A Study of the Operation and Maintenance of Computer Systems to Meet the Requirements of 10 C.F.R. 73.55

Prepared by J. R. Lewis, K. R. Byers, J. D. Fluckiger, K. C. McBride

Pacific Northwest Laboratory
Operated by
Battelle Memorial Institute

Prepared for
U.S. Nuclear Regulatory
Commission

NOTICE

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, or any of their employees, makes any warranty, expressed or implied, or assumes any legal liability of responsibility for any third party's use, or the results of such use, of any information, apparatus, product or process disclosed in this report, or represents that its use by such third party would not infringe privately owned rights.

NOTICE

Availability of Reference Materials Cited in NRC Publications

Most documents cited in NRC publications will be available from one of the following sources:

1. The NRC Public Document Room, 1717 H Street, N.W.
Washington, DC 20555
2. The Superintendent of Documents, U.S. Government Printing Office, Post Office Box 37082,
Washington, DC 20013-7082
3. The National Technical Information Service, Springfield, VA 22161

Although the listing that follows represents the majority of documents cited in NRC publications, it is not intended to be exhaustive.

Referenced documents available for inspection and copying for a fee from the NRC Public Document Room include NRC correspondence and internal NRC memoranda; NRC Office of Inspection and Enforcement bulletins, circulars, information notices, inspection and investigation notices; Licensee Event Reports; vendor reports and correspondence; Commission papers; and applicant and licensee documents and correspondence.

The following documents in the NUREG series are available for purchase from the GPO Sales Program: formal NRC staff and contractor reports, NRC-sponsored conference proceedings, and NRC booklets and brochures. Also available are Regulatory Guides, NRC regulations in the *Code of Federal Regulations*, and *Nuclear Regulatory Commission Issuances*.

Documents available from the National Technical Information Service include NUREG series reports and technical reports prepared by other federal agencies and reports prepared by the Atomic Energy Commission, forerunner agency to the Nuclear Regulatory Commission.

Documents available from public and special technical libraries include all open literature items, such as books, journal and periodical articles, and transactions. *Federal Register* notices, federal and state legislation, and congressional reports can usually be obtained from these libraries.

Documents such as theses, dissertations, foreign reports and translations, and non-NRC conference proceedings are available for purchase from the organization sponsoring the publication cited.

Single copies of NRC draft reports are available free, to the extent of supply, upon written request to the Division of Technical Information and Document Control, U.S. Nuclear Regulatory Commission, Washington, DC 20555.

Copies of industry codes and standards used in a substantive manner in the NRC regulatory process are maintained at the NRC Library, 7920 Norfolk Avenue, Bethesda, Maryland, and are available there for reference use by the public. Codes and standards are usually copyrighted and may be purchased from the originating organization or, if they are American National Standards, from the American National Standards Institute, 1430 Broadway, New York, NY 10018.

A Study of the Operation and Maintenance of Computer Systems to Meet the Requirements of 10 C.F.R. 73.55

Manuscript Completed: November 1985
Date Published: January 1986

Prepared by
J. R. Lewis, K. R. Byers, J. D. Fluckiger, K. C. McBride

Pacific Northwest Laboratory
Richland, WA 99352

Prepared for
Division of Risk Analysis and Operations
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, D.C. 20555
NRC FIN B2877

ABSTRACT

The Pacific Northwest Laboratory has studied the operation and maintenance of computer-managed systems that can help nuclear power plant licensees to meet the physical security requirements of 10 CFR 73.55 (for access control, alarm monitoring, and alarm recording). This report of that study describes a computer system quality assurance program that is based on a system of related internal controls. A discussion of computer system evaluation includes verification and validation mechanisms for assuring that requirements are stated and that the product fulfills these requirements. Finally, the report describes operator and security awareness training and a computer system preventive maintenance program.

SUMMARY

The Pacific Northwest Laboratory (PNL) has studied the design, installation, operation, and maintenance of computer-managed physical access security systems at commercial nuclear power plants (NPPs). The purpose of this work was to provide information for licensees who are planning to use or upgrade such a computerized system to meet the security requirements of 10 CFR 73.55 (access control, alarm monitoring, and alarm recording) and 10 CFR 73.70 (reporting and logging). The information can also be used to assist NRC license reviewers and inspectors in their evaluation of a licensee's computer-managed system.

A previous report, NUREG/CR-4298, concentrated on the design, selection, and installation of the computer systems.

Once a system has been installed, a great deal of effort must yet be expended to ensure its proper operation and maintenance. The literature was reviewed and sites were visited to obtain information in these areas involved in the operation and maintenance of a system: quality assurance, system evaluation, operator training, and preventive maintenance.

COMPUTER SYSTEM QUALITY ASSURANCE PROGRAM

A computer system quality assurance program is based on a system of related internal controls: management controls, documentation guidelines, configuration management, and operation controls. Internal control comprises the plan of organization and all the methods and procedures to safeguard data and system integrity and to encourage adherence to prescribed management policies.

Management control embraces the entire system of organization, policies, procedures, and practices employed in managing affairs and promoting the execution of assigned responsibilities. The specific techniques employed to exercise control over security system operations and computer resources vary depending on the size, type, complexity, and geographic location of the facility.

Management should exercise control over the computer system to determine how well it operates, where improvements are needed, and what capabilities will be needed in the future. To accomplish these objectives, control techniques should be considered in the following areas: organization and planning; policies, standards, and procedures; personnel; reporting; and internal auditing.

Documentation contributes to the successful operation and maintenance of a system. Documentation should not be approached as a task occurring only after the system has been developed and implemented. Good documentation practices must be followed at every step of the system life cycle. Specific documentation necessary for the operations phase of a system includes a user's manual, test plans, a program maintenance manual, and configuration management procedures.

Configuration management is "the discipline of identifying the configuration of a system at discrete points in time for purposes of systematically controlling changes to this configuration and maintaining the integrity and traceability of this configuration throughout the system life cycle." (Bersoff et al. 1979). Software configuration management has been further described as a "housekeeping" function established to assure that an orderly approach is taken during all phases of the software life cycle, that all changes are traceable and have proper approval, and that administrative requirements are met (Doggett 1983).

The objectives of configuration management are to:

- improve the reliability of a system
- decrease total costs over the life cycle of a project
- provide a method of determining the state (configuration) of a system for any time during the system life cycle
- provide traceability and therefore responsibility
- assure that user requirements have been met
- approve and monitor all changes to the system
- support the functions of auditing, quality assurance, and verification and validation.

Operations control involves the security of the computer system as well as the central alarm station/secondary alarm station (CAS/SAS). Operations control involves: physical security controls; hardware controls; system software controls; data input, processing, and output controls; auditing; problem reporting and preventive maintenance; contingency plans; and personnel.

Physical security controls for a computer system are those procedures and equipment installed to assure that only authorized individuals gain physical access to the computer equipment or facility. Hardware controls involve preventing unauthorized use of the devices themselves, corruption of data in the devices, or interruption of the equipment's use. System software controls are used so that actual machine processing is not affected by outside influences that might corrupt or destroy data, or that might cause improper operation of the computer system.

COMPUTER SYSTEM EVALUATION

Verification and validation (V&V) are processes that assure that the system requirements are stated and that the product fulfills these requirements. Mechanisms used to verify and validate software products include: reviews, walk-throughs, inspections, and testing. V&V should be applied throughout the life cycle of a software system. If a utility is purchasing a complete system from a vendor, the major V&V concern is

acceptance testing of the system. This will involve a review of the problem statement, requirements specification, design, test plan, and test results.

TRAINING

An integral part of the operation of a computer-managed physical access security system is the training provided to the operators of the system. Four steps must be completed to provide a comprehensive training program: reality search, research, materials preparation, and testing and evaluation. Successful training provides reasonable assurance that decisions and actions of the security personnel during normal and abnormal conditions will be such that the plant is protected from the design basis threat. For operators of the CAS/SAS this would involve specific training in: security officer training, decision making, computer operation and proficiency, and communications.

Computer security awareness training is also important to the secure operation of the computer-managed physical access security system. An effective computer security training program can help ensure that computer security officials and computer resource users acquire and maintain the necessary skills and knowledge to reduce the risks of security compromise.

PREVENTIVE MAINTENANCE

Adopting sound practices for hardware and software design and implementation will contribute significantly to the availability of the computer-managed security systems. A preventive maintenance method needs to address at least four needs: to develop an effective preventive maintenance plan, to provide a maintainable computer system, to make available competent maintenance personnel, and to audit the maintenance program to assure that it conforms to the plan.

CONTENTS

ABSTRACT	iii
SUMMARY	v
ACKNOWLEDGMENTS	xiii
1.0 INTRODUCTION	1.1
1.1 BACKGROUND	1.1
1.2 STUDY OBJECTIVES AND APPROACH	1.2
1.3 REPORT ORGANIZATION	1.3
2.0 COMPUTER SYSTEM QUALITY ASSURANCE PROGRAM	2.1
2.1 MANAGEMENT CONTROLS	2.1
2.1.1 Organization and Planning	2.2
2.1.2 Policies, Procedures, and Standards	2.3
2.1.3 Personnel	2.5
2.1.4 Records and Reports	2.7
2.1.5 Internal Auditing	2.8
2.2 DOCUMENTATION GUIDELINES	2.9
2.2.1 Purpose of Documentation	2.9
2.2.2 Documentation in the System Life Cycle	2.10
2.2.3 Recommended Documents for Operations	2.12
2.3 CONFIGURATION MANAGEMENT	2.16
2.3.1 Introduction to Configuration Management	2.16
2.3.2 Objectives of Configuration Management	2.17
2.3.3 Configuration Management Implementation Plan	2.18
2.3.4 Automated Configuration Management Systems	2.22
2.4 OPERATIONS CONTROLS	2.25
2.4.1 Physical Security Controls	2.26
2.4.2 Hardware Controls	2.27
2.4.3 System Software Controls	2.27
2.4.4 Data Input, Processing, and Output Controls	2.30
2.4.5 Auditing	2.30
2.4.6 Problem Reporting and Preventive Maintenance	2.30
2.4.7 Contingency Plans	2.31
2.4.8 Personnel	2.31

3.0	COMPUTER SYSTEM EVALUATION	3.1
3.1	DEFINITIONS OF VERIFICATION AND VALIDATION	3.1
3.2	VERIFICATION AND VALIDATION MECHANISMS	3.1
3.2.1	Reviews	3.1
3.2.2	Walk-throughs	3.2
3.2.3	Inspections	3.2
3.2.4	Testing	3.3
3.3	VERIFICATION AND VALIDATION OVER THE SYSTEM LIFE CYCLE	3.8
3.3.1	V&V Planning During System Initiation	3.8
3.3.2	V&V During System Design and Development	3.8
3.3.3	V&V During System Installation ..	3.9
3.3.4	V&V During System Operation	3.9
3.4	AUTOMATED V&V	3.10
3.5	V&V AND NUCLEAR POWER PLANT COMPUTER-MANAGED SECURITY SYSTEMS	3.11
4.0	TRAINING	4.1
4.1	COMPUTER SYSTEM OPERATION TRAINING	4.1
4.1.1	CAS/SAS Officer Training	4.2
4.1.2	Decision Making	4.3
4.1.3	Computer Operation and Proficiency	4.3
4.1.4	Testing and Evaluation	4.4
4.2	COMPUTER SECURITY AWARENESS TRAINING	4.5
4.2.1	Computer Security Awareness	4.6
4.2.2	Risk Management	4.6
4.2.3	Computer Security Program Management	4.7
4.2.4	Security for Sensitive Applications	4.7
4.2.5	Contingency Planning	4.8

5.0	PREVENTIVE MAINTENANCE	5.1
5.1	REQUIRED MAINTENANCE FUNCTIONS	5.1
5.2	MAINTENANCE ALTERNATIVES	5.2
5.2.1	Vendor-Supplied Maintenance	5.2
5.2.2	Third-Party Supplied Maintenance	5.3
5.2.3	In-House Maintenance	5.3
5.2.4	Combination Approach	5.4
5.3	MAINTENANCE SOURCE SELECTION CRITERIA	5.4
5.4	CURRENT MAINTENANCE TRENDS	5.4
6.0	REFERENCES	6.1
APPENDIX A	- REVIEW CHECKLISTS	A.1

FIGURE

2.1	Documentation in the System Life Cycle	2.11
-----	--	------

TABLES

2.1	User's Manual Recommended Contents	2.14
2.2	Program Maintenance Manual Recommended Contents	2.15

ACKNOWLEDGMENTS

This report could not have been completed without the cooperation and assistance of many people in numerous organizations. The authors wish to extend their appreciation to the personnel at utility headquarters and power plants who so willingly assisted us in this endeavor. We extend our gratitude to Battelle colleagues who help us in technical analysis. In particular, we are grateful to Steve Matsumoto for editing and manuscript preparation. All these contributions make this document a better product than it otherwise would have been. However, the authors take full responsibility for the content of this report.

1.0 INTRODUCTION

Commercial nuclear power plant (NPP) licensees tend to rely on computer-managed systems to help meet the physical security requirements of 10 CFR 73.55 (for access control, alarm monitoring, and alarm recording) and the reporting and logging requirements of 10 CFR 73.70 (Energy Regulations 1984). The failure (or manipulation) of these computer-managed access control systems has a potential to threaten plant safety. Recognizing this impact, the Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission, has contracted with the Pacific Northwest Laboratory (PNL) to study the design, operation, and maintenance of computer-managed systems for physical security at NPPs. This report provides information on computer system operation, evaluation, and maintenance. A previous report provided information on computer system design and installation (NUREG/CR-4298).

1.1 BACKGROUND

The requirements for physical protection of nuclear power reactors against radiological sabotage are defined in 10 CFR 73.55. Many licensees are using or are in the process of installing or upgrading computer-managed security systems to help them meet the requirements of 10 CFR 73.55.

The NUREG/CR-4298 report concentrated on the design, selection, and installation of the computer systems. Security functions that could be managed by computers were identified; i.e., access control, alarm monitoring, alarm display, and record keeping. The study identified potential conflicts between safety and a computer-managed security system, and guidelines were proposed to minimize the system's impact on the safe operation of the plant.

For the computer-managed physical security system to meet its functional requirements and minimize safety/security conflicts, the system design must consider system performance criteria in five areas:

- level of protection (how much security desired?)
- system capacity (how large a system?)
- system response (how fast?)
- system reliability (how dependable?)
- user interface (how operated by people?)

Three design alternatives were identified for system implementation: dedicated stand-alone systems for each function, centralized systems that monitor alarms, assess alarms, and control access through a mini-computer that receives and processes input from the sensors in the plant, and distributed systems where the minicomputer interfaces with intelligent multiplexers that oversee the sensors in the plant.

