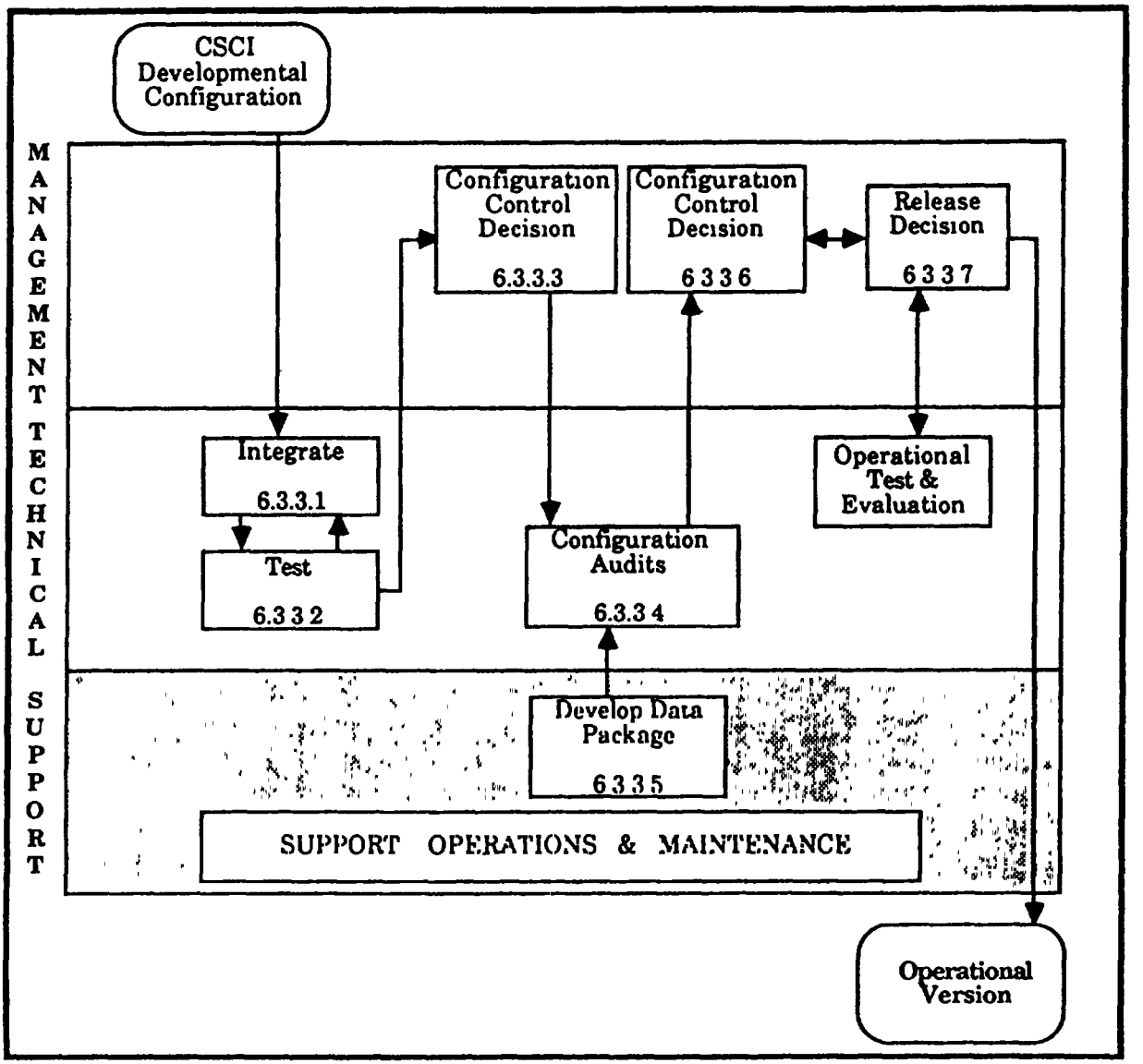

FIGURE 7. Detailed PDSS process model: system integration and testing.

6.3.3.1 Activity: Integrate.

6.3.3.1.1 Input. Authenticated CSCI test version

6.3.3.1.2 Description. This activity involves combining CSCI's to create an integrated software system. It is appropriate for test or configuration management personnel to generate the executable image in order to verify software generation procedures, and to

ensure that software changes have been identified and properly documented. Depending on the size and complexity of the software program, several interim versions may be evaluated before the complete software program has been generated and is ready for "system" testing. The term interim versions is used herein to mean two or more CSCI's that have been integrated into a single executable image. Each test level should be identified and scheduled in the STP.

6.3.3.1.3 Output. (Interim) test version

6.3.3.2 Activity: Test.

6.3.3.2.1 Input. (Interim) test version

6.3.3.2.2 Description. Each integration test is intended to ensure that requirements are satisfied in the (interim) test version. Formal testing should include regression testing and CSCI interface testing at every level, and should be witnessed. The purpose of regression testing is to reasonably ensure that faults have not been introduced during the PDSS process. The purpose of CSCI interface testing is to reasonably ensure that the component parts of each (interim) test version are working together properly and satisfy quality criteria. The integrate and test cycle is often repeated several times until the complete software program is generated and tested. If required by the PDSS concept, IV&V activities may be scheduled to coincide with integration or system testing.

6.3.3.2.3 Output. Test results

6.3.3.3 Activity: Configuration control decision.

6.3.3.3.1 Input. Test results

6.3.3.3.2 Description. The purpose of this configuration control activity is to review test results, validate requirements, and release the test version (developmental configuration) for configuration audits (verification of performance and design) and data package development. If the test version does not satisfy requirements or the documentation is not adequate, the CCB may return the test version, via configuration control channels, to the appropriate activity for rework.

6.3.3.3.3 Output. A software program test version released for configuration audits

6.3.3.4 Activity: Configuration audits.

6.3.3.4.1 Input. Software program test version, current configuration identification with proposed changes, test results and approved problem/change report(s) from the initial analysis phase (activity 6.3.1.4)

6.3.3.4.2 Description. After requirements have been satisfied, functional and physical configuration audits (FCA and PCA) should be conducted as independent activities under the purview of quality assurance or configuration management organizations. Audits provide important management visibility and control, and are an opportunity to evaluate software product supportability, integrity, and quality. Configuration audits should be performed before the test version is approved. The PDSS process model depicts configuration audits conducted after system integration and testing. However, audits may be conducted after FQT. When audits are conducted after FQT, the input to the system integration and testing phase is an approved software version and associated configuration identification. Audits of less than one hundred percent of the software program may be approved to reduce cost and personnel requirements.

6.3.3.4.3 Output. Audit results

6.3.3.5 Activity: Develop data package.

6.3.3.5.1 Input. Draft Specification Change Notice

6.3.3.5.2 Description. Draft SCN's, which are a products of the software development phase, contain preliminary specification change pages or "redlined" text resulting from the associated ECP/software change request implementation. A SCN should be prepared for each specification or controlled document that is affected by the implemented problem/change report. The document production activity involves the development, technical editing, verification and production of change pages or document revisions in their "camera-ready copy" or ready for publication form. If several SCN's have been promulgated, a revision should be published. A Notice of Revision (NOR), DD Form 1695 may be used to notify custodians of a revision. (MIL-STD-480B and MIL-STD-490A refer.).

6.3.3.5.3 Output. Formal SCN's or revision

6.3.3.6 Activity: Configuration control decision.

6.3.3.6.1 Input. Test version, proposed changes to the current configuration identification, test results, configuration audit results, and data package(s)

6.3.3.6.2 Description. This decision by the CCB establishes a new approved software version and places the updated configuration identification under formal control. Depending on the level of change and the authority delegated, the CCB may take one of the following actions:

a. Forward the test version, test results, and audit results with a recommendation;

b. Approve the test version and associated changes to the configuration identification; or

c. Do not approve the test version.

