DEVELOPMENT OF A COMPUTERIZED INFORMATION SYSTEM WHICH SUPPORTS MEASUREMENT AND ASSESSMENT OF AEGIS COMBAT SYSTEMS CENTER PERFORMANCE

by:

Michael Purello

Project and Report submitted to the Faculty of

the Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

IN

SYSTEMS ENGINEERING

APPROVED:

L. Kenneth Harmon, Chair

Konstantinos P. Triantis

Antonio Couto

April 24, 1997

Blacksburg, Virginia

Keywords: Performance, Assessment, Evaluation, Measurement

DEVELOPMENT OF A COMPUTERIZED INFORMATION SYSTEM WHICH SUPPORTS MEASUREMENT AND ASSESSMENT OF AEGIS COMBAT SYSTEMS CENTER PERFORMANCE

by:

Michael Purello

Committee Chairman: L. Kenneth Harmon
Systems Engineering

(ABSTRACT)

A series of effectiveness measures which can be used by the AEGIS Combat Systems Center (ACSC) to measure combat system support effectiveness as it relates to satisfying customers will be presented. Utilizing the effectiveness measures identified in the first part of the report, a computerized performance assessment system (in more generic terms, a computer based management information system) will be proposed.

At the present time, performance assessment is a labor intensive effort. A computerized performance assessment system will be extremely useful and is the solution to the labor intensive effort. The results of the project will make performance assessment data collection and subsequent analysis less time consuming, freeing ourselves to work on solutions.

DEDICATION

I would like to dedicate this project to my wonderful wife Anne and our two children, Joey and Amy.

ACKNOWLEDGEMENTS

I would like to acknowledge the support and guidance provided to me by Professor Ken Harmon and Dr. Triantis of Virginia Tech.

I would also like to acknowledge the help I received from my peer and friend, Mr. Tony Couto who acted as a sounding board and provided useful critique comments.

Finally, I would like to acknowledge the AEGIS Combat Systems Center team of military, civilian and contractor personnel who consistently strive to do the best job they can in support of our customers (the AEGIS fleet and U.S. taxpayer).

TABLE OF CONTENTS

1.0 INTRODUCTION 1

1.1 Purpose 1
1.2 Background 3
1.2.1 Background of the AEGIS Program 4
1.2.2 Background of the ACSC 6
1.2.3 ACSC Users (Customers) 7
1.2.4 User Support Drivers 9
1.2.5 Multi-User Capability 10
1.3 Evolution 11

2.0 SYSTEMS ENGINEERING CONTEXT 13

2.1 Premise 13
2.2 History of Performance Measurement at ACSC 13
2.3 Identification of Need 14
2.3.1 Justification for Change 15
2.3.2 Description of Needed Changes 16
2.3.3 Plan of action 16
2.4 Identification of Current Performance Assessment Criteria 16
2.4.1 Background, Objectives and Scope 16
2.4.2 Operational Policies and Constraints 16
2.4.3 Description of Current System or Situation 17
2.4.4 Current hardware operational environment 18
2.4.5 Users or Affected Personnel 19
2.4.6 Support Concept 19
2.5 Performance Assessment Shortfall Analysis 19
2.5.1 User satisfaction feedback analysis 20
2.5.2 Planning 20
2.5.3 Operations Support 21
2.5.4 Detail Event User Planning Support 21
2.5.5 System Readiness 21
2.5.6 Maintenance Efficiency 21
2.5.7 Workaround Techniques 21
2.5.8 Critical Systems Assessment 22
2.5.9 Contractor Performance Evaluation Process 22
2.5.10 System Loading throughput 22
2.6 Action 23
2.7 Criteria to Cover Shortfall 23

2.7.1 Planning.. 24
2.7.2 Scheduling.. 26
2.7.3 Readiness... 27
2.7.4 Operations.. 28
2.7.5 Maintenance... 30
2.7.6 Engineering... 31
2.7.7 Combat System Support.. 32
2.8 Possible Grading System.. 33
2.8.1 Outstanding... 34
2.8.2 Good.. 34
2.8.3 Satisfactory.. 34
2.8.4 Unacceptable.. 35

3.0 COMBAT SYSTEM PERFORMANCE ASSESSMENT SYSTEM....... 35

3.1 CSPAS Background.. 35
3.2 CSPAS Overview.. 35
3.3 CSPAS Design Drivers... 37
3.3.1 Required States and Modes....................................... 39
3.3.2 System Capability Requirements.................................. 39
3.3.3 System External Interface Requirements.......................... 45
3.3.4 System Internal Interface Requirements.......................... 46
3.3.5 System Internal Data Requirements............................... 47
3.3.6 Adaptation Requirements... 48
3.3.7 Safety Requirements... 48
3.3.8 Security and Privacy Constraints................................ 48
3.3.9 System Environment Requirements................................. 48
3.3.10 Computer Resource Requirements................................. 51
3.3.11 System Quality Factors... 52
3.3.12 Design and Construction Constraints............................ 53
3.3.13 Personnel Related Requirements................................. 54
3.3.14 Training Related Requirements.................................. 54
3.3.15 Logistics Related Requirements................................. 54
3.3.16 Other Requirements... 54
3.3.17 Packaging Requirements... 54
3.3.18 Precedence and Criticality of Requirements..................... 55
3.4 Description of the new or modified system.......................... 55
3.5 Proposed CSPAS Operational Scenario................................ 56
3.5.1 Generation of Predefined Reports................................ 56
3.5.2 Generation of Ad-hoc Reports.................................... 57
3.5.3 Acquisition of Preventative and Corrective Maintenance Data...... 57
3.6 Analysis of Proposed System.. 57
3.6.1 Summary of Advantages... 57

3.6.2 Summary of Disadvantages/Limitations.................... 58
3.7 Qualification Provisions.. 58
3.7.1 General Qualification Methods.............................. 59

4.0 DETAILED SOFTWARE DEVELOPMENT ACTIVITIES................ 59

4.1 Project Planning and Oversight... 59
4.1.1 Software Development Process.............................. 59
4.1.2 Software Development Methods............................. 61
4.1.3 Software Development Planning............................. 62
4.1.4 CSU/CSCI Test Planning...................................... 62
4.1.5 System Test Planning.. 63
4.1.6 Software Installation Planning............................... 64
4.1.7 Software Transition (Implementation) Planning.......... 64
4.1.8 Management Reviews... 64
4.2 Established Software Development Environment................... 66
4.2.1 Software Engineering Environment......................... 66
4.2.2 Software Test Environment.................................. 67
4.2.3 Software Development Library.............................. 67
4.2.4 Software Development Files................................. 68
4.3 Systems Requirements Analysis.. 68
4.3.1 Analysis of User Input.. 68
4.3.2 Operational Concept... 68
4.3.3 System Requirements.. 68
4.4 System Design.. 69
4.5 Software Requirements.. 69
4.6 Software Design.. 69
4.7 Software Implementation and Testing................................ 69
4.8 Unit Integration and Testing... 69
4.9 CSCI Qualification (acceptance) Testing.............................. 70
4.10 CSCI/HWCI Integration and Testing.................................. 70
4.11 System Qualification (acceptance) Testing.......................... 70
4.12 Preparation for Software Use... 70
4.12.1 Prepared Executable Software............................. 70
4.12.2 Prepared Version Descriptions for the User Site........ 70
4.12.3 Prepared User Manuals..................................... 71
4.12.4 Installation at User Site..................................... 71
4.13 Preparation for Software Transition.................................. 71
4.14 Configuration Management.. 71
4.14.1 Configuration Identification................................ 71
4.14.2 Configuration Control....................................... 72
4.14.3 Configuration Status Accounting.......................... 73
4.14.4 Packaging, Storage, Handling and Delivery............. 74
4.15 Software Product Evaluation.. 74

4.16 Software System Quality Assurance................................ 74
4.16.1 Quality Assurance (QA) Plan.............................. 74
4.16.2 Quality Assurance Records................................ 74
4.16.3 Independence in Software Quality Assurance.......... 75
4.17 Corrective Action... 75
4.17.1 Problems/Change Reports................................. 75
4.17.2 Corrective Action System.................................. 75
4.18 Technical Reviews... 75
4.18.1 Monthly In-Process Briefs................................. 75
4.18.2 Mid-Task Review Brief..................................... 76
4.19 Other Development Activities.. 77
4.19.1 Final Technical Report..................................... 77
4.19.2 Subcontractor Management............................... 77

5.0 PROPOSED SOFTWARE SYSTEM DEVELOPMENT SCHEDULE... 77

6.0 CONCLUSIONS.. 78
6.1 Operational impacts.. 78
6.2 Organizational impacts.. 79
6.3 Impacts during development... 79
6.4 Concluding Remarks.. 79

APPENDICES.

A. Measurement Definitions
B. Standard Reports
C. Selection of CSPAS/MMOSS Development Software
D. Coding Standards and Naming Configurations
E. Diskette label "SENSITIVE UNCLASSIFIED"
F. Analysis of Technical and Support Documentation Requirements
G. CSU Code Inspection and Test Procedures
H. CSCI Code Inspection and Test Procedures
I. ACSC Non-tactical Computer Program Change Request
J. Quality Assurance Evaluation Plan
K. Referenced Documents
L. List of Acronyms

LIST OF FIGURES AND TABLES

Figure 1. Current CSPA Process.. 18
Figure 2. CSMMOSS Main Program Modules.......................... 37
Figure 3. CSPAS External Interface Requirements..................... 45
Figure 4. Proposed CSPA Process.. 56
Figure 5. Waterfall Model.. 60

Figure 6. Software Development Activities................................ 62
Figure 7. Software Design Team.. 66
Table 1. Assessment Factors and Related Reports....................... 41
Table 2. CSCI Identification.. 72
Table 3. Software System Development Schedule....................... 78

1.0 INTRODUCTION

1.1 Purpose

The ultimate purpose of this ENGR 5904 project is to work it in conjunction with an ongoing ACSC project so that system engineering principles can be used to enhance the end result. This project consists of two parts. First, I will identify a series of effectiveness measures which can be used by the AEGIS Combat Systems Center (ACSC) to measure combat system support effectiveness as it relates to satisfying our many and varied users. Second, utilizing the effectiveness measures identified in the first part of this report, I will propose a computerized performance assessment system (in more generic terms, a computer based management information system). I chose this topic for several reasons which are:

First, to fulfill requirements for ENGR 5904

Second, by embedding the principles and techniques which I learned during the classes I took in pursuit of my Systems Engineering masters I know that the ACSC project has been significantly enhanced. This ACSC project actually began as I was into my third class of the System Engineering curriculum. Since I had the lead in the project, I realized that the end product would be far superior by using system engineering principles.

Third, the successful completion and implementation of this project will allow ACSC to realize significant savings in several different ways.

First, there is approximately 5 manyears worth of effort being expended on the manual collection, interpretation, analysis and portrayal of data and information in the Combat Systems Department at ACSC. It is anticipated that a system such as that being proposed in this report will reduce this effort to one manyear.

Second, in addition to saving the four manyears of effort, these manyears can now be utilized in making improvements to those processes and areas which data analysis has determined are in need of improvements. This is particularly significant in that ACSC has realized a downward trend in personnel resources over the past four years. ACSC human resources consists of military, civilian and contractor personnel. In 1991 total personnel was approximately 310. The number currently is 267. At the same time, the number of hours with which we support our customers has increased from 11,000 hours per year to almost 23,000 hours per year. To use an overused expression, ACSC has truly had to learn to do more with less.

Third, in November 1996, ACSC submitted a detailed application for the Presidential Quality Award (PQA). We did not win. The area where we took the largest hit was in performance measurement. Specifically, how do you know in what areas you are weak or strong, which areas need your immediate attention, or if improvements are made, how do you know they are working.

Finally, within the past three years, this command has conducted multiple drills for the Base Realignment and Closure (BRAC) Committee. These drills are extremely labor intensive. Not only will an automated performance assessment system help us to respond to these BRAC inquiries, but more importantly we will be able to develop a comprehensive and detailed package which will allow us to put our best foot forward.

At the present time, performance assessment is a labor intensive effort. A computerized performance assessment system will be extremely useful to this command and is the solution to the labor intensive effort. The results of the ACSC project will make performance assessment data collection and subsequent analysis less time consuming, freeing ourselves to work on solutions. Furthermore, Navy wide command inspections are moving towards performance assessment criteria (e.g. how do you know how good or how bad you are doing and what are you doing about it?). And, as mentioned earlier, ACSC submitted a package for the Presidential Quality Award and received critical comments from the PQA committee on the lack of performance measurement metrics.

This ENGR 5904 project will utilize the Systems Engineering approach and process guided by project management principles as they apply to performance measurement and software development. Again, this project is being used to enhance and add value to an ongoing ACSC performance measurement project. The approach which I will take with this report will be to: (1) Identify a Need (Operational Problem), (2) Conduct Shortfall Analysis (Statement of Requirements), (3) Development of Effectiveness Measures to Address Shortfalls, and (4) Automation of Performance Assessment.

1.2 Background

1.2.1 Background of the AEGIS Program. The goals of the AEGIS Program are to allow the Navy to continue to provide "War Ready" AEGIS ships to the fleet, maintain AEGIS ships at peak operating efficiency, and continuously improve and upgrade the AEGIS ships to meet new and hostile threats. Through the integration of people, parts, paper, and programs, the AEGIS Program is able to provide the fleet with combat capable warships that are not only pacing the threat but are also keeping pace with technology advancements.

The cruisers and destroyers in these two ship classes are both equipped with the AEGIS Combat System (ACS). The ACS is unique in that it is a "top down" design, with performance specifications at every level. The intent of implementing the top-down design process, the detailed specifications, and the rigorous testing program, often characterized as "build a little, test a little" was to put to sea in the new AEGIS ships, a thoroughly tested combat system fully able to meet current and near-term threats from any possible aggressor. This goal has been achieved. Another ongoing AEGIS program goal is to ensure that, at the minimum, no regression of performance occurs throughout each ship's operational life. Another way of stating this goal is that each AEGIS ship will be at least as capable at the end of its operational life of 30 years or more as it was when the ship first entered the fleet. The need for dedicated AEGIS shore sites was recognized early in the mid-1970s. Several issues drove this need:

- A lesson had been learned from difficulties experienced during previous introductions of new shipboard combat systems that were not preceded by a rigorous, system level test program before shipboard installation.

- ACS proof-of-concept testing was vital to successful fleet introduction; that is, to demonstrate that the performance requirement behind the AEGIS cornerstones were met.

- It was important to demonstrate that the ACS equipment and computer programs could be controlled and operated by Navy personnel. A stringent training requirement existed to integrate prospective AEGIS ship officers and men with the ACS to ensure effective and efficient employment, operation, and maintenance of the ACS. This training requirement includes initial crew training, replacement crew training, and refresher training.

AEGIS Program Office planning incorporated a technical control feature known as "baselines" to allow orderly management of changes (changes were expected for a number of reasons (technological advances being perhaps the principal one) within the ACS. Baselines represent specific groups of ships that have a similar version of combat system capability. Ships can obtain a specified baseline capability either through the "forward fit" process; i.e. during initial ship construction; or as a "backfit" or retrofit installation during a scheduled Ship Restricted Availability or Regular Overhaul. Use of the baseline concept allows introduction of these upgrades while the ships are still in construction rather than "freezing" the ACS design until all the ships are built.

Additionally, using baselines breaks the overall AEGIS ship introduction and ship life cycle support task into manageable packages, while taking advantage of commonalities between the various baselines, which also helps in designing the AEGIS landbased sites or upgrading current systems. Baselines also help to identify those ACS elements at the sites that need to be shared among the ACS baselines supported by the landbased sites. Currently, there are five major baselines, numbered 1 through 5. However, variations within some of the baselines - largely reflected in ACS integration and computer program differences - result in several variants of each major baseline at sea in AEGIS ships. In addition, baselines 6 and 7 are planned.

1.2.2 Background of ACSC. The AEGIS Combat Systems Center, located at Wallops Island, first operational in December 1987, was formally established as an Echelon III command by the Chief of Naval Operations in August 1989. ACSC is a tenant of the National Aeronautics and Space Administration's Goddard Space Flight Center at the Wallops Flight Facility, Wallops Island, Virginia. Approximately six miles south of the Maryland and Virginia border, ACSC's location on Virginia's eastern shore in a maritime environment and in close proximity to Navy controlled air space, makes it an ideal site for addressing the Navy's current and future combat systems needs. The ACSC Combat Systems Site on Wallops Island is comprised of the AEGIS Cruiser and Destroyer systems and is the "Waterfront" part of the ACSC.

ACSC has become a critical AEGIS site for conducting key engineering and training events in an operational setting. The Navy has established replicas of AEGIS cruiser and destroyer combat systems in a land-based engineering and training site along with associated Quality of Life facilities. In the execution of the ACSC command Mission, stated below, combat systems equipments and overall systems must be exercised at the same level of tactical performance expected of deployed AEGIS cruisers and destroyers. For those equipments and systems required but not installed at our center, extensive simulation systems are provided for all normally expected combat systems warfare elements.

The mission of ACSC as established in OPNAV NOTICE 5450, dated 14 August 1989, is:

"To provide support for the conduct of AEGIS Combat System Lifetime Support Engineering (LSE), In-Service Engineering (ISE), System Level Operations and Maintenance Training, the conduct of engineering experiments and warfare exercises, and participation in battle group exercises and evaluations in the adjacent VACAPES OPAREA."

1.2.3 ACSC Users (Customers). ACSC accomplishes it's mission by providing support for five types of activities which are:

- **Lifetime Support Engineering.** Lifetime Support Engineering involves keeping pace with the threat by providing continuous upgrade of operational capability. This is accomplished through certification of the AEGIS weapon system and integration of the AEGIS combat system prior to ship delivery, identifying system test requirements, and developing and conducting system-level tests to thoroughly stress the system, help measure its endurance, provide fault analysis, recommend work around procedures, and ensure safety when the change is installed aboard ship. The entire tactical potential of a baseline being delivered to the fleet must be verified fully operational and certified.

- **System Level Operations and Maintenance Training.** The AEGIS Training Center, Dahlgren, Virginia, maintains a unit at the ACSC.

The AEGIS Training Unit conducts and coordinates training in operational and maintenance functions which strategically exposes the Officers and Enlisted men to each AEGIS system element and then stresses them with tactical scenarios at the full system level and tempo of a real fleet exercise or actual occurrence.

The training facility at ACSC provides a total system which looks and acts like the AEGIS weapon system aboard ship. The Cruiser Combat Information Center at ACSC, for example, is an exact representation of the CIC found on an AEGIS cruiser with the types and locations of consoles, large screen displays, lighting, etc. being the same as that found aboard ship. Through the use of extensive simulator equipments, more ships, aircraft, missiles, and other threats can be placed into the training environment (ie. increased track loads), which otherwise would not be able to occur at sea during peacetime. Sailors leave ACSC with an appreciation of what the system can do when stressed in complex scenarios.

- **In-Service Engineering.** Responsibilities of In-service Engineering involve keeping current systems working by providing fleet engineering services to address emergent shipboard problems and maintain day-to-day operational readiness. These responsibilities encompass all ship centered AEGIS combat system engineering, technical, and logistic support and include investigating fleet program casualty reports, computer program certification, ORDALT and field change proof-ins, and documentation and verification for: technical manuals, Preventative Maintenance Systems, drawings, and test plans and procedures.

- **Engineering Initiatives.** One of the cost effective by-products that resulted from having the capability of replicating any AEGIS Baseline combat system for Lifetime Support Engineering, In-Service Engineering, and Training, was that our facility could be used by Research and Development personnel who could now for example come to ACSC with their new experimental radar, and test it on each AEGIS Baseline in one waterfront location. ACSC's flexible and modular design accommodates the temporary installations of engineering experiment equipments. Engineering Initiative projects involve a wide range of R&D type experiments designed to enhance AEGIS and other fleet operational capabilities associated with the AEGIS Combat System as well as provide system enhancements, technology upgrades, new threat upgrades, tactical improvements, and deficiency solutions.

- **Warfare Exercises.** Warfare exercises include participation in battle group exercises and evaluations in the adjacent Virginia Capes Operating Area. For example, during February 1992, ACSC successfully supported Commander, Operational Test and Evaluation Force along with ten other government organizations for the Anti-Air Warfare portion of the operational test of DDG 51.

Please note that the word user and customer are used interchangeably throughout this report.

1.2.4 User Support Drivers. There are two terms which are used frequently in the ACSC community. They are fidelity and capacity (throughput), and are considered the technical, operational, and design drivers for ACSC.

Fidelity is the capability to faithfully represent all operational interfaces with the AEGIS Combat System. Capacity is the capability to increase system throughput by means of selected redundant equipment, switching capability, computers, and commercial gear without having to increase personnel or provide dedicated suites of unique baseline equipments. This capability allows ACSC to fully support 50 to 75 user events per week and pack 300 to 500 user hours into a normal operating work week of 80 hours.

1.2.5 Multi-User Capability. ACSC contains the most complete suite of combat systems elements, equipments and simulators which can be configured to represent any in-service AEGIS ship for combat system level interface testing and operational activities. As new ship classes are introduced into the AEGIS program, ACSC will be required to assume a support posture for these ships and the baselines they represent. User agencies identify their requirements, both long range and immediate, which must be fully supported by ACSC. Of particular importance is the interrelation between the tactical and simulated elements within the baseline.

Providing AEGIS combat system elements for all baselines would be prohibitively expensive and unnecessarily redundant, since there are baseline commonalities. But, providing a method to share common elements, equipments or simulators between the various baselines significantly reduces cost without sacrificing performance or capability. The ACSC interface switching system provides for this equipment sharing and allows for a single equipment group to be scheduled and utilized by various baselines. The AEGIS Data Reduction system, allows users to verify and analyze test results in near real time. An audio visual system is available to allow for real-time viewing and monitoring of operator actions and equipment responses as well as providing a historical record for later analysis.

Activation and upgrade of ACSC systems must pace the upgrades of newly delivered fleet ships and modernization or upgrades of Combat System Baselines now at sea. ACSC is constantly preparing for the activation of new baseline developments to be ready to support the first ship of an operational baseline group. The site must continually upgrade its equipment to keep abreast of ship equipment Ordnance Alterations and Field Change installations.

1.3 Evolution

When ACSC was established in 1987, initial efforts of the Combat Systems Department were expended doing basic things such as activation, weekly scheduling, defining operations, interfacing with NASA and the outside world, developing user guides, etc. Today, ACSC has more users (COMNAVSEASYSCOM tasking to ACSC has grown from zero user hours to 300 to 500 user hours per week and will continue to increase, to perhaps 1000 - 1500 hours of system level test and training, as new baselines are added), more events, new Baselines, more equipments to operate and maintain, more ADP oversight, increased contract monitoring, and more rules, restrictions, and instructions to abide by.

The mission critical effectiveness measure goal for ACSC is USER SATISFACTION. Successful achievement of this goal involves several aspects including: providing the desired environment, equipment availability, system hours, scheduling/operations, maintenance/logistics support, BQ/Galley operations, Public Works support, environmental services and more. With ACSC's many diverse objectives, we may not be capable of meeting all support requirements without limitation. However, our services must provide Users with their best opportunity to evaluate their software, hardware, and training effectiveness. To accomplish this, we must identify soft spots in our support system, improve our processes, and continuously communicate with all of our users.

There is a real need for definition of effectiveness measures which allow subjective assessment of ACSC and contractor performance. At the same time, routine feedback is needed by users and internal ACSC personnel so that we may improve the quality of overall ACSC support to the AEGIS program. We also need to be able to integrate all of our resources so Engineering, Operations, Planning, and Maintenance are all marching to the beat of the same drummer. A systems engineering approach needs to be used where we will look at the whole and not just the parts. How do actions used to support one customer impact our other customers? Are we pleasing one customer at the expense of the others? We must adopt a longer term prospective to enable our users to work with us in dovetailing support for effective throughput as system loading increases.

Therefore, in order to improve the services we provide to our customers, we must continuously:

- Enhance ACSC/User communications
- Analyze current performance assessment systems
- Define methodology for measuring support effectiveness
- Determine user satisfaction measurement factors with ACSC

support

- Improve our assessment systems as required
- Improve our real time process to collect and analyze performance data
- Determine areas needing further improvement and corrective actions to enhance them.

2.0 SYSTEMS ENGINEERING CONTEXT

2.1 Premise

Dependable availability of AEGIS systems technical support for AEGIS program users is essential to our mission. Current assessment systems are inadequate for present needs and unsatisfactory for mission operations of the future. A systems engineering approach must be utilized which will allow for examination of the current situation in its full environment leading to a systems solution of the whole problem or challenge. The challenge in this case is performance assessment of the Combat Systems Department functions at ACSC.

2.2 History of Performance Measurement at ACSC

Fully successful achievement of mission objectives at AEGIS Combat Systems Center requires the use of multiple CSLAN based information management and control systems. Since inception of mission user support, operation event planning and scheduling remain the key to aligning capability, readiness and availability with command user requirements. Scheduling tools were developed which have fundamentally met ACSC user needs to date. A customer requirements definition system was added as well as statistical applications, forecasting systems and a user satisfaction evaluation system.

Independently, software tools were developed to help manage the large volume of media required to support users. Also independently, a small local ACSC application, the Combat Systems Officer of the Watch Logbook Equipment Program (CLEP) was added to track the status of equipment and systems that were in need of supply parts and maintenance. Also, ACSC provides significant input to the Navy Micro-Snap data base and to AEGIS Centralized Equipment Data System (ACEDS) for configuration management. Other locally generated software system applications exist for system and facility drawings.

In addition to the few automated but separate applications mentioned above, there are also numerous reports prepared by manual means which are used to assess performance. These methods are not automated but rather have to be performed again and again as the need arises. Each collection and assessment of any kind of data has always been a time intensive exercise.

2.3 Identification of Need

2.3.1 Justification for change. With reduced resources and growing mission workloads, ACSC is faced with continually improving the quality of customer service and overall support performance. Scheduling conflicts often result in sharing of site resources by ACSC users. This, along with the need to provide reliable technical services for all ACSC users, necessitates systematic assessment of AEGIS Combat Systems (ACS) performance indicators at ACSC.

The independent nature of the current CSLAN configuration has not significantly impacted ACSC effectiveness until this past year where more complete sharing of operations and readiness information to support mission loading and maintain a growing list of installed systems was needed. The variety and complexity of ACSC mission loading and the requirement for peak system performance are driving factors for an improved and fully integrated technical information management system. Driven by loading, complexity and personnel constraints, combat systems operations and readiness requirements mandate timely and accurate exchange of data and decision information to insure effective mission support operations for customers.

Obtaining and assessing meaningful data of this nature is currently performed through a variety of methods. Contractor personnel provide operations analysis support by copying, converting, and analyzing event-related data at ACSC. This process is often repetitive and somewhat labor intensive.

2.3.2 Description of needed changes. In order to facilitate ACSC performance assessment, a more flexible, user-friendly, and reliable system is needed. The program must be flexible enough to accommodate changes in ACSC's operating environment. Data integrity must be addressed and a database engine should be placed into operation to ensure efficient and reliable data storage and retrieval. Software to be developed must minimize the effort required to collect and assess data about ACSC mission support performance. Reports which are produced regularly need to be available on the CSLAN for easy generation by appropriate users.

2.3.3 Plan of action. Based upon the description of needed changes mentioned above, a plan of action needs to be developed which would contain the following actions:

- Define existing performance levels
- Identify current measures of performance
- Identify areas where additional performance measures are needed
- Structure a performance monitoring and assessment system
- Investigate problem area causes
- Develop new measures and modify existing techniques as required
- Develop options for improved resolution
- Recommend optimum solution
- Implement solution
- Receive and evaluate feedback

2.4 Identification of Current Performance Assessment Criteria

2.4.1 Background, objectives and scope. With the AEGIS program and Navy-wide focus on continuous customer support process improvement, a structured performance measurement system is essential. An improved system for assessing the overall effectiveness of combat systems technical support, engineering, system readiness, scheduling, and mission loading in the conduct of mission support operations for ACSC is needed.

2.4.2 Operational policies and constraints. The current newly automated process of the ACSC performance assessment reporting is a Microsoft Disk

Operating System (MS-DOS)-based application, utilizing separate, non-integrated data files on the ACSC Combat Systems Local Area Network (CSLAN). It requires manual copying of data from the current program by Operations Analysis personnel. Consequently, data from one application is not readily available for use in another application, creating redundancy in data storage and entry and the possibility of a compromise in the retrieval of data for a particular period of time.

2.4.3 Description of current system or situation. Currently, many methods are utilized to track performance at ACSC. Data from the ACSC scheduling software, currently the AEGIS Training and Engineering Accessibility Model (ATEAM), is copied daily by LM Operations Analysis personnel and converted to a local MS ACCESS database application. After duplication, statistics are compiled to report on the data collected. Additional data is entered into the MS ACCESS database, including event setup times and event availability. The data is exported to other software applications such as MS EXCEL, MS WORD and STATISTICA to produce reports and charts which are delivered to Code 2400 for distribution to key command personnel on a weekly or required basis. Figure 1 depicts the Current Combat Systems Performance Assessment Process.

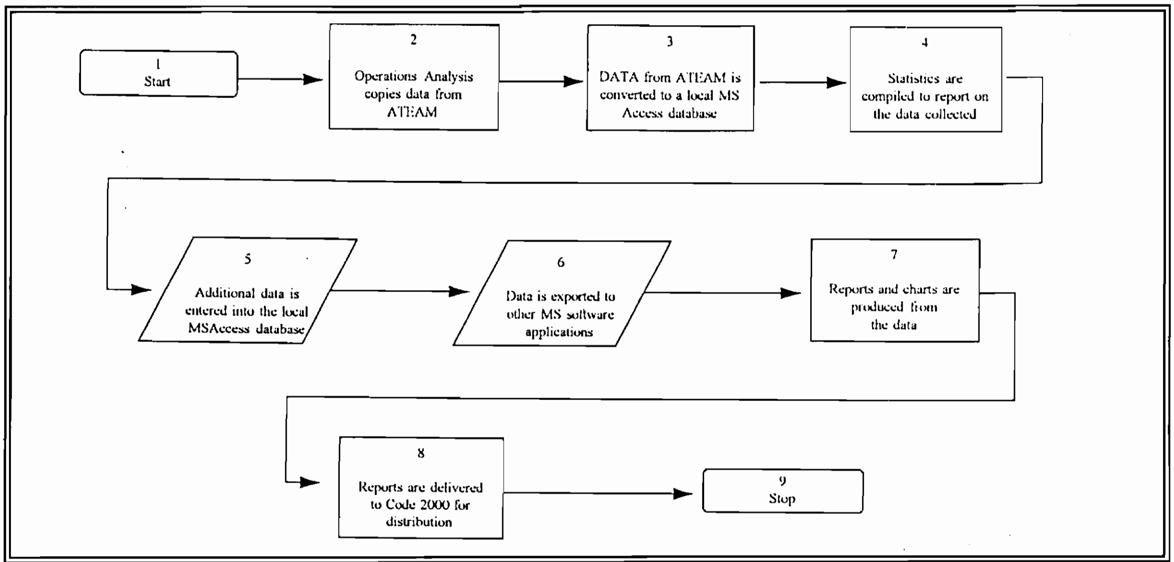


Figure 1. Current Combat Systems Performance Assessment Process

Other areas through which we assess how well we are doing include:

- User surveys - conducted randomly and assessed quarterly
- PMS400 feedback
- Internal equipment maintenance reports
    - PMS records (3-M)
    - Equipment casualty reports
    - Equipment down time
- Meetings
- CCmail and Email
- Total Quality Leadership initiatives
- Informal communications

2.4.4 Current hardware operational environment. The current hardware operational environment is a Novell NetWare client/server configuration consisting of approximately 125 local workstations residing on the CSLAN and capable of supporting up to 100 users simultaneously through the ACSC

Wide-Area Network (AWAN). More detailed information on both the physical environment and the installed software can be found later in sections 3.3.9.1 and 3.3.9.2 respectively.

2.4.5 Users or affected personnel. Personnel affected by the current performance assessment methods employed at ACSC are Code 2000 personnel, the ACSC Commanding Officer, and LM Operations Analysis personnel.

2.4.6 Support concept. Presently, there is no single software package in use for the accomplishment of performance assessment at ACSC. Support is provided in the form of analysis and data collection efforts by Code 2400 and LM Operations Analysis personnel.

The current ACSC user feedback process consisting of the number 1 (mission accomplished - no problems), 2 (mission accomplished - minor problems), 3 (mission not accomplished due to ACSC), and 4 (mission not accomplished due to outside influence such as power outage or lightning strike) grades is workable but crude. Even with the crude system we get lots of press if we receive a three grade. There are times when the dissatisfied user will bypass the Code 2000 Department and call the Commanding Officer directly. This is not a desirable situation. Better measures and trends must be used to promote fairness, problem definition, and effective enhancement of the overall quality of combat systems support.

2.5 Performance Assessment Shortfall Analysis

There are several combat systems department areas that require a carefully structured systems engineering effort in Code 2000. Our goal is to quantify performance parameters and assessments in a manner that allows inputting of data and generation of real time assessment. The approach taken to achieve this goal will be to:

- solicit user feedback
- utilize existing data from current system to the fullest extent possible
- establish additional assessment systems where necessary
- tailor assessment system report to various management activities
- implement a real time system for rapid effectiveness assessment

In keeping with this approach, we have conducted multiple brainstorming sessions with our users to come up with a list of performance indicators through which we could judge ourself. By priority, the shortfall primary areas of concern related to effectiveness measures are as follows:

2.5.1 User satisfaction feedback analysis. An overlay of the degree of satisfaction achieved by the individual ACSC user (LSE, ISE, ATC, or Project) upon completion of each user event for which ACSC has established a system which involves different quantities and types of participants. It is extremely important for ACSC to consistently produce a high level of user satisfaction. This satisfaction level is the essence of our value to the AEGIS Program.