NUREG/CR-4298 identified needs for the security of the computer system. The conclusion was that those needs could be met by implementing security

measures in five areas: 1) facility design, 2) access control, 3) administrative restrictions, 4) specifications in the procurement process, and 5) catastrophe planning.

NUREG/CR-4298 considered three categories of human factors associated with a computer-managed physical security system: 1) equipment layout and displays, 2) system space needs, and 3) workstation environment. Human factors design principles that should be applied to each category to make the hardware compatible with the human operators were identified. The report also discussed human factors guidelines regarding how the information could be displayed to the operator at the terminal and the way in which the operator enters commands and data to the system.

For the next step in the system development process, NUREG/CR-4298 provided information to aid in the writing of a system specification that would ensure that the vendors bid a system that meets the needs of the utilities. The report also focused on current vendor information and bid review procedures. Four general approaches to bid review were outlined: low bid, fixed price selection, evaluation of intangibles, and analytic hierarchic process. The final topic of NUREG/CR-4298 was system installation, including: site selection, site planning, supply storage, acoustics, environmental specifications, and electrical considerations.

NUREG/CR-4298 provided information on the stages of system development from conceptual design to system installation. This report provides information on system operation and maintenance.

1.2 STUDY OBJECTIVES AND APPROACH

Once a system has been installed, a great deal of effort must still be expended to ensure its proper operation. The objectives of this second phase of the study are to:

- study system validation and verification approaches and techniques
- evaluate training programs for users of computer-based security systems
- study hardware and software maintenance procedures
- suggest methods to assure that appropriate control is being exercised in the operation and maintenance activities of the security computer system to preclude unsafe or unsecure operating conditions
- recommend techniques and procedures that would protect against carelessness and malevolent actions.

To meet these objectives, the staff at PNL obtained information from literature reviews and site visits. The areas of study included:

- hardware and software maintenance

- user and maintenance personnel training
- configuration management system
- verification and validation
- user and system documentation
- management control
- operations control and procedures
- system audits.

Each of these topics is involved in the operation and maintenance of a system.

1.3 REPORT ORGANIZATION

The information obtained from the literature reviews and site visits was studied and organized into four discussions for this report:

- Computer System Quality Assurance Program (Section 2)
(management controls, documentation guidelines, configuration management, operations controls)
- Computer System Evaluation (Section 3)
(methods and steps for verifying and validating the computer system)
- Training (Section 4)
(for computer system operation and security awareness)
- Preventive Maintenance (Section 5)
(required functions, alternative sources of maintenance, selection of sources, trends).

2.0 COMPUTER SYSTEM QUALITY ASSURANCE PROGRAM

A computer system quality assurance program is based on a system of related internal controls: management controls, documentation guidelines, configuration management, and operations controls. Internal Control is the means by which errors in data are detected and corrected. It comprises the plan of organization and all the methods and procedures to safeguard data and system integrity and to encourage adherence to prescribed management policies. Each internal control measure is discussed in detail in this chapter. Each discussion considers the purpose of the internal control, the effect it has on a computer-managed security system operation, and important considerations that must be factored into the security system design, operation, and evaluation process for an effective quality assurance program.

2.1 MANAGEMENT CONTROLS

Management control embraces the entire system of organization, policies, procedures, and practices employed in managing affairs and promoting the effective execution of assigned responsibilities.

Recognition of the need to direct the efforts of people assigned to perform the work is important in establishing and operating a system of control. Most elements of management control are designed to control the activities of the employees. The specific techniques employed to exercise control over operations and resources will vary depending on the size, type, complexity, and geographic location of the plant. However, the following basic steps should be provided for an effective control system:

- set standards, goals, and objectives
- check on performance
- appraise results
- take corrective action, when needed.

Each of these steps involves action of some kind. The effectiveness of the control system depends on the competence and attitudes of all employees. The management process is not confined to a few people. It involves every supervisory level in an organization, and the effectiveness of the management process is a composite of the efforts of numerous employees.

Management should exercise sufficient control over the computer system in order to determine how well it operates, where improvements are needed, and what capabilities will be needed in the future. To accomplish these objectives, control techniques should be considered in the following areas (GAO 1975):

- organization and planning
- policies, standards, and procedures
- personnel
- reporting
- internal audit.

Collectively, these areas make up the network of management controls.

2.1.1 Organization and Planning

Effective controls need to be established over the computer processing and data management operations. The facility organizational structure needs to assure that data and program elements are safeguarded and that information is produced reliably. A management committee consisting of representatives from all groups associated with security and computers should be organized to formulate policies for computer system installation, operation, maintenance, and quality assurance.

The planning process needs to establish and document functional requirements, strategy, and overall system goals and objectives. It should also include individual responsibility for specific actions to be undertaken. Individual responsibility for alarm data entry, personnel key card information entry, records and reports generation, security system characteristics updates, and periodic system maintenance must be clearly defined so that responsibility cannot be sidestepped or exceeded, either intentionally or unintentionally. Such definition would also prevent blaming others for inaction or improper action. An individual who assigns responsibility and delegates authority to subordinates needs an effective system of follow-up to determine whether assignments are properly carried out. The individual to whom authority is delegated should be required to operate on prescribed terms but to check with the supervisor when exceptions occur. Even specific instructions cannot provide for all contingencies, and employees should be expected to recognize unusual situations. Every person must be required to account to their immediate supervisor for the manner in which responsibilities were discharged and for the results obtained, both positive and negative.

Duplications and conflicts in the assignment of functions, duties, and responsibilities should be avoided. If the assignment is made as part of a verification, then personnel working in the areas where verifications are required must be aware of that function. Verifications should not be viewed as duplication of work; they are required for proper quality assurance. The assignment of job responsibilities must also consider separation of duties that are critical to the secure operation of a computer-managed security system. These internal control measures include:

- separating the security department processing functions from other organizational functions, for example maintenance or computer programming
- separating duties within the security department, for example, authorizing area access and entering personnel area access information
- separating individual responsibilities for systems analysis, acceptance testing, data control, computer equipment operation, and configuration management.

Generally, the organizational structure should be as simple as possible. Establishing too fine a division of duties and responsibilities usually

results in inefficient operations and unnecessary positions. The organization needs to be flexible enough to easily allow changes in its structure required by changes in operating plans, policies, and objectives.

2.1.2 Policies, Procedures, and Standards

A policy is any rule which requires, guides, or restricts action. Policies are the predetermined patterns of conduct to be observed in carrying out specific activities. They are statements of management intention to act in certain ways under specified types of circumstances. All organizational entities have policies. The key is whether or not they are adequately understood by all personnel.

In contrast, procedures are the methods or means used to carry out activities in conformity with prescribed policies. A standard includes procedures and rules employed and enforced to prescribe a disciplined, uniformed approach that helps to ensure system quality. A standard provides the tool needed to measure system value.

Policies, standards, and procedures must exist to serve as a basis for management planning, control, and evaluation.

The following principles and requirements apply to all policies.

- Policies need to be clearly stated in writing and organized in handbooks, manuals, and other forms of company publications. Lack of clarity can result in serious misunderstandings and improper or inconsistent interpretations.
- They must be communicated to all employees.
- They should be designed to permit effective and efficient use of resources.
- They must provide assurance that data and information are suitably safeguarded.
- They need to be periodically reviewed and revised when necessary because of changing circumstances.

After basic policies, organizational plans, and responsibilities have been established, procedures are developed to carry out specific activities. The same principles and requirements for policies apply to procedures and in addition:

- A program for continuous or periodic internal review and improvement of procedures is necessary. For computer systems this review should be made by all parties involved; i.e., the computer development, maintenance, and security organizations.
- To reduce the possibility of error or system compromise, procedures should be coordinated such that one employee's work is automatically

checked by another employee who is independently performing prescribed duties. The extent of these internal checks should consider the importance of the data that is available from the system, availability of personnel, and the feasibility of implementing such procedures.

- Procedures should not be so detailed that they are hard to follow and understand when unusual situations occur.
- Procedures should not duplicate or conflict with others.

The U.S. Office of Management and Budget has prescribed internal control standards as the minimum level of quality acceptable for internal control systems in operation and as the criteria against which these systems are evaluated (OMB 1978, OMB 1983). These standards are divided into three groups: general standards, specific standards, and the audit resolution standard. The following general standards apply to all aspects of internal control (DOE 1984).

- Internal control systems are to provide reasonable assurance that the objectives of the system will be accomplished.
- Managers and employees are to maintain and demonstrate a positive and supportive attitude toward internal controls at all times.
- Managers and employees are to have personal and professional integrity and are to maintain a level of competence that allows them to accomplish their assigned duties as well as to understand the importance of developing and implementing good internal controls.
- Internal control objectives are to be identified or developed for each activity and are to be logical, applicable, and reasonably complete.
- Internal control techniques are to be effective and efficient in accomplishing their internal control objectives.

The following specific standards are the fundamental techniques that provide the greatest assurance that internal control objectives will be achieved:

- Internal control systems and all transactions and other significant events are to be clearly documented, and the documentation is to be readily available for examination.
- Transactions and other significant events are to be promptly recorded and properly classified.
- Transactions and other significant events are to be authorized and executed only by persons acting within the scope of their authority.
- Key duties and responsibilities in authorizing, processing,

recording, and reviewing transactions should be separated among individuals.

- Qualified and continuous supervision is to be provided to ensure that internal control objectives are achieved.
- Access to resources and records is to be limited to authorized individuals, and accountability for the custody and use of the resources is to be assigned and maintained.

The audit standard states:

- Managers are to promptly evaluate findings and recommendations reported by auditors, determine proper response to audit findings and recommendations, and complete, within established time frames, all actions that correct or otherwise resolve the matters brought to management's attention.

2.1.3 Personnel

An important management function is to assign duties and responsibilities to persons capable of carrying them out in a satisfactory manner. Personnel recruiting and promotion practices should be based on objective criteria and should consider education, experience, and security risks relevant to the job requirements and to the degree of responsibility (FIPS 1974). After determining what the job requirements are, employees who have the necessary qualifications or who can be trained to perform the work must then be obtained. If the computer system developed for the security alarm station operation is difficult to understand or learn, then many of the security personnel may be eliminated since they may have neither the skill nor the desire to operate the system. The alarm station operator position requires knowledge of the security operation at the facility, familiarity with computer operation in general, specific operation of the computer-generated menus and commands, leadership qualities, and the ability to handle both routine and high-stress situations.

Management must address the potential problem of absences, morale, and poor performance that can occur when overtime work combines with the pressures of high-stress situations. Some of the internal controls that management should consider are:

- Alarm station operator position descriptions should be in writing, be clear in delineating authority and responsibility, be kept current, be accompanied by definitions of technical skills needed, and be usable as a basis for performance evaluation.
- Employee performance should be evaluated regularly, and any negative performance should be appropriately addressed.
- When hired, employees should be provided with an orientation of the internal controls and security and with ongoing training to maintain their technical knowledge, skills, and abilities.

- Training programs should exist to build and maintain skills and knowledge in system technology and internal control and computer security requirements.
- Absentee and turnover rates among alarm station operators should be monitored for potential personnel problems.
- Alarm station operators should be required to take regularly scheduled vacations.
- To the extent possible, a personnel rotation plan should be in effect within the different functional areas of security and computer operations.

The most direct form of control over the performance of individuals is supervision. The quality of supervision is often the focal point of satisfactory control over the operations. Therefore, high standards of supervision should be established, and provisions set forth to review the quality of supervision at all levels in the organization. An important part of supervision is assuring that all employees understand their duties and responsibilities, understand where they fit into the organization, and are competent in the performance of their job. The correct processing of alarm and access authorization transactions require that each employee have clearly understood duties that are followed consistently. This division of duties is not possible if each person does not have a job title and complete position description.

The following list provides job classifications for personnel operating a computer-managed security system:

- Systems analyst - Analyzes and defines the requirements for new and existing applications. Designs new or improved processing systems to meet these requirements. Prepares specifications for the systems to guide programmers. Writes procedures and user instructions.
- Systems programmer - Is responsible for the maintenance, improvement, and testing of operating systems, library software, and system utility programs. Coordinates the operating and control interrelationships of application and systems programs.
- Application programmer - Identifies programs required by the systems design, and flowcharts the logic of these programs. Codes the logic in a programming language. Tests and debugs the programs. Prepares the program documentation.
- Data base administrator - Is responsible for the development and maintenance of a data base system.
- System operator - Performs tasks such as operating the computer equipment. Operates the computer in accordance with standard procedures and documentation for each application. Responds to and acknowledges

system alarms. Provides information to the system regarding the alarms. Communicates alarm status to responding forces. Controls personnel access to the plant via system data changes. Generates required system operating logs and reports.

- Librarian - Has custody of files, programs, and documentation. Checks files, programs, and documentation in and out of the library according to management policy. Catalogs all documentation. Maintains a record of all file, program, and documentation usage.
- Quality control manager - Establishes systems design, programming, and documentation standards. Reviews documentation to ensure compliance with standards and approves of documentation when standards are met.
- Security supervisor - Establishes procedures for access to equipment, data, programs, and other aspects of physical security. Monitors work of the security force. Investigates procedure violations, reviews violation logs and statistics, and carries out spot checks and audits of physical security procedures. Issues passwords or security codes and is responsible for ensuring that they are used correctly. This person is usually qualified as a system operator.
- Internal auditor - Assures management that all systems operate in conformance with management standards and approved design specifications to produce consistently reliable results. Acts as a user to assure that all systems contain necessary audit trails.

It may not be convenient or necessary that each of these jobs be performed by different individuals. Because the primary source of security system computers and their software are established vendors, the only computer personnel needed may be a single person who understands the system and is able to make maintenance adjustments and system modifications to support expanding requirements.

Two organizations should be directly involved with the computer operations: the security department and the computer systems department. The security data base and system operation should be strictly controlled such that neither organization has access to all of the data. Other people may also require access to various parts of the computer system and its data. These include security clearance personnel for entering area access information, security supervision for changing access authorizations and generating specific reports, and quality control personnel who audit the performance of the entire system.

2.1.4 Records and Reports

Computer-generated reports are necessary to provide management with current information about how well the security system is functioning, the progress of any changes or accomplishments, and where management action is needed. This information is a basis for management action and control. In establishing report requirements, the following principles should be considered:

- The reports should be made in accordance with NRC and management requirements and assigned responsibilities, for example, security event logs and personnel vital area assignments, area access history, and misuse of access control system operation.
- Reports should be as simple as possible and include only the information required for the report. Reports containing too much information are usually not reviewed at all.
- Progress or performance reports should compare past and present data in order to judge system effectiveness and monitor performance; significant deviations might indicate the need for corrective action.
- Reports should be timely and reviewed in a timely manner. This helps to assure that system performance is maintained at acceptable levels.

