Office of Inspector General

Report of Audit

MANAGEMENT OF APPLICATION SOFTWARE MAINTENANCE AT EPA

March 31, 1995

Audit Report E1NMF3-15-0072-5100240

Inspector General Division Conducting the Audit:	ADP Audits and Assistance Staff
Region Covered:	Agency-wide
Program Offices Involved:	OARM OAR OECA OPPTS OSWER OW

UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
WASHINGTON, D.C. 20460

MAR 31 1995

OFFICE OF
THE INSPECTOR GENERAL

MEMORANDUM

SUBJECT: Report of Audit--Management of Application Software Maintenance in EPA
Audit Report No. E1NMF3-15-0072-5100240

FROM: Kenneth A. Konz
Acting Deputy Inspector General (2410)

TO: Jonathan Z. Cannon
Assistant Administrator for Administration and Resources Management (3101)

Attached is our final report entitled "Management of Application Software Maintenance in EPA." This is part of a governmentwide effort to examine the management of software maintenance activities for computer-based information systems (i.e., application systems). Our office is leading this governmentwide effort involving eight Federal agencies under the auspices of the President's Council on Integrity and Efficiency (PCIE). The primary objectives of the audit within EPA were to evaluate the: (1) Agency's software maintenance policies and procedures; (2) Agency's management of application system software maintenance during the system life-cycle; (3) processes by which the Agency manages contractors' performance of application system software maintenance; and (4) quality and quantity of cost information on application software maintenance.

This audit report describes problems and recommended corrective actions the Office of Inspector General (OIG) has identified. The report represents the opinion of the OIG. Final determinations on the matters in the report will be made by EPA managers in accordance with established EPA audit resolution procedures. Accordingly, the findings described in this report do not necessarily represent the final EPA position.

In accordance with EPA order 2750, you, as the action official, are required to provide this office a written response to the audit report within 90 days of the final report date. For corrective actions planned but not completed by your response date, reference to specific milestone dates will assist this office in deciding whether to close this report. In addition, please track all action plans and milestone dates in the Management Audit Tracking System.

We appreciate your positive response to the recommendations presented in the report and the many actions you and your staff have initiated to improve the management of application software maintenance at EPA.

We have no objection to the further release of this report to the public. Should you or your staff have any questions about this report, please contact Gordon Milbourn, Acting Director, ADP Audits and Assistance Staff on (202) 260-7784.

Attachment

EXECUTIVE SUMMARY

PURPOSE

This audit is part of a governmentwide effort to examine the management of software maintenance activities for computer-based information systems (i.e., application systems). Our office is leading this governmentwide effort involving eight Federal agencies under the auspices of the President's Council on Integrity and Efficiency (PCIE). The primary objectives of the audit within EPA, as well as within the other participating agencies, were to evaluate the: (1) Agency's software maintenance policies and procedures; (2) Agency's management of application system software maintenance during the system life-cycle; (3) processes by which the Agency manages contractors' performance of application system software maintenance; and (4) quality and quantity of cost information on application software maintenance.

BACKGROUND

The General Accounting Office pointed out in a 1981 report that software maintenance in the government was largely undefined, unquantified, and undermanaged. These conditions may not have changed since definitive Federal requirements are still lacking.

During fiscal 1994, the Office of Management of Budget (OMB) estimated that the Federal government would spend over $25.2 billion on information technology. EPA ranked 13th among all Federal agencies in information technology expenditures, estimated at almost $300 million for fiscal 1994. EPA has over 500 information systems as well as computer models to support its mission. These systems and models incur operations and maintenance costs of at least $1 billion over their life cycles[1].

RESULTS IN BRIEF

Software maintenance is very costly. Each year, EPA spends almost $100 million operating and maintaining its information systems. Prudent, cost-effective management of this function is critical under any circumstances, but especially now, when EPA is being asked to cut expenses dramatically.

EPA has taken a number of steps in recent years to improve its management of application software maintenance. However, we found

[1] We used an estimated 12-year life cycle period. EPA's System Life Cycle Management policy (Chapter 17 of EPA Directive 2100) recognizes that the average life cycle of application systems is 12 years.

that most EPA managers do not really know how much this function costs, so effective decision-making is greatly hindered about things like what software changes to make, when to make them, and whether to replace old systems with new ones. Further, software maintenance is not adequately managed in areas such as recording and analyzing system failures to help identify needed improvements, and tracking changes dictated by new legislative mandates. Finally, the Government Performance and Results Act of 1993 mandates that the Federal government begin measuring its performance, but EPA generally does not have adequate performance measurement indicators for, tracking techniques for, or management involvement in the software maintenance process. Consequently, critical data could be damaged or lost, and costs for correcting software problems could increase beyond what is already being spent. Process improvements are needed to ensure that desired software maintenance outcomes are achieved, and better Federal guidance is needed, which we will address in our governmentwide report at a later date.

PRINCIPAL FINDINGS

Software Maintenance Function Is Not Adequately Managed

EPA has taken some significant steps to strengthen the management of the software maintenance function. For example, the Office of Information Resources Management (OIRM) recently issued Chapter 17 of EPA Directive 2100, which outlines requirements for managing application software maintenance. Additionally, EPA established the Systems Development Center (SDC) in 1990, which is an Agency activity which can serve as a model for promoting the best software maintenance practices.

Nevertheless, system managers for major information systems do not adequately manage the software maintenance process. Specifically, they do not: (1) monitor and record failures as corrective maintenance; (2) monitor and record changes which result from changes in legislation, hardware, or operating system as adaptive maintenance; (3) monitor and record changes which are made to a system to meet changing user needs as perfective maintenance; (4) monitor resource utilization; and (5) periodically review all software resources to identify and prevent obsolescence of software.

As a result, both system managers and senior program managers do not have the information needed to make critical decisions and to manage the risks associated with software maintenance. System managers cannot effectively set software maintenance priorities, manage resource utilization, or manage removal of software defects. Further, senior Agency managers do not have the management information they need to make a decision to maintain or replace a

major information system. As a result, major systems were not replaced until they failed to support program needs.

These deficiencies are attributable to a number of causes. Under current circumstances, system managers and program managers may not fully appreciate the extent of resources consumed by individual systems and, as a result, may not recognize software maintenance as an area which warrants attention and discipline. Additionally, EPA does not examine how software is maintained, exercise control over the process, and ensure effective software maintenance techniques and tools are employed. In addition, system managers do not have mechanisms to adequately track, record, and classify software defects (or failures). Further, Agency managers do not use maintainability or economic criteria for determining when to replace major information systems. Another contributing cause is inadequate Federal guidance, which is an issue we will address in our governmentwide report to the Federal oversight agencies. Nevertheless, prudent business practices would still necessitate individual Federal agencies establishing their own guidance in the absence of Federal guidance.

Software Maintenance Costs Are Not Available For Decision-Making

EPA is creating a Working Capital Fund (WCF), so that it can more cost effectively administer services, including ADP and telecommunications services. However, it is still questionable whether the WCF can separate application software maintenance activity from operations activity. We found that EPA did not develop, review, and update software maintenance costs by individual systems throughout their life cycles. As a result, EPA is not in a position to make informed system and budget decisions regarding systems operation and maintenance, which costs almost $100 million annually, and at least $1 billion over their life cycles. In addition, financial statements did not accurately reflect capitalization of software maintenance costs and some system costs were not accurately reported to OMB. These deficiencies are primarily due to the lack of a comprehensive process or system to accumulate costs.

Software Change Control And Configuration Management[2] Processes Are Not Adequately Managed

OIRM has taken a number of significant steps to implement controls over the management of software modifications to its application systems. For example, OIRM took the initiative to research and

[2] Software configuration is defined as an arrangement of software parts, including all elements necessary for the software to work. Configuration management refers to the process of identifying and documenting the software configuration and then systematically controlling changes to it to maintain its integrity and to trace configuration changes.

implement a software configuration management tool on its Integrated Financial Management System (IFMS). The product, ENDEVOR, will strengthen impact analyses for software changes, ensure version controls are in place, and force audit trails for emergency software changes. OIRM also recently updated its Change Management System (CMS) processes to improve the basic features and controls of the tracking system.

However, overall changes to major national EPA systems were not performed in a structured and controlled manner. The ten EPA application systems reviewed for change controls[3] had displayed varying degrees of weaknesses. In particular, EPA management did not use adequate performance measurement indicators, tracking techniques, management review techniques, quality assurance procedures, or supplemental software tools. In several cases, management involvement was limited to the initial stages of review and approval, with EPA management relinquishing control over the final test and review stages to contractor personnel. Overall, EPA management did not consistently or effectively control software changes.

As a result, continuity of system operations and orderly evolution of EPA's application systems cannot be guaranteed. Critical functional production problems, damage or loss of data and, most likely, additional unwarranted costs for corrective procedures could also result. Software changes could be processed without adequate audit trails, and unapproved, unintentional, or malicious modifications could be introduced and proceed undetected through the change process. Also, without measurement indicators, managers are unable to identify existing maintenance trends, detect unnecessary or inefficient maintenance, or make informed decisions regarding the future stability of the application system. In the end, implemented changes may not satisfy user requirements or may negatively impact the successful performance of other application functions.

The change control deficiencies are attributable to a number of factors, one of which indirectly relates to management's inability to view software operations and maintenance cost information on a system by system basis. Even in defined change control systems, most managers do not place sufficient importance on software modifications

---

[3] Aerometric Information Retrieval System (AIRS)
Comprehensive Environmental Response, Compensation, and Liability Information System (CERCLIS)
Contract Payment System (CPS)
EPA Payroll System (EPAYS)
Facility Index System (FINDS)
Grants Information and Control System (GICS)
Integrated Financial Management System (IFMS)
Permit Compliance System (PCS)
Resource Conservation and Recovery Information System (RCRIS)
Toxic Chemical Release Inventory System (TRIS)

which require limited program office resources and, therefore, controls over these numerous changes are often minimal. The overall attitude toward software maintenance is apparent when many system managers decline to use available Agency and Federal policies and guidance to define their change control processes. However, in some respects, neither EPA nor Federal guidance provide sufficient information to adequately manage maintenance projects, and many key management issues, such as the importance of version controls[4], are not addressed. In addition, Agency officials place too much reliance on contractor personnel by not building adequate oversight controls into the maintenance process. Program office coordination with contractor personnel during design, coding, and testing of changes is not adequate to control the quality and content of work performed.

RECOMMENDATIONS

We are recommending that the Assistant Administrator for Administration and Resources Management, in his role as the Designated Senior Official (DSO) for Information Resources Management (IRM), and, when appropriate, in conjunction with the Executive Steering Committee for IRM, promote a more consistent and structured approach to managing application software maintenance across the Agency. This should include the establishment of mechanisms and practices for application maintenance to ensure valid performance measurement, accountability for costs, adequate management review, and quality control.

AGENCY COMMENTS AND OIG EVALUATION

In a memorandum dated March 17, 1995, the Assistant Administrator for Administration and Resources Management responded to our draft report (see Appendix I). To provide a balanced understanding of the issues, we have summarized and commented on the Agency's position in appropriate locations throughout our report.

Productive discussions with Office of Administration and Resources Management (OARM) representatives resulted in a revised set of recommendations which alleviated some of OARM's major concerns and yet adequately addressed the conditions noted in our draft report. In summary, the Agency agreed with sixteen of the twenty-one revised recommendations in our draft report, partially agreed with four recommendations, and disagreed with one recommendation. In addition, the Agency initiated action on six recommendations. Additional

[4] Version controls allow program developers and maintainers to locate the latest version of a software program accurately, reliably, and consistently. Version controls also enable system managers to roll-back to prior operable configurations of an application should a newly modified version fail to operate correctly once installed in the production environment.

concerns which relate to the four partially agreed upon recommendations are addressed in the individual chapters. Considering the National Institute of Standards and Technology's (NIST) recent announcement of their intention to rescind a number of Federal Information Processing Standards (FIPS) publications, we have withdrawn recommendation 4-5 with respect to the IRM Policy Manual. However, the planned revisions of the Operations and Maintenance (O&M) Manual need to adequately address the topic of independent V&V testing and stipulate thresholds for implementation which would clearly define the level of effort and other criteria used to determine which software changes are subject to Verification and Validation (V&V) testing.

The Agency response expressed concern about several of broad issues:

- Agency officials considered our recommendations for additional policies or procedures to be in direct conflict with EPA's efforts to reduce internal mandates. As previously stated, we updated our recommendations to limit the implementation of additional formal policies and procedures. We firmly believe that while efforts to reduce *unnecessary* EPA mandates have value, improvements are still needed in critical areas where *weak*, or *no*, guidance exists.

- Several policy-related recommendations focused on improving internal Agency processes, which EPA officials believed was contrary to the mission orientation of the National Performance Review (NPR) and the Government Performance and Results Act (GPRA). The revised recommendations eliminated, whenever possible, the need for creating additional Agency processes. We agree with the emphasis of NPR and GPRA on the importance of outcomes. Where EPA's software maintenance processes need to be made more effective to help ensure that desired outcomes are achieved, we have retained our recommendations.

- The Agency took exception to the statistical data used in Chapter 2. They disagreed with the factual basis of those audit conclusions which, in their opinion, formed the core of the draft report. Although we had already stated it in the draft report, we nevertheless modified Chapter 2 to more strongly acknowledge that the lack of a measurement program also meant that the data we used for analysis contained weaknesses. The message resulting from the analysis should not be focused on the details of individual application systems, but rather on the need for a consistent set of measurements, which EPA does not have.

- Agency officials believed the report overemphasized the need for mandatory Agency implementation of Federal guidance that is actually optional. We recommended implementation of Federal

guidelines because they offer relevant information on techniques, procedures, and methodologies to improve the maintainability of a software system. The rapidly evolving nature of this Agency's information systems makes it imperative that the best available practices are used to control that development, whether mandatory or not.

- EPA officials believed that much of the report was aimed at the efficiency of internal processes. Therefore, they were uncertain whether adopting the recommendations will really improve mission accomplishment. In our opinion, the efficiency of a process is linked with its effectiveness, so the efficiency of the Agency's internal software maintenance processes is important to achieving dependable and desirable end results.

Overall, the Agency agreed that more attention and discipline should be placed on software maintenance activities, but was concerned that greater benefits might result from improvement in other areas of information resources management.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

PURPOSE

This audit is part of a governmentwide effort to examine the management of software maintenance activities for computer-based information systems (i.e., application systems). Our office is leading this governmentwide effort involving eight Federal agencies under the auspices of the PCIE. This effort is Task 4 of the PCIE Computer Systems Integrity Project (CSIP) which is a multi-task review of controls, security, and other integrity issues related to the data processing systems life cycle. Our involvement includes reviewing EPA's management of the application software maintenance process, as well as preparing a consolidated governmentwide report on the results of reviews in this area conducted by participants from other agencies.

The PCIE selected the application software maintenance area as Task 4 of CSIP based primarily on two reasons. First and foremost, inadequate control of software maintenance exposes an organization to corruption of system information that can cause erroneous management decisions and an inability to meet organizational missions. Second, without the ability to quantify costs of software maintenance, managing it becomes a formidable task. The General Accounting Office (GAO) pointed out in a 1981 report[5] that software maintenance in the government was largely undefined, unquantified, and undermanaged. This condition may not have changed, since definitive Federal requirements for controlling software maintenance costs are still lacking.

The primary objectives of the audit within EPA, as well as within the other participating agencies, were to evaluate the: (1) Agency's software maintenance policies and procedures; (2) Agency's management of application system software maintenance during the system life-cycle; (3) processes by which the Agency manages the performance of application system software maintenance; and (4) quality and quantity of cost information on application software maintenance.

[5] Report AFMD-81-25, titled "Federal Agencies' Maintenance of Computer Programs: Expensive and Undermanaged."

BACKGROUND

The 1981 General Accounting Office report cited that Federal agencies spend millions of dollars annually on computer software maintenance, but little is done to manage maintenance. In spite of the high cost, agencies have a very limited overview of their software maintenance operation and have made little concentrated effort to effectively manage and minimize the resources required to maintain their computer software. The report further cited that ADP managers have done little to identify common causes of maintenance problems or to take action to reduce maintenance costs. Managers generally have neither cost accounting data nor management data on software maintenance activities and thus know little about how much maintenance really costs, or which types of maintenance cost the most. Additionally, agencies have established neither goals or standards to measure the efficiency of their maintenance operation nor criteria for acceptable maintenance costs.

During fiscal 1994, OMB estimated that the Federal government would spend over $25.2 billion on information technology. EPA ranked 13th among all Federal agencies in information technology expenditures, estimated at almost $300 million for fiscal 1994. EPA has over 500 information systems as well as computer models to support its mission. These systems and models incur operations and maintenance costs of almost $100 million annually, and at least $1 billion over their life cycles.

SCOPE AND METHODOLOGY

The primary focus of this audit was on the overall management and cost tracking of application software maintenance within EPA. The audit fieldwork was conducted from May 1993 to November 1994, primarily at EPA Headquarters, Washington, DC and the National Data Processing Division (NDPD), Research Triangle Park (RTP), North Carolina. We selected 11 major application systems[6] in 6 program offices (see Appendix II) for review to determine a representative Agency-wide approach toward the management of software maintenance. See Appendix III for detailed discussion regarding our methodology.

