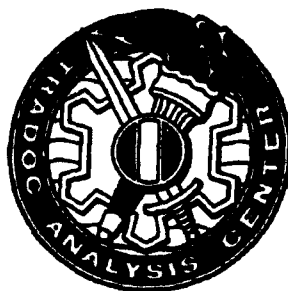


AD-A280 212



①

**NATIONAL MISSILE DEFENSE
POST DEPLOYMENT SOFTWARE SUPPORT:**

**A METHODOLOGY FOR MANPOWER AND
PERSONNEL REQUIREMENTS ASSESSMENT**

May 1994

DTIC
ELECTE
JUN 13 1994
S G D

Submitted to:

U.S. ARMY TRAINING AND DOCTRINE COMMAND (TRADOC)

ATTN: ATCA

Fort Eustis, Virginia 23604-5538

WPF **94-17966**

Prepared by:

AEPCO, Inc.

15800 Crabbs Branch Way

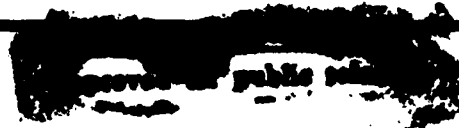
Suite 300

Rockville, Maryland 20855-2622

**DYNAMICS RESEARCH
CORPORATION**

60 Concord Street

Wilmington, Massachusetts 01887-2193



DTIC QUALITY INSPECTED 1

94 6 10 071

REPORT DOCUMENTATION PAGE

Approved
B No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments, regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE May 1994		3. REPORT TYPE AND DATES COVERED Final Technical Report (CDRL A007)	
4. TITLE AND SUBTITLE National Missile Defense Post Deployment Software Support: A Methodology for Manpower and Personnel Requirements Assessment				5. FUNDING NUMBERS #DABT60-90-D-0010	
6. AUTHOR(S) Hemenway, Mark (DRC) Lafond, Ronald H (AEPCO, Inc.)					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Dynamics Research Corporation (DRC) 60 Concord Street Wilmington, MA 01887-2193				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Training & Doctrine Command (TRADOC) ATTN: ATCA Fort Eustis, VA 23604-5538				10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES					
12a. DISTRIBUTION / AVAILABILITY STATEMENT Distribution Statement A: Approved for Public Release; Distribution is Unlimited				94-17966 	
13. ABSTRACT (Maximum 200 words) The purpose of this study was to define a methodology for estimating manpower and personnel requirements for Post Deployment Software Support (PDSS) of Department of Defense (DOD) software systems. This study was initiated by the United States Army's Training and Doctrine Command (TRADOC) Analysis Center (TRAC) at Fort Lee (TRAC-LEE), Virginia. The Statement of Work (SOW) required the AEPCO/DRC Team to accomplish the following: (1) Develop a methodology for determining PDSS manpower and personnel resource requirements and training prerequisites; (2) Outline a procedure for implementing this methodology; and (3) Discuss the practical application of the methodology to the Army's National Missile Defense (NMD) System. Typically, PDSS accounts for more than two thirds of the total lift cycle cost for software within the DOD. This study defined a methodology to assist program managers within DOD in estimating PDSS manpower and personnel (MP) requirements and training standards in order to mitigate life cycle software engineering (LCSE) costs.					
14. SUBJECT TERMS Manpower, Personnel and Training (MPT) Manpower & Personnel Integration (MANPRINT) Post Deployment Software Support (PDSS)				15. NUMBER OF PAGES 30	
				16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT		

A B S T R A C T

1. PURPOSE. The purpose of this study was to define a methodology for estimating manpower and personnel requirements for Post Deployment Software Support (PDSS) of Department of Defense (DOD) software systems. This study was initiated by the United States Army's Training and Doctrine Command (TRADOC) Analysis Center (TRAC) at Fort Lee (TRAC-LEE), Virginia. The Statement of Work (SOW) required the AEPCO/DRC Team to accomplish the following: (1) Develop a methodology for determining PDSS manpower and personnel resource requirements and training prerequisites; (2) Outline a procedure for implementing this methodology; and (3) Discuss the practical application of the methodology to the Army's National Missile Defense (NMD) System. Typically, PDSS accounts for more than two thirds of the total life cycle cost for software within the DOD. This study defined a methodology to assist program managers within DOD in estimating PDSS manpower and personnel (MP) requirements and training standards in order to mitigate life cycle software engineering (LCSE) costs.

2. TECHNICAL APPROACH. The PDSS manpower requirements methodology was applied to the NMD System to assess the validity of the technical approach. The PDSS manpower requirements were then translated into respective personnel occupational specialties (POSSs). Recommended personnel skills, knowledge, and abilities (SKAs) and training prerequisites were also defined for these job specialties. This methodology was reviewed by contractor and government Subject Matter Experts (SMEs) for validity with feedback provided.

3. PDSS MANPOWER REQUIREMENTS ESTIMATION METHODOLOGY (PRAM). The PRAM was applied to the Command and Control and Fire Control components of the NMD system as a model excursion. PDSS manpower requirements were calculated as a function of random workload generators (e.g., the number, complexity, function, and size of the lines of code).

4. PERSONNEL AND TRAINING ATTRIBUTES. The objective of the PDSS Personnel and Training Analysis was to identify, at a high level, the various POSSs derived from the PRAM. Prerequisite SKAs were determined for the following computer POSSs: systems

engineer, systems analyst, software engineer, computer programmer, computer operator, computer repairman/maintainer, and field technician. Training prerequisites for the PDSS computer POSSs included both formal education categories of courses and types of degrees as well as informal on-the-job training (OJT) programs.

5. CONCLUSIONS AND RECOMMENDATIONS. Areas of the PRAM that warrant additional research and study include the following:

A. Conducting a random sampling of PDSS workload generated in conjunction with major DOD system acquisitions. This study would validate error, requirements-based and technology-based software change request (SCR) volume relationships over time. This validation would improve the reliability of the PRAM and enhance its value to DOD Program Managers (PMs) as a tool to estimate the quantity and quality of manpower and personnel needed.

B. Researching the duties and functions of several DOD users since they are the key element in the process who write the SCRs in response to changes in doctrine and policy.

C. Performing a trial run excursion of the PRAM using an actual U.S. Army materiel system for validation purposes.

D. Developing a PM guide for applying the PRAM for estimating PDSS MP requirements and determining training prerequisites.

NATIONAL MISSILE DEFENSE POST DEPLOYMENT SOFTWARE SUPPORT:

A METHODOLOGY FOR MANPOWER AND PERSONNEL REQUIREMENTS ASSESSMENT

May 1994

FINAL TECHNICAL REPORT

(CDRL A007)

Submitted to:

U.S. ARMY TRAINING AND DOCTRINE COMMAND (TRADOC)

ATTN: ATCA

Fort Eustis, Virginia 23604-5538

Prepared by:

AEPCO, Inc.

**15800 Crabbs Branch Way
Suite 300
Rockville, Maryland 20855-2622**

**DYNAMICS RESEARCH
CORPORATION**

**60 Concord Street
Wilmington, Massachusetts 01887-2193**

Accession For	
NTIS	CRA&I ☒
DTIC	TAB ☐
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and / or Special
A-1	

PREFACE

The purpose of this study was to define a methodology for estimating manpower and personnel requirements for Post Deployment Software Support (PDSS) of Department of Defense (DOD) software systems.

This study was initiated by the United States Army Air Defense Artillery School (USAADASCH) in connection with National Missile Defense (NMD) Issues.

Typically, PDSS accounts for more than two thirds of the total life cycle cost for software within the DOD. This study defines a methodology to assist program managers within DOD in estimating PDSS manpower, personnel, and training (MPT) requirements in order to mitigate life cycle software engineering (LCSE) costs. The PDSS manpower requirements methodology was applied to the Army's National Missile Defense (NMD) System to assess the validity of the technical approach. The PDSS manpower requirements were translated into respective personnel occupational specialties. Recommended personnel skills, knowledge, and abilities (SKAs) and training prerequisites were defined for potential PDSS occupational specialties. Conclusions and recommendations for further PDSS study are listed in the final chapter. This methodology was reviewed by contractor and government Subject Matter Experts (SMEs) to provide an initial check on the soundness of the concepts and proposed procedures.

The principal findings of the analysis are contained in this report. The appendices contain supportive information and reference data. This report will be on file with the Defense Technical Information Center (DTIC).

The AEPCO/DRC Team Leader was Ron Lafond. Mark Hemenway was the Principal Investigator for the study with support provided by the following contributors:

☐ Senior Analyst	Dan Risser
☐ MPT Analyst	Steve Crump
☐ Graphic Artist	Scott Marshall
☐ Technical Editor	Cheryl Lincoln

EXECUTIVE SUMMARY

This Technical Report was prepared by Advanced Engineering & Planning Corporation (AEPCO)/Dynamics Research Corporation (DRC) for the United States Army Training and Doctrine Command Analysis Center (TRAC), Ft Lee, Virginia as CDRL A004 under Contract Number DABT60-90-D-0010.

The objective of the study was to develop a methodology for estimating the manpower and personnel requirements for Post Deployment Software Support (PDSS) and apply this methodology to the Army National Missile Defense (ANMD) System. This report:

- ☐ Defines a conceptual basis for estimating PDSS manpower and personnel requirements;
- ☐ Outlines a procedure and methodology for implementing that concept; and
- ☐ Discusses the application of the methodology to ANMD.

PDSS includes the planning, design, programming, installation, and testing of enhancements and modifications to software after fielding. PDSS is costly. It accounts for over two thirds of the lifecycle cost of software systems. In a world which is increasingly dependent on software, PDSS is also essential to maintaining system availability and effectiveness.

Methodologies and tools now used to estimate PDSS manpower and personnel requirements treat PDSS as an extension of the software development process. They do not recognize the unique nature of PDSS tasks and workload. The most popular tools are the COConstructive COst Model (COCOMO) and its derivative REVised Intermediate COCOMO (REVIC) methodologies. These models both estimate PDSS requirements as a function of development effort and program size.

The PDSS Requirements Assessment Methodology (PRAM) applies task and workload-based manpower and personnel requirements assessment techniques to assist program managers in estimating PDSS Manpower requirements.

PDSS is an iterative process that involves the repetitive execution of common tasks, without regard to the type of modification or software. These tasks are:

- ☐ Change Request Preparation and Management
- ☐ Impact Analysis
- ☐ System Release Planning
- ☐ Execution (Design, Program, Code)
- ☐ Test
- ☐ Install and Implement

The PDSS process is based on the creation and execution of Software Change Requests (SCR) generated by the users. Each SCR is evaluated and prioritized by the PDSS authority. Once approved, the SCR is executed, tested, and sent to the field for installation and implementation.

Workload is the product of task frequency and performance time. PRAM estimates PDSS requirements by calculating PDSS task frequency as a function of SCR volume, and performance time by PDSS task. Workload is then allocated among job categories, and personnel requirements are calculated.

PRAM is summarized as follows:

- Step 1.* Determine PDSS task frequency. Error correction rates, technological and environmental changes, and user generated changes are used to estimate SCR volume.
- Step 2.* Determine Mean Level of Effort per PDSS task. Task performance times form a distribution about an average value. The shape of the distribution depends on several factors associated with the task and the individual under consideration.
- Step 3.* Calculate Workload per Task. Workload is the product of task frequency and mean level of effort per task.
- Step 4.* Total workload per task is allocated or assigned among job categories to produce workload per task per job category. Allocated workload is summed by job category.
- Step 5.* Personnel availability and capacity rates are applied to workload to calculate personnel requirements by job category.
- Step 6.* Supervisory and Administrative Requirements are calculated and added to the total.

Sample job categories are presented and qualifications and training requirements are discussed in the Personnel and Training Analysis section.

Although constrained by lack of detailed data to a high level assessment, the PRAM is applied to the NMD System to assess the validity of the technical approach.

TABLE OF CONTENTS

Introduction	1
Objective	1
Scope	1
Background	1
Definitions	2
PDSS	2
Software Maintenance	3
Technical Approach	4
PDSS Tasks and Functions	4
PDSS Environment	4
Configuration Management	6
PDSS Cost & Resource Estimation Tools	6
OVERVIEW	6
SLIM	7
PRICE S/SL	7
COCOMO	7
REVIC	10
REVIC and COCOMO	10
Post Deployment Software Support Requirements Assessment	
Methodology (PRAM)	13
General	13
Overview	13
Tasks	14
Determine Workload	15
Allocate Workload	15
Manpower Determination	16
Methodology	16
General	16
Procedure	17
Army National Missile Defense (ANMD) PDSS Assessment	27
Objective	27
System Description	27
Methodology Application	28
Assumptions	29
Step 1. Determine Change Request Volume	29
Step 2. Determine Mean Level of Effort	32
Step 3. Calculate Workload	34
Step 4. Allocate Workload	34
Step 5. Calculate Personnel Requirements	34
Step 6. Determine Overhead	36
Step 7. Analysis Results	36
Conclusions and Recommendations	37
Conclusions	37
Recommendations	37
Appendix A—Acronym List	A-1

TABLE OF CONTENTS (CONTINUED)

Appendix B—References	B-1
Appendix C—Personnel and Training Analysis	C-1
Overview	C-1
Personnel and Training Analysis Assumptions and Constraints	C-1
Personnel Occupational Specialties, Qualifications, and Training	C-1
Systems Engineer	C-1
Systems Analyst	C-2
Software Engineer	C-3
Computer Programmer	C-4
Computer Operator	C-5
Computer Repairer/Maintainer	C-6
Field Technician	C-7
Personnel Specialties Versus PDSS Tasks	C-9

LIST OF FIGURES

1. Software Life Cycle Cost Distribution	2
2. Software Change Request Process	5
3. The Rayleigh Distribution	8
4. The Manpower and Personnel Requirements Determination Process	14
5. The PDSS Manpower/Personnel Requirements Assessment Methodology	16
6. Error Correction Change Request Volume	18
7. Environmental/Technological SCR Volume	20
8. Mission Essential Requirements—Based Change Request Volume	22
9. Normal Distribution for Level of Effort Per Task	23
10. Mean Level of Effort—Impact Analysis	24
11. Site Layout	28
12. ANMD SCR Volume by Category and Total	31
13. Total SCR Volume	32
14. Mean Level of Effort Distribution By Task	33

LIST OF TABLES

1. COCOMO Cost Driver Attributes	9
2. PDSS Tasks	15
3. PDSS Methodology Overview	17
4. Software Change Request Volume Factors	18
5. The Workload Allocation Process	26
6. Demonstration Workload Calculation—ANMD	35
7. Demonstration Workload Allocation—ANMD	35
8. Demonstration Calculation, Personnel Requirements By Position	36
9. Occupational Specialties and PDSS Tasks	C-9

INTRODUCTION

OBJECTIVE

The objective of this study is to define a methodology for estimating manpower and personnel requirements for Post Deployment Software Support (PDSS).