2.5.2 Planning. Site unique pre-planning, assessment and negotiations required to arrive at a workable combat systems support shift that fully satisfies all approved users of ACSC while allowing adequate system time for maintenance and activation activity. Planning issues include meeting requirements, commitments, event linkage(s), analysis and site operations control.

2.5.3 Operations Support. The total range of required elements essential to the successful conduct of ACSC supported operations which as a minimum includes tactical and/or technical operator personnel, special services, documentation, critical assessment of system readiness, data support and, often, external services such as air craft.

2.5.4 Detail Event User Planning Support. The depth of capability available to provide complete mission event services including test (or training) plans, procedures, data reduction/analysis, workaround configurations (where all live systems may not be available) and other tactical or non-tactical detailed planning essential to scheduling parallel mission event activity at ACSC.

2.5.5 System Readiness. Assessment and maintenance of a specified state of readiness for all ACSC systems, equipments, computer programs, and simulation systems installed and available for mission users at ACSC. System readiness also includes the time required for set up of the required system configuration.

2.5.6 Maintenance Efficiency. As system and equipment time is critically short with many users waiting, efficiency of maintenance planning and execution is a valid measure of success.

2.5.7 Workaround Techniques. The degree of ACSC site integration and interdependency is extremely high; thus, when selected systems or equipments, etc. fail or are degraded, the process by which alternative techniques are developed to continue to provide required user support (defined as workarounds) is also a measure of effectiveness.

2.5.8 Critical Systems Assessment. Tracking of all work on all functions and deliverables and a systematic review and reporting process which is kept up to date in a computer based environment is key to our present and future.

2.5.9 Contractor Performance Evaluation Process. An effective and dependable performance evaluation process with reliable measures must be developed to support effective contractor performance. The overall process must be able to cover ACSC performance so that contractor performance can be cleanly defined. For example, the contractor should not be held accountable if the system and personnel are up and ready but the visiting user decides to take high risk with his development software or procedure. Conversely, the contractor should be held accountable if his recommended throughput process for multiple user shift support fails to work for reasons he should have corrected. Products from this process definition effort are mandatory for contract performance assessment and would be very useful on our present contract.

2.5.10 System Loading throughput. This area involves innovation of design and procedure to make more efficient, those capabilities and capacities available at ACSC to meet increasing mission needs. Some AEGIS sites avoid this problem by dedicating a single user hour by hour of shift by shift. With multiple AEGIS mission areas, our combat systems and equipments must be ready to support a full load of parallel user agents with effectiveness for each. Improved techniques need to be defined; and the search for workarounds and better use of our systems requires study.

2.6 Action

Given valid shortfalls in process and procedure, there is a need to take corrective action and evaluate closure on problems identified for combat systems performance and overall support effectiveness. The first step therefore is to establish the model and define the parameters before developing equations to solve the problems.

2.7 Criteria to Cover Shortfall

As a result of the shortfall areas identified above, I convened additional meetings with a selected group of individuals to identify areas which we could target for measuring effectiveness. As a guide to assist us in this process we used the ACSC User Support Process which can be found on the first page of Appendix 1. This was useful for several reasons. First, it provided a reference point which everyone could work from. Second, we were fairly sure that any measure which could be linked to this User Support process (a process which has been validated by the ACSC users), was probably a worthwhile measure. Third, if we could link these measures to a customer support process flow chart, we would be in a much better position to handle the deficiencies sighted by the PQA committee. Finally, it would be extremely beneficial to link challenges or problems with a particular User, shift, day of week, specific element, etc. Therefore, we identified the following 25 Quality Effectiveness Measures and affinitized them into one of 7 (Q1 - Q7) categories:

2.7.1 Planning (Q1). AEGIS Program approved users of the AEGIS Combat Systems and equipments submit immediate, short term and long term requirements for ACSC system support. Critical analysis of the user requirements based upon ACSC capability and capacity to provide system time is conducted. This normally permits orderly development of operations and test plans as required. Such planning provides the lead time to evaluate user operational and functional support requirements together with ACSC loading commitments to best satisfy all user requirements. ACSC long term requirement support is translated into optimum resource allocation planning and drives activation and engineering of new user support capability. User feedback is requested in the form of lessons learned and customer satisfaction rating which complete the planning model. Specific measurement criteria include:

- **User Requests Timely.** User request submitted in advance of published schedule. Requests submitted within 30-90 days prior to event date are considered very timely, those submitted three to six months (with no changes) exceptionally timely. Requests submitted by users two weeks prior to their event can normally be accommodated. However, waiting till two weeks prior to event to place a request almost always ensures that a conflict between users needing the same system/element/equipment will take place. When requests are submitted after the schedule has been published timeliness is negative.

- **User Requests Achievable.** Whether or not User requests are achievable is dependent on the following situations: (1) ACSC capability and what cannot be supported, (2) planned capability (equipment on site but not installed), (3) capability not planned (equipment is not planned or funded to be installed at the site), (4) priority of workload resources (i.e. sailor not qualified) and (5) capacity or loading issues. If we have educated our Users so that they understand ACSC capabilities, the likelihood of achievable requests being submitted becomes much greater. However, a high percentage of unachievable requests does not necessarily mean that the User is uneducated about ACSC capabilities or that ACSC is doing a poor job of scheduling. Rather, a high percentage of unachievable requests could be used as justification for more equipments or resources to accommodate our Users.

- **Event Planning Effective.** Event was scheduled and equipment available - Plan executable. Event executable to plan and schedule without having to move or rearrange other events or modify requirements. The percentage of schedule changes or modifications would be tracked and monitored.

- **External Interfaces Closed.** Services with NASA, PAX River, FACSFAC, Oceana and Coast Guard are sometimes required by ACSC users and need to be fully coordinated with the ACSC Scheduling Department. These support services often involve aircraft and as such bring with them the uncertain element of the weather. Although weather cannot be helped, good planning and coordination of the event and how it effects the other ACSC customers is important. Which services are hardest to coordinate and do the users cancel the services in advance when not needed are two questions which can be asked and answers tracked.

2.7.2 Scheduling (Q2). ACSC user workload is the catalyst for the development and maintenance of our paperless scheduling process (as defined by ACSCINST 3967.1B). This ADP/LAN allows ACSC to maximize user load throughput, and not degrade support for user requirements. ACSC scheduling techniques allow for equipment multiplexing, or shared use, whenever practical; thereby, allowing ACSC to maximize user load throughput and satisfy user requirements for support. The process requires analysis of user requirements, preliminary scheduling thruput, rotating week scheduling, user coordination, scheduling of the event week, publishing of ACSC schedules and rapid schedule adjustments (by the Combat Systems Officer of the Watch (CSOOW)) during the period of schedule distribution through the week of event execution. Limited availability and high cost of combat system equipment demand multiplexing the support load to meet the multiple user needs at ACSC. Specific measurement criteria include:

- **Forecast Accuracy.** Is our ability to forecast good, useful and accurate? The ratio of forecasted activity to actual is determined by the length of the forecasted period and accuracy (e.g. 3 months 90%, 18 months 60% and 8 years 30% accurate). For example, if it is January and we prepare a forecast for April, we would want to compare our forecast for April to what actually happened in April, noting the similarities and differences. Where did the biggest discrepancies occur and with what user and during what shift and were there any unusual circumstances such as a major fleet problem, weather, unplanned maintenance, etc.

- **Published vs. Actual Schedules.** The difference between the published weekly schedule and the events which actually take place.

Planning effectiveness and event supportability come into play in assessment. Similar to above, this metric is used to compare what is scheduled with what actually happens, noting the differences and similarities. Schedule stability is key to effectiveness.

- **Throughput Level/shortfalls**. Throughput is ACSC ability to handle the total load regardless of the priority. What equipment or system interfaces may be causing shortfalls of throughput? Is shortfall occurring on a regular basis? Which user is most frequently encountering shortfalls? Which piece of equipment is causing a shortfall? Mission shortfall created by the lack of equipment can lead to better procurement and activation of equipments also provides justification or increased or new capabilities. One measure is multiple combinations of the following metrics: requested hours required, scheduled hours, actual hours, systems hours provided, clock hours (that is the hours that the site is available to users which is 80 (8 hours per day X 5 days per week X 2 shifts per week)) or a straight 168 hours per week.

2.7.3 Readiness (Q3). Ensuring 100% availability for all ACSC scheduled users while being prepared to rapidly respond to last minute schedule changes dictates the level of required readiness for ACSC. Configurations and set up of the AEGIS Combat System by ship baselines to the user's needs in a timely manner, providing system flexibility (work around) and proper system set up is considered a function of ACSC support readiness. System/equipment availability is critical to the readiness status. Readiness implies high levels of training for support personnel; particularly in the areas of switching and overall equipment maintenance. Specific measurement criteria include:

- **System Available on Time**. System/element/equipment support is available and on time, in working order, manned and configured properly and ready for use at start of event. Because there is a schedule with multiple users scheduled throughout the day at different times, we cannot simply just shift the schedule to accommodate the system not being ready. A system not ready for a user negatively impacts the time available to that user and puts successful completion of the event at risk. Another area that has to be addressed with this measure are those Users who are not getting off the system promptly when their scheduled time has elapsed. Both of these measurements would be measured in minutes.

- **Systems Remain On line**. System/elements/equipments remain up and fully operational for the scheduled user event for the total time scheduled. Systems remain at a high state of readiness for use at any time.

- **Key Element Down Time.** If element or interface goes down during the event, what potential does it have to take system down. How much are the sub-elements of the AWS or ACS down? This would be recorded in minutes and treated as a ratio of system/element up vs. down.

2.7.4 Operations (Q4). All user events are considered ACSC "operations". Operations are conducted at ACSC to assist training, lifetime support engineering (LSE), in-service engineering (ISE) and engineering in accomplishing their objectives. The effectiveness of ACSC's support to user operations is directly related to the completeness and accuracy of user provided OP/Test Plans and other planning documents. Operations can entail the use of an entire combat system suite of equipment or individual elements of systems. Interaction between AEGIS Combat systems and external activities (ships at sea, aircraft, and other land activities) provide users with total support capability. Operational support may include providing interfaces with NASA facilities and resources, communications circuits including links with FACSFAC, PAX River and Dam Neck and establishing satellite links. Military support is provided in the areas of aircraft control, console and computer operators and test operations management. (The ACSC operations process is defined in ACSCINST 3121). Specific measurement criteria include:

- **Operations Completed to Plan**. The difference between scheduling (mentioned above) and Operations planning is that the User is involved in the scheduling process. How we accommodate his her schedule request is up to us and is detailed in the Operations Plan. The more time allowed for OPS planning usually means a greater chance for event success and full utilization of our facility. Deviation of Operational Plan involving major external users will be assessed. If the plan was not complete at submittal there is a greater chance that the event will fail or at a minimum, experience some problems.

- **Problems low/documented**. All problems will be documented. The number and type of problems should be addressed. Air ops status should be written down and passed on to the department head (through the chain of command).

- **Safe/Secure/Coordinated.** All safety and security regulations must be followed and services coordinated with external participants. Any deviation from this shall be recorded. Services requested which were not used (e.g. adverse weather condition (no fault)) will be recorded.

2.7.5 Maintenance (Q5). Maintenance efficiency is defined as maintaining the required availability and user readiness requirements with minimum dedicated system level maintenance time (i.e. 8 hours per day or less). Preventive maintenance (PM) of the combat system equipment is scheduled to satisfy 3M requirements. 3M feedback reports are analyzed to improve the efficiency of the maintenance process. Effective PM execution reduces the amount of system down time for corrective maintenance (CM) thereby achieving the optimal system availability for all users while ensuring all baseline systems capabilities. ACSC's goal is maximum availability of all systems, equipment and supporting computer program software for the user. Specific measurement criteria include:

- **All 3M Requirements Met.** 3M requirements have been met by assigned or tasked personnel. What equipments required dedicated system maintenance time? How can we be able to have less dedicated system time for maintenance of equipment with more maintenance to be accomplished?

- **Maintenance Throughput Ratio.** How many hours of CM and PM did we do for every piece of equipment (internal and external). How many of these repairs require dedicated system time. What is the ratio of total maintenance accomplished to dedicated systems hours available?

- **Corrective Maintenance Low.** Regular preventive maintenance should reduce the need for CM thereby lessen down time. Recording both CM and PM will allow comparison analysis.

- **All System Elements/Equipments Modes Up.** Mission critical system/elements as defined in all modes of operations are up (e.g. SPY-1) All modes are available to the user as required. Downtimes would be recorded and compared to PM and CM performed on the respective elements/equipment.

2.7.6 Engineering (Q6). Long term technical requirements are identified by ACSC users for future planning and engineering designs leading to upgraded capability or capacity for future user support. Engineering support during site operations includes: design of special interfaces between user and combat systems, review of ECPs, CADs and tech briefs for potential site impact. System engineering provides the process to maximize system throughput capability while meeting design specifications. Achieving Required Operational Date(s) (ROD) to schedule and fully integrate AEGIS combat systems allows users to perform multiple events on multiple baselines at the same time to meet increasing fleet support requirements. Specific measurement criteria include:

- **Design Requirements Met.** Most of the engineering work performed at ACSC is driven from outside sources. For example, new baselines are developed at the Combat Systems Engineering Development Site (CSEDS) in Morrestown, New Jersey. Once a baseline is developed and validated at CSEDS, the baseline is delivered to ACSC for incorporation into our facility. Prior to the baseline arriving at ACSC, engineers at ACSC have prepared a design for baseline installation. Metrics for this measurement include number of design changes and delays.

- **System Interface Work**. Do the system interfaces work? After an event has been set up, the system is turned over to the user. Interface problems (if any) will be uncovered at this point. The system interfaces have to be correct particularly if the user is trying to replicate a fleet problem(s).

- **Capability Throughput Up**. As we activate more equipment, are we degrading or increasing our capability to handle multiple users. A comparison and analysis of hours provided to the users, event grades and number of activated pieces of equipment could provide useful information.

- **Required Operation Date (ROD) Capability to Schedule**. ROD can be scheduled; if it slips, by how much and why? Required Operation Date for a new piece of equipment getting installed and ready for use as planned. Any deviation from the schedule can plotted and analyzed for impact. If not installed on schedule, what are the impacts to the user(s)? What are the impacts to the fleet?

2.7.7 Combat System Support (Q7). Users may request services from ACSC in logistics, vault services, quality assurance, operations support, facilities services, data reduction, environmental impact assessments, configuration management, CSOOW controlled shifts, duty test director and ADP/LAN operations for an event. Assessment of functional support products in these areas and overall contribution to achieving user objectives leads to effective use of total command resources and feedback into the planning process. Specific measurement criteria include:

- **Facility Operational.** Support fully planned and provided in the following typical metric areas: AEGIS Data Reduction (ADAR), COMMS, environmental assessments (weather, RADHAZ), library (personnel, technical publications and manuals), console and computer operator (present/not present), configuration management (properly recorded for future use), and Combat Systems Officer of the Watch (CSOOW) comments.

- **Logistics Effective.** Logistics support functions provided to have parts on site in timely manner. How long does it take to get mission critical parts on site after part has been ordered? What are our spare parts requirements (COSBAL/COSEL)

- **ADP/LAN Operational.** 100% availability of approved LAN applications for all users.

- **Environment Support.** Water, air, electrical power, security access, etc. up and available.

2.8 Possible Grading System.

Many of these measurements are related to one another and a deficiency in one area will more than likely result in a deficiency in one or more other areas. Because none of these measurements operate in a vacuum, I feel it is a good example of how system engineering principles can be applied to measure user satisfaction. I am continuing to refine these measurements but the logical next step is to begin the process of converting these verbal descriptions into mathematical formulas. Once the formulas have been written and tested, a grading scheme will need to be created. I have taken a grading scheme which I developed for another effort and modified it to provide a sample of what one may look like for Maintenance (2.7.5 - All System Elements/Equipments Modes Up).

2.8.1 Outstanding. (93% - 100%) The system (defined as all elements - people, parts, paper, and programs - controlled by ACSC) is fully and completely responsive to the user requirements. Any minor faults are considered to be of such small consequence that their impact on the overall event is negligible. "Outstanding" indicates that in terms of the specific event parameters, criterion or subcriterion, the system meets or exceeds the highest expectations of the user.

2.8.2 Good. (80% - 87%) The system is responsive to the requirements of the user with no major weaknesses. "Good" means the system meets or exceeds the expectations of the user and the test was conducted and met test plan requirements.

2.8.3 Satisfactory. (66% - 73%) Although major weaknesses were detected in the system, workarounds enabled the user to successfully conduct the test with minimal negative impact. "Satisfactory" indicates that in terms of the specific event parameters, criterion, or subcriterion, the user satisfactorily completed the test.

2.8.4 Unacceptable. (0%) The system failed to meet specific event parameters, criterion or subcriterion. The test as conducted did not meet the test plan requirements.

3.0 COMBAT SYSTEM PERFORMANCE ASSESSMENT SYSTEM

3.1 CSPAS Background

ACSC users are requested to assess the achievement of their individual mission event objectives in test and training activities. The most visible of the currently available tools is the evaluation (grading) system referred to in Section 2.4.5. When mission users and ACSC support personnel provide mission event comments, a strong basis for problem identification and process improvement is provided. Obtaining and assessing meaningful data is currently labor intensive, because many possible measures of performance are not being recorded. For example, system operator proficiency in system reconfiguration for upcoming mission events is not measured at this time. With the AEGIS program and Navy-wide focus on continuous customer support process improvement, a structured performance measurement is essential. Although this data is collected and analyzed through various semi-automated methods, a centralized system to facilitate this process is required.

3.2 CSPAS Overview

The AEGIS Combat Systems Center (ACSC) Combat Systems Performance Assessment System (CSPAS) will be a comprehensive and compatible computer program module designed to evaluate total ACSC mission support capabilities. It will be a subset module integrated within the new Combat Systems Media, Maintenance and Operations Scheduling System (CSMMOSS). CSMMOSS, which is still under development, is an integrated set of Windows based applications with a single Database Management System consisting of:

- PowerBuilder and C++ Object Oriented programming languages which:
    - are optimized for Windows95
    - provide user friendly graphical user interface
    - provide extensive on-line help files
- Oracle DBMS for data management, storage, and retrieval which:
    - reduces requirements for redundant data entry
    - minimizes possibility of data errors and duplicate entries
    - maximizes ability to access data in the right format for all CSMMOSS applications

CSPAS will be an ACSC Wide Area Network (AWAN)/Combat System Local Area Network (CSLAN) compatible computer program module. It will be used for data collection, assessment and reporting of ACSC performance.

In general, CSPAS will provide performance assessment reporting with reproducible and retrievable report capability. Performance tracking will occur in the primary areas of system performance listed in Section 3.3

CSPAS is intended to extract, manipulate, and display information from these areas to aid in the evaluation of ACSC's total mission support capability and performance. Data to be used for reporting will originate in the CSMMOSS Site Scheduling Module, the Navy's MicroSnap program, and other CSMMOSS modules as they are developed and introduced onto the CSLAN. This data will include historical data to be collected from existing ACSC software systems (such as ATEAM and CLEP) and placed into a database for reporting purposes. Figure 2 depicts the remaining CSMMOSS program modules to be developed.

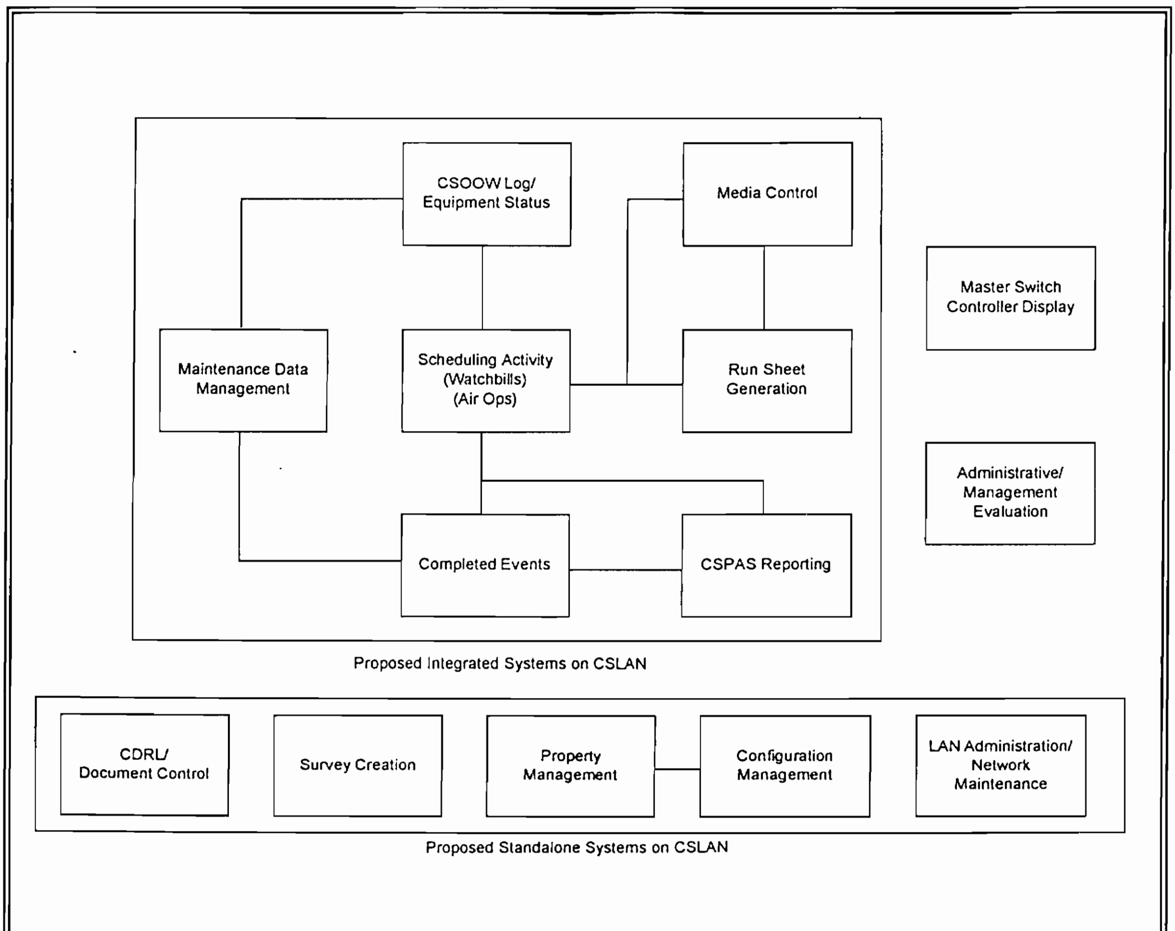


Figure 2. Remaining CSMMOSS Program Modules to be Developed

3.3 CSPAS Design Drivers

This paragraph and its subparagraphs specify the system requirements for

CSPAS, including characteristics which are conditions for its acceptance. This project will not specify design characteristics such as data file descriptions or code module descriptions. That information will be provided in the CSPAS Functional Software System Specification. Parameters for performance tracking are defined for all primary areas of system performance. These mission areas include:

- **Mission Loading** (planning, forecasting, scheduling, throughput of loading);

- **Event Readiness** (availability, reliability, maintenance effectiveness, reconfiguration timing, setup);

- **Event Operations** (individual user event support quality, dependability);

- **Technical Effectiveness** (capability, engineering, activation, design controls); and

- **Quality Assurance** (for combat systems and equipments)

Appendix A describes the calculations used by ACSC to measure performance. These values, once calculated, will be entered into the CSPAS database to facilitate the monitoring of ACSC equipment, computer programs, personnel and effectiveness of operations through the use of predefined, standard reports. The structured reports will provide some flexibility (within the basic framework of each report) for the user to vary the data items to display. Appendix B contains examples of the required standard reports to be included in CSPAS. In addition, an ad hoc query capability will be provided through the use of commercial off-the-shelf (COTS) software packages (such as Impromptu and Powerplay) to allow analysis of data not represented in the standard reports.

3.3.1 Required states and modes. No specialized states or modes are required.

3.3.2 System capability requirements. This paragraph and its subparagraphs itemize the identified area of related requirements for each function of the subsystem proposed.

- **General requirements.** The following requirements pertain to the overall operation of the CSPAS software system.

    - **Reproducibility.** All reports generated within CSPAS shall be reproducible. They can be created and saved for future re-use at the user's discretion.

    - **Retrievable report format files.** A retrievable report format file will be provided.

    - **Negative impact findings.** On color charts or reports, negative impact findings (as defined for each report) will be shown in red.

    - **Use of color.** Colors and patterns will be drawn from the Microsoft Windows default color palette.

- **User responsiveness.** CSPAS shall be menu driven with pull-down help screens in default Microsoft Windows color schemes and function with minimal keystrokes to accomplish the standard formatted reports for users. The reporting module shall allow selection of report options for standard reports. A progress indicator shall be displayed during report execution.

- **Flexibility.** An ad hoc query capability shall be provided via the COTS packages Impromptu and PowerPlay for accessible combat systems performance data. Appendix C contains reports on the selection of CSPAS/MMOSS development software. The ability to import/export a dBASE III/IV data format file and an Excel spreadsheet file shall be provided. In addition, the ad hoc reporting module will provide alternatives for presenting on-line statistical analysis and review of trends.

- **Specific requirements.** The following requirements address specific areas of CSPAS operation.

  - **Assessment factors.** Several factors are to be considered in the performance assessment of ACSC Combat Systems support processes for mission customers. These include system loading, scheduling, forecasting, equipment availability, system readiness, user satisfaction, and process improvement feedback. Maintenance and engineering effectiveness shall be addressed, although data may be partially narrative in nature. In general, CSPAS will track ACSC customer experiences from advance planning through scheduling and completion of mission events. Table 1 below contains assessment factors and related reports for areas of operations which will be assessed. The format, options, and calculations for each report are described in detail in Appendix B, Standard Reports.

Table 1. Assessment Factors and Related Reports

SHORTFALL AREA (Sect. 2.5)	**ASSESSMENT AREA (Sect. 2.7)**	**SPECIFIC REPORT(s) (Appendix B)**
System readiness	Element Failure Rate	**B.2** ACSC Element Failure Rate
System Loading Throughput	Equipment Shortfalls	**B.14** ACSC Equipment Shortfalls
Maintenance Eff	Maintenance Costs	**B.3** ACSC Maintenance Costs
Maintenance Efficiency	Maintenance Effectiveness	**B.6** ACSC Monthly Preventive and Corrective Maintenance **B.7** ACSC Causes of Corrective Maintenance **B.8** ACSC Impact of Maintenance on User Satisfaction **B.9** ACSC Preventive Maintenance (Actual v. Scheduled)
Planning	Mission Loading	**B.11** ACSC Mission Loading (Scheduled v. Requested) **B.12** ACSC Mission Loading (Completed v. Scheduled) **B.13** ACSC Mission Loading (Requested v. Completed)

SHORTFALL AREA (Sect. 2.5)	ASSESSMENT AREA (Sect. 2.7)	SPECIFIC REPORT(s) (Appendix B)
System Readiness	Equipment Availability	**B.17** ACSC Equipment Availability
Maintenance Efficiency	Equipment Reliability	**B.2** ACSC Element Failure Rate **B.7** ACSC Causes of Corrective Maintenance
User Satisfaction Feedback Analysis	Event Dependability	**B.10** ACSC Dependability
User Satisfaction Feedback Analysis	Event Execution	**B.4** ACSC User Feedback **B.5** ACSC Mission User Evaluation
System Readiness	Setup Time	**B.15** ACSC Setup Time by User Group **B.16** ACSC Setup Time by Event Classification

As can be seen by the above example, a highly disciplined approach was taken to ensure that the charts produced by CSPAS can be directly linked to an effectiveness measure and shortfall. However, it is important to note that a CSPAS chart has not been created for all of the shortfalls and effectiveness measures identified in this report. There are two reasons for this. The primary reason is that some of the effectiveness measures and shortfalls have not been adequately studied. The other reason is that it was determined that CSPAS would be an evolving process in which new charts would be created as they were needed. Charts can be created in one of two ways. The first way will be for a programmer to create the chart (based on validated requirements) and add it to the CSPAS program making it a standard chart. The second way will be for the user of CSPAS to create the chart with the add hoc reporting tool capability of the program. It was not by accident that this ad hoc capability was designed into CSPAS. As new types and kinds of data are added to the CSMMOSS database, users of CSPAS will need a way to be able to perform analysis of this information.

- **Measurement criteria.** Appendix A contains the background information pertaining to specific performance assessment factors which will provide certain data items for the CSPAS database. Criteria for performance assessment algorithms shall be defined by formula and/or data source, and shall be directly related to the areas of operation addressed in section 3.3.2.2.1. Provision shall be made for additional criteria to be developed or relationships to parameters to be modified while minimizing reprogramming effort.

- **Data availability.** Data shall be obtained from existing data available on the CSLAN. Source data files created by programs currently in existence on the CSLAN will not be modified or opened by any program having write capabilities, but will be copied at scheduled intervals after determining that the source files are not in use. Selected information from the copied files will then be placed into a centralized relational database for further manipulation. Interface capability with external databases (including SNAP, CLEP, and the CSMMOSS database) shall be exploited to minimize manual activity in collection and assessments. The capability to manually enter data and narrative assessment information shall be provided for those data items which do not currently exist and for which analysis is desired. More detail regarding acquisition of data can be found in paragraph 3.3.3, System external interface requirements.

- **Historical analysis.** CSPAS shall provide reporting of historical data. Historical analysis normally encompasses data from the previous three years and can be extracted and expressed in yearly, quarterly, monthly or weekly formats.

- **Predictive analysis.** The capability to graphically forecast assessment factors or criterion performance for five years is desired. Requirements for this feature are currently unspecified and will likely require modifications to ACSC's scheduling programs.

- **Future compatibility.** Design and programming of CSPAS will consider advancing computer systems and information science. Wherever possible, the system will be designed to maximize compatibility with future software packages. A phased or graduated level of development is desired for module development and validation.

3.3.3 System external interface requirements.

- **Interface identification and diagrams.** Figure 3 depicts the external interface required for CSPAS.

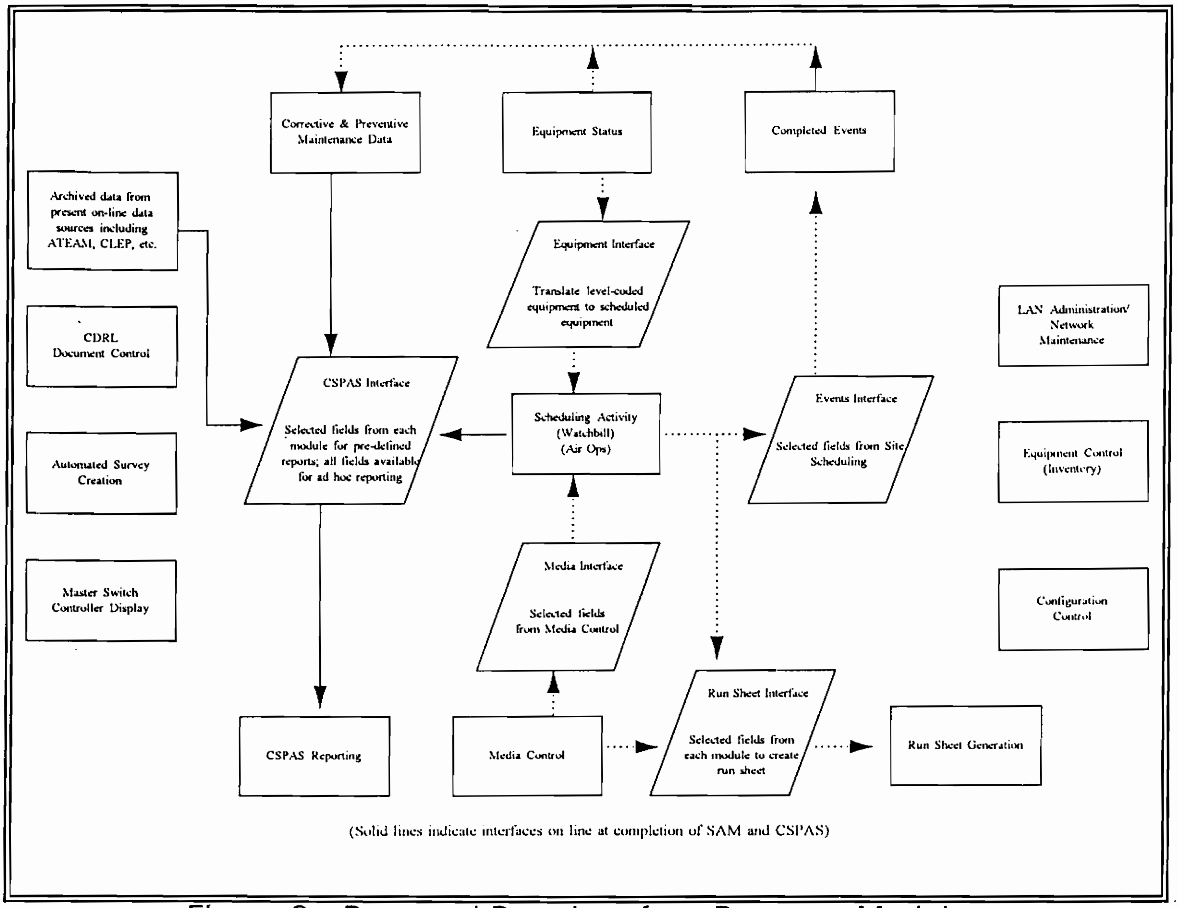


Figure 3. Proposed Data Interface Between Modules

- **Corrective maintenance data.** Corrective maintenance data management at ACSC is accomplished through the implementation of Navy Micro-Shipboard Non-Tactical Automatic Data Processing II System (Micro-SNAP). Data from this program is regularly transmitted via COTS modem lines to the Naval Sea Logistics Center (NAVSEALOGCEN) Maintenance and Materials Management (3-M) database. Methods will be established to periodically obtain corrective maintenance data from the Ship's 3-M Open Architectural Retrieval System (OARS) software which is currently available to enable capturing of this data. Required reporting data from the 3-M database includes: (but is not limited to) equipment identification code, maintenance man hours, active maintenance time, date of maintenance action, and other available information necessary to create standardized reports on corrective maintenance. Some translation processing will be necessary to properly relate fleet equipment data (as identified by equipment identification code) to ACSC site equipments. Other data files currently in existence for maintenance data will be utilized to provide interface translation data. Once retrieved, the data will be stored in the CSMMOSS database to provide access for CSPAS standard reports, as well as for ad hoc reporting.