Whenever an updated configuration identification is established, a SCN is published for each specification changed in accordance with MIL-STD-490.

6.3.3.6.3 Output. Approved software version and updated configuration identification documented by SCN's

6.3.3.7 Activity: Release decision.

6.3.3.7.1 Input. Approved software version and updated configuration identification

6.3.3.7.2 Description. The release decision refers to the release of the approved software version to the using community (i.e., operational release decision). The PDSS concept for many MCCR systems may require an operational test and evaluation (OT&E) before a new software version is released to operational forces. OT&E is conducted by Service operational test agencies in an environment as realistic as possible using typical operators and support personnel. The OT&E decision includes funding, scheduling, and commitment of operational forces to support the evaluation. Therefore the decision to conduct and schedule OT&E is usually made above the program management level. The release decision is made in accordance with Service regulations after OT&E results have been reviewed. If the decision is made to release the approved software version to the using community (i.e., operational release), a tentative delivery schedule is prepared.

6.3.3.7.3 Output. Operational release decision, tentative delivery schedule, and an operational version

6.3.4 Product logistics phase. See figure 8.

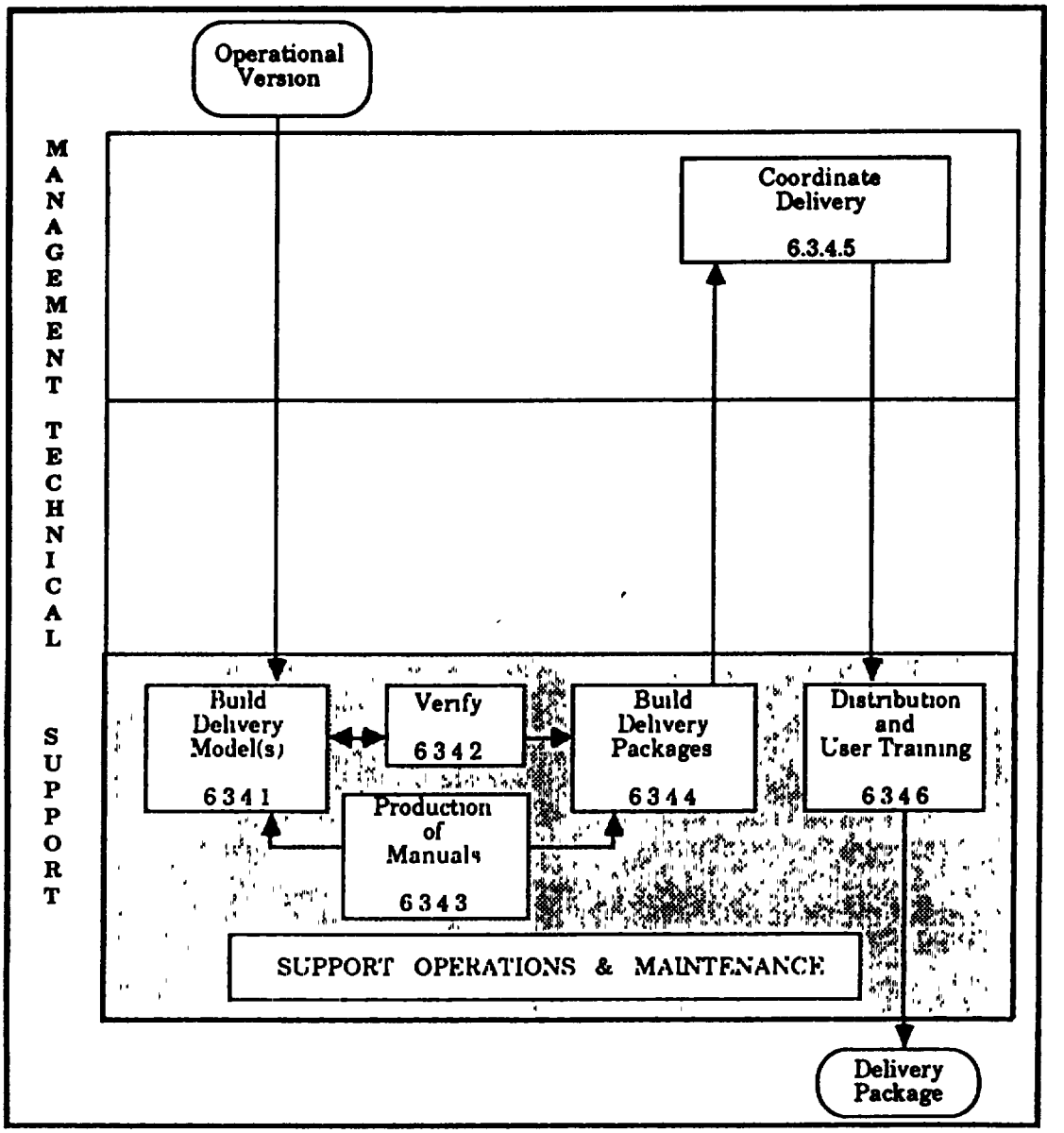

FIGURE 8. Detailed PDSS process model: product logistics.

6.3.4.1 Activity: Build delivery model(s).

6.3.4.1.1 Input. Operational version

6.3.4.1.2 Description. It is not uncommon for weapons systems to require hardware, site, or model specific data. In cases where adaptation requirements exist (e.g., magnetic declination data tables, altitude/height above water data tables, radar

sweep/rotation constants), a single approved software version may include many such types, each separately identified and controlled. The implementation and verification of adaptation data is normally accomplished during the product logistics phase. These unique versions are configuration types and do not constitute new configurations in accordance with MIL-STD-483A. This activity also includes fabrication or initial production and verification of model delivery packages which include, as appropriate, the approved software version in all deliverable forms of magnetic media; user/operator manual change pages or revisions; firmware or integrated circuit boards; and new diagnostic software. A model of each deliverable item should be built and verified before mass production. If the SSA will conduct on-site training during delivery, the model delivery package should include training objectives, lesson outlines, and training materials to be used or provided.

6.3.4.1.3 Output. Delivery model(s)

6.3.4.2 Activity: Verify.

6.3.4.2.1 Input. Delivery model(s)

6.3.4.2.2 Description. This activity involves a final check for completeness, correctness, and verification of delivery package components. This is an important step because production units are fabricated from the delivery model(s). Faults not identified and corrected will be carried through into production.
6.3.4.2.3 Output. Verified delivery package model(s)

6.3.4.3 Activity: Production of manuals.

6.3.4.3.1 Input. Verified delivery package model(s)

6.3.4.3.2 Description. This activity can be considered a component of the "build delivery packages" activity. Because it is important, sometimes overlooked, and requires considerable lead time, it is separately highlighted. The manual production activity should include tasks such as document verification, technical editing, illustration and production. Early planning for manual production should include personnel requirements, which may include skills not commonly associated with the PDSS process, funding requirements, and lead time requirements. The SSA should retain camera-ready copies, illustrations, and graphics for future use.

6.3.4.3.3 Output. Manuals

6.3.4.4 Activity: Build delivery packages.

6.3.4.4.1 Input. Verified delivery package model(s)

6.3.4.4.2 Description. This activity involves tasks associated with the production of delivery packages. It may include for example the fabrication, manufacture or production of magnetic media, firmware, integrated circuit cards, user manuals, lesson outlines, student documents, maintenance kits, diagnostic kits, etc. An important aspect of the build activity is an effective quality control program. A sufficient number of delivery packages should be produced to satisfy known logistic requirements (e.g., stockage levels, depot needs, spares provisioning).