The reporting system should be reviewed periodically to determine whether it meets the needs of the security management: producing informative reports containing sufficient but no unnecessary data.

The following records could be available from a computer-managed physical security system and could serve as input to reports:

- security watch tours, inspections, and tests
- alarm annunciations
- security responses to alarms
- personnel badging records
- authorizations for access to protected and vital areas
- visitors, vendors, and other non-employee records
- personnel accountability by area
- security personnel training records
- security system maintenance records
- system software change records
- system malfunctions and audits.

2.1.5 Internal Auditing

Another important mechanism for providing management with necessary information is a system of independent internal review of operations, methods, systems, procedures, and practices (GAO 1975). The facility's internal audit staff should be responsible for assuring top management that

systems are developed in accordance with system objectives, contain the needed internal controls to produce consistently reliable results, and operate in conformance with management standards and approved design specifications. The internal audit staff's charter should allow it to conduct independent reviews and to report its findings and recommendations to the facility management.

The internal audit group should review general controls to determine that controls have been designed according to management direction and that these controls are operating effectively to provide reliability of, and security over, alarm data, access control data, and operational information. The internal audit group should actively participate in reviewing the development of a new system and any significant modification of an existing system.

Some basic principles and policies pertaining to an internal review system are:

- Top management should define the nature and scope of the internal review so that duties and responsibilities are adequately recognized.
- To promote adequate consideration and action on its findings, the internal review group should be responsible to management that is independent of the operations under review.
- Internal review is a staff or advisory function and should not include supervision or performance of line operations.
- A manual outlining the scope, objectives, and performance standards of the internal audit or review function should be developed.
- Internal reviewers should place primary emphasis on promoting improvement of operations.

Internal audits or reviews should not be considered as replacing, supplementing, or duplicating reviews by the NRC. Internal reviews should be an integral part of internal management control and are subject to the same review by NRC as any other form of control.

2.2 DOCUMENTATION GUIDELINES

Documentation that is developed concurrently with the development and testing of a computer-managed security system is important to the operation and maintenance of the system. This section discusses the purpose of documentation, describes the system life cycle as it relates to documentation, and describes in detail the operations documents recommended for this type of system.

2.2.1 Purpose of Documentation

Documentation is an area of software engineering where the professional computer scientist is usually not experienced. Few in the field have the

opportunity to learn to document satisfactorily. Documentation is usually treated as an add-on task to the software development project. Frequently, time provided for documentation is not sufficient. Many program documents are incomplete and often incorrect. The importance of documentation is often considered only as an afterthought whenever some difficulties have occurred later in the life cycle.

There are many reasons for preparing good documentation. Documentation:

- aids in planning and managing resources
- provides a means of communicating within the organization
- promotes common understanding and expectations between the buyer and seller of purchased software
- defines what the seller is expected to deliver and helps the buyer to verify correct operation of the delivered product
- gives the user the ability to install and use the software with minimal aid of the developer
- provides continuity and flexibility as personnel change
- aids in training the users of the system
- aids in making modifications to extend applications
- aids in implementing audits and evaluations
- helps in planning and implementing computer security procedures.

Documentation contributes to the successful operation and maintenance of a system. Documentation should not be approached as a task that is performed after the system has been developed and implemented. Good documentation practices must be followed at every step of the life cycle.

2.2.2 Documentation in the System Life Cycle

For the development and operation of a computer-managed security system, we have divided the system life cycle into four distinct stages as shown in Figure 2.1: system initiation, system design and development, system implementation, and system operation. Each phase of the life cycle has documentation that should be produced. Although this report concentrates on the system implementation and operation stages, each of the stages is briefly described along with the documentation that is recommended.

During system initiation the purpose and description of the system and the needs for development resources are identified. Documents produced in this phase describe the user requirements, define a general approach for the

FIGURE 2.1. Documentation in the System Life Cycle

proposed system, discuss alternative design strategies, and recommend the best approach for the development of the system.

The system design and development stage resolves all design features of the proposed system. All input, output, data base, and processing aspects of the design are developed and documented. The next objective of this stage is to transform the design into functional system components through program design, coding, and testing. At this stage, the computer code should be documented with internal comments in the code. This is perhaps the greatest aid that can be provided to the programmer who must understand and maintain the code during the system operation stage. Various tools, well-documented in the software industry, can aid in structuring the code for future changes. The User's Manual should be planned and begun in this stage.

The system installation stage includes conducting system tests, performing necessary conversions, training users, and beginning operations. User acceptance is a desired part of this stage. A test plan should be developed and documented so that test results can be compared with test plan predictions to ensure that the system is operating properly. As part of the system test, users conduct acceptance tests to verify that their specifications have been met.

The User's Manual should be finalized during this stage and tested by the user. The first draft of the Program Maintenance Manual should be completed during this stage.

During the final stage, system operation, the objective is to maintain an efficient operating system by periodically evaluating it and making changes as additional requirements are identified. Documents produced during this phase describe problem resolution procedures such as abnormal program completions, correction actions, condition-caused problems, requests and documentations for minor system enhancements, and documentation of program module changes.

2.2.3 Recommended Documents for Operations

Although documents can be identified for each stage of the system life cycle, documentation should not be considered as a process that is finished once a stage is complete. Documentation preparation should be treated as a continuing effort, evolving from preliminary draft through changes and reviews, through the documentation and software delivery, to subsequent updates indicated by user feedback.

The documentation recommended below may be supplemented by comments in the program code or on-line user documentation. Standards and recommended contents are provided for each document. These should be viewed as a checklist of items to be addressed in the document. Specific formats are not provided. The recommendations provided below are compiled from the EPRI Software Development and Maintenance Guidelines (EPRI EL-3089), the FIPS publications (FIPS 38), and the NBS FIPS Software Documentation Proceedings of a Workshop held March 3, 1982 at the National Bureau of Standards, Gaithersburg, MD.

The documentation necessary for the operations phase of a system includes a User's Manual, test plans, a Program Maintenance Manual, and configuration management procedures. The test plan and configuration management procedures are discussed in detail in later sections of this report.

User's Manual

The purpose of the User's Manual is to sufficiently describe the functions performed by the software so that the user can determine when and how to use it. The User's Manual should serve as a reference document for preparation of input data and interpretation of results. The information that should be included in the manual is shown in Table 2.1.

The content of the User's Manual should focus primarily on procedures, i.e., how to use the software. General information about data bases and other features of the system is provided as background information.

Documentation is more understandable to non-programmers when the software is described as the user sees it--as a set of functions that are performed by a set of commands or menu selections. A good User's Manual would list the functions and then list the commands or responses that cause the functions to be performed. Examples of the commands and expected responses help the users' learning. The documentation should be organized by functions, thus helping to make it clear to the user what the system can do and the commands necessary to do it.

The User's Manual should be used during the training period. By doing this, the users learn to use the documentation as well as the system. This is also a good time to obtain feedback about the manual and make the necessary revisions.

In addition to the formal User's Manual, on-line documentation is an important issue for computer-managed security systems. Considerable research about on-line documentation still needs to be done. Few guidelines can be specified based on work that has already been done in this area.

HELP functions need to provide detailed and useful information that does not require the user to refer to additional source references. To provide sufficient information without erasing current screen formats, some form of a windowing or split-screen technique should be used for displaying the HELP information. In this way, the user might be able to carry out a function while viewing the HELP information.

Error messages are one of the most discouraging and problem-filled areas of documentation. The error messages should be integrated with the HELP functions for on-line documentation. Error messages might recommend which HELP function to use.

Table 2.1. Users Manual Recommended Contents

General Information

Summary	Summarizes the general functions of the software and briefly describes the operating environment.
Application Description	Describes when and how the software is used and how it fits in the overall operating environment.

Procedures and Requirements

Initiation	Description of the step-by-step procedures required to initiate processing.
Inputs	Defines parameters and the requirements for preparing input data. Specific instructions should be provided along with sample inputs.
Outputs	Describes the requirements relative to each output. Should include output formats, where applicable, and sample outputs.
Error and Recovery	Lists error codes or conditions generated by the software and corrective action to be taken by the user.

Supplemental Information

Performance Statistics	Describes the performance capabilities of the system including response times, processing times, and information on flexibility and reliability.
Data Base	Describes all files that are referenced, supported, or modified by the software.

Program Maintenance Manual

The Program Maintenance Manual should provide a maintenance programmer the information needed to understand the program, the operating environment, and the maintenance procedures. This manual is very important when the maintenance programmer was not involved in the original code development. Maintenance refers to the activities of correcting code or documentation and the retesting that occurs with codes in a production environment.

Table 2.2 provides suggested contents for a Program Maintenance Manual.

Table 2.2. PROGRAM MAINTENANCE MANUAL
RECOMMENDED CONTENTS

General Information

Summary: Summarize the general nature of the software and the operating environment.

Module/Program Descriptions (for each module/program)

Overall Structure	Describe the role of the software and the role of each significant component in the operation of the software.
Models & Algorithms Applied	List the models and algorithms used in the program.
Input	Describe in detail the input to the program.
Processing	Describe processing features and purposes important to the maintenance programmer such as: processing logic, linkages, variables and constants, formulas, error handling provisions, restrictions and limitations, shared storage locations, settings, internal switches and flags.
Output	Describe the output from the program.
Error Messages	Describe all error messages, their causes, and suggested corrective action.
Interfaces	Describe interfaces with other software.
Data Tables	Describe arrays or sets of data items internal to the computer memory and files of data external to the main computer memory.
Execution Description	Describe or reference the operating procedures to execute the module/program.
Flexibility & Expandability	Describe array dimensions, constants, tolerances, or problem sizes that need to be changed.
Operating Environment	Describe the computer hardware and software that is required for the code to operate.

Maintenance and Updating Procedures

Programming Conventions	Describe any programming conventions used in the programs.
-------------------------	--

Table 2.2. PROGRAM MAINTENANCE MANUAL
RECOMMENDED CONTENTS
(Continued)

Verification and Validation Procedures	Describe the verification and test procedures used to check the correctness, validity and performance of the programs.
Error Correction Procedures	Describe error conditions, error messages, error sources, and procedures for correction of errors.
Special Maintenance Procedures	Describe any special procedures required for the maintenance of the programs.

2.3 CONFIGURATION MANAGEMENT

Configuration management (CM) has been described (Bersoff et al 1979) as "the discipline of identifying the configuration of a system at discrete points in time for purposes of systematically controlling changes to this configuration and maintaining the integrity and traceability of this configuration throughout the system life cycle." Software configuration management has been further described as a "housekeeping" function established to assure that during all phases of the software life cycle there is an orderly approach taken, that all changes are traceable and have proper approval, and that administrative requirements are met (Doggett, 1983).

To help the reader gain a better understanding of CM and its possible application to computer-managed security systems at nuclear power plants, this section of the report introduces configuration management, describes the goals and objectives, discusses a CM implementation plan, and briefly discusses automated CM systems.

2.3.1 Introduction to Configuration Management

CM has evolved as a methodology to solve recurring problems of complex system development. These problems have been: software systems do not provide the functions the user requested; the evolution of system configurations cannot be traced; changes to requirements, software design implementation and documentation are not controlled; documentation for project phases is not structured, consistent, or complete; system reliability is uncertain; responsibility is non-existent or not well defined; and system maintenance is extremely difficult.

Failures of complex systems to meet users' needs have resulted mainly from lack of discipline in system development. Initial requirements specifications have often been adequate, but mechanisms have not existed to control and trace changes. Unstructured documentation has often resulted in

a "crazy quilt" documentation package when subcomponents of a system are integrated. Either errors in systems have not been identified and corrected, or correct code has been modified unnecessarily. Changes to systems (who made them, what they were, when) have been difficult to trace, making system maintenance more difficult. Any disorder of complex systems tends to increase with time.

CM was first implemented with hardware-oriented systems and has brought order to the development of complex systems. Software configuration management (SCM) has applied the same tools as hardware CM to software development projects but with varying degrees of success. The primary stumbling block of SCM has been the definition of software. Two extreme definitions of software are: 1) only the program code, or 2) all documentation which in some way determines the code. Applying CM using the first definition leads to incomplete configuration description, limited traceability, and (therefore) questionable integrity. Using the all-inclusive second definition tends to increase costs, complexity of system development, and redundancy.

SCM is continuing to mature, as indicated by an increasing number of successful implementation accounts found in the literature. Hardware and software CM do not need to be approached separately. A system-wide approach (hardware and software) has been used successfully in developing some advanced flight systems (Petersen and Flores 1983).

Configuration management is a management tool. As a tool, it is set up to ensure that recognized development steps are taken and the resulting products are consistent with requirements. CM must assure that changes are controlled and accounted for.

2.3.2 Objectives of Configuration Management

The objectives of CM are to improve the reliability of a system, decrease total costs over the life cycle of a project, provide a method to determine the state (configuration) of a system for any time during the system life cycle, provide traceability and therefore responsibility, assure that user requirements have been met, approve and monitor all changes to the system, and support the functions of auditing, quality assurance, validation and verification, and test and evaluation.

Doggett, Wilburn, and Carey (1983) have suggested adopting CM for the following types of systems:

- systems essential to the design, development, construction, operation, or maintenance of a nuclear program or facility
- any system where malfunctions would have serious impact on critical plant functions
- computer systems that will be maintained over an extended period

- any computer program that is used to support the design, construction, operation, or maintenance of a facility that has a potential for licensing
- any software system where the development costs or processing costs represent a significant part of the organization's budget
- software systems with a high potential for use at other government sites.

The best time to implement CM is at the onset or during the preplanning phase of a project; however, implementation at any time during a system's life cycle is better than not at all. Planning and implementing CM takes time, and experience is necessary before it will be used. The lessons learned about successfully implementing CM for one system will be beneficial for future systems development.

The advantages of using CM include:

- System integrity is improved.
- Changes are approved, understood, and documented.
- Design or implementation errors or changes are traceable.
- The possibility of unauthorized changes is greatly reduced.
- System life cycle costs are reduced.
- The relationship between users and developers is improved.

The disadvantages include:

- A new discipline needs to be learned and implemented.
- Early life cycle costs will likely be greater, though costs over the entire life cycle may be lower.
- Corrections or changes may be delayed, pending approvals and/or documentation.

The advantages of CM are significant, and the development and operation of a computer-managed security system will benefit from the application of CM. The next section describes a plan for implementing CM.

2.3.3 Configuration Management Implementation Plan

There are four basic elements of CM:

- configuration identification
- configuration control

- configuration status accounting
- configuration auditing.