- **Preventive maintenance data.** A manual system is currently used by the Navy to manage and track preventive maintenance data. Methods will be established to periodically obtain preventive maintenance data required for the CSPAS standard reports. Some translation processing may be necessary to properly translate preventive maintenance data for reporting purposes. Once retrieved, the data will be stored in the CSMMOSS database to provide information for CSPAS standard reports, as well as for ad hoc reporting. A limited amount of historical data will be available for reporting purposes.

3.3.4 System internal interface requirements.

- **Interface identification and diagrams.** Internal interface requirements for CSPAS will be detailed in the CSPAS Functional System Specification.

- **CSMMOSS database interface.** For creation of the CSPAS standard reports (as depicted in Appendix B), interface with the CSMMOSS database must be provided to impart the following reporting data:

  - **Event-related data.** Data such as equipment usage, user satisfaction, dependability, setup times, availability, scheduled event hours, completed event hours, and requested event hours will be obtained from the CSMMOSS database.

  - **Maintenance data.** Corrective and preventive maintenance data will be obtained as described in paragraphs 3.3.3.2 and 3.3.3.3 above. Once obtained, this data will be stored in the CSMMOSS database and will be available for retrieval and reporting.

In addition to standard reports, interface with all data in the CSMMOSS database (including historical data where available) will be provided in order to facilitate ad hoc reporting through a COTS reporting package. In previous versions of this document, this capability to produce ad hoc reports has been described as "x-y chart capability".

3.3.5 System internal data requirements. All information pertaining to specific system internal data requirements will be provided in the Functional Design Specification for CSPAS. In general, data must be provided to enable creation of the reports as specified in Appendix B. In addition, data will be structured, wherever possible, to facilitate the creation of ad hoc reports through an COTS reporting package. In previous versions of this document, this capability to produce ad hoc reports has been described as "x-y chart capability".

3.3.6 Adaptation requirements. No adaptation requirements are anticipated for this system.

3.3.7 Safety requirements. "Safety" refers to system requirements concerned with preventing or minimizing unintended hazards to personnel, property, and the physical environment. No requirements pertaining to this issue are anticipated.

3.3.8 Security and privacy constraints. System security shall be provided via the login security provided by Novell Netware as currently installed on the CSLAN. Further security will be provided via additional user permissions for this specific system. At least three distinct levels of user access (see paragraph 3.3.13 below) will be required for the CSPAS software. Access levels and permissions will be detailed in the CSPAS Functional Software System Specification.

3.3.9 System environment requirements. The CSLAN has four Novell Netware version 3.1X file servers that comprise a part of the AWAN system. An upgrade to Novell Netware version 4.1 is anticipated in the near future. The servers are interconnected to support residents of ACSC buildings V-10, V-20, Q-29 and R-30, as well as the PRC company offices. Network support includes access to a wide variety of commercial and customer software application programs, databases and communications to remote network systems. The following paragraphs provide a general overview of the CSLAN design.

- **Physical environment.** The CSLAN is a local area network consisting of four file servers, numerous workstations, printers and a communications server. CSPAS will be fully compatible with the ACSC network environment as described in the CSMMOSS Operational Concept Description. The system must support multiple-user access and function in a timely and reliable manner. It must provide accurate reports with minimal delay, regardless of LAN utilization. All reports shall be available to both the video display and to LAN printers and plotters (where applicable).

- **File servers include:**

  a. **Wallops Island Mission Support (WIMS).** WIMS hosts the applications specifically developed to support the ACSC mission.

  b. **Wallops Island Office Automation (WIOA).** WIOA hosts the commercial business applications.

  c. **AEGIS Facilities Documentation Management System Two (AFDMS2).** AFDMS2 hosts the ACSC drawing libraries and interfaces with AFDMS at the Naval Surface Warfare Center

(NSWC) Dahlgren, VA.

d. **Wallops Island Snap (WISNAP).** WISNAP hosts the Micro Shipboard Non-tactical Automatic Data Processing (SNAP) II system.

- **Workstations:** In its present configuration, the CSLAN cable layout consists of 125 workstations which can accommodate up to 100 users simultaneously. Typical workstations are comprised of 386/33, 486/50 and Pentium Personal Computers (PCs) with both Video Graphics Array (VGA) and Super VGA monitors Many variations in workstations exist.

- **Print Servers:** At the present time, four print servers are in use at ACSC. These print servers provide a total shared printer capacity of 34 printers, 12 of which are currently installed.

- **Communications Server:** The communications server provides remote user access to the AWAN via three dial-in-modems in addition to supporting five dial-out modems.

- **CSMMOSS Server.** The addition of one server for the CSMMOSS database and programs is anticipated before completion of the CSPAS software. In addition, a total of six workstations are to be added to the CSLAN in support of CSPAS and Scheduling Activity Module (SAM) software.

- **Installed Software.** Both commercial off-the-shelf (COTS) software applications such as word processors, and various programs to support unique ACSC operations are currently installed on the CSLAN. The major software applications that are installed on each file server are:

- **WIOA**. MicroSoft Windows, MicroSoft Office, WordPerfect 5.1, Harvard Graphics 3.0, dBASE IV, Lotus 1-2-3, PCAnywhere, Delrina Formflow, Novell PerfectOffice, Groupwise Mail and CC:Mail.

- **WIMS**. AEGIS Training and Engineering Accessibility Module (ATEAM), CSOOW Logbook and Equipment Program (CLEP), Library Inventory Management System (LIMS), and ACSC Technical Library Automated System (ATLAS).

- **AFDMS2**. AUTOCAD, the AFDMS program, and DocMan/IMCENTER.

- **WISNAP**. SNAP.

3.3.10 Computer resource requirements.

- **Computer hardware requirements.** The recommended workstations for routinely accessing CSPAS should consist of (minimally) an 80486DX-33 processor (CPU), 16 megabytes of random access memory (RAM), super video graphics array (SVGA) monitor and video card with a resolution of 1024x768 pixels and be running Microsoft Windows with network drivers.

- **Computer hardware resource utilization requirements.** The CSPAS software will reside on the CSMMOSS server when installed. Resource utilization for this program will be physical storage space required for this software and any data specific to CSPAS reporting only. In addition, physical storage space will be required for the COTS ad hoc reporting package. Detailed resource utilization requirements will be outlined in the CSPAS Functional Software System Specification.

- **Computer software requirements.** CSPAS will be generally compatible with current versions of standard Microsoft software applications currently installed on the CSLAN, such as Excel (a spreadsheet program), Word (word processing), and Access (database manipulation). "Compatible" refers to the format of data produced by the CSPAS software (for Excel and Access), or the use of graphic data for printing and/or inclusion in documents (for Word).

- **Computer communications requirements.** No communications requirements are anticipated for this system.

3.3.11 System quality factors.

- **Ease of use.** Wherever possible, execution of the standard reports (as defined in Appendix B) shall be accomplished within five menu selections.

- **Report execution time and reliability.** Report execution time shall correspond as closely as possible to the established acceptable processing time (currently two minutes). Reporting results will be correct and consistent.

3.3.12 Design and construction constraints.

- **Development tools.** CSPAS will be developed using the following development tools:

  - PowerBuilder Enterprise 5.0 (or higher version) (PowerSoft, Inc.)
  - PowerSoft FUNCky development library
  - Ad-hoc reporting and graph package (to be determined)
  - Borland C++ (Borland, Inc.)
  - First Impressions graph OCX (Visual Components, Inc.)
  - Windows 3.11 Software Development Kit (Microsoft)

- **Database management system.** CSPAS will utilize a server-based, relational database management system compatible with all of the above-listed development tools.

- **Design standards.** Design standards to be used in the development of this software system should adhere to industry-accepted methods for object-oriented software design and relational database design. Appendix D contains Coding Standards and Naming Configurations.

- **Future growth.** As well as to accommodate changes in ACSC's AWAN/CSLAN technology environment, mission loading, and increased combat systems operational capability; CSPAS must be designed in a manner which permits system modification while minimizing reprogramming effort. The data structure shall also allow for future expansion or modification and interface with advancing commercial applications and changes to the AWAN/CSLAN environment.

3.3.13 Personnel-related requirements. Three distinct levels of user access will be available in CSPAS: analyst, management, and user. The analyst level of access will provide all functions available in CSPAS. The management level of access will show all reports, but with fewer options available. The user level of access will (possibly) restrict the number and/or types of reports which can be created. On-line help will be provided to assist users in the operation of the software.

3.3.14 Training-related requirements. Training will be provided in the use of this software. Methods and plans for training will be addressed in a CSPAS Implementation and User Training Plan and User Guides will be provided.

3.3.15 Logistics-related requirements. CSPAS will be supported and maintained by government and contractor ADP personnel. No other logistics-related requirements are anticipated.

3.3.16 Other requirements. Additional requirements include but are no limited to system documentation, user guides, test plans and procedures, and installation of system on CSLAN.

3.3.17 Packaging requirements. The CSPAS program and all modifications or updates will be delivered via high-density (3.5 inch 1.4 MB) diskettes for program installation. A copy of the source code shall be delivered to Code 6900, the AIS Security Manager.

All diskettes and the source code listings will be labeled with the program name, version identification, and date. "SENSITIVE UNCLASSIFIED" classification markings (NAVMC 11196(1-88) SN:0000-00-007-0010 11/1) shall be used on all diskettes. The diskette label format is shown in Appendix E.

3.3.18 Precedence and criticality of requirements. All requirements in this specification are deemed to be of equal weight.

3.4 Description of the new or modified system

The proposed Combat Systems Performance Assessment System (CSPAS) subsystem of the CSMMOSS will be a Microsoft Windows-based, fully integrated computer program. Users will be able to select reports and options using a mouse for an intuitive, point-and-click interface.

CSPAS will extract data directly from the CSMMOSS database (currently under development) which will provide the basis for a number of statistical reports currently in use at ACSC or to be developed specifically for this program. Data to be extracted will include event-related and scheduling data (from the Scheduling Activity Module currently under development), completed events data (from the Completed Events Module to be developed as part of the CSMMOSS development effort), and some preventive and corrective maintenance data (from the Navy's MicroSnap and PMS Sked software currently in use). The CSPAS software will enable users to quickly and easily produce pre-defined reports on several areas of mission evaluation. Figure 4. is a depiction of the Proposed Combat Systems Performance Assessment System (CSPAS) Process.

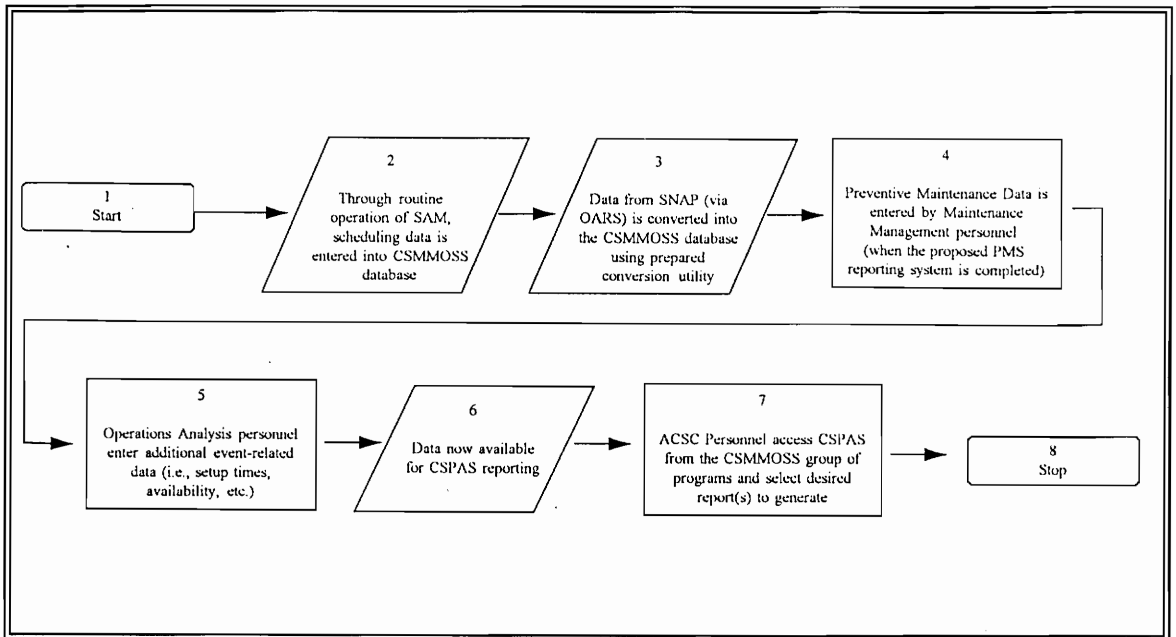


Figure 4. Proposed Combat Systems Performance Assessment System (CSPAS) Process.

In addition to the pre-defined reports to be specified for this software, the ability to perform ad hoc reporting of the data will be provided through the use of a commercial off the shelf (COTS) reporting package. Additional capabilities will be obtained through the integration of CSPAS data and reports with other software applications currently in use on the CSLAN, such as Microsoft Excel (spreadsheet) and MS Word (word processor).

3.5 Proposed CSPAS Operational Scenario

3.5.1 Generation of pre-defined reports. The CSPAS program will be available as a menu selection on the CSMMOSS main application window. The program user will select the desired report from the menu and will be presented with the requested report (usually in a graphical representation) with a default supplied for each report option. If all defaults meet the criteria of the user, the report will be generated by simply pressing on a window button or icon. Any changes to the default selections can be made in a similar manner. After report generation, the user will choose to print the report, save the report data in a variety of formats, copy the report to the Windows clipboard area for use in another application (such as Microsoft Word), change some characteristics of the report graph, or modify selection criteria and produce another report. For selected reports, the user will have the additional ability to "drill-down" on the report data, i.e., by clicking directly on a data bar in a graph, the user will be able to see the underlying data directly from the CSMMOSS database. This data can also be printed or saved to a file.

3.5.2 Generation of ad hoc reports. Using a COTS ad hoc reporting package (currently undetermined), the user will have the ability to generate, print, export and save reports on any data from the CSMMOSS database.

3.5.3 Acquisition of preventive and corrective maintenance data. Periodic retrieval of preventive and corrective maintenance data will be required to support the operation of CSPAS. Currently, a Navy-supplied data retrieval software system known as the Open Architecture Retrieval System (OARS) is available to retrieve a variety of corrective maintenance from the Naval Sea Logistics Center (NAVSEALOGCEN). Preventive maintenance data is also available from the PMS Sked software (also Navy-supplied). Electronic preventive maintenance data collection is a relatively recent development at ACSC. Little historical data for this activity will initially be available for reporting. A facility to store this maintenance data in the CSMMOSS database will be provided in CSPAS.

3.6 Analysis of Proposed System

3.6.1 Summary of advantages. The following advantages are expected to be obtained from the use of CSPAS as proposed: (1) more reliable and efficient preparation of reports which are desired for routine use and/or distribution at ACSC; (2) the availability of virtually all event-related, scheduling, and maintenance data stored in the centralized CSMMOSS database for automated reporting through the use of the ad hoc reporting package; (3) the potential for the improvement of services provided to mission support users at ACSC.

3.6.2 Summary of disadvantages/limitations. The most apparent limitation of this module will be its dependence on user participation in order to operate correctly. Users will be required, particularly at the initial installation of this computer program, to adhere to newly established procedures very carefully. Failure to do so will result in impaired performance of the system. These disadvantages/limitations will be minimized through the use of a CSPAS user guide, training sessions, and an on-line tutorial which is planned as the first major modification to CSPAS.

3.7 Qualification Provisions

The CSPAS computer program shall be verified to ensure compliance with the written requirements of this specification. A Test and Acceptance Plan will be developed to define the scope of testing for the CSPAS software program. The test plan will designate the qualification method to be used for verification of each functional requirement. Preliminary tests will be conducted within the software engineering environment. Software test
descriptions will identify the input data, expected output data, and the evaluation criteria of all formal tests. A Government representative will observe all testing and demonstrations. If required, instrumentation or special test equipment will be specified in the test plan.

3.7.1 General qualification methods. The following methods of qualification will be used in evaluating acceptance of the CSPAS software operation. For each acceptance item, the method of qualification to be used will be detailed in the CSPAS Test and Acceptance Plan.

- **Demonstration.** The operation of the system, or a part of the system, that will rely on observable functional operation not requiring the use of instrumentation, special test equipment, or subsequent analysis.

- **Test.** The test method of qualification is carried out by operation of the software in a LAN environment and requires the collection and subsequent examination of data. Instrumentation or special test equipment may be required.

- **Procedure.** Procedures for system use will be available as user documentation and will be demonstrated.

- **Analysis.** The analysis method of qualification is accomplished by machine processing of accumulated data and comparing to hand processing of the same data.

- **Inspection.** Inspection is the visual examination of system components, documentation, etc. In general, this method of qualification will be used for any user manuals, administrator manuals, or associated system documentation.

4.0 DETAILED SOFTWARE DEVELOPMENT ACTIVITIES

4.1 Project Planning and Oversight

4.1.1 Software development Process. The software development process includes the following major activities that are described in the text which follows: (a) project planning and oversight; (b) establishment of a software development environment; (c) system requirements analysis and documentation; (d) system design; (e) software requirements analysis; (f) software design; (g) configuration management; (h) quality assurance; (i) corrective action; (j) creation of programmers' notes; (k) technical and management reviews; (l) computer software unit (CSU) integration; (m) CSU testing; (n) computer software configuration item (CSCI) integration; (o) CSCI testing; (p) system acceptance testing; (q) preparations for software use and transition; (r) training and implementation; (s) creation of User Training Guides; (t) completion of the Final Technical Report; and (u) Maintenance and Operations. This process can be depicted using a waterfall model as shown in Figure 5 below. Arrows pointing to the right indicate progression to the next level whereas arrows pointing up are feedback loops.

```
REQUIREMENTS ->
 ^---  DEFINITION ->
        ^--- DESIGN ->
              ^---  CODING ->
                     ^---  IMPLEMENTATION ->
                            ^--- SYSTEM TESTING ->
                                  ^---  ACCEPTANCE TESTING ->
                                                     DELIVERY
```

Figure 5. Waterfall model.

Specific products which will be required of the development team are as follows:

- **Programming Language Trade-Off Analysis.**
- **Advanced Summary Level Description.**

- **Technical Software Development Plan.**
- **CSPAS Project Schedule.**
- **Operational Concept Description.**
- **CSPAS Functional Software System Specification.**
- **CSPAS System Software Specification.**
- **Programmer's Technical Support Document.**
- **CSPAS User Guides.**
- **Introduction of CSPAS onto LAN.**
- **Implementation (Introduction) and User Training Plan.**
- **Monthly In-Process Briefs.**
- **Mid-Task Review Brief.**
- **Test and Acceptance Plans.**
- **Final Technical Report.**
- **LAN upgrade materials/equipments Listings.**
- **CSPAS Quality Assurance Report.**
- **Final Review Brief.**
- **CSPAS Software Installation on the CSLAN.**

Descriptions of each of these products can be found in Appendix F - Analysis of Technical and Support Documentation Requirements

4.1.2 Software development methods. The development of software can be grouped into four distinct areas. The first area, the preparations for software development, included items (a) and (b) of paragraph 4.1.1 above. The second area, the performance of software development, was a direct application of items (c) through (f). The third area, maintaining quality of software development, includes items (g) through (p). The final area, preparing for CSPAS delivery, included items (q) through (t). Figure 6 depicts the basic CSPAS software development activities in a broad sense.

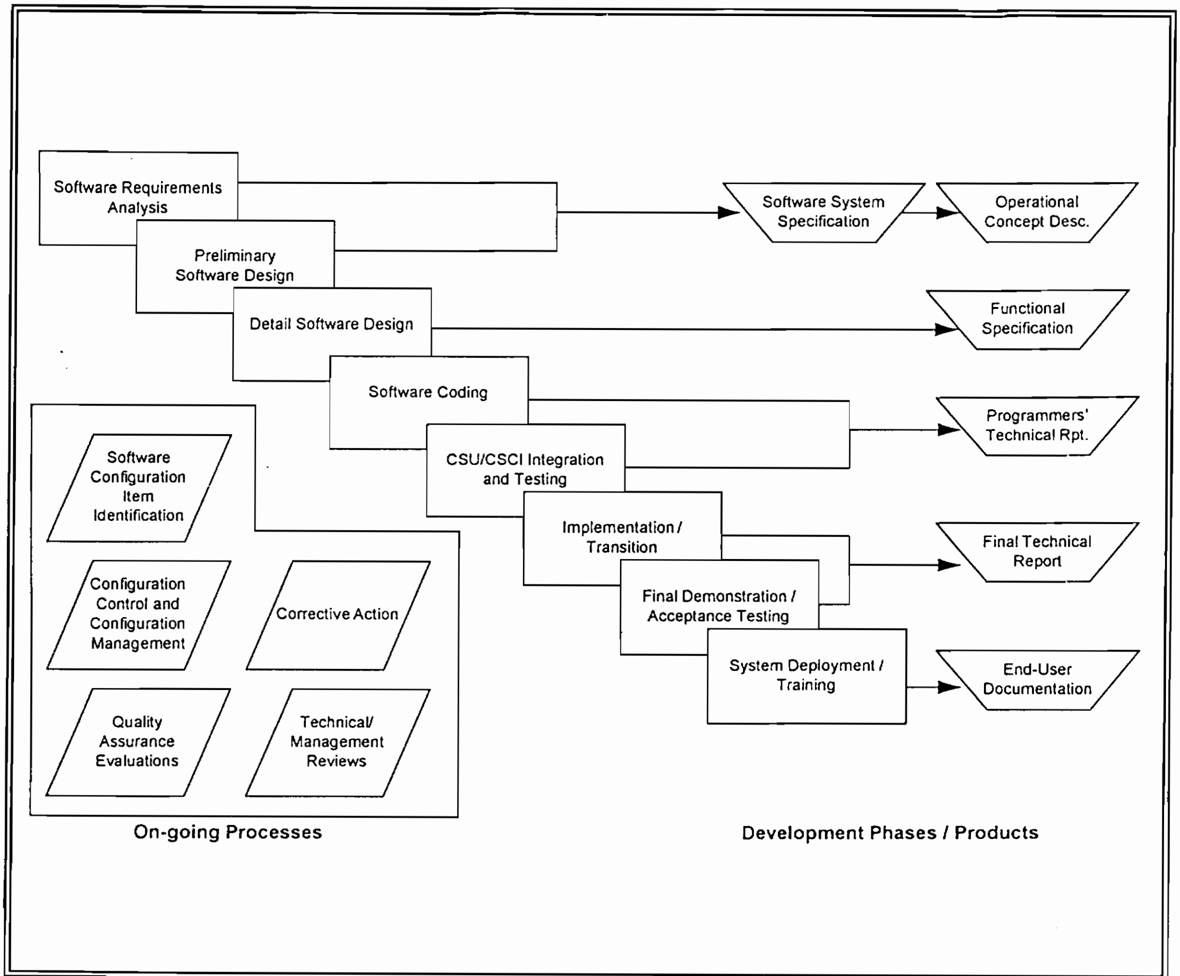


Figure 6. Software Development Activities

4.1.3 Software development planning. As modifications to the Preliminary Software Development Plan (SDP) are deemed necessary, the document will be updated accordingly. The Final SDP will include modifications and corrective actions as they were required.

4.1.4 CSU/CSCI test planning. As stated in paragraph 4.1.1 above, a CSCI (computer software configuration item) is defined as a specific functional unit of software. CSCIs are comprised of CSUs (computer software units), smaller subsets of code which are combined to meet the functional requirements of a CSCI. All CSUs included in a CSCI were tested. Tests will be conducted internally by the programmers during program development, using the form and procedures as outlined in **Appendix G, CSU Code Inspection and Test Procedures.** Upon completion of all CSU testing for a specific CSCI, testing of the CSCIs will be conducted internally using the form and procedures as outlined in **Appendix H, CSCI Code Inspection and Test Procedures.** Results of all tests will be reviewed for correctness by the Systems Designer, evaluated for quality assurance by the Configuration Manager, and placed under configuration management in the Archives Library (**see paragraph 4.2.3**). Coding test result evaluations and coding test results will be described in a CSPAS Final Technical Report.

4.1.5 System test planning. The purpose of the system testing phase is to verify the ability of the system to satisfy all requirements as they are specified. System testing will consist of two formal stages, internal testing and acceptance testing. System testing will also be conducted informally by all development personnel throughout software development.

During the internal test phase, testing of the overall system will be performed within the development environment. Internal system tests will be conducted according to the instructions as outlined in Appendix G, CSU Code Inspection and Test Procedure and Appendix H, CSCI Code Inspection and Test Procedures. Results will be reported for each tested item. Corrective action will be taken where necessary, and any corrected code will be re-integrated and re-tested (including regression testing). Results of the internal system tests will be reviewed for correctness by the Sr. Systems Designer, evaluated for quality assurance, and placed under configuration management in the Archives Library (**see paragraph 4.2.3**).

The acceptance testing phase will consist of testing identified software requirements as delineated in the CSPAS Software System Specification. Testing will proceed according to a Test and Acceptance Plan. The test will be accomplished by the Sr. Systems Engineer with a member of the ACSC government personnel and the Configuration Manager in attendance. Acceptance testing results will be included in the CSPAS Final Technical Report.

4.1.6 Software installation planning. Completed software will be installed in the planned operating environment (the CSLAN) by development personnel, and conducted with proper notification to ACSC personnel and minimal disruption of CSLAN operations.

4.1.7 Software transition (implementation) planning. Procedures regarding training and implementation of the software will be defined in the User Training and Implementation Plan. The plan will include information regarding transition from the existing system to the newly installed system. User Training Guides will be produced as easy reference tools for creating reports using CSPAS. CSPAS training sessions will be conducted.

4.1.8 Management reviews. Reviews will be conducted to provide regularly scheduled monitoring of the project. Although three different types of reviews will be required, there are basically four questions which should be addressed at these reviews. The questions are:

- Is the development plan being followed?
- Is the project team making satisfactory progress?
- Are there any indications of future problems?
- Is the team prepared to move on to the next phase?

The three different types of reviews are:

- **Preliminary Design Review (PDR)**. After the completion of the functional design phase of CSPAS development, a Preliminary Design Review will be held to present the preliminary design during a monthly In Process Briefing. Topics discussed will include: (1) resource utilization, (2) project scheduling, (3) traceability of requirements to functional design, (4) alternative design approaches considered, (5) any technical risks, and (6) interface descriptions. Results of this review will be recorded in the IPB Minutes. The results will be evaluated for Quality Assurance, placed under configuration management, filed in the Archives Library (**see paragraph 4.2.3**), and included in the CSPAS Final Technical Report.

- **Critical Design Review (CDR)**. After the detailed design of the CSPAS Module is complete, and before implementation has begun, a Critical Design Review will be held to present the detailed design. Results of this review will be included in the IPB Minutes. The results will be evaluated for Quality Assurance, placed under configuration management, filed in the Archives Library (**see paragraph 4.2.3**), and included in the CSPAS Final Technical Report.

- **Operational Readiness Review (ORR).** After acceptance testing is completed, remaining critical issues and questions will be discussed at the Operational Readiness Review. Key issues for this discussion include: (1) status of remaining technical documentation, (2) configuration control items, (3) transition and implementation plans and (4) and User Guides and training. Results of this review will be included in the meeting Minutes, evaluated for Quality Assurance, placed under configuration management, filed in the

Archives Library (**see paragraph 4.2.3**), and included in the CSPAS Final Technical Report.

4.2 Established software development environment.

4.2.1 Software engineering environment. A software engineering environment will be established for the development of CSPAS. The engineering environment will provide the software development team with the tools required to accurately code and test all CSUs. Tools used in the software engineering environment include application languages (PowerBuilder 5.0), the relational database management system (Oracle 7.2) graphic and icon editors (PowerBuilder, Microsoft Paintbrush), word processors and text editors (WordPerfect, Microsoft Word, PowerBuilder text editor, RoboHelp), a spreadsheet program (Microsoft Excel), and an ad hoc report generator (Impromptu 3.0). A proposed Software Design Team is shown below in Figure 7.

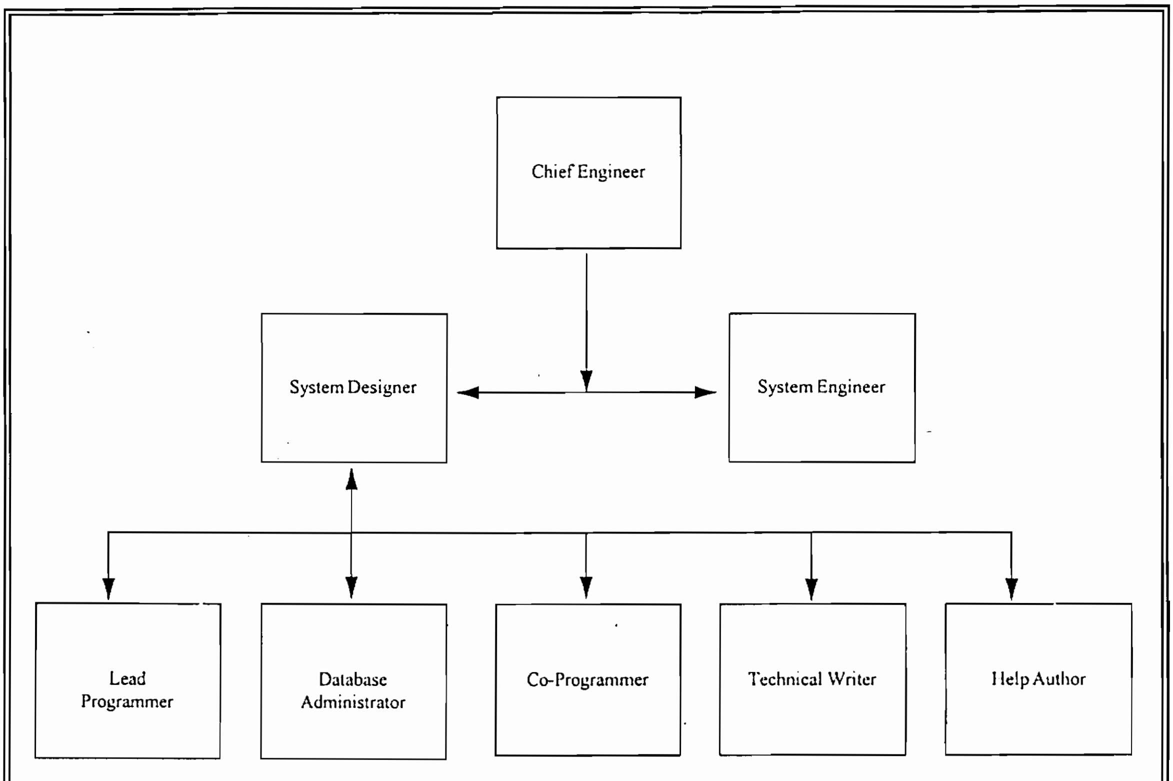


Figure 7. Software Design Team

4.2.2 Software test environment. Testing of software will occur at both government and contractor facilities.

4.2.3 Software development library. As part of configuration management and quality assurance, a software development library will be developed and maintained within the software development environment. The library will contain three separate libraries: the Archives Library, the Application Library, and the Configuration Control Library, described below:

- **Archives Library.** This library will contain programmers' notes as compiled throughout software development. It will also contain all other technical and non-technical documentation including (but not limited to) Quality Assurance Evaluation Reports, management review reports, development plans and results, test plans and results, User Training Guides, and other deliverables. The Archives Library will be managed by the Sr. Technical Writer.

- **Application Library.** This library will be the location for CSUs and CSCIs under development prior to their completion and placement in the Configuration Control Library. The Application Library will be managed by the Sr. Systems Designer.

- **Configuration Control Library.** This library will be the location for source code listings. It will define the evolving configuration of the CSCIs until their evaluation and filing. Once filed, they cannot be modified without following the formal procedures for corrective action as described in **section 4.17, Corrective Action**. The Configuration Control Library will be managed by the Sr. Systems Designer.

4.2.4 Software development files. Software development files maintained in the Configuration Control Library will include:

- **Source Code Files.** Source code listings in programmer-readable format.
- **Data Dictionaries.** Dictionaries listing file names, field names, field descriptions, field characteristics, and data sources for each item in the CSPAS main database.
- **Executable Files.** Those output files which are produced by the assembly, compilation, or linkage process as completed by the software development personnel.