SCOPE

The complete development of a methodology for estimating PDSS requirements is well beyond the scope of this study. With the resources allocated to this effort, we have:

- ☐ Established the scope of PDSS activities for which human support workload must be assessed.
- ☐ Identified factors that drive the workload for each of these functional activities.
- ☐ Established an analytical approach for estimating workload in each function.
- ☐ Discussed the potential application of the methodology to the National Missile Defense (NMD) system.

BACKGROUND

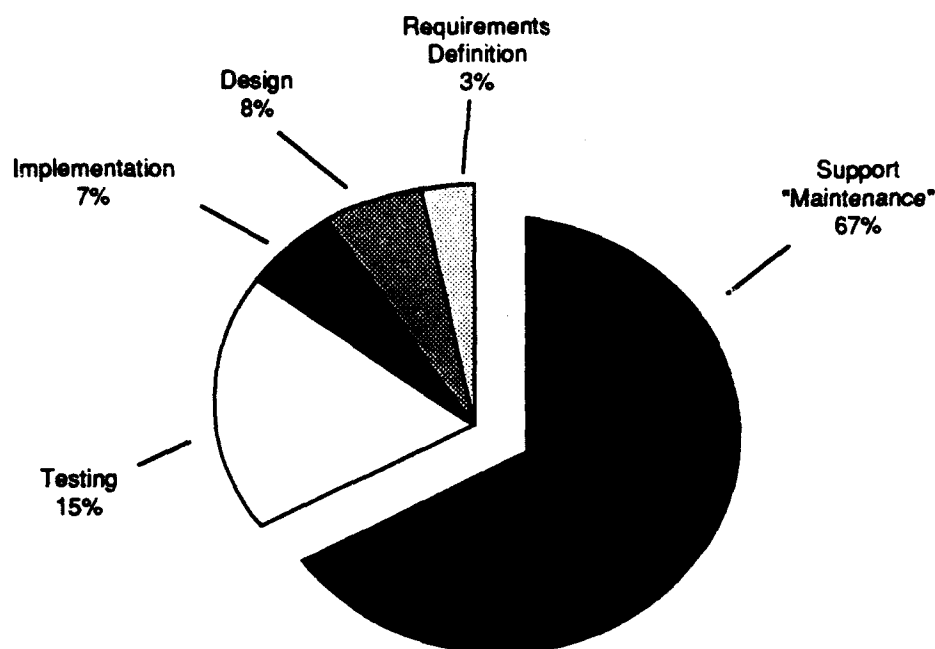
Software support does not end when a system is fielded; rather, it continues throughout the system's lifecycle. Continuous revision, update, and repair are required to achieve and maintain effectiveness and relevance.

PDSS is critical. It is also costly. PDSS accounts for more than two thirds of the total lifecycle cost of DoD software¹, as shown in Figure 1. This is an area of particular concern now, as the Services struggle to keep up with changing requirements. That concern will only increase as plans for digitization of the combat environment are implemented.

The PDSS process is not well understood, and planning for and managing PDSS is a challenge for program managers. Resource requirements and workload are difficult to forecast. Functions and activities are difficult to define and organize. Factors that drive these functions and activities are difficult to define and measure. Costs are difficult to control. Frequently, PDSS is performed outside of the Program Manager's purview, at contractor facilities or at specialized, dedicated government facilities.² Finally, the entire PDSS process occurs in an environment of heavy demand, conflicting priorities, and limited funding.

¹ Caro, I., Higuera, R., et al.

² Caro, I., Higuera, R., et al., pp.7-8, 7-3.



Source: Arthur, 1988, provided by DSMC, 1992

Figure 1. Software Life Cycle Cost Distribution

The techniques and methodologies used to estimate PDSS support requirements are largely based on the concepts and approaches used to estimate software development costs. These, however, fail to consider the unique aspects of the post deployment support function. A new approach is needed.

PDSS is of particular interest to the NMD system which will rely heavily on complex Command and Control (C2), fire control, and target acquisition software systems. It will be required to perform under extreme reliability and availability requirements to perform a critical defense function. The NMD post deployment support environment will be austere, and the accurate estimation of PDSS will be critical to the program's supportability.³

DEFINITIONS

Manpower - The quantity of individuals required to operate or maintain a system.

PDSS. This includes "all activities required to ensure that, during the production/deployment phase of a mission critical software system's life, the implemented and fielded software/system continues to support its original operational mission and subsequent mission modification and production

³ NMD Integrated Logistics Support Plan, p. 2-44.

improvement efforts." ⁴ It begins during Phase III (Production and Deployment) of the acquisition/development process and continues through Phase IV, Operation and Support.

Immediately on fielding, a number of factors begin generating PDSS requirements. These requirements can be classified as ⁵:

- ❑ *Error Correction.* An error is a logical mistake in the software code which results in an operational mission event that deviates from stated system requirements. Due to the size and complexity of modern software systems, errors are unavoidable in even the most carefully managed development projects. Testing is designed to surface the most serious and most common errors. However, as a system is operated, new modes of operation are exercised and unpredictable errors or bugs emerge. Error correction accounts for the smallest proportion of PDSS support requirements.
- ❑ *Technology/ Operating Environment.* A number of external influences create demands for PDSS. Operating systems evolve, host hardware is updated or changed, and Commercial Off-the-Shelf Software (COTSS) packages are updated or changed. Existing software must be modified to reflect these changes. This PDSS category is large and growing larger.
- ❑ *User Requirements.* User requirements change and evolve after fielding. User experience with the system generates new uses, and uncovers inefficiencies or human factors shortfalls. New missions, doctrine or operational pressures generate new user needs and requirements. This category of PDSS represents the largest proportion of support workload.

Personnel - The qualification or characteristics (skill, knowledge, and aptitudes) required by system operator or maintainer positions.

Software Maintenance. This term is often used to describe PDSS. We do not recommend the use of this term, and will avoid using it in this report. The word "maintenance" implies hardware-like failure behavior and an emphasis on repair. This can be confusing when discussing software. First, software does not break in the same way that hardware does. Although software errors are discovered and fixed, they do not reoccur and they are not random or dependent on the length of operation. In addition, PDSS is much broader than the word "maintenance" implies. In fact, the correction of errors is the smallest part of PDSS workload⁶.

⁴ Caro, I., Higuera, R., et al., p. 7-5.

⁵ United States Department of Commerce/National Bureau of Standards, p. 5.

⁶ Piersall, J., p. 38.

TECHNICAL APPROACH

The methods and approaches now used to estimate PDSS support requirements are often perceived as inadequate in both concept and execution. This study proposes an alternative approach that reflects the unique characteristics of the PDSS function. This approach focuses on describing the average task and its associated completion time and average rate of occurrence rather than attempting to predict specific failures and maintenance requirements.

PDSS TASKS AND FUNCTIONS

PDSS Environment

Software development is a linear process. A series of related activities are performed sequentially. Development tasks and workload are directly related to design requirements and challenges. PDSS is a repetitive, concurrent execution of a process. This process is based on the management and execution of software change requests (SCR)⁷. Simply stated, SCRs are prepared and submitted by the user, evaluated and approved by the system support management structure, completed by the PDSS organization, and installed by the user or field support group. These tasks are repeated for each software change. Total numbers and types of change requests are a function of the PDSS environment; they are, however, independent of the development effort.

Actual software programming and design are only two aspects of PDSS. A large portion of PDSS workload is accounted for by administration, and control of changes.⁸

The SCR process is shown in Figure 2. Major tasks are shown in a box. Important subtasks are displayed immediately below the box. The process is described as follows⁹:

Prepare Change Request. This task involves the user preparing and submitting an SCR, deficiency report, software improvement request, etc. It also includes receipt and processing of the request by the PDSS organization.

Prepare Change Request
Process Request

Conduct Impact Analysis. After the PDSS organization receives and processes the SCR, it must be assessed for feasibility, and impacts on cost, software architecture, functionality, data bases, code, other systems, etc.

Review Change Request
Identify Impacts

⁷ Arthur, L.J., p. 7.

⁸ Cline, R.

⁹ Arthur, p. viii.

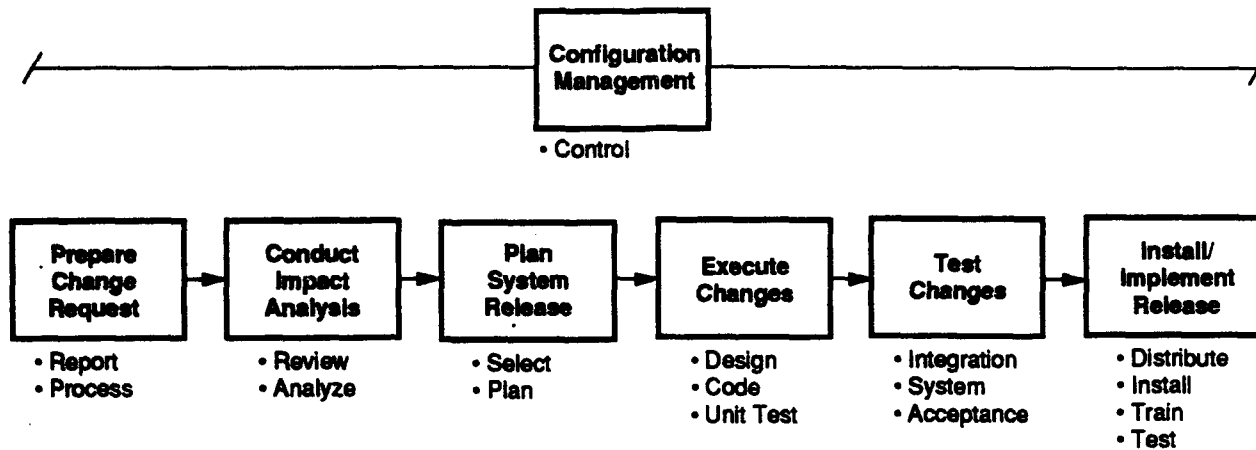


Figure 2. Software Change Request Process

Plan System Release Planning. The review authority and user representative select and prioritize SCRs for execution. At this point, non-emergency changes are bundled to create a software release package.

Review Change Request
Rank and Select Change Requests
Design Change Release
Prepare/Release Documentation and Test Plan

Execute Change. This task includes designing, programming, and coding the SCR. Requirements and problems are defined, design solutions are developed, code is written, unit testing is performed, and documentation is updated. In the current micro-computer environment, these steps are often performed concurrently. Design solutions may be developed during the Impact Analysis Phase. Change execution support requirements are adequately estimated by existing software development estimation tools and methods.

Design
Problem Definition
Hypothesis Development and Test
Program Design
Module Design
Data Design
Design Code
Write Code

- Code Walk Through
- Unit Test
- Update System Software Documentation

Test Changes. Modifying existing programs involves not only modifying the code itself, but also its links with other modules within and without the system. Thorough testing of software changes is essential to avoid introducing new errors or problems into the program, and to make sure integration with the existing system is complete.

- Develop Test Plan
- Develop Test Procedures and Cases
- Integration Test
- System Test
- Acceptance Test

Install/Implement Release. Once the SCR is complete and tested, it must be packaged, transferred to the appropriate media, and distributed to user sites for installation. Installation can be a significant workload factor if there are many installation sites, if the installation process is complex or difficult. User training must also be considered.

- Package Release
- Document Release
- Deliver Release
- Install Release
- Test Release
- Train Users

Configuration Management. Configuration management is performed throughout the PDSS process. It is used to establish and maintain control of PDSS process.

- Define Configuration

These tasks define the PDSS process and are always executed for all types of SCRs.

PDSS COST & RESOURCE ESTIMATION TOOLS

OVERVIEW

Analytical tools and models have been developed to estimate software development resource requirements. These tools and models have been adapted, or include modules which estimate PDSS requirements. Several models are discussed below. Two models, Constructive Cost Model (COCOMO) and its derivative REVISED Intermediate COCOMO (REVIC) are currently much used within the Department of Defense.