Configuration identification is the process of determining project base lines and correlating them to life-cycle phases. A base line is defined as a milestone in the system life cycle and each base line represents reference points in the development of a system. Section 2.2.2 describes four life-cycle stages of a system: initiation, design and development, implementation, and operation. Each of these stages may be viewed as distinct base lines. Base lines may change if requirements change or if deficiencies are identified and corrected. Thus base lines are dynamic and should be time traceable.

Base lines can be broken down into elementary subcomponents called software configuration items (SCIs). SCIs may be identified as physical entities such as subroutines, functions, information files (for example, FORTRAN "INCLUDE" files), or data files. SCIs may be functionally represented entities which encompass parts of program modules or files. Also, SCIs may be simply parts of documentation found, for example, in the system initiation stage.

The configuration identification function involves determining the constituent parts of the software and the relationship of those parts, assigning a name and label to each part, and depicting the identified software in a tabular or graphical form. Proper identification of software configuration is important so that the state of the software can be formally defined at any time. This establishes a common basis for negotiation among all parties with an interest in the system.

An organization must provide a unique method of identifying each SCI in a way that ensures accountability for all changes. One identification scheme might be to assign a number consisting of three elements:

Computer program designator - SCI designator - SCI revision number

These identification numbers could be the basis for configuration control.

Configuration control is the process of approving, monitoring, and controlling the conversion of the requirements specification into software, and managing the system changes. It involves recognition of the need for changes, determining how and when changes can be made, and the mechanisms for tracking changes. Controlling a system involves management planning, change approval, and documentation.

Doggett, Wilburn, and Carey (1983) have suggested that at the outset of any project, a management plan should be activated. This management plan should address five areas:

- SCM organization (i.e., structure, responsibilities, interfaces, configuration control board, and associated policy and directives)
- Details of SCM functions (identification, control, audit, and accounting)
- SCM procedures and forms
- SCM resources such as funding, staff, etc.

A configuration control board (CCB) should be established to provide a forum for management control. The board should have representation from functional and/or project management, user groups, the contractor or developing group, and CM representatives. It is the CCB which administers change approval. The administration of change control is the function that will strongly affect the performance of CM. Considerable time and effort will be required from those represented to review proposed changes, to assure CCB understanding of proposed changes, and to make timely decisions.

CCBs may either meet regularly or as the need arises to review change requests. They review and either approve the change, deny the request, or request more information or action before making a decision. There are two extreme approaches which CCB members could take that would negatively affect CM goals and objectives. First, CM members might act as "rubber stamps" allowing all change requests to be quickly approved and implemented. This approach is one of no control and the CCB would not be serving its function. The second extreme is that of excessive deliberation and therefore failure to act in a timely manner. This could result, for example, if CCB representatives have little technical comprehension of the system being developed. The eventual outcome for this second extreme would be schedule delays and cost overruns.

Procedures for documentation of changes should be prepared using formally accepted formats and structures. One practice is to appoint a librarian to actually modify SCIs and assure that documentation is completed according to CM policy. A librarian's function might be as in the following process. First, the user decides to effect a change in the system. Then the user and developer discuss the change and prepare a proposal for change. The CCB members review the change request and approve the action, and the developer determines what technical action is required to implement the change. At this point the librarian could be responsible for actually making the change to the software, including documentation to describe the change, and any non-executable code changes (comments), and for compiling the changed module and linking the system modules, etc. The librarian would have saved the complete software package on magnetic media (tape) before making the specified modification. A base line revision number would have been saved with the "old" snapshot and a new one assigned for the current system. Testing and evaluation would then be performed by the developers/users, and problems would be noted by the librarian.

Depending on the complexity and extent of a particular system and the staffing constraints, a nuclear power plant utility may or may not identify a

specific individual with librarian responsibilities. However, successful configuration management requires that the librarian functions be performed, or the CM being implemented will not meet its objectives of effecting order and discipline.

Status accounting is the formal process of documenting the configuration identification, control, and auditing. Its purpose is to allow an authorized person to review the system software configuration as it existed during any stage of the system's life cycle.

Configuration auditing is the process of verifying that a system meets specifications and that change implementations and releases are traceable. Doggett et al. (1983) separated configuration auditing into reviewing and auditing. The following are suggested reviews:

- System requirements review
- Preliminary design review
- Critical design review
- Verification readiness review
- Software verification review.

The following are suggested audits:

- Functional configuration audit
- Physical configuration audit
- Software audit.

Reviews serve the purpose of: separating base lines so they may be more easily identified, evaluating development phases, and providing mechanisms for follow-up reviews when changes are implemented or deficiencies have been corrected. Systems requirements reviews verify that the users' needs have been correctly identified and documented. Preliminary design reviews verify that the functions identified to meet requirements are complete and correct, and that the design requirements will form an adequate base for the detailed design. Critical design reviews verify the technical approach for the detailed design and grant approval for code implementation to begin. Verification readiness reviews verify that the completed system code and associated documentation (i.e., user manuals and program maintenance manuals) are complete and that the project is ready for verification. Software verification reviews verify that the developed system satisfies requirements, that system components interface correctly, and that the system is ready for routine operation. The review process as just described is a tool for identifying and completing base lines.

Once project base lines have been completed and reviewed, the system is ready for auditing. The functional configuration audit is a process to

verify that a completed system has been tested and that the system performance meets requirements. The physical configuration audit verifies that software, supporting documentation, and all system hardware are installed, complete, and acceptable for routine operation. The software audit includes the following:

- audit/review results
- identification of software components
- project development references
- the audit procedure
- quality assurance findings
- configuration identification
- findings of the requirements evaluation
- test results for traceability of requirements to software product
- verification findings
- validation findings
- configuration status accounting (record of SCI modifications)
- conclusion of all reviews and audits
- recommendations.

2.3.4 Automated Configuration Management Systems

For each of the primary elements of CM (identification, control, status accounting, and auditing), clerical support plays a principal role. Failure of clerical efforts to harmonize with engineering and management may mean that a particular CM project will fail. Automation is being used to improve the reliability of CM clerical functions and expedite CM clerical tasks in many applications.

One example of computer-supported CM is the systems-wide approach to digital flight control systems reported by Petersen and Flores (1983). A microcomputer-based system and a database program were used as tools to provide CM automation support for clerical tasks. This system tracks CM progress and provides status reports. Since this is a hardware/software integrated approach, the automated system tracks hardware as well as software documentation changes. The database is used to store, change, and retrieve the status of discrepancy reports, program changes, work orders, and verification and validation findings. Because it is a database-centered system, information may be retrieved in a wide variety of ways, and

corresponding hard-copy reports may be generated. This automated approach as a CM tool is reported to improve accuracy and efficiency.

Another automated CM system, Automated Revision Control System for Software - ARCS, is described by Duerling (1981). The FORTRAN system executes either on a Univac 1100 or a VAX 11/780. The system performs the following functions:

- controls changes
- creates release version
- creates configured version
- generates problem reports, change reports, status reports, and version descriptions
- maintains logs
- provides cross references
- checks changes on a module-by-module or a system-wide basis.

One of the ARCS strong points is its line editor. The ARCS editor is similar in function to other software editors, but all revisions of files may be reviewed for any time period since file creation. It is beyond the scope of this report to describe all detailed functions; however, it seems that CM goals and objectives have been the driving force in the design of this system. For example, all program modules are automatically compiled on the existing editor, and ARCS has been designed to compile modules only via the editor. The user may, however, exit the editor without compiling, which occurs when program coding has not been completed. The following is a summary of ARCS editor functions:

- create and update files
- show any version of a file since its creation
- store configured and non-configured file versions
- store current file version information
- store date, time, and source version of all file versions
- save the reasons for change
- provide file names (in case file names are changed)
- store instructions for which compiler to use or which processor is being used.

ARCS is a CM tool that can be used directly or as a starting model for the development of similar systems.

The final CM automation tool reviewed was The Configuration Management System (CMS) described by Zucker (1983). The methodology of this system is similar to the one just described with the exception that all CM tracking is done using the computer. The CMS runs on a VAX system, works with any type of file (i.e., executable data or information) and thus performs CM on all items in the configuration. Bookkeeping and status accounting is done interactively with the computer. After being approved, configuration items are created or modified using the system. Like ARCS, CMS uses a database approach to follow system status. A variety of standard and user-designed reports have been generated by CMS using the database system. CMS is reported to be user friendly with HELP functions available on a need-to-know, password basis. User interactions may be stored for future retrieval and review (audit trail).

In addition to ARCS-like functions of producing discrepancy reports, change proposals and work orders, CMS reportedly does error checking (i.e., the system flags suspect user input, data-format errors, illegal access requests, etc.).

CMS places configuration items into one of three categories:

- development
- test
- release.

The movement of configuration items (CIs) is allowed after an item has been completed and approval obtained. For example, a CI in development can be moved to test after the completion of all "paperwork" and QA requirements have been satisfied. (This is actually a "paperless" approach.) Appropriate forms are completed via terminals by those responsible. Even the engineer review board (ERB) approvals are done using the computer system.

Advantages and disadvantages of automated CM systems are similar to those for the automation of most processes.

Advantages of Automated CM Systems

- They are faster than manual systems.
- Associated CM documentation is easier to maintain in pre-determined formats.
- The documentation may be retrieved faster.
- Base line and SCI status and historical data may be stored in a database for quick retrieval and relatively simple report formatting.

- Audits may be more quickly and easily completed.
- Responsibility for work and changes can be quickly traced.
- They save clerical time and money.
- They make it easier to standardize CM for different projects.
- Their reliability is better.

Disadvantages of Automated CM Systems

- More highly skilled personnel are needed to operate these systems.
- Rigid structures may result in some information loss.
- No automated system will execute on all computers.
- No automated system is recognized as a standard.
- Loss of project data on magnetic media due to a disc head crash or tape destruction could result in the loss of considerable data unless proper backup procedures are enforced.

2.4 OPERATIONS CONTROL

An operational security program applies to all staff, including those who require only infrequent access to the facility in the process of doing their work. Maintenance personnel and visitors would be in this category. To assure a continuing high level of security awareness, it is necessary to regularly reinforce that awareness for all personnel. Spot checks, posters in conspicuous places, newsletters, etc. can aid this effort.

The following discussion of the security of the computer center as well as the central alarm station/secondary alarm station (CAS/SAS) is divided into the following topics:

1. Physical security controls
2. Hardware controls
3. System software controls
4. Data input, processing, and output controls
5. Auditing
6. Problem reporting and preventive maintenance
7. Contingency plans
8. Personnel.

2.4.1 Physical Security Controls

Physical security controls for a computer system are those procedures and equipment installed to assure that only authorized individuals gain physical access to the computer equipment or facility. This includes all users of the system, employees needing regular access (such as janitorial and plant security personnel), and those needing infrequent access (such as visitors or computer vendor maintenance personnel). The following controls are not intended to be all inclusive but do apply to both the computer center and the CAS/SAS.

Access Control - Positive access protection should be kept in place even when the facility is occupied. This would require either that the occupants take positive action that would allow any additional access to the facility or that the additional people enter the facility using a key, magnetic card, etc. This eliminates the need for those working in the facility to continually monitor the coming and going of personnel. Any casual access that would not be desirable in these facilities is also eliminated.

Computer Center Alarm - If the computer center is considered a sensitive area, it should be locked and alarmed. A strict access procedure should assure that any person requesting access is authorized to do so.

Hardware Access Control - Computer hardware should at least be behind lock and key, magnetic card, or equivalent with the access device (keys or cards) being tightly controlled. It is important to realize that this is not a fool-proof solution since "tailgating" can occur; for example, a person with authorized magnetic card entry through a door could be followed by an unauthorized person who could pass through the door before it closed. It would be up to the person with legitimate access to notice what was happening and contact security; otherwise it would go completely unnoticed.

Visitor Control - It is necessary to maintain and review visitor logs to assure the need for access to the computer room or CAS/SAS is valid. Visitors should be escorted at all times by a person who knows enough to easily determine whether a visitor is doing something that should not be allowed. Employees should understand that it is important to challenge personnel if there is any question about what a visitor is doing.

Shift Changes - Under normal operating procedures, regular shift changes of personnel will be necessary in the CAS/SAS. These changeovers should follow predetermined procedures, taking into consideration the following:

- a. The same steps should be followed every time.
- b. All steps should be logged, indicating time, date, and personnel.
- c. The checkoff procedures for shutdown and startup should be signed by both the person coming on shift and the one leaving.

2.4.2 Hardware Controls

Hardware controls in the operation of a computer security monitoring system normally involve preventing unauthorized use of the devices themselves, corruption of data in the devices, or the interruption of the equipment's use. Several existing devices can provide an additional level of control over the standard use of the usercode/password. These devices can be installed as part of the terminal.

Terminals can be fitted with magnetic card readers, some of which use encryption as an additional level of security. These devices require the user to insert the card to gain access. In some cases the device makes the actual contact with the main computer system. This kind of connection with the computer system allows additional control as to what procedures the system allows the terminal (and user) to do. In effect the magnetic card code would be used to check the computer database to find out what functions the owner can access. This would allow the user to work at different consoles, if necessary, and not be limited to a specific station. This could be significant during an emergency situation when it may not be known where a specific individual is located or needed.

Standard key locks can be installed on the terminal; these would have to be turned before the machine is usable. Locks can be installed on the power unit so that the system will not function at all, or on the keyboard so that the system power can be left on but no input can occur.

Other possibilities for access control through the terminals include voice-activated devices, palm readers, hand-held laser devices, etc.

It is necessary to assure that the sensors are working properly. The most common way of doing this is to design circuit testing routines that run from the central system on an irregular basis. The set of expected results can be compared with actual results to assure the integrity of the circuits and sensors. This will reduce the possibility that the monitoring circuits could be tampered with by an aggressor or fail without the knowledge of the CAS/SAS operators.

2.4.3 System Software Controls

System software controls are put in place so that the actual machine processing is not affected by outside influences that might corrupt or destroy data, or that might cause improper operation of the computer system. Some steps that can be taken to prevent this kind of occurrence are described in the following paragraphs.

1. Dial-Up Telephone Lines - No dial-up telephone lines should be allowed on the system. If this is not practical, then special care needs to be taken to limit the number of lines and protect the system from unauthorized access by hackers.