6.3.4.4.3 Output. Delivery packages

6.3.4.5 Activity: Coordinate delivery.

6.3.4.5.1 Input. Tentative delivery schedule (See 6.3.3.7)

6.3.4.5.2 Description. This activity really encompasses all final delivery preparations. Coordination milestones should be established to coordinate production and publication schedules with delivery schedules. Delivery can be completed in a few days or, with a large and geographically separated using community, delivery can take months. Sometimes, delivery schedules should be delayed and coordinated with hardware changes such as a system retrofit, overhaul, or rework associated with a scheduled service life extension program. Additional considerations may be required when the installation of integrated circuit boards, chips or magnetic media needs to be coordinated with a hardware change. Such situations often require coordination with other ILS agencies. Although depicted as a single event, delivery planning should begin once an operational version has been established and the operational release decision made. Long lead items, such as training plans, installation instructions, and diagnostic support tool requirements, should be identified and receive attention much earlier. Often, delivery must also be coordinated with unit deployments, availability of personnel, funding, and operational commitments. Many factors need to be considered and coordinated before the final delivery is scheduled. Delivery alternatives include the U.S. mail, couriers, electronic transmission, or delivery teams.

6.3.4.5.3 Output. Coordinated production/publication and delivery schedule and completed delivery packages

6.3.4.6 Activity: Distribution and user training.

6.3.4.6.1 Input. Approved delivery schedule and verified delivery packages

6.3.4.6.2 Description. On-site deliveries should be more than just turnover of the new software version. At a minimum, five things should be accomplished by the delivery team at each delivery site:

a. turnover and verify receipt of the delivery package,

b. disposition of outdated delivery package,

c. installation of the software,

d. training which emphasizes new functional capabilities and changes to the man-machine interface, and

e. update any computer aided instruction software.

But, there should always be two additional items on the informal agenda. Delivery of a new software product is an opportunity to reinforce lines of communication between the SSA and the using community and to promote the support role of the SSA. It is important that the attitude of each delivery team member be one of support and cooperation with the using community.

6.3.4.6.3 Output. Delivery package

APPENDIX A

SAMPLE SSA'S CHARTER

10 Purpose. The purpose of this appendix is to provide a sample charter which designates the SSA and defines authority, responsibility, organizational relationships, and support channels.

20 Information. The SSA should be designated early during concept exploration, but no later than milestone one. The SSA's charter should also be published early and maintained throughout the pre- and post-deployment stages of software support. Eventually the charter should be incorporated into the CRLCMP. The external environment of the SSA can be relatively simple or it can be a complex array of command, coordination, and support channels. In each case, however, the lines of authority and support need to be specifically identified. Single Service, consolidated PDSS is the simplest scenario. In such a case, organizational boundaries, lines of authority, and support channels are usually well established and clearly understood. However, joint PDSS scenarios often include widely dispersed activities, performed by a variety of service organizations and support contractors. Responsibilities, support channels, and lines of authority are not so clear in such circumstances and can lead to a confusing PDSS environment. Two facets to the "authority" issue should be addressed in the SSA's charter. Management authority, which normally flows along organizational lines, typically includes the right to direct, plan, organize, control, and allocate resources. The second facet of the authority issue, configuration control authority, includes the right to determine the functional and physical configuration of the supported software. This authority is always exercised within the context of documented and approved configuration management procedures. In a single Service scenario, the designation and charter of the SSA will reflect appropriate Service directives. When the PDSS concept calls for joint PDSS, the designation and charter of the SSA should be accomplished by a memorandum of agreement.

30 Software support manager. It is not uncommon for a single SSA to support multiple systems. Often it is beyond the capabilities of one individual to effectively manage software support activities for all systems assigned to the SSA. In such cases, a software support manager should be appointed. The software support manager is a member of the SSA and is the individual responsible within the normal chain of command for the

software support of a single designated system. The software support manager is also the software support advocate for the system. When multiple systems are contending for shared resources this can be an important role. When appropriate, the SSA's charter should identify the software support manager and define authority and responsibility.

40 Sample SSA's charter. Figure 9 contains a sample SSA charter outline.

I DESIGNATION OF THE SOFTWARE SUPPORT ACTIVITY (SSA)

[Identify SSA] is designated the Software Support Activity for [Identify system], hereafter referred to as the designated system, effective [Date]

II. DESIGNATION OF THE SOFTWARE SUPPORT MANAGER

[Identify software support manager] is designated the software support manager for the designated system, effective [Date]

III SSA TASKINGS

The SSA is responsible in accordance with applicable Department of Defense (DoD) Directives, Service regulations, and program direction for:

A. Providing pre- and post-deployment software support within the scope of this charter and the approved PDSS concept for the designated system as provided for in [Identify CRLCMP]

B Supporting the Program Manager in taskings pertaining to software support of the designated system

IV SSA RESPONSIBILITIES AND AUTHORITY

A. SSA responsibilities

1. The SSA is responsible to plan, coordinate, and control software support activities in support of the designated system

FIGURE 9. Sample SSA's charter

2. During the pre-deployment software support stage, the SSA is responsible to assure that the accepted software is supportable, that software requirements are satisfied, and to plan for transition and Post-Deployment Software Support (PDSS).

3. During transition, the SSA is responsible to implement the transition plan and to demonstrate that the software can be supported in accordance with the PDSS concept.

4. During the PDSS stage, the SSA is responsible to the Program Manager to provide cost effective PDSS, evaluate and maintain software quality, and to coordinate related PDSS activities.

5 During the entire life cycle, the SSA is responsible for the software configuration identification, status accounting, and configuration audits of Government controlled baselines associated with the designated system.

B. SSA Authority

1. [The specific authorities delegated to the SSA should be identified to include, for example, authority over contractors, financial authority, organizational authority, and authority to communicate with members of the using community]

2. Configuration control authority The configuration control authority and responsibilities of the SSA are defined in [The appropriate configuration management plan or inter-Service agreement should be referenced].

V. RESOURCE CONTROL

A The SSA will ensure that financial and personnel requirements to accomplish the above responsibilities are submitted in accordance with established procedures for inclusion in the Program Objective Memorandum (POM) for applicable target program years

B Funding to accomplish the above responsibilities will be provided to the SSA as direct mission funding by [Identify PDSS funding source or sources for the designated system] The SSA will, in turn, allocate the necessary funding and provide direction, as applicable, to participating organizations for services provided in accordance with current regulations, policies, and procedures.

FIGURE 9. Sample SSA's charter - continued

C As early as practical and continuing throughout the pre-deployment software development phase, the SSA will solicit approval and advocacy from the Program Manager or the Integrated Logistic Support (ILS) Manager regarding acquisition and support of the Software Engineering Environment (SEE) and the Software Testing Environment (STE) in accordance with the approved PDSS concept

D PDSS resource requirements will be identified in the Computer Resources Life Cycle Management Plan (CRLCMP).

VI RESOURCE AND FACILITY SUPPORT

A Resource support·

The [Major Service support organization] located at [Organization and address] will coordinate and provide support to the SSA in accordance with the CRLCMP Liaison/field offices may be established by the Service support organization to facilitate support planning, administration, and control.

B Facility support

Facility support (office space, utilities, etc) for the SSA and for other agencies assigned PDSS activities will be provided in accordance with the CRLCMP

VII COMMUNICATION CHANNELS

Direct communication is authorized among participants involved in implementation of the PDSS concept and support of the designated system to ensure timely and effective coordination and exchange of information

FIGURE 9. Sample SSA's charter - continued

APPENDIX B

THE PDSS CONCEPT

10 Purpose. The purpose of this appendix is to discuss the PDSS concept and its components and to examine alternative PDSS concepts.