In the spirit of the Integrity Act process and in response to the National Performance Review recommendations to the Inspector General community to focus more on recommended improvements and less on effects, we took a proactive approach during this audit. Specifically, we concentrated on internal control improvements to

[6] Ten application systems were reviewed under each section of the audit on software maintenance. However, the selection of the ten application systems differed slightly for each major audit area, due to varying circumstances. The circumstances are explained in detail in Appendix II. In total, this report identifies findings related to eleven EPA information systems.

offset potential adverse effects of identified conditions rather than prolonging the audit to identify actual adverse effects.

It is possible that some of the effects we identified could be mitigated through the use of compensating management controls or that some of the application systems reviewed employed adequate compensating controls to mitigate the unfavorable occurrences we described. However, this claim could not be made for all of the application systems reviewed and, therefore, the effects identified depict real and potentially damaging situations which cannot be overlooked. Furthermore, in many cases we could not produce actual examples of damaging effects because the lack of adequately detailed and formatted historical data regarding software maintenance activities prevented a thorough analysis of specific conditions.

We conducted this audit in accordance with Government Auditing Standards (1994 revision) issued by the Comptroller General of the United States. Our audit included tests of management and related internal controls, policies, standards, and procedures specifically related to the audit objectives. Because this review disclosed EPA's management of application software maintenance as a material weakness, we also reviewed the Integrity Act evaluation process for OARM, the Office of Prevention, Pesticides and Toxic Substances (OPPTS), the Office of Solid Waste and Emergency Response (OSWER), and the Office of Water (OW) to determine why these weaknesses were not identified internally. No other issues came to our attention which we believed were significant enough to warrant expanding the scope of this audit.

PRIOR AUDIT REPORT COVERAGE

A September 28, 1994, OIG audit report, entitled "EPA's Integrated Financial Management System" identified application software maintenance and cost tracking deficiencies related to IFMS. The report cited the following as contributing causes to escalating costs and delays relating to the implementation of IFMS: (1) not adequately following a generally accepted system development life cycle (SDLC) approach; (2) over-customization of the off-the-shelf software; and (3) lack of a comprehensive process or system to accumulate costs. The report made 16 recommendations to address the software maintenance, cost tracking, and other problems identified by the audit. Actions were taken or are currently underway to implement the recommendations.

A March 24, 1994, OIG special review, entitled "Special Review of EPA's Information Systems Program" identified application software maintenance deficiencies. Specifically, the report cited that:

The vast majority of Headquarters program and Regional offices do not use a standard approach to manage software development and maintenance. The primary reason appears to be the use of contractors for the majority of EPA's systems development projects; these contractors are allowed to use their own methodologies, which vary widely and do not always meet Federal requirements. Further, some of the steps in this standard approach are perceived to add time to the development of a system, so some organizations tend to perform them cursorily or not at all. Participants in one focus group summed it up by stating that "management often will not let us do it right, but there is always time to do it again". Part of this process involves assessing whether older systems are obsolete, something EPA does not consistently do.

The report made 35 recommendations to address the software maintenance and other problems identified by the review. Actions were taken or are currently underway to implement the recommendations.

CHAPTER 2

SOFTWARE MAINTENANCE FUNCTION IS NOT ADEQUATELY MANAGED

SOFTWARE MAINTENANCE REQUIREMENTS AND GUIDANCE

We used the following Federal requirements and guidance to conduct our audit in the area of software change control and configuration management. Federal guidelines, as well as a number of industry publications, were used to form a framework of sensible, stable business practices and, therefore, served as a means to evaluate software maintenance activities. Appendix IV contains a more detailed discussion of this Federal criteria.

- Public Law Nos. 99-511, 99-591;
- OMB Circular Nos. A-130, A-132;
- Government Performance and Results Act of 1993;
- FIPS Publication No. 106;
- General Services Administration (GSA) Guide For Acquiring Software Development Services, Chapter 16, Software Operation and Maintenance; and
- EPA Directive 2100, entitled "Information Resources Policy Manual."

NEED FOR A SOFTWARE MEASUREMENT PROGRAM

EPA has not established a software measurement program. As a result, hard data about software maintenance was not available. The lack of a measurement program also means that the data we have used for this analysis contains weaknesses. One weakness is that data identified to any particular application system may in fact have little or no relationship to that application. In the absence of established measurements, we used FIPS 106 guidelines to provide a framework against which we could evaluate software maintenance activities. These measurements should lead to a lively debate about what should be measured, and how to properly identify and aggregate data.

The message from this analysis is not about the details of individual applications, and it is not about the accuracy of the underlying data. Any data that is available can be used to find indicators of system performance. The measures that we have used do not represent

an "ideal" measurement program, but do help to illustrate the difficulty in obtaining valid measurements in the current environment. However, the weaknesses in the underlying data do not prevent us from using the data as if it were accurate in order to *illustrate* various uses of measurement for managing software maintenance, system reliability and efficiency, or the need to replace the system.

In addition, EPA has taken some significant steps to strengthen the management of the software maintenance function. For example, OIRM recently issued Chapter 17 of the EPA Directive 2100, which outlines requirements for system life cycle management, including the management of application software maintenance. Additionally, EPA established the SDC in 1990, which is an Agency activity which can serve as a model for promoting the best software maintenance practices.

System Managers Do Not Monitor And Record Job Failures As Corrective Maintenance

During the maintenance and enhancement phases of the SDLC, both user satisfaction and defects, i.e., something wrong that needs to be fixed, should be measured. It is at this point that it becomes possible to carry out retrospective analyses of defect removal efficiencies of each specific review, inspection, and test, and of the cumulative efficiency of the overall series of defect removal steps. However, based on the information available, many system managers for the ten systems in this audit do not routinely record job failures[7] and do not distinguish between corrective, adaptive, and perfective maintenance. They do not normally evaluate job failures for trends or identify problem programs or jobs.

Dr. Bill Hetzel of the Software Practices Research Center defines the basic building blocks of any bottom-up software measurement effort as:

- Resource tracking: estimating and tracking resource use, tasks, deliverables, and milestones;
- Work product tracking: tracking and control of source code and document versions and changes; and
- Problem tracking: tracking and control of problems, defects, and open issues.

[7] For the purposes of this discussion, a failure was defined as an event which resulted in loss of system use for any amount of time or which resulted in a data base restore. Abnormal job terminations (ABENDS) and Job Control Language (JCL) errors represent failures according to these definitions. Each ABEND and JCL error represents the requirement to fix the error and then resubmit the job. Fixing the error may be an operational issue for JCL errors and data errors, but fixing ABENDS will normally involve corrective maintenance.

The use of defect counts is common to most measurement programs. Problems are entered into a tracking system to make sure they are not lost or forgotten. Whenever a problem is discovered, a problem record is "opened" and basic data about the problem (i.e., activity or phase, symptom, suspected cause, and type or classification) is recorded. After the fix is "approved," a change is prepared and tested, and the problem can be "closed." Upon closing, additional data is usually entered into the record (i.e., actual cause, source of the problem, and effort to fix).

We asked system managers how many failures they had experienced during calendar year 1992. The data available to us suggests that system managers do not consider abnormal job ends as failures, and thus had no idea of the extent of the job failures. Most thought that their system had less than twenty failures during 1992. We used the MVS Integrated Control System (MICS) records of abnormal job ends (ABENDS) and job control language (JCL) errors in this analysis as an indicator of job failures, and arrived at a much higher failure count. While the number of failures for each system varies widely, as an average there were 3,171 JCL errors and 4,057 ABENDS for each of the ten systems during 1992.

The tracking systems used by system managers do not provide the comprehensive data that Dr. Hetzel's model would provide. System managers have established systems for tracking production jobs and reporting status at the completion of the job, but these are strictly operations activity logs and are not effective problem tracking systems. Job failures are evaluated and, when possible, corrected and resubmitted. Failures which are corrected and resubmitted are only recorded on the job tracking logs as job submissions, and not as problems, even though resources were used to correct the failure. Failures which cannot be easily corrected are referred by the maintenance staff to the program office staff for further analysis and corrective action. However, system managers do not have systems which record historical data on problems incurred to identify trends and problem programs or jobs. Current historical data does not distinguish system-related ABENDS and JCL errors from those resulting from user-created non-production jobs. Any problem tracking system of measurement must be able to account for all ABENDS and JCL errors if they are to be appropriately classified and analyzed; problems cannot be collectively considered irrelevant on the basis that their origin is inconclusive.

Figure 1 shows mainframe data for ABENDS (both system and user abend codes) and JCL errors as a percent of all jobs submitted for the system. All numbers are taken from MICS records for 1992. All of these systems exhibit error rates that are higher than expected for mature production systems, although one system, RCRIS, was an immature system during the period reflected in the graph. In a production system, it is expected that JCL errors are corrected early and that the rate of JCL errors would be approaching zero. In our opinion, we would also expect management attention when ABEND rates exceed two percent. Seven of the ten systems we reviewed exceed these rates. The other three systems have ABEND rates lower than two percent, but combined ABEND and JCL error rates between 3.25 and 3.75 percent.

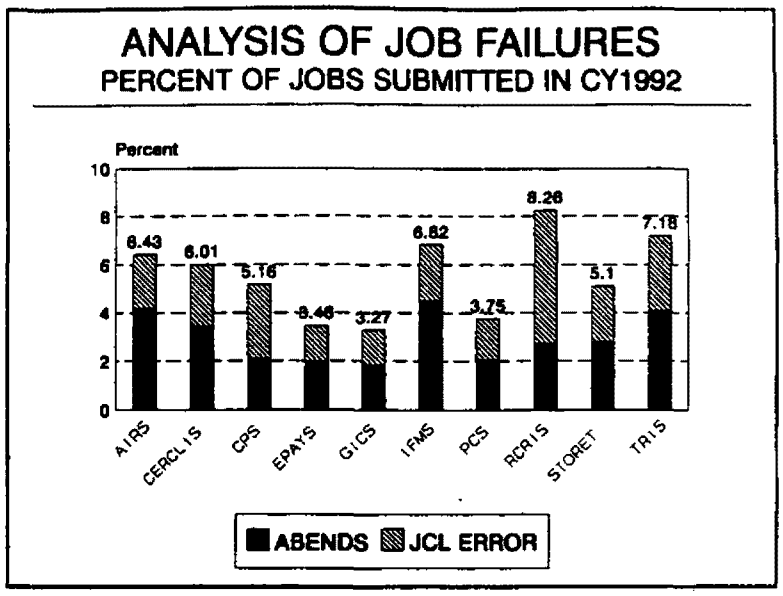

Figure 1 Reported Problems and ABENDS

Figure 2 shows the monthly record of JCL errors and ABENDS for four of the ten systems, and demonstrates the inconsistency of software maintenance management, as well as the volatility of the systems. The EPA Payroll System (EPAYS) is a mature, robust system and the graph reflects consistency and stability. The Resource Conservation and Recovery Information system (RCRIS) completed implementation in December 1991, and the graph represents the first year of production. The basic quality of the software is reflected in a low and declining ABEND record. RCRIS implemented approximately 1,200 changes in 1992, and is one of two systems in our study in which the JCL error rate exceeded the ABEND rate. Again, this is an indication of a new application system with inexperienced or untrained users. The Toxic Chemical Release Inventory System (TRIS) was extensively modified in 1992 to accommodate new required data elements for the Pollution Prevention Act, but its JCL Error and ABEND rates were more erratic. This suggests that the change management process was not as rigorous as for RCRIS. The graph for the IFMS family of systems reflects significant system difficulties, perhaps as a result of implementing approximately 3,000 changes during the year.

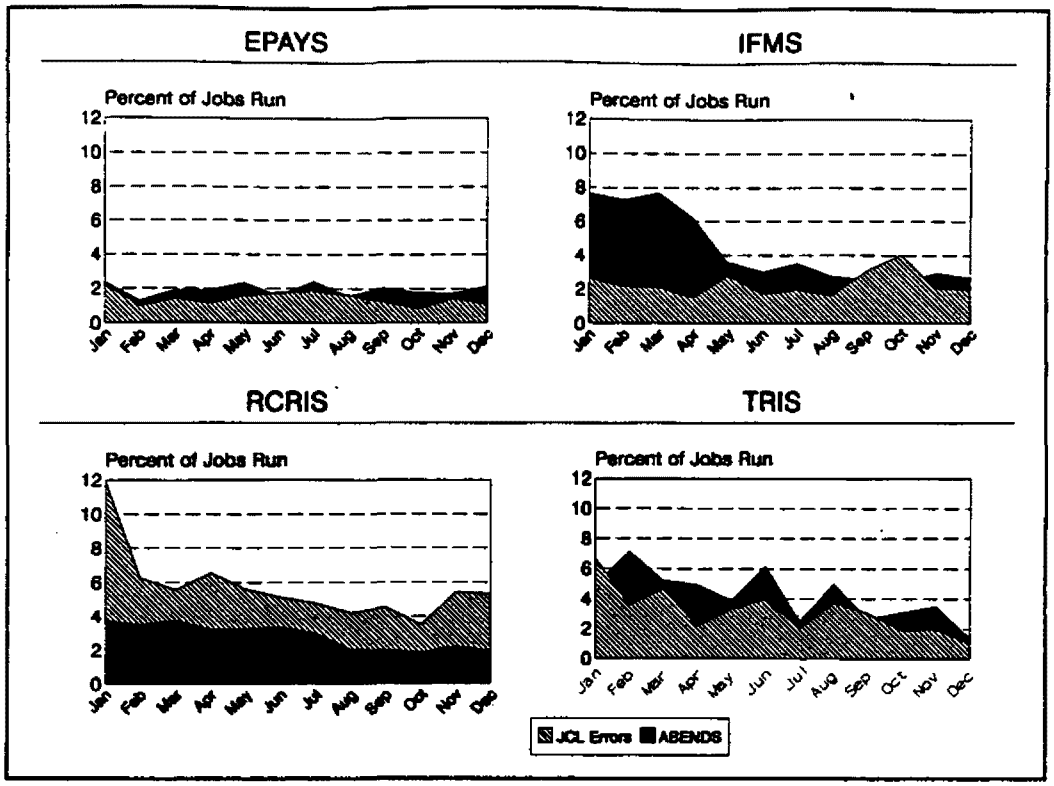

Figure 2 Percent Failures By Month CY1992

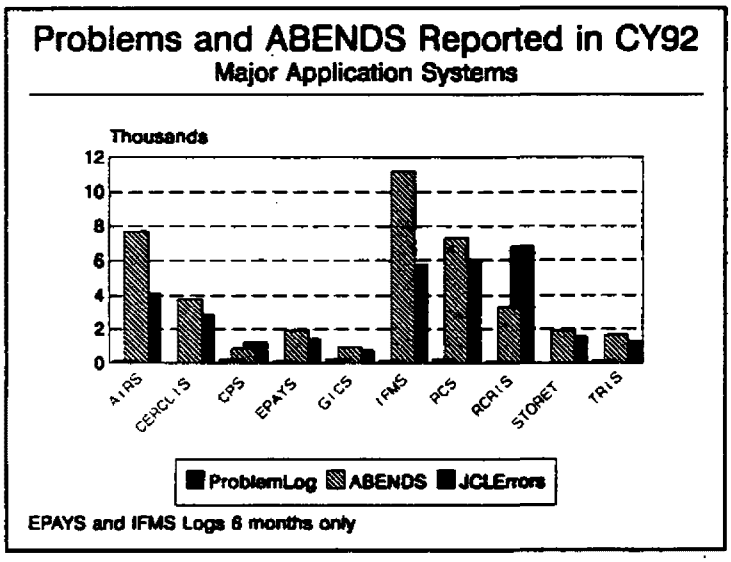

Figure 3 shows the problems recorded on Problem Logs for the ten systems and the number of ABENDS indicated in the system management data for 1992. Only two of the systems report more than two hundred problems logged during 1992. However, while eight of the systems had more than 1,000 ABENDS, very few ABENDS in any application system are recorded as problems. However, in some cases, application system managers maintain multiple logs to track system problems. It is possible that some of these problems may be logged elsewhere, but were not included in the system logs provided during the audit.

Figure 3 Problem Logs and ABENDS

Figure 4 compares the changes reported on the change control logs for each of the ten systems with the number of modifications reported by system managers for 1992. Four systems reported implementing more than one thousand changes during 1992. This includes RCRIS which completed fifteen months of implementation in December 1991, and then processed more than one thousand changes during the first year of operation. This figure illustrates the major inconsistency between offices or systems managers with regard to how changes are counted. Some system managers count each individual module change as a change. Other managers group modifications into a major change, release or version, and reflect only the version on the change control log. And still other system managers do not maintain change control logs.