4.3 System requirements analysis.

4.3.1 Analysis of user input. User input to the requirements of this system consisted of an existing preliminary software system specification dated 30 September 1995, and ongoing meetings and interviews with appropriate ACSC personnel.

4.3.2 Operational concept. The Operational Concept Description (OCD) for this system will address the following areas:

- What is the current process?
- Description of proposed scenario.
- What are the reasons for changing the current process?
- Description of the operating process with supporting flow charts.

4.3.3 System requirements. Preliminary requirements for this system were specified in the CSPAS Preliminary System Software Specification dated 30

September 1995. As these requirements were modified, changes will be reflected in the Final Software System Specification.

4.4 System design. System-wide design decisions will be addressed in the final CSPAS Software System Specification. System architectural design decisions will be addressed in the Functional Software System Specification.

4.5 Software requirements. CSPAS software requirements will be fully defined in the Functional Software System Specification. CSCIs (as defined in **paragraph 4.1.2, Software development methods**) will be identified and related to the requirements outlined in that document.

4.6 Software design. Software design will be described in the Final Functional Software System Specification. The detailed functionality of each CSCI is included.

4.7 Software implementation and testing. Software implementation and testing will be addressed in the User Training and Implementation Plan, and in the Test and Acceptance Plan, respectively.

4.8 Unit integration and testing. CSU testing will be conducted internally by the system programmers and the Sr. Systems Designer during CSPAS project development. The form and procedures which will be used are as outlined in **Appendix G, CSU Code Inspection and Test Procedures.** Upon completion of all CSU testing, CSCI testing will be conducted internally using the form and procedures as outlined in **Appendix H, CSCI Code Inspection and Test Procedures.** Results of those tests will be reviewed for correctness by the Sr. Systems Designer, evaluated for quality assurance by the Configuration Manager, and placed under configuration management. Testing is described in **paragraph 4.1.4 CSU/CSCI test planning.**

Test results will be included in the CSPAS Final Technical Report.

4.9 CSCI qualification (acceptance) testing. This section is intended to address acceptance testing of individual CSCIs. Qualification testing will be described in the CSPAS Software Test and Acceptance Plan and in **paragraph 4.1.5 System test planning** of this document. Test results will be described in the CSPAS Final Technical Report.

4.10 CSCI/HWCI integration and testing. A hardware configuration item (HWCI) is defined as an aggregation of hardware that satisfies an end-use function and is designated for separate configuration management by the acquirer. No HWCIs are included in this system implementation.

4.11 System qualification (acceptance) testing. Procedures for acceptance testing of the system are as outlined in **paragraph 4.1.5** and **section 4.9.**

4.12 Preparation for software use.

4.12.1 Prepared executable software. Prior to software delivery, development personnel will coordinate and compile (according to the constraints of the development language) source code modules and resources (i.e., bitmaps, file listings) into an executable software product.

4.12.2 Prepared version descriptions for the user site. All software code modules will be assigned a version control number. This number will be used to identify each module while within the Configuration Control Library, and will continue to be used after transition to the operating site.

4.12.3 Prepared user manuals. User Training Guides will be written according to a government-approved format. One will describe applications and capabilities of the software with examples of CSPAS reports. A double-sided Quick Reference Guide will describe the CSPAS reporting process in brief simple steps.

4.12.4 Installation at user site. Procedures for installation of developed software at ACSC will be included in the User Training and Implementation Plan.

4.13 Preparation for software transition. The prepared executable application will be placed on magnetic media (high density 3.5 inch diskettes (1.4 mb) or CD-ROM) for delivery. The diskettes and source files will be labeled with the program name, version identification, and the date. Source code listings and Programmers' Notes will be supplied in the CSPAS Final Technical Report. The updated fres.

4.14 Configuration management. Configuration management is defined as the formal management of configuration items which have been placed under configuration control. Configuration management will be accomplished through the unique identification of configuration items, through the establishment of a Change Control Board (CCB) (**see paragraph 4.14.2.2, Change Control Board (CCB),** the establishment and maintenance of the Archives Library and Configuration Control Library, and with routinely held quality assurance evaluations (**see paragraph 4.16.2 Quality assurance records**).

4.14.1 Configuration identification. A configuration item is defined as any deliverable item which is subject to acceptance by the acquirer. Examples of configuration items include technical documentation, CSCIs and their component

CSUs, deliverable electronic media, files, and bitmaps. Once identified for this project, each item will be assigned a number as in the example contained in Table 2 below:

Table 2. CSCI Identification

CSCI	CSU	Version	Revision	Release Status	Item Type
1-n	1-n	1-n	1-n	**d** = under development **c** = complete **r** = released	**doc** = document **csu** = computer software unit **bmp** = bitmap **res** = resource file **med** = electronic media **dbf** = database file (table) **rec** = database record

The format of the identifier will be as follows:

<CSCI>.<CSU>.<Version.>.< Revision>.<Release Status>.<Item Type>

4.14.2 Configuration control. Configuration control consists of the following two areas:

- **Configuration control process**. Once a configuration item is determined, it will be placed under configuration control. This means that any changes or modifications after that time will require approval by the Change Control Board (CCB). Proposed changes to any configuration item can be initiated via an ACSC Non-Tactical Computer Program Change Request 5231/3 (1/95) (CPCR), an example of which is included in Appendix I. The change control process is as follows:

- User has problem or requests change to the application
- User enters problem into MRQ log
- LAN Administrator reviews and processes MRQ
- Program problems or change requests forwarded to module TPOC
- Module TPOC prepares CPCR
- CCB reviews and approves CPCR
- CCB forwards CPCR to support contractor
- Support contractor completes work and installs change
- Installation verified by LAN Administrator and DBA
- Change accepted by module TPOC
- Support contractor sends completed documentation to CCB

After a CPCR has been submitted, it will be reviewed by the CCB. If approved, the change will be implemented by software development personnel and documented accordingly.

- **Change Control Board (CCB)**. The preliminary CCB will be comprised of the Chief Engineer, Sr. Systems Designer, Sr. Systems Engineer, Configuration Manager, the government POC for CSPAS, and the government Automated Information System (AIS) Security Manager. The CCB will be organized to review Computer Program Change Requests (CPCRs) and problem reports as required, prioritize solution implementation, and assign responsibility for problem resolution and revision of documentation if required.

4.14.3 Configuration status accounting. The software development personnel and technical staff will maintain records of the configuration status of all items that have been placed under configuration control. These records will be maintained in the appropriate library until project completion. They will include, as applicable, the current version, revision, or release of each item and a record of changes to the item since placed under configuration control.

4.14.4 Packaging, storage, handling and delivery. ACSC has established procedures for packaging, storage, handling and delivery of all deliverable items. The ACSC AIS Manager will maintain master copies of delivered software products and other deliverables. Floppy disks and source code listings will be labeled with the program name, identification number and date. "SENSITIVE UNCLASSIFIED" classification markings will be used on all disks.

4.15 Software product evaluation. This paragraph refers to the internal testing of software under development prior to acceptance testing and delivery which is addressed in **section 4.1.5 System test planning.**

4.16 Software system quality assurance.

4.16.1 Quality assurance (QA) plan. The objective of a QA plan is to assure the quality of the deliverable computer program and associated documentation. A plan will be written for the Preliminary Software Development Plan and maintained throughout the course of software development by retrieving and compiling Quality Assurance Evaluation Reports from Quality Assurance personnel on all quality assurance evaluation categories.

4.16.2 Quality assurance records. The quality assurance evaluator will prepare records of each required quality assurance activity as defined by the Sr. Technical Writer. This will be accomplished by the timely completion of Quality Assurance Evaluation Reports (QAERs). **Appendix J. Quality Assurance Evaluation Plan** contains an example of this report form, and instructions for their completion for each activity. All completed QAERs will be archived with a file number in the Archives Library and included in the CSPAS Final Technical Report.

4.16.3 Independence in software quality assurance. Organizational independence will be established between government and contractor's two unique development facilities, resources, and personnel. Software quality assurance evaluations will be performed by Quality Assurance personnel outside of the immediate software development group.

4.17 Corrective action.

4.17.1 Problems/change reports. The format for CPCRs will be shown in the CSPAS Final Technical Report.

4.17.2 Corrective action system. Changes, corrections, and modifications can be initiated via the CPCR form as approved by the CCB. Implementation may be initiated on the CPCRs as well. The Sr. Technical Writer will maintain a log of approved changes in the Archives Library subject to Quality Assurance Evaluation Reports if they occur.

4.18 Technical reviews.

4.18.1 Monthly in-process briefs. During the CSPAS Module development, in-process briefs will be held. As required, locations and times will be organized and scheduled by the contractor. Minutes of all briefs will be the responsibility of the contractor's Sr. Technical Writer acting as Archives Librarian.

Meeting format:

Presenters: Government TPOC, Sr. Systems Designer and Chief Engineer

Participants: Representatives from ACSC Code 2000 and Technical Point of Contact

Agenda: Action items from previous meeting.

Status of CSPAS development

Problems/solutions as needed

Action items for next meeting

4.18.2 Mid-task review brief. This meeting will be held as a formal review midway through the project. The status of the CSPAS development effort and software specifications are presented by the Sr. Systems Designer and Sr. Systems Engineer. The Sr. Systems Designer will analyze the preliminary software system specifications and determine through updated quality assurance and internal testing that the software requirements are consistent and complete. The review will include a demonstration on a reduced LAN configuration.

Meeting Format:

Presenters: Government TPOC, Sr. Systems Designer, Sr. Systems Engineer and Chief Engineer (The Database Administrator, computer programmers and Sr. Technical Writer will be available to answer questions).

Participants: Representatives from ACSC Code 2000 and
Technical Point of Contact

Agenda: Discussion about deliverables already submitted

Timeline for balance of deliverables

Problems encountered/solutions

LAN demonstration of completed software

Focus will be placed on the following:

- Satisfaction with the progress of the project
- Indications of future problems
- Comparison of original software development plan with development course
- Plans for next phases of development course
- LAN demonstration

4.19 Other development activities.

4.19.1 Final technical report. The CSPAS Final Technical Report will summarize task milestones, testing results, programmers' notes and technical rationale. Summaries of all in-process briefs and reviews will be included. Quality assurance evaluations will be included. Technical support documentation, user training and implementation, and User Training Guides will be discussed. The final CSPAS Schedule is described below in **section 5.0 Proposed Software System Development Schedule.**

4.19.2 Subcontractor management. Subcontractors will not be required for completion of this project.

5.0 PROPOSED SOFTWARE SYSTEM DEVELOPMENT SCHEDULE

A proposed software system development schedule is presented below in Table 3:

Table 3. Software System Development Schedule

Activity Name	Start Date	Completion Date
Software Requirements Analysis	28 February 1997	5 March 1998
Preliminary System Design	28 February 1997	15 April 1998
Software Configuration Item Identification	16 April 1998	30 September 1998
Detail Software Design	16 April 1998	6 June 1998
Software Coding	7 June 1998	30 September 1998
Software Testing	1 July 1998	1 July 1998
Final Demonstration and Testing	9 September 1998	9 October 1998
Implementation/Transition	16 September 1998	30 September 1998
System Deployment/Training	30 September 1998	30 October 1998
Configuration Control and Configuration Management	As configuration items are completed and/or CPCRs are required	
Quality Assurance Evaluations	Ongoing after 4 March 1998	

6.0 CONCLUSIONS

6.1 Operational impacts.

Initially, the most noticeable operational impact to the CSPAS process will be a temporary setup period during which user priorities will be established and entered, and standard user equipment configurations will be created. Once it becomes a fully automated system, the elimination of the manual steps currently required to standardize performance assessment reporting may impact the personnel man-hours currently required. Impact to the CSLAN will be minimal, although some file transfer overhead for use in data extraction and manipulation is to be expected.

6.2 Organizational impacts.

No negative impact on the ACSC organization is anticipated. However, several positive impacts are anticipated.

6.3 Impacts during development.

The following impacts on the government during development and at initial installation are anticipated: (1) discussions with development personnel regarding the concept and operation of the new system and (2) parallel operation of the current and new systems for a period of time.

6.4 Concluding remarks.

There is a wide variety of support equipment installed at the ACSC that is available to users of the Site. Numerous simulation systems support system engineering, training, and replace tactical systems supporting other AEGIS baselines. The Operations Control Center (OCC) serves as the central control point or problem control center to support testing and training events. Switching systems are available so that a single equipment group can be scheduled and utilized by the various baselines. The AEGIS Data Reduction system, ADAR, allows users to verify and analyze test results. An audio visual system is available to allow for real-time viewing and monitoring of operator actions and equipment responses as well as providing a historical record for later analysis.

With the increasing number of AEGIS ships comprising a primary force of the surface combatant Navy, operations at ACSC have become vital to operational readiness world wide. Through direct systems level interface, sailors and systems alike can experience a stressed, realistic combat environment with on the spot technical analysis and feedback. Ship life engineering support ashore is essential for investigating emergent shipboard problems, equipment alterations, tactical computer program modifications, and new technology.

The total Combat Systems Capability, including support systems, must be maintained at a high state of readiness to provide maximum and quality system time to multiple site users. Alternate support strategies are held in high value in such an operating environment and must be relied upon when key elements or equipment are degraded or down for repair/activation. Appendix K contains a list of Referenced documents used in this project. Appendix L contains the acronyms found throughout this report.

Appendix A - Measurement Definitions

The definitions in this Appendix are provided for background information only. They describe methods of calculation and/or measurement for data items which will reside in the CSPAS database, but which will be calculated outside of the CSPAS program.

A.1 ACSC User Support Process

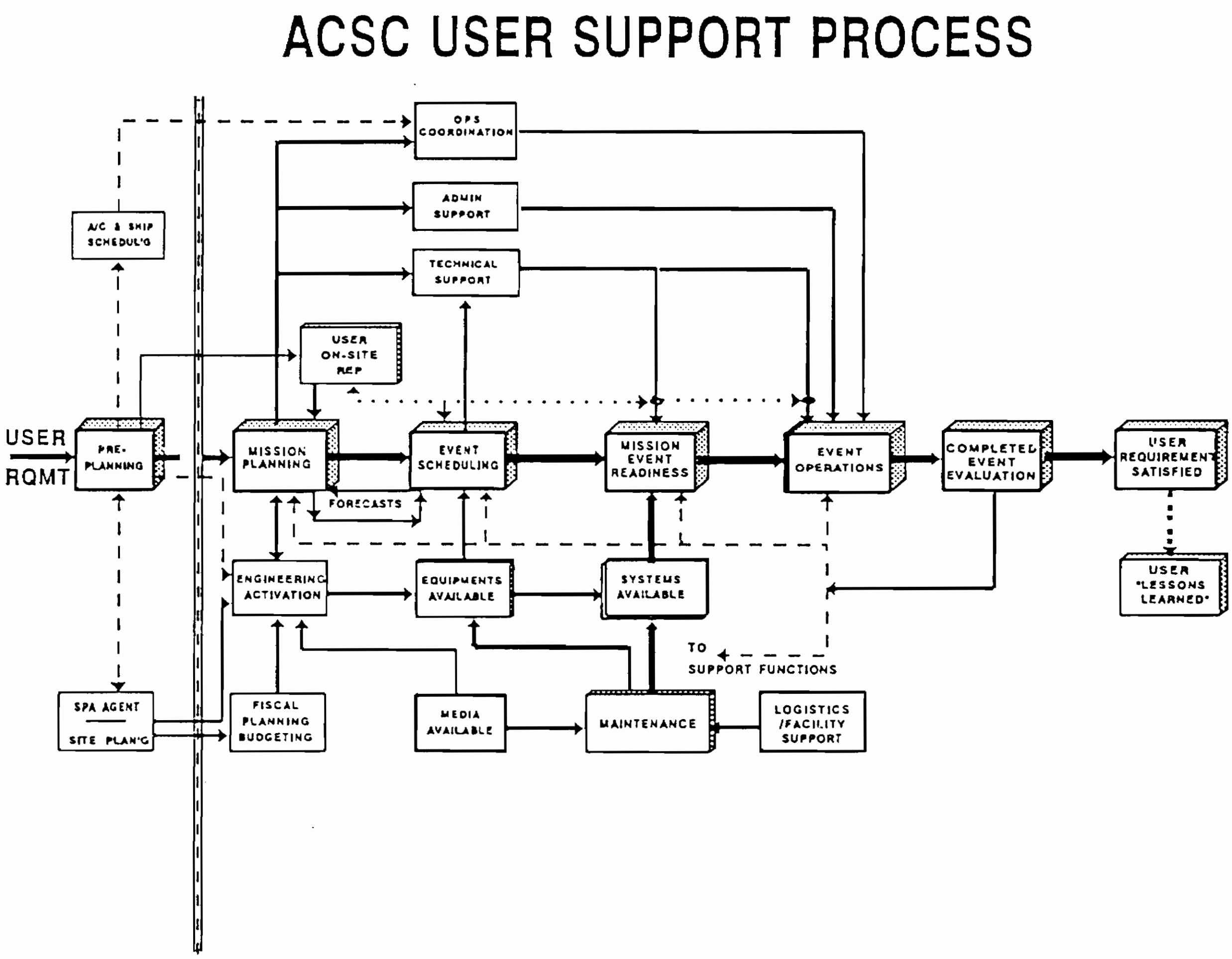


Figure A-1. ACSC User Support Process

Figure A-1 provides an overview of the total ACSC combat system support process as typically experienced by our customers. Supporting system processes are indicated both above and below the activities found in the center of the chart. Assessment measures will be applied to establish trends of ACSC support performance in mission support activities.

A.2 Mission Planning

Mission planning is the advance planning and coordination of activities by site and user personnel necessary to conduct a successful mission event at ACSC. These activities and events include, but are not limited to, forecasting, securing schedule commitment, obtaining outside assets, configuration adjustments and security arrangements. Representatives from each ACSC department, user organization, contractor or other activities may be called upon to participate in these planning activities.

A.3 Scheduling

A.3.1 Event Planning. Event planning is the advance request and commitment of time, equipment, people and resources to future events. This usually requires coordination among users to free up time for that event.

A.3.2 Event Scheduling. Event scheduling is the allocation and commitment of scheduled time, equipment, systems, personnel and other resources requested by the user in accordance with command priority direction and constraints. It is accomplished by providing the customer with the necessary personnel, equipment and system interfaces to successfully conduct a requested event.

A.4 Event Setup

Event setup is necessary to make the system available and consists of the following functions.

1. Take down from the prior event - terminate recording; stop the computers; remove all program media and data recording media from the drives.

2. Switch setup - run and verify the switch configurations required by the next user according to a user provided run sheet.

3. Load the computers - mount the program media and load all scheduled computers following a user provided run sheet.

4. System check out - to verify all programs loaded correctly and interfaces have been established as requested.

The time required to accomplish setup will vary with the complexity of the user requirements and should not be considered the same for all users.

A.5 Mission Event Readiness

Mission event readiness is defined as ACSC's ability to provide a complete system, ready to support a given user event. Mission event readiness is measured by the availability (AV) of the elements, interfaces, systems, personnel and computer programs required to complete the scheduled mission activity. AV will be decided for each user event using scheduled elements, support equipment, operators and technicians for that event. An element is a subsystem of the ACS as identified by its level code AXX-00-00 (A = baseline reference, XX = element number).

At ACSC, AV is defined as follows:

1. All scheduled combat system elements and support systems are energized and in a ready condition at the end of setup time.

2. All switch configurations run, and links have been verified according to user provided run sheet.

3. All requested and approved operators and technicians are on station at the designated time.

4. All requested and approved site-provided computer programs are loaded according to the user-provided run sheet and are ready for use at the start of the user's event.

The calculation of AV is as follows:

$$AV = \frac{\frac{(E_S - E_N)}{E_S} * 100}{2.222222} + \frac{\frac{(S_S - S_N)}{S_S} * 100}{20} + \frac{\frac{(O_S - O_N)}{O_S} * 100}{4} + \frac{\frac{(P_S - P_N)}{P_S} * 100}{4}$$

AV	Availability (in percent)
E_S	Elements scheduled
S_S	Switch configurations scheduled
O_S	Operators scheduled
P_S	Programs scheduled
E_N	Elements not available at end of setup
S_N	Switch configurations not available at end of setup
O_N	Operators not available at end of setup
P_N	Programs not available at end of setup

Note: Any casualties occurring after the start of an event will not be considered in determining AV, but will be evaluated as part of the event operation.

A.6 Event Operations:

Event operations denotes the conduct of the mission or event by the user. Mission execution begins when the mission event ready phase is complete and the system is turned over to the user. The event is completed when the user's magnetic media is removed from the drives, which may occur before, at, or after the end of the scheduled time. In event operations the user conducts his mission, operating all scheduled equipment according to its design, assisted by ACSC staff personnel if requested. Some of the factors affecting event operations include, but are not limited to, dependability, personnel support and user preparedness.

Dependability (D) is defined as: A measure of the degree to which a system is operable and capable of performing its required function during a specified mission profile, given its availability at the start of the mission.

The calculation of D is as follows:

$$D=\frac{(ST-DT)}{ST}$$

D	Dependability
ST	Scheduled time
DT	Down time (Lost time)

Down time begins when an element is reported as inoperative to the CSOOW or TD, and ends when the system is restored to its initial functional capability or at the end of event time. Dependability will be decided by element, not specific equipment within an element. Problems caused by user provided computer programs or equipment will not be used to decide dependability.

Specific event dependability is referred to as the event's Sustainability (SU) of the required configuration. SU is calculated as a percentage of the usable duration of the event, modified by the total number of elements scheduled.

The calculation of SU is as follows:

$$SU=\frac{(E_S*ET)-DT}{E_S*ET}*100$$

SU	Sustainability
E_S	Elements scheduled
ET	Event time
DT	Total down time (Lost Time)

A.7 Multi-Element Impacts on Performance Calculations

In general the calculations of previously described AV and SU will be determined by element, not specific equipment within an element. Performance for an element will be decided by the ability of the element to support the scheduled user needs, but does not necessarily require that all elements operate at 100%. For example, the SPY radar transmitter can be operational in reduced power versus full power or the Signal Processor is still operational with one side down. Such conditions may be considered available for one user but not for another. An element will be considered available by using simulators or alternate equipment if the use of substitutes causes no adverse effect on the user's ability to complete his objective. Final AV and SU determinations will be based on consensus of the user and a site representative.

The following conditions will not be used to determine AV or SU:

1. Elements requested but not scheduled due to scheduling conflict.

2. Elements not scheduled but requested at start of or during an event.

3. Elements scheduled down for ORDALTs, field changes or long term PMS maintenance actions.

4. Problems caused by user provided computer programs.

A.8 Completed Event Evaluation

The completed event evaluation is the last stage of mission user event support. Completed event evaluation is the overall assessment of scheduled events with respect to availability, dependability and personnel support. This evaluation will be completed by the user test conductor by assigning a satisfaction grade of 1, 2, 3, or 4 and providing feedback in the comment section of the ATEAM completed event form.

A.9 Combat System Measures of Quality

Figure A-2 provides an overview of the interrelationships of performance measurement criteria.

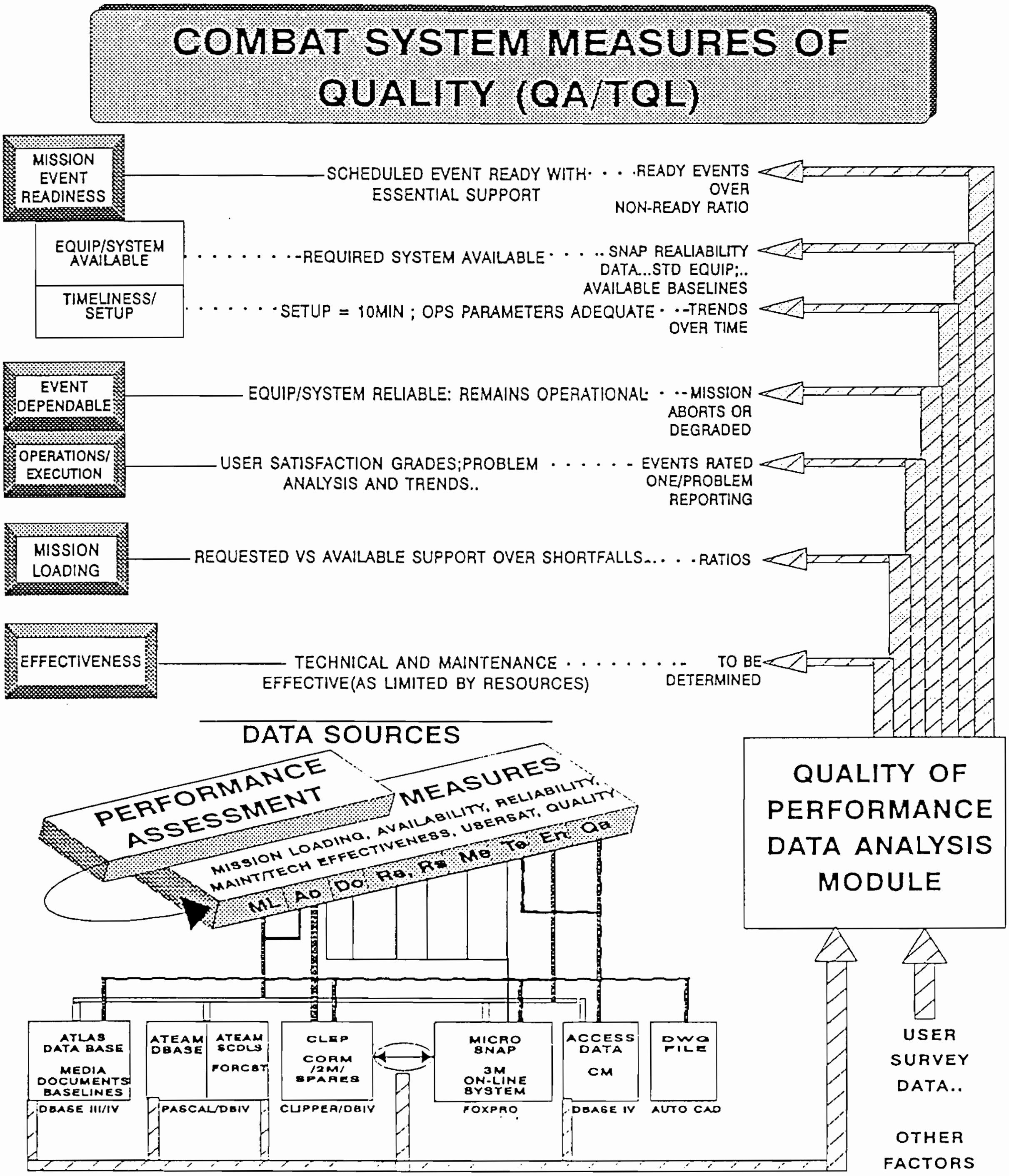


Figure A-2. Performance Evaluation Process

Appendix B - Listing of Standard Reports

B.1 Detail Report
B.2 ACSC Element Failure Rate
B.3 ACSC Maintenance Costs
B.4 ACSC User Feedback
B.5 ACSC Mission User Evaluation
B.6 ACSC Monthly Preventive and Corrective Maintenance (FF)
B.7 ACSC Causes of Corrective Maintenance
B.8 ACSC Impact of Maintenance on User Satisfaction
B.9 ACSC Preventive Maintenance (Actual v. Scheduled Maintenance)
B.10 ACSC Dependability (all users)
B.11 ACSC Mission Loading (Scheduled v. Requested)
B.12 ACSC Mission Loading (Completed v. Scheduled)
B.13 ACSC Mission Loading (Requested v. Completed)
B.14 ACSC Equipment Shortfall
B.15 ACSC Average Setup Time by User Group
B.16 ACSC Setup Time by Event Classification
B.17 ACSC Equipment Availability

Appendix B - Standard Reports

Appendix B defines the standard reports to be generated. Reporting periods and values used for each report are as specified below.

Many of the data items reported by CSPAS have only recently been collected and are currently being analyzed. Negative impact criteria for these reports are currently undefined, and will be determined for each report following analysis of specific data trends.

1.0 General Reporting Requirements

1.1 User reporting options. The following user options apply to all CSPAS reports unless otherwise specified in this Appendix:

User Options	Calculations	Miscellaneous Functions
Select time range and increment: **FY/CY** quarterly monthly **Quarter** monthly weekly **Month** weekly **Four Quarters** quarterly monthly **Date Range** Varies depending upon length of time selected - limited to 15 reporting periods **Data Display** Choose to display or hide data values on graph **Detail Report** Click on a data value and show background detail information.	**Trend lines:** standard linear regression analysis $slope(m) = \frac{\bar{y}\sum x - \sum xy}{\bar{x}\sum x - \sum x^2}$ $y\ intercept(b) = -m\bar{x} + \bar{y}$ Where: $\sum x$ = *sum of the x data* $\sum xy$ = *sum of the products of x and y* $\bar{x}$ = *mean of the x data* $\bar{y}$ = *mean of the y data* $\sum x^2$ = *sum of the squares of the x data*	Clipboard support Printing to MS Windows printers ***Resizable graph window Color selections Graph type selection Progress indicator ***Cancel reporting option Message window indicating data within negative impact range.

1.2 Report generation window. Figure B-1 depicts the general (preliminary) format of the CSPAS report generation window. Figure B-2 shows the general format of the Event Detail Report (representative depictions only).

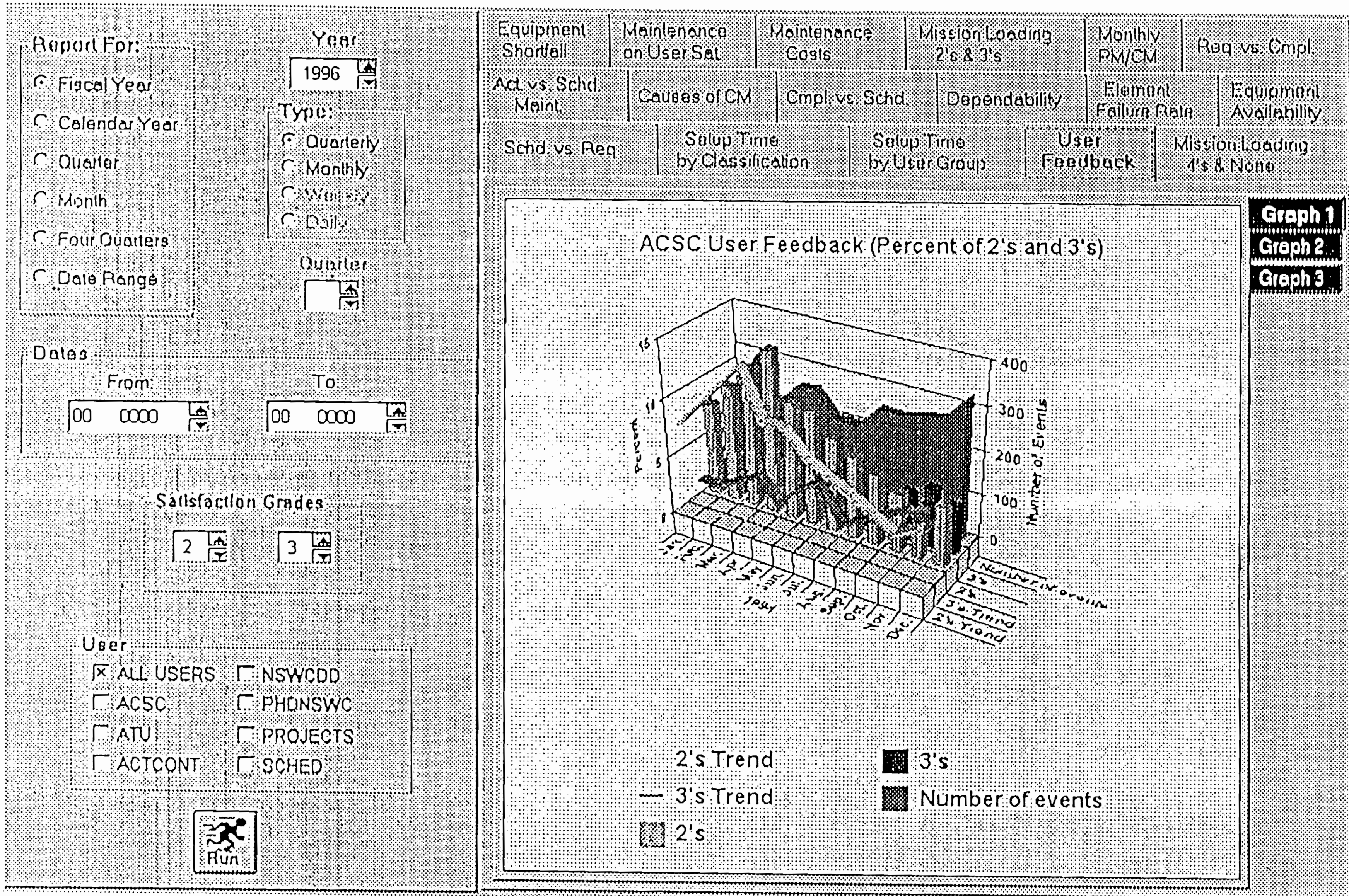


Figure B-1 Preliminary Report Generation Window

1.2.1 Operation of the report generation window. To execute any report, the user will select the desired report (by clicking on the appropriate report name tab). The user will then be presented with a default set of selection criteria, and may choose to run the report immediately or modify any or all of the selection criteria and then execute the report.

1.2.1.1 Viewing multiple versions of the same report. The user will have the option to create and view up to 3 versions (i.e., reports based upon up to 3 distinct sets of selection criteria) through the report window.