20 Information. The PDSS concept, the basis for all subsequent PDSS planning, must be conducted early in the acquisition process. Since the PDSS concept is directly affected by operational, support and funding issues, PDSS concept decisions should be approved at the program management level or higher. The approved PDSS concept should be documented in the CRLCMP, the ILSP, the Test and Evaluation Master Plan, and reflected in all other software support plans.

30 PDSS concept. Specific PDSS concept content and format will vary from program to program. However, a complete PDSS concept should answer each of the following questions as they pertain to the designated system.

a. What is the PDSS objective?

b. What is the tailored PDSS process?

c. Who will perform each PDSS activity?

d. What are the initial cost estimates for PDSS?

40 PDSS objective. Since the tailored PDSS process is derived from the PDSS objective, the PDSS objective should be approved before modeling. System characteristics, the operational environment, software volatility and the needs of the using community generally determine the PDSS objective. At a minimum, the PDSS objective should describe the scope of PDSS, and it may include provisions for consolidating resources or centralizing PDSS activities.

40.1 Scope of PDSS. Responsiveness to the using community is the primary consideration in determining the scope of PDSS. The scope of PDSS should be tailored to satisfy operational response requirements. Scope relates to how responsive the SSA will be to proposed changes submitted by the using community. For example, a full scope PDSS concept suggests that the SSA will provide full support for the entire deployment phase. This includes responding to all approved software change categories (i.e.,

corrections and enhancements) within a reasonable period. PDSS concepts that limit the scope of PDSS are referred to as "limited scope concepts". Limited scope concepts limit the support period, the support level, or both. While limited scope concepts can result in major cost reductions, they do impact the SSA's ability to respond to proposed software changes and can result in reduced system readiness.

40.1.1 Limited scope concepts

40.1.1.1 The support period. It is not always necessary to maintain an on-line PDSS capability for the entire deployment period of the system. This is an idea carried over from hardware ILS. Unlike hardware support, software support can sometimes be terminated while the supported system is still deployed. Since software doesn't break or wear out, repair parts and stockage levels are not maintained, and preventative maintenance is not performed, PDSS can be temporarily suspended, modified, or terminated. That is, once the software has matured, the environment has stabilized, and essential needs have been satisfied, it may not be necessary to continue PDSS. The temporary suspension of PDSS is feasible, but risk and re-start costs must be considered. A more extreme example of limiting the support period is a PDSS concept that says, "no PDSS". In some circumstances, throw away software can be a viable solution to reducing PDSS cost. Also, consider another example which virtually reduces PDSS operating cost to zero after an initial and pre-determined support period:

> The PDSS objective for system X is to release two operational versions during the service life of the system. The target dates and the content of each release will be determined by the Program Manager. The second and final release will be no later than five years after initial deployment. After the second release and at the direction of the Program Manager, routine PDSS activities will be terminated and software technical data will be archived by the SSA. The disposition of other computer resources will be determined by the responsible command(s).

40.1.1.2 The support level. The scope of PDSS can be limited by controlling PDSS operating cost. A program budget can be a simple but effective means to limit PDSS operations. It then becomes the responsibility of the SSA to provide efficient PDSS

for the system within the budget ceiling. An alternative is to control software volatility. This can be accomplished through budgetary controls above minimum levels necessary to sustain PDSS operations, annual software "change rate" ceilings, or by establishing criterion for software change (e.g., essential software adaptation changes will be the only software changes approved for implementation.).

40.2 Resource consolidation. Resource consolidation is a PDSS concept which involves the collocation of "PDSS resources" for two or more supported systems. Resource consolidation, or simply sharing resources across program boundaries, can reduce costs, provide redundancy and introduce flexibility at the SSA. A shared resource strategy requires early planning and a coordinated effort above the program office level.

40.3 Centralizing PDSS functions or activities. Centralizing PDSS functions or activities involves the concentration of PDSS functions or activities for two or more systems at a single SSA. Joint PDSS, a common form of centralized PDSS, is when one SSA has PDSS responsibility for one system or similar systems deployed by two or more military Services. For similar systems, systems which posses some common system or support characteristic, a centralized PDSS approach can be very cost effective. Support costs can be reduced by reducing training costs, reducing total personnel requirements by employing similar skills across program boundaries, avoiding the procurement and support of common hardware items in the software support environments, and reducing management overhead costs. Decentralization, the other end of the same spectrum, involves multiple SSAs, through a coordinated effort, supporting a single system by allocating PDSS functions or activities. One SSA must be designated the lead activity and coordinate the entire PDSS process, including system configuration management and system integration and testing. If a decentralized PDSS concept is implemented, functions and activities must be carefully defined. Configuration control becomes more critical and must be closely managed.

50 Tailored PDSS process. The second component of a complete PDSS concept is a description of the tailored PDSS process. The tailored PDSS process, which resembles the detailed PDSS process model described in Section 6, should identify each PDSS activity and the organization responsible. Responsible organizations can be the SSA, other Service organizations, other DoD organizations or commercial contracting organizations. Detailed PDSS planning can not commence and requirements can not be projected until the

tailored PDSS process has been defined. At this point in the planning process, specific organizations may not be identified, but at a minimum, activities should be assigned and generally grouped (e.g., by Service, agency or contractor support). During planning, it is often worthwhile to perform a preliminary PDSS resource analysis to examine the allocation of PDSS resources and to derive alternative PDSS concepts. Improving resource utilization, through consolidation or centralization for example, should always be considered as a cost saving alternative.

50.1 Critical activities performed by the Government. Some PDSS functions are critical to system performance and the Government's capability to continue software support. Such functions are inherently management responsibilities and should be performed by Government agencies. The following functions are in this category:

a. Basic program management functions (program planning/budgeting, organizing, staffing, directing, controlling);

b. PDSS concept decisions;

c. Configuration audits;

d. Configuration control and configuration identification of Government controlled baselines;

e. Software quality assurance; and

f. System level testing and OT&E.

50.2 Critical SSA activities. As the DoD agent for PDSS, the SSA should participate in the following activities:

a. Basic PDSS management functions (PDSS planning, organizing, staffing, leading, controlling);

b. Identifying SSA requirements;

c. Software configuration control and configuration identification; and

d. Software quality assurance (specifying and evaluating software quality requirements).

APPENDIX B

60 PDSS concept decision criterion. A decision criterion for PDSS concept selection should be determined. Decision criteria are most often stated as PDSS characteristics to be optimized (e.g., maximized or minimized) such as cost, responsiveness, software reliability or risk. Multiple decision criteria should be prioritized, and alternatives that do not satisfy minimum requirements should be discarded. For example, if a system operates in a stable environment and is critical to the national defense, system reliability is probably the determining criterion. If an alternative does not satisfy the criterion for reliability, it can be deleted from the set of feasible alternatives. Other PDSS concept decision criteria, such as cost and responsiveness, are not even considered. In this case, the tailored PDSS process might be very structured and controlled with an emphasis on software quality, formal testing, OT&E and strict configuration control procedures. Often, a single characteristic is of such overriding importance, such as system reliability, that a single selection criterion can be used.

70 Costs. For many systems cost is a determining criterion for the PDSS concept decision. It is useful to briefly discuss the subject of software costs, and at the same time, introduce a few fundamental software support cost concepts. Figure 10 relates software life cycle cost (SLCC), software life cycle support cost (SLCSC), and PDSS cost. When cost is the decision criterion for selecting a PDSS concept, SLCC should be used because many costs can be shifted between acquisition and PDSS. Sometimes decisions that appear to be cost-effective during acquisition, may not be in the best long-term interest of the Government. The following paragraphs briefly describe the software cost terms introduced in this handbook.