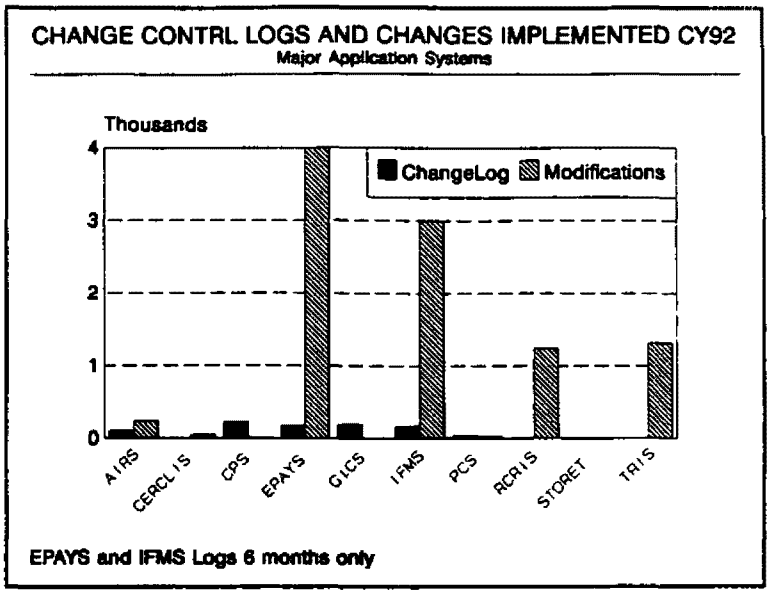

Figure 4 Change Control LOGS versus Reported Changes

Most System Managers Do Not Record "Environmental" Changes As Adaptive Maintenance

Most system managers do not monitor and record as adaptive maintenance software changes corresponding to environmental changes in laws and regulations, system software configuration, and hardware configuration, and do not distinguish between corrective, adaptive and perfective maintenance. Adaptive maintenance is performed in response to an external event, and must be performed so the system will be compatible with its environment. However, in practice these external changes are seldom accomplished without extensive coordination. A manager who can demonstrate the impact of a proposed change with accurate schedule and cost data may be able to influence the timing of the change or obtain additional resources necessary to meet the schedules.

For example, regulatory changes have a direct impact on the types of changes to RCRIS. While OSWER officials maintain detailed cost records, they are unable to identify costs associated with changes directly attributed to regulatory actions without significant analysis of their data. They also indicated that the lack of coordination between those individuals writing and issuing policy and those functions directly affected by the policy is a major source of frustration not only for IRM personnel, but for enforcement personnel as well. RCRIS costs between 1985 and 1991 totaled $18.3 million for

design, development, implementation and initial operation. First year costs alone for operation and maintenance were almost $3.4 million.

Regulatory changes also have a direct impact on the types of changes to TRIS. Projected maintenance costs alone for the six years since implementation are $7,200,000. OPPTS reports that the following factors require significant effort to alter TRIS to accommodate the changes: the system is required to produce an exact replica of an industry submission; and approximately 85 percent of the 250 program modules must be changed within a very short time before the first forms are received from industry. All contract expenses fall into one object class and OPPTS does not maintain detailed cost records, so it is unable to identify costs associated with changes directly attributed to regulatory actions. TRIS cost approximately $300,000 to develop in 1988, and has cost approximately $3,500,000 per year to operate and maintain since 1988.

System Managers Do Not Monitor Resource Utilization

Monitoring resources utilization will become much more important to system managers in the future with the implementation of the working capital fund -- which will encompass charge back to the program offices for resources utilization. However, the data available to us suggests that system managers do not monitor resource utilization. We asked the system managers for the ten systems to provide monthly processing hours in their response to our vulnerability questionnaire. We defined processing hours as CPU hours for consistency.

Figure 5 compares monthly average CPU time with the CPU hours reported by the systems managers. We developed an average monthly CPU time by dividing the total CPU hours used in 1992 as reported in the MICS system by twelve. We compared reported processing times with the average of actual monthly CPU hours for 1992. The results indicate that most system managers do not have a realistic view of the amount of CPU time used each month by their system.

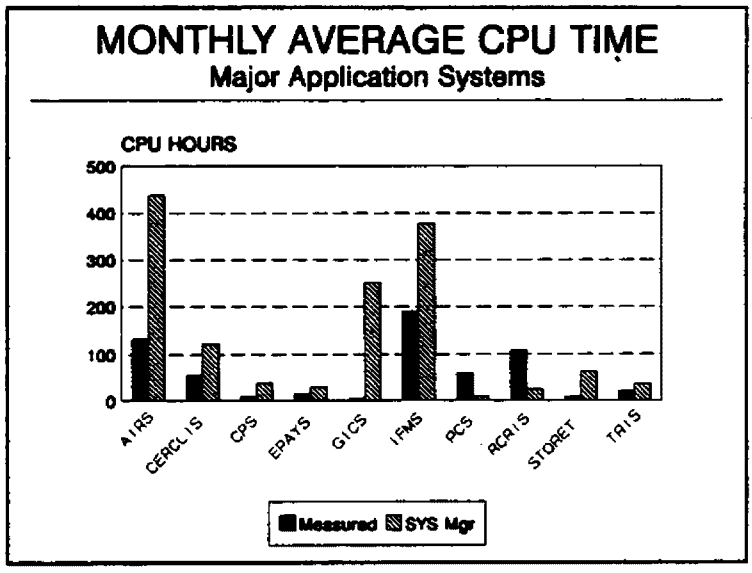

Figure 5 Reporting Variance

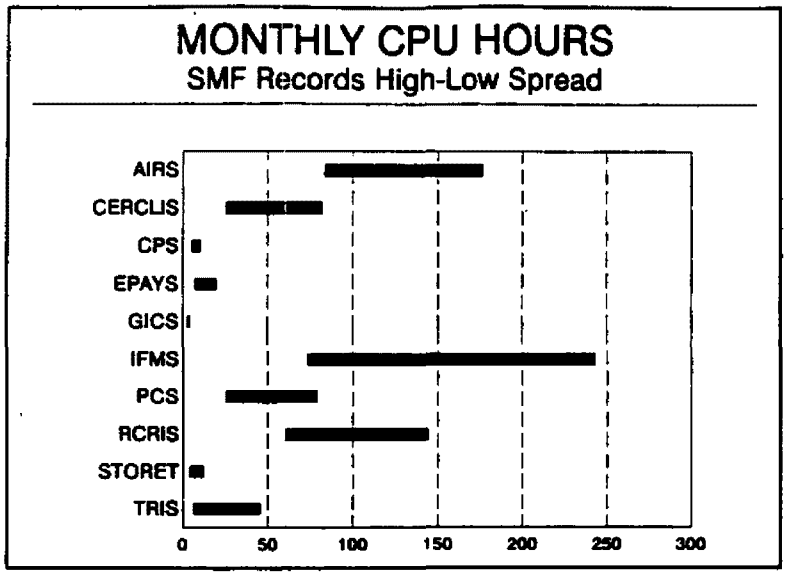

Figure 6 shows the significant variation in CPU usage from month to month. This figure shows the spread between the lowest monthly CPU time (slowest month) and the highest monthly CPU time (busiest month) for each application. Typically, the busiest month uses 2-3 times as much CPU time as the slowest month.

Figure 6 Monthly CPU Spread

A review of the peaks is an example of a mechanism for monitoring resource utilization which may reveal patterns of processing which can be leveled out by a change in processes. The peak requirements are especially important from a capacity planning perspective. The planner must ensure that there is sufficient capacity available to satisfy the demand and still meet negotiated service agreements. This can have a significant impact on billing rates in cost recovery systems.

We selected the four systems with the greatest total CPU utilization for the year, Aerometric Information Retrieval System (AIRS), IFMS, Permit Compliance System (PCS) and RCRIS, for a more detailed review of usage patterns.

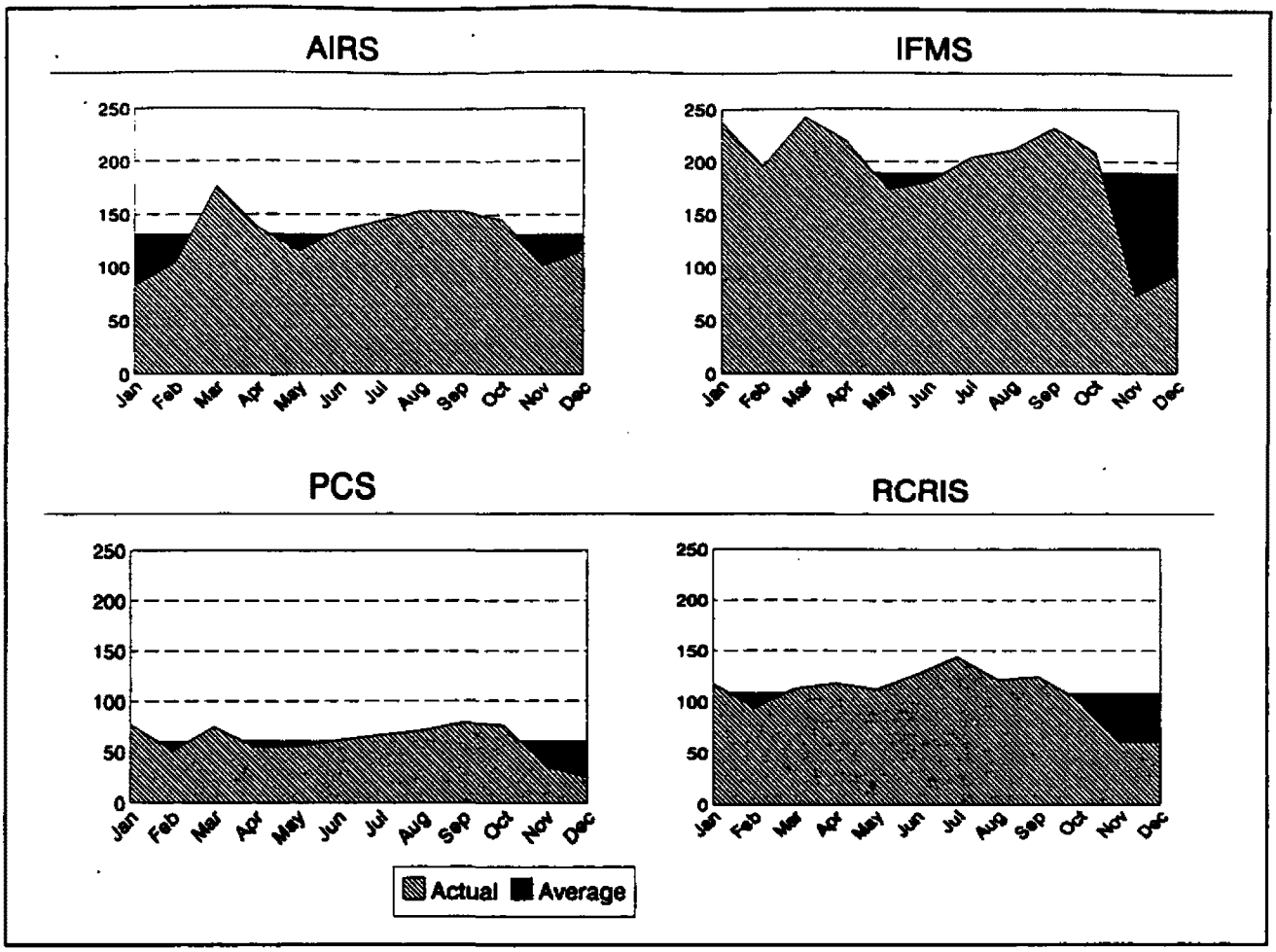

Figure 7 CPU Hours by Month

Figure 7 shows the graph of usage by month for the four systems, compared to the graph of their average monthly usage. All of these systems show distinct peaks and valleys in the usage. All of these systems show a sharp drop in activity during October, November, and December. This corresponds to the beginning of the fiscal year when budgets have not yet been finalized, and to calendar year-end with the holidays and use-or-lose leave time.

System Managers Do Not Periodically Review Systems To Prevent Obsolescence Of Software

System managers do not periodically review all software resources to identify and prevent obsolescence of software in accordance with the IRM Policy Manual, which states:

> *EPA program officials will periodically review all software resources to determine and prevent obsolescence of software. Indicators of obsolescence include more than five years since the last substantial redesign.*

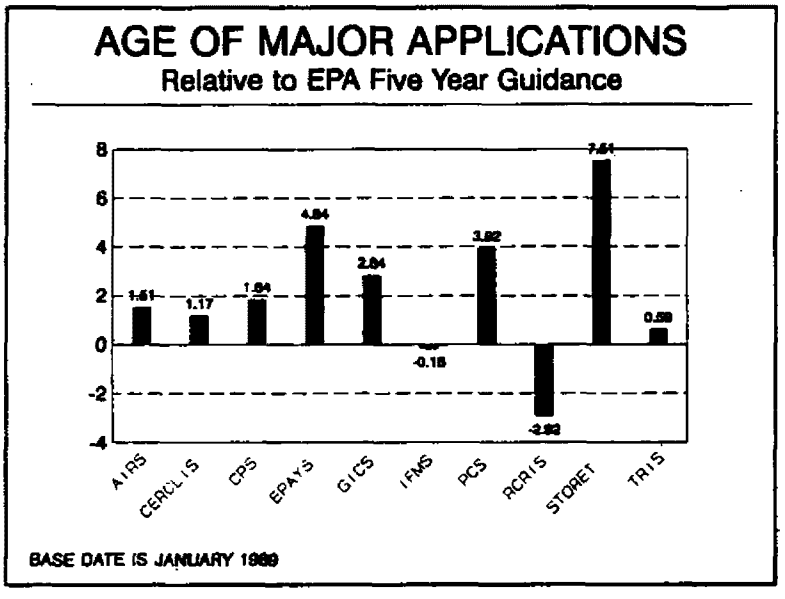

Figure 8 shows the age of the ten applications in this review relative to the 5-year age guideline. All of the application systems except IFMS and RCRIS are older than the 5-year guideline. Based on age alone, these systems meet Agency requirements for an ADP Review to determine obsolescence.

Figure 8 Age of Applications

In direct response to a 1991 OIG report recommendation, IFMS management contracted for the performance of a Functional Requirements Analysis, as well as a Cost/Benefit Analysis, to assess the adequacy of IFMS's present functionality. The results of these analyses were addressed in a Decision Paper which was signed by the Chief Financial Officer (CFO).

Interviews with system managers disclosed that while these systems have been subject to an ADP Review, the reviews did not examine all software resources for obsolescence. Furthermore, the reviews were not conducted in accordance with OIRM guidance. The systems also have not been subject to recent management control reviews under the Integrity Act, with a focus on obsolescence. Both ADP reviews and management reviews could be expanded to meet the requirement for a

review for obsolescence. It would be beneficial if the reviews also considered the ability to maintain the system in a cost-effective manner.

System managers do not consider age of the application to be a factor in the determination of obsolescence. For example, CPS managers indicate that a determination of obsolescence would be based on the subjective judgement of the division director based on his experience managing and developing administrative systems, as well as his perception of budgetary implications and technology considerations. RCRIS system managers report that system obsolescence would be driven by an extensive combination of system-level and programmatic requirements which could not be met, or a complete revision of the Resource Conservation and Recovery Act (RCRA) legislative mandates. PCS management has recently developed National Pollution Discharge Elimination System (NPDES) Information Management Strategic Plan involving Headquarters, Regional, and State program managers in a series of Total Quality Management (TQM) processes. The TQM processes concluded that PCS currently supports program functions and data needs. This plan defines a process for conducting an Integrated Strategic Plan (ISP) and Business Area Analysis (BAA) to determine system obsolescence.

MORE INFORMATION NEEDED TO EFFECTIVELY MANAGE SOFTWARE MAINTENANCE

System Managers Need Information For Setting Priorities, Planning, And Defect Removal

System managers do not have the management information they need to effectively set software maintenance priorities, conduct capacity planning, or manage removal of software defects.

> *When you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind.*
>
> *- Lord Kelvin*[8]

Software Maintenance Priorities. In times of budget cuts, managers are required to make painful decisions about deferring software maintenance. Corrective and adaptive changes frequently cannot be deferred. This leaves perfective changes as the source of discretionary changes which may need to be deferred for budget reasons. If managers cannot identify which changes are corrective, adaptive, or perfective, they cannot make effective decisions about software maintenance priorities. In fact, some system managers have

[8] Quoted by Lloyd K. Moseman

indicated that they receive a budget for contractor maintenance and a list of changes approved by the Change Control Board, and process changes from the approved list until the contract dollars have been exhausted.

System failures have an immediate impact on daily operations of the office. Each ABEND and JCL error represents the requirement to fix the error and then resubmit the job. Fixing the error may be an operational issue for JCL errors and data errors, but fixing ABENDS will normally involve *corrective maintenance*. Usually, fixes to system failures cannot be deferred, and in fact they are often made under emergency conditions.

Regulatory changes, hardware changes, and operating system changes have a direct impact on the types of changes to information systems. Detailed quantitative data is necessary to predict, review, assess, and negotiate with NDPD about the administrative overhead of proposed hardware and software changes. If program officials cannot quantify the impact of the change, they cannot effectively influence the schedule of the change, they can only adapt the software when the change is imposed. We have seen this with changes to the Toxic Release Inventory Form R where 85 percent of the modules were changed between May 19 and September 1, 1992. A minimum of 25 people are involved in adaptive maintenance annually.