SLIM

The Software Lifecycle Management (SLIM) model is a project planning model used to estimate lifecycle cost, manning, and time-to-complete for software development projects. Manning levels calculated by SLIM are derived from the Rayleigh Distribution of project personnel over time.¹⁰

The user enters historical data from completed development projects to establish organizational productivity factors. These factors and the descriptive characteristics of the software determine the shape of the Rayleigh curve. Tradeoff and sensitivity analyses can be conducted against schedule, cost, effort, risk, reliability, peak manpower, and size. Linear programming techniques can be applied to determine optimal resource allocations.¹¹

The Rayleigh Distribution is a mathematical distribution (Figure 3) of manning levels over the duration of the project. It has been shown to¹² give a close approximation of manning levels for R&D projects and some software development projects.

SLIM treats PDSS as an extension of the development phase, and it addresses only error correction PDSS.¹³

PRICE S/SL

PRICE S and PRICE SL are parametric models of software development and operating and support costs. Historical data is used to establish curves for various parameters. These parameters include software type, complexity, platform characteristics, error rates and repair, post deployment growth, and others.¹⁴

This parametric approach to software cost estimation assumes the continuing applicability of historical data to new systems and is dependent on the availability and quality of that historical data.

COCOMO

COCOMO is one of the most used software development cost models. The objective of COCOMO is to estimate total effort in man-months to develop a software product. Lines of code is the basis for resource estimation¹⁵.

¹⁰ Boehm, B.W., p. 513.

¹¹ Quantitative Software Management, Inc., p. 7-3 ff.

¹² Boehm, p. 68

¹³ Quantitative Software Management, Inc. p. 7-40

¹⁴ Fugate, C.S., pp. 5-7.

¹⁵ United States Air Force Cost Center, p. 1.

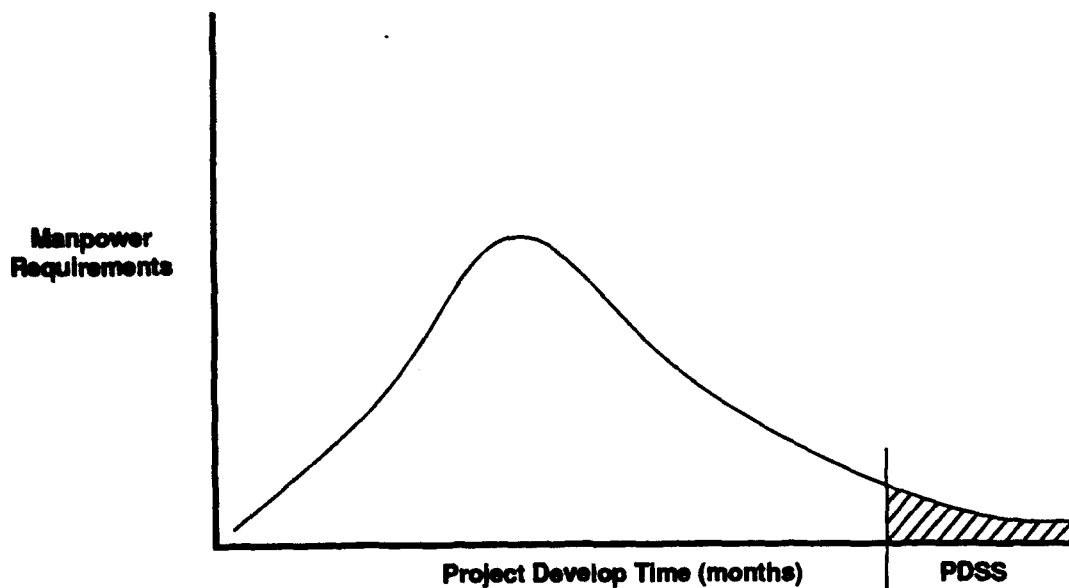


Figure 3. The Rayleigh Distribution

The equations used in COCOMO are derived from empirical data from actual program experience. Regression analysis was applied to this data to develop a parametric equation for estimating software development effort.¹⁶

The basic COCOMO equation is¹⁷ :

$$\text{Man months}_{\text{Development}} = A (\text{LOC})^k B.$$

LOC = Lines of Code

A,B,k = Constant factors reflecting characteristics of the software and the development environment

Factors are based on assessments of software size, type, and attributes. The 15 cost driver attributes used in COCOMO are displayed in Table 1.

¹⁶ Boehm, p. 494.

¹⁷United States Air Force Cost Center, p. 1.

Table 1

COCOMO Cost Driver Attributes

1. Required Software Reliability
 2. Data Base Size
 3. Product Complexity
 4. Execution Time Constraint
 5. Main Storage Constraint
 6. Virtual Machine Volatility
 7. Computer Turn Around Time
 8. Analyst Capability
 9. Applications Experience
 10. Programmer Capability
 11. Virtual Machine Experience
 12. Programming language Experience
 13. Use of Modern Programming Practices
 14. Use of Software Tools
 15. Required Development Schedule
-

COCOMO covers the lion's share of software tasks and functions; however, it does not cover several important ones, including requirements definition and management; installation, acceptance testing, user training or data base administration¹⁸. Unfortunately, these tasks and functions are among most critical during PDSS.

COCOMO estimates annual software "maintenance" (PDSS) to be a proportion of the total effort applied during development based on the volume of changes to the software code. The equation is¹⁹:

$$\text{ManMonths}_{\text{Maintenance}} = (\text{ACT})(\text{MM}_{\text{devel}})(F_{\text{env}})$$

ACT = Annual Change Traffic, Lines of Code changed annually as per Total Program Lines of Code

MM_{devel} = Total Development Effort, Man Months

F_{env} = Environmental Factor reflecting the characteristics of the software and the maintenance environment.

¹⁸ Boehm, p. 52.

¹⁹ United States Air Force Cost Center, para 1-4.

This approach assumes:

- ☐ PDSS is similar to development workload, and PDSS workload can be estimated based on the development workload;²⁰ and
- ☐ PDSS workload is proportional to changes in lines of code.

These assumptions are problematic for the analyst attempting to determine PDSS workload. First, as we have seen, PDSS work is quite different from software development work. Secondly, although PDSS workload is proportional to annual SCR volume. SCR volume does not translate directly to lines of code.

REVIC

The REVIC model is based on the intermediate version of COCOMO and uses COCOMO equations for estimating both development effort and PDSS.²¹ REVIC is frequently used within the DoD to estimate software lifecycle costs.

REVIC and COCOMO. Although REVIC and COCOMO are built on the same conceptual base, they differ significantly in implementation of the model.²² These differences are discussed below:

- ☐ *Calibration Data.* REVIC uses historical data from recent DoD software projects to calibrate the coefficients used in the basic equations. These values are monitored and updated periodically. COCOMO uses static data from commercial software development projects.
- ☐ *Phase Distribution.* REVIC includes an automated routine for distributing effort and schedule over development phases. The model provides a default single weighted average distribution that the user can adjust directly. Within COCOMO, effort and schedule distribution by phase is dictated by static reference tables.
- ☐ *Risk.* REVIC applies statistical techniques to describe lines of code. The ability to calculate standard deviation and risk associated with lines of code, effort, and schedule estimates follows from this capability.
- ☐ *User Interface.* REVIC allows the user to directly define parameters for the software development process. It does not require extensive knowledge of the COCOMO model or its workings.

²⁰ Boehm, p. 536.

²¹ United States Air Force Cost Center, p. 1.

²² United States Air Force Cost Center, para 1-5.

To use REVIC, the operator:

1. Enters environmental factors, resource parameters, and basic program information;
2. Enters a range for lines of code by module; and
3. Enters parameters for expected effort on existing code.

The environmental factor used in the basic COCOMO/REVIC equation is calculated by taking the product of the factor values assigned to the 15 COCOMO cost driver attributes. Each of the attributes is weighted equally in the process.

REVIC PDSS. REVIC uses the COCOMO equation for software maintenance:

$$\text{ManMonths}_{\text{Maintenance}} = (\text{ACT}) (\text{MM}_{\text{devel}}) (\text{F}_{\text{env}})$$

The weaknesses of REVIC for estimating PDSS requirements are the weaknesses of COCOMO. Problems with the equations used by COCOMO for estimating PDSS have been discussed. Additionally, REVIC does not explicitly model the software requirements engineering phase or system level testing, two critical components of PDSS.²³

REVIC is widely used within DoD for estimating both development and PDSS requirements. Its flexibility and ease of use ensures its popularity. However, lacking sufficient detailed data on development effort and PDSS costs, and in the absence of a firm dependency between development and PDSS, the analyst is left to his/her own resources to develop accurate and usable input values.

²³ United States Air Force Cost Center, para 1-3.

POST DEPLOYMENT SOFTWARE SUPPORT REQUIREMENTS ASSESSMENT METHODOLOGY (PRAM)

GENERAL

The PDSS environment differs from the software development environment in several important ways. These differences affect the nature of workload and the approach required to estimate that workload.

PDSS consists of the repetitive execution of a limited set of tasks.²⁴ It is more similar to a manufacturing operation than to a construction project. Software development follows the linear progress of a construction project.

PDSS demands are unconstrained. The potential volume of SCRs generated by user requirements and environmental changes is unlimited. Program experience shows there are always more change requests than can be serviced within resource constraints.

User requests account for the largest share of SCRs. This volume is layered on a base of error correction and environmentally driven changes. The number of system users is often large, and each user is a potential source for a broad range of SCRs based on user needs and preferences. Software development workload, although significant and variable, is limited by design specifications, and the practical limits to personnel loading efficiencies.

PDSS is ultimately constrained by resources/funding. Given unlimited SCR volume and PDSS workload, funding levels influenced by non-PDSS priorities become the effective limit of PDSS effort. The question then, is not how many people are required to meet the total workload requirements, but what level of effort is sufficient to maintain essential levels of effectiveness and readiness.

OVERVIEW

A model is a representation of a process or a system. It provides a conceptual structure for working with and thinking about that process or system. Figure 4 is a model of the manpower and personnel requirements determination process on which the PDSS Requirements Assessment Methodology (PRAM) is based. It can be summarized as follows:

²⁴ Arthur p. viii.

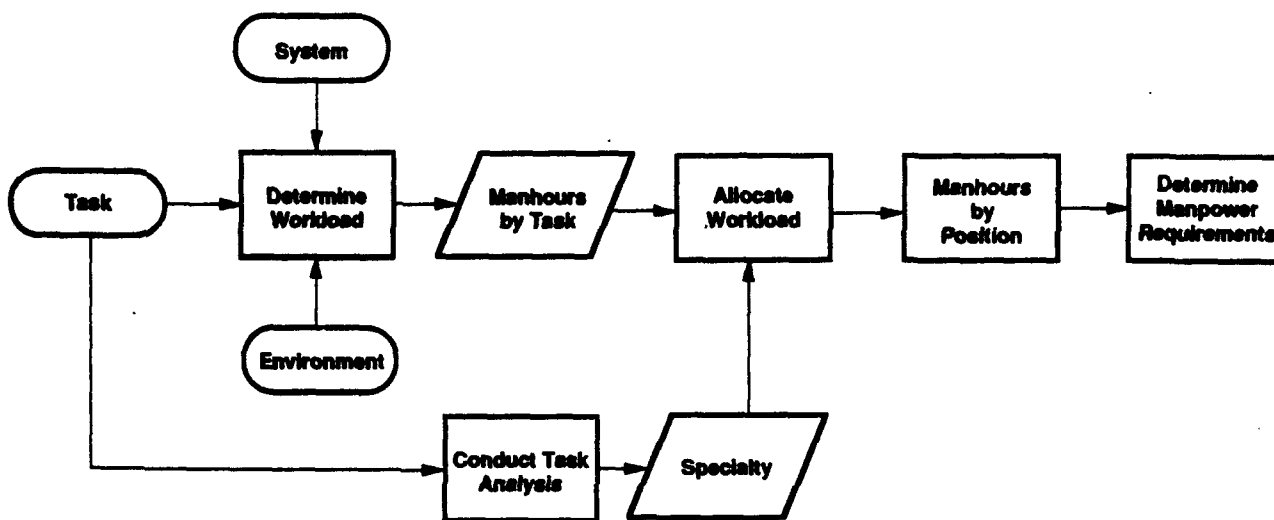


Figure 4. The Manpower Requirements Determination Process

- ☐ Task performance, which is affected by the system and environment, creates workload; in this, case measured in man hours;
- ☐ Workload is allocated or assigned to personnel categories defined by occupational specialty and skill or grade level and
- ☐ The volume of workload and the work capacity of personnel determines manning levels.

Tasks

Tasks define what activities must be performed. Tasks are the basis for organizing work. Task performance is the source of workload. The content and characteristics of tasks are the basis for allocating workload among job categories.

Tasks used in PRAM are listed in Table 2. Tasks can be defined at virtually any level of detail or indenture. In this study, the highest possible level of indenture is used. As data becomes available and PDSS processes become better understood, lower levels of detail within the work breakdown structure can be addressed.

COCOMO does not define PDSS tasks below a single general level of software support. This approach is consistent with the objective of determining project cost and resource requirements. It is not adequate, however, for determine specific PDSS manpower requirements.

Table 2

PDSS Tasks

- ☐ Prepare/Submit Software Change Request
 - ☐ Conduct Impact Analysis
 - ☐ Perform System Release Planning
 - ☐ Execute Change
 - ☐ Conduct System Level Testing
 - ☐ Install/Implement
-

Determine Workload

Workload is the effort required to perform a task. In PRAM, the basic transfer function is the product of task frequency and task level of effort:

$$\text{Workload} = (\text{Task Frequency}) (\text{Mean Level of Effort per Task})$$

COCOMO links workload with lines of code of the software system. This approach may be valid when considering software development. It is not valid for PDSS. PRAM links workload to the execution of SCRs. This better represents the repetitive, task-oriented nature of PDSS workload.