70.1 Software life cycle cost. SLCC is the sum of all initial software development costs and software life cycle support costs. The first component of the SLCC equation, initial software development cost, includes software acquisition and contracting costs such as costs associated with software development, Government overhead, system integration, software IV&V, and prototype development.

70.2 Software life cycle support cost. SLCSC, which is the second component of SLCC, is the sum of all software support costs that occur during the software's life cycle. This includes pre- and post-deployment software support costs.

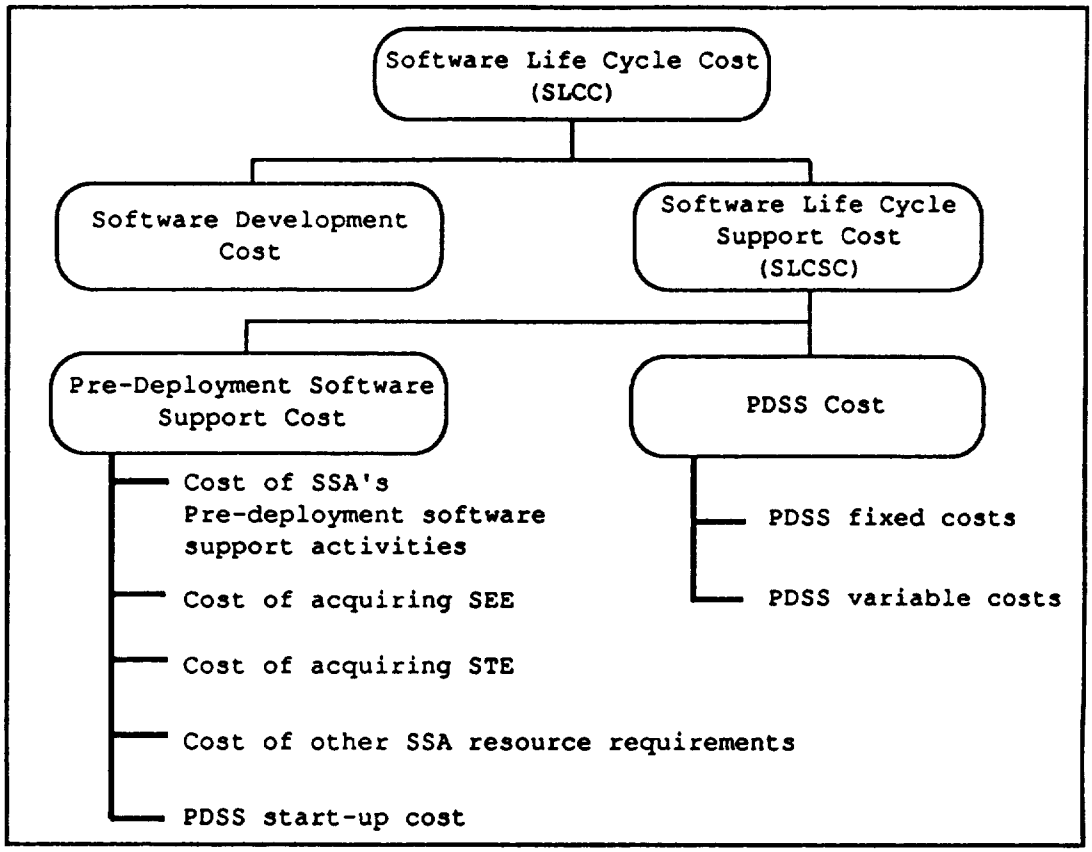

FIGURE 10. General software cost components.

70.2.1 Pre-deployment software support cost. Pre-deployment software support cost is the sum of all software support costs that occur prior to deployment. For the most part, these are costs associated with the SSA's pre-deployment activities described in Section 5 of this handbook. However, pre-deployment software support cost also includes transition costs and PDSS start-up costs. PDSS start-up cost includes direct Government costs for transition, equipment, staffing, initial training, acceptance tests and OT&E, SEE/STE relocation and installation, and facility construction or modification.

70.2.2 PDSS cost. PDSS cost is the sum of all costs associated with PDSS. It includes PDSS fixed costs which are necessary to sustain minimum PDSS operations and PDSS variable costs which are directly related to the level of PDSS operations. PDSS variable costs can be adjusted by varying the level or scope of PDSS.

80 Steps in developing a PDSS concept. Like most planning, developing a PDSS concept is a very creative process, and does not lend itself to cookbook instructions. However, the following

sequence of steps may be helpful when setting out to develop a PDSS concept:

a. Consider the impact of system environmental factors such as:

   1. system and software criticality,

   2. anticipated software volatility,

   3. interface requirements and interface volatility, and

   4. characteristics of the using community (e.g., size, dispersion, experience with automated systems);

b. Identify constraints (e.g., budget, staffing, organization, facilities);

c. Define the PDSS objective;

d. Develop feasible PDSS process alternatives. Each alternative should describe a tailored PDSS process (i.e., activities and organizations);

e. Define selection criteria or criterion which reflect the PDSS objective;

f. Analyze and evaluate each alternative; and

g. Select an alternative for implementation.

APPENDIX C

PDSS RESOURCE ANALYSIS

10 Purpose. The purpose of this appendix is to provide a general methodology for performing PDSS resource analysis and projecting SSA resource requirements.

20 Resource categories. After the SSA has been designated and a PDSS concept has been approved, resource analysis can begin. The objective of this analysis is to identify and project computer resources necessary to implement the approved PDSS concept. The process involves identifying SSA resource requirements in five resource categories:

software resource requirements,
hardware resource requirements,
facility resource requirements,
other resource requirements, and
personnel resource requirements.

This categorization of SSA resource requirements, which is consistent with the requirements structure used in the Computer Resources Integrated Support Document (CRISD, DI-MCCR-80024A), will be used throughout the resource analysis discussion. Once SSA resource requirements have been identified and approved, they should be specified in the CRLCMP which is the source and principal reference for resource requirements throughout the life cycle. Assumptions and models used to estimate requirements should be included or referenced in the CRLCMP.

30 Identifying software and hardware resource requirements. As a general rule, the software engineering and test environments (SEE and STE) during early PDSS should be the same as during initial software development. It should be pointed out that the SEE and STE do not include facility, personnel or other resource requirement categories. Components of the SEE and STE do include all hardware and software necessary for the development, code generation, integration and testing of software. If SSA, program or Service SEE or STE requirements exist, they should be identified early and contractually specified as PDSS acquisition requirements to be incorporated in the SEE/STE by the developing agency. SEE/STE PDSS acquisition requirements might include an automated program design language, source code editor, compiler (for the operational program and support software), static code analyzer, host computer, computer aided software engineering tools, configuration management tools, formal specification and

verification tools, system simulator, emulators, etc. Planners should avoid attempting to change components in the SEE or STE during transition. As we have discussed, the transition period is difficult enough without introducing additional complexity or another source for software faults or problems. Attempting to introduce changes in the SEE/STE during transition, usually indicates inadequate early planning. One of the risks associated with this situation is that it is very difficult to separate latent (i.e., existing) faults or problems from faults or problems introduced as a result of changes in the SEE or STE. Software and hardware computer resource requirements which make up the SEE and the STE are identified in the CRISD. Additional requirements may also be found in other acquisition documents (e.g., Logistic Support Analysis, Integrated Logistics Support Plan).