2. Usercodes/Passwords - The security device most commonly used for access control is the usercode/password combination. While usercodes are rarely protected, passwords should be protected in every way practical. This can be done by implementing the following procedures:

- a. Only the person who actually uses the password to access the system should know what the code is. It is not even necessary or desirable that the system manager know each user's password.
- b. The passwords should be no less than six characters long. Longer passwords increase the difficulty of breaking the codes.
- c. Passwords should be randomly generated rather than thought up by the user; user-generated passwords tend to be easier to guess.
- d. Passwords should not be written down in any uncontrolled manner. For practical reasons, this means that the random generator creating the passwords needs to generate pronounceable passwords. Any written lists should be kept under lock and key.
- e. The passwords should be changed periodically. Longer times between changes make it more likely that the passwords will be compromised. On the other hand, if they are changed too often, such as weekly, the users are more likely to write them down instead of memorizing them, thereby creating a vulnerability to compromise. A middle ground is necessary and depends on the number of users and the environment. However, it is recommended that the interval between password changes exceed one month but be less than one year. A password should also be changed whenever there is any suspicion that the password has been compromised.
- f. It may be desirable to issue different passwords for different functions.

3. Operating System Restrictions - Through the use of the operating system and the utilities there are several restrictions that, if used properly, can enhance the level of security of the system without unreasonably affecting system use. Some of these restrictions are as follows:

- a. Limit the number of times that a user may enter wrong data during the logon procedure. The number of times commonly allowed is three. This takes into consideration the fact that some authorized users do not type well or could make other mistakes. However, if after three tries the user is still unsuccessful, then someone should investigate the attempted access. While the problem is being investigated, the computer should not recognize further attempts by the user to gain access.

This procedure eliminates the possibility that someone can use a computer program with a loop that tries all possible combinations of usernames and passwords until it finds the combination that is successful and gains unauthorized access.

- b. It may be beneficial to further restrict the actions of a user from a specific terminal to protect highly sensitive capabilities.
- c. Procedures should restrict who is allowed to bring the system up into an operating state or to shut the system down during either regular exercises or an emergency. The procedures would include situations when the system goes down on its own (crashes) and maintenance periods when only a reboot might be required.
- d. Additional restrictions can be put on the data and program files to limit the kind of access (read, write, update) that a user is allowed.
- e. Restrictions would be appropriate on which terminals are allowed to do what procedures both inside and outside the facility. This can be implemented by restricting the terminals, using passwords, or both methods.

4. Protecting Sensitive Media - Under no circumstances should sensitive media be left unprotected. This includes program files or listings as well as data.

5. Data Backups - Backup of the data and working files as well as the applications programs needs to be done on a regular basis as defined by the needs of the facility. However, there should not be a heavy dependence on the "system" backup because most of these backups are only good for total disasters when everything must be rebuilt. In addition to the "system" backup, each facility or user must decide what kind of a backup routine is needed and implement it accordingly.

6. Review and Testing of Program Modifications - In cases where it will be necessary to modify the applications programs, it is important that all modifications be reviewed by appropriate management to assure that no effects other than those desired will occur. Following the modifications, the programs need to be tested to assure that the computer systems operate as intended.

7. Separation of Duties - A major concern not addressed in many facilities is the concept of separation of duties. The greater the diversity of tasks that any one individual is given to do without appropriate checks and balances, the easier it becomes to cause a system malfunction or to abuse the system through improper use, corruption of data, destruction of data, or use for personal gain.

For example, consider the case of having only one person design the computer system, write the code, test the programs, run them in production, and modify them with no oversight. That person could insert any desired code, such as trap doors and logic bombs, without fear of being detected. For this reason both separation and rotation of duties are important management tools to increase security and reduce vulnerability.

2.4.4 Data Input, Processing, and Output Controls

Scanning routines that review job streams as they are entered by comparing them to predefined criteria (such as what jobs are expected to be run, when they are expected to be run, and by whom) can be helpful in recognizing the misuse of resources. Consideration can be given to limiting when certain work, such as code development, printing of output, etc. can be done.

Procedures can also be established for recognizing when users are attempting to run jobs not common to their areas of responsibility.

The capability to shred documents that might be considered sensitive should be provided in a convenient usable location. A workable policy is: when in doubt, shred.

2.4.5 Auditing

Computerized auditing is a basic tool for protecting a computing resource. Audit files are the key to verifying that only legitimate accesses are allowed to the system, data files, and programs. When unauthorized accesses are attempted, audit files will log what compromised passwords were used and what unauthorized attempts to access systems or files were made. If the password procedure allows, computer use attempts that exceed a user's authorization can also be logged.

The information that can be collected during system operation depends heavily on the power of the operating system and the auditing package. Because the logging capability is an overhead function of the computer and can affect the speed with which work on the system is executed, audit files must be carefully implemented. If there is too much data to review, then the logged data will not be used and is worthless. If the wrong data is logged, then the review will be meaningless. Instead, if the audit files and the level of detail are carefully chosen, the audit files can be a very powerful tool in monitoring the security of the system and in investigating computer abuse.

2.4.6 Problem Reporting and Preventive Maintenance

The most important part of a preventive maintenance and problem solving program for a computer facility is documentation and routine preventive maintenance. Almost all systems maintain log files of hardware errors that occur as the system operates. These should be reviewed regularly, not just when the system shuts down with a problem. Routine preventive maintenance can assure fewer problems during production time. Maintenance should be scheduled ahead of time so that preparations can be made and so that the system will not be down any longer than necessary. This is especially true of computer systems operated in a production mode around the clock.

For security reasons, the scheduled preventive maintenance should not be always at the same time. The schedule should be considered sensitive and known to as few individuals as possible.

For systems that are considered sensitive, the maintenance personnel should protect the integrity of the production software by using their own versions of all software when working on the system. This requires that the system be shut down in the normal way and, when the maintenance personnel finish, that the system be put back into production mode as described in the operations guide.

It is essential that all maintenance personnel have the same level of clearance required for those employees that are responsible for the operation of the system.

2.4.7 Contingency Plans

Contingency plans should be in place addressing all possible threats to the computer facility. This could be as simple as just recognizing that an event could happen and that, for the duration, security would have to be a manual procedure. At the very least, the time necessary to bring in new equipment and people necessary to put the system back on-line should be estimated. Plans should also be developed in at least a brief form for much less catastrophic occurrences (FIPS 1974).

2.4.8 Personnel

To prevent the intentional destruction of data or hardware by a disgruntled employee that may be dismissed, it is absolutely necessary that the employees not be given any kind of access to the computer system, not even through a remote terminal, once they know that they are being let go. If it is company policy to give several weeks notice to an employee, it is best that the employee be transferred to some less sensitive duties for that period. Individuals being let go with no notice should be escorted while being allowed to clean out their desks, etc. Under no conditions should an individual be put on probation and still be allowed access to a sensitive computer system.

The amount of damage that an individual with access to a company's computer system can do in only a few seconds is simply not worth risking. The cost of paying an individual to do nothing at all during the separation period is small compared to the possible loss if the employee were allowed access to the computer system.

3.0 COMPUTER SYSTEM EVALUATION

Software verification and validation (V&V) are defined and reviewed in this section. The objectives of this discussion are to: 1) describe V&V mechanisms and methodology; 2) explain how V&V apply to each stage of a system life cycle; 3) show how V&V can be automated, and give some examples of automated V&V; and 4) discuss how computer-managed security systems in nuclear power plants (NPPs) might benefit through the use of V&V.

3.1 DEFINITIONS OF VERIFICATION AND VALIDATION

Verification and validation are processes that assure that the specified requirements are stated and that the product fulfills these requirements. Deutsch (1982) has provided formal definitions of verification and validation:

Verification: an activity that assures that the results of successive steps in the software development cycle correctly embrace the intentions of the previous step.

Validation: an activity that assures that each product functions and contains the features as prescribed by its requirements specifications at the corresponding level.

These definitions for V&V are the current philosophy for using V&V during the system life cycle. Earlier applications approaches for V&V during project life cycles were somewhat different. The traditional approach to validation has been to use validation activities after the system design and development stage instead of during the entire life cycle. The primary advantage of the current V&V approach of validating each end item product is the early recognition and correction of problems.

3.2 VERIFICATION AND VALIDATION MECHANISMS

Using the above V&V definitions as a basis, this discussion addresses some mechanisms used to verify and validate software products: reviews, walk-throughs, inspections, and testing.

3.2.1 Reviews

Reviews are essential to V&V and should be performed by someone who understands the processes involved in generating software. Software products may be re-examined during all of the life cycle phases. Doggett (1983) has generated some checklists for reviewing software requirement specifications, the software design, and code review (see Appendix A). The checklists in Appendix A are representative of the questions that should be asked during detailed reviews.

Program testing may also be re-examined by a review team. Testing may be considered acceptable if:

- all program requirements have been tested
- anticipated program applications were tested
- primary design features and major logical paths were tested
- sufficient program validation tests were performed.

Testing is discussed in more detail in 3.2.4.

3.2.2 Walk-throughs

Myers (1979) reports that walk-throughs and inspection are two primary "human testing" methods which have been found to detect an average of 38% of total program errors. It is the mission of walk-throughs to identify errors, but not to correct them.

A walk-through is a team-oriented method that includes a walk-through team leader, a "tester," a programmer, and a secretary. The team leader coordinates the activities of scheduling the walk-through meeting, leading the meeting discussion, distributing materials, recording errors, and following errors to ensure that they are corrected. A "tester" performs desk-checking by reading through the code to check for errors. The programmer is the person who wrote the code. The secretary records the proceedings of the walk-through.

The normal procedure for the walk-through is to distribute program materials and allow the team several days to become familiar with the program to be tested. A walk-through meeting is scheduled, and during that meeting the "tester" presents test cases generated on paper. Myers (1979) states: "In most walk-throughs, more errors are found during the process of questioning the programmer than are found directly by the test cases themselves." In general the number of program errors will be more a function of the complexity of the task that the program is doing than of the capability of the programmer.

3.2.3 Inspections

Inspections, like walk-throughs, are an important part of V&V. Inspections have been found to be effective in identifying design and programming errors. They are also team-oriented activities. The inspection team may have the same member structure as the walk-through team except that the program's designer may be included if the programming and designing are done by different people.

The inspection meeting consists of the programmer/designer reviewing the program logic step-by-step. Usually this process will reveal program errors, primarily to the programmer. The second phase of the inspection

consists of team members examining the program for common errors. Common sources for errors are:

- data reference errors
- data-declaration errors
- computation errors
- comparison errors
- control-flow errors
- interface errors
- input/output errors (Myers 1979).

3.2.4 Testing

Reviews, walk-throughs, and inspections have been described and are, in part, testing mechanisms. The purpose of this section is to briefly describe the test plan, test-case design, module testing, generalized testing, debugging, and testing methodology.

3.2.4.1 Test Plan

Implementation of a computer-managed security system may be done:

- completely by the NPP utility, including hardware/software design and installation
- partly by an off-site vendor and partly by the utility
- completely by an off-site vendor.

Regardless of the approach chosen, a utility will be responsible for directing how testing and V&V efforts will be done. A test plan can be an effective tool to aid in system implementation. The test plan should include the following:

1. test plan description - This should include the program to be tested, the scope and objective of the test plan, and traceability to requirement specifications.
2. testing approach and evaluation - These include requirements tested, acceptance criteria, testing procedure, and test evaluation approach.
3. test problem definitions - These should include the basis for selection and test problem descriptions.
4. test case specification - The test case description is needed.

5. test data base plan - This should include the construction, update mechanisms, maintenance, and recording.
6. test procedures
7. planned reporting formats and procedures - These should include discrepancy reports, test case log, and test report (ANS 10.4 1985).

3.2.4.2 Test-Case Design

There is no test case that can be used to find all errors. If such a test case did exist, software testing would be a relatively simple task. It is not within the scope of this document to describe all test case designs in detail. However, two primary test case designs will be presented: "white-box" and "black-box" methodologies.

"White-box testing is concerned with the degree to which test cases exercise or cover the logic (source code) of the program" (Myers 1979). White-box testing is concerned with all internal mechanisms (decision code, loops, etc.) of programs. If it were possible to execute every path in a program, then the white-box testing objective would be satisfied. So far this has not been feasible.

Even as white-box testing is concerned with the internal program structure, black-box testing, as the name implies, is concerned with what a program output will be for a given input. Black-box testing seeks to discover, using exhaustive input testing, whether error conditions exist. Unfortunately, exhaustive testing is usually not feasible, and so black-box testing also fails to guarantee that all possible errors can be identified.

It might seem at this point that software testing is impossible because black-box and white-box testing taken separately fail to reach their objectives. In practice it has been found that black-box testing can be used to identify modules that contain errors, and then white-box testing can be conducted on "error-prone" modules to help discover error sources and other errors not found using black-box techniques.

3.2.4.3 Module Testing

Module testing is the testing of the modular components with which software systems are constructed. These components are subroutines, functions, procedures, etc. The purpose of module testing is the same as for all software testing: to discover errors and to associate them with unique programs.

Because software modules are designed to be limited in function, the number of execution paths are limited and white-box testing is the predominant testing mechanism. Two types of testing are routinely performed: top-down and bottom-up. Top-down testing tests modules starting with the first in a program module hierarchy and proceeding to lower-level modules. Conversely, bottom-up testing first tests the most primitive modules and proceeds up the highest level.

3.2.4.4 Debugging

Programmers can debug programs (find and correct errors) in several ways. Three debugging techniques are: using a core dump to locate errors, using the "wolf-fence" approach, and using automated debugging tools.

Core dump analysis was much more common in the early days of programming. Earlier generation computers usually allowed the user only 16K to 32K words in a working area, and one could review the core dump to discover what happened when an error occurred. The modern computing environment includes multi-megabyte addressing, virtual memory, multi-tasking, etc. Trying to read a 4-megabyte core dump is a ridiculous task and this method is seldom used.

The "wolf-fence" approach of locating program errors involves inserting print statements at the beginning and near the end of a program, and then using binary search techniques (using print statements) to converge on the "wolf" or error location.

Most modern compilers have internal debugging tools. These tools are automated to do things such as: print out the name of the current subroutine or function; select a variable and print its value whenever it changes; list sources where an array exceeds its dimensions, etc.

Other debugging approaches such as induction, deduction, backtracking, and testing rely on the human thought process. Myers (1979) has described these debugging approaches along with some error-locating and error-repairing principles that will be described next.

Induction involves proceeding from the particular to the general. Induction should include the following steps:

1. Locate the pertinent data.
2. Organize the data.
3. Devise a hypothesis that would explain the data.
4. Test the hypothesis.

Debugging by deduction is the process of proceeding from the general to the particular. This approach to debugging includes the following steps:

1. Enumerate the possible causes.
2. Use the data to eliminate unlikely causes.
3. Refine the remaining causes or hypothesis.
4. Test the remaining hypothesis.

Debugging by backtracking is a technique that has been effective in debugging small programs, and thus is potentially applicable for systems that have been modularly designed to keep program size limited. This approach is simply starting where the logic has failed, and working backwards until the error is located and corrected.