Managers make decisions on the basis of the best available information, and performance measurement can improve the quantity and quality of this information. A complete management system includes a performance monitoring feedback loop with successive cycles of goal setting, performance monitoring, and regular reporting. Such a system requires regular, efficient information collection, analysis, and review. A performance measurement system in some instances simply formalizes, makes more efficient, and makes more explicit the decision-making process managers use intuitively. Such a system also forces managers to confront hard evidence about program efficiency and effectiveness.

Changes, insertions, deletions, modifications, extensions, and enhancements which are made to a system to meet changing user needs represent the evolution of the information system. They generally make the system more responsive to the way users do their work. But these changes are often discretionary in nature, and thus could be deferred during periods of tight budgets.

Capacity Planning. Managers also need resource utilization information to adequately conduct capacity planning. Capacity planning is becoming increasingly important to the program offices because the Agency is currently establishing a Working Capital Fund in which program offices will be billed for the IRM services provided. However, system managers base budget requests on

historical costs for onsite assistance (OSA), staff, hardware and software maintenance, and capital investment in new hardware and software.

NDPD's Architectural Planning and Management Branch (APMB) is responsible for ensuring sufficient mainframe computing resources for the Agency. APMB uses 24 EPA target application growth rates to determine mainframe requirements. APMB collects and summarizes CPU data from the MICS Capacity Planning data bases. The Capacity Planning staff uses modeling software to project response times based on growth ratios applied to a baseline model.

APMB categorizes the CPU usage and CPU usage growth rates by both Program Office and Program Office applications. Thus APMB can determine which Program Offices and applications are driving the consumption of the mainframe resources. The mainframe is a limited resource, and efficient utilization is necessary to keep costs down and accommodate growth. Ten steps for more cost effective computing have been identified, including offloading, upsizing and downsizing application processes.

We can expect to see the fee structure of full cost recovery via the Working Capital Fund encourage the desired behaviors represented in these ten steps. Also, we can expect to see the rate structure impose premiums on the less desired behaviors. Offices which can manage their scheduling to minimize the variance between high and low demand will benefit.

Defect Removal. Detailed record keeping of failures (i.e., defect counts -- JCL errors and ABENDS) is critical to quality assurance.

> *There are two major components of a quality measurement program: user satisfaction measures and defect removal measures. In a well-planned measurement program, defect counts are one of the key hard-data measures. Defect counts are continuously recorded during project life cycles starting as early as requirements reviews and continuing through maintenance.*[9]

Since many application systems do not report ABENDS and JCL errors as problems, any attempt to remove defects or improve software quality would start with incomplete information. For most of the systems reviewed, it would not be possible to identify which JCL fails consistently, which program ABENDS most often, which program is sensitive to bad data, which Region or State consistently enters bad

[9] Jones, Capers, Applied Software Measurement, McGraw-Hill, Inc., New York, 1991

data. Analysis of these issues could identify programs to be changed, JCL to be changed, edits for bad data, and training needs. Without hard data, the system manager cannot determine which defects not to fix to avoid destabilizing the software.

Senior Agency Managers Do Not Have The Information Needed To Replace Systems Based On Economic Value

Senior Agency managers do not have the management information they need to make a decision to maintain or replace a major information system, or to manage the risks associated with software maintenance. Our review of ten major information systems indicates that several senior managers within the Agency would not replace major systems until conditions are extreme. We believe, based on the information available to us, that each of these systems, or family of systems, as shown in the chart below, exhibit one or more of the characteristics FIPS Pub 106 defines as the factors to consider in weighing a decision to maintain or redesign including: (1) frequent system failures; (2) code over seven years old; (3) overly complex program structure and logic flow; (4) excessive resource requirements; and (5) difficulty in keeping maintainers.

Table I System Replacement Factors

System	Frequent Failures (ABENDS)	Resource Requirements (CPU Hours)	Age (Date Implemented)	Complex Logic (# Programs)	Obsolete Software
AIRS	7707	1569	Jul 1987	5,379	
CERCLIS	3811	662	Nov 1987	449	System 2000
CPS	846	92	Mar 1987	649	
EPAYS	1911	177	Mar 1984	413	PL/1
GICS	909	46	1972/1986	5,802	
IFMS	.11179	2275	Mar 1989	3,360	
PCS	7333	730	1975/1985	951	
RCRIS	3360	1294	Dec 1991	3,403	
STORET	1865	95	1965/1980	405	PL/1
TRIS	1648	250	Jun 1988	260	

For example, five of the systems we reviewed had more than 3,000 ABENDS in 1992. This is an average of more than 12 ABENDS each

working day of the year. Five of the systems used more than 500 CPU hours in 1992, or an average of 2 or more CPU hours each working day of the year. At the 1992 billing rate of $700 per CPU hour, this amounts to more than $350,000 per year. Three of the systems originated twenty or more years ago. Another system was transferred from another agency in 1984. Four of the systems are very complex, consisting of more than 3,000 programs. Just keeping track of all the programs becomes a major logistical exercise. And finally, three of the systems are using obsolete software platforms. TRIS is the only system which does not meet these conditions. However, TRIS suffers from excessive adaptive maintenance. During 1992 virtually all of the 1300 objects were modified to accommodate the requirements of the Pollution Prevention Act.

The Comprehensive Environmental Response, Compensation, and Liability Information System (CERCLIS), which uses the System 2000 Data Base Management System, meets the criteria of difficulty in finding maintainers. We conducted an informal survey of five technical recruiting and technical consulting firms to determine the availability of programmers with System 2000 skills as well as industry demand for such skills. None of the five had a request for such skills in the last five years. These companies characterize System 2000 as archaic and obsolete. One firm has a data base of 11,500 consultants. Only ten consultants in that data base admit to System 2000 experience, and they tend to have 25-30 years experience and command premium rates. Another company indicated that these consultants will not take a System 2000 assignment if any other assignment is available, and even then require incentive rates. Additionally, because CERCLIS is the primary user of System 2000, OSWER pays the full cost of maintaining the System 2000 software.

We developed a risk model based on a formula taught by the GSA training center. The model uses three components to "score" application systems: (1) sensitivity impact; (2) risk criteria weight; and (3) element risk score. Using this model we produced normalized risk scores for the ten major systems. The weighting factors of the risk model tended to level the scores (i.e., factors with high risk would be leveled by factors with low risk). For example, factors such as age which would increase the risk may be cancelled by the low number of programs in the system. Factors such as the

Table II Risk Scoring

Risk Score	Risk Ranking
0 - 14	Very Low
15 - 44	Low
45 - 60	Low Moderate
61 - 74	High Moderate
75 - 89	High
90 - 100	Very High

number of changes would be cancelled by the program-exclusive impact of system failure. Table II shows the risk assigned to each score in our risk model. The low and moderate risks are defined to accommodate a wide range of scores.

Figure 9 shows the risk scores for the ten systems. All are grouped within a narrow range between 60 and 80. Eight of the ten systems in our study have high-moderate risk based on software maintenance factors. The two major financial systems have high risk primarily because of the Agency-wide impact of system failures.

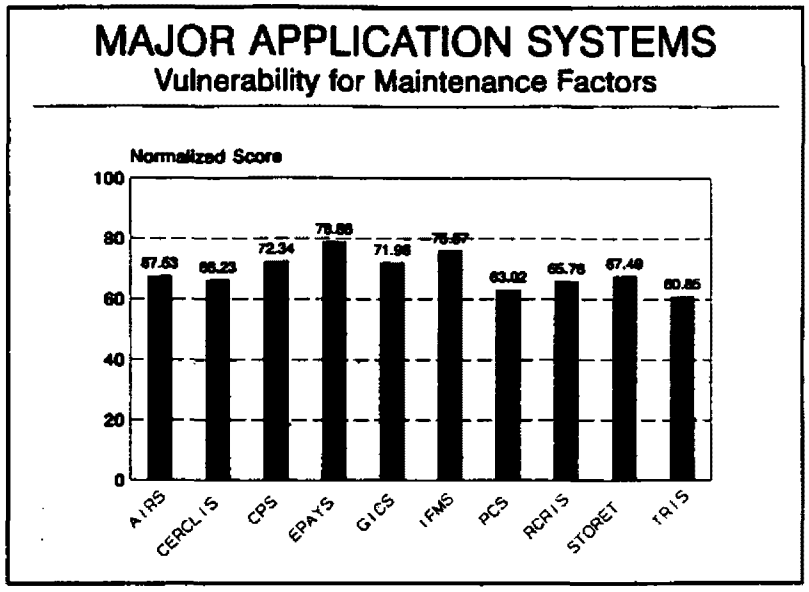

Figure 9 Vulnerability Assessment

Additionally, due to the lack of management information on software maintenance, managers are delaying cost/beneficial decisions to redesign or replace systems well beyond when the systems begin to become inefficient and fail to support program needs. For example:

Storage and Retrieval of Water Quality Information (STORET). OW is modernizing the STORET system and indicated several reasons for the modernization initiative:

1. STORET was last upgraded in 1980. The system was falling apart, and only two or three people knew how to maintain the system;

2. STORET lacked the flexibility to meet the present need of it users. The present system lacks the capability to enter Quality Assurance/Quality Control (QA/QC) data about the data values, thereby limiting the usefulness of the data to secondary users; and

3. To gain the necessary flexibility, system managers wanted to move to a relational-based system.

Additionally, combined operations and software maintenance costs were high. In February 1993, the system manager reported that annual operating costs were approximately $2.5 million and software maintenance was an additional $275,000. However, when we calculated 1993 operations and maintenance costs based on OMB reporting, and allocations for timeshare and telecommunications, we found these costs totaled $1.3 million. Total funding for 1990, prior to the modernization initiative and the decision to suspend perfective maintenance, was approximately $5 million (including OW and OIRM costs for CPU time, contractor costs, FTE salaries, travel costs, etc.).

RCRIS. OSWER obtained funding in 1986 to begin the design for an information system to replace the Hazardous Waste Data Management System (HWDMS). The two major reasons for this effort were:

1. HWDMS was only used by the EPA Regional offices. The States maintained their own data. As a result, EPA and the States were providing two separate and sometimes conflicting sets of data to Congress; and

2. Constant changes to the software by EPA Headquarters irritated the Regional offices.

Development of the RCRIS system represented a change in the way the States and EPA did business together. They agreed that the program entity responsible for the program was also responsible for the data. This addressed the issue of separate and conflicting data.

OSWER officials describe the process for replacing HWDMS as a model of the process that would be used to replace RCRIS:

> A determination of system obsolescence derived from a combination of system level and/or programmatic requirements or constraints not being met by the HWDMS system. The determination involved extensive review of national software programs, information management processes, conversion requirements, and program implementor skills. The final determination involved office director evaluation as well as OSWER Assistant Administrator review through the OSWER IRM Steering Committee, with heavy involvement by Regions and States.[10]

[10] RCRIS system management, 11/24/93

Federal Reporting Data System (FRDS). OW was forced to modernize FRDS by creating FRDS II when OIRM discontinued support for the System 2000 Data Base Management System. While FRDS was not part of our review, it provides an excellent example of a system that was not replaced until no other choice was available. Once the decision was made to replace FRDS, OW chose to design the new system to meet current needs.

When OW was forced to replace the system they were having significant maintenance difficulties; one official stated "When we fixed a problem, we were not really sure that it was fixed. A fix to one problem had a ripple effect which identified other problems." OW decided that it was a matter of bandaids versus a new system, so they did a comparative cost analysis for continued maintenance versus a new system.

Data quality has been inconsistent, with the biggest problems being in the small drinking water systems. As a result, OW has implemented a policy that they will not look at compliance issues if the data has not been recorded in FRDS. This has provided an incentive for the States to accurately record the data. The Regions and States rely on Headquarters to keep the system available, and Headquarters relies on the States for data quality.

The existing FRDS did not help in regulatory development. Current data in FRDS was not broad enough to monitor trends, or to evaluate performance to adjust regulatory strategy. OW could not evaluate values below the standard because they are not reported.

OW initially identified business centers to be supported, and implemented those centers which were directly supported by the existing FRDS system. They are using work groups, user groups, demonstration meetings, and pilot projects to sell FRDS II to their user community. They see a significant benefit from having all Regions and States using the same system. FRDS II has been designed to be flexible enough so that individual States can customize parts of the system to meet their needs while maintaining the integrity of the overall system. OW has currently spent about $1.5 million of the estimated $7 million replacement effort.

INADEQUATE MANAGEMENT ATTENTION TO AND INSUFFICIENT CONTROLS OVER SOFTWARE MAINTENANCE

Management Did Not Recognize The Significance Of System Operations And Maintenance Cost Information

Currently, EPA has not implemented a cost accumulation process for major information systems. The account code structure in the Agency's financial system does not provide for cost tracking by

information systems and no cost accounting process has been established to properly allocate cost by system.

Cost awareness enables managers to control and save costs, as well as find and avoid waste. Reliable cost information helps managers ensure that resources entrusted to them are spent wisely and alerts them to waste and inefficiency. Therefore, if system managers cannot view relevant system cost data, they cannot recognize the magnitude of actual dollars related to these activities within their particular application system(s). This lack of information could significantly influence their ability to perceive growing software maintenance costs as a major expenditure of their budgeted resources.

EPA's Policies And Procedures[11] Do Not Establish Adequate Guidelines Or Provide Adequate Direction For The Software Maintenance Process

EPA's Operation and Maintenance Guidance (O&M) defines a configuration management process which includes primarily quality assurance activities. However, it does not implement FIPS Pub 106 guidelines to examine how EPA's software is maintained, exercise control over the process, and ensure that effective software maintenance techniques and tools are employed. Neither does it establish guidelines for managing the software life cycle, process, and products in compliance with EPA Directive 2100. All of the areas defined in the O&M guidance implement FIPS Pub 106 guidelines for assuring adherence to system standards in all phases of the maintenance effort.

EPA's Operations and Maintenance Manual defines software maintenance requirements in the following terms:

- Documentation update;
- Source code standards;
- Coding and review process; and
- Testing standards and procedures.

However, existing OIRM Operation and Maintenance Guidance does not address the following key management issues which would establish controls over software maintenance:

- version control techniques to manage the maintenance process;
- definition of the quality assurance functions to manage the maintenance process;

---

[11] EPA Directive 2100, Information Resources Management Policy Manual
EPA System Design and Development Guidance (June 1989)
EPA Operations and Maintenance Manual (April 1990)

- definition of the appropriate project status reporting and quality assurance tasks for software maintenance activities; and
- assurance that the operational system uses an optimum, least-cost mix of resources to meet user functional, data, and other systems compatibility requirements.

The IRM Policy Manual, Directive 2100, dated July 21, 1987, requires that the operational system continue to conform with applicable copyright laws and Federal standards and guidelines, but does not specifically address software maintenance or significant maintenance-related issues, such as:

- provisions for incorporating new hardware and software technologies into the software maintenance processes; and
- restricting application programmers from production data.

Program Managers Have Not Issued Software Maintenance Polices Or Goals

Many Agency program managers have neither issued policies and procedures for software maintenance nor established goals or objectives for software maintenance. Both OMB Circular A-130 and EPA Directive 2100 assign responsibility for managing information technology to the program manager whose program is supported by information technology.

We sent our management survey questionnaire to nine Assistant Administrators, ten Regional Administrators, and twelve Laboratory Directors. Twenty-one of these officials responded that they do not have a software maintenance policy. The other ten program offices developed their own standards, policies and procedures related to software development and maintenance. Only five of these officials reported that they have established goals and objectives for the performance of software maintenance. Only three officials measure the performance of software maintenance based on goals and objectives.

Agency Managers Do Not Adequately Track Software Failures

Many Agency system managers do not have mechanisms in place which adequately track, record, and classify software defects (or failures). Commercial defect tracking software is available which provides robust capabilities to record and classify the severity of defects, and provide up-to-date information about the status of the defect. Commercial defect tracking software can also provide standard reports of defects by project, program, and developer. Problem records are analyzed to produce a variety of defect measurements:

- Defect types: Counts by category of various kinds and classes of defects;
- Defect distributions: Location and distribution of defects throughout the software;
- Defect rate: Plots over time showing defects per unit of time or effort;
- Defect age: Time from when a defect is first entered into the system until it is detected;
- Defect response times: Time from detection to fix;
- Defect cost: Cost of failure (system impact of the failure) as well as the cost of analyzing and fixing the defect; and
- Defect density: Defects detected per unit of work, e.g., page of design specification, or per thousand lines of code (KLOC), or per page of documentation.

Agency Managers Do Not Use Maintainability Or Economic Criteria For Determining When To Replace Major Information Systems

Agency managers generally consider the retirement or replacement of an information system to be a major undertaking which should be avoided whenever possible. This is articulated well by OSWER management. Generally, OSWER management agreed that major system changes, such as a determination of obsolescence, are predicated on changes in mission, audience, budget, and technology.

However, Agency managers do not consider maintainability and economic criteria, which should be major factors, for retirement or replacement of information systems. Capers Jones[12] states that as computers evolved from being specialized military and academic curiosities, economic necessity has made measurement imperative. The need for accurate measurement of software productivity and quality is directly related to the overall economic importance of software to industry, business, and government.

EPA Directive 2100, Information Resources Management Policy Manual, declares that it is EPA policy to enhance the management of software throughout its life cycle. It provides 13 specific statements of policy for managing software including a requirement to periodically review all software resources to identify and prevent obsolescence of software. The policy manual requires that ADP reviews conducted by the program offices must be coordinated with the Office of the Inspector General.