1.2.1.2 Saving a created report. The user will be given the option to save any generated report. The report can be saved at any time through a menu selection or icon. In addition, upon exiting the software without previously saving a report, the user will be prompted to save if desired. The formats for saved data are outlined in paragraph 3.2.1.1.1, Saving a created report.

1.2.1.3 Recalling a saved First Impression chart. A window will be provided which allows for the retrieval of a previously saved First Impression chart. The saved chart can be recalled by the user and can be manipulated (i.e., modified, printed, re-saved) by the user as desired.

1.2.1.4 Saving a report format file. The format file for a generated report can be saved. The format file includes report selection criteria. The user will be given the option to save the report format in his default directory. See paragraph 3.2.1.2.1, Saving a report format file.

1.2.1.5 Recalling a saved report format file. A list of previously saved report format files (from the user's default directory) will be presented to the user. From this list, the user will select the desired report format file file for further use.

2.0 Detailed data report (data drilldown)

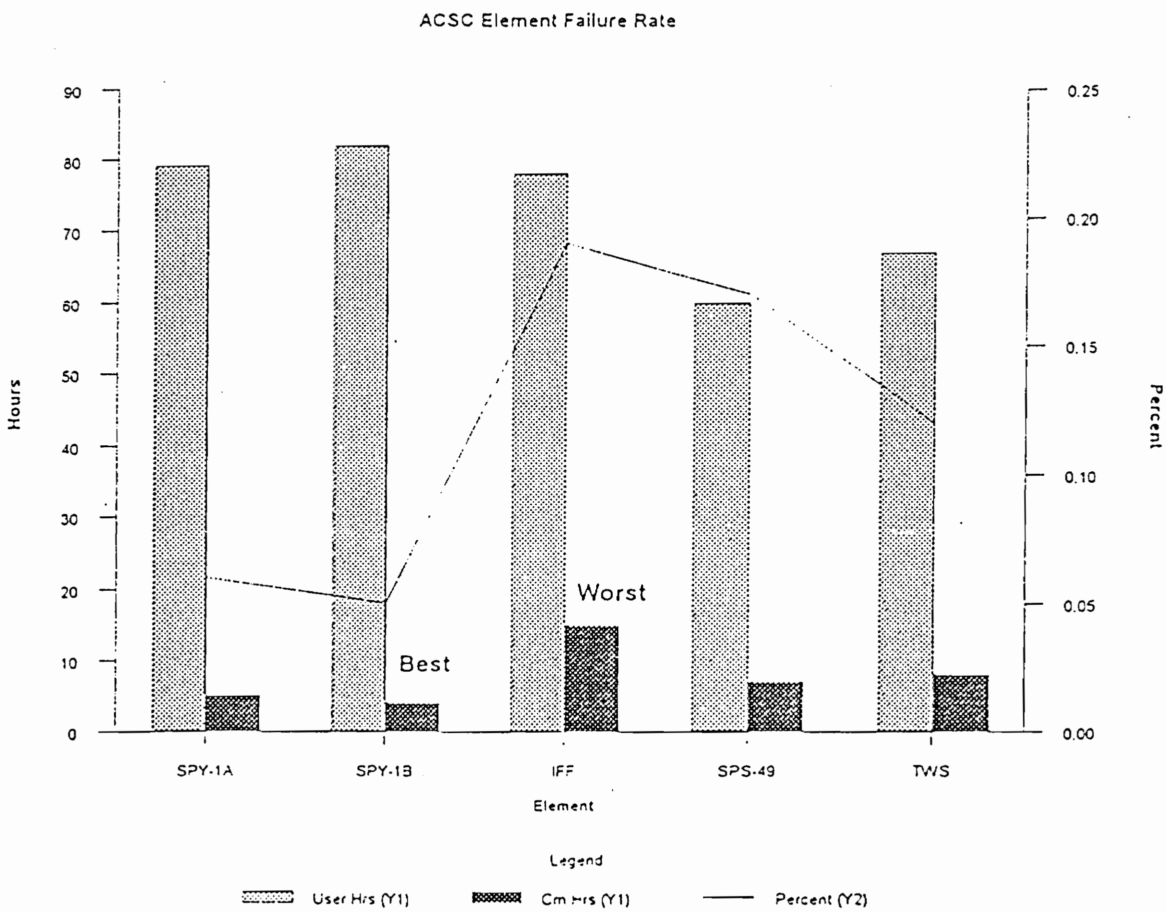
2.1 Accessing the detailed data report. This report is accessed by clicking on a data point on the graph. Information (either event- or maintenance-related, based upon the type of report currently in use) pertaining to the selected date will be displayed. This information will be for display purposes only; it will not be editable by the user.

2.2 Additional selection of detailed data. Initially, the information displayed on this window will represent date and user selections from the graph, i.e., if the user clicks on a graph object representing data from the first quarter of 1995 for all users, that data will be selected and displayed. The user will have the ability to change those parameters if desired and perform a new search.

3.0 Standard report formats. The following pages describe the formats of the standard reports to be developed for CSPAS.

3.1 Element Failure Rate

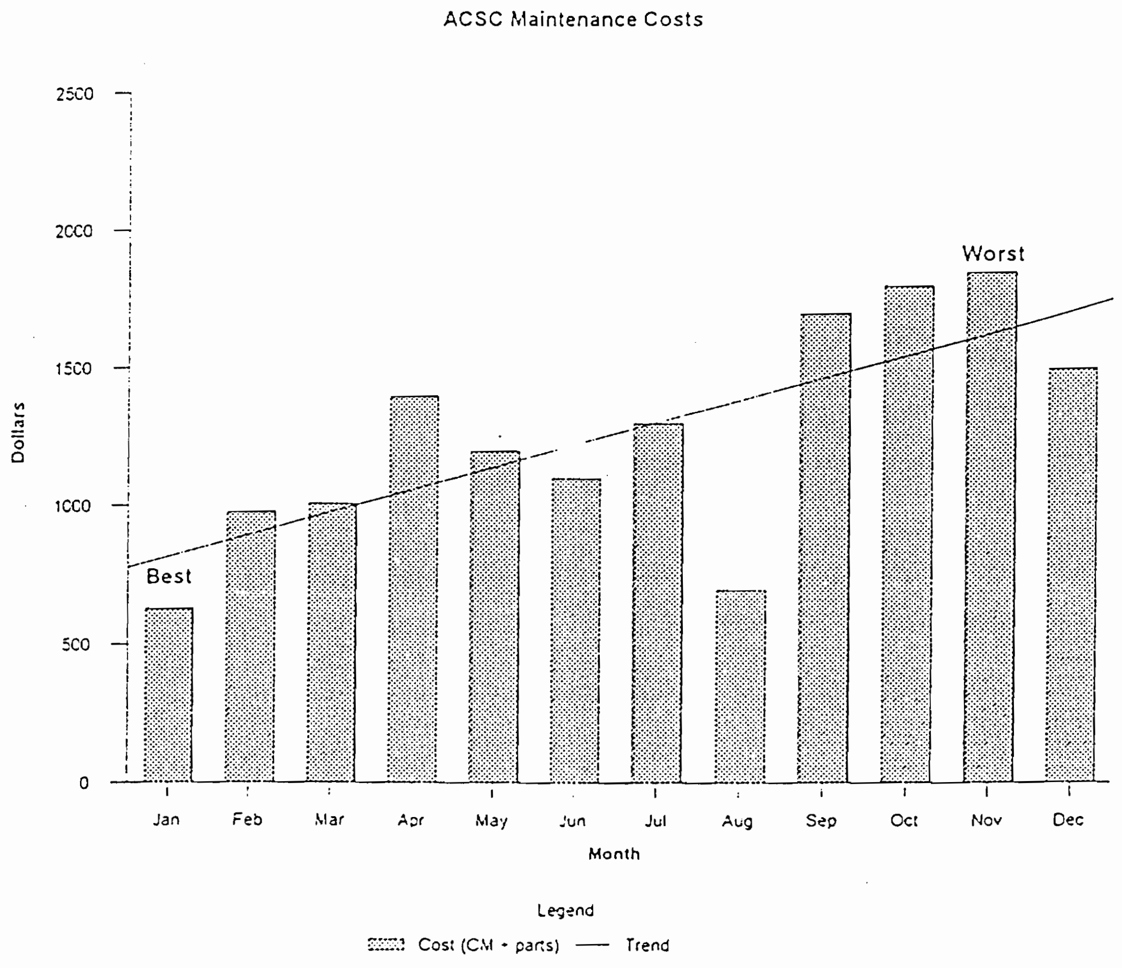
The Element Failure Rate report will portray element failure hours (through the use of corrective maintenance (CM) data) and compare this to element usage hours. The information presented in and subsequent analysis of this type of chart could be used to justify new and/or additional elements/equipments or as an indicator of elements/equipments requiring management attention.



User Options	Data Source(s)	Calculations	Negative Impact Criteria
Select elements to include Best/worst performer	CSMMOSS database, OARS	Hours: a = Count (user hours) b = Count (CM hours) Percent: b/a Best = lowest number of CM hours Worst = highest number of CM hours	To be determined

3.2 Maintenance Costs

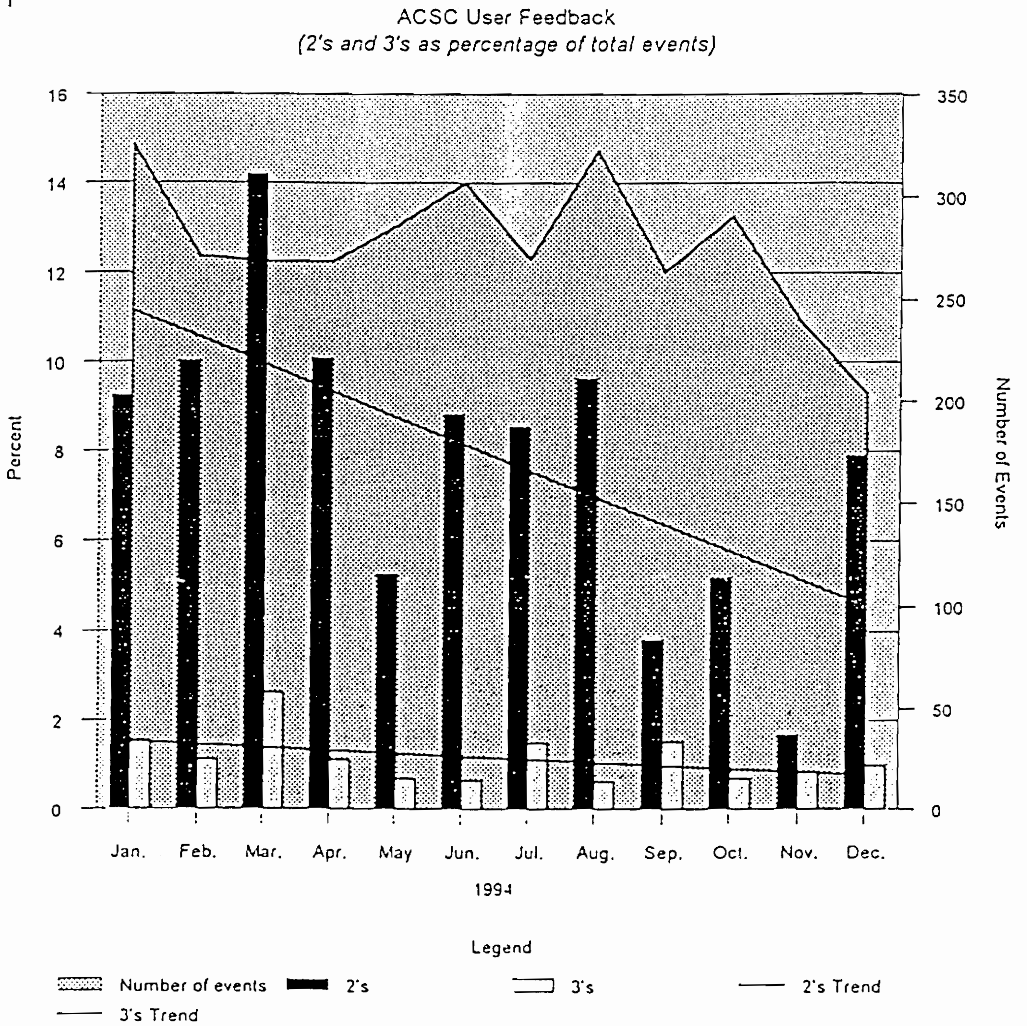
The Maintenance Costs report will display the cost of corrective maintenance (definiton of this parameter to be determined) and the cost of repair parts. A trend line will be displayed.



User Options	Data Source(s)	Calculations	Negative Impact Criteria
User must enter a labor rate. (default rate will be supplied) Best/worst performer	OARS, other undefined Cost of repair parts to be available in FY 96.	Maintenance costs = Cost of repair parts + (corrective maintenance hours * corrective maintenance rate) Best = lowest CM cost Worst = highest CM cost	To be determined

3.3 User Feedback

This chart will portray the percentage of two selected User Satisfaction Grades (as received by the Combat Systems Department from ACSC users) relative to the total number of events for a selected period. Data will be trended over time. The chart will enable ACSC to partially assess its fulfillment of user requirements.



User Options	Data Source(s)	Calculations	Negative Impact Criteria
Select ALL or specific user Select 2 grades to compare	Historical scheduling data, CSMMOSS database	a = Count (all events in date range) b = Count(all grade x) c = Count (all grade y) Percent(b) = b/a * 100 Percent(c) = c/a * 100 Trend lines for each grade reported	To be determined

3.4 Mission User Evaluation

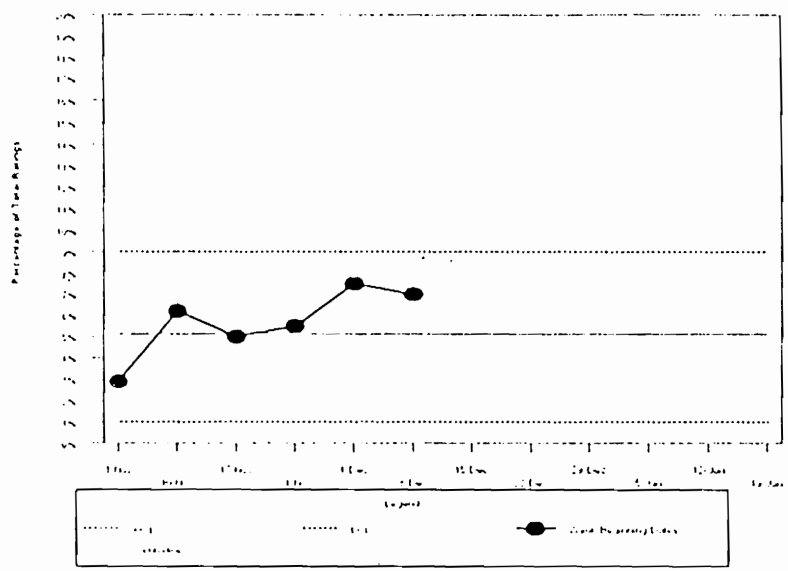
The Mission User Evaluation charts will show, for each individual user satisfaction grade, the percentage of total ratings, as well as upper and lower control limits, on a weekly basis. The upper and lower control limits will be recalculated on demand, i.e., only when specified by the program user.

<REPORT FORMAT SHOWN ON FOLLOWING PAGE>

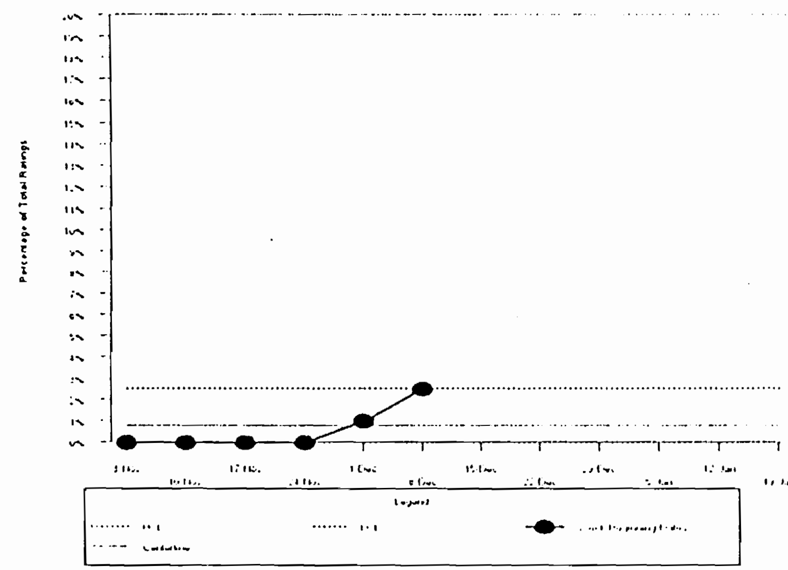
User Options	Data Source(s)	Calculations	Negative Impact Criteria
Select ALL or specific user Print a single chart or all charts on a single page	Historical scheduling data, CSMMOSS database	a = count(total events for the period) c = count(events for specific user satisfaction grade for the period) a' = count(all events for time period) / # periods (typically all events for a year / 12 months) u' = $\sum c_i / \sum a_i$ Upper Control Limit (UCL) = $u' = 3 + \sqrt{u'/a'}$ Lower Control Limit (LCL) = $u' = 3 - \sqrt{u'/a'}$	Items outside control limits

Mission User Evaluation 3 November - 8 December 1995

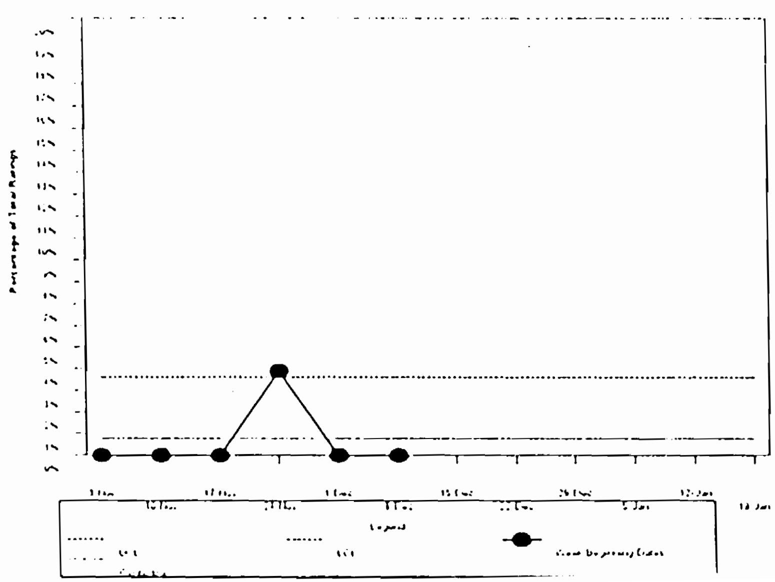
TWO'S



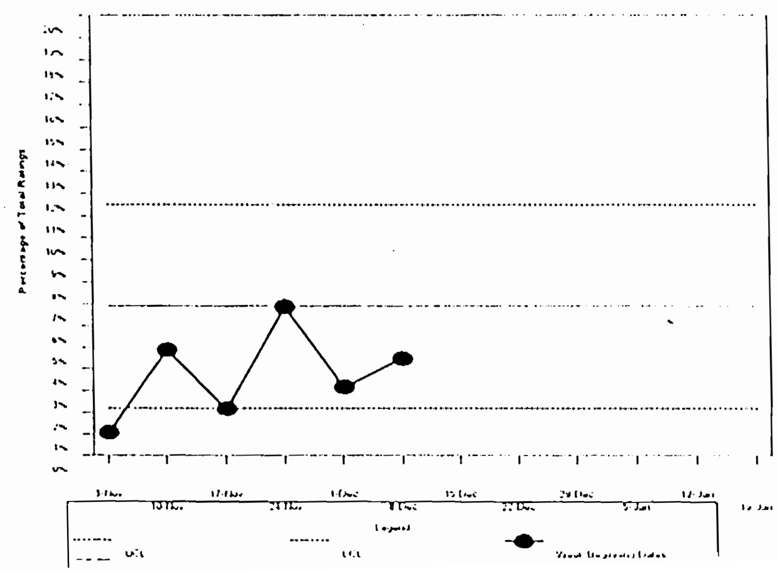
THREE'S



FOUR'S

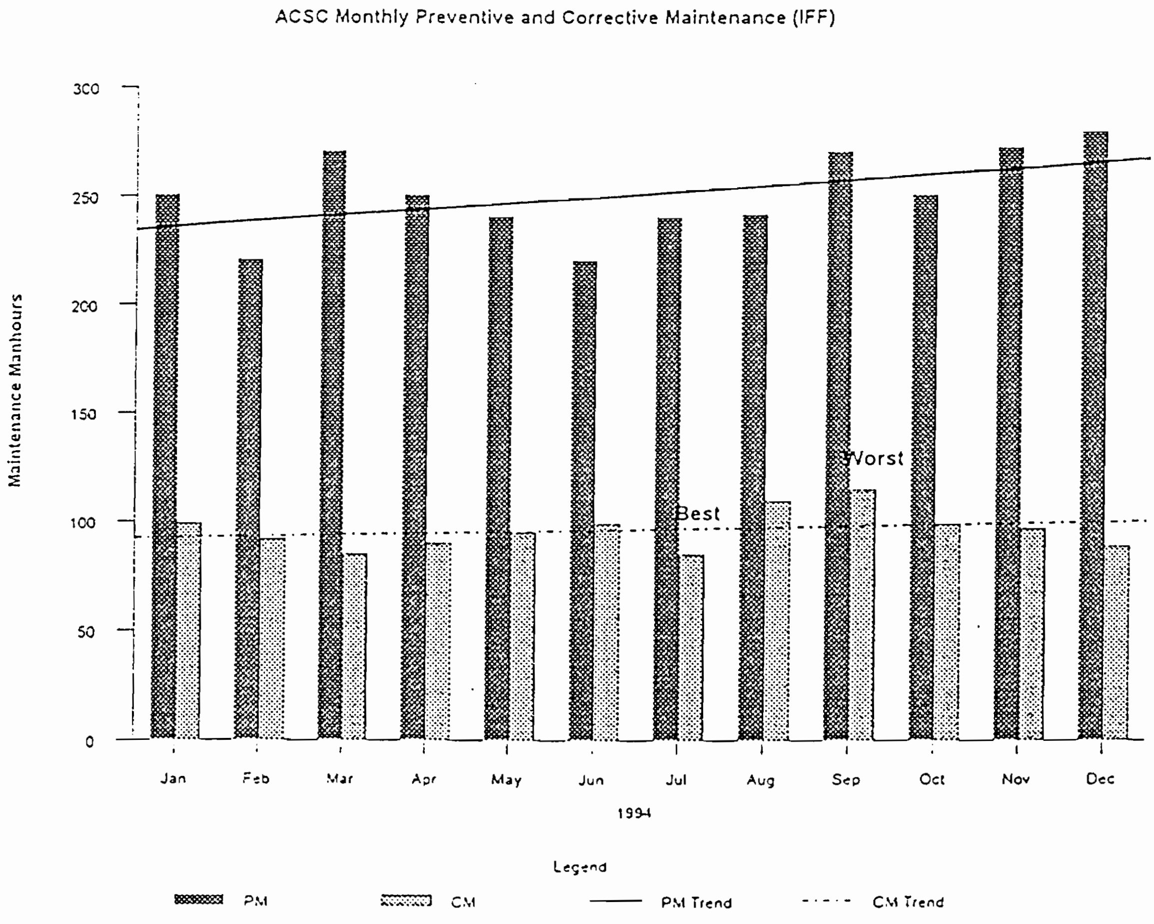


NONE'S



3.5 Monthly Preventive and Corrective Maintenance

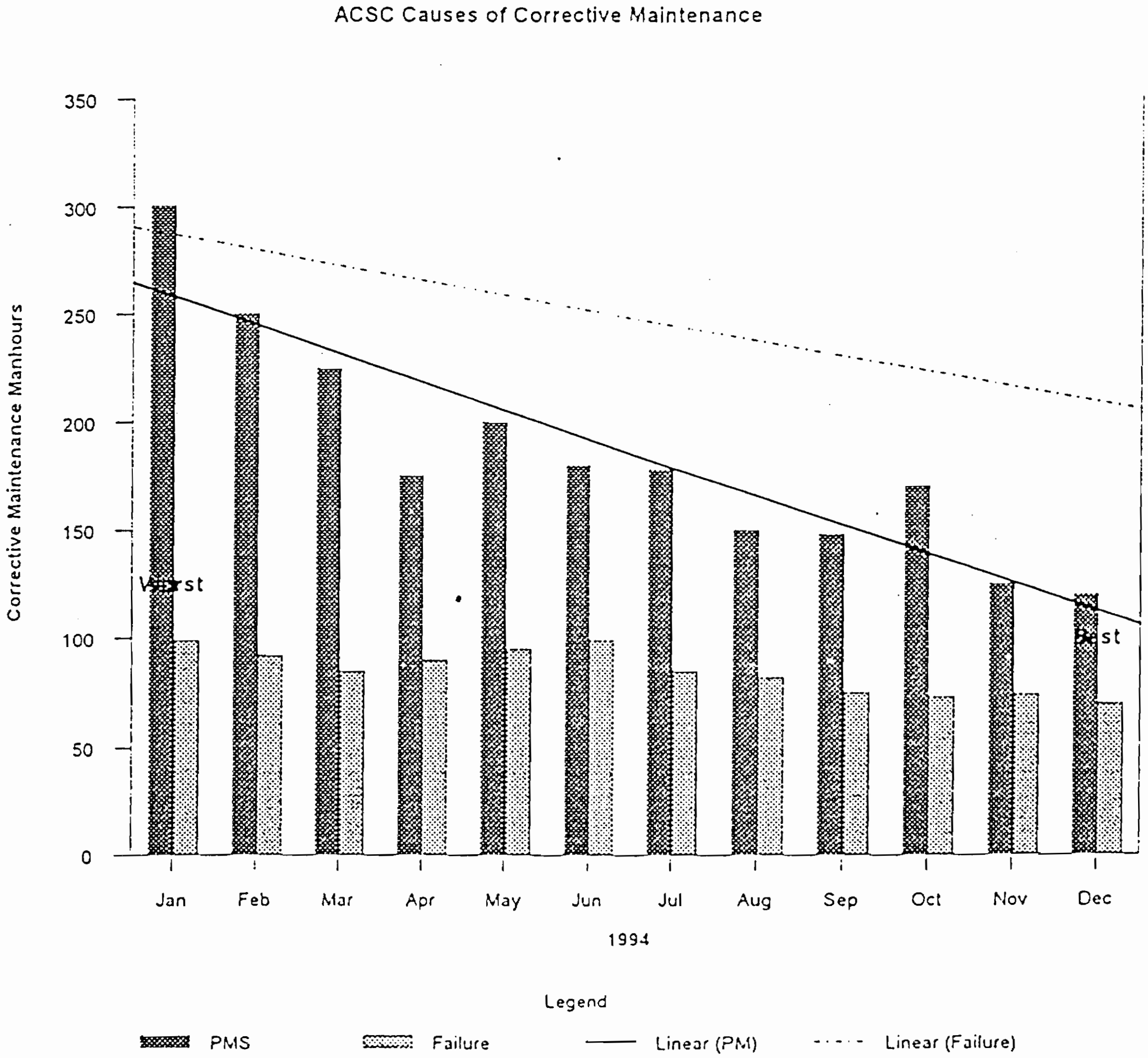
This report will display the number of hours (if hourly preventive maintenance data is obtainable) for both preventive and corrective maintenance actions. This data can be used to demonstrate the impact of actual maintenance hours conducted as compared to measurement parameters from other sources (e.g., cost of repair parts, type of maintenance, etc.).



User Options	Data Source(s)	Calculations	Negative Impact Criteria
Select element Best /worst performer	OARS	Count (preventive maintenance hours / tasks)****** Count (corrective maintenance hours) Trend lines for each count Best = lowest CM hours Worst = highest CM hours	To be determined

3.6 Causes of Corrective Maintenance

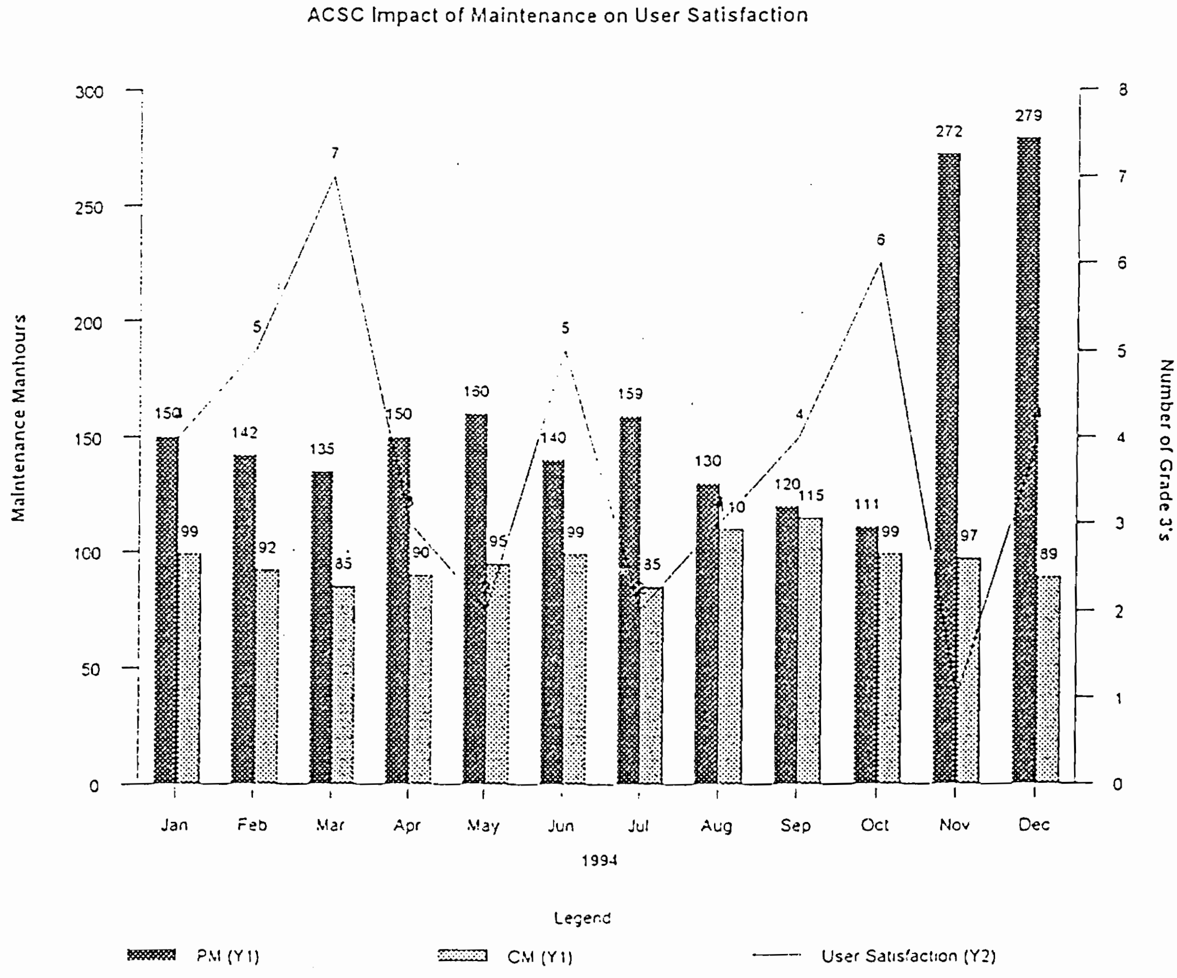
This report will show hours of corrective maintenance as a result of PMS and as a result of failure.