40 Identifying facility resource requirements. Facility resource requirements are also described in the CRISD which is prepared by the developing agency. However, SSA facility requirements are often different. It should be pointed out that this situation is not the same as the situation discussed in the previous paragraph (i.e., changing the SEE or STE during transition). Facility requirements can be modified without introducing significant risk in the software development, code generation, integration or test process. If facility related problems do occur, they are usually easily identified. Planning for SSA facility requirements can involve new military construction, renovation of existing Government facilities, lease arrangements or a variety of other facility alternatives. Facility requirements can be a major cost consideration and alternatives need to be considered when developing the ILSP and the PDSS concept. For example, if the PDSS concept involves providing PDSS operations for a specified period during the deployment phase, a lease arrangement might be more cost effective than new construction. PDSS facility resource analysis should include the following facility requirement considerations:

a. Office requirements,

b. SEE/STE laboratory space,

c. Software engineer workstation requirements,

d.. Local/long-haul network requirements,

e. Power requirements,

f. Temperature requirements,

g. Vault and secure storage requirements (including vaults with environmental controls for magnetic media storage),

h. Library requirements,

i. Equipment maintenance facility requirements,

j. Storage facility requirements to include off-site or backup storage facilities,

k. Security requirements (physical security, special security, TEMPEST), and

l. Communication requirements.

50 Identifying other resource requirements. Other resource requirements can include technical data requirements, data rights, funding requirements, data management requirements, administrative requirements, logistic requirements, etc. Again, the CRISD is a good beginning point to identify other miscellaneous requirements.

60 Estimating personnel resource requirements. Personnel requirements are a major PDSS cost component, and, at the same time, the most difficult to accurately estimate. Unlike other resource requirements which are "identified", personnel requirements are estimates. The distinction is meaningful. Before personnel requirements can be estimated, it is important to understand PDSS process characteristics, which are discussed later in this appendix. In general, other resource requirements are identified during acquisition and in a sense inherited by the SSA as necessary requirements of the SEE or STE. The costs associated with these requirements are usually up-front, one time costs. Personnel requirements on the other hand are much more subjective and depend on many variables. These variables in many cases are not even known until much later (e.g., PDSS concept, software volatility, software design, software quality, automated tools). Estimating PDSS personnel resource requirements is often a "best guess" endeavor. Many parametric models exist to assist PDSS personnel planners. Most models, however, do not consider all PDSS phases and activities, and none of them are intended to replace sound judgement or experience. In addition, there are two sources of information which can help; the CRISD and the LSA

Skills Analysis Task. The following approach offers some structure to the process of analyzing PDSS resource requirements.

70 Approach to PDSS resource analysis. This paragraph provides a very simple but generic sequence of four steps which will lead to complete identification of SSA resource requirements. Although this example only addresses the "personnel" resource category, the same general process can be applied to all five resource categories.

70.1 Determine assigned PDSS activities. PDSS activities and responsible organizations are identified in the detailed PDSS process model. The detailed PDSS model was modified during early planning to satisfy the requirements of the approved PDSS concept. This initial step in the PDSS resource analysis process simply involves each participating organization analyzing the detailed PDSS process model to determine the PDSS activities for which they are responsible.

70.2 Identify/estimate activity requirements. After assigned activities have been determined, the responsible organization is then ready to identify or estimate (in the case of personnel resource requirements) resources necessary to perform these activities. Each participating organization should identify resource requirements necessary to conduct each assigned activity in accordance with and at the level of effort indicated in the PDSS concept. The planned level of effort is an important consideration because it provides some indication of PDSS process frequency. Let's say for example the Head of the Configuration Management (CM) Section has been tasked to estimate personnel requirements for CM support of a designated system. Head CM first goes to the PDSS concept and the detailed process model to determine that:

a. Formal configuration audit activities are the responsibility of the Head, Configuration Management Section. This includes planning, organizing, conducting and follow-up for all functional and physical configuration audits.

b. Formal configuration audits will be conducted for every operational release and will be conducted only after system integration and testing has been successfully completed.

c. Operational releases are scheduled every two years for the entire life cycle.

With this and other information concerning the size of the program (or the size of the software to be sampled), data/test requirements, and staffing factors based upon past SSA experience, Head CM can estimate minimum personnel requirements for the configuration audit activity.

70.3 Integrate like requirements for assigned activities. At this low level in the estimating process, Head CM next looks at related activities assigned to the CM Section and, where possible, integrates like requirements within each resource category. This step involves PDSS CM planners working through the following considerations:

a. Based on the approved PDSS concept, configuration audits will occur approximately every two years. Head CM estimates that 2560 person-hours (eight people for two months) will be required to complete the activity. This is an infrequent requirement, which will not require permanent staffing, as opposed to a continuing requirement, which will require permanent staffing.

b. Next, Head CM examines the personnel requirements planning data for this system, and determines that the permanent staffing objective for the CM Section is three people. Head CM estimates that he can provide 640 person-hours from the CM Section and continue to provide CM support.

c. In addition to two people from the CM Section, Head CM needs:

1280 person-hours (four people for two months) from the Software Engineering Section

320 person-hours (one person for two months) from the Software Quality Assurance Section, and

320 person-hours (one person for two months) from the Software Test Section.

d. Once requirements have been identified and integrated, permanent staffing estimates and unsatisfied infrequent staffing requirements are forwarded.

70.4 Combine and integrate like requirements for all PDSS activities. After each participating organization has identified support estimates and unsatisfied requirements, the SSA then performs a final step to combine and integrate all requirements. This step is like the previous step except instead of assigned activities, the SSA considers the input from all participating organizations. In our example, SSA planners would refer to the Heads of Software Engineering, Software Quality Assurance and Software Test Sections to determine if the infrequent staffing requirement for configuration audits can be satisfied from permanent personnel resources. Once infrequent requirements have been satisfied, which may involve increasing permanent staffing estimates, they are combined and totaled. The SSA should always consider as a cost saving measure opportunities to integrate resources across projects. Ultimately, the product of this step is a complete list of requirements by resource category necessary to implement the approved PDSS concept and process.

80 PDSS process characteristics. In many aspects the PDSS process is unique. Before concluding our discussion of PDSS resource analysis, it is worthwhile to consider these unique characteristics, characteristics that need to be understood and considered by PDSS planners. Consider the following observations for example: (Note: Much of this discussion applies as well to initial software development.)

a. PDSS is a protracted process. The entire process, from one operational release to the next, can last months or even years.

b. The PDSS process is personnel intensive, and personnel cost is a significant factor in the PDSS cost equation.

c. The PDSS process can be iterative or recursive (DoD-STD-2167A states that activities in the software development process may overlap and may be applied iteratively or recursively.). Rework is the rule rather than the exception.

d. The PDSS process consists of a set of ordered activities each requiring unique skills and experience.

e. PDSS is an intellectual and often creative process. The time required for each task is often difficult to estimate.

f. Progress is difficult to quantify.

g. Inputs into the process (software change requests, ECPs) and the level of effort associated with each input are difficult to project.

h. The PDSS process is a "token" process, not a production process. In a production process, multiple units are in the process, and each unit is acted upon at the same time by a different group. Scheduling is easy and high levels of productivity can be achieved. In a "token" process there is only one unit in the process at a time, and the unit is acted upon sequentially by each group. Groups are idle when the "token" is not at their station. The token process is more difficult to manage and to achieve high levels of productivity. In the PDSS process, the "token" is the current software configuration. Consider for example, the test section cannot test a new software configuration until the engineering section has developed it, and it has been released for test.