[12] Op. Cit. 5

EPA Directive 2115, Guide for ADP Reviews, provides a methodological and procedural framework for the planning and execution of ADP Reviews. An ADP review is defined as an evaluation of an information system, ADP equipment, operations, or an organization, to determine if the intended and expected functions are being accomplished. Procedures are defined for mission support reviews, management reviews, and technical reviews of application systems. Appendix C states that ADP systems should be reviewed and audited on an annual basis to determine:

(1) if the system is still needed to satisfy valid EPA requirements;
(2) if the system is performing adequately or needs to be modified;
(3) if the costs involved in operating the system are justified by the benefits received;
(4) if the costs/benefits associated with the system justify its continued existence given the current fund limitations and overall priorities within the user's organization; and
(5) if adequate user documentation and system maintenance documentation exists to use and maintain the system.

Clearly, the EPA directives suggest that there are reasons other than 'Change in Mission' for making major system changes, including the retirement of the system. While EPA Directive 2115 was issued in 1984, the guidelines for conducting ADP reviews are still valid. However, the guidelines do not address measurement issues such as defect tracking or customer satisfaction surveys.

FIPS Pub 106 Does Not Define Software Maintenance Management Processes

FIPS Pub 106 recognizes the importance of management in the software maintenance process, stating that management is clearly one of the most important factors in improving the software maintenance process. It states that management must examine how the software is maintained, exercise control over the process, and ensure that effective software maintenance techniques and tools are employed. In addition, software maintenance managers must make decisions regarding the performance of software maintenance, assigning priorities to the requested work, estimating the level of effort for a task, tracking the progress of the work, and assuring adherence to system standards in all phases of the maintenance effort.

However, FIPS Pub 106 -- which is the only Federal guidance on software maintenance -- assumes that all managers will know how to fulfill these responsibilities. It does not provide guidance about the effective techniques and tools which managers must employ. It does not define techniques and tools which aid in exercising control over the software maintenance process or in estimating the level of effort for a task. Agencies are left to exercise these technical responsibilities without adequate guidance. This is an issue which

we intend to address in our governmentwide report to the Federal oversight agencies. Nevertheless, it is still incumbent on individual Federal agencies to establish their own guidance in the absence of Federal guidance.

RECOMMENDATIONS

We recommend that the Assistant Administrator for Administration and Resources Management, in his role as the Designated Senior Official for IRM and, when appropriate, in conjunction with the Executive Steering Committee for IRM:

2-1. Identify the measurements needed to support Agency-wide management of software maintenance. The measurements should include:

- resource tracking -- quantification in dollar amounts of intramural and extramural resources used as the input for production of a service or product, i.e., estimating and tracking resource use, tasks, deliverables, and milestones;
- work product tracking -- the number of units of the product or service provided to the customer; the level of service or product quality, both in terms of customer satisfaction (external quality) and of work performed to provide the service (internal process quality), e.g., tracking and control of source code, test case, and document versions and changes; and measures of size and complexity, e.g., Halstead code measurements, function points, cyclomatic complexity, Kiviat diagrams; and
- problem tracking -- tracking and control of problems, defects, and open issues.

2-2. Based on the metrics defined in our first recommendation, require that OIRM modify its Operations and Maintenance Guidance to establish processes to:

- define appropriate project status reporting and quality assurance tasks for software maintenance activities;
- manage the software life cycle, maintenance process, and products within Agency programs in compliance with EPA Directive 2100; and
- implement FIPS Pub 106 guidelines to examine how the software is maintained, exercise control over the process, and ensure that effective software maintenance techniques and tools are employed.

2-3. Based on the metrics defined in our first recommendation, require OIRM to update EPA Directive 2115 to make the ADP Review a comprehensive review of the system and its support for Agency goals and missions. Include review requirements that would:

- require quantitative measures of performance, and a user satisfaction survey of the system;
- require that the program office demonstrate the extent to which the system supports Agency and program office strategic objectives;
- require a periodic review of the effectiveness, accuracy, need, and economic justification for continued operation for each information system; and
- ensure that operational systems use an optimum, least-cost mix of resources to meet user functional, data, and other systems' compatibility requirements.

2-4. Evaluate commercial defect tracking software, and determine whether any available package should be included as an Agency standard for problem tracking and defect removal in Agency roadmap planning and hardware/software standards documents.

AGENCY COMMENTS AND OIG EVALUATION

In their March 17, 1995, response to our draft report, OARM officials agreed with all four of our *revised* recommendations. OARM has initiated action on one recommendation while NDPD is performing action on another recommendation. Completion of the two remaining recommendations are dependent on the implementation schedule for recommendation 2-1.

However, OARM's proposed actions do not fully meet the intent of two of our recommendations. Although OARM agreed with recommendation 2-2, the corrective actions do not specifically state that the metrics, defined in recommendation 2-1, will be incorporated in the revised version of the Operations and Maintenance Manual. Rather, OARM's response focuses on revising the document to strengthen the concepts presented in FIPS 106. Similarly, it is not immediately clear from the response to recommendation 2-5, whether the revised ADP Review Program will incorporate the defined software measurement metrics as a required part of the review process. Instead, OARM's response emphasizes that the review program will be revised to meet the requirements of the Paperwork Reduction Act (PRA) in a more comprehensive manner, and that the revised program's infrastructure will consist of developing evaluative tools to assist in the review of IRM activities. The intent of these two recommendations was to integrate the defined measurements into software maintenance management processes, under the instruction of Directive 2100 and the O&M Manual.

CHAPTER 3

SOFTWARE MAINTENANCE COSTS NOT AVAILABLE FOR DECISION-MAKING

COST ACCUMULATION AND TRACKING REQUIREMENTS AND GUIDANCE

We used the following list of Federal and EPA requirements and guidance in conducting our audit in the area of software maintenance cost accumulation and tracking. For more detailed information about the criteria see Appendix IV.

- OMB Circular Nos. A-11, A-109, A-130;
- 1994 GAO Executive Guide, entitled "Improving Mission Performance Through Strategic Information Management and Technology: Learning from Leading Organizations";
- Chapter 3 of the EPA System Design & Development Guidance, Volume B (June 1989);
- EPA Operations and Maintenance Manual (April 1990);
- 1991 memorandum from the Chief, Financial Reports and Analysis Branch, to Financial Management Officers, entitled "Reconciliation and Verification of Capitalized Equipment with Property Management Officers and Accountable Officers"; and
- Title 2: GAO Policy and Procedures Manual for Guidance of Federal Agencies (May 1988).

COST INFORMATION FOR SOFTWARE MAINTENANCE NOT AVAILABLE TO MANAGEMENT

Lifecycle Cost Information Was Incomplete For Management Decisions And Reporting

In fiscal 1993, EPA spent an estimated $98.8 million on application systems O&M. Further, while accurate figures are not available, we estimate that over the life cycle of its systems (i.e., an estimated 12-year life) that the cumulative costs for O&M[13] will exceed $1 billion. We recognize that EPA has initiated efforts to create a

[13] Operations and Maintenance amounts include contracts to provide services associated with operations of existing systems. This includes systems hardware and software maintenance, capacity and facility management, data entry support, maintenance/operation of tape/disk libraries, etc. It also includes maintenance furnished as a part of software purchases and license arrangements or for rental/lease contracts when significant and readily identifiable in the contract or billing.

Working Capital Fund, so that it can more cost effectively administer services, including ADP and telecommunications services. However, contrary to the requirements of OMB Circular A-109 as well as good business practices, systems managers were not developing, reviewing, and updating the life cycle costs of their individual systems.

We reviewed the cost management of ten major application systems (see Table III) in five program offices which accounted for a total of $60 million of fiscal 1993 O&M costs. We found that system managers were only responsible to manage program funds, which were about 51 percent of total system costs. Timeshare and telecommunications costs were about 49 percent of overall system costs, but these costs were managed in aggregate by the Director, OARM, Research Triangle Park. The implementation of the Working Capital Fund should provide for better system cost accounting.

Most system managers were not accurately collecting the lifecycle costs of the ten systems reviewed. When asked for cost information, system manager responses varied greatly. One system manager indicated that she was providing all costs from conception through implementation. However, most system managers were only able to provide information on contract dollars spent by their program office in support of the system. The costs that the system manager should accumulate include direct, indirect, recurring, nonrecurring, and other related costs in the design, development, production, operation, maintenance and support of a major system.

In addition, system managers were not accurately accumulating usage-based costs for specific systems. When asked to provide their O&M costs, only four system managers provided information on usage-based costs. However, the information provided was widely inaccurate and incomplete. The other six system managers provided only contract dollars, not usage-based cost information. System budgets, in general, were not affected by ADP utilization, and utilization costs were not considered when making decisions regarding these systems.

ANNUAL COSTS ASSOCIATED WITH MAJOR SYSTEMS - FISCAL 1993[14]						
PROGRAM OFFICE	SYSTEM NAME[15]	FISCAL 93 PROGRAM COSTS	FISCAL 93 NDPD TIMESHARE COSTS	FISCAL 93 NDPD TELECOMM COSTS	TOTAL FISCAL 93 COSTS	PROGRAM PERCENT OF COST
OAR	AIRS	$4,660,000	$3,184,642	$3,206,934	$11,051,576	42.17%
OARM	CPS	$736,800	$384,773	$387,466	$1,509,039	48.83%
	EPAYS	$1,120,600	$793,595	$799,150	$2,713,345	41.30%
	GICS	$611,100	$198,458	$199,847	$1,009,405	60.54%
	IFMS	$7,427,400	$4,246,700	$4,276,400	$15,950,500	46.57%
OPPTS	TRIS	$8,328,000	$580,154	$584,215	$9,492,369	87.73%
OSWER	CERCLIS	$1,090,000	$1,410,461	$1,420,334	$3,920,795	27.80%
	RCRIS	$3,179,000	$2,135,958	$2,150,910	$7,465,868	42.58%
OECA	PCS	$2,671,700	$1,050,157	$1,057,508	$4,779,365	55.90%
OW	STORET	$537,100	$395,939	$398,711	$1,331,750	40.33%
TOTALS		$30,361,700	$14,380,837	$14,481,475	$59,224,012	51.27%

Table III - FY 93 Annual Costs

Absence Of Cost/Benefit Analysis

Federal criteria states that requests for system modifications should be carefully reviewed and evaluated before any actual work is performed on the system. The evaluation should consider, among other things, the costs and benefits of the change. In particular, perfective maintenance changes must be thoroughly analyzed, since

[14] Methodology and scope are addressed in Appendix II.

[15] Seven of these systems reported in EPA's OMB Circular A-11 43A, dated October 5, 1993 (AIRS, IFMS, TRIS, CERCLIS, RCRIS, PCS, and STORET).

they are optional in the sense that failure to implement them will not adversely affect system performance. Changes should be approved only if the benefits outweigh the costs. Because corrective and adaptive maintenance are not optional, cost-benefit analysis is most appropriately used to determine the best option for applying the required changes.

Change control processes for six of the ten application systems reviewed did not include cost-benefit analysis. Management too often relied on simple contractor estimates to determine the level of effort and costs involved with a proposed modification. For many systems, these estimates, if performed at all, were solicited *after* the requested modification had been approved by the change control committee. These estimates did not weigh the benefits to be derived by implementation of the proposed software change as opposed to the costs involved in the modification process. Without proper analysis, scarce budgetary resources could be mis-allocated and non-cost effective software changes could be implemented.

Only the IFMS, AIRS, PCS, and RCRIS application systems had formal processing requirements which stipulated preparation of cost-benefit analyses. In the case of IFMS, the Finacial Management Division/Financial Systems Branch (FMD/FSB) IFMS Procedures Manual outlined definite procedures for the preparation of a cost-benefit analysis. In addition, PCS performed a cost-benefit analysis for programmatic enhancements, within the confines of the 'feasibility study'. However, no such analysis was performed for non-critical corrective maintenance, despite the magnitude of those software changes.

Cost-benefit analyses were not prepared for the remaining six application systems. For example, the contractor for TRIS submitted cost estimates, but did not perform a cost-benefit analysis. During audit interviews, several system managers responded that decisions to implement a change were based on whether sufficient funds were available to do the job and whether system functionality was affected. Software maintenance was performed until available funds ran out; the benefits to be realized were a secondary consideration.

IMPACT OF NOT MONITORING AND TRACKING SOFTWARE MAINTENANCE COSTS

System Life Cycle Decision-Making Impaired

Program managers were not in a position to make informed and effective short- and long-term decisions, such as choosing the best enhancement, upgrade, or improvement based on accurate cost information or cost-benefit analysis. Proper cost-benefit analysis serves a valuable purpose, especially when allocating scarce budgetary resources. These analyses ensure that the most cost-

effective alternative which satisfies system requirements is chosen. Without it, scarce budgetary resources may be mis-allocated. Non-cost-effective changes to the software could be approved by management and implemented into the production environment, despite the marginal benefits which would be realized from the modification. This could result in a lack of funds for those proposed software changes which are either mandatory in nature or necessary for proper application functioning.

Additionally, because system managers were unable to separate maintenance costs from operations costs, they could not effectively evaluate the economy and efficiency of operations. In the current environment of shrinking budgets, it is crucial that senior program managers use cost as a performance factor and make decisions regarding available resources in the most effective manner possible thereby supporting increased accountability as required by the Government Performance and Results Act of 1993.

Furthermore, NDPD managers are relying on inaccurate system usage information for budget decisions. Capacity planning is based on mainframe usage predictions for major information systems derived from the computer utilization statistics accumulated in the Facility Impact Monitoring and Analysis System (FIMAS), through the use of a FIMAS code[16]. For example, the "HWDM" FIMAS code was assigned to HWDMS, which was archived in January 1992. Yet, as of September 1993, NDPD's Mainframe Capacity Report used the HWDM FIMAS code, listed this system as the 20th largest system in usage and predicted a 9.2 percent increase in usage for this system.

Incomplete Reporting To OMB

The lack of a cost accumulation process to collect the lifecycle costs of these systems has led to incomplete reporting of these costs to OMB. During the period of this review, Circular A-11 required reporting based on a $25 million lifecycle cost threshold or $10 million annual cost threshold. Seven of the systems reviewed were reported to OMB as major information systems. Table IV below summarizes the lifecycle costs of these seven systems. However, over $26 million of annual timeshare and telecommunications costs (48 percent of the total costs) were not reported on an individual system basis to OMB on the Circular A-11 Report on Information Technology Systems -- Exhibits 43A and 43B). These costs were only reported in the aggregate for the Agency rather than identified as system costs.

[16] The FIMAS code (also known as the ADP Utilization Identifier) is a four-character code which identifies a specific ADP system or activity and associates computer utilization statistics with that activity. These codes are obtained by the ADP Coordinator or Project (Account) Manager from the FIMAS Office at NDPD. Each user must obtain the correct FIMAS code from his ADP Coordinator or Project (Account) Manager.

TOTAL COSTS AS REQUIRED BY A-11 - FISCAL 1993					
PROGRAM OFFICE	SYSTEM NAME	INCLUDED IN A-11	OMITTED FROM A-11	TOTAL FISCAL 93 COSTS	PERCENT REPORTED TO OMB
OAR	AIRS	$4,660,000	$6,391,576	$11,051,576	42.17%
OARM	IFMS	$7,427,400	$8,523,100	$15,950,500	46.57%
OPPTS	TRIS	$8,328,000	$1,164,369	$9,492,369	87.73%
OSWER	CERCLIS	$1,090,000	$2,830,795	$3,920,795	27.80%
	RCRIS	$3,179,000	$4,286,868	$7,465,868	42.58%
OECA	PCS	$2,671,700	$2,107,665	$4,779,365	55.90%
OW	STORET	$537,100	$794,650	$1,331,750	40.33%
TOTAL		$27,893,200	$26,099,023	$53,992,223	51.66%

Table IV Total Costs Required By A-11

Because of this incomplete cost tracking, other major information systems which meet OMB reporting thresholds are not being reported. For example, EPAYS should have been reported to OMB under the requirements of OMB Circular A-11. This system was transported from another department and baselined in 1984. Using actual costs for fiscal 1987 through 1993, we were able to project costs for fiscal 1984-1986 and fiscal 1994-1995. Operations and maintenance on this system from fiscal 1984 through fiscal 1993 total $21.4 million. The projected system costs show that EPAYS obligations should exceed $25 million in fiscal 1995. Therefore, this system meets the threshold for reporting to OMB.

While OMB Circular A-11 Section 43 no longer requires system level reporting for all major information systems, Section 40 maintains this requirement for financial and mixed financial systems. It is therefore important to ensure that the process for reporting to OMB under Section 40 includes all obligations for these systems, including timeshare and telecommunications costs.

Inaccurate Financial Reporting

The Agency's current policy (see Appendix IV) requires the capitalization of ADP software valued at $5,000 or more, with a useful life of two years or greater. This requirement is consistent with GAO's "Title 2, GAO Policy and Procedures Manual for guidance of Federal Agencies." However, EPA did not properly capitalize at least $38.1 million of software costs for the ten systems we reviewed,

which distorts the accuracy of EPA's current financial statements. For example, all costs associated with purchase, development, and enhancement of the ten systems reviewed were treated as annual expenses. Table V, below, lists the individual system development cost estimates, as provided by the system managers.