Debugging by testing is used as a tool with both inductive and deductive debugging. The purpose of the test cases is to generate information that will be helpful in locating errors. Each test case is concerned with a single condition or a few conditions. Test cases for testing are different than the cases for debugging in that testing seeks to cover many conditions with a small number of test cases.

The following are error-finding directives:

- THINK! (human reasoning debugging versus automated approaches).
- If you reach an impasse, sleep on it.
- If you reach an impasse, describe the problem to someone else.
- Use debugging tools only as a second resort.
- Avoid experimentation. (Use it only as a last resort).

The following are error-repairing principles:

- Where there is one bug, there is likely to be another.
- Fix the error, not just a symptom of it.
- The probability of the fix being correct is not 100%.
- The probability of the fix being correct drops as the size of the program increases.
- Beware of the possibility that an error correction creates a new error.
- The process of error repair should put one back temporarily in the design phase.
- Change the source code, not the object code.

3.2.4.5 Generalized Testing

The last area of testing to be addressed is "Big Picture" testing. After a software system has been developed and all the components (modules) thoroughly tested, testing must still be done to satisfy the end user that the desired product has been delivered. Another way of thinking about error is stated by Myers (1979): "A software error is present when the program does not do what its end user reasonably expects it to do."

The following testing approaches not discussed in detail, but rather just defined:

- function testing: the process of trying to find errors by comparing functional performance to a functional specification
- system testing: the process of trying to find errors by comparing a program performance to user documentation and program objectives
- facility testing: often a manual process to check each program objective to verify that it was implemented
- volume testing: an approach to require a system to process abnormally high volumes of data where the throughput rate is not critical (for example, activating all alarms simultaneously over an extended time period)
- stress testing: the process of subjecting a software system to heavy loads where the throughput rate is critical (for example, a computer managed security system where multiple alarms are exercised simultaneously)
- usability testing: testing of a software system where human factors are the primary considerations (for instance, testing the response effectiveness of a newly trained security system operator to an alert drill)
- security testing: tests designed to compromise a system's security protective features (for example, trying to bypass access protective mechanisms in a computer-managed security system)
- performance testing: tests designed to evaluate efficiency objectives such as response times and throughput rate (for example, testing card access stations in an NPP to simulate peak load ingress/egress)
- storage testing: test cases designed to test storage limitations for high data volumes, or special cases requiring temporary storage
- configuration testing: a process to test a system with minimum and maximum hardware configurations to check for corresponding errors (for example, operating systems, data base management systems, etc.)
- compatibility/conversion testing: testing that seeks to show that a replacement module does not meet program objectives
- installability testing: review of system installation to discover errors on startup
- recovery testing: strategy of allowing or causing a program (such as a data base program) to fail and then testing for error conditions on re-initialization

- documentation testing: tests seeking to discover errors in the accuracy of the user documentation
- acceptance testing: testing of a software system by the user or contracting organization to discover discrepancies between the original requirements, or contract, and the program's operation. For most cases where an NPP is acquiring a computer-managed security system, acceptance testing will be the predominant approach because most security systems will be contracted with outside vendors.

3.2.4.6 Testing Methodology

The testing methodology, or the approach used by NPP utilities, will be acceptance testing for the initial installation of a computer-managed security system. After a working system is implemented, changes to the software will be done in-house or via contract personnel. Testing for maintenance changes to the software may be done using one or more of the testing approaches just described. The following will probably be used by NPPs for testing computer-managed security systems: acceptance testing, facility testing, performance testing, and compatibility testing.

3.3 VERIFICATION AND VALIDATION OVER THE SYSTEM LIFE CYCLE

Verification and validation should be applied during the life cycle of a software system. The next four sections discuss how V&V is implemented for each stage in the life cycle.

3.3.1 V&V Planning During System Initiation

Section 2.2.2 of this document is an overview of the system initiation phase. The recommended approach for the development of a system should be documented in a plan. This plan should be oriented to system development and V&V activities. Verification and validation activity specification should include: 1) who will perform the V&V, 2) V&V and program development effort levels, 3) V&V activities to be performed, 4) resource requirements for each activity, 5) development and V&V testing responsibility assignments, 6) the development products that will be used with V&V, 7) a development schedule showing milestones and products, 8) the manner and frequency of V&V reporting, and 9) the way in which V&V findings will be implemented into the development system (ANS 10.4 1985).

3.3.2 V&V During System Design and Development

V&V should be included in the system design and development phase described in Section 2.2.2. V&V activities during the phase should be:

- V&V of design
- V&V of preliminary program documentation
- update of the test plan(s) to include design information

- V&V of the test plan(s)
- V&V of source code
- finalization of the test plan(s)
- development of the test data base(s)
- V&V of the test plan(s) and test data base(s)
- V&V of the integrated program (program components combined into an overall program)
- execution of the test plan(s)
- V&V of the test results.

3.3.3 V&V During System Installation

V&V during the system installation phase should include the following activities:

- V&V of the program installation (for example, the program as installed in the operational environment)
- V&V of the final program documentation
- preparation of the final V&V report.

3.3.4 V&V During System Operation

System operation will need V&V as changes to the system are implemented. Each discrete product that is changed needs to be checked for continuity of purpose and traceability. The reasons for changes in the final phase of the life cycle are changes in the product mission (new requirements emerge) or changes in the operating environment.

If new requirements emerge because operating errors are discovered, or the operating environment changes, or algorithms change, or performance requirements change, or new capabilities are added, then the software system will change and V&V is required to preserve system integrity. When changes are required, the following may need modification:

- requirements specification
- the system design
- the source code and associated comments and documentation.

Only the changed products require V&V activities. The associated test plan(s) should focus on those products.

3.4 AUTOMATED V&V

V&V automation produces the same advantages as automation for other applications:

- More can be done in a fraction of the time required for manual V&V.
- V&V can be done consistently (discipline).
- Exhaustive testing goals are possible.

Deutsch (1982) has described an automated verification system (AVS) and listed five basic functions:

1. source code analysis and database creation
2. static source code analysis to discover errors and identify software control and data structure
3. insertion of software probes into the source code to permit data collection on code segments executed and values computed and stored
4. test results analysis
5. generation of test assistance reports to help organize testing.

The AVS gets its information solely from the source code of the software system being analyzed.

The first function is to partition the source code by using conditional (decision) instructions (those instructions that determine which program instruction(s) will be executed next, based on possible parameter values). Then users of the system can determine how many times that specific blocks of code are executed during different test cases. A data base is employed to track these code blocks and to enhance the process of test thoroughness analysis. The second function is to generate a hierarchical tree structure that shows relationships among the program modules. This function allows rapid determination of which program module calls which module or module subset. Thus for specific test cases, the frequencies of module invocation and parameter changes may be tracked. In the third function, statements are inserted in the source code to monitor execution flow and record program performance statistics and signals. This will allow the automated V&V system to keep track of which statements are executed how many times, and how often program modules are called. The fourth function is the analysis of test results. For a specific test case the analysis will produce the following:

- statements or statement blocks executed
- the frequency of statement or statement block execution
- the data ranges for data variables

- frequencies of data exceeding expected ranges.

A number of automated V&V applications have been described in the literature, but are not reviewed in detail here. The primary objective of this section has been to report the existence of automated V&V systems and to give a brief introduction to the concept.

3.5 V&V AND NUCLEAR POWER PLANT COMPUTER-MANAGED SECURITY SYSTEMS

Generally, NPP utilities purchase computer-managed security systems from manufacturers and then use both their own programmers and contract programmers to maintain the software for their systems. For those sites visited, most software maintenance was done by utility staff with continual update and consulting support supplied by the original system suppliers.

V&V used by most utilities will be most helpful applied to:

- an existing product installed with changes made to suit site requirements
- software product changes as they occur.

V&V concepts reviewed in this document have been presented to address all phases in the life cycle. NPPs will probably be most interested in the V&V of existing programs because they will probably be purchasing a computer-managed security system as a finished product with specific enhancements being made to suit a specific site. This type of V&V is usually referred to as "posterior" V&V review (ANS 10.4 1985). This "posterior" V&V is normally done by first gathering all existing V&V data already generated for the system in question and then performing V&V for any part of the system which has incomplete V&V or none at all.

If a system has just been purchased and installed, acceptance testing needs to be performed by utility personnel. Essentially, what needs to be done is a reconstruction of products and requirements so that a V&V review can be done "a posteriori." The program and its documentation should be made available for review. The problem statement, requirements specification, design, and test plan and test results should be obtained.

A utility may elect to review program testing of the purchased system. This review should include:

- a review to verify that all program requirements have been tested
- a review to verify that anticipated program applications have been tested
- a review to determine that the design has been tested
- a verification that program logic is tested (ANS 10.4 1985).

Utilities may also choose to use automated testing programs to assure product V&V. V&V of existing systems is essentially a process to determine that an acquired system has been verified and validated as effectively as if it were a new program just being developed--but V&V now is "after-the-fact."

Also, as software maintenance is performed by utility or contract personnel, the products, after making changes, should have V&V practices applied even though the changes may be small. V&V of changed modules or new modules is limited in scope but is important to the software system because of the need for traceability and visibility.

Although V&V will cost time and money to implement effectively, it will improve the visibility of the software development progress and establish traceability-during the life cycle of the project. Software changes made well into a system's life cycle are much more costly than the same changes early in the life cycle. V&V, when properly implemented, will be an additional cost early in a software system's life but will pay dividends during the project life cycle by minimizing retrofitting. Retrofitting is caused by: 1) the discovery of errors in the design or 2) new requirements emerging later in the life cycle.

Design flaws caused by errors in the requirements specification or the requirements analysis can be particularly damaging because of resulting cost overruns on slipped schedules. V&V can be a valuable aid in eliminating most of these design flaws.

4.0 TRAINING

An integral part of the operation of a computer-managed physical access security system is the training that is provided to the operators of the system. Computer security awareness training is also important to the secure operation of the system. Each of these topics is discussed in this section of the report.

4.1 COMPUTER SYSTEM OPERATION TRAINING

Certain aspects of a general training program apply to the training of operators of a computer-managed security system. Four steps must be completed to provide a comprehensive training program:

- 1) defining scope and program requirements
- 2) research
- 3) materials preparation
- 4) testing and evaluation.

Defining the scope and requirements of a training program involves answering questions such as:

- What management objectives will this program meet?
- What specific requirements must be met?
- Are there additional needs identified by the individuals or plant supervision that are necessary to fulfill the job function?
- How much will the training program cost (in salaries, materials, overtime, etc.)?
- What are the criteria for selecting and training personnel?
- How many people need to be trained?
- How many people will be trained at any given time?
- What is the frequency of re-training or re-qualification?
- What are the time requirements for the training program?
- Will the time spent in training affect the quality or capability of the security force?
- What are the training documentation requirements that must be met for audit purposes?

This step gives the licensee the information needed to lay the framework for the training program. The objective is to train security personnel that have a combination of education, experience, health, and skills commensurate with their functional level of responsibility. Thus successful training provides reasonable assurance that decisions and actions of these security personnel during normal and abnormal conditions will be such that the plant is protected from the design basis threat (10 CFR 73.1a). For operators of the CAS/SAS this would involve specific training in the following areas:

- security officer training
- decision making
- computer operation and proficiency

These areas are discussed in more detail below.

The second step for developing a comprehensive training program, research, is necessary to cost-effectively establish a training program. After identifying what is needed in step 1, research into established training programs at similar facilities will prevent unnecessary re-development of training practices, procedures, and special equipment. Licensees have generally been willing to share ideas, concepts, and experiences dealing with computer-managed systems.

The third step, materials preparation, is dedicated to the development of a topical agenda, dividing the training program into modules of discrete skills and concept areas, determining objectives for each module, and writing lesson plans for each module. Using the research performed in step 2 and reference materials (such as training manuals, regulatory guides, specific NRC requirements, and literature reports), the program format is established. This format includes at least the minimum training standards. Every security force manager should strive for performance levels above the minimum to present an image of excellence to the NRC, to the public, and to the security personnel themselves.

The last step is to develop an effective testing and evaluation program that will assure that the training program successfully provides personnel who exemplify the desired high performance level.

4.1.1 CAS/SAS Officer Training

Training and qualification of security officers are necessary to assure that each officer knows the approved Security Plan, Safeguards Contingency Plan, security procedures, post orders, and details of the facility layout and operation. Training with respect to the Security Plan should be adequate to ensure that the licensee's security officers are fully aware of their duties and responsibilities under normal operating conditions, emergency conditions, and threat situations. In addition to security training, the training program should emphasize general employee training with special attention to certain aspects of plant operation, radiological and health safety, and quality assurance. This aspect of the CAS/SAS operator's

training assures that proper responses to system alarms are made in a timely manner. Criteria for a program for training security personnel have been identified (10 CFR 73 App B).

The CAS/SAS operators should be trained from a pool of skilled, assertive security personnel that have knowledge of each security post and the general operation of the security department. This knowledge is best obtained by working at each post and by being evaluated on post operation and responsibilities. This selection process assures that the CAS/SAS operators are intimately familiar with all post operations and their responsibilities.

The position of CAS/SAS operator requires special skills, attitudes, and ability to quickly apply a correct solution to any given situation. This last point is especially important and care should be taken to adequately provide an evaluation system in the training program to test this ability. Development of personnel for this position should be an important aspect of the security management responsibilities. Specifically, the security department needs to establish career paths that provide achievable opportunities to advance into the corporate structure (NUREG/CR-2297).

4.1.2 Decision Making

One of the important qualifications of personnel selected for CAS/SAS operator positions is the ability to make decisions. This consideration is not usually emphasized enough during personnel selection, and it may be the difference between otherwise qualified security personnel and those suitable for the CAS/SAS operator position. This assumes, of course, that the training and evaluation program has a mechanism to measure decision-making ability. If such a mechanism is not present, then the result could be an operator who cannot achieve acceptable security system performance.

The training program should stress oral response to simulated plant and security situations that are unusual. Part of an operator's decision-making ability can be improved by impressing on the operator the correct response to a given situation. As long as situations are familiar to the operator, then this method works effectively. (This fact emphasizes the need for periodic refresher training.) However, if situations involving major security responses occur infrequently or if an unforeseen situation arises, then the operator may be in an unusually high-stress condition that could affect judgement and the ability to properly communicate the situation to the proper plant or response personnel. For this reason the CAS/SAS operator selection process should include provisions for detecting people with inherent decision-making abilities and should provide the training to assure that these people are selected for CAS/SAS operators.

4.1.3 Computer Operation and Proficiency

The CAS/SAS operation requires some knowledge of the operation and maintenance of system components. Most of the operational aspects of a computer-managed security system can be learned through on-the-job training (OJT) working with an experienced operator. Common operations would be:

- acknowledging alarms
- assessing alarms
- entering personnel area access information
- checking personnel location status
- performing personnel accountability.