User Options	Data Source(s)	Calculations	Negative Impact Criteria
Best/worst performer	OARS	Count (corrective maintenance hours as a result of PMS) Count (corrective maintenance hours as a result of failure) Trend lines for each count Best = lowest failure hours Worst = highest failure hours	To be determined

3.7 Impact of PM/CM on User Satisfaction

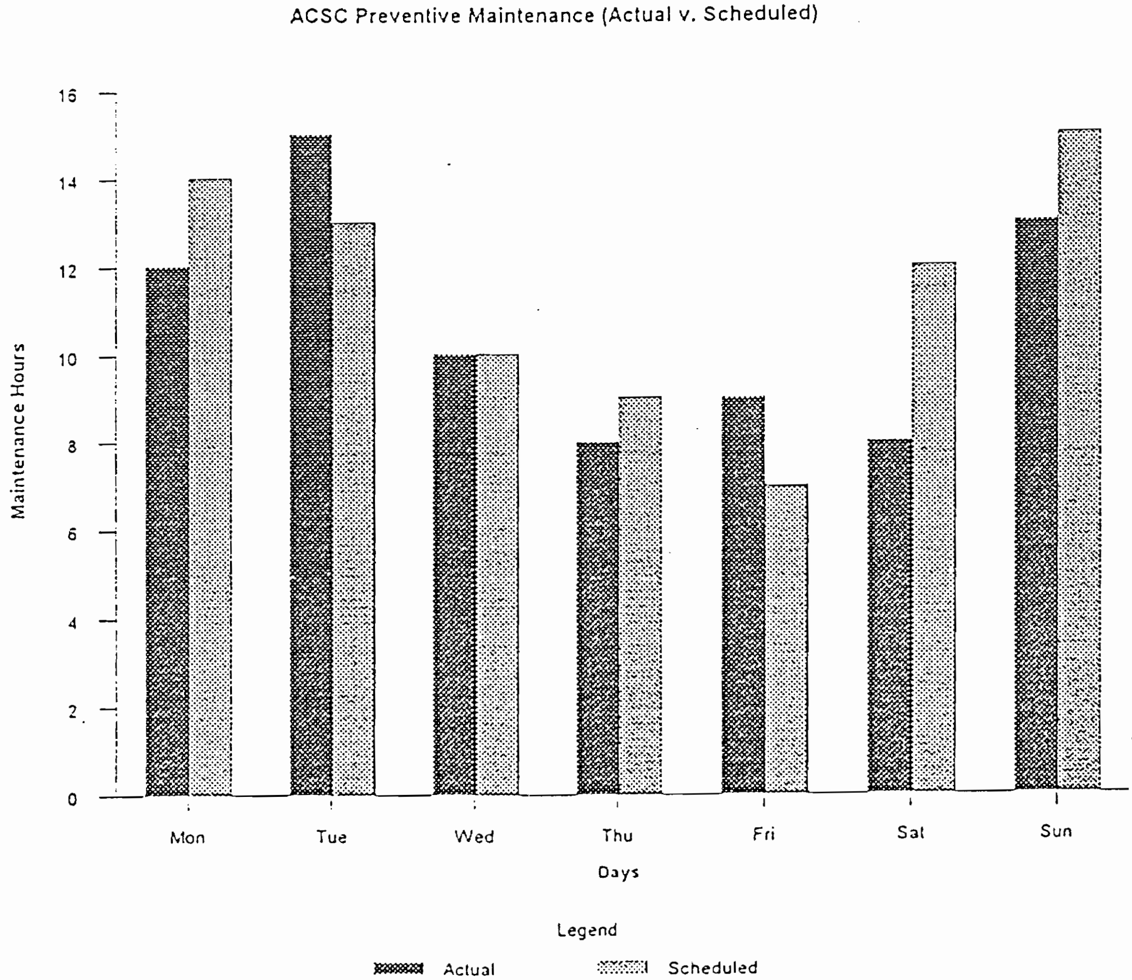
This chart will show the number of hours for both preventive and corrective maintenance, along with the count of the selected User Satisfaction Grade as received by the Combat Systems Department from ACSC users for a given period.



User Options	Data Source(s)	Calculations	Negative Impact Criteria
Select any user grade to display	CSMMOSS database	Count (corrective maintenance actions as a result of PMS) Count (corrective maintenance hours as a result of failure) Count (selected user satisfaction grade)	To be determined

3.8 Actual v. Scheduled Maintenance

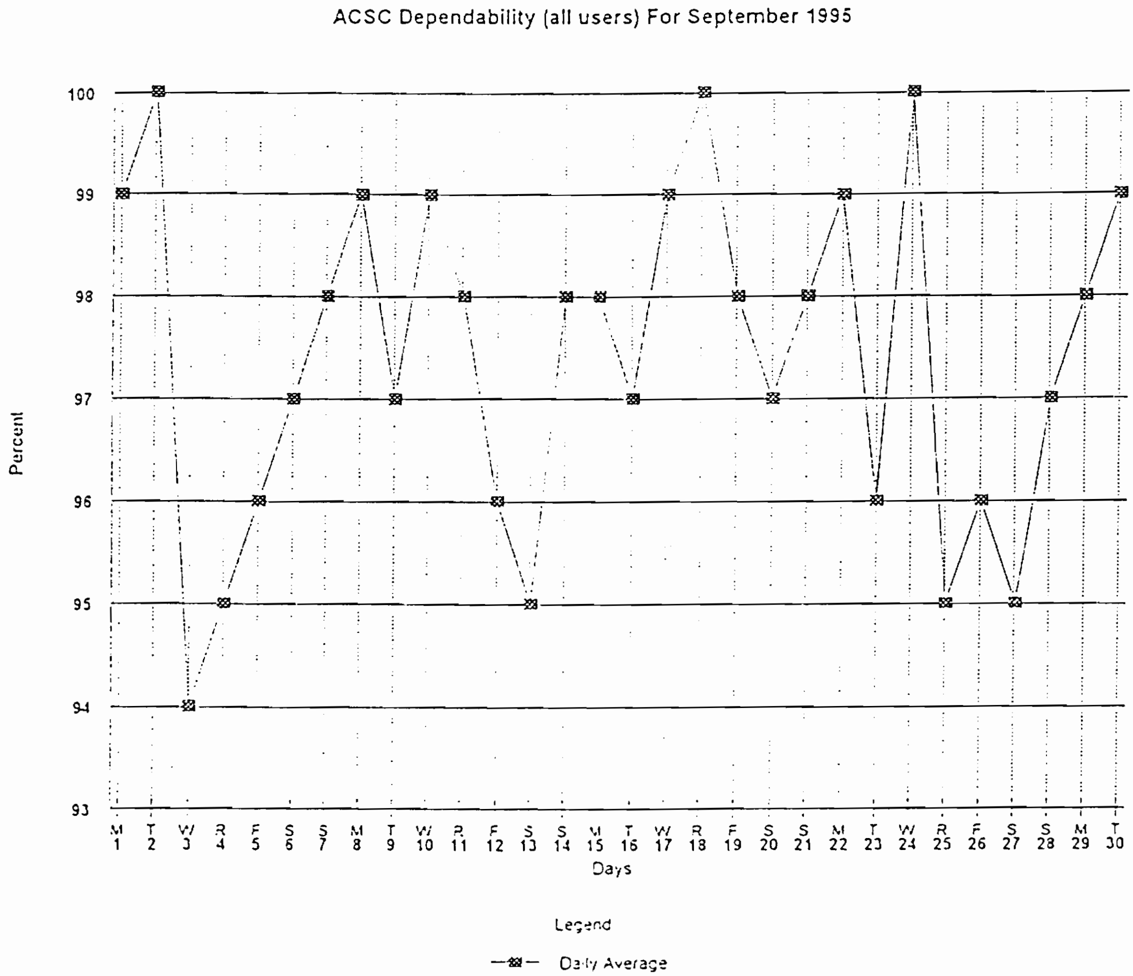
This report displays actual preventive maintenance hours and scheduled preventive maintenance hours. Creation of this report is contingent upon the ability to obtain hourly preventive maintenance data.



User Options	Data Source(s)	Calculations	Negative Impact Criteria
Select element	CSMMOSS database	Count (scheduled maintenance hours (or tasks)) TBD Count (actual maintenance hours (or tasks)) TBD	To be determined

3.9 Dependability

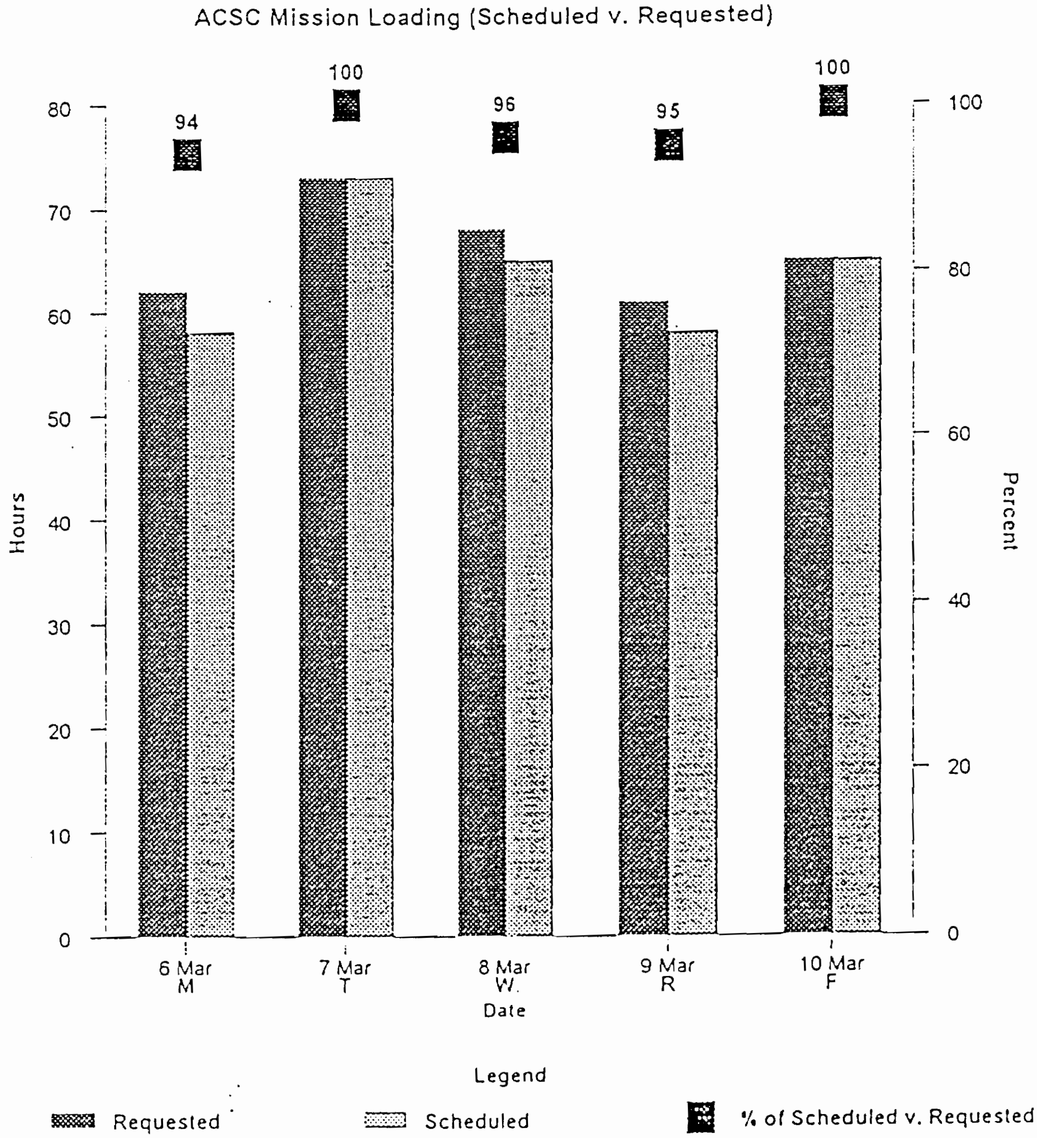
This report shows the average dependability (as defined in Appendix A) of one or all users on a daily basis. Dependability is used as a measure of ACSC general operations performance.



User Options	Data Source(s)	Calculations	Negative Impact Criteria
Select all or one user	CSM/MOSS database	a = Sum(all dependability values for the selected user(s) and date range) b = Count(all events for period) Daily Average = a/b	To be determined

3.10 Mission Loading

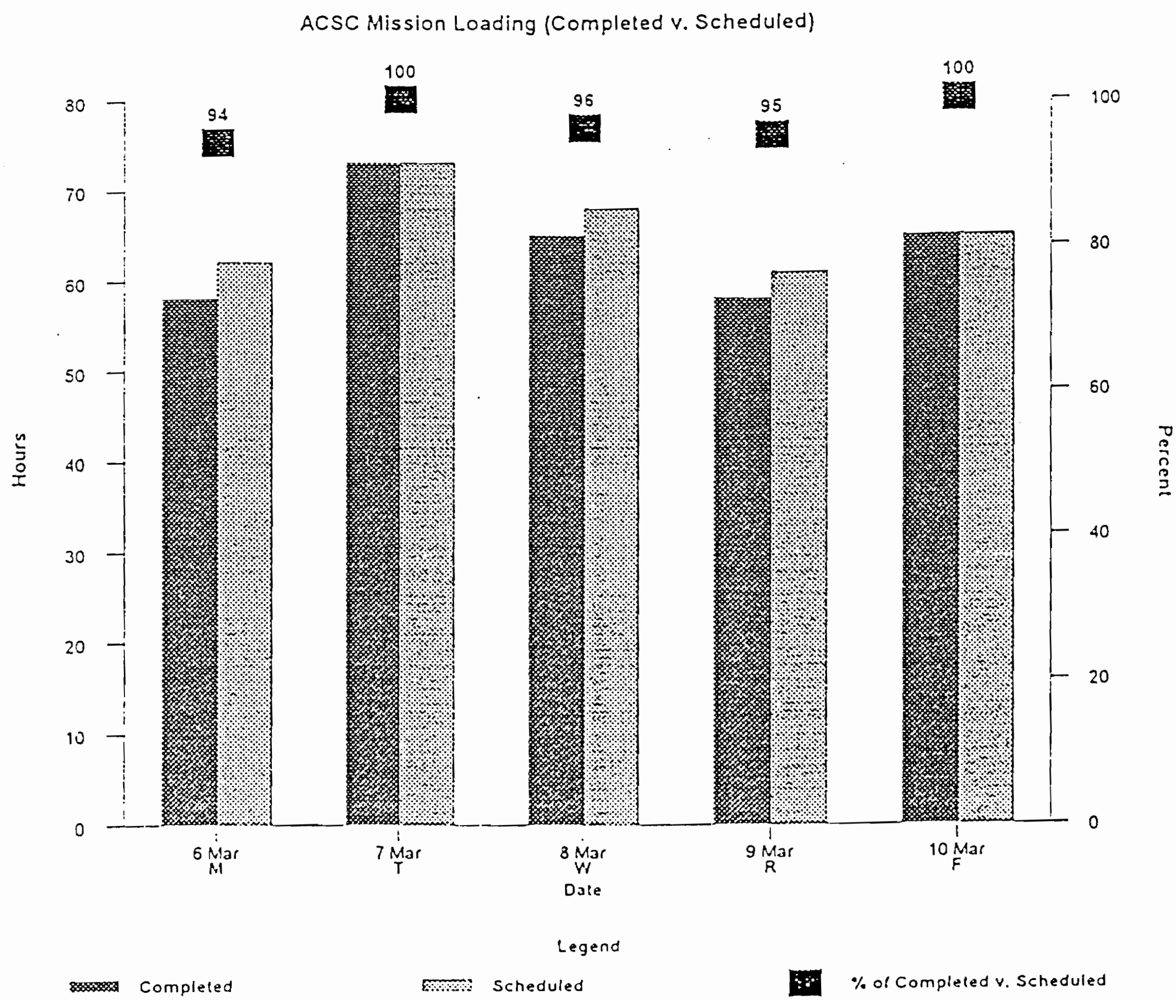
This report depicts mission loading data in the form of scheduled event hours vs. requested event hours.



User Options	Data Source(s)	Calculations	Negative Impact Criteria
Select ALL or one user	CSMMOSS database	a = Count (requested event hours) b = Count (scheduled event hours) Percent = b/a	To be determined

3.11 Mission Loading

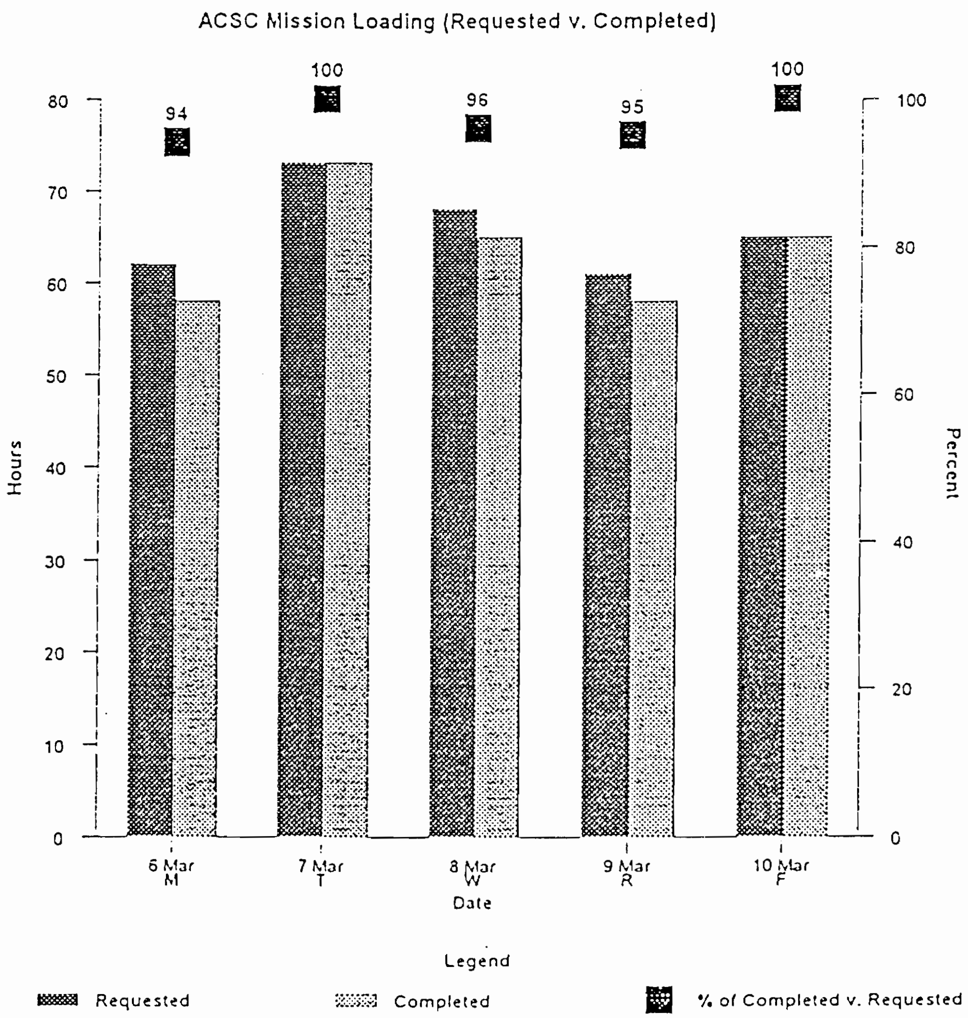
This report depicts mission loading data in the form of completed event hours vs. scheduled event hours.



User Options	Data Source(s)	Calculations	Negative Impact Criteria
Select ALL or one user	CSMMOSS database	a = Count (scheduled event hours) b = Count (completed event hours) Percent = b/a	To be determined

3.12 Mission Loading

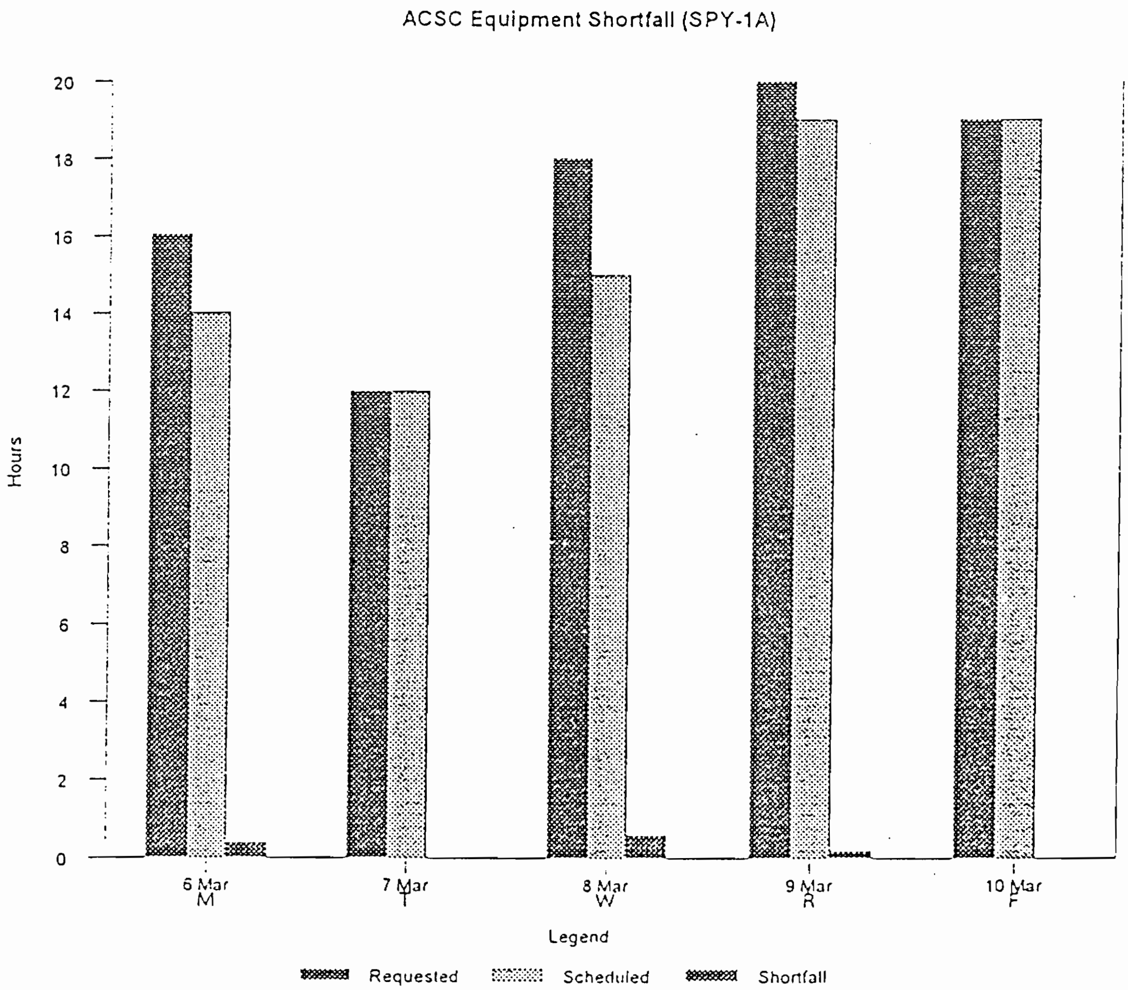
This report depicts mission loading data in the form of requested event hours vs. completed event hours.



User Options	Data Source(s)	Calculations	Negative Impact Criteria
Select ALL or one user	CSMMOSS database	a = Count (requested event hours) b = Count (completed event hours) Percent = b/a	To be determined

3.13 Equipment Shortfalls

This chart is a variation of the Mission Loading report which depicts scheduling shortfalls for a specific piece of equipment. This data will assist ACSC in assessing how well it is meeting user requirements on an element by element basis.



User Options	Data Source(s)	Calculations	Negative Impact Criteria
Select ALL or one user Select equipment	CSMMOSS database	a = Count (requested equipment hours) b = Count (scheduled equipment hours) Shortfall = a - b	To be determined

3.14 Setup Time

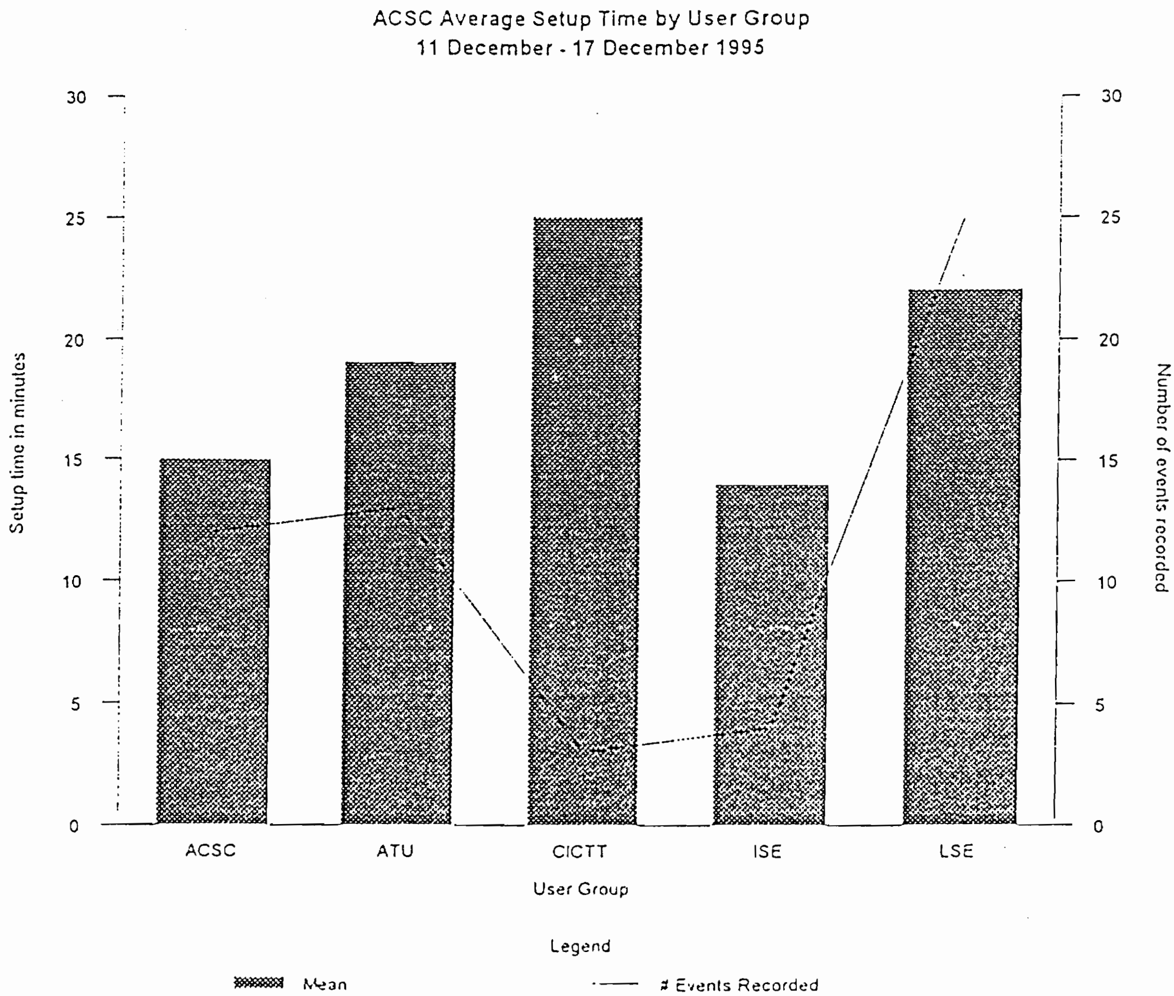
The setup time reports display several views of setup time. Information shown here will be used to formulate improvements in the event setup process. This report is a variation of the Availability report (defined below).

Setup Time by User Group
11 December - 17 December 1995

	# Events Recorded	Mean	Median	Mode	1-15 Minutes	16-30 Minutes	Over 30 Minutes
ACSC	13	16.23	15	10	8	5	0
ATU	13	19.62	15	15	8	4	1
CICTT	3	26.6667	20	N/A	1	1	1
LSE	3	15.00	15	N/A	2	1	0
LSE	27	22.22	15	15	15	9	3
PROJ	0						

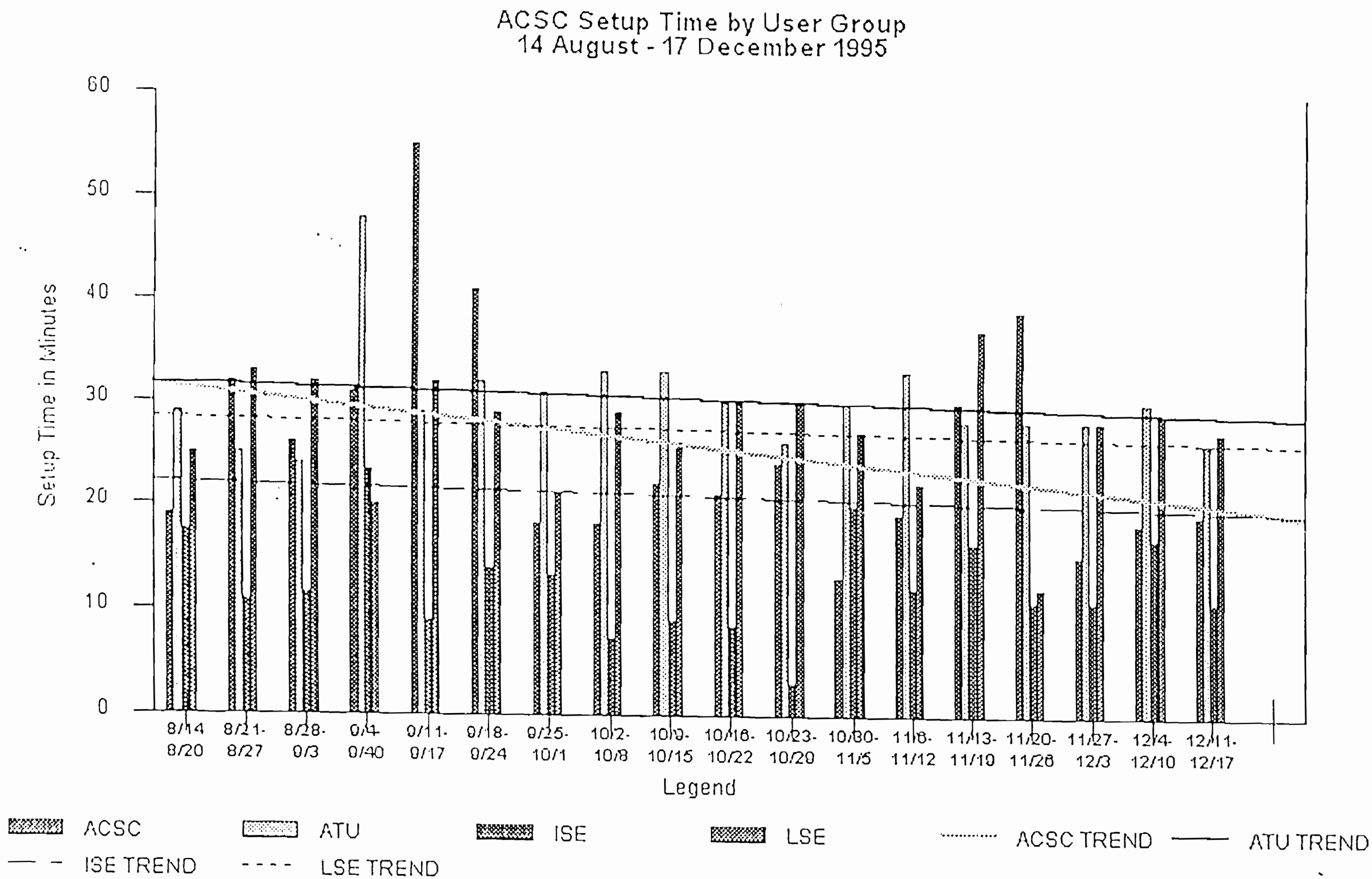
•Total number of events = 110 •Number of events that do not require setup time = 51
•Number of events without recorded setup time = 0

3.14 Setup Time (con't)



User Options	Data Source(s)	Calculations	Negative Impact Criteria
Select User Select date range	CSMMOSS database	a = Sum (setup time in minutes for each user) b = Count (number of events recorded for period for each user) Mean = a/b for each user	To be determined

3.14 Setup Time (con't)

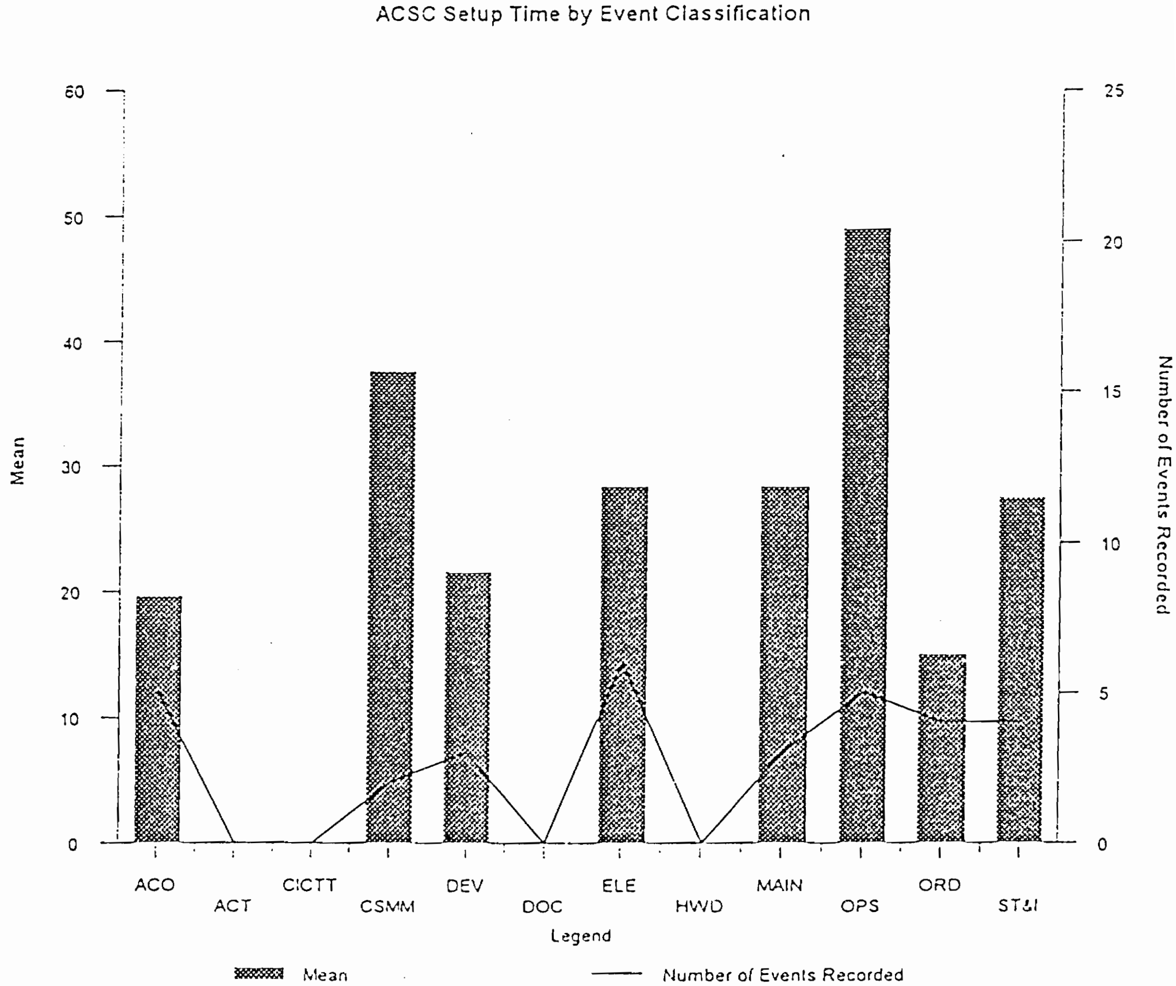


3.15 Setup Time by Event Classification

Setup Time by Event Classification
11 December - 17 December 1995

	# Events Recorded	Mean	Median	Mode	1-15 Minutes	16-30 Minutes	Over 30 Minutes
ACO	5	19.60	15	15	3	2	0
ACT	0						
CICTT	0						
CSMM	2	37.50	37.5		0	1	1
DEV	3	21.67	15		2	0	1
DOC	0						
ELE	65	28.33	15	15	4	0	2
HWD	0						
MAIN	3	28.33	30		0	2	1
OPS	5	49.00	60	75	1	1	3
ORD	4	15.00	15	15	4	0	0
ST&I	4	27.50	22.50	20	0	3	1