ESTIMATED INITIAL DEVELOPMENT COSTS		
PROGRAM OFFICE	SYSTEM NAME	DEVELOPMENT COST
OAR	AIRS	$8,000,000
OARM	CPS	$550,000
	EPAYS	Not Available
	GICS	$2,000,000
	IFMS	$3,204,000
OPPTS	TRIS	$285,000
OSWER	CERCLIS	$3,900,000
	RCRIS	$18,313,000
OECA	PCS	$885,000
OW	STORET	$1,000,000
TOTAL		$38,137,000

Table V Initial Development Costs

BETTER COST ACCUMULATION FOR SOFTWARE MAINTENANCE IS NEEDED

EPA Has Not Implemented A Cost Accumulation Process For Major Information Systems

The 10-digit account code structure did not allow for project cost accumulation. However, the Agency recently replaced this account code structure with an expanded account code. In addition, the Chief Financial Officer's 5-year plan initiated a project to develop and

implement a project cost accounting system by October 1995. This would provide an acceptable method to accumulate costs as required by OMB Circulars A-109, A-11, and A-130.[17]

Most of these ten major information systems are funded through several program elements and no one person is responsible in total for all their costs. OIRM has developed a consolidation process to respond to the requirements of the A-11 43A submission. This process relies on estimation of system costs, because expense information is not accumulated in the accounting system. Individual system cost reports from all the program elements are collected and consolidated into one report. Other significant costs are not specifically identified to major information systems, such as timeshare, telecommunications, and data entry. Timeshare and telecommunications charges are paid for through the NDPD budget while data entry costs are incurred by the Regions and States. Thus, these costs are generally not specifically identified to these major systems.

The dispersion of funding also contributes to the general lack of management of information system costs throughout the Agency. In interviews, several system managers stated the reason they do not perform cost-benefit analyses is because they are not used in determining their budget. Other system managers added they do not complete the analysis because they compete with other functions within their program for resources and a cost-benefit analysis might hurt their chances of receiving adequate funding. None of these systems is tracked as a separate line item in the budget of the program it supports. On the other hand, one program official indicated he believes that such tracking would increase the visibility of major information systems and therefore improve their management.

System managers are only generally responsible for production accounts and accounts used by Headquarters personnel. Even though timeshare and telecommunications charges for the systems reviewed total more than $28 million, system owners and managers are not aware of user costs of operations by application and therefore do not concentrate on the cost and demand areas warranting attention.

Although OMB Circular A-130 requires agencies to implement a system to distribute and recover the obligations incurred for providing services to all users, no effort was made to charge program offices for the use of these resources. These costs are available for use by system managers if they choose to look for them. Without controls to

[17] There is some inconsistency between the accumulation of costs by categories of the system development lifecycle in OMB Circular A-109 and the functional categories provided by the current object class codes in OMB Circular A-11. OMB Circular A-109 requires specific accumulation by lifecycle stage, i.e. operations separate from maintenance. However, Circular A-11 groups operations and maintenance into one budget object class code.

ensure the accuracy of this information, it is not useful to system managers. EPA's project to establish the WCF would address A-130 requirements for billing. However, no accounting process that would be an acceptable basis for billing usage-based costs to users of the systems is currently in place. We support the concept of a working capital fund which would make users responsible for funding these system costs, because it would make them accountable for all material system costs.

EPA Officials Are Not Capitalizing Software

An Financial Management Division (FMD) official stated they had received no requests for software cost information, other than those made by the OIG. A Financial Reports and Analysis Branch (FRAB) official stated that no processes have been established to track or capitalize software costs, including those associated with major enhancements.

Although IFMS does not have a project cost accounting system or account code structure to accumulate these costs, FMD has an ongoing effort to identify and capitalize equipment and other assets in excess of $5,000. However, FMD has made no attempt to identify software development and major enhancements which frequently involve much higher figures.

A system manager stated that it is too time consuming and costly to require separate tracking of contract costs for operations, development, enhancements, and maintenance. Contracts generally accumulate all support costs by information system. However, without contractors breaking out maintenance, development, enhancements and operations costs, it is impossible to determine the actual cost of these activities for these systems.

Until IFMS is modified to provide cost and project accounting capability, FMD needs to establish an interim process to identify system costs to be capitalized. These costs should then be incorporated into the Agency's financial statements as appropriate.

FIMAS/Account Structure

NDPD officials stated that the FIMAS list was very unreliable since it showed any account that may have used the FIMAS ID for one of the systems involved. In addition, they stated the process used may not identify all accounts using a particular application but it was the best they could do.

It is extremely difficult for system managers or NDPD to accurately accumulate costs associated with CPU usage by system, primarily due to the number and variety of accounts assigned to each information system. In order to locate and summarize all the accounts for each

system, it is necessary for them to cross-reference data from three different sources: (1) the Superfund layoff reports, (2) a report showing the accounts that accessed particular FIMAS IDs, and (3) a listing of accounts decentralized in the Resource Access Control Facility (RACF).

When users[18] log onto the mainframe, they are asked for user-ID, account, FIMAS code, and password. The user-ID is specific to an individual. The account number is tied to a particular organization for billing purposes. The FIMAS code can reflect the applications being used, but it is not linked to specific users, accounts, or organizations. In addition, no controls or edits are imposed on the FIMAS code. Since the log on screen defaults to the last FIMAS code used by a particular user, the code can be ignored and/or used indiscriminately. This makes it difficult or impossible to accumulate these costs by information system.

In addition, a 1984 Management Review of EPA's National Computer Center reported that "It is now possible for a user ID to be logged in on one account and then do work on an application that should be billed to another account. Users therefore may spend timeshare funds for applications other than for which they were intended." Although this has been a known problem since 1984, it still exists today and prevents system managers from having the ability to monitor and control costs incurred by their systems. This longstanding problem extends to every major information system and involves at least the five program offices reviewed.

Furthermore, system managers are not maintaining appropriate internal controls over utilization of their systems. Users are granted access through numerous accounts in Headquarters, Regions, and States. These accounts are coordinated through ADP Coordinators in each program office. This creates a complex and unacceptable process of assigning and controlling system usage. For example, one system uses a total of 157 accounts with over 1,200 registered users. Also, the proposed WCF will need an acceptable process to accumulate and allocate system utilization to help calculate billings to program offices.

Clearer Criteria Needed

While several OMB circulars require management to track and report life cycle costs, clarification of policies, terminology, definitions, and responsibilities is needed on life cycle costing. For example, OMB Circular A-11 does not require the separation of operations, software maintenance, and hardware maintenance costs in

[18] A "user" is any individual who logs onto the system to achieve a specific purpose (e,g., data entry, report generation, application maintenance).

the A-43. As a result, Federal accounting systems do not provide mechanisms for separate accounting of these costs. These are issues which we plan to address in our governmentwide audit report to OMB.

In the absence of adequate Federal criteria, we are pleased that the Agency is establishing its own policies, procedures, and guidance. For example, Chapter 17 of the IRM Policy Manual was revised and clarified the use of life cycle costing. Also, an OIRM official indicated that additional guidance, procedures, and standards will be forthcoming which cover life cycle cost methodology for application systems. Finally, on March 15, 1995, the Executive Steering Committee approved the first Agency-wide Strategic IRM plan which explicitly linked Information Resources to the Agency's budget.

RECOMMENDATIONS

We recommend that the Assistant Administrator for Administration and Resources Management, in his role as the Designated Senior Official for IRM and, when appropriate, in conjunction with the Executive Steering Committee for IRM:

3-1. Ensure that the CFO project related to project cost accounting provides the ability to accumulate system level costs. Continue to coordinate these efforts with the working capital fund initiative.

3-2. Incorporate requirements for the accumulation and capitalization of all new development costs and major enhancements which meet the $5,000 capitalization threshold into the project cost accounting project. Establish an interim process to accumulate major system costs to be capitalized. These costs should be incorporated into the financial statements as appropriate.

3-3. Change the OMB Circular A-11 40B report on financial system obligations to reflect system costs for telecommunications and timeshare reports.

3-4. Require the completion of a feasibility study for replacing or modifying the timeshare management system to provide accurate levels of workload accumulation for individual major systems for both NDPD capacity planning and system managers.

3-5. Provide the capability within the system access and accounting systems to capture and accumulate resource utilization costs for the different life cycle phases of each information system (e.g., maintenance programming, operations, user access, etc.).

3-6. Establish thresholds to enforce the requirement of cost-benefit analyses for all major changes to application systems, using the criteria for system classification outlined by EPA Directive 2100, Chapter 17. The cost-benefit analyses should provide managers, users, designers, and auditors with adequate cost and benefit information to analyze and evaluate alternative approaches. The cost-benefit document should contain a summarization of the criteria used in the evaluation as well as the estimated costs and benefits.

AGENCY COMMENTS AND OIG EVALUATION

OARM's March 17, 1995, response to our draft report indicated that they agreed with two of the six *revised* recommendations and partially agreed with the other four recommendations. In addition, OARM has already initiated action on four of the recommendations.

OARM partially agreed with Recommendation 3-1. However, it is unclear from the response whether or not the Project Cost Accounting System will be used to track system costs or whether OARM intends to include the tracking of system costs as part of their annual guidance on use of the expanded account code structure.

OARM partially agreed with Recommendation 3-2. While OARM does not agree that the Project Cost Accounting System is the most appropriate place for implementation of capitalization, they agree that appropriate software costs should be capitalized and reflected in the Agency's financial statements.

OARM also partially agreed with Recommendation 3-5. As stated in the response to Recommendation 3-4, OARM has already initiated enhancements to TSSMS which will improve the accuracy of usage information. However, OARM stated that TSSMS will not be able to track the purpose of system usage and that they believe such tracking to be a system management function. We believe that the introduction of the enhanced TSSMS software should include system manager training in establishing more reasonable account codes for tracking costs associated with various life cycle stages (e.g., maintenance programming, operations, user access, etc.).

Finally, OARM partially agreed with Recommendation 3-6. While they agreed with the intent of this recommendation, OARM expressed concern regarding incremental cost-benefit analyses. However, the recommendation does not require comparison of total system benefits with incremental changes. The main point of the recommendation is to establish thresholds as criteria to determine *when* a cost-benefit analysis is needed.

CHAPTER 4

SOFTWARE CHANGE CONTROL AND CONFIGURATION MANAGEMENT PROCESSES ARE NOT ADEQUATELY MANAGED

SOFTWARE CHANGE CONTROL REQUIREMENTS AND GUIDANCE

We used the following list of Federal requirements and guidance to conduct our audit in the area of software change control and configuration management. Federal guidelines, as well as a number of industry publications, were used to form a framework of sensible, stable business practices and, therefore, served as a means to evaluate software maintenance activities. Refer to Appendix IV for more detailed discussion regarding the Federal criteria. For details on applicable Agency criteria and applicable maintenance services, refer to Appendix V.

- Public Law Nos. 99-511, 99-591;
- OMB Circular Nos. A-130, A-132;
- FIPS Publication Nos. 38, 106, 132;
- National Bureau of Standards Special Publication 500-129; and
- EPA Directive 2100, entitled "Information Resources Policy Manual."

EPA'S SOFTWARE CHANGE CONTROL PRACTICES COULD RESULT IN CRITICAL PROBLEMS

Numerous aspects of software maintenance were reviewed in connection with the audit of change control and configuration management practices. A synopsis of the audit findings by application system is presented in Appendix VI, to provide a quick reference for system managers, thereby facilitating their review of these software maintenance activities.

Throughout this chapter, we used examples of application system practices in order to illustrate the vast differences in change control and configuration management practices within the Agency. The examples describe both structured and weak processes, and we recognize that the complexity or uniqueness of an individual application system may necessitate certain differences, in some cases. Whereas the observations relate examples from the ten

application systems reviewed, our remarks and recommendations correspond to the Agency's general practices in the area under discussion.

In addition, EPA's OIRM has taken a number of significant steps to implement controls over the management of software modifications to its application systems. For example, OIRM took the initiative to research and implement a software configuration management tool on its Integrated Financial Management System. The product, ENDEVOR, will strengthen impact analyses for software changes, ensure version controls are in place, and force audit trails for emergency software changes. OIRM also recently updated its Change Management System processes to improve the basic features and controls of the tracking system.

Performance Measurement Indicators Needed

Everything done to software affects its quality. Therefore, Federal guidelines on software maintenance suggest that "measures" be established to aid in determining which category of changes are likely to degrade software quality. Performance measurement is a recognized tool which can improve the quantity and quality of information needed for decision-making purposes.

Management did not *quantify* effectiveness of change control processes through measurable indicators (e.g., number of problems or changes, types of changes, number of changes to each module, etc.) in any of the ten application systems reviewed. Although various types of historical data were recorded in change control logs, discussions with application system managers disclosed that the collected data was not measured against pre-established targets and that no managerial analysis was made of the collected data.

As a result, system managers do not have the information they need to effectively analyze and manage software maintenance. A complete software management system would include a performance monitoring feedback loop with successive cycles of goal setting, performance monitoring, and regular reporting. Such a system requires regular, efficient information collection, analysis, and review. A performance measurement system in some instances simply formalizes, makes more efficient, and makes more explicit the decision-making process managers use intuitively. Such a system also forces managers to confront hard evidence about program efficiency and effectiveness.

Without defined measurable performance indicators, system managers lack a valuable tool to help them: (1) evaluate the effectiveness of their change control process; (2) assess the overall stability of the application system; or (3) draw conclusions as to how program office resources should best be utilized. Therefore, system managers will

continue to authorize maintenance activities without the benefit of information which could significantly influence their decisions.

In addition, attempting to manage software maintenance processing without benefit of measurable indicators may result in EPA paying excessive amounts in contract and award fees. Since most of EPA's software maintenance is performed under cost-plus-award-fee contracts, contractors may be inclined to perform unnecessary changes to a system or to rework faulty changes with the intent of charging EPA for unnecessary labor costs. Various management controls, if consistently applied, may effectively mitigate these potentially threatening effects.

Inconsistent Use Of Standardized Change Control Request Form

FIPS Publication 106 states "There must be a well-defined mechanism for initiating a request for changes or enhancements to a system." A standardized form ensures adequate information is collected for classifying, reviewing, and processing change requests. Therefore, all changes should be formally requested and submitted on a standardized form. The decision and reasons for acceptance/rejection of a change request should also be recorded and included in the permanent documentation for the system.

Five of the application systems reviewed (AIRS, CERCLIS, CPS, IFMS, and TRIS) had procedures which demonstrated a standardized approach for collecting pertinent data for *all* types of software changes. Four of the five systems used a standardized written request form to initiate all software modifications regardless of the level of effort which might be expended in implementing the requested change. The standardized form required the initiator to provide *specific* facts regarding the proposed change to facilitate its review and subsequent tracking. Through the automated CMS, IFMS change request information was collected in a standardized on-line format. The five remaining application systems (EPAYS, FINDS, GICS, PCS, RCRIS) had procedures which used a mixture of formal and informal methods to initiate software modifications. In several cases, the anticipated level of effort was the key factor in determining the exact submission channel and whether or not standardized submission data was required.

FINDS and GICS had neither a formal set of procedures to instruct users on how to initiate a request for software maintenance, nor a standard form to facilitate that process. In addition, RCRIS used a standardized "issue paper" format to obtain sufficient data for proposed major system and programmatic changes. However, minor enhancements and routine operations and maintenance changes were initiated in a different manner and did not require a sufficient or standardized level of information. The Centralized Problem Management System (CPMS) tracked these routine types of changes, but the data it contained was not sufficient for evaluation or monitoring

purposes. Formal policies for PCS also required that requests for changes be submitted in writing, through appropriate channels. However, the policies did not identify a standard format for submission of change requests, or indicate the type of detailed information required for proper identification, review, and processing of a change request.

EPAYS procedures only required use of a standardized change request form for particular *types* of proposed software changes. EPAYS users used a System Enhancement Request Form (SERF) to initiate changes pertaining to minor system enhancements or routine maintenance requests. However, due to the nature of EPAYS, an abundance of software maintenance items were received and processed as *emergency* fixes with Emergency Technical Assistance Documents (ETADS) or "hotline" changes. No defined format existed for reporting the details related to problems identified and fixed through these two channels. The magnitude of changes effected through these non-regulated changes was substantial. Of the 4,000 modifications made to EPAYS[19] during 1992, only 107 were submitted using a SERF.

Without a standardized change request form, sufficient data may not be available to adequately evaluate the nature of a proposed software change or its resulting benefits, cost, or impact on the application system. Management may waste valuable time and resources trying to discern the nature of the requested change or returning a request for further information. In addition, since some change control boards only convene periodically, inadequately justified or detailed change requests might be returned to the requestor for clarification and, therefore, have to wait until the next meeting to be reconsidered.

It is also possible that a review board may disapprove an important and necessary change due to a lack of information. Conversely, a marginally justified or defined request for change could be approved despite the lack of available data. In addition, approved changes could be misclassified or ranked inappropriately due to insufficient information.

Varying Classifications Of Requests For Change

As we discussed in Chapter 2, FIPS Publication 106 identifies the three following categories for monitoring and controlling software maintenance activities: corrective[20], adaptive[21], or perfective[22].