If the analyst's objective is determining workload or total effort for cost or planning purposes, the analysis is complete at this point. Additional steps are required to complete the personnel requirements determination process.

Allocate Workload

Workload allocation is conducted to determine who will do the work. The workload is assigned to job categories. The content and difficulty of each task must be analyzed and matched with categories defined by job or occupational specialty, and level of skill or experience. In the military, these categories are represented by Occupational Specialty (OS) and paygrade or rank, respectively.

Manpower Determination

The productivity, capacity, and availability of personnel in each job category are applied to workload to determine the number of workers needed in each specialty to meet workload requirements at a given location.

COCOMO does not determine manpower requirements by job category. Although personnel determination can be performed outside the COCOMO process, it is not integrated with the COCOMO analysis and is not directly supported by the methodology.

METHODOLOGY

General

The PRAM is based on two principles:

- ☐ Workload is the product of task frequency and level of effort; and
- ☐ PDSS workload is the sum of workload for individual task performance.

The PRAM methodology is summarized in Figure 5 and discussed in detail on the following pages. Application of the methodology to the National Missile Defense (NMD) system is demonstrated in a later chapter of this report.

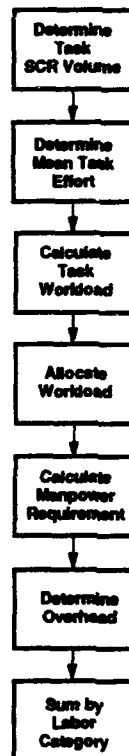


Figure 5. The PDSS Requirements Assessment Methodology

Procedure

- ☐ Determine change request volume
- ☐ Determine mean effort per change request
- ☐ Calculate workload
- ☐ Allocate workload
- ☐ Determine overhead
- ☐ Calculate personnel requirements

The parameters for each analysis step are summarized by task in Table 3.

Determine Change Request Volume. SCR frequency or volume is the primary dependency in estimating PDSS workload. It is a function of several influences, the understanding of which allows us to estimate that volume. Three types of factors which directly effect SCR volume: They are summarized in Table 4.

Error Correction. Software is never error free. Immediately on release of the new system, bugs are discovered that must be corrected. The number and discovery rate of these bugs are a function of software quality achieved during development. The estimation of software error rates is a technical function. Techniques are available to perform this estimation. Software errors, unlike equipment failures, once discovered and corrected, do not re-occur. The number of errors requiring fixes, therefore, decreases over the life of a system as a fixed number of errors are discovered and eliminated. The change request volume for error-driven changes is displayed graphically as a function of time in Figure 6. Software errors account for a small proportion of the PDSS effort.

Table 3

PDSS Methodology Overview

TASK	FREQUENCY	EFFORT	ALLOCATION
Change Request	Change Request Volume	Distribution	User/Field Tech
Impact Analysis	Change Request Volume	Distribution	Systems Analyst Software Engineer
System Release Planning	Down Select %	Distribution	Systems Analyst
Execution (Design Program, & Code)	Down Select %	Distribution	Systems Analyst Software Engineer Programmer Documentation Specialist
Test	Down Select %	Distribution	Systems Analyst
Install/Implement	Release Strategy	Requirements	Systems Analyst Software Engineer Field Tech User/Operator

Table 4

Software Change Request Volume Factors

CATEGORY	SOURCE	DEPENDENCY
Error Correction	Software Quality	Error Rate
Environmental/ Technological	Hardware External/System Links Operating Systems COTSS Doctrine	Rate of Change
User Requirements	Interface Functionality Data	Interface Type System Age Volatility of Operating Environment

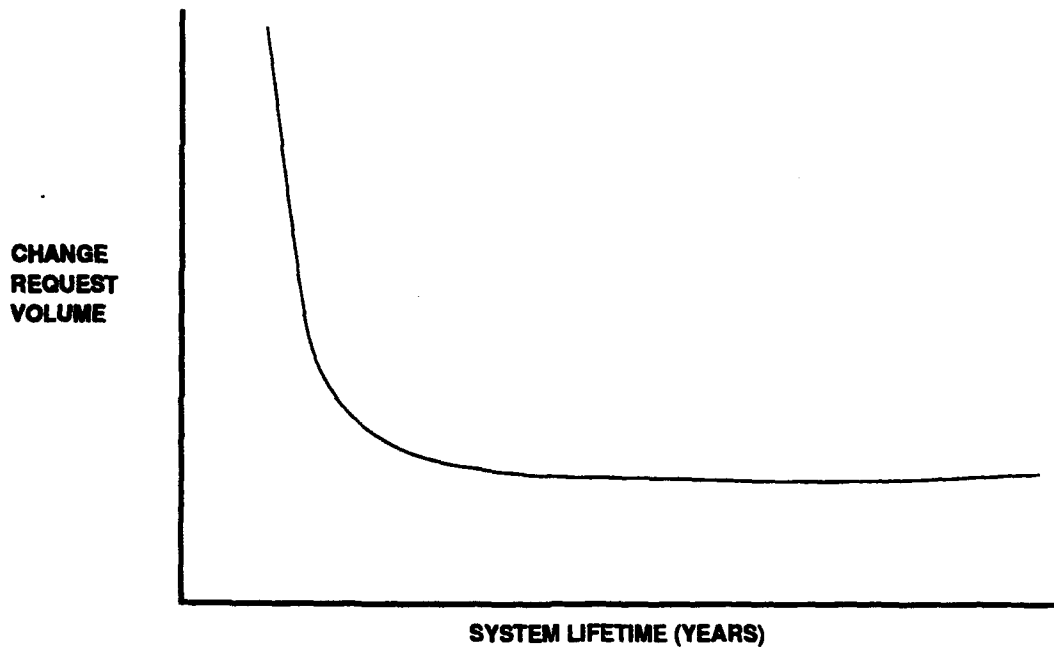


Figure 6. Error Correction Change Request Volume

Environmental/Technological Factors. Throughout the software lifecycle, external factors generate SCRs independently of the operational requirements and performance of the system. The PDSS workload generated by these factors is unavoidable, and in many cases significant. Conceptually, these changes are cyclic and can be predicted. These influences include:

- ❑ **Hardware Changes.** Evolution of technology and re-equipping of the system can be expected to occur once, if not repeatedly during the system's lifetime. Whenever hardware-related changes occur in any component of the system, the software must be reviewed and often modified to assure compatibility with the new components. Hardware changes include rehosting, technology insertion (e.g. new generation micro-computer chips), module replacement (new displays), and the like.
- ❑ **Operating System.** The operating system (DOS, UNIX, VMS, etc.) supporting functional software is updated regularly by the manufacturer, in response to its own SCR release cycle, design evolution, marketing plans, and so on. Changes in the functional software system are often required to ensure compatibility, and to take advantage of new functions and capabilities incorporated in the revised operating system.
- ❑ **Commercial Off-The-Shelf Software (COTSS).** The use of COTSS in DoD and commercial software systems is increasing. COTSS programs undergo their own PDSS cycle, independently of the functional software system. COTSS changes must be documented, evaluated, and installed; and functional software must be modified to ensure compatibility.
- ❑ **External System Links.** Data, processing and operational links among remote systems are common to DoD software systems. Changes in any component of the larger "information system" generates changes or the possibility of change in other members of the system. For example: a change in data record format in a single logistics management system requires a corresponding change by all other systems using those data records as input to their processes.
- ❑ **Doctrine.** Doctrine defines the objective, and provides the basic logic and framework of the software system. Although not directly related to computer hardware or software, changes in doctrine generate changes in programs, files, and data bases.

Doctrine changes referred to here include fundamental paradigm shifts in national strategy or warfighting theory. The current shift from forward deployment for land battle in Western Europe to force projection for operations short of war is one example. Doctrinal changes at lower levels, such as changes in weapon capability and employment, are included in user requirements changes discussed below.

Figure 7 shows a curve representing SCR volume for a single category of Environmental Factors over the life of the system. The system is stable until a change in the environment is introduced. This causes a steep rise in change request volume. Change volume tapers off as the environmental implementation of the change is completed.²⁵

Although the general trends and change rates for each category of environmental factors can be predicted, it is less clear when those changes will occur for each factor relative to another. For that reason, each category of changes due to environmental factors is assumed to be staggered along a curve. That is, the system is always undergoing the effects of a change in one or another environmental factor throughout its lifecycle. The volume of environmental/technological changes is smooth curve which follows the peaks of individual factors.

User Requirements - based changes. Historically, user-generated SCRs constitute the largest proportion of PDSS workload.²⁶ They take several forms:

- ☐ **Interface Changes.** These include user requirements for display changes of all types, report changes, and all other user interface issues. Both functional and visual changes are included.
- ☐ **Functionality.** New or changed functionality is needed to respond to new or changed missions, or operational requirements. New applications and capabilities for the system are discovered by the operator through familiarity with the system and its capabilities.

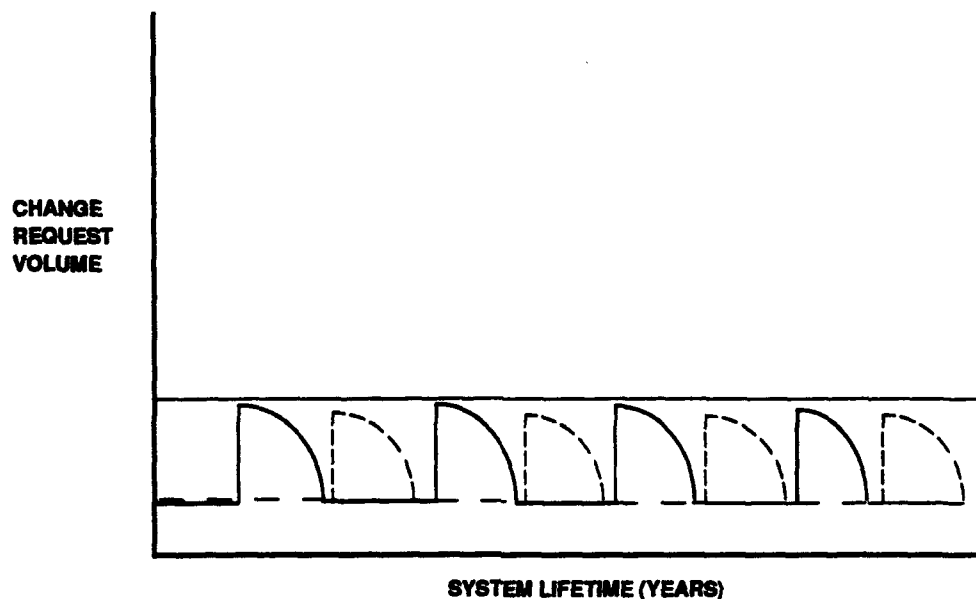


Figure 7. Environmental/Technological SCR Volume

²⁵ Piersall, p. 39.

²⁶ Boehm, p. 549.

- ❑ **Data.** The form, content, and sources of data change frequently. For example, new map data may be required to support a change in unit mission, or new force structure information may be required to accommodate downsizing. User needs and data characteristics are equally important in determining data change request rates.

Requirements-based change rates are dependent on three factors:

- ❑ **Interface Type.** The type of interface, screen, or paper reports affects the volume of mission essential change requests generated by users. Systems with greater user interface will experience change more frequently as more people are involved and experience a longer exposure to the system. Systems with rigid interfaces, such as fixed-format reports, will require more frequent changes to respond to user requirements. Embedded software will have a very low user-generated change volume because the operation of the software is invisible to the user.
- ❑ **System Age.** As the system ages, requirements change with a greater rate relative to system design. More effort is required to make the system design current.
- ❑ **Volatility of the Environment.** The stability or, conversely, the volatility of the operational environment (frequent mission changes, applications changes, data changes) affects the change request frequency.

At this point, recall that user requirements changes are unbounded in terms of content, frequency and complexity. Discussions with Program Office support personnel and evidence from the literature, confirm this. This translates to a constant volume of SCRs during the system lifecycle. In this study we will address only mission critical or essential SCRs.

The curve in Figure 8 shows SCR volume for mission essential user requirements-based changes. At fielding, there are a large number of changes as the user becomes familiar with the capabilities of the system and requirements are modified to coincide with operational realities. As the initial fielding phase and "break-in" of the system is completed, the system reaches a level of stabilization. Then as the system ages, change volume rises at a constant rate. Finally, the rate of change increases significantly during the final stages of the lifecycle when obsolescence occurs.

All SCRs are considered during the SCR preparation and impact analysis phases. At System Release Planning, however, priorities are established and non-critical changes are eliminated. The percentage of SCRs selected for implementation is a function of several parameters; however, it is most reflective of desired resourcing levels.

Release Frequency. SCRs are "bundled" for implementation as "releases." This approach facilitates management and control of the PDSS process. The size and frequency of releases is a function of resources available to support the PDSS effort and the urgency of the change. The release cycle determines installation/implementation requirements.²⁷

²⁷ Arthur, p. 74.