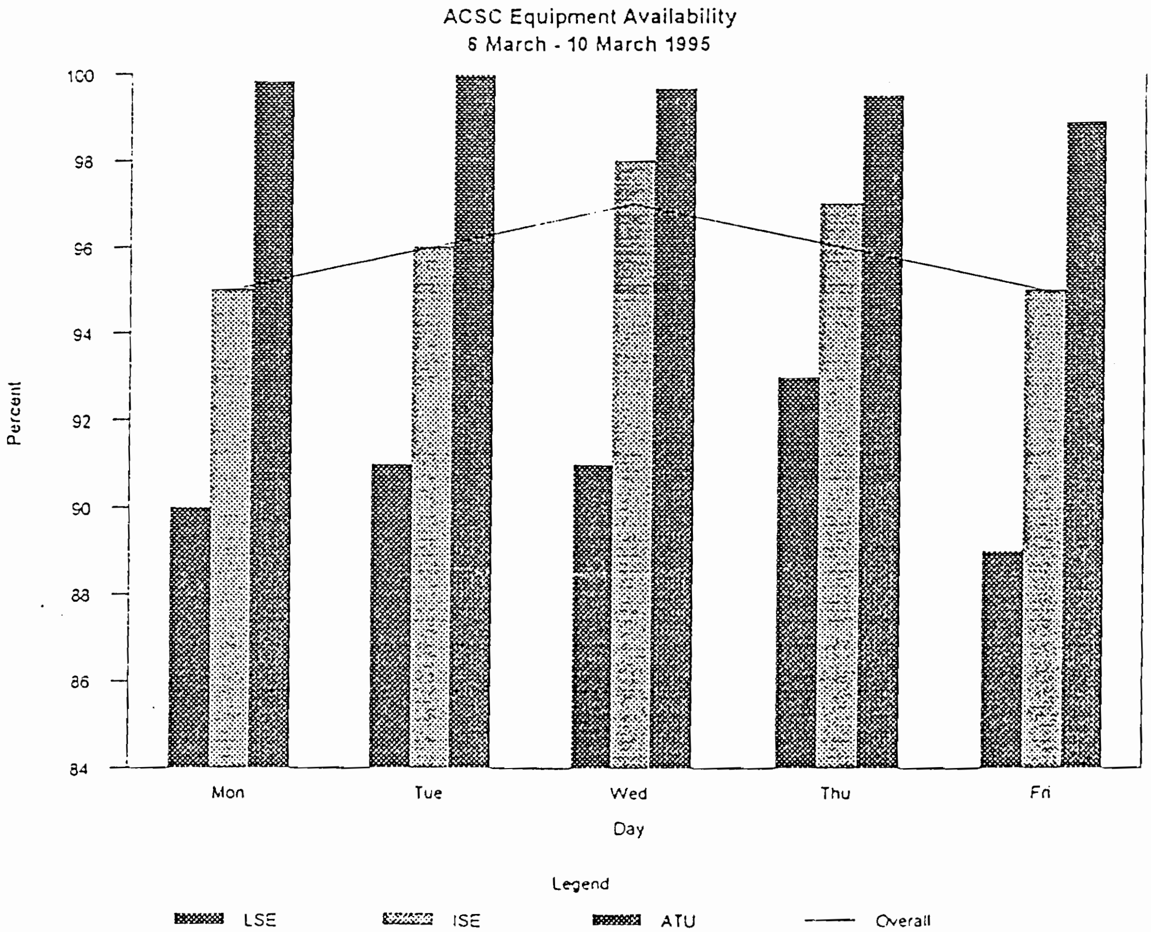
3.15 Setup Time by Event Classification (con't)



User Options	Data Source(s)	Calculations	Negative Impact Criteria
Select date range	CSMMOSS database	a = Sum (setup time in minutes for each event classification) b = Count (number of events recorded for period for each event classification) Mean = a/b for each user	To be determined

3.16 Equipment Availability

This chart shows the availability of system and personnel (as defined in Appendix A) for event conduct at the start of the event. The information presented in and subsequent analysis of this chart will be used to target certain specific user event setups for process improvement.



User Options	Data Source(s)	Calculations	Negative Impact Criteria
Select ALL or up to 3 users USER SELECT: User can enter a series of characters to search for in the event description, which will be reported as a separate bar on the graph.	CSMMOSS database	None NOTE: Some data from "User Select" search may be contained in other data items on the graph.	To be determined

Appendix C - Selection of CSPAS/MMOSS Development Software

Included in this appendix are notes and summaries regarding the selection of CSPAS development software. The Sr. Systems Designer selected development software based upon the technical rationale provided within these notes and summaries.

Please see below some information on Oracle in the government. Oracle has a 70% market share in the RDBMS market and has been the best choice for government customers requiring powerful, scalable and portable information management solutions. You should be receiving some literature on the WebServer in the next week or so. Let's talk then about how that solution would benefit the Aegis Combat Support Center.

ORACLE WINS HUMAN RESOURCES SOFTWARE DEAL

BETHESDA, Md., Feb. 12, 1996-Oracle Corp. announced today plans to upgrade two of the U.S. Defense Department's largest personnel systems with Oracle Human Resources software. The Office of the Secretary of Defense selected Oracle to build the DoD's new Civilian Personnel Data System (DCPDS), while the U.S. Air Force picked the same application software to modernize the service's military Personnel Data System (PDS).

The combined orders represent what is believed to be the world's largest human resources application. The new systems will manage information on nearly 800,000 civilians in the Defense Department and more than 1.2 million Air Force military and retired personnel. The Air Force Personnel Center anticipates that more than 40 percent of all federal employees not accounted for in this initial implementation ultimately will be migrated to the DCPDS.

All aspects of the personnel life cycle will be managed through these systems, including procurement, education and training, development and utilization and personnel data management functions. Oracle software, maintenance support and services will be delivered through Logicon, Inc.'s I-CASE contract.

The software deal can be delivered under a three-phase, multi-year package that includes the Applications software, the Oracle7 database, parallel query, parallel server software and advanced replication technology for distributing information to the more than 350 DCPDS sites and 100 PDS locations worldwide. The order also includes supporting development tools for Oracle Human Resources and Oracle's Designer/2000 CASE tools.

For DoD and Air Force officials, flexibility, scaleability and DoD standards compliance were key issues. Oracle's selection also is consistent with Defense Department policy to use commercial off-the-shelf software when possible. Each of the personnel system modernization efforts require open-system architectures to address the very large database requirements and responsiveness issues associated with supporting the changing force structures within the Department of Defense.

Oracle's solution includes pre-built modules and consulting services to create extensions to the database and Human Resource modules, as well as integrate CASE-developed applications into the HR system. The platform is designed to meet or exceed all current requirements, forecasted National Performance Review objectives and DoD commitments.

"We are proud to be supplying the U.S. Defense Department, one of the largest employers in the world, with Oracle Human Resources software," said Jay Nussbaum, senior vice president and general manager, Oracle Government. "Customers are now realizing the benefits of the tremendous investment Oracle has made over the last several years in developing best-of-breed applications software."

Oracle Human Resources manages all aspects of employee records, including job skills, training history, benefits tracking and dependent history. Because it maintains a single data repository, record duplication is eliminated and data entry time is reduced. The software manages the entire recruitment cycle, designs organizational business models and performs compensation administration. The result is a cost-effective method to analyze and control all employee-related expenses.

Oracle Corp. is the world's largest supplier of software for information management, and the world's second largest software company. With annual revenues of approximately $3.5 billion, the company offers its database, tools and application products, along with related consulting, education and support services, in more than 90 countries around the world.

For more information about Oracle, please contact Oracle's headquarters at (415) 506-7000. Oracle's World Wide Web address is (URL) http://www.oracle.com/.

Regards,

Greg Bateman

Oracle Corporation
196 Van Buren Street
Herndon, VA 22070
gbateman@us.oracle.com
Phone: 703-708-6803
Fax: 703-708-7922

SUMMARY OF FEATURES

COGNOS *IMPROMPTU* v. ORACLE *DISCOVERER 2000*

SCALE = 1 (Best) to 5 (Worst)

FEATURE	*IMPROMPTU*	*DISCOVERER 2000*	*COMMENTS*
Use of component packages	2	2	Impromptu can be purchased in the Administrative and User editions. Another package, *PowerPlay*, can be purchased separately to work with Impromptu data. *PowerPlay* allows the creation of graphic "dimensional" data which can be used for presentations, etc. Discoverer 2000 includes 2 packages, *Data Query* and *Data Browser*. *Data Query* is the primary reporting module which allows creation of both text reports and graphs. *Data Browser* is used for quick creation and execution of queries into the database.
Ease of initial setup	1	1	Both packages require the creation of a database description which is made available to end users. For the most part, this happens with very little intervention on the part of the database administrator. In Discoverer, however, certain features must be manually set up and enabled. Impromptu has slightly better presentation for the database information.
Ease of text report creation	2	1	Discoverer is easier to use for initial selection of data, for adding summary calculations, and creating crosstab reports.
Availability of formatting features	1	4	Impromptu offers the ability to add graphics to any report, and to graphically format text data.

FEATURE	*IMPROMPTU*	*DISCOVERER 2000*	*COMMENTS*
Charting capability	3	2	Impromptu charting is easy to use, but it is difficult to create an overall chart for a report. Discoverer will create an overall chart. Neither package will allow the user to fully annotate the chart once it is created. Some flexibility can be gained by exporting to Excel.
Outward compatibility with other software packages.	1	3	Impromptu will export data in a variety of formats which are directly compatible with many Windows applications. Discoverer will export in four formats which may be used to import data into other applications.
Inward compatibility with other software packages.	1	3	Impromptu will automatically import many different data formats, and will create reports on many different database formats. Discoverer will import only SQL queries, and works only with the Oracle database.

Mary Lotz

LOCKHEED MARTIN

Martin Marietta Services, Inc.
a Lockheed Martin company
33531 Chincoteague Road
Wallops Island, VA 23337

ST096-025

To: Distribution

From: M. L. Lotz

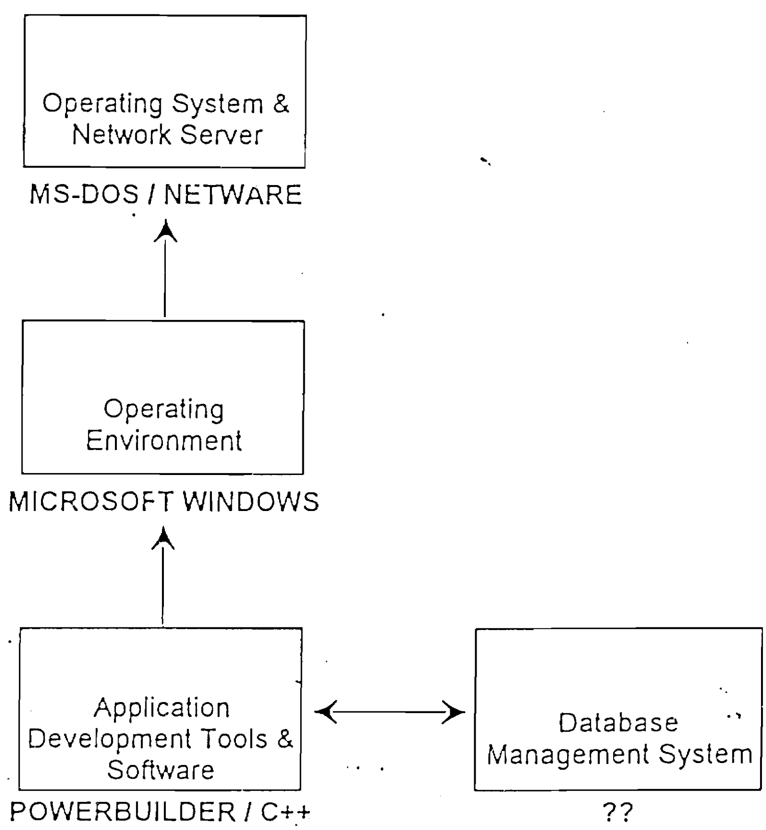
Date: March 27, 1996

Phone: (804) 824-5740

Re: RE: Database Management System Selection for CSMMOSS

Background

During the initial requirements definition phase for the CSMMOSS development effort, it was determined that the applications to be developed required the procurement of a relational database management system (DBMS). The current development environment is shown below:



A DBMS can provide many capabilities for data storage and retrieval operations, including improved speed and reliability of data retrievals. Some other advantages gained from the use of a DBMS are listed below:

1. A DBMS provides a common storage area for data which can be managed (administered) as a single unit.

2. Through Structured Query Language (SQL), data may be retrieved from a database in many combinations and forms. SQL is the software standard by which users pull data via queries to relational databases.

3. A DBMS can provide data integrity - methods for ensuring that data is complete and non-redundant.

4. Data security can be provided through a DBMS. Through the DBMS administrative functions, users can be granted detailed access privileges.

5. A DBMS provides optimization for queries. The DBMS gathers and stores comprehensive statistics about database structures. The DBMS optimizer will choose the most efficient access paths to data based on the information gathered. If data is extremely dynamic, this optimization may involve estimating statistics by scanning random samples of the data.

6. In a client/server environment, a DBMS provides partitioning of processing by using stored triggers and procedures. Database triggers and procedures are pieces of procedural code actually stored within the database. When used, they perform requested processing at the server, rather than at the client. This speeds processing and results in fewer data transfers between the client and server.

7. A DBMS can manage multi-user access and can provide assistance in dealing with concurrency issues, or situations which arise within the database when many users request access to a particular record of the database at once. For example, some DBMS's contain sequence number generators used for creating unique key values, eliminating the necessity for the use of one common database table to perform the task for many record creations.

Many other capabilities can be provided through a DBMS. Many DBMS products are available; each one may provide these features in a different way.

Selection process

The following factors were considered in the DBMS selection process:

1. Reputation of DBMS company. Because so many DBMS products exist, and because time for selection is considered critical, it was necessary to quickly eliminate as many products as possible. As with any product, a DBMS which has been in use for an extended time period and is well known within the software industry is most desirable.

2. Performance. Database performance can be measured in terms of the DBMS's throughput capacity. This is defined as the number of transactions per second a system can execute given a

certain number of users and given a certain response time for those users.

3. **Concurrency control methods.** As mentioned above, a DBMS enables multi-user access to data. In conjunction with this ability, methods are required to prevent more than one user from attempting to change the same database record at the same time. Each DBMS has one or more methods for controlling concurrency. The concurrency control method used can be a significant factor in the efficient operation of the database.

4. **ODBC compliance.** Open Data Base Connectivity (ODBC) is an industry standard for database drivers which allows connection to serveral different types of databases. This is particularly important in a client/server environment where many different, local databases may be in use on the network.

5. **Ease of database administration.** A database must be administered (i.e., managed by a person knowledgeable in the structure and requirements of the data). All DBMS's provide some facility for performing administrative tasks. Most desirable is a system which does not require extensive knowledge of SQL to accomplish database administration.

6. **Stored triggers and procedures.** As indicated above, stored triggers and procedures are essential in obtaining the best performance in a client/server environment.

7. **Scalability.** Scalability of a DBMS refers to its ability to be used in a much larger environment (for example, as the number of clients in a client/server environment grows), or in a much smaller environment (on a local PC, for example). Not all DBMS products are completely scalable; often some conversion process is required to change environment size. This feature is important because of its impact on the future growth potential of the database system.

Selection Results

Many DBMS products are available with very similar features and capabilities. As outlined above, reputation of the DBMS company was chosen as the initial selection criterion. On this basis, two DBMS products were selected: Oracle Workgroup 2000 (a product of Oracle Corporation), and Sybase SQL Server 11.0 (a product of Sybase Corporation), and their features further evaluated. Below is a summary of their capabilities:

1. **Reputation of DBMS company.** Both Oracle and Sybase have excellent reputations for reliability and support. Both have had versions of their primary database products on the market for five years or more. Currently, Oracle is the marketplace leader; Sybase is its closest competitor.

2. **Performance.** An independent evaluation group known as the Transaction Processing Performance Council (TPC) performs benchmark testing on a wide variety of software and hardware, including DBMS software. Their benchmark specifications are made available to the public at no cost. The tests are conducted using identical and/or similar hardware, operating systems, and data

configurations. Data from these tests can be difficult to interpret; however, in most tests, Oracle appears to have surpassed Sybase's performance on similar hardware/OS configurations.[1]

3. **Concurrency control methods.** Oracle and Sybase have different concurrency control methods.

Sybase's most precise method is "page level locking". (A database "page" is a manageable area of the database (often 2048 bytes).) This means that only a page of the database can be locked at any given time. Thus, when one user updates a row in the database, the entire page is locked and other users are blocked from updating (and sometimes reading) rows in that page.

Oracle's most precise method is "row level locking". This means that multiple users can simultaneously update different rows on the same database page. Each user, when performing an operation on the row, only locks that row in question and does not interfere with other users in the same page. This method permits the highest degree of concurrency.

The nature of the CSMMOSS database (in particular, the scheduling-related data) requires row-level locking capabilities. In scheduling, event requestors and schedulers will most frequently be working on the same page of database data. Page level locking will result in concurrency problems for the scheduling application.

4. **ODBC compliance.** Both Oracle and Sybase provide ODBC-compliant database drivers.

5. **Ease of database administration.** Both products offer a full set of administrative tools and interfaces.

6. **Stored triggers and procedures.** Both Oracle and Sybase utilize stored triggers and procedures. Oracle offers more functionality in that its stored procedures can return all data types for queries, including arrays and sets. Sybase does not offer these features.

7. **Scalability.** Both Oracle and Sybase offer scalability options which can be obtained by purchasing additional software.

Sybase sells two basic database products:

1. PC Anywhere. This is Sybase's PC-based, standalone database package. It is a reworked version of the Powersoft Watcom database engine for 1 user only.

2. SQL Server 11.0. (evaluated herein) This is Sybase's enterprise-level database for 2+ users.

---

[1] According the the latest TPC benchmark results available (March 11-22, 1996).

Oracle sells three database products:

1. Personal Oracle 7. This is Oracle's equivalent of PC Anywhere.

2. Oracle Workgroup 2000. (evaluated herein) This product supports from 2 to +- 150 users.

3. Oracle 7.3 Enterprise Server. The Oracle product which supports 25+ users.

Oracle's advantage lies in compatibility. The same database kernel (basic set of software routines) is used in the Personal Oracle 7 product, the Workgroup 2000 product, and the Oracle 7 Enterprise Server product. This means that any size transition within the Oracle products (either up or down) will have the greatest likelihood of compatibility success.

A brief comparison of all features outlined is shown below:

	Criterion	Sybase	Oracle
1	Reputation	=	=
2	Performance	=	=+
3	Concurrency control methods	-	+
4	ODBC compliance	=	=
5	Ease of database administration	=	=
6	Stored triggers and procedures	=	+
7	Scalability	=	+

\+ Stronger performance; additional capabilities or features
= Acceptable performance, capabilities or features
\- Weaker performance; fewer capabilities or features

Summary

Both Oracle and Sybase offer excellent choices in terms of performance and general features. However, because of its superior concurrency control methods and scalability features, Oracle Workgroup 2000 was chosen as the recommended database for the CSMMOSS development and operation.

Initiated By: ____________________
M. L. Lotz, System Designer
ACSC Systems

Approved By: ____________________
L.T. Terrell, Manager
ACSC Project

Distribution: J. Eaton
D. Kaye
D. Latham
R. Mason
M. Purello
F. Robinson
D. Surran

LOCKHEED MARTIN

To: Distribution

From: M. L. Lotz
D. H. Kaye

Date: 19 December 1995

Martin Marietta Services, Inc.
a Lockheed Martin company
33531 Chincoteague Road
Wallops Island, VA 23337

Phone: (804) 824-7287/7092

ST095-071

Re: SELECTION OF CSPAS/MMOSS DEVELOPMENT SOFTWARE

Background

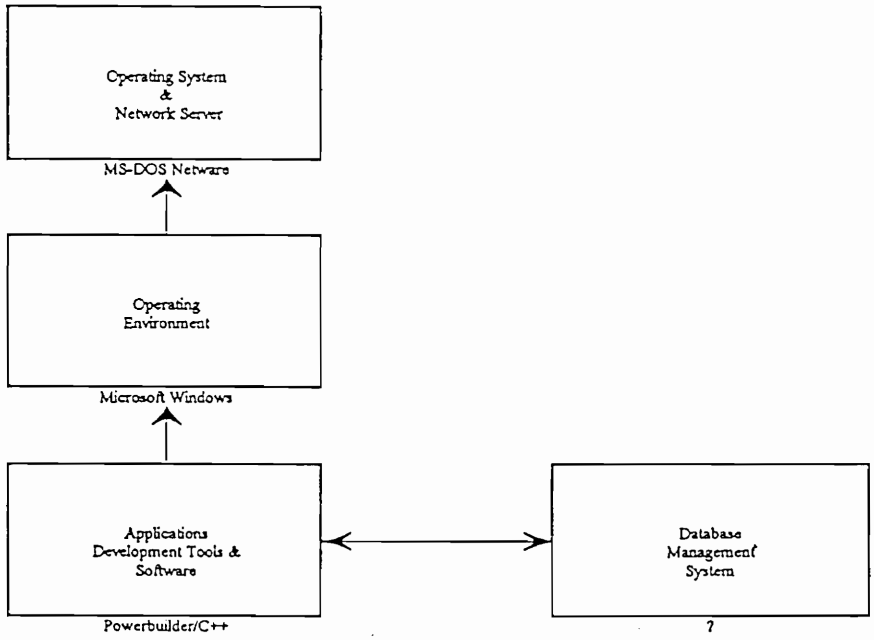
Development of software applications requires the consideration of several factors: (1) Operating environment, (2) purpose of the software to be developed, and (3) requirements of the software acquirer. Based upon the stated requirements for MMOSS and CSPAS, three basic design decisions were made:

1. Use of Microsoft Windows® operating environment. The recent installation of this operating environment on the Combat Systems LAN requires development of software which utilizes its capabilities.

2. Utilization of a database management system (DBMS). Most applications to be developed for CSPAS and MMOSS are database-oriented. The use of a DBMS will greatly enhance the speed and reliability of data retrievals. In addition, client/server capabilities can be more fully accessed through the use of a database system.

3. Use of a visual software development tool in addition to a standard programming language. Applications developed for Windows® require the creation and management of many visually-oriented objects. Use of a visual software development tool greatly facilitates this process by providing easier access to Windows® functions.

The proposed development environment is shown below:



<u>Selection Process</u>

Selection of the Microsoft Windows® operating environment was directed by the stated requirements of the government. Research is continuing for the selection of the database management system which will best support these projects. The choice of the proposed CSPAS and MMOSS development language has been made. This memorandum describes the selection process.

The following development products are currently available and were considered as part of the selection process:

Borland C++ 4.5
Borland dBase V
Borland Delphi 1.0
Borland Paradox 5.0
Borland Pascal
Borland Visual dBase
Computer Associates Clipper 5.3
Microsoft FoxPro 2.6
Microsoft Visual Basic
Powersoft PowerBuilder 4.0

The languages listed can be placed into three broad categories:

1. *General-purpose programming languages (GPLs).* These languages have been created to enable the production of almost any type of program (with the possible exception of parallel processing software). General purpose languages enable access to data, dialog with users, and calculations for nearly any problem. They are a step above assembler code in the programming language hierarchy. Because of this, they are ideal for direct control of lower-level software functions. Applications developed using GPLs generally contain large amounts of source code. Languages in the category include C++ and Pascal.
2. *PC-based database languages.* Languages of this variety are used for development of database applications for a single PC only. Software developed with these languages is created for local program execution, even within a networked environment. Data from these applications is not dynamically shared. Examples of PC-based database languages include dBASE V, Paradox, Visual dBASE, Clipper, and FoxPro.
3. *Client/server visual development languages.* These languages utilize an architecture which provides a method of using hardware and software together. Applications, data, and processing can be distributed between a centralized repository (the server) and all of the computers that access it (the clients). In addition, the visual component of these languages makes possible the rapid development of graphical user interface (GUI) applications (such as those in Microsoft Windows). The utilization of these advanced, special purpose languages greatly facilitates development of software for network connectivity, interfacing, database management and data sharing. Languages in this category include Visual Basic, Delphi, and PowerBuilder.

General programming languages were eliminated as candidates for a primary development language because of their deficiencies in the following areas:

1. *Maintainability.* Large source code sizes create difficulties with application maintenance and modification. Ease of maintainability is a stated requirement for the CSPAS and MMOSS systems.

2. *Development time*. Development using GPLs is very time intensive. The most minute operations must be completely designed and implemented by the developer.

3. *Graphical interface development*. Applications development of Windows programs is extremely complex, requiring the utilization of more than 800 separate programming functions in order to effectively interface with the Windows environment. GPLs provide little or no support for this type of development.

PC-based database languages were also eliminated as primary development tool candidates for the following reasons:

1. *Network environment*. These tools have no direct interface to the client-server environment. Processing is accomplished entirely on a single workstation; no distribution of processing can occur. Network operation is a stated requirement for the CSPAS and MMOSS systems.

2. *Data sharing*. Although data files can be opened and/or accessed in a networked environment, the databases used with these development languages do not utilize any performance features of the client/server environments. Data integration is also a stated requirement for these software systems.

3. *Functionality*. The general functionality of many PC-based languages is not as complete as either general purpose languages or higher-level development tools.

After elimination of these two categories of products, the only remaining option was a client/server visual development language. These development tools have been designed to specifically overcome the problems inherent in general purpose languages while utilizing the performance capabilities of a client/server environment. Preliminary research of industry publications showed that, in documented industry comparisons, PowerBuilder and Delphi consistently outperformed Visual Basic. This resulted in the elimination of Visual Basic from further consideration.

Because the remaining development tools investigated appeared to have similar capabilities regarding the criteria already cited, the following additional features were used as selection guidelines:

1) Platform optimized for large scale data file manipulation with structured query language (SQL) queries
2) Fully Open Data Base Connectivity (ODBC) compliant. ODBC is an industry standard for database drivers which allows connection to several different databases if desired.
3) Capability to export both database and spreadsheet files in a variety of formats
4) Speed of program execution
5) Object-oriented for ease of programming and support of the end product
6) Inheritance factors - maintainability and code size
7) Proven, widely-used, well-supported language
8) Team development capabilities
9) Industry-wide availability of knowledgeable programming personnel
10) Object linking and embedding (OLE) 2.0 Support

Our objective was to achieve a balance of execution speed, ease of use, file integrity, and system capabilities.

	Criterion	Delphi	PowerBuilder
1	Optimized for large scale SQL manipulation	=	=
2	Fully ODBC compliant	=	=
3	Capability to export data in a variety of formats	-	+
4	Speed of program execution	+	=
5	Object-oriented	=	=
6	Inheritance factors	-	+
7	Proven, widely-used, well-supported	-	+
8	Team development capability	=	=
9	Availability of personnel	-	=
10	OLE 2.0 support	=	=

+ Stronger performance; additional capabilities or features
= Acceptable performance, capabilities or features
- Weaker performance; fewer capabilities or features

Final determination of the programming tools selection was as follows:

Industry-wide, both Delphi and PowerBuilder are respected, widely-used development tools. PowerBuilder is a better-known, more mature development product. Although it is relatively new, Delphi is the product of Borland International, a company known for the quality of its software development products. Generally, Delphi and PowerBuilder are very similar in capabilities and performance; however, the nature of the CSPAS and MMOSS development projects indicated the need for specific database-oriented features for which PowerBuilder provides stronger support. For these reasons, PowerBuilder was chosen as the primary development tool, with Borland C++ as the secondary tool to provide support for any low-level programming functions which may become necessary during development.

Initiated By: ______________________
M.L. Lotz, System Designer
ACSC Systems

Initiated By: ______________________
D.H. Kaye, Sr. Engineer
ACSC Systems

Approved By: ______________________
L.T. Terrell, Manager
ACSC Project

Copy To:

J. Eaton
R. Mason
M. Purello
F. Robinson

Appendix D. Coding Standards and Naming Conventions.

Identifier Naming Conventions

Purpose: Identifier naming conventions are established to assist in the readability and maintainability of program source code.

Syntax Conventions

[] Single brackets indicate an identifier name element that is required.
[[]] Double brackets indicate an identifier name element that is optional.

The general format used will be: [[scope]] [[access]] [[prefix]] [[type]] [name]

The following tables provide the standard abbreviations for each identifier name element. The [[scope]] qualifiers are listed first, followed by the [[acess]] qualifiers. After these elements the required [prefixes] are listed. Finally the valid [type] abbreviations are listed.

Scope Qualifiers

Variable or Function	Scope Qualifier	Example
Argument	a	astr_new_author
Global	g	gstr_user_name
Instance	I	iprii_age
Local	l	ll_row
Shared/Class	s	sb_state

Access Qualifiers

NOTE: Unless otherwise specified, all access is assumed to be public.

Variable or Function	Access Qualifier	Example
Private	pri	ipri
Protected	pro	ipro

Public	[None]	

Prefixes

Class	Prefix	Example
Application	a_	a_defect_tracking_system
Data Window	d_	d_authors
Function	f_	f_get_symbol
Menu	m_	m_system
NonVisualObject	n_	n_book
OleObject	o_	o_calendar
Pipeline	p_	p_finance_refresh
Project	<None>	Defect_Tracking_System
Query	q_	q_delivery_codes
Structure	s_	s_person
UserObject	u_	u_security
Window	w_	w_m_library

Reference Variables	Prefix	Example
Application	ap_	ap_sam
Checkbox	cbx_	cbx_name
CommandButton	cb_	cb_ok
DataWindow	dw_	dw_users
DataWindowChild	dwc_	dwc_states
DragObject	drg_	drg_user
DrawObject	drw_	drw_line
DropDownListBox	ddlb_	ddlb_items
DynamicDescriptionArea	dda_	dda_sqlda

DynamicStagingArea	dsa_	dsa_sqlsa
EditMask	em_	em_start_date
Environment	env_	env_user1

Prefixes continued

Reference Variables	Prefix	Example
Error	err_	err_mask
Function	N.A.	
Graph	gr_	gr_user_data
GraphicObject	go_	go_mybox
HscrollBar	hsb_	hsb_my_window
Line	ln_	ln_1
ListBox	lb_	lb_names
MailFileDescription	mf_d	mfd_my_mail
MailMessage	mm_	mm_my_message
MailRecipient	mr_	mr_my_recipient
MailSession	ms_	ms_my_session
Menu	me_	me_main
Message	msg_	msg_msg_parm
MultiLineEdit	mle_	mle_names
NonVisualObject	nv_	nv_graph_constants
OleControl	oc_	oc_report1
OleObject	oo_	oo_report1
OleStorage	osto_	osto_report1
OleStream	ostr_	ostr_report1
Oval	oval_	oval_users
Picture	pic_	pic_my_picture

PictureButton	pb_	pb_exit
Pipeline	pl_	pl_add_users
PowerObject	po_	po_my_object
Query	N.A.	
RadioButton	rb_	rb_users
Rectangle	rec_	rec_my_rectangle

Prefixes continued

Reference Variable	**Prefix**	**Example**
RoundRectangle	rrec_	rrec_my_round_rectangle
SingleLineEdit	sle_	sle_names
StaticText	st_	st_title
Structure	str_	str_my_struct
Transaction	tr_	tr_my_trans
UserObject	uo_	uo_my_user_object
VscrollBar	vsb_	vsb_right_scroll_bar
Window	w_	w_my_window
WindowObject	wo_	wo_my_window_object

Standard Variables	**Prefix**	**Example**
Any	a_	a_current
Blob	blb_	blb_word_doc
Boolean	b_	b_is_selected
Character	ch_	ch_gender
Date	dt_	dt_payday
DateTime	dtm_	dt_logged_at
Decimal	dec#_	dec5_pi
Double	dbl_	dbl_result

Integer	i_	i_my_integer
Long	l_	l_my_long
Real	r_	r_my_real
String	s_	s_my_string
Time	tm_	tm_my_time
UnsignedInteger	ui_	ui_my_unsignedint
UnsignedLong	ul_	ul_my_unsignedlong

Prefixes continued

Function	**Prefix**	**Example**
Application Function	f_	f_my_function
Menu Function	mf_	mf_my_function
NonVisualObject Function	f_	f_my_function
UserObject Function	f_	f_my_function
Window Function	wf_	wf_my_function
Note: All non-global functions should be qualified by the class name or a pronoun.		