[19] This number reflects both source code and data modifications.

[20] Corrective maintenance refers to software changes which are necessitated by actual errors and, therefore, are a reactive process required to keep the system operational.

[21] Adaptive maintenance includes changes which are beyond management's control.

These three categories adequately cover all possible types of software maintenance and facilitate analysis procedures.

System managers did not classify requests for change in a manner which would facilitate analysis of software maintenance trends. None of the systems reviewed used the classification types generally identified in industry publications and Federal guidance to categorize software maintenance requests for change. Instead, many systems categorized requests for change based on the level of effort involved with the maintenance activity.

Use of the categories identified in Federal criteria would facilitate meaningful management analysis of historical maintenance data and promote consistency within Agency software change control processing. For example, "corrective" maintenance customarily accounts for 20 percent of all software maintenance and would encompass design errors, logic errors, and coding errors. Therefore, system managers noticing a 40 percent corrective maintenance rate would be alerted to existing or potential problems regarding: (1) system stability; (2) inadequate programmer design or coding; or (3) inadequate change controls testing or review procedures.

Not one of the application systems reviewed distinguished proposed changes as "adaptive," although numerous instances were noted in which this would have been the proper classification. Adaptive changes are considered beyond the control of the software maintainer, since they constitute effort which is initiated as a result of changes in the environment in which the system must operate. Adaptive changes include changes in laws and regulations, as well as hardware and system software configuration.

Although most of EPA's major application systems performed a "classification" process during the initial review of a proposed change request, these classifications did not correspond to the Federally recognized categories. For example, CPS managers classified proposed software changes based on the estimated level of effort required. Proposed changes requiring more than 160 hours of effort were classified as "development," while all other changes were considered "maintenance." The procedural paths for CPS changes differed based on this classification. Changes initially identified as "development" effort did not undergo further classification and were never tagged as being related to perfective, corrective, or adaptive type maintenance. "Maintenance" changes within CPS were further categorized as either problems, enhancements, or emergency changes, but never as a mandatory adaptive change.

Similarly, TRIS classified change requests as either Class I or Class II. This distinction was primarily based on the extent of the effect of the modification on the functional or technical characteristics of

the product baseline. Therefore, the classification made was irrelevant to the nature of a change or the motive behind its initiation.

CMS, which acts as the recognized change control system for IFMS, presented the closest adaptation of Federal and industry guidance by classifying requests in one of five "priority" categories:

- Emergency or System Crash,
- Major System Faults,
- Statutory (Regulatory) Requirements,
- Changes to Save Resources, and
- Nice to Have

While this approach is more informative than a level of effort classification, it also could inhibit constructive analysis, since "corrective" changes could be classified as one of two possible categories: Emergency, or Major System Faults. Similarly, "perfective" modifications could be classified as either Major System Faults, Changes to Save Resources, or Nice to Have. The abundance of classification categories could obscure the monitoring and interpretation of maintenance trends.

If managers cannot identify which changes are corrective, adaptive, or perfective, they cannot make effective decisions about software maintenance priorities. In times of budget cuts, managers are required to make painful decisions about deferring software maintenance. Corrective and adaptive changes frequently cannot be deferred. This leaves perfective changes as the source of discretionary changes which may need to be deferred for budget reasons. During interviews, some system managers indicated that they receive a budget for contractor maintenance and a list of changes approved by the Change Control Board, and process changes from the approved list until the contract dollars have been exhausted.

Based solely on the current types of classifications used to distinguish requests, many system managers may find it difficult, if not impossible, to efficiently identify trends or patterns in maintenance activity which could indicate weaknesses in the change control process. Unless the following issues are routinely and thoroughly evaluated by other measurable review mechanisms, these problems could proceed undetected:

- excessive contractor rework of continuing problems;
- degrading software performance;
- inadequacies in the maintainer's performance;
- inadequacies in established review and testing procedures; or
- excessive maintenance costs for a particular type of software change.

In addition, management lacks insight regarding the true nature of the proposed change (adaptive, perfective, or corrective) and cannot reasonably evaluate the stability of the application system. Therefore, management might not be able to make informed decisions regarding the appropriateness of future modifications.

Centralized Change Control Review Not Always Performed

According to FIPS 106, the key to controlling changes to a system is the formal requesting of changes and the centralization of change approval. A centralized approval process will enable one person or group of persons to have knowledge of all the requested and actual work being performed on the system. Prudent management practices suggest that executives and senior management fully participate in and take responsibility for all major information management project decisions, throughout their lifecycle.

Seven of the ten application systems reviewed did not have a fully[23] centralized change review process through which *all* proposed software changes flowed. Only RCRIS, CERCLIS and CPS had *fully* centralized approval processes for system software changes. For CPS, all proposed changes (excluding emergency changes) were reviewed and approved by the CPS Steering Committee, which met on a monthly basis. Emergency changes were approved by the National Contract Payment Division's Financial Systems Section (FSS) Chief, and a general reporting was made to the Steering Committee regarding the number and type of ad hoc requests performed. RCRIS used a predetermined panel of reviewers to evaluate and prioritize major system and programmatic enhancements, and funnelled routine O&M and minor system enhancements through a separate panel of Office of Solid Waste (OSW) personnel. For CERCLIS, the Project Manager made the final decision on approval of change requests. However, although procedures demonstrated a centralized control review, current practices indicated that senior management did not actively participate in the process through steering committees or periodic reviews.

The remaining seven systems varied greatly with respect to the degree of centralization in their change request review process. Some application systems did not have policies or procedures stipulating a centralized approval process; other systems stipulated an approval process, but the approval committees were either decentralized or did not meet in a timely manner.

[23] The term "fully," in this context, relates to the fact that all types of software changes flow through the same centralized control review point. Changes are not allowed to discretionally circumvent the established control point. Factors such as the following do not exempt the request from the established review and approval process: (1) department of origin for the change request; (2) level of effort associated with the proposed change; and (3) whether the proposed change is of a corrective or enhancement nature.

For example, AIRS used a *decentralized* approach for processing software changes, since it was comprised of four separate subsystems. Each subsystem's change requests were handled independently of other subsystems, but management used a central committee which met at regular intervals to coordinate overlapping issues. In addition, informal AIRS procedures required that a common subsystem "work assignment team" evaluate and approve the results of approach studies, performed in relation to proposed subsystem changes.

The PCS Steering Committee reviewed, approved, and prioritized requests for PCS software "enhancements," but only on a *yearly* basis. PCS "corrective" maintenance was coordinated by the EPA Software Coordinator or other EPA Information Management Section Staff; however, no set policies governed this process.

Both GICS and FINDS relied on SDC policies to dictate formal change control procedures. However, applicable SDC procedures did not address approval points within the system software maintenance process.

Without a centralized review and approval point of *all* proposed software maintenance projects, the following effects could occur:

- Similar enhancements to the system might be processed individually, thereby wasting limited resources, rather than combining their modification and implementation;
- A proposed software change could be implemented without evidence of an audit trail;
- A proposed software change could be implemented without sufficient impact analysis and, therefore, interact negatively with other application software components;
- A software change could be implemented despite the cost-effectiveness of the proposed modification;
- A proposed change of lesser importance might be implemented ahead of more significant and necessary modifications; and/or
- A software change could be implemented without proper update of related software documentation.

In addition, if software changes are processed without management's full knowledge, then managers might approve and process subsequent changes which would interact negatively with a previous, unrecognized modification. Similarly, two or more independent changes might conflict with one another thereby negatively impacting the application system's functions. In either case, the ultimate risk would be a production failure due to incompatible software modifications. Although program offices would ultimately be held accountable for their systems' production failures, most of the systems reviewed are ranked as "Major Agency" systems, due to their extremely high lifecycle costs and the "mission critical" nature of their influence on Agency operations. Therefore, a production

failure, regardless of accountability, would impact multiple Headquarters offices or Regions, perhaps even States and the general public, and could affect a substantial portion of normal business operations.

Also, as a result of inadequate program office review, *un*approved modifications; unintentional, malicious and/or fraudulent insertions of source code; or poorly structured or inadequately tested program code could exist and proceed undetected through the change process. In extreme instances, critical functional production problems, damage or loss of production data and, most likely, additional unwarranted costs to the cognizant program office could result.

In addition, without final program office review and approval, software changes could be implemented without adequately fulfilling all documentation and system requirements. This is a serious and factual risk, since inadequate system documentation could mislead program maintainers during future maintenance efforts or force them to analyze the source code, line by line, to understand the system. By the same token, out-of-date or incomplete user documentation could detrimentally affect the performance of the application users and result in inadequate training of new employees. Also, missing or incomplete design specifications increase the difficulty of future maintenance on the system. Without final management approval, it is also possible that modifications could be implemented without adhering to established rules and guidelines for review, testing, and quality assurance inspection.

Historical Review Process Needed

One of the important activities necessary to change control management is the analysis of problem reports and software changes. Requests for system changes, as well as system problems, should be reported and documented in a standard manner.

The data gathered by a change request and problem reporting system can be used to describe the changing quality of a system. Experience has indicated that sources of errors are rarely uniformly distributed across all modules of a system. Therefore, use of historical change control and problem data enables managers to: (1) identify modules which experience a high degree of problems requiring corrective changes; (2) identify modules repeatedly reworked due to deficiencies in the maintainers' performance; (3) identify modules requiring rework due to problems in testing and review procedures; or (4) determine whether it would be more cost-effective to redesign rather than continue maintaining these modules.

None of the application system managers periodically reviewed change control logs to discern patterns or trends which might be indicative of these types of problems. Likewise, management did not routinely

evaluate the stability of the application software or consider the effects of its possible instability in their management decisions. In most cases, available tracking systems did not contain a level of detail sufficient to facilitate informative trend analysis of prior software changes.

Our review disclosed that the historical logs maintained for the ten systems varied greatly with regard to the type and extent of information tracked. In most cases, the information accumulated proved insufficient for any reliable managerial analysis of software maintenance. For example, software changes affecting TRIS were tracked on daily and weekly user support reports. However, the collected information was often incomplete and numerous action items were not numbered for tracking purposes. In addition, the information contained within the reports would not facilitate management decisions, since it identified neither the type of change being processed (i.e., corrective, perfective or adaptive) nor the modules affected by the change. During fiscal 1994, TRIS management attempted to establish a "QA Tracker System" to serve as a comprehensive monitoring system for reported problems and requested software changes. The system, however, was abandoned after five months in operation.

For several of the application systems reviewed, we examined copies of NDPD Problem Management Detail Reports, as evidence of their change/problem tracking logs. These reports did not provide sufficient information to discern: (1) the type of software change being processed; (2) the modules to be changed during the process; or (3) whether an item was indeed a software change or actually a non-software related problem.

IFMS management required two separate systems to track a software request from its initiation to its final implementation into the production environment. OARM identified CMS[24] as the system which encompassed all aspects of change control and configuration for the IFMS family of programs. CMS is a prototype system designed to provide standardized processes and procedures to control and track changes to application systems. However, in our opinion, CMS was limited in its usefulness and did not qualify as a "self-sufficient" system, since it was not able to track the software change requests through to their implementation in the production environment. Despite recent improvements in its software, CMS still did not identify the programs or modules affected by a software modification, nor track the date the software change was processed into production.

[24] Although still reportedly "in development," CMS is used by IFMS in a production capacity and it is anticipated that CMS will become the model for many major EPA systems in the near future. In fact, EPAYS management expected CMS would be installed on their application system before the end of Calendar Year 1994.

Within CMS, the designation "Implemented" referred to the closure of a CMS Work Order; it did not mean that the modified source code was put into production.

The tracking and reporting abilities of CMS would need to be expanded to increase the usefulness of its data from a management and historical perspective. Despite its name, CMS did not operate as a fully-functioning change control system. In our opinion, the focus of CMS lay in the tracking of contractor work order requests and pertinent documentation related to those requests. However, since requests for software modification represented only 80 percent of the work order requests processed through CMS, action items initiated in response to requests for software modification could lose their visibility.

Without routine assessment of historical information, system managers cannot adequately evaluate the effectiveness of the change control process or determine whether the costs involved in the maintenance of system software justify a partial or overall redesign of the application. Similarly, a potential risk exists that continuing and perhaps excessive software maintenance costs will be incurred due to the constant rework of systemic application problems or the maintainers' inability to produce effective corrective changes. In addition, fixing and then re-fixing problems on a continual basis tends to increase the difficulty of programming and could result in overly confusing, complex, and unstructured source code.

Also, repetitive changes to application systems may result in the deterioration of a system's performance or its inability to satisfy functional user requirements. A potential risk exists that an application system might deteriorate due to an endless succession of "quick fixes" and "patches" to the source code, without timely detection by responsible management officials. Without routine and thorough reviews of the changes processed to the application system, management may not be able to:

- determine existing maintenance trends or detect their probable cause(s);
- assess the overall stability of the system;
- identify programs or modules which are routinely modified or reworked;
- make informed decisions with regard to the future of the system, its impending obsolescence, or necessary redesign of the application system; and
- assess whether unnecessary or inefficient effort was expended or rework of previously reported problems is evident. Both could contribute to excessive contract and award fee costs.

Coding Standards And Reviews Needed

Source code guidelines and standards aid maintainability by providing a structure and framework within which systems can be developed and maintained in a common, more easily understood manner. Source code standards also promote productivity, software sharing, and reuse. Code standards should be applied whenever designing a new system application or modifying an existing one, and must be continually enforced via technical review and examination of all work performed by the software maintenance staff.

Modified source code should be reviewed during the maintenance process to determine how well code adheres to the established coding standards and to guarantee a high degree of uniformity across the software. This becomes a critical factor when someone other than the original developer must understand and maintain the software. Since most of the systems reviewed were of a considerable age, it is reasonable to conclude that numerous maintainers had performed modifications to the source code.

None of the ten application systems reviewed developed supplemental coding standards for software development or maintenance purposes. No application system used a source code policy which encompassed the following Federally-defined, basic principles: (1) single high-order language; (2) coding conventions; (3) structured, modular software; (4) standard data definitions; (5) well-commented code; and (6) compiler extensions. Although some application systems managers cited general EPA guidance as a basis for source code standards, all referenced procedures were inadequate for source coding purposes.

For example, several application system managers cited EPA's "System Design and Development Guidance, Volume C," and its accompanying "Operations and Maintenance Manual," as the standard for source coding requirements. The Agency guidelines listed various important, but decidedly general, characteristics for source code standards. However, the Agency guidance did not establish rules or measurement criteria to specify "how well" the code must be written, organized, or formatted. Without such a mechanism, it would be difficult to assess the quality or clarity of the written code, two aspects which directly affect the maintainability of the source code.

Documentation of FINDS, GICS, and certain other application systems referenced the applicable SDC policies for source code standards. However, although these policies included naming standards, screen standards, and error message standards, they did not address source code standards for software maintenance.

During an audit interview, a system manager remarked that ADABAS "Natural" was a self-documenting language and, therefore, presumed that programming would be relatively easy to interpret. Managers also stated that they relied on NDPD Natural coding standards and the associated reviews performed by NDPD technical consultants to detect poorly written source code.

In fact, NDPD's Natural 2 Program Code Techniques did provide more details regarding how source code should be written and what pitfalls should be avoided by code programmers. However, the referenced code standards did not contain general rules which addressed the basic coding principles or identified and set measures against which written code could be evaluated. In addition, NDPD technical consultants advised us that their pre-production testing was limited to "production" acceptance requirements. Standardized checklists were used to facilitate the NDPD Test & Assurance (T&A) process, but few, if any, checks were made regarding adherence to Natural source code standards. Rather, the NDPD T&A procedures routinely checked the program's use of standard function keys, report headers, etc.

CPS was a unique case since maintenance effort was not contracted out, but instead performed in-house by full-time EPA employees. Again, these standards pertained to existing JCL, program naming conventions, and screen layout standards, rather than establishing guidelines to ensure that new or modified code was consistently structured in an efficient and comprehendible manner.

Since every programmer has their own unique style and creativity, source code written as a result of continuing maintenance activities may become unstructured, overly complex, and not adequately commented[25]. The absence of adequate coding guidelines may cause confusion and difficulties for future maintainers of the system. Inadequately formatted and commented program source code could lead to a lack of continuity in system operations and/or the unnecessary expenditure of significant funds. If the current maintainers change, poorly structured or overly complex source code could disrupt the continued flow of emergency and routine software maintenance operations.

In addition, not having a person other than the original author review the code increases the risk that improperly written or structured code could be approved and implemented into the production environment. It may not have an immediate effect on the maintainability of source code, but the cumulative effect of numerous modifications could jeopardize system operations.

[25] "Commented" refers to any notes a software developer includes in the source code. The comments are made in English and may identify the purpose of the particular code or relate to other notable details which are pertinent to future modification of the program.

Inadequate Test Plans And Analysis Of Test Results

Testing is a critical component of software maintenance. Testing procedures promote software quality, maintainability, and overall system integrity. As such, the test procedures must be consistent and based on sound principles.