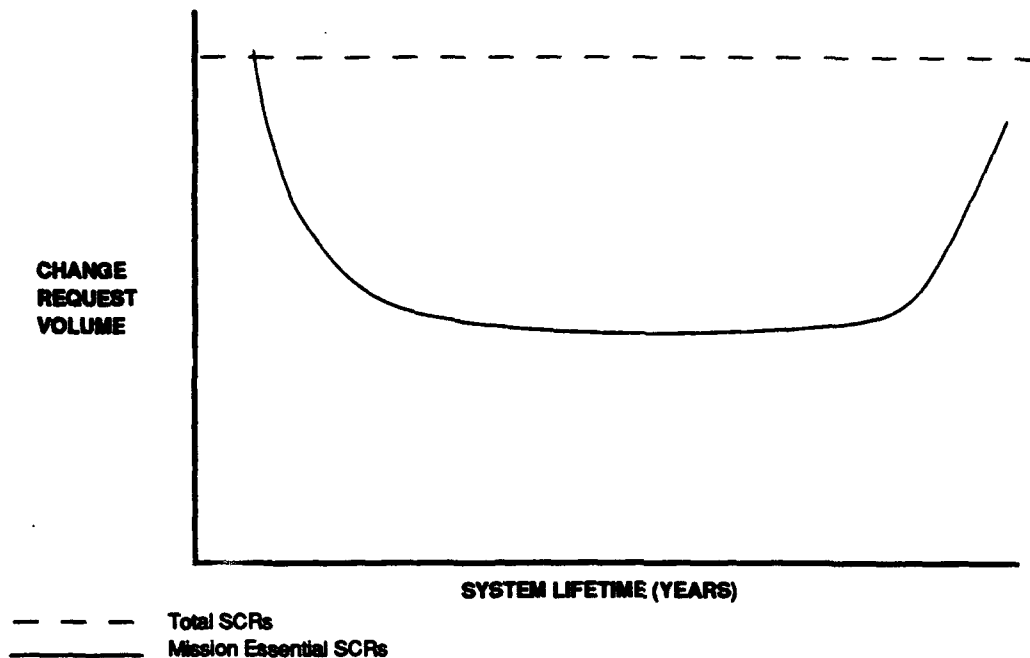


Figure 8. Mission Essential Requirements-Based Change Request Volume

Determine Mean Effort per Change . The second aspect of workload is effort. Many PDSS requirements determination methodologies attempt to predict PDSS workload from the characteristics of SCRs. This is not feasible because:

- ☐ Interaction of these variables is too complex to model reliably with available data and understanding of the software maintenance process; and
- ☐ Volume of changes is "always" greater than available resources. The downselect process adds another variable to the process.

PRAM estimates effort based on average PDSS task performance times. This is consistent with workload measurement systems based on Mean Time to Repair (MTTR). This approach is based on an assumption that task performance times constitute a random distribution about an average or mean value, as shown in Figure 9.²⁸

This approach simplifies the assessment task while providing a realistic value for workload determination. Furthermore, this value can be established by relatively straight-forward sampling and data collection techniques, facilitated by statistical method, avoiding difficult regression analysis and complex data reporting and collection efforts.

²⁸ Dahl, S., Adkins, R., et al.

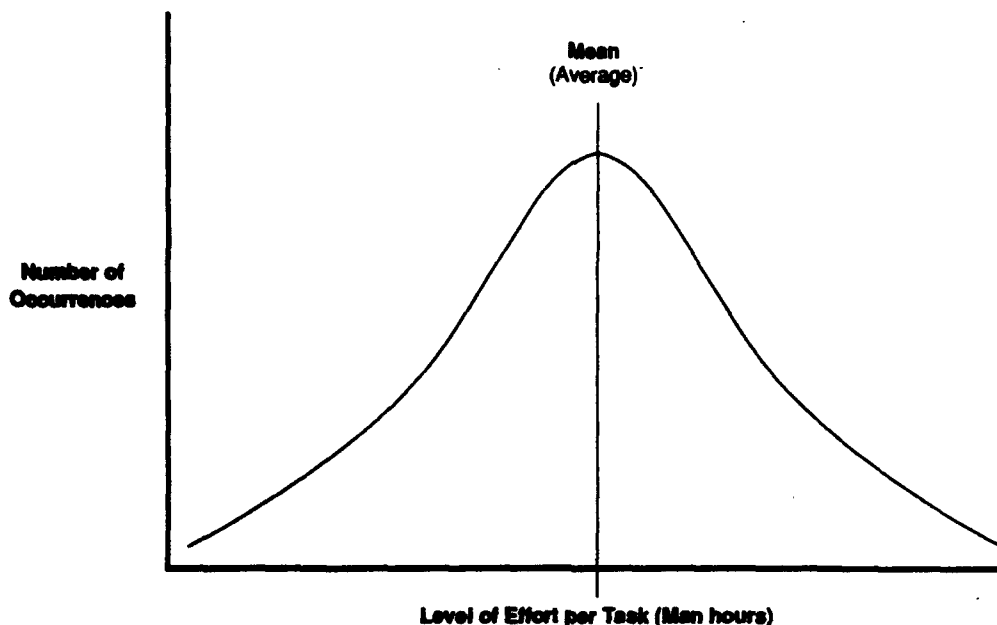


Figure 9. Normal Distribution for Level of Effort Per Task

The curve describing the distribution of effort per change request will vary by task. As data are collected and more is learned about the PDSS process, other dependencies may be uncovered.

Change Request. The mean level of effort required for the user to prepare and submit an SCR will be a narrow distribution, approaching a constant. The effort required to complete this task includes formulating the need and preparing the physical request. Because the user can state his/her requirements in operational terms, the level of effort should be fairly constant for all types of changes; that is, it should form a relatively tight distribution about the mean.²⁹

Impact Analysis. Specific steps must be executed and a given number of characteristics must be assessed in conducting impact analysis for any SCR.³⁰ This establishes a minimum level of effort for almost all SCRs. Due to complexity or potential impact, impact analysis will take longer for some. Therefore, the distribution of the mean level of effort required to conduct impact analysis is skewed toward the higher end of the range of task times as shown in Figure 10.

²⁹ Arthur, p. 19.

³⁰ Department of Defense, Appendix XIV.

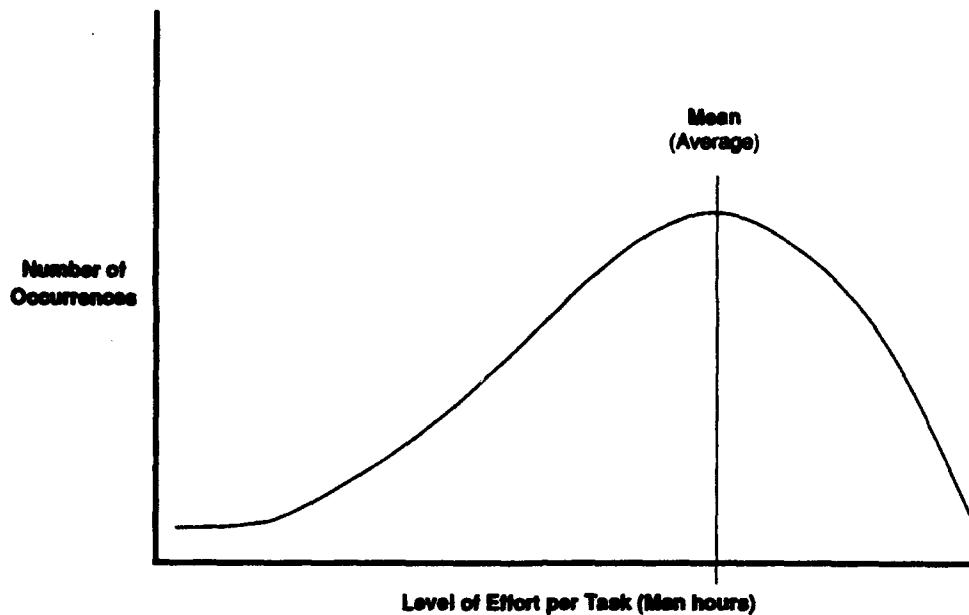


Figure 10. Mean Level of Effort - Impact Analysis

System Release Planning. System release planning is a critical, but low resource task in the PDSS process. All SCRs are evaluated by the members of a Change Control Board (CCB) and by user representatives. A finite level of effort is required for Board Members to review recommendations and participate in the board meetings.

Execution Task. The Execution task includes the design, coding, and unit testing of changes selected for execution during System Release Planning. The mean level of effort associated with this task can be expected to have a wide distribution. Time to complete design and coding tasks can vary significantly based on a number of variables.

Testing. The level of effort for system testing will have a wide distribution. The effort required here depends on the nature and extent of the changes completed in the previous step and on a variety of system and environmental characteristics.

Installation/Implementation. The level of effort required for change implementation is a function of system configuration (installation sites, media), training requirements and system release strategy.

Calculate Workload. After task volume and mean level of effort are determined, the product of these two values is calculated for each task.

Allocate Workload. Workload calculated in the previous analysis step is then allocated or assigned to job categories within an organization. The following representative positions were obtained from the "Occupational Outlook Directory" published by the Department of Labor. Refer to Appendix C for more detailed information on these job positions.

-
- ☐ Systems Engineer
 - ☐ Systems Analyst
 - ☐ Software Engineer
 - ☐ Programmer
 - ☐ Technician

The organization of a PDSS facility is also an important factor to be considered during the allocation process.

Additional research, beyond the scope of this study, is needed to define job categories and to define a procedure for allocating workload. In the near term, a parametric approach can be used, allocating workload based on percentage for each task and category.

Workload allocation is performed for each task. Workload is then summed by labor category. The output is total man-hours by labor category.

The workload allocation process is portrayed in Table 5.

Calculate Manpower Requirements by Occupational Specialty. Workload must now be converted to manpower requirements. The bases for this process include:

- ☐ Capacity. What is the individual work rate?
- ☐ Availability. How many hours per day and year is an individual available for direct productive effort.

The following equation provides a basic approach to determining personnel requirements:

$$\text{Positions} = (\text{mh}) (1/\text{avail hrs})$$

$$\begin{array}{ll} \text{mh} = & \text{Manhours workload required} \\ \text{avail hr} = & \text{available productive man-hours} \end{array}$$

This calculation is performed for each job category. The output is the number of personnel required in each job category. By converting this value to an annual value, using available days per year, a staffing level is calculated.

Determine Overhead. Some overhead or non-workload driven manpower is required to support a complete PDSS operation. Overhead requirements are primarily dedicated to management and administration.

Table 5**The Workload Allocation Process**

Task	Workload*	Allocation*	Total Workload By Position (manhours)
Impact Analysis	120 Manhours	Systems Analyst (50%) - 60 manhours Software Engineer (50%) - 60 manhours	Systems Analyst - 160 Software Engineer - 210 Programmer - 250
Execution	500 Manhours	Systems Analyst (20%) - 100 manhours Software Engineer (30%) - 150 manhours Programmer (50%) - 250 manhours	

* Quantities are for illustrative purposes only

Management. Management overhead will be determined from standard planning factors that provide the necessary number of supervisors and support personnel per worker, consistent with organizational structure and industry standards.

Configuration Management. Configuration Management continues throughout the PDSS process. Records are maintained and updated, and control documentation is prepared and processed. A minimum level of support is required based on the size of the number of Computer Software Configuration Items (CSCI) and anticipated SCR volume.

ARMY NATIONAL MISSILE DEFENSE (ANMD) PDSS ASSESSMENT

OBJECTIVE

PRAM principles are applied to a high level PDSS supportability assessment of the ANMD. The objectives of this study are to:

- ☐ assess the general relevance and applicability of the concepts and principles defined in this report;
- ☐ determine directions for future analysis; and
- ☐ identify weaknesses requiring additional study or modification.

Detailed data is not available on ANMD; relationships and concepts proposed in this report have not been quantified or validated; nor have detailed analysis procedures been developed. Therefore, a detailed analysis is not possible at this point.

SYSTEM DESCRIPTION

The ANMD consists of ground-based defenses for the protection of the entire United States against limited strategic Inter Continental Ballistic Missile (ICBM) and Submarine-Landed Ballistic Missile (SLBM) attacks. NMD is one segment of the Global Protection Against Limited Strikes (GPALS) Program. The design concept for the ANMD has undergone a number of iterations. This study focuses on a single site configuration as shown in Figure 11.

The ANMD Site includes:

- ☐ Ground-based Radar (GBR)
- ☐ Ground-based Entry Point (GEP), satellite communications installation;
- ☐ Ground-Based Interceptor (GBI), missile and missile launch facility;
- ☐ System Operation Area consisting of a Command and control building, Readiness Station, Training Facility, and Security Facility;
- ☐ Communications Support; and
- ☐ Admin/logistics support area.

Leased commercial lines will provide Command Control (C2) links with NMD/NORAD. A fiber-optic network will link nodes within the site. The GEP provides a link with surveillance assets and missiles in flight.