Event	**Prefix**	**Example**
User Defined	ue_	ue_cst_fileopen

Types

Windows Type	**Type Qualifier**	**Example**
Child	c_	wc_my_child_window
Main	m_	wm_my_main_window
MDI Frame	mdi_	wmdi_my_mdiframe
MDI Frame with MicroHelp	mdih_	wmdih_my_mdiframewhelp

Popup	p_	wp_my_popupwindow
Response	r_	wr_my_responsewindow

Types continued

Standard UserObject Type	**Prefix**	**Example**
Checkbox	cbx_	cbx_users
CommandButton	cb_	cb_ok
DataWindow	dw_	dw_events
DropDownListBox	ddlb_	ddlb_states
EditMask	em_	em_date
Graph	gr_	gr_mygraph
GroupBox	gb_	gb_allcontrols
HscrollBar	hsb_	hsb_test
Line	ln_	ln_test
ListBox	lb_	lb_add
MultiLineEdit	mle_	mle_test
OleControl	oc_	oc_report
Oval	oval_	oval_1
Picture	pic_	pic_name

PictureButton	pb_	pb_name
RadioButton	rb_	rb_choices
Rectangle	rec_	rec_myrec
RoundRectangle	rrec_	rrec_myrec
SingleLineEdit	sle_	sle_getnames
StaticText	st_	st_mytext
VscrollBar	vsb_	vsb_myscrollbar

Other UserObject Type	**Type Qualifier**	**Example**
Custom	cst_	cst_my_object
External	ext_	ext_my_object
VBX	vbx_	vbx_my_object
C++	cpp_	cpp_my_object

Types continued

Std NonVisualObject Type	Type Qualifier	Example
DynamicDescriptionArea	dda_	dda_sqlda
DynamicStagingArea	dsa_	dsa_sqlsa
Error	err_	err_errmsg
MailSession	ms_	ms_mysession
Message	msg_	msg_mymail
OleStorage	osto_	osto_myolestorage
OleStream	ostr_	ostr_myolestream
Pipeline	pl_	pl_mydatapipe
Transaction	tr_	tr_mytrans

Function Type
Note: The [[type]] element of an identifier name for a function should represent the simple data type of the function's return value. See the variable prefixes section for this list.

Event Type	Type Qualifier	Example
Custom	cst_	ue cst filesave
Windows Message	pbm_<windows msgname>	ue pbm char

Arguments	Type Qualifier	Example
Value	val_	ai val
Reference	ref_	acbx ref

Types continued

DataWindowObject Source	Type Qualifier	Example
SQL Select*	sql_	d_sql_employee
Quick Select*	sql_	d_sql_employee
Query*	sql_	d_sql_employee
External	ext_	d_ext_customer_upload
Stored Procedure	sp_	d_sp_orders
**These datawindows all become SQL select source after creation.*		

Presentation Type	Type Qualifier	Example
Composite	cmp_	cmp_mydata
CrossTab	crs_	crs_mycrosstab
FreeForm	ff_	ff_myfreeform
Graph	gph_	gph_mygraph
Grid	grd_	grd_mygrid
Group	grp_	grp_mygroup
Label	lbl_	lbl_mylabel
N-Up	nup_	nup_myn_up
Tabular	tab_	tab_mytabular

Indentation Style

The indentation style for control structures such as IF...THEN...ELSE statements will match the style of statements obtained by using the PasteStatement tool in the PowerScript Painter. An example of one of these structures follows:

```
IF <conditional> THEN
        <action1>
```

```
ELSEIF <condition2> THEN
      <action2>
ELSE
      <action3>
END IF
```

PowerBuilder keywords are capitalized like IF and THEN. Indentations are 2 characters or 1 tab character.

Documentation Headers

Two styles of comments are supported in PowerBuilder. Each may be used throughout the software development process as deemed necessary by development personnel. These styles are as follows: (1) use of the double-slash (//) marks at the beginning of the commented line, and (2) use of the slash-asterisk (/*) and asterisk-slash (*/) combinations to begin and end comment blocks.

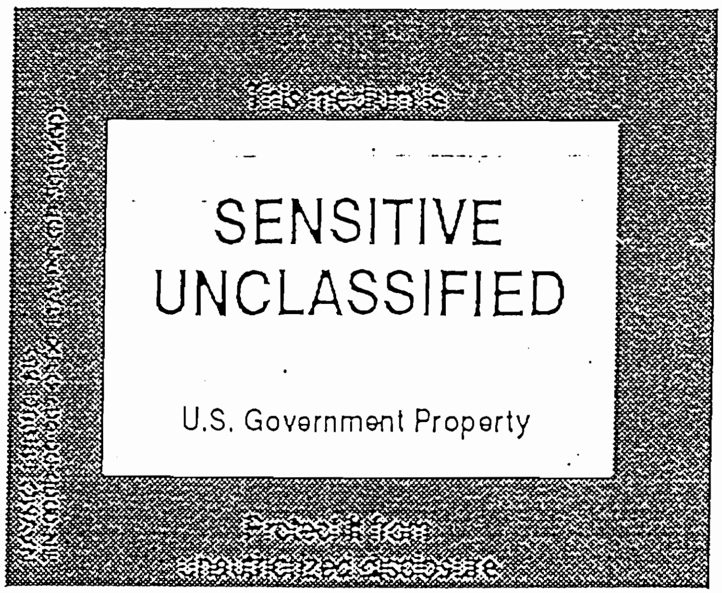
Functions and events

The documentation header for a function will describe its purpose, function and context in a description paragraph. It will also describe any preconditions that must be met before the function can be executed, input parameters, postconditions that must be true when the function has completed execution, and the value returned by the function.

Description:
Precondition:
Parameters:
Postcondition:
Returns:
Revision history:
Programmer   **Date**   **Comment**

The body of the function itself should be next.

Appendix E - Diskette label "SENSITIVE UNCLASSIFIED"



Appendix F - Analysis of Technical and Support Documentation Requirements

A. Programming Language Trade-Off Analysis. Development of software applications requires the consideration of several factors: (1) Operating environment, (2) purpose of the software to be developed, and (3) requirements of the software acquirer. Based upon these factors, a Programming Language Trade-Off will be required and delivered.

B. Advanced Summary Level Description. This document will provide a high level description of the contractor's understanding of the job. Areas to be addressed include (1) Overview of the software module functionality, (2) Analysis of tasking and documentation requirements, and (3) a proposed schedule for completion.

C. Technical Software Development Plan. This document provides information on how the contractor intends to get the job done. Areas to be addressed include (1) general and detained description of the software development process, (2) project planning and oversight, (3) software development environment, (4) configuration management, (5) quality assurance methods, (6) schedule, (7) integration of software onto the CSLAN, (8) documentation, and (9) required resources.

D. CSPAS Project Schedule. A Project Schedule for the CSPAS Module software development will be included in the Software Development Plan. It provides a major actions and milestones schedule from project beginning to CSPAS transition and implementation.

E. Operational Concept Description. This document describes the current processes, why they need to change, and the proposed changes which are to be implemented. Areas to be addressed include (but are not limited to) (1) What is the current system and process, (2) Description of proposed scenario, (3) What are the reasons for changing the current process? and (4) Description of the operating process with supporting flow charts.

F. CSPAS Functional Software System Specification. This document will detail function operations which are aligned with support applications for CSPAS actions, and will be traceable to the Operational Concept Description described above. This document will also provide detailed views of proposed screen presentations as well as a detailed description of the database structure and functionality.

G. CSPAS System Software Specification. This document defines the actual capabilities and requirements which are to be implemented in the new software. Areas to be addressed include (1) General software requirements, (2) System interfaces and computer resource requirements, (3) System quality factors, (4) Design and construction constraints, and (5) Qualification methods.

H. Programmer's Technical Support Document. On 30 September 1996, a programmer's technical support document will accompany the working CSPAS Module. Programming guidance, standards, and design and option decisions will be compiled for inclusion. A complete listing of programmer tools will be supplied. All commercial software used in the development of the module will be listed. Programmer hardware required for development and completion of the module will also be listed. Additional information provided in this document will include: a data dictionary, LAN interfacing tools, a source code master copy, and all programmers' notes.

I. CSPAS User Guides. A formal user guide will be prepared and delivered as part of this effort. The government will provide a format to follow, and a two-volume document will be written. Volume I will describe applications and capabilities of the CSPAS computer program. Volume II will more specifically address the needs of the internal combat systems analyst. Specific examples of the use of the CSPAS Module will be included (i.e., how to build your own chart).

J. Introduction of CSPAS onto LAN. A limited demonstration of CSPAS Module applications on the LAN will be held for review purposes.

K. Implementation (Introduction) and User Training Plan. A preliminary copy of this document will be submitted approximately 30 days prior to completion of the project, in order to allow time for revisions and comments by ACSC Code 2400. The document will describe the content and location of the training to be held for users of the new application as well as how the new system will be implemented with regards to:

- software and database installation, data conversion, and setup activities
- hardware requirements and method to implement with minimum impact to the existing system and users
- transition plan

A final implementation document will accompany the CSPAS Module.

L. Monthly In-Process Briefs. During CSPAS development, in-process briefs (IPBs) will be held at least monthly. As IPBs are required, locations and times are organized and scheduled by the contractor. Minutes of all IPBs shall be the responsibility of the contractor. The meeting format will be as follows:

Presenters: System Designer and Chief Engineer

Participants: Representatives from ACSC Code 2000 and Technical Point of Contact, Code 2400

Agenda:
- Action items from previous meeting
- Status of CSPAS Module development
- Problems/solutions as needed
- Actions items for next meeting

M. Mid-Task Review Brief. This meeting shall be held as a formal review. The status of the CSPAS Module development effort and computer program specifications will be presented. The System Designer, having analyzed the preliminary system software specifications, will determine, through updated quality assurance methods, whether the computer program requirements are consistent and complete. Other items to be addressed at this meeting are:

- Satisfaction with the progress of the project
- Indications of future problems
- Comparison of original software development plan with development course
- Plans for next phases of development course
- LAN upgrade

N. Test and Acceptance Plans. A CSPAS Module Test and Acceptance Plan will be developed and written for the new assessment program. Items to be included in the CSPAS Module Test and Acceptance Plans are (1) Traceable to individual software configuration items and specification requirements, (2) Procedures to test individual items, (3) Quality Assurance Methods and forms, (4) Schedule Function Testing on the LAN, (5) Test Analyses Methods, and (6) Acceptance Criteria CSLAN demonstration.

O. Final Technical Report. A Final Technical Report will be submitted with the CSPAS Module and will include:

- Summary of All Task Events
- Optional Design Decisions
- Specification Requirements which were not met
- Technical Recommendations for future improvements
- Copies of IPB/IPR Minutes
- Progress Development Reports
- Quality Assurance Reports

- Acceptance Test discrepancies and deficiencies with corrective action

The Final Technical Report will accompany:

- Programmer's Technical Support Document
- CSPAS User Guide
- Implementation and User Training Plan
- Final Quality Assurance Report
- CSPAS Module

P. LAN upgrade materials/equipments Listings. A listing of LAN modifications and additional equipment will be supplied by the contractor.

Q. CSPAS Quality Assurance Report. Quality Assurance Evaluation Reports shall be detailed in the CSPAS Software Development Plan and submitted with the Final Technical Report.

S. Final Review Brief. The Final CSPAS Review Brief will be held in accordance with the on-line CSPAS Module demonstration. The location and time of the review will be organized and scheduled by the contractor. Minutes of the Final Review Brief will be the responsibility of the contractor. Agenda items will include:

- Final action items
- Summary of task events
- Successes/problems/solutions
- Comparison of CSPAS program development plan with course
- Demonstration of CSPAS computer program

T. CSPAS Software Installation on the CSLAN. The CSPAS Module will be installed on the CSLAN in fully operational mode ready for ACSC use at project completion.

Appendix G. CSU Code Inspection and Test Procedures.

CSU Code Inspection and Test Procedures

1. Configuration ID	**CSCI: CSU: Ver: Rev: Rel: Type:** **code / dbf / bmp**
2. CSU Descriptive Name	
3. CSU object/filename Include application library location (if list of objects, include all names)	

4. General description of CSU item:	
5. Expected input:	
6. Required output/operation:	
7. Referenced functions or objects: (include application library location)	

8.	**Inspection questions**	**Yes**	**No**	**N/A**	**Corrected**
a.	Any unused input arguments. Any output argument not used?				
b.	Any data types incorrect or inconsistent?				
c.	Are any local, instance, or shared variables used before initialization?				
d.	Are any external functions improperly declared or missing declarations? Are they declared at the incorrect level?				
e.	For data entry operations, are any items lacking proper validation?				
f.	For data entry operations, is the user prompt for save, cancel, or exit when data is modified or missing?				
g.	Are any controls not keyed with accelerator key sequences?				
h.	Do any accelerator keys not follow development group standard?				
i.	For a reusable object, are any constraints in place which will adversely affect reusability?				
j.	Are non-standard error messages provided? If so, are there any unclear messages provided to the user?				

k.	Are any sizing or placement constraints inconsistent with development group standards?				
l.	Are any naming conventions not used properly?				
m.	Is the file/object name unclear or assigned to an improper development library?				
n.	Are comments (internal and./or library level) unclear or missing?				

CSU Code Inspection and Test Procedures

9. Testing Procedure:	
a. Describe testing procedure used for this item (by programmer).	
b. Were any tests (as initially performed by the programmer) unable to be replicated by inspector and verified as accurate? (If yes, state inconsistent results) Yes No	
c. Is any expected output or action inconsistent with requirement as stated above? (If yes, state inconsistent results) Yes No	

10. Inspection/Testing Results:	
a. Programmer Name:	**Date:**
b. Inspector name:	**Date:**
c. If all items above are checked "NO" or N/A, the code unit passes. Inspector signs here to indicate completion or checks here to indicate corrective action is needed	**d. Corrective action required (if any)**

Instructions for completing the CSU Code Inspection and Test Procedures Form

Items 1 through 7 are to be completed by the programmer and verified by the code inspector.

1. Enter the configuration identification of the applicable computer software unit (CSU).

2. Enter the descriptive name of the applicable CSU.

3. Enter the CSU object/filename. Include the directory name, application library name (if applicable), and object name.

4. Provide a general description of the CSU item.

5. Enter the expected input or entry actions into this CSU.

6. Enter the required output and/or operation of this CSU.

7. Add any functions and/or objects which this CSU references. Include the application library location (including directory) of any referenced functions and/or objects.

Item 8 is to be completed by the code inspector.

8. For each lettered item in this section, check "Yes", "No", or "N/A" as appropriate. If this inspection is the result of corrective action, check "Corrected" as applicable. Each lettered item must be completed.

Items 9 and 10 are to be completed by the programmer and the code inspector.

9. a. The programmer provides a description of the testing procedure(s) (including results) used for this CSU.

   b. The code inspector attempts to replicate the testing procedures as described in item **a** and completes item **b** as appropriate.

   c. The code inspector evaluates the results of the testing procedures performed in item **b** and completes item **c** as appropriate.

10. The programmer and code inspector enter their names and the current date in the appropriate areas of this item. The code inspector indicates in item **c** whether the CSU has passed inspection by signing the inspection form. If any area of the inspection has failed, item **c** is checked to so indicate. Any further comments regarding required corrective action are indicated in item **d**.

Appendix H. CSCI Code Inspection and Test Procedures

CSCI Code Inspection and Test Procedures

1. Configuration ID	**CSCI: CSU: Ver: Rev: Rel: Type: code / dbf / bmp**
2. CSCI Descriptive Name	
3. CSCI object/filename Include application library location (if list of objects, include all names)	

4. General description of CSCI item:	
5. CSUs to be included in this CSCI (list all)	
6. Required output/operation:	
7. Referenced functions or objects: (include application library location)	

8.	**Inspection questions**	**Yes**	**No**	**N/A**	**Corrected**
a.	Any unused input arguments. Any output argument not used?				
b.	Any data types incorrect or inconsistent?				
c.	Are any local, instance, or shared variables used before initialization?				
d.	Are any external functions improperly declared or missing declarations? Are they declared at the incorrect level?				
e.	For data entry operations, are any items lacking proper validation?				
f.	For data entry operations, is the user prompt for save, cancel, or exit when data is modified missing?				
g.	Are any controls not keyed with accelerator key sequences?				
h.	Do any accelerator keys not follow development group standard?				
i.	For a reusable object, are any constraints in place which will adversely affect reusability?				
j.	Are non-standard error messages provided? If so, are there any unclear messages provided to the user?				

k.	Are any sizing or placement constraints inconsistent with development group standards?				
l.	Are any naming conventions not used properly?				
m.	Is the file/object name unclear or assigned to an improper development library?				
n.	Are comments (internal and./or library level) unclear or missing?				

CSCI Code Inspection and Test Procedures

9. Testing Procedure:	
a. Describe testing procedure used for this item (by programmer).	
b. Were any tests (as initially performed by the programmer) unable to be replicated by inspector and verified as accurate? (If yes, state inconsistent results) Yes No	
c. Is any expected output or action inconsistent with requirement as stated above? (If yes, state inconsistent results) Yes No	

10. Inspection/Testing Results:	
a. Programmer Name:	**Date:**
b. Inspector name:	**Date:**
c. If all items above are checked "NO" or N/A, the code unit passes. Inspector signs here to indicate completion or checks here to indicate corrective action is needed	

Instructions for completing the CSCI Code Inspection and Test Procedures Form

Items 1 through 7 are to be completed by the programmer and verified by the code inspector.

1. Enter the number of the applicable computer software configuration item (CSCI).

2. Enter the descriptive name of the applicable CSCI.

3. Enter all CSU object/filename for CSUs included in this CSCI. Include the directory name, application library name (if applicable), and object name of each CSU.

4. Provide a general description of the CSCI item.

5. Enter the expected input or entry actions into this CSCI.

6. Enter the required output and/or operation of this CSCI.

7. Add any functions and/or objects which this CSCI references. Include the application library location (including directory) of any referenced functions and/or objects.

Item 8 is to be completed by the code inspector.

8. For each lettered item in this section, check "Yes", "No", or "N/A" as appropriate. If this inspection is the result of corrective action, check "Corrected" as applicable. Each lettered item must be completed.

Items 9 and 10 are to be completed by the programmer and the code inspector.

9. **a.** The programmer provides a description of the testing procedure(s) (including results) used for this CSCI.

b. The code inspector attempts to replicate the testing procedures as described in item **a** and completes item **b** as appropriate.

c. The code inspector evaluates the results of the testing procedures performed in item **b** and completes item **c** as appropriate.

10. The programmer and code inspector enter their names and the current date in the appropriate areas of this item. The code inspector indicates in item **c** whether the CSCI has passed inspection by signing the inspection form. If any area of the inspection has failed, item **c** is checked to so indicate. Any further comments regarding required corrective action are indicated in item **d.**

Appendix I - ACSC Non-tactical Computer Program Change Request

ACSC NONTACTICAL COMPUTER PROGRAM CHANGE REQUEST

Request for Program Change (CPCR)

Computer Program Identification: ____________________

Description of Problem or Change: ____________________

____

____

____

Justification: ____________________

New Report Format: ☐ Attached ☐ Not Required

Date Required: __________ Priority: 1 2 3 4 5

Critical Routine

Originator: __________ Code: __________ Date: __________ Ext: __________

POC: __________ Code: __________ Date: __________ Ext: __________

For Technical Evaluation

AIS Manager: ____________________ Date: __________

TECHNICAL EVALUATION

Impact on Program: ☐ Major ☐ Minor ☐ None

Description: ____________________

____

____

Requires a version update? ☐ Yes ☐ No Current Version: __________

Impact on Documentation: ☐ Major ☐ Minor ☐ None

Description: ____________________

Other Impacts: ____________________

____

Estimated Scope of Effort:

System Analyst/Engineer Hours __________
Computer Programmer Hours __________
Manual Update Hours __________
Other Cost __________
Total Estimated Cost __________

Reviewer: ____________________ Code\Activity: __________ Date: __________

ACSC Form 5231/3 (1/95)

Change Authorization

☐ Approved ☐ Disapproved

Comments: ____

____

____

Authorization: ____

AIS Manager Code Date

Record of Completion

	Yes	No	Not Required
Program Modified	☐	☐	☐
Source Code Listing Updated	☐	☐	☐
Specification/Requirement Updated	☐	☐	☐
User Guide/Manual Updated	☐	☐	☐
Other (Describe Below)	☐	☐	☐

New Version No. ______________

Comments: ____

____

____

Completed by: ______________________________ Date: ______________

Installation Verified by: ______________________________ Date: ______________

Accepted by POC: ______________________________ Date: ______________

Notes:

Appendix J. Quality Assurance Evaluation Plan

This plan contains full details of the Quality Assurance Evaluation Reports (QAERs) which were developed, implemented and archived during CSPAS development and testing. Full details of the QAER process and instructions for completing reports follows:

Quality Assurance Evaluation Reports (QAERs).

Quality assurance evaluations ensure that each activity described in the Software Development Plan was performed as specified therein. In addition, periodic evaluations were conducted to assure that each software product required remained intact in the appropriate software development or archive library. Software quality assurance was performed by the Sr. System Engineer and recorded on a Quality Assurance Evaluation Report (QAER) by Quality Assurance personnel after the Archives Librarian determined the type of evaluation to perform. The personnel members who performed quality assurance and evaluation reporting were outside of the immediate software development group. They were chosen for their experience and objectiveness in computer program development, documentation, and quality assurance.

There were three types of quality assurance evaluation reports:

- **Documentation evaluations:** For this type of evaluation, the evaluator compared the schedule of deliverables as specified in the Advanced Summary Level Description with the status of those items, and noted whether they were in compliance with the requirements.

- **Software evaluations:** The evaluator determined that uniform conventions were being employed for directories and subdirectories, and that naming conventions for files and subdirectories were consistent within each computer program development item. The evaluator performed code walkthroughs and ensured that all code inspection forms were properly completed for all software which has been placed under configuration control.

- **Library evaluations:** The evaluator determined that the Archives Library contains the most recent and authorized version of all required documentation materials, including Technical Support Documentation in the form of Programmers' Notes, as well as all briefing and review Minutes. The evaluator evaluated the Configuration Control Library for recovery and backup capabilities, naming conventions and identification consistencies.

The Quality Assurance Evaluations took place according to the following formal procedures and instructions:

A. Preparation for software evaluation. The appropriate QA (quality assurance) item will be determined for each evaluation from the numbered QA Evaluation Item List below:

QA EVALUATION ITEM LIST

1. Monthly evaluations. The following items will be evaluated on a monthly basis:

QA-1. Computer program objectives and requirements. This evaluation item will document that completed portions of the CSPAS software meet program objectives and requirements in accordance with the Software System Specification as specified in TI 9014.

QA-2. Computer program quality. This evaluation will verify that a quality program activity is taking place, and that corrective action will be initiated when an area needs improvement.

QA-3. Computer program records. This evaluation item will document that programmers' notes are complete and up to date, and properly filed in the Archives Library. This evaluation will begin after completion of the Preliminary Design Review.

QA-4. Code evaluations. The monthly code evaluation process will ensure that completed CSUs have been properly documented and that a CSU Code Inspection and Test Procedure Form has been completed for each CSU.

QA-5. Documentation evaluations. The evaluation of documentation will assure that document deliverables are in keeping with the schedule as stipulated in the CSPAS Advanced Summary Level Description and the requirements of TI 9014.

QA-6. Development library evaluations. The development library evaluations will begin after completion of the Preliminary Design Review, and will continue through CSPAS delivery to assure that the Archives Library, Application Library, and the Configuration Control Library adhere to the Software Development Plan (SDP). These evaluations will also assure that the most recent and authorized version of materials under configuration control are clearly identified, and that archiving, recovery, and backup capabilities are up to date.

QA-7. In-Process Review (IPR) evaluations. As IPRs are held and deliverable schedules are met during the software system development phases, the evaluator will review IPR Minutes and action items accordingly on a monthly basis. The evaluator will document that requirements as described in the Advanced Summary Level Description have been/not been met.

2. Mid-task evaluations. The following evaluations will take place at mid-task:

QA-8. Development process evaluations. This item will evaluate the software development process. It will also evaluate the quality assurance activities by LM personnel for internal unit CSU and CSCI testing as implemented through the completion of the CSU and CSCI Code Inspection and Test Procedures Forms (see Appendices C and D). A code walkthrough will be performed by the evaluator to assure that the process complies with the CSPAS Software System Specification.

3. Evaluations just prior to program delivery. The following quality assurance activities will take place just before delivery of the completed software and documentation:

QA-9. Electronic media evaluation. Prior to the preshipping evaluation below, electronic media will be inspected to assure that all have been properly identified and that configuration control standards have been met.

QA-10. Preshipping evaluation. This will be performed just prior to acquirer acceptance. The procedure will ensure that preparations for computer program and documentation delivery have complied with CSPAS project requirements as set forth in TI 9014.

4. As-required evaluations. These quality assurance evaluations will be accomplished on an as-required basis, determined by the software development schedule and completed tasks.

QA-11. Computer program corrective action. This evaluation item will assure that timely and positive corrective actions will be taken when they are required. It will help to define immediate corrective action needs (such as critical process noncompliance), as opposed to timely ones (such as typographical errors). It will also aid in timely responses to CPCRs.

QA-12. Design evaluations. This item will assure that management design reviews as described in paragraph 5.1.6 have been held as scheduled. The evaluation will be scheduled accordingly. It will document that reports of reviews have been completed, that they are on file in the Archives Library, and that design changes have been specified and incorporated in project documentation.

B. Performance of evaluation. After the preparation for evaluation has been completed, the evaluation will be performed. The steps to be performed for QA evaluation follows:

a. If a QAER already exists from a previous quality assurance evaluation, it will be used to aid the performance of ongoing evaluations.

b. After the item to be evaluated has been determined, the evaluator will be notified, and the evaluation will be scheduled.

c. The evaluator will complete the QAER as per the instructions.

d. The evaluator will coordinate with the appropriate personnel, resolutions for convention conflicts and documentation status conflicts.

e. Any corrective action required will be noted, with an anticipated date for completion.

f. A Computer Program Change Request (CPCR) will be initiated.

g. If there is disagreement between the evaluator and the individual responsible for the item evaluated, the issue will be escalated to LM software development management.

h. The completed QAER will be given an Archives Library File Number and will be filed in the Archives Library for later inclusion in the Final Technical Report.

A blank QAER form and instructions on how to complete it follows.

QUALITY ASSURANCE EVALUATION REPORT (QAER)

PROJECT	

REASON FOR EVALUATION	MONTHLY	MID-TASK	PRIOR	AS REQ'D	DATE

ITEM #/TYPE

QA-1	QA-2	QA-3	QA-4	QA-5	QA-6	QA-7	QA-8	QA-9	QA-10	QA-11	QA-12
Program Objectives	Program Quality	Program Records	Codes	Document	Library	IPR	Develop Process	Electronic Media	Pre-shipping	Correct Action	Design

EVALUATION SUMMARY:

CORRECTIVE ACTION REQUIRED:

CORRECTIVE ACTION DATE OF COMPLETION:

EVALUATOR NAME:

EVALUATOR SIGNATURE:

DATE: ___________________

ADDITIONAL COMMENTS:

ARCHIVES LIBRARY FILE

NO.___________

MEETS REQ.	DOES NOT MEET REQ	CORRECTIVE ACTION REQ
		YES NO

Instructions for Completion of the QAER

1. Insert Project name, date, and reason for evaluation; whether monthly, mid-task, prior to delivery, or as-required.

2. Circle the appropriate Item # and Type as determined from the QA Evaluation Item List.

3. Briefly describe the evaluation which took place in the space provided. (If more space is needed, use the Additional Comments section.)

4. Briefly describe the corrective action required. If no corrective action is required, then write the word NONE in the space provided.

5. If a corrective action is required, estimate the completion date and write that date in the space provided.

6. The evaluator must print his/her full name in the space provided.

7. The evaluator must sign his/her full name in the space provided.

8. Fill in the date according to the order day/month/year.

9. If it has been determined that the evaluated item meets all of the requirements, place a checkmark in the appropriate box. If it has been determined that the evaluated item does not meet all of the requirements, place a checkmark in the appropriate box.

10. If it has been determined that a corrective action is required, circle the word YES. If no corrective action is required, circle the word NO in the appropriate box.

11. The completed QAER will be given an Archives Library file no.

Appendix K - Referenced Documents.

ACSCINST 3121
ACSCINST 3967.1B
ACSCINST 5234.2 Non-Tactical Computer Program Development and Maintenance
MIL-STD-498, DI-IPSC-81427 Software Development Plan
CSPAS Software System Specification
CSMMOSS Preliminary Software Development Plans
CSPAS Functional Software System Specification
CSPAS Advanced Summary Level Description
CSPAS Final Technical Report
CSLAN Upgrade/Equipment List
CSPAS User Training and Implementation Plan
CSPAS Test and Acceptance Plan
CSMMOSS Operational Concept Description
CSPAS Programmers' Notes
CSPAS User Training Guides
AEGIS Technical Instruction No. 9014
Blanchard, B.S., and W.J. Fabrycky, Systems Engineering and Analysis (2nd edition), Prentice Hall, Inc. Englewood Cliffs, NJ, 1990.
Hunger, J., Engineering the System Solution by Jack Hunger, Prentice Hall PTR, Englewood Cliffs, NJ, 1995
Kasser, J., Applying Total Quality Management to Systems Engineering, Artech House, Boston, MA, 1995
Whitten, N., Managing Software Development Projects (2nd Edition), John Wiley & Sons, Inc., New York, NY, 1995
SEL-84-101, Manager's Handbook for Software Development, NASA, November 1990
SP-6105, NASA Systems Engineering Handbook, NASA 1995

Appendix L - List of Acronyms.

ACEDS	AEGIS Centralized Equipment Data System
ACS	AEGIS Combat Systems
ACSC	AEGIS Combat Systems Center
ACSCINST	AEGIS Combat Systems Center Instruction
ADAR	AEGIS Data Reduction
ADP	Automatic Data Processing
AFDMS2	AEGIS Facilities Documentation Management System Two
AIS	Automated Information Systems
ASCII	American Standard Code for Information Interchange
ATC	AEGIS Training Center
ATU	AEGIS Training Unit
ATEAM	AEGIS Training and Engineering Accessibility Model
ATLAS	ACSC Technical Library Automated System
AWAN	ACSC-Wide Area Network
AWS	AEGIS Weapon System
BQ	Bachelor Quarters
BRAC	Base Realignment and Closure
CCB	Change Control Board
CDR	Critical Design Review
CD-ROM	Compact disk - read-only memory
CIC	Combat Information Center
CLEP	CSOOW Logbook and Equipment Program
COSAL	Coordinated Shipboard Allowance List
COSBAL	Coordinated Shore Base Allowance List
COTS	Commercial off the shelf
CPCR	Computer Program Change Request
CPU	Central Processing Unit
CSCI	Computer Software Configuration Item
CSEDS	Combat Systems Engineering Development Site
CSLAN	Combat Systems Local Area Network
CSMMOSS	Combat Systems Media, Maintenance, and Operations Scheduling System
CSOOW	Combat Systems Officer of the Watch
CSPAS	Combat Systems Performance Assessment System
CSU	Computer Software Unit
DBA	Data Base Administrator
DBMS	Data Base Management System
ECP	Engineering Change Proposal

FACSFAC	Fleet Area Control and Surveillance Facility
HWCI	Hardware Configuration Item
IPB	In-process Briefs
IPR	In-progress Review
ISE	In Service Engineering
LAN	Local Area Network
LIMS	Library Inventory Management System
LM	Lockheed Martin
LSE	Lifetime Support Engineering
MB	Mega Byte
MRQ	Maintenance Request Query
MS-DOS	Microsoft Disk Operating System
NMRM	Network Maintenance Request Module
NSWC	Naval Surface Warfare Center
OARS	Open Architecture Retrieval System
OCC	Operational Control Center
OCD	Operational Concept Description
OP	Operation
OPAREA	Operational Area
ORDALTs	Ordnance Alterations
ORR	Operational Readiness Review
PC	Personal Computer
PDR	Preliminary design Review
PMS	Preventative Maintenance System
POC	Point of Contact
PQA	Presidential Quality Award
QAER	Quality Assurance Evaluation Reports
RAM	Random Access Memory
SAM	Scheduling Activity Module
SCIN	Site Change Implementation Notice
SDP	Software Development Plan
SNAP	Shipboard Non-Tactical Automatic Data Processing II System
SVGA	Super VGA
TI	Technical Instruction
TPOC	Technical Point of Contact
VACAPES	Virginia Capes
VGA	Video Graphics Array
WIMS	Wallops Island Mission Support
WIOA	Wallops Island Office Automation
WISNAP	Wallops Island SNAP

Vita

Michael Purello is a Systems Engineer at AEGIS Combat Systems Center, Wallops Island, VA. where he is the Engineering Analysis Division Manager. Prior to AEGIS Combat Systems Center, he worked at the Norfolk Naval Shipyard, Portsmouth, VA and in the food manufacturing and service industries. He is a registered Professional Engineer in the Commonwealth of Virginia and holds a B.S. in Industrial Engineering from the New Jersey Institute of Technology, a Masters in Business Administration from Chaminade University of Honolulu, HI and a Masters in Systems Engineering from Virginia Tech.