90 Impact of PDSS process characteristics on personnel planning. Given the preceding complexities of the PDSS process, it should be no surprise that long-range estimates are more "best guess" than exact science. Even near term personnel estimates are difficult. PDSS planners should consider the following when estimating personnel requirements:

a. In practice, most PDSS planning tends to view PDSS as a steady state process wherein a constant level of effort is applied. Consequently, most schedules and milestones are usually based on steady state production rates. If schedules start to slip, adding

more people, especially new and untrained people, to the PDSS process will generally not improve near term productivity. In fact, productivity may be reduced due to new training requirements and increased management complexity.

b. Managers should avoid placing high risk tasks on the critical path. High risk tasks are tasks where the estimated time and personnel required to complete the task are difficult to project.

c. Within a single PDSS cycle, there is generally idle time to be considered (characteristic of a "token" process). Managers need to recognize this and plan accordingly. In some cases, slack-time can be applied to other programs. Combining and integrating personnel requirements at the program level or across program boundaries is an important step in reducing personnel requirements and in efficient PDSS operations.

d. Managers should incorporate personnel flexibility into staffing plans and be prepared to respond to changes in scheduling or to changes in the SSA's environment.

e. Managers should impose controls on the PDSS process which focus management attention on schedules, milestones, levels of effort applied to a task and software quality.

f. The use of quantitative or parametric models to estimate personnel requirements can be a useful tool, but these tools should be used with caution. Most popular models apply to software development which is only one phase of PDSS. All PDSS phases and activities must be considered.

g. An alternative to the "token" process is the "pipeline" approach. This requires rigorous configuration identification and control, but can reduce process slack-time. The pipeline

approach involves multiple software versions or developmental configurations in work simultaneously, each at different stages in the PDSS process. For example, one version can be in design, another version in code, and, yet another, undergoing system test. The key to success is to ensure that all changes are implemented and traceable in formal baseline updates.

90.1 Integrating personnel requirements. To facilitate the process of integrating personnel requirements, all activities in the detailed PDSS model were identified as management, technical or support functions. The following list categorizes specific PDSS skills by functional area.

a. Management functional areas

Project management

Configuration management

Budgeting

b. Technical functional areas

Systems engineering

Software engineering (requirements analysis, preliminary design, detailed design, coding)

Software quality evaluation

Software testing

System integration and testing

IV&V

Technical writing

c. Support functional areas

Logistics/supply

Computer equipment operations and maintenance

APPENDIX C

Miscellaneous equipment operations and maintenance

Document/magnetic media production

Facilities support

Training/training support

Packaging, handling, storage and transportation

Data management

Network/communications management

APPENDIX D

CRLCMP TOPICS

10. Purpose. The purpose of this appendix is to provide topics and amplifying information that should be contained in a Computer Resources Life Cycle Management Plan (CRLCMP).

20. General. The CRLCMP should be closely coordinated and may become part of the Integrated Logistics Support Plan. In addition to reflecting current plans, the CRLCMP provides a chronology of management decisions, assumptions, evolving support requirements and developments which impact software support. The CRLCMP is the source document for computer resource planning throughout the life cycle of the system. The following are subject areas covered in most CRLCMP's, and not a statement of DoD policy or reporting format. Consult the latest CRLCMP format used by your organization.

30. CRLCMP objectives. The objectives of the CRLCMP are to:

a. Provide a plan for controlling the acquisition, management, and support of MCCR throughout the acquisition process.

b. Document MCCR support concept agreements and plans for the orderly transition of computer resources from the developing agency to the SSA.

c. Document procedures for applying various systems engineering and management disciplines (e.g., CM, quality evaluation, testing, security) throughout the life cycle.

d. Identify the resources required to acquire, manage and support MCCR.

e. Identify the organizational roles, responsibilities, and interrelationships of all agencies involved in acquiring, managing and supporting MCCR.

f. Document the level of independent verification and validation to be applied to software in MCCR.

40. CRLCMP Topics. The following topics, as applicable, should be addressed in the CRLCMP for MCCR items (operational hardware

and software, automatic test equipment, trainers, etc. in accordance with the scope of the CRLCMP.).

40.1 Introduction

a. A brief statement of the purpose of the CRLCMP.

b. The original version of the CRLCMP should be identified, and a brief discussion of how changes will be approved and implemented.

c. Complete identification of the system to include formal and common names, nomenclature, identification number and system abbreviations.

d. Identification of subsystems and interfacing systems.

40.2 System

a. Describe the operational mission of the system to include mission need and employment.

b. Identify interoperability requirements.

c. Describe system functions.

d. Describe technical requirements and provisions for system growth and excess capacity of current configuration.

e. Identify probable changes in the system necessary to accommodate new operational or environmental requirements.

f. Identify planned product improvements.

g. System description. Describe the system to include descriptions of: system architecture, components and interfaces; MCCRs (mission hardware and software). Use separate subparagraphs to describe each subsystem and major MCCR component.

h. Describe the system support concept, including the PDSS concept. Identify the Integrated Logistics Support Plan (ILSP).

i. Address hardware issues (e.g., quality assurance, configuration management, warranties, interoperability and testing) as applicable.

j. Analyses. Describe or refer to the results of any early analyses conducted (e.g., hardware/software tradeoff studies, support analyses, risk analysis).

k. Identify the user(s), developing and supporting agencies, to include the SSA.

l. Identify security requirements to include:

1. Any special facility security procedures or requirements and the impact;

2. Any special computer resources security procedures or requirements and the impact; and

3. The possibility of foreign sales, hardware and software that is not releasable to foreign countries, and support concept implications.

m. Identify major MCCR milestones to include:

1. Contractual milestones,

2. Acceptance evaluation(s) and delivery date(s),

3. OT&E schedule,

4. Production schedule,

5. System initial operational capability, and

6. Planned product improvements.

40.3 Initial Software Development and Software support

a. References and standards. Identify references and standards that apply to software development, transition and PDSS.

b. PDSS concept. Describe the PDSS concept to include the PDSS objective, risk analyses (technical, cost and

schedule), and a description of the detailed PDSS process that includes identification of organizational agencies responsible for performing each PDSS activity (e.g., impact analyses, detailed problem analysis, production of manuals).

c. Organizational roles and responsibilities. Identify and discuss the roles and responsibilities of all agencies involved in software development, transition and PDSS.

d. Boards and committees. Identify all boards and committees created to manage computer resources during initial software development, transition and PDSS.

e. Financial planning. The CRLCMP should contain or refer to documents which contain financial planning information, planning procedures, historical cost data (e.g., developmental costs), budgets, etc.

f. PDSS acquisition requirements. Identify or refer to PDSS acquisition requirements imposed during initial software development (e.g., Government standard hardware, software or programming languages).

g. Software development methodologies. Describe the software development process to include specific software engineering methods, techniques and management controls. The description should include criteria for measuring progress. The Software Development Plan should be identified.

h. Software milestones. Identify initial software development, transition and PDSS milestones to include software:

1. Contractual milestones,

2. Technical reviews and audits,

3. Transition period and transfer date,

4. Test schedules,

5. Support period, and

6. Planned releases and deliveries

i. Supportability demonstration. Discuss plans to conduct a supportability demonstration prior to software transfer.

j. Software engineering environment (SEE). Identify and describe all components of the SEE (software, hardware, facilities, personnel and other resources).

k. Software test environment (STE). Identify and describe all components of the STE (software, hardware, facilities, personnel and other resources).

l. MCCR acquisition. Describe how each MCCR component will be acquired (e.g., competitive development, non-developmental procurement, product improvement).

m. Software configuration management. Identify all approved plans, requirements and agreements which govern software configuration management activities during initial software development, transition and PDSS.