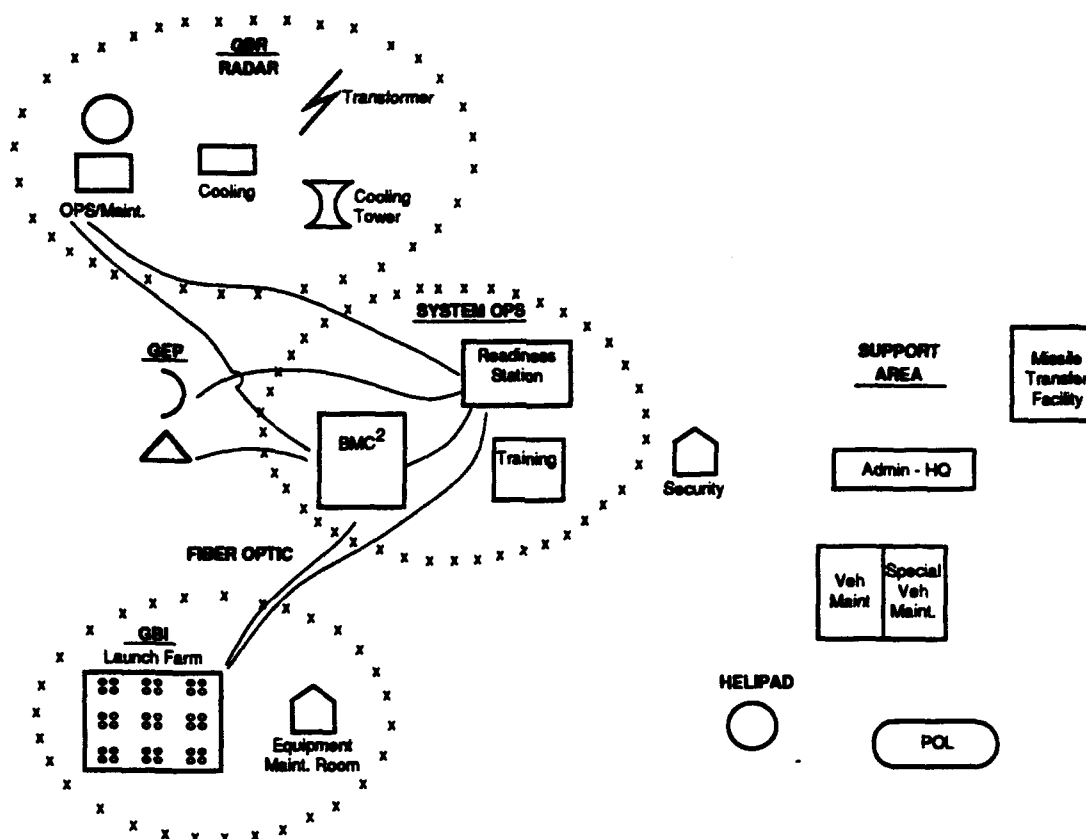


Figure 11. Site Layout

METHODOLOGY APPLICATION

General. The ANMD Site will be software intensive. Critical functions performed or supported by software include:

- ☐ C2 between the site and regional or national missile defense command centers, and among nodes within the site;
- ☐ Management of internal and external communications;
- ☐ Node operation;
- ☐ Fire control and missile guidance;
- ☐ Maintenance;
- ☐ Training; and
- ☐ Administration/Logistics support.

In general, the software will be very critical for operations functions, and less critical for Training and Administration/Logistics functions.

Operation, maintenance and support manning levels will be austere, forcing a heavy reliance on software functionality and availability.

An analysis of this type requires detailed information on the characteristics of the software to be assessed. Application of the methodology requires a definition of the shapes and dimension of the curves which define analysis inputs. Neither of these requirements could be met for this application. Therefore, a subjective estimate of relative orders of magnitude (high, medium, or low) was made for each variable.

Assumptions.

1. The ANMD software system is very large.
2. PDSS level of support will be austere.
3. Analysis will focus on mission critical software components (fire control, Battle Management C2, communications, etc).
4. System-Lifetime is 20 years.

Step 1. Determine Change Request Volume.

Error Correction Rates. Methods are available for predicting software error rates. These are discussed elsewhere and will not be addressed in detail here ³¹.

We will assume that ANMD software follows a standard pattern of decreasing error correction requirements over the lifetime of the system.

Technology/Environment Change Volume. Technology/environment change volume is a function of hardware, operating system, COTSS, linking systems and doctrine. Although the pattern of change for each of these factors is a sawtooth curve, variation in change rate distributes impacts across a continuum. The combined impact is a constant following the mean level of the combined curves. Lacking detailed quantitative data, we will establish general classifications of high, medium or low in this change category.

Environmental/technology SCR volume will be high for ANMD for the following reasons:

- ☐ Mission criticality (CONUS defense) and rapidly changing system and system support technology indicate an expected high, constant rate of change throughout the system lifecycle;
- ☐ Heavy dependence on, and interaction with, external systems (Regional and National Centers, Missile Guidance, etc) means reaction to and accommodation of change to those systems by ANMD;

³¹Boehm, p. 494.

-
- ❑ The ill-defined defense environment and evolving concept of national missile defense indicates a future need for change in basic doctrine as it relates to targeting, opposing weapons, and defense strategies for the US.

User Requirements Changes. User requirements based-changes are the result of change and evolution in user needs, preference, and capabilities. User-generated change volume is high initially, then decreases to a minimum level, before rising again to the end of the lifecycle and obsolescence. The total volume of user-generated change requests is unlimited, and virtually impossible to satisfy. We, therefore, focus on mission essential changes as the more accurate measurement on which to base workload determination.

If we establish high, medium, and low classifications of user requirements-generated changes, ANMD will fall below the medium level for the following reasons:

- ❑ As currently envisioned, the ANMD will be fielded at a single site. This means fewer operators generating user-defined changes in a single operating environment. This environment should serve to limit user-derived change volume.
- ❑ Functionality is relatively fixed. Given an austere O&M environment, and the limited functional requirements assigned to the system, requirements for new functionality will be limited. This aspect of the fielded system will also tend to flatten the center portion of change volume curve.
- ❑ Volatility of the operating environment is fairly stable relative to other systems (i.e tactical, logistics). The size, complexity and cost of changing a system of this size militate against high change rates.
- ❑ Interface design will generate a larger number of change requests, thereby holding up the average rate of change. ANMD interface will be primarily through screen displays. The user is unable to modify these to suit his own needs as he would a locally generated report, therefore. SCR must be initiated and executed for any and all screen changes.

Curves representing change volume for each category are shown in Figure 12.

Total SCR Volume. The three curves described above are combined to determine total change volume. The results are shown in Figure 13. Note that the x axis represents system lifecycle in years. To determine change frequency in a given year, the total for that year is obtained from the graph. This value, more than any other determines annual manning levels.

Down-select Change Volume. Once SCR's are evaluated, a down select is made and only a percentage of total submissions are implemented. For the ANMD, the percentage of SCRs approved for implementation will be on the low side. The relative stability of the operating environment, and the austere support environment will tend to constrain SCR activity.

Change Release Frequency. Installation workload is a direct function of the frequency of releases. Given the austere support environment, assume a single annual release.

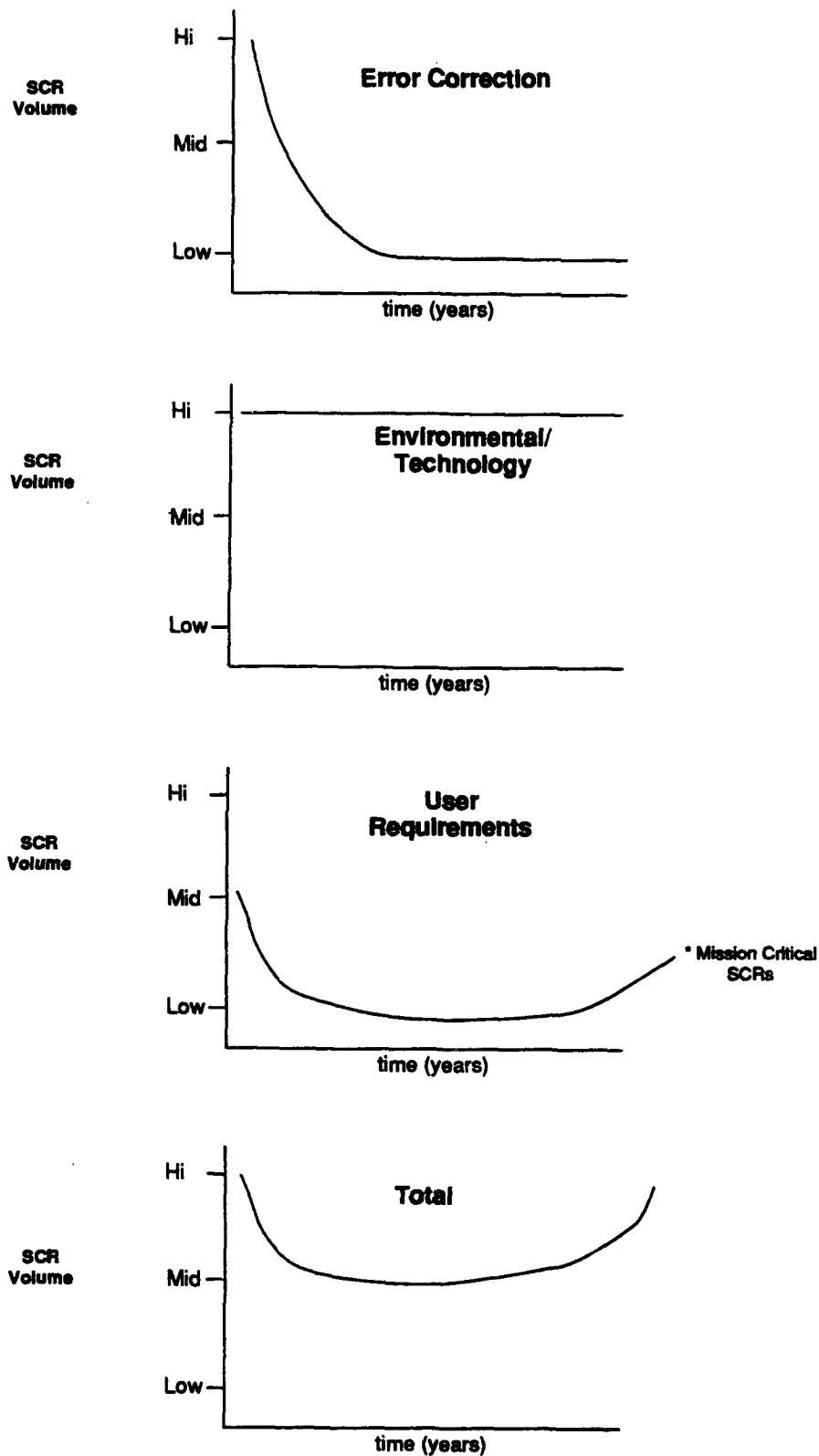


Figure 12. ANMD SCR Volume by Category and Total

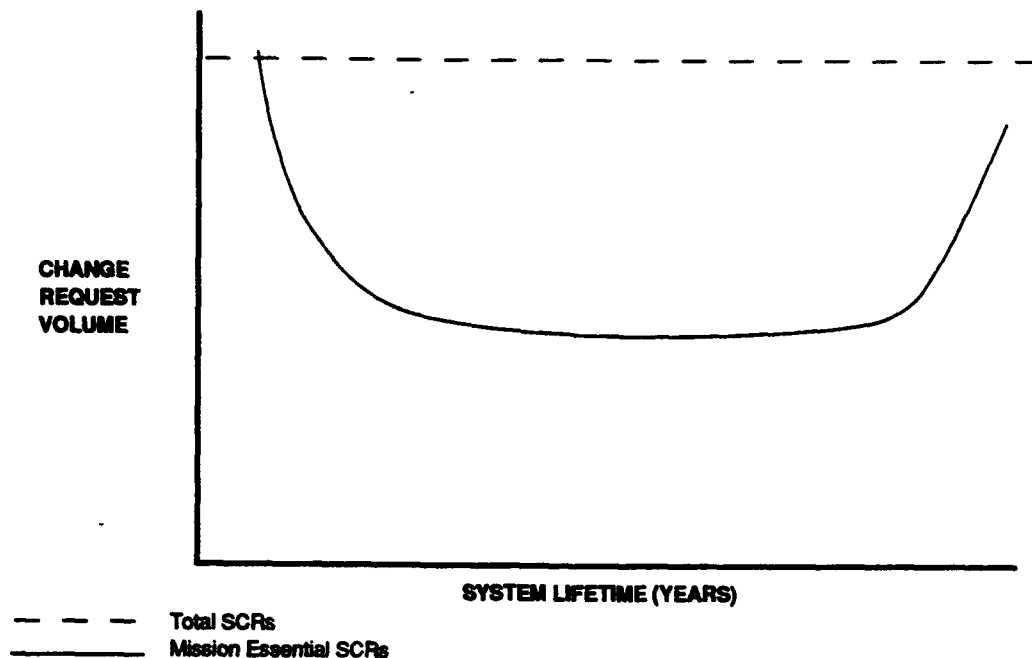


Figure 13. Total SCR Volume

Step 2. Determine Mean Level of Effort

PRAM Workload is the product of SCR-frequency and mean effort per PDSS task. In this step, the analyst determines mean effort per task. Research has shown that task performance workload is adequately represented by a distribution of values about a mean. Sample distributions for the tasks included in PRAM are shown in Figure 14.

Change Request Management. This task includes the preparation, processing and submission of an SCR. The mean effort required to submit a change request is a normal distribution tightly distributed about the mean. This means there is little variation in the effort required to perform this task. SCRs are defined in functional terms by the user, and should require little research or analysis. Forms are simple, although there may be some variation in preparation time for electronic versus paper submissions.

Impact Analysis. This task includes the assessment and review of each SCR. Impact analysis effort is a distribution skewed towards the upper-end of the scale. This reflects a scenario in which a minimum effort is required for assessment of any SCR. There are specific areas which must be checked regardless of the complexity or magnitude of the proposed change, and the analyst cannot easily determine impacts until an assessment of these basic areas is made. On the other hand, complex SCRs will take more effort to assess.