The ability of an operator to learn the operational system depends upon the difficulty of the machine-user interface. System interfaces, such as menus, function keys, touch screens, etc. have been addressed in a previous report (NUREG/CR-4298).

Maintenance operations that a CAS/SAS operator might perform include changing data storage tapes and printer paper.

An operator's proficiency depends on the time spent under actual or simulated conditions. Using a simulator that provides the operator with realistic event scenarios would be useful for training personnel in proper response without affecting plant security. The use of the simulator would be more beneficial to facilities that have opted for only one operator at each of the CAS and SAS locations. Facilities where two operators are stationed at the CAS might find that on-the-job training is sufficient for their needs and would not impact security as long as an experienced operator was immediately available.

4.1.4 Testing and Evaluation

The training staff should have at least one person dedicated to testing and evaluating CAS/SAS operators. In addition to the routine schedule for training and re-training operators, evaluations must constantly be made for all operators, including relief or emergency coverage operators. The operators must continually be observed for proficiency, decision making, fatigue, stress, morale, behavioral changes, and absenteeism.

Both written and oral tests should be given periodically to assure that the operators are knowledgeable and can handle stress situations. Written tests should dwell upon plant knowledge, normal operational details, operating procedures, post orders, etc. Oral examinations are useful for situation testing. Examiners would consider the responses of the tested personnel: Is the response appropriate for the given situation? Was the response timely? Alternatives should be developed for likely plant emergencies or threat situations, and discussions should be entertained that point out advantages and disadvantages of certain actions.

The development of a quality security training program is based on the ability of the training department to establish criteria and standards that can be used to evaluate both the overall performance of security personnel and the adequacy of the training program itself.

4.2 COMPUTER SECURITY AWARENESS TRAINING

Computer security awareness training is important to the secure operation of the computer-managed physical access security system. An effective computer security training program can help ensure that computer security officials and computer resource users acquire and maintain the necessary skills and knowledge to reduce the risks of security compromise.

The objectives to be met to achieve the overall goal of the computer security training program are:

- Make all employees aware of the scope and magnitude of the computer security risks and the potential impact on computer resources.
- Train computer resource users to identify actual and potential threats to application systems and the vulnerabilities of these systems.
- Train computer security officials to identify, prescribe, and design cost-effective safeguards for applications software and the computer installation.

Any training program should be designed to assure that continuity of operations will be maintained and that the system's integrity will be secure.

The individuals to be trained can be divided into two groups: 1) computer security management personnel and 2) computer resource providers and users. The resource providers and users include computer operations personnel, security personnel in CAS and SAS, computer system maintenance personnel, audit personnel, and the utilities personnel. Each of these groups require different training.

Top management within the utility needs to understand the fundamental computer security policies and the potential impact of computer fraud and abuse. Managers who are directly responsible for the operation of the system should have detailed knowledge in computer security disciplines that pertain to assuring continuity of operation (e.g., physical security, contingency planning).

Computer operations personnel are responsible for recommending methods and procedures to enhance physical and environmental security, contingency planning, data preparation controls, and media controls. Security training for maintenance personnel should include awareness presentations and instructions in their duties during emergency and contingency operation situations. Audit personnel require knowledge of computer security problems, risk analysis, contingency planning and computer security program auditing. Engineering personnel who are responsible for identifying power, air-conditioning, and structural design requirements, need to have a basic awareness of the environmental concerns of a computer security program under normal operating conditions. They should be knowledgeable in the areas of contingency planning, emergency management, and recovery operations. The personnel responsible for the routine day-to-day operation of the system should have training in data integrity controls, protection of systems

documentation, and programming practices that enhance security. Programmer/systems analyst personnel are responsible for identifying and documenting application systems and associated data security requirements, developing security specifications, programming security controls in application software, and performing software security certification tests. Programmers/systems analysts also need to have an understanding of the security requirements of systems and data back-up and recovery operations.

The overall computer security training program should meet the training requirements for each of the audiences just identified. Subject areas should range from general awareness training to specific courses in such areas as risk analysis, and programming practices for application system development.

Five subject areas are recommended to meet the computer security training needs of the utility personnel. These are:

1. Computer Security Awareness
2. Risk Management
3. Computer Security Program Management
4. Security for Sensitive Applications
5. Contingency Planning.

4.2.1 Computer Security Awareness

Training in this area should identify the major issues underlying the computer security problem, such as growing dependency on computers; need for accurate, efficient, and reliable systems; and the need to protect personal proprietary, classified, or otherwise sensitive information. Also, an overview of relevant cases of computer fraud and abuse should be reviewed to indicate the types of threats to computer resources and their vulnerabilities.

4.2.2 Risk Management

Training in this subject should address the steps necessary to conduct a risk analysis. Specific subjects should include instruction in defining: 1) the various risks that can affect the computer installation, 2) the consequences of various risk scenarios, 3) the degree to which risks can be controlled, and 4) the steps that are being, have been, or can be taken to reduce the occurrence, minimize the consequences, or transfer the impact of risks. Recommended subjects which should be included in this training area are:

- threat and vulnerability analysis
- risk analysis methodologies/approaches
- risk reduction analysis

- developing/implementing risk management plans
- physical security
- computer hardware security
- operating system security.

4.2.3 Computer Security Program Management

This training should cover the assignment of responsibilities and authority of those utility personnel specifically charged with management of the computer security program, the computer center, and computer security plans. Instruction should be provided on how to develop, implement, maintain, and audit computer security programs.

Recommended subjects which should be included in this area are:

- computer security management responsibilities
- developing, implementing, and maintaining computer security plans
- personnel security/clearances
- auditing computer security plans.

4.2.4 Security for Sensitive Applications

This training area should address the management and technical aspect of establishing a management control process to assure that appropriate administrative, physical, and technical safeguards are built into all existing and new applications as well as changes to existing applications. Specific instruction should address identifying sensitive applications, security planning for applications, developing security specifications, programming practices, and system tests and evaluations. Requirements and approaches for certifying and recertifying security controls in applications should also be addressed.

Recommended subjects which should be included in this training area are:

- criteria for the identification of sensitive applications
- software life cycle and security
- data integrity
- document protection
- defining application security requirements
- developing application security specifications

- programming practices for sensitive applications
- system test and evaluation
- certification of sensitive applications
- recertification requirements and approaches.

4.2.5 Contingency Planning

This training area should focus on the development and maintenance of plans for assuring the continuity of computer operations. Subjects to be included in this area should address the criteria and approaches for defining critical applications, alternative back-up strategies, and the selection of an alternate data processing operations site. Instruction should include a discussion of the planning steps involved in emergency response, backup operations, and recovery activities. Various methods for testing plans should also be discussed. Recommended subjects which should be included in this training area are:

- relationship between risk analysis and contingency planning
- identifying critical applications
- selection of a back-up strategy
- planning for emergency response
- planning for back-up operations
- planning for recovery operations
- documenting the contingency plan
- testing the contingency plan.

5.0 PREVENTIVE MAINTENANCE

Adopting sound hardware and software design and implementation practices will contribute significantly to the life cycle effectiveness for computer-managed security systems. The adoption of a methodology to assure availability of these security systems also needs to be considered as an important part of an overall program. The key concept here is preventive maintenance (PM): "An ounce of prevention is worth a pound of cure."

In this section PM is reviewed by addressing: 1) the required functions of PM, 2) alternatives from which NPP utilities may choose to incorporate a PM plan, 3) maintenance source selection criteria, 4) and finally, current trends for PM.

5.1 REQUIRED MAINTENANCE FUNCTIONS

Preventive maintenance methodology needs to address at least four needs:

- Develop an effective PM plan.
- Provide a maintainable computer system.
- Make available competent maintenance personnel.
- Use auditing to assure conformance to the PM plan.

The first task that a utility should perform is the development of the PM plan. This plan will need to:

- specify equipment elements (equipment inventory)
- develop a PM schedule
- specify detailed equipment checkout procedures (i.e., how components are tested)
- identify those components that may be taken off-line for checkout (printers, tape drives, terminals, etc.)
- identify those components which require "down-time" for checkout (disk drives, computer memory, input/output boards, graphics boards, etc.)
- describe procedures for PM conformance documentation
- specify who may perform which parts of PM
- provide continuing training if maintenance will be done "in-house"

- detail spare parts use and replacement procedures if these are provided "in-house"
- make available audit procedures and findings.

If computer-managed security system components cannot be supported, either because spare parts are not available or maintenance personnel cannot be found, those components should be replaced. Because of the trend toward modularity, component replacement should be a feasible solution.

For utilities using "in-house" personnel to do PM, continuing training programs to improve technical competence will provide long-term dividends. If third-party (contract) or vendor maintenance is provided, utilities will want to monitor the performance to assure conformance to the PM plan.

Spare parts control is one of the burdens imposed by "in-house" maintenance. Reliable sources will need to be identified so that spare parts needs can be met. Redundant computer systems and peripherals is an excellent way to provide readily available spare parts. Backup computer systems may have PM performed off-line and may also be placed on line when primary systems are taken off line for PM. All of the utilities visited to gather user survey data for this study used redundant computer systems. The redundant approach is an excellent way to minimize system down-time.

Auditing the PM documentation to verify conformance is an important aspect of PM. Auditing is also an excellent tool to identify weaknesses of the PM plan. The auditing may be done by utility personnel or by third-party agencies.

5.2 MAINTENANCE ALTERNATIVES

Computer system maintenance can be supplied by the vendor, the utility, a third party (contract), or any combination of these. Each of these maintenance alternatives offers advantages and disadvantages, and are discussed here in generalized form.

5.2.1 Vendor Supplied Maintenance

In general if vendor-supplied maintenance is available, this is a good approach because:

- The vendors have the expertise to repair their own equipment.
- Vendors can do PM on a service contract.
- No spare parts are required by a utility.
- A utility does not have to contract for the continuous manpower.
- A utility does not have to train technicians.

The following disadvantages of vendor maintenance should be considered:

- Immediate response to critical situations may be hampered by travel time, spare parts, shipping time.
- Contracts must be negotiated before service.
- If a utility has modified components, the vendor will be reluctant to service those devices.
- All equipment may not be supplied by the same vendor. In this case multiple-vendor maintenance or combination maintenance may be the best choice.
- The competence level of vendor-supplied technicians varies and cannot be controlled by the utility.

Vendor maintenance usually costs annually about 10 percent of the original equipment purchase cost (Collins et.al. 1981).

5.2.2 Third-Party Supplied Maintenance

Advantages and disadvantages of third-party (or contract) maintenance are similar to vendor maintenance with some exceptions. An additional advantage of the third-party approach over vendor maintenance is that multi-vendor systems may be combined into a single maintenance contract.

5.2.3 In-House Maintenance

A third alternative for computer-managed security system maintenance is "in-house" maintenance by the utility. In-house maintenance has the following advantages:

- The maintenance staff is most readily available.
- Contracts for service are not required.
- Mixed-vendor equipment are maintained by one maintenance group.
- Technicians can be trained for one specific system and so can become very familiar with that system.

In-house maintenance comes with its own price tag as shown by the following disadvantages:

- It is the most expensive of all alternatives. Personnel must be hired on a continuous basis and they must receive extensive training.
- A spare parts inventory must be maintained and controlled to support quick response.

- In general, in-house technicians are not as familiar (at least initially) with the equipment as the respective vendors.
- Utilities may not be able to justify the hiring of personnel comparable to vendor-supplied technicians.

5.2.4 Combination Approach

A fourth alternative is for utilities to adopt some combination of vendor, third-party, and in-house maintenance. Using the combination approach, utilities can analyze system PM requirements and maximize the benefit for each particular site. This approach allows utilities to focus on optimal solutions for each set of components in a computer-managed security system.

5.3 MAINTENANCE SOURCE SELECTION CRITERIA

Section 5.2 of NUREG/CR-4298 reviewed several approaches that can be used to select an optimal system. Utilities may choose to use one or more of those approaches (low-bid, fixed-price, evaluation-of-intangibles, and the analytic hierarchy process) to assist in deciding how PM will be performed. The selection process may be done, for example, by first grouping common components into separate subsystems and then analyzing which maintenance approach would be favorable for each subsystem.

If a utility does not favor the "combination" approach to preventive maintenance, then the approach may be decided by using the same selection criteria as for purchasing a system (NUREG/CR-4298, Section 5.2). This approach would be to arrive at a single PM source using the selection strategies just referenced.

5.4 CURRENT MAINTENANCE TRENDS

Current work in optimizing computer maintenance has been in five primary areas: 1) spacing for minicomputer systems, 2) diagnostic software for maintenance engineers, 3) hardware modularity, 4) disposable components, and 5) self-testing computers.

A system has been developed to provide computer systems with spare boards included in the computer system (Tolleth 1984). The selection of which boards to include as spares was based on a queuing model that used a time-dependent number of repairers, time between failures approximating empirical repair times, and the time required to switch in spares. This approach would add to the initial purchase cost of a computer system but should prove beneficial over the equipment life cycle.

Diagnostic software is not new to the computing industry and essentially relies on a computer system to find its own failure sources. Usually diagnostic software is executed in two modes. First, the computer system is shut down and rebooted with only the diagnostic software executing. Secondly, diagnostic software is executed routinely and concurrently with

other software to detect and flag system hardware faults. This second approach requires a computer system that is functioning well enough to execute diagnostic software, and it is relatively fast at locating fault sources. Diagnostic software can reduce costs (Chavez 1982).

The concept of hardware modularity has developed to support relatively rapid maintenance. Computer maintenance technicians simply replace boards that they suspect are the source of hardware faults. Diagnostic software just described is one tool to locate suspect boards. Once hardware failures have been corrected through board swapping, the failed or suspect boards can be delivered to a service center for repair and eventual re-use. This concept has directed design engineering of computer systems to produce "black-box" components (boards, power supplies, communications terminals, etc.) that can be quickly exchanged to maximize availability and simplify computer maintenance.

One consequence of the rapidly decreasing costs of computer components has been the development of "disposable" components. Often when the cost to repair a computer board is compared to the cost of shipping and repairing, it may be feasible to discard the board without repairing it (Walko 1981).

The final future maintenance trend is that of computer self testing. Most computer systems now do some self-testing when power is first supplied. It is conceivable that future PM may be done automatically (Dahler 1980). Self testing is very often done using read-only-memory because the self test software is not lost when power is turned off. Self testing will probably increase significantly in the near future and become an effective PM tool.

6.0 REFERENCES

ANS. 1985. Guidelines for the Verification and Validation of Scientific and Engineering Computer Programs for the Nuclear Industry. ANS 10.4.