    1. Describe the developing agency's/SSA's software status accounting procedures and policies.

    2. Describe the developing agency's/SSA's software configuration identification procedures and policies.

    3. Describe the developing agency's/SSA's software configuration control process.

    4. Identify and describe the results of all software configuration audits.

    5. Identify and describe plans for transitioning configuration management from the developing agency to the SSA.

n. Data management. Describe the plans and procedures for document preparation, update, control and distribution during initial software development, transition and PDSS.

o. Software distribution and installation. Describe the plans and procedures for distribution of software

(magnetic media and firmware) during initial software development, transition and PDSS.

p. Software quality program. Describe the software quality program, responsibilities, related activities and evaluation techniques employed by the developing agency's/SSA's software quality group. All related plans (e.g., Software Development Plan, Software Quality Program Plan) should be identified.

q. Software quality requirements. Identify software quality requirements to be implemented.

r. IV&V requirements. If applicable, identify IV&V agents and activities to be performed, schedules and supporting organizations. The results of IV&V activities should be referenced and summarized. Identify and describe the transition of IV&V products from the IV&V agent to the SSA.

s. Test and evaluation. Identify and describe the scope, nature, and organization for testing during initial development and PDSS. All related test plans, reports, models, and requirements should be identified.

t. Security requirements. Identify all security requirements and responsibilities, to include the Government agency responsible for obtaining security certification and accreditation.

u. Warranty, data rights and license requirements. Describe contract provisions which ensure the government's rights concerning the delivered software and data. Include references to any required licensing agreements or warranties in this paragraph.

v. SSA resource requirements. Identify all SSA resource requirements (software, hardware, facilities, personnel and other resources) necessary to implement the PDSS concept. All supporting information, assumptions, models and calculations should be included or referenced. SSA resource requirements should be identified as satisfied or unsatisfied requirements.

w. Training. Identify transition and PDSS training requirements and approved training plans.

APPENDIX E

SOFTWARE TRANSITION

10 Purpose. The purpose of this appendix is to discuss and provide a checklist for software transition.

20 General. The single objective of software transition is for the Government to acquire the resources, data and knowledge to be able to fully implement the approved PDSS concept. This usually involves a transition from the agency performing initial software development to the SSA; Government acquisition of other SSA resource requirements; training; and construction or modification of the facilities necessary to perform PDSS. Transition may occur over a period of a few days or it can cover an extended period. Transition plans, developed during the pre-deployment software support stage, should be complete and detailed. The SSA must plan for transition to ensure that all requirements and activities are included and closely coordinated. Separate transition plans for each subsystem may be combined into a set of multi-volume plans for a major weapon system.

30 Software transfer. Software transfer, the final transition milestone, is when the SSA assumes responsibility for PDSS. Generally, software transfer occurs before acceptance and after software development functions and resources have been passed to the SSA. At a minimum, the SSA should have assumed responsibility for all critical activities. (See Appendix B, paragraphs 50.1 and 50.2). Software transition and transfer must integrate with program planning to ensure system support during the transition period. When software transfer occurs the SSA should be fully capable of conducting PDSS operations.

40 Transition scheduling. If possible, software transition should be completed before system deployment. The very early phases of PDSS are often characterized by an initial wave of software change proposals and requirements from the using community for software support. It is especially important at this point in the evolution of software support that the SSA establish credibility and a "support" relationship with the using community. On-site training provided by the SSA coincident with initial fielding is an excellent way to establish rapport and lines of communication. Focusing on the SSA's user support role can be difficult if the SSA is still trying to complete transition. Therefore, when deployment begins transition should be complete and PDSS resources should be ready to respond to the needs of the using community.

APPENDIX E

50 Software support transition plan. DI-E-7142 contains a proposed software support transition plan outline.

60 Transition planning checklist. The following list contains considerations that should be made when reviewing the transition plan.

1. Does the transition plan completely and correctly implement the approved PDSS concept?

2. Have significant software and SSA transition milestones been identified?

3. Have all applicable PDSS management plans been implemented and documented (e.g., quality assurance, configuration management, test)?

4. Are the milestones from the software transition plan, the SSA transition plan and the contract delivery schedule properly coordinated?

5. Will transition milestones be performed and completed while the software developing contractor is still under contractual obligation to the Government?

6. Is the transition schedule reasonable and has sufficient time been allowed to ensure that transition will be completed before the first systems are fielded?

7. Has sufficient slack-time been provided in the schedule to handle unanticipated circumstances?

8. Have activities on the critical path been identified in the transition plan? Have planners avoided placing high risk transition activities on the critical path?

9. Have management controls been put into place which highlight anticipated scheduling problems?

10. Does the SSA's transition plan include adequate procedures for monitoring the progress of external organizations with a PDSS responsibility?

11. Has PDSS planning included all security procedures and safeguards required by the Security Classification Guide?

12. Have system unique training requirements been identified and addressed?

70 Software transition checklist. The following list contains items that should be considered as milestones in the transition schedule and verified before the SSA assumes PDSS responsibility.

1. Has SEE and STE hardware been delivered, installed and tested, including all special or unique interface hardware requirements?

2. Has SEE and STE software been delivered, installed and evaluated?

3. Have licensing agreements been properly transferred for commercial software?

4. Have documents, notebooks and manuals necessary for the software development process been identified and turned over to the SSA?

5. Have change status reports been delivered and are ECPs/software change requests available?

6. Have formal configuration audits been completed?

7. Have identified discrepancies been accounted for and properly documented in the corrective action system?

8. Has the current configuration identification (i.e., the functional, allocated and product baselines plus approved changes) been delivered to the SSA?

9. Have version description documents been turned over which properly identify each Government CSCI baseline?

10. Do version description documents properly identify each SCN with cross references to implemented ECPs and software change requests?

11. Has a Computer Software Configuration Index been delivered for each CSCI (MIL-STD-483A)?

12. Are sufficient personnel with the skills and experience necessary available to sustain the projected level of effort?

13. Has functional area and system specific training identified in the transition plan been completed?

14. Has adequate office, work and storage space been identified to support PDSS operations? Has adequate secure office, work and storage space been identified to satisfy physical security requirements? Has appropriate accreditation approval been obtained?

15. Are adequate utilities available (e.g., electrical power, air conditioning, heat) to support PDSS operations?

16. Has computer cabling been properly laid, connected and tested?

17. Has network wiring been properly laid, connected and the network tested?

18. Has a Software Transition Board been identified, with SSA and contracting agency members representing each functional area, to manage transition and to resolve transition issues?

19. Has the complete software development and generation process been exercised and the SSA's ability to generate and support the operational software in accordance with the PDSS concept been demonstrated?

20. Have software change requests and other discrepancy reports been turned over by the developing agency?

21. Has each software change request been classified by category and priority and accounted for in the SSA's corrective action system?

22. Are PDSS management procedures, techniques and controls in place and understood?

23. Does the Government have the necessary data rights to commence PDSS operations?

24. Have the computer system operator's manual and software user's manual been accepted, mass produced and available for issue?

25. Have provisions of the ILS plan and the PDSS concept been accomplished and coordinated with the Program Manager?

26. Have system fielding plans, which include provisions for software distribution, been approved, promulgated and funded?

27. Have training and diagnostic/maintenance devices been developed and are they ready for production and distribution?

28. Have support contracts, as appropriate, been approved, funded and implemented?

29. Has the PDSS budget been approved and is it adequate to support PDSS operations at the projected level of effort?

30. Has the software replication facility (equipment, media, etc.) been installed?

31. Have hardware and software maintenance responsibilities been identified?

32. Have all MCCR security requirements been met?

CONCLUDING MATERIAL

Custodians:
Army - CR
Navy - EC
Air Force - 10, 70

Review activities:
Army - AM, CR, ER, MI, AR
Navy - AS, EC, SH, SA, MC, CG, OM
Air Force - 1, 10, 11, 70, 71, 80, 82, 84, 89
DLA - DH

Preparing activity:
Air Force - 10

(Project MCCR-0030)

☆U.S. GOVERNMENT PRINTING OFFICE: 1998 - 704-034/30716