Due to the complexity, the number of internal and external modules, and criticality of ANMD software, it is expected that the mean level of effort for impact analysis will be above average and the distribution skewed heavily towards higher values.

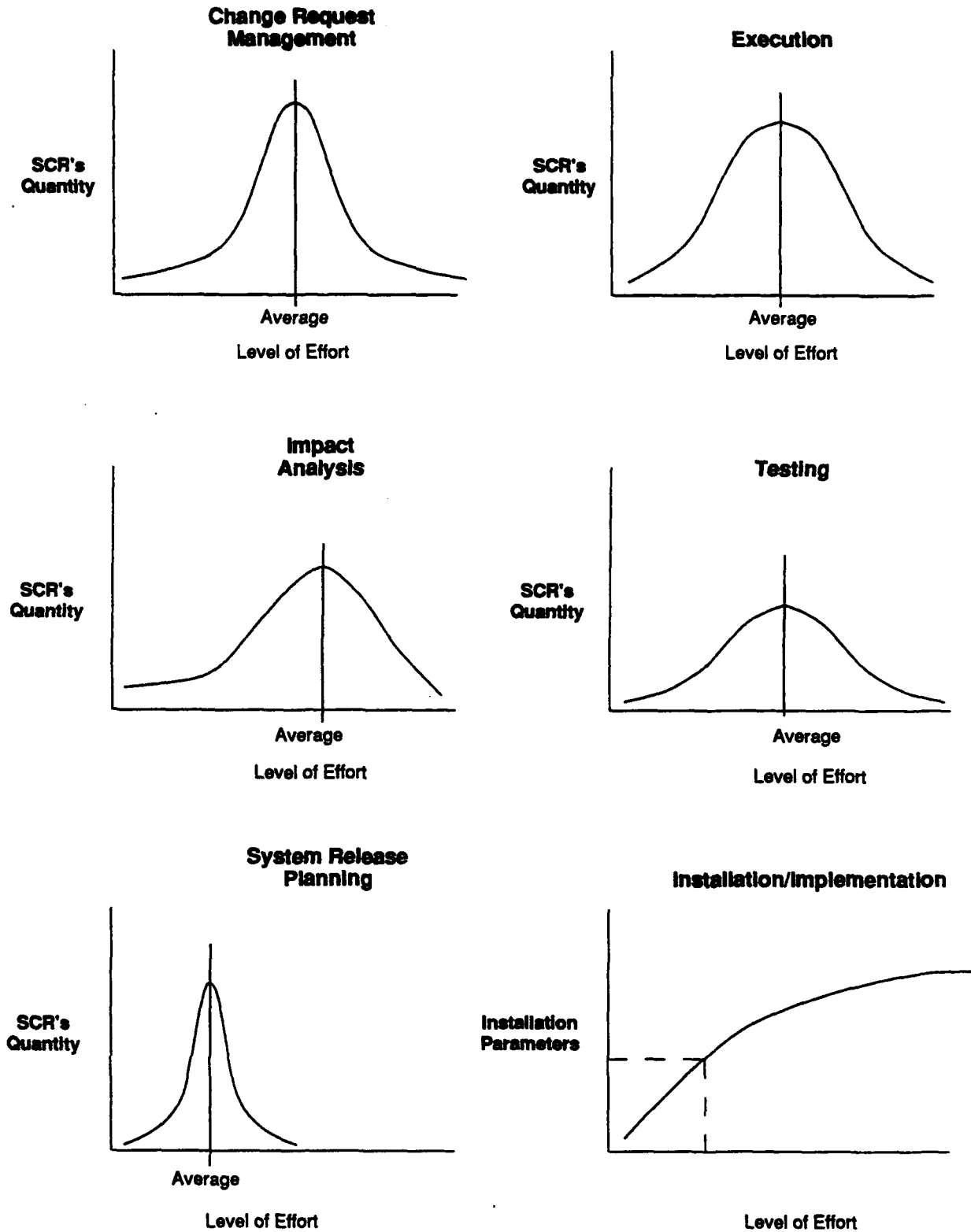


Figure 14. Mean Level of Effort Distribution by Task

System Release Planning. System release planning is a critical but low resource PDSS task. It includes Change Control Board review, SCR selection and ranking, and release planning. Actual workload includes document review and participation in meetings. This is a routine process and the mean level of effort should form a narrow distribution.

Execution. SCR execution includes the planning, design, programming, coding and unit level testing of software changes. Given the number of variables and random nature of SCR content, it is infeasible to attempt to predict mean level of effort from individual software engineering tasks just as it would be infeasible to predict automotive repair workload by predicting the specific type of failures and repair actions required. Research has shown that workload, time to repair, etc follows a normal distribution. This is the approach taken by PRAM.

Testing. The level of effort required to conduct system and integration testing is dependent on a variety of factors including the design of the change, modules and lines of code affected, the characteristics of the system itself. These factors are inherently unpredictable and therefore the mean level of effort required for testing is also a distribution around a mean or average value.

Installation/Implementation. Installation workload is a function of the number of sites, the level of expertise of site operators, and training required. This should be quite low for ANMD. Installation is required at the single site only and the level of expertise of operator's at that site will reduce training and installation/testing requirements.

Step 3. Calculate Workload

Sample calculations of ANMD workload are summarized in Table 6. The values used are not actual; but were selected only for demonstration purposes.

Step 4. Allocate Workload

Table 7 demonstrates the allocation of workload among occupational specialties. These positions were loosely based on the Department of Labor's Occupational Outlook Handbook and aligned by task based on SME experience. Lacking more detailed data, workload was distributed proportionally between these positions. Percentages are not actual, but were created for demonstration purposes only.

Step 5. Calculate Manpower Requirements

Manpower Requirements are a function of workload, worker capacity and worker availability. The standard calculations for determining man-power requirements for variable positions are found in Army Regulation 570-2³².

³² Department of the Army, p. 10.

Table 6

Demonstration Workload Calculation - ANMD

Task	Annual SCR Volume	Mean Level Of Effort per SCR-Task (Manhours)	Workload (Manhours) per year
Change Request	1000	2	2,000
Impact Analysis	1000	80	80,000
System Release Planning	800	12	9,600
Execution	800	240	192,000
Testing	800	120	9,600
Installation	1	120	120

NOTE: These are not actual values, but were selected for demonstration purposes only!

Table 7

Demonstration Workload Allocation - ANMD

Task	Position	Allocation (%)	Allocated Workload (manhours)
Change Request	User/Field Rep.	100	2,000
Impact Analysis	Systems Analyst	40	32,000
	Software Engineer	60	48,000
System Release Planning	Systems Analyst	100	9,600
Execution	Systems Analyst	10	19,200
	Software Engineer	20	38,400
	Programmer	70	134,400
Testing	Systems Analyst	100	9,6000
Installation	Systems Analyst	20	24
	Software Engineer	30	36
		50	60

NOTE: These are not actual values, but were selected for demonstration purposes only!

$$\text{MPR} = \text{WL} / \text{APM}$$

MPR = Manpower Requirement

WL = Productive man-hours required

APM = Available Productive Manhours

Available productive manhours must account for the work capacity of the individual as well as the structure of the organization. The calculation of manpower requirements by position are shown in Table 8.

Step 6. Determine Overhead

Supervisory and administrative support requirements are calculated from standard tables of authorization such as those found in AR 570-2. This step was not included in this demonstration.

Step 7. Analysis Results

Requirements are summed by personnel position to determine total personnel Requirements. Results are summarized in Table 8. Values used are for demonstration purposes only and were analytically or empirically derived.

Table 8

Demonstration Calculation, Personnel Requirements by Position

	Annual Productive Man-hours Required	Annual Productive Man-hours Available per Individual	Positions Required
User/Field Rep.	2,000	1824	1
System Engineer	156,824	1824	86
Software Engineer	86,436	1824	47
Programmer	134,400	1824	74
Field Rep	60	1824	1
Total			209

NOTE: These are not actual values, but were selected for demonstration purposes only!

CONCLUSIONS AND RECOMMENDATIONS

CONCLUSIONS

The concept proposed for PRAM is valid. At the analytical level at which this study was conducted, no significant inconsistencies were discovered. During the course of the study, subject matter experts in Air Force and Army Program and PEO organizations confirmed the relevance and apparent validity of PRAM assumptions, concepts and overall approach. Further study is required to provide empirical validation of PRAM and to provide a detailed foundation for a practical methodology.

RECOMMENDATIONS

Recommend a phased approach to PRAM development which will include data collection, extension of the methodology and validation of the conceptual approach.

Phase 1. Conceptual Prototype. The objective of Phase 1 is development of a baseline prototype which will support limited applications for additional research. Development of the Conceptual Prototype will include:

- ☐ Data collection and analysis to determine shape of frequency and effort curves and distributions, and to validate other PRAM assumptions;
- ☐ Research of occupational category descriptions and workload allocation methodologies;
- ☐ Investigation and identification of variables and dependencies for workload determination. These may include system functionality, program characteristics and organizational structure.
- ☐ Application of the Conceptual Prototype to a DoD system.

Phase 2. Verification Prototype. Phase 2 will produce an integrated PRAM tool. The objective of this phase is verification of the PRAM methodology and a PRAM analysis tool. Phase 2 Tasks include:

- ☐ Continued data collection and analysis;
- ☐ Development and documentation of a full methodology;
- ☐ Testing of variables and dependencies;
- ☐ Development of an automated PRAM analysis tool; and
- ☐ Application of the PRAM Verification Prototype and analysis tool to a DoD system in the laboratory environment.

Phase 3. Operational Prototype. The Operational Prototype will be an operational version of the PRAM analysis tool. Validation and final verification of the PRAM Concept will be conducted during this phase. Phase 3 task include:

- ☐ Analysis and documentation of the PRAM methodology and concepts;
- ☐ Development and documentation of an automated prototype PRAM analysis tool;
- ☐ Operational test of the PRAM concept and tool by user representatives in the analysis environment.

APPENDIX A—ACRONYM LIST

ACT	Annual Change Traffic
AEPCO	Advanced Engineering and Planning Corp.
AI	Artificial Intelligence
ASIOE	Associated Support Items of Equipment
ANMD	Army National Missile Defense System
BM/C2	Battle Management/Command and control ACT Annual Change Traffic
CAD	Computer Aided Design
CASE	Computer-Aided Software Engineering
CCB	Change Control Board
CHS	Common Hardware and Software
CLS	Contractor Logistic Support
COCOMO	Constructive Cost Model
CONUS	Continental United States
COTSS	Commercial Off-the-Shelf Software
C2	Command and Control
CSCI	Computer Software Component Integration
DA	Department of the Army
DAB	Defense Acquisition Board
DoD	Department of Defense
DRC	Dynamics Research Corporation
DSMC	Defense Systems Management College
DTIC	Defense Technical Information Center
EDP	Electronic Data Processing
GBI	Ground-Based Interceptor
GBR	Ground-Based Radar
GEP	Ground-Based Entry Point
GFE	Government Furnished Equipment
GFI	Government Furnished Information
GPAL	Global Protection Against Limited Strikes Program
ICBM	Intercontinental Ballistic Missile
ILSP	Integrated Logistic Support Plan
JCL	Job Control Language
LAN	Local Area Network
LCSE	Life Cycle Software Engineering
LOC	Lines of Code

MANPRINT	Manpower and Personnel Integration
MOS	Military Occupational Specialty
MM	Man Months
MPT	Manpower, Personnel, and Training
M/R	Maintenance Ratio
MRD	Manpower Requirements Determination
MTTR	Mean Time to Repair
NMD	National Missile Defense
OJT	On-the-Job Training
OMS	Operator, Maintainer, Support
O&M	Operations and Maintenance
ORD	Operational Requirements Document
PDSS	Post Deployment Software Support
PEO	Program Executive Office
PM	Program Manager
PMO	Program Management Office
POI	Program of Instruction
POS	Personnel Occupational Specialty
PRAM	PDSS Requirements Assessment Methodology
RAM	Reliability, Availability, and Maintainability
REVIC	Revised Intermediate COCOMO
RFP	Request for Proposal
SCR	Software Change Request
SKAs	Skills, Knowledge and Abilities
SLBM	Submarine-Launched Ballistic Missile
SLIM	Software Life Cycle Management
SME	Subject Matter Expert
SOW	Statement of Work
SRB	Software Review Board
TA	Task Analysis
TAD	Target Audience Description
TMDE	Test, Measurement and Diagnostic Equipment
TRAC	TRADOC Analysis Center
TRADOC	U. S. Army Training and Doctrine Command
USAADASCH	United States Army Air Defense Artillery School

APPENDIX B—REFERENCES

- Arthur, L.J. (1988). Software Evolution: The Software Maintenance Challenge. New York: John Wiley and Sons.
- Boehm, B.W. (1981). Software Engineering Economics. Englewood Cliffs, NJ: Prentice-Hall, Inc.
- Cane, S. (1993). Software Maintenance Manning (a briefing for Computer Resources Working Group, Project Peace Shield). Wilmington, MA: Dynamics Research Corporation.
- Caro, I., Higuera, R., et al. (circa 1989). Mission Critical Computer Resources Management Guide. Ft. Belvoir, VA: Defense Systems Management College.
- Cline, R. (1989). E-3A (AWACS) Post Deployment Software Support. Seattle, WA: The Boeing Company.
- Curtis, B., Sheppard, S.B., et al. (March 1979). Measuring the Complexity of Software Maintenance Tasks with the Halstead and McCabe Metrics. IEEE Transactions on Software Engineering, Vol. SE-5, No. 2.
- Dahl, S., Adkins, R., et al. (1990). Final Report for Concepts of MPT Estimation (Development of MANPRINT Methods). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.
- Department of the Army. (1992). Manpower Requirements Criteria (MARC)—Tables of Organization and Equipment (AR 570-2). Washington, D.C.: Author.
- Department of Defense. (1985). Configuration Management Practices for Systems, Equipment, Munitions and Computer Programs (MIL-STD-483A). Washington, D.C.: Author.
- Department of Labor (1992). Occupational Outlook Handbook. Washington, D.D.: Bureau of Labor Statistics.
- Fugate, C.S. (Undated). Assessing Long-Term Software Costs. General Electric.
- NMD Integrated Logistics Support Plan. (March 1993). Global Protection Against Limited Strikes (GPALS). Huntsville, AL: Program Executive Office (PEO), NMD Program Management Office.
- Piersall, J. (January-February 1994). The Importance of Software Support to Army Readiness. Army Research and Acquisition Bulletin.
- Quantitative Software Management, Inc. (1987). SLIM User Manual. McLean, VA: Author.
- USAADASCH, Directorate of Combat Developments (DCD). (February 1993). Operational Requirements Document (ORD) for NMD System. Fort Bliss, TX: Author.
- United States Air Force Cost Center. (February 1991). REVIC Software Cost Estimating Model User's Manual, Version 9.1. Arlington, VA: Author.
- United States Department of Commerce/National Bureau of Standards. (June 1984). Guideline on Software Maintenance (FIPS Pub. 106). Gaithersburg, MD: Author.