Bersoff, E. H., V. D. Henderson, and S.G. Siegel. 1979. "Software Configuration Management: A Tutorial." Computer. IEEE.

Chavez, C.F. 1982. "Diagnostic Programming: Costs and Benefits". In Proceedings of Annual Reliability and Maintainability Symposium, Los Angeles, California.

Collins, J. C., S. B. Higgins, T. R. Harris, F. Levin, and T. G. Arnold. 1981. "Minicomputer Self-Maintenance: The Case for and Against." In IEEE Southeastern 1981 Conference Proceedings, Huntsville, Alabama, April 1981.

Dahler, G., and R. McGowan. 1980. "The Case for Self-Testing Computers." Mini-MicroSystems (USA). B (7): 97-99, 101, July 1980.

Deutsch, M. S. 1982. Software Verification and Validation: Realistic Project Approaches. Prentice-Hall Inc., Englewood Cliffs, New Jersey.

DOE 1984. ADP Internal Control Guideline. DOE/MA-0165, U.S. Department of Energy, National Technical Information Service, Springfield, Virginia.

Doggett, R. B., N. P. Wilburn, Z. E. Carey. 1983. Guidelines--Software Configuration Management. HEDL-TC-2263, Hanford Engineering Development Laboratory, Westinghouse Hanford Company, Richland, Washington.

Duerling, C. L. 1981. "An Automated Revision Control System for Software." In Proceedings of the IEEE 1981 National Aerospace and Electronics Conference, Dayton, Ohio, May 1981.

Energy Regulations. 1981. U.S. Code of Federal Regulations (CFR). Title 10, Parts 9 to 99, Chapter I, "Energy".

Part 73 - "Physical Protection of Nuclear Power Plants"

Part 73.55 - "Requirements for Physical Protection of Licensed Activities in Nuclear Power Reactors Against Radiological Sabotage"

Part 73.70 - "Records"

FIPS 1974. Guidelines for Automatic Data Processing Physical Security and Risk Management. FIPS PUB 31, Federal Information Processing Standards Publication, U.S. Department of Commerce/National Bureau of Standards.

FIPS 1976. Guidelines for Documentation of Computer Programs and Automated Data Systems. FIPS PUB 38, Federal Information Processing Standards Publication, U. S. Department of Commerce/National Bureau of Standards.

FIPS 1982. NBS FIPS Software Documentation Proceedings of a Workshop Held March 3, 1982 at NBS, Gaithersburg, Maryland. PB83-140632, U.S. Department of Commerce, National Technical Information Service, Springfield, Virginia.

GAO 1975. Comprehensive Audit Manual, Part I (revised). U.S. General Accounting Office.

Lewis, J. R., K. R. Byers, J. D. Fluckiger, K. D. McBride, S. C. Vickroy. 1985. Design and Installation of Computer Systems to Meet the Requirements of 10 CFR 73.55. NUREG/CR-4298 (PNL-5490), Pacific Northwest Laboratory, Richland, Washington.

Myers, G. J. 1979. The Art of Software Testing. John Wiley and Sons, New York, New York.

OMB 1978. "Security of Federal Automated Information Systems". Circular No. A-71. Office of Management and Budget, Washington D.C.

OMB 1983. "Internal Control Systems". Circular No. A-123 revised. Office of Management and Budget, Washington, D.C.

Petersen, K. L., and C. Flores, Jr. 1983. "Software Control and System Configuration Management: A Process That Works." In Proceedings of the IEEE/AIAA Fifth Digital Avionics Systems Conference, Edwards, California.

SAI. 1983. Software Development and Maintenance Guidelines. EPRI EL-3089. Prepared for the Electric Power Research Institute by Science Applications, Inc.

Schurman, D.L., G.H. Datesman, Jr., J.O. Truitt. 1982. Security Management Techniques and Evaluative Checklists for Security Force Effectiveness. NUREG/CR-2297. Applied Science Associates, Inc. Valencia, Pennsylvania.

Tolleth, D. W. 1984. "System Spacing for Minicomputer-Based Operation Systems." Atandt Bell Laboratory Technical Journal. (U.S.A), 13 (6), Part 2, July-August 1984.

Walko, J. 1981. "Cheapest Way to Repair Computer Boards -- Just Throw Them Away." Engineer. (GB), 252 (6525), April 16, 1981.

Zucker, S. 1983. "Automating the Configuration Management Process." In Proceedings of Softfair, A Conference on Software Development Tools, Techniques, and Alternatives. Arlington, Virginia, July 1983.

APPENDIX A

REVIEW CHECKLISTS

APPENDIX A

REVIEW CHECKLISTS

Summary checklists for reviewing requirements specifications, software design approach, and coding practices have been included here to provide guidance in developing an overall review plan. The following checklists are presented in summary form to give an overall view of what is needed for effective reviews. An NPP utility developing a review plan may want to refer to the complete checklists (Wilburn 1983) and should feel free to either expand the suggested checklist items or to delete those not applicable to a computer-managed security system for a specific site.

REQUIREMENTS REVIEW CHECKLIST

- A. Do the requirements conform with Company guidelines, including those for documentation?
- B. Do the requirements reflect an understanding of the problem to be solved?
- C. Are the requirements complete?

OVERALL

- 1. Are the products completely defined, and is there adequate documentation required?
- 2. Are documentation standards established for all deliverable and non-deliverable software?
- 3. Are all the user requirements adequately addressed?
- 4. Are human engineering requirements and problem areas identified?
- 5. Do the requirements include all functions implied by the Statement of Work?

MODELS/FUNCTIONS

- 1. Do the models/functions consider each type of information necessary for satisfying the user's need?
- 2. Do the functions carry out all the necessary transformation of information?

INPUT/OUTPUT

- 1. Are all the inputs and outputs of the computer program identified?

DATA

- 1. Are procedures for purging and updating all data bases identified?
- 2. Are data security and protection against loss of data provided for?
- 3. Are the logical data base and its access mechanisms defined?
- 4. Is each entity or relationship mentioned in the requirements also defined in the data dictionary and vice versa?

INTERFACES

1. Are the man-machine interfaces and operational procedures clearly defined?
2. Is adequate attention given to both the hardware-to-software and software-to-software interfaces?
3. Do the requirements identify all external interfaces and fully specify required program behavior with respect to each?

PERFORMANCE

1. Are performance requirements for each function described in separate paragraphs? Do these paragraphs include source and type of inputs, and destination and type of outputs?
2. Are requirements for system resource margins adequately specified?

ERROR PROCESSING

1. Are provisions made for transition to degraded or manual modes if the system/subsystem fails?
2. Are adequate provisions made for system backup and redundancy?
3. Are the software and hardware diagnostic capabilities adequate?
4. Is error processing logic described, giving performance when improper, incorrect, or out-of-range inputs are received?

ENVIRONMENT

1. Do the specifications tell what the computer program must do, how well, and under what conditions? Do they describe the environment in which it is to operate?
2. Have support tools, facilities, and the recruitment and training of support personnel been addressed?

CONSTRAINTS

1. Are explicit limits for the system (i.e., what it should and shouldn't do) defined, and are constraints identified?
2. Are timing and sizing constraints adequate?

- D. Are the requirements correct?
 - 1. Are all requirements consistent with the Statement of Work for the program?
 - 2. Are all requirements consistent with documented descriptions and known properties of the operational environment into which the program must fit?
 - 3. Do requirements concerning models, algorithms, and numerical techniques agree with standard references, where applicable?
- E. Are the requirements consistent?
 - 1. Are the models, algorithms, and numerical techniques that are specified mathematically compatible?
 - 2. Are the style of presentation and the level of detail consistent throughout the document?
 - 3. Is the mapping of software requirements from the System Specifications consistent, complete, and accurate?
- F. Are all requirements clear and unambiguous?
 - 1. Are the requirements organized and presented in a way that promotes clarity (for example, use of tables and lists in place of text, where applicable)?
 - 2. Are the requirements sufficiently detailed to prevent misinterpretation?
 - 3. Are requirements stated singularly, clearly, and concisely and could a reader understand the specification even though he is not completely familiar with terminology peculiar to software engineering?
- G. Are the requirements feasible?
- H. Do the requirements contain adequate provision for program verification and acceptance?
 - 1. Are the acceptance criteria consistent with use of any of the following:
 - Results obtained from similar computer programs
 - Classical solutions

- Accepted experimental results
 - Analytical results published in the technical literature
 - Benchmark problems.
2. Are all requirements traceable to, or derived from, a higher level specification, and is the source of each requirement or derivation shown?

DESIGN CHECKLIST

- A. Does the software design conform to the design documentation guideline and any other company guidelines?
- B. Is the translation of the requirements into the design correct?
- C. Have all the overall considerations in the design been dealt with?
- D. Is the design complete?

DATA BASE

- 1. Is the detailed definition of the data base content, structure, and control complete?

INTERFACES

- 1. Is the detailed definition and documentation of software internal and external interfaces completed?
- 2. Are all external interfaces well defined (control, support, test)?
- 3. Is the design compatible with external interfaces?

CONTROL LOGIC

- 1. Is the control flow clear and acceptable?
- 2. Is the program control system clearly outlined?

INPUT/OUTPUT

- 1. Are the specified data rates, accuracies, response time, and the like reflected in the design?

ALGORITHMS/MODELS/FUNCTIONS

- 1. Is the development and documentation of all required computational algorithms completed?
- 2. Do the algorithms satisfy accuracy, interface, timing and response requirements?

- E. Does the design have the desired quality?

- 1. Does the design specification reflect the quality criteria required in the requirements?

- F. Does the design have the required performance?
 - 1. Does the design contain adequate provisions for debugging, testing, and maintenance?
 - 2. Are the test requirements and test tools sufficient to determine whether all the functions meet requirements?
- F. Is the design complete and clear?
- G. Is the design feasible, i.e., achievable within allocated resources, schedules, and acceptable risks?
- H. Does the design appropriately meet all the constraints?
 - 1. Have all constraints specified by the requirements been met?

CODE REVIEW CHECKLIST

- A. Are there any DATA REFERENCE defects?
- B. Are there any DATA-DECLARATION defects?
- C. Are there any COMPUTATION defects?
- D. Are there any COMPARISON defects?
- E. Are there any CONTROL-FLOW defects?
- F. Are there any INTERFACE defects?
 - 1. Are the interfaces between routines, and to external devices, well designed and described?
- G. Are there any INPUT/OUTPUT defects?
- H. Have all possible OTHER CHECKS on the program been made?
 - 1. Is the program or module sufficiently robust?
 - 2. Are there functions missing from the program?
 - 3. Are there any syntax errors?
 - 4. Are there compiler diagnostics or errors?
- I. Is the program SELF DESCRIPTIVE? A software product is self-descriptive to the extent that it contains enough information for a reader to determine or verify its objectives, assumptions, constraints, inputs, outputs, components, and revision status.
- J. Does the software conform to existing STANDARDS, PROCEDURES, CONVENTIONS?
 - 1. Does the coding conform to ANSI standard coding language?
 - 2. Does the coding conform to company procedures?
 - 3. Is the documentation of the code complete, descriptive, consistent, and in conformance with applicable standards.
- K. Is the software consistent internally and with earlier base lines?
 - 1. Is the coding logically consistent with the design specifications?
 - 2. Have any functions been omitted, or have any extra functions been inserted, except as necessary coding considerations to support the given design?

3. Is the software coded in accordance with the requirements?
- L. Has the software been adequately tested by the developer?

REFERENCES

Wilburn, N. P. 1983. Guidelines -- Software Verification. HEDL-TC-2425, Hanford Engineering Development Laboratory, Westinghouse Hanford Company, Richland, Washington.

DISTRIBUTION

No. of
Copies

No. of
Copies

OFFSITE
LABORATORY

34 PACIFIC NORTHWEST

10 Dr. Sandra Fratalli
 Human Factors & Safeguards
 Branch
 U. S. Nuclear Regulatory
 Commission
 5650 Nicholson Lane
 Rockville MD 20850

 U. S. Nuclear Regulatory
 Commission
 Division of Technical
 Information and Document
 Control
 7920 Norfolk Avenue
 Bethesda MD 20014

K. R. Byers
M. D. Erickson
J. D. Fluckiger
N. L. Harms
J. R. Lewis (20)
K. C. McBride
R. J. Sorenson
C. S. Vickroy
Publishing Coordination (2)
Technical Information (5)

BIBLIOGRAPHIC DATA SHEET

SEE INSTRUCTIONS ON THE REVERSE

1. REPORT NUMBER (Assigned by TIDC, add Vol. No., if any)

NUREG/CR-4473
PNL-5680

2. TITLE AND SUBTITLE

A STUDY OF THE OPERATION AND MAINTENANCE OF COMPUTER
SYSTEMS TO MEET THE REQUIREMENTS OF 10 CFR 73.55

3. LEAVE BLANK

4. DATE REPORT COMPLETED

MONTH: November YEAR: 1985

6. DATE REPORT ISSUED

MONTH: January YEAR: 1986

5. AUTHOR(S)

J. R. Lewis K. R. Byers
J. D. Fluckiger K. C. McBride

7. PERFORMING ORGANIZATION NAME AND MAILING ADDRESS (Include Zip Code)

Pacific Northwest Laboratory
P.O. Box 999
Richland, Washington 99352

8. PROJECT/TASK/WORK UNIT NUMBER

9. FIN OR GRANT NUMBER

B2877

10. SPONSORING ORGANIZATION NAME AND MAILING ADDRESS (Include Zip Code)

Division of Risk Analysis and Operations
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 12555

11a. TYPE OF REPORT

Task Final Report

b. PERIOD COVERED (Inclusive Dates)

June 1985 - November 1985

12. SUPPLEMENTARY NOTES

13. ABSTRACT (200 words or less)

The Pacific Northwest Laboratory has studied the operation and maintenance of computer-managed systems that can help nuclear power plant licensees to meet the physical security requirements of 10 CFR 73.55 (for access control, alarm monitoring, and alarm recording). This report of that study describes a computer system quality assurance program that is based on a system of related internal controls. A discussion of computer system evaluation includes verification and validation mechanisms for assuring that requirements are stated and that the product fulfills these requirements. Finally, the report describes operator and security awareness training and a computer system preventive maintenance program.

14. DOCUMENT ANALYSIS & KEYWORDS/DESCRIPTORS

b. IDENTIFIERS/OPEN ENDED TERMS

computer security, security systems, power plant security, access
control, physical security15. AVAILABILITY
STATEMENT

unlimited

16. SECURITY CLASSIFICATION

(This page)
unclassified(This report)
unclassified

17. NUMBER OF PAGES

18. PRICE