During the test stage, the software and its related documentation should be evaluated in terms of readiness for implementation. The goal of testing is to find errors and, therefore, a test plan should: (1) define the degree and depth of testing to be performed; (2) describe the expected output; and (3) test for valid, invalid, expected, and unexpected cases. Federal guidelines outline the format and content of test plans and test analysis reports, and emphasize the importance of:

- identifying and segregating the various functions of the program to be tested;
- describing the strategy and limitations of the testing; and
- describing the input data and expected output data for each planned test.

Four application systems (AIRS, TRIS, IFMS, RCRIS) were chosen for the review of specific software modifications and applicable test documentation. Test documentation was not available for many of the software changes selected for review. In all four application systems, we determined that test plans and related test analyses did not meet the level of detail established by either: (1) the individual system's standards for test documentation; (2) OIRM's Operations and Maintenance Manual standards; or (3) Federal standards. The only aspect of testing to be consistently applied within the examined systems was the use of a separate environment for the performance of test activities.

RCRIS software maintenance was handled under a contract and was subject to EPA SDC policies and procedures governing the development and execution of test plans, as performed by the contractor. Although the SDC procedures were lacking in some specific respects, they did require that both formal written test plans and test incident reports (TIRs) be prepared by the contractor's independent testing organization. However, RCRIS test plans consistently did not include the basic factors identified in SDC or Federal guidance. Notably, the test plan did not define the expected output or identify steps to test for valid, invalid, expected, and unexpected cases. Similarly, test analysis reports were not included for any of the software changes sampled under RCRIS. Although the RCRIS changes sampled did include computerized printouts of test runs, these documents did not clearly indicate whether the tests passed or failed.

Software changes to IFMS, implemented through the Administrative Systems Division (ASD)'s Change Management System, were subject to the test procedures and requirements outlined in ASD/Systems Support Branch (SSB) Joint Application Development Appendices, dated April 20, 1992. These procedures outlined detailed requirements for the formation and review of test plans. However, during our review of recently implemented software changes, no documentation could be provided to support the use of test plans or test analysis reports.

Changes to AIRS were only subject to the guidelines established in EPA System Design and Development Guidance, as well as its companion guide, EPA Operations and Maintenance Manual. AIRS management stated that test results were often reviewed on-line, and therefore, they were not able to provide test analysis reports for the software changes selected for review. Since AIRS management had not maintained the audit trail by electronically storing their test analysis results, we could not be assured that test results were generated, reviewed, or analyzed. In addition, AIRS test plans consistently did not include the basic features outlined in Federal guidelines.

The development of TRIS test plans and results was governed by a *draft* policy entitled EPA/OPPT Test and Evaluation Development Plan, dated December 14, 1992. This was a very comprehensive policy which adequately defined the variety of tests to be performed and the degree and depth of testing required. However, a majority of the TRIS changes sampled did not include test plan documents or test analysis reports. Those software modifications which included test documents did not adhere to their draft policy requirements or Federal guidelines.

As a result, changes to software may not be sufficiently tested to account for all valid, invalid, expected, and unexpected outputs. The relevant limitations on the test, due to the test conditions, may not be recognized by the performing contractor, and therefore, test results could be interpreted to be conclusive when, in fact, they are not truly representative of the actual production environment. Similarly, not all functions of proposed software changes may be exercised during overall tests. Insufficient testing and analysis of test results could result in source code which fails when introduced to the production environment, due to unforeseen transaction conditions, interfaces, or user input.

Inconsistent Testing And Acceptance By Functional Users

"Acceptance[26]" testing is commonly performed by the functional users after system testing has been completed. The software maintenance process is not considered complete until the user has accepted the modified system and all documentation has been satisfactorily updated.

The functional user community was not given the opportunity to perform user acceptance testing in two (FINDS and GICS) of the ten application systems reviewed. The results obtained from the contractor's unit and system tests were forwarded to NDPD for implementation without thorough review by technically competent program office personnel. In these cases, the responsible program office did not perform a *functional* assessment of the modified software to ensure that only authorized work was performed and that all requirements of the change request had been met and the system functioned according to specifications.

The remaining eight application systems provided varying degrees of user interface before implementing a modification into the production environment. RCRIS provided an excellent four phase sequence for software testing. In addition to unit[27] and integration[28] tests performed by the development programmers, changes were subjected to functional quality assurance testing by an independent SDC test group. In the fourth phase, the user community evaluated changes for a period of approximately 30 days. This acceptance testing was completed prior to placing the modified software into production.

Modifications to AIRS subsystems were reviewed by environmental engineers, associated with the particular subsystem's review group, in order to assess functionality of the software. Both CERCLIS and PCS system managers stated that "beta[29]" testing was performed to fulfill the need for user acceptance testing. PCS management stipulated that both Regional and State offices performed beta testing of all major system enhancements for a period of at least

---

[26] Acceptance testing is considered the "last line of defense" for the end user. The end users perform functional tests on the modified software using live data, test data, or a combination of data.

[27] The testing of a single module or a related group of modules.

[28] Integration or "string" testing is performed to ensure that the interfaces between programs/modules are functioning properly and that transactions or data pass between programs. System integration testing is the progressive linking and testing of system components to ensure that they work as a complete system.

[29] Beta testing, although not a universally-recognized term, refers to a second level of testing which is performed by a separate group of individuals from those who executed the first set of tests on the modified software. Beta testing is performed by a group independent of the software programmers and is used to confirm the results of the first functional tests.

three weeks. Likewise, OSWER Management Systems Staff performed the testing for CERCLIS, but stated that the period of time allotted for such testing varied. The scheduling and length of time allotted for user acceptance testing could significantly effect the ability of users to thoroughly evaluate system functionality. For example, if user testing was restricted to weekends, the number of users available to access and evaluate system performance would be limited. In addition, a severely limited test period might restrict the range of transactions which were tested against the modified software.

CPS large-scale development efforts were subjected to user acceptance testing, while lessor functional software changes required systems personnel to conduct "Train the Trainer" sessions with a user representative from the affected functional area. Once trained, that user was responsible for training other users. Although training users on functional changes is an essential part of the cycle, it cannot serve as a substitute for testing the boundaries of modified software's performance. In the aggregate, smaller changes can have a profound impact on the performance of the system and, in fact, might introduce additional functional problems which might not surface during simple training exercises.

Similarly, ASD identified "user acceptance testing" as one of the defined processes for testing those modifications to IFMS which originated from ASD. However, the written procedures did not specify the required duration of user testing nor identify the types of software modifications subjected to this type of testing. OARM indicated that the Financial Management and Budget Divisions function as "clients" to ASD and as such, have specific branches which are responsible for interpreting and managing user requirements as well as changes. However, despite the fact that OIG requests for IFMS test documentation were very explicit and comprehensive, the supplied documentation did not provide support that "user acceptance" testing is performed by any of these "client" divisions.

Draft TRIS procedures identified "user acceptance testing" as one of the formal tests for software changes. However, management advised us that it was only the Work Assignment Manager (WAM) who reviewed, tested, and accepted the software changes prior to forwarding the modifications to NDPD for implementation. In fact, our review of detailed TRIS "incident reports" confirmed that a WAM for TRIS knew that problems existed with a particular software modification and still accepted the work for implementation into production.

In addition, for several of the ten systems, cognizant program offices placed too much reliance on the final testing performed by NDPD Technical Consultants (TC)[30]. The testing performed by NDPD TCs[31] only qualified as "production acceptance testing." Under production acceptance testing, the consultants reviewed acceptance packages for performance issues and for interaction with the central database environment. Since the TCs did not know the purpose of the modified application code, they could not guarantee its performance with regard to other aspects and functions of the application. NDPD TCs did not "accept" an application with respect to its entire function. Therefore, NDPD's production acceptance testing should not take the place of functional end-user acceptance testing.

Also, in seven (CPS, EPAYS, FINDS, GICS, IFMS, RCRIS, and TRIS) of the ten application systems, there was varying, if minimal, evidence to support that software changes were *formally* reviewed, approved, or "closed" by either the cognizant program office's Change Control Board or responsible originating management officials prior to their migration to the production environment. Formal management review would officially recognize that the software modification had been reviewed by non-contractor technical personnel and that the end-user community found the modification satisfactory. Additionally, our review of change control processing documents disclosed that the formal authorization to proceed with implementation was seldom evident.

As a result, implemented changes, both perfective and corrective, may not satisfy user requirements, since neither the functional user community nor responsible system managers provide feedback on the adequacy of a modified change prior to its implementation. A lack of interaction by the user community could waste time and budgeted resources and also result in repetitive requests to solve previously unfulfilled user needs.

More Comprehensive Verification And Validation Testing Needed

Software Verification and Validation (V&V)[32] testing is recommended for both critical and non-critical application systems, and is

[30] The IFMS application was granted exceptions to the central database management system review and testing requirements. The contractor, AMS, controlled and performed approximately 85% of the production reviews normally performed by NDPD. After the fact, AMS provided copies of acceptance activities to the technical consultant at NDPD.

[31] Technical Consultants are contractor personnel. Consultants are assigned to a particular application system(s) in order to: (1) oversee the implementation of software changes; (2) monitor software performance; and (3) prevent and/or detect potential technical problems.

[32] Verification and Validation testing is a formal check and balance effort which monitors and evaluates software as it is being built. V&V consists of tasks from a broad spectrum of analysis and test techniques and is performed to determine functionality, uncover performance of unintended functions, and ensure the production of quality software.

usually performed separate from the development groups' testing. V&V uses a structured approach to analyze and test the software against all system functions and all hardware, system users, and other software interfaces. Through V&V techniques, high risk errors are detected early, software performance is improved, and higher confidence is established in software reliability.

Since independent V&V testing is an added expense to the change control process, we recognize that this procedure is best directed towards those software modifications which represent major changes to system functionality or require a large level of re-programming effort. In these cases, the cost of conducting independent V&V is offset by cost advantages of early error detection and improved software reliability and quality. FIPS Publication 132 encourages the use of minimum requirements for the format and content of Software Verification and Validation Plans (SVVP). Even if the originally developed software was not verified under this standard, a new SVVP should be written to test major modifications made during the lifecycle's maintenance phase.

Nine of the application systems reviewed conducted some degree of formal and independent V&V testing for software maintenance changes, as indicated by their respective written procedures and policies described below. Modifications to TRIS and PCS underwent formal tests which might be loosely interpreted as partial V&V testing. Similarly, CPS defined quality assurance test and acceptance procedures which might parallel some of the objectives of V&V testing. However, no AIRS modifications, regardless of their level of significance, were subject to independent V&V testing prior to implementation in the production environment.

Software maintenance changes to RCRIS, CERCLIS, GICS, EPAYS, and FINDS were subject to EPA SDC product assurance policies. These SDC product assurance activities, defined in SDC Guideline #2, identified four processes, two of which would be classified as independent V&V testing. A separate independent group within the maintainer's organization performed these tests. In addition, software maintenance changes to CERCLIS were subjected to separate formal V&V testing, as stated in the EPA SDC Policy defining configuration management procedures for particular Superfund systems. Under this policy, an independent product assurance group within EPA's SDC was responsible for ensuring that products were congruent with past similar products and all requirements and specifications were satisfied. However, neither of the aforementioned EPA SDC procedures identified various V&V tests and analysis techniques which might be used to achieve product quality assurance. Nor did the applicable SDC procedures state which types of software modifications would definitely be subjected to V&V testing.

TRIS also had some degree of V&V testing for software modifications. Draft TRIS policies, covering test and evaluation plans, stipulated a comprehensive list of formal tests to be completed prior to implementation. This formal approach incorporated tests which could be interpreted as V&V testing. However, the test approach did not include reviews to check for unintended code modifications. In addition, the draft policy stated that formal tests would be the responsibility of "Quality Improvement" (QI), but did not identify or ensure the independence and objectivity of QI participants.

Similarly, PCS management indicated that a reviewer performed testing to ensure that the software had been developed as specified in the general design and reflected in the PCS user manuals. However, that testing, referred to by PCS management as "alpha testing," did not fully characterize V&V testing.

When major software modifications are not subjected to independent V&V testing, system management lacks *objective* assurances regarding the content of application source code, the completeness and usefulness of system and user documentation, the interaction of application system baseline components, and the stability of system operations. Therefore, any of the following effects could occur:

- Software changes could be implemented without the completion of relevant system and user documentation;
- Unapproved software changes could be introduced into the production environment; or
- Important testing, review or approval procedures could be inadvertently or intentionally omitted from the maintenance cycle, in an attempt to expedite the modification process.

These effects could result in application software which is difficult to maintain. Also, the adequacy of user documentation could be compromised, thereby affecting the training of new employees or the performance of established application users.

Inconsistent Use Of Scheduled Maintenance Plans

Planned and scheduled maintenance activities furnish users with periods of stable operation and known system performance characteristics. Where changes are implemented in a regulated and scheduled environment, users can be informed of pending changes in a timely manner and receive appropriate instruction with regard to the new operating procedures or functional capabilities. Limiting implementation of software changes to regularly scheduled events enables management to exercise tighter version control over the numerous modules and programs which comprise these application systems.

Notable differences existed between the ten application systems reviewed regarding planned and scheduled maintenance activities. Some application systems, such as RCRIS, PCS, CPS, and CERCLIS, performed software maintenance activities at regularly scheduled times, and employed "packaging" techniques to group similar software changes for processing at the same time. Projected implementation dates and level of effort data were established at the onset of maintenance effort. Also, IFMS held quarterly Executive Management Group meetings to encourage consolidation of compatible modifications, thereby minimizing the number of adjustments application users would be forced to accommodate.

However, other application systems, such as EPAYS and AIRS, did not implement software changes on a periodic, scheduled basis. Although AIRS management indicated that users were informed prior to an impending change, changes were not made at predesignated intervals. In fiscal 1993, 4,000 software modifications were made to EPAYS and over 200 to AIRS. Therefore, users of these application systems could be confronted with constantly changing application capabilities and user requirements.

Continuous, unscheduled software modifications could contribute to the instability of an application system. Without policies to regulate the implementation of software modifications, application users could be continuously bombarded by new operating procedures, changing data input requirements, varying screen presentations, and new or modified system functions or reporting capabilities. Also, constantly changing software might create situations where user training would be compromised. In such instances, user training might be performed with little notice or preparation or the instructions provided might be less comprehensive due to time limitations. In addition, continuous modifications to a system of interrelated programs, modules, and tables would add to the difficulties of ensuring proper and effective control over successive versions of the software components. This could complicate a return to a prior software version should a software modification fail when put into production.

Version Control In Software Configuration Not Used

According to the GSA Guide for Acquiring Software Development Services, software configuration is defined as an arrangement of software parts, including all elements necessary for the software to work. Configuration management refers to the process of identifying and documenting the configuration and then systematically controlling changes to it to maintain its integrity and to trace configuration changes. Since no real-world software exists in only one version, it is very important to be able to identify which version of a module is associated with a particular program configuration. Version control allows program developers and maintainers to locate the latest

version of a program accurately, reliably, and consistently. Version control also enables system managers to roll-back to prior operable configurations of an application should a newly modified version fail to operate correctly once installed in the production environment.

Many of the application systems reviewed had a "baseline[33]" which dated back to the development or purchase of the system. Since implementation, numerous software changes were made to the components which comprised those baseline configurations. In fact, several of the application systems reviewed had software configurations composed of thousands of computer programs or modules.

Our review determined that many of the application systems did not use a library management system or a software configuration management (SCM) tool to track previous versions of software components, store backup copies of the source code, or identify the software configuration for each prior version of the application. At least three application system managers stated that they would rely on NDPD staff to provide backup copies of the software, in case a recently implemented version of the software failed. Even if each module within an application contained a "constant" to track successive version modification numbers, it would not assure the software configuration for the total production version of the application was recognized and recorded.

Prior to the commencement of this audit, OIRM initiated action to research and select a suitable SCM tool which could be installed on its various information systems. In December 1994, the chosen SCM product, ENDEVOR, was installed on IFMS. To date, no other application systems under OIRM's control have implemented the ENDEVOR product.

Several application systems chose to control access to source code through library management systems[34] (i.e., Librarian). Although librarian management tools are able to track successive versions of software programs, such utilities cannot recognize which version of each software component comprised the prior production release of the application. Nor can librarian products group modified software components together as a subrelease for migration to the production environment. Librarian management utilities are lacking in other respects that significantly affect configuration management. Unlike specific configuration management products, librarian utilities lack features which perform impact analysis, force audit trails for

[33] Baseline refers to a specification or product which has been formally reviewed and agreed upon and that will serve as the basis for further development or maintenance.

[34] Library software is a set of programs which organizes and maintains control files of program source-language. Its automated functions include the retention and identification of prior program versions and limited edits over program statement format and content.

emergency software changes, or create automated approval mechanisms which can be customized to the change management process. However, use of a SCM product would ensure versioning control for systems which are constantly undergoing modifications. Also, the implementation of a product such as ENDEVOR would not interfere with librarian management systems already in use by several applications. If application systems installed ENDEVOR, system managers could "rollback" to a prior version of the system with assurance that all system components would be synchronized to perform properly.

Most application systems reviewed lacked the ability to "rollback" to the prior functional configuration of their application software. Under the current circumstances, system managers do not have sufficient tools to ensure adequate control over the software configuration. Responsible system management may not be able to quickly and efficiently return to a prior version of the application, if a production failure occurs