Vallabhaneni, S.R. (1987). Auditing the Maintenance of Software. Englewood Cliff, NJ:
Prentice-Hall, Inc.

APPENDIX C - PERSONNEL AND TRAINING ANALYSIS

OVERVIEW

The objective of the PDSS Personnel and Training Analysis was to identify, at a high level, personnel occupational specialties for the PDSS Requirements Assessment Methodology. In addition, training prerequisites were determined for these personnel specialties.

PERSONNEL AND TRAINING ANALYSIS ASSUMPTIONS AND CONSTRAINTS

The following assumptions and constraints were applied to the manpower, personnel and training analysis:

- ☐ PDSS manpower requirements will be supported as a "level-of-effort" by contractor personnel.
- ☐ Training prerequisites include both formal education, categories of courses and types of degrees, as well as informal on-the-job training (OJT) programs.

PERSONNEL OCCUPATIONAL SPECIALTIES, QUALIFICATIONS, AND TRAINING

Systems Engineer

Description. Systems Engineers apply scientific and engineering efforts to:

1. Transform an operational need into a description of system performance parameters and a system configuration;
2. Integrate related technical parameters and ensure compatibility of all physical functional and program interfaces to optimize the total system definition and design; and
3. Integrate reliability, maintainability, safety, survivability, requirements etc., to meet cost, schedule, and technical performance.

Duties

Analyze user requirements to establish functional requirements
Develop a system's design
Develop and prepare detailed machine logic flow charting and input/output specifications
Formulate logical statements of problems and devising procedures for solutions
Analyze a system's logic difficulties
Revise a system's logic and procedures
Verify program logic
Provide technical support to users

SKAs

Skills

Operate computer consoles
Operate computer peripheral equipment

Knowledge

Models of computation, performance measures, and system architecture types
Algorithms and their performance and relationship to system architecture
Software concepts, data structures, file systems, operating systems, development methodologies, and protocols
Hardware concepts, computer architectures, communication systems, peripheral control systems, and bus architectures
Mathematical theory and concepts

Abilities

Understand abstract concepts and ideas
Communicate highly technical concepts, ideas, and instructions to users, analysts, and programmers

Training Prerequisites. Post graduate level education and experience in both hardware and software design.

Types of Degrees. BS degree in electronic engineering. Senior engineering positions - usually require a post graduate degree.

Systems Analyst

Description. Systems analysts apply analysis techniques and procedures to

1. Manage the development, enhancement, and implementation of new or revised computer systems;
2. Establish test parameters; and
3. Define the integration of developed software and commercial-off-the-shelf-software (COTSS).

Duties

Develop design specifications
Analyze procedures to be automated
Specify number and types of records and files
Prepare work flow diagrams, and data flow charts
Prepare test procedures
Monitor and conduct system level tests
Provide technical support to programmers

SKAs

Skills

Operate computer consoles
Operate computer peripheral equipment
Well versed in a variety of areas

Knowledge

Technical characteristics of computer hardware and peripheral equipment
Testing methods
Objectives and design of applications software
COTSS and operating systems software
Software concepts, architecture, data structures, file systems, operating systems, development methodologies, and protocols
Hardware concepts, computer architectures, communication systems, peripheral control systems, and bus architectures
Mathematical theory and concepts

Abilities

Understand abstract concepts and ideas
Communicate highly technical concepts, ideas, and instructions to users, engineers, and programmers
Interpret system design specifications; and
Interpret software flow diagrams

Training Prerequisites. Performance at this level requires graduate level education and experience in software and hardware design.

Types of Degrees. BS degree in electronic engineering, mathematics, or computer science. Senior systems analyst positions usually require a post graduate degree, and strong problem solving skills.

Software Engineer

Description. Software engineers apply software engineering efforts to

1. Perform product design
2. Develop engineering standards and procedures involving software technology and its applications and
3. Apply systems analysis techniques and procedures where system' requirements are predetermined.

Duties

Conduct fact finding and analysis
Establish procedures where the nature of the system, feasibility, computer equipment, and programming language have already been decided
Assist in the preparation of detailed specifications required by computer programmers
Analyze user problems
Recommend modifications to the existing system

SKAs

Skills

Operate computer consoles
Operate computer peripheral equipment

Knowledge

Models of computation, performance measures, and software architecture types
Algorithms and their performance and relationship to software architecture
Software concepts, data structures, file systems, operating systems, development methodologies, and protocols
Mathematical concepts

Abilities

Understand technical concepts and ideas
Communicate technical concepts, ideas, and instructions to analysts and programmers
Interpret system specifications
Interpret software flow diagrams

Training Prerequisites. Performance at this level requires graduate level education and experience in software design.

Types of Degrees. BS degree in mathematics or computer science. Senior software engineering positions usually require a post-graduate degree.

Computer Programmer

Description. Computer programmers

1. Use detailed designs to develop precise steps and processing logic to convert a process into machine-usable code and
2. Test and validate individual coded steps and processes.

Duties

Interpret work flow diagrams and data flow charts
Encode or modify computer software programs
Analyze errors in encoded programs (debugging)
Test software modules and programs
Prepare software documentation
Install software programs

SKAs

Skills

Operate computer consoles
Operate computer peripheral equipment

Knowledge

Data process functions
Testing methods
Objectives and design of applications software
COTSS and operating systems software
Software data structures, file systems, operating systems, and protocols

Abilities

- Understand technical concepts and ideas
- Communicate technical concepts, ideas, and instructions to users, engineers, and analysts
- Interpret software flow diagrams

Training Prerequisites. There are no universal training requirements for programmers; however, a college degree is not needed in order to work in this field. Because of rapidly changing technology, programmers must continue their education throughout their career. Computer programming is taught at public and private vocational schools, community and junior colleges, and universities. Many programmers have taken courses in areas such as accounting, inventory control, or other business areas to supplement their programming expertise.

Types of Degrees. BS degrees in computer science, mathematics, or computer information systems. In the absence of a degree, 2-5 years' experience may be needed.

Computer Operator

Description. Computer operators are responsible for

1. Coordinating and executing activities related to the installation of software
2. Operating computer hardware and peripheral equipment, and
3. Troubleshooting software problems.

Duties

- Perform system "cold" and "hot" start procedures
- Perform system back-up procedures
- Install new software programs
- Identify system malfunctions
- Initiate corrective actions
- Purge records and databases
- Initialize, operate, and control computer hardware and peripheral equipment
- Monitor job flows

SKAs

Skills

- Operate computer consoles
- Operate computer peripheral equipment

Knowledge

- Computer operations and procedures
- Peripheral equipment operations and functions
- Operating system's job control language (JCL)
- COTSS and application software
- Computer hardware and peripheral equipment configuration
- Testing methods

Abilities

- Understand technical concepts and ideas
- Communicate technical concepts, ideas, and instructions to engineers, analysts, programmers, and repairers/maintainers
- Interpret technical manuals

Training Prerequisites. There are no universal training requirements for computer operators; however, a college degree is not necessary to work in this field. Because of rapidly changing technology, programmers must continue their education throughout their career. 1-3 years' experience is minimum.

Categories of Courses. Some secondary or business school training is usually required for entry-level positions. Computer operating systems are taught at public and private vocational schools, community and junior colleges, and universities. Training is also offered in the military services and by some computer manufacturers.

Types of Degrees. BS degree in computer information systems may be required for senior computer operators' positions or supervisory personnel.

Computer Repairer/Maintainer

Description. Computer repairers/maintainers are responsible for coordinating and executing activities related to the installation, repair and preventive maintenance of computer hardware and peripheral equipment.

Duties

- Install computer hardware and peripheral equipment
- Assist with system and component-level tests
- Diagnose, troubleshoot and isolate malfunctions
- Perform built-in-test, built-in-test-equipment, and computer aided diagnostics procedures
- Remove, replace, and repair failed parts
- Maintain the organization's spare parts inventory

SKAs

Skills

- Operate computer consoles
- Operate computer peripheral equipment
- Operate Test, Measurement and Diagnostic (TMDE)
- Operate power tools
- Soldering

Knowledge

- Computer operations and procedures
- Peripheral equipment operations and functions
- Electronic theory

Physics
Mathematics
Computer hardware and peripheral equipment configuration
Testing methods

Abilities

Understand technical concepts and ideas
Communicate technical concepts, ideas, and instructions to users, engineers, analysts, and programmers
Interpret technical manuals
Interpret equipment schematics

Training Prerequisites. There are no universal training requirements for computer repairers/maintainers; however, a college degree is not needed to work in this field. Some secondary school training is usually required for entry-level positions. Because of rapidly changing technology, repairers/maintainers must continue their education throughout their career.

Categories of Courses. Most positions require 1-2 years' post-high school training in basic electronics, data processing equipment maintenance, or electrical engineering. Electronic theory and repair procedures are taught at public and private vocational schools, community and junior colleges, and universities. Training is also offered in the military services and by some computer manufacturers.

Types of Degrees. Vocational/Technical training in electronics repair would be required. Senior computer repairer/maintainer positions or supervisory personnel would require 5-10 years work experience.

Field Technician

Description. The field technician is normally provided by a contractor and resides on-site or is on-call in a centralized maintenance facility. The field technician

1. Resolves both hardware- and software-related problems
2. Executes activities related to the installation of computer hardware, peripheral equipment, and software; and
3. Processes field software change requests.

Duties. These include both computer repair- and operation-related duties:

Computer repair-related duties

Performing system "cold" and "hot" start procedures
Perform system back-up procedures
Install new software programs
Identify system malfunctions
Initiate corrective actions
Purge records and databases
Initialize, operate, and control computer hardware and peripheral equipment
Monitor job flows
Assist the organizations computer operator

Computer operation related duties

- Install computer hardware and peripheral equipment
- Assist with system and component-level tests
- Diagnose, troubleshoot, and isolate malfunctions
- Perform built-in-test, built-in-test-equipment, and computer-aided diagnostics procedures
- Remove, replace, and repair failed parts
- Maintain the organization's spare parts inventory
- Assist the organization's computer repairer/maintainer

SKAs. These include both computer repair and computer operation.

Skills

- Operate and repair computer consoles
- Operate and repair computer peripheral equipment
- Operate TMDE
- Operate power tools
- Soldering

Knowledge

- Computer operations and procedures
- Peripheral equipment operations and functions
- Operating system's JCL
- COTSS and application software
- Computer hardware and peripheral equipment configuration
- Electronic theory
- Physics
- Mathematics
- Testing methods

Abilities

- Understand technical concepts and ideas
- Communicate technical concepts, ideas, and instructions to engineers, analysts, programmers, operators, and repairers/maintainers
- Interpret technical manuals
- Interpret equipment schematics

Training Prerequisites. BS degree with studies in electronics, computer repair procedures, hardware configuration, and computer operation is normally required for this position. Other training (such as military courses or vendor-provided specific equipment training) coupled with experience may qualify a person to work as a field technician.

Types of Degrees. BS in electrical engineering or computer science is required.

PERSONNEL SPECIALTIES VERSUS PDSS TASKS

The seven personnel occupational specialties are arrayed against the various PDSS tasks depict the level of involvement of each personnel specialty, as shown in Table 9.

Table 9

Occupational Specialties and PDSS Tasks

JOB SPECIALTY	PDSS TASKS/SUBTASKS									
	Requirements Analysis	Design	Coding	Testing	Integration	Software Configuration Management (CM)	Software Quality Assurance	Update Documentation	Computer Program Configuration Sub-Board Participation	Training
Systems Engineer	X	X		X	X	X			X	
Systems Analyst	X	X		X	X	X			X	
Software Engineer	X	X	X	X	X	X	X	X	X	
Computer Programmer			X	X			X	X		
Computer Operator				X						X
Computer Repairman/Maintainer				X		X				X
Field Technician				X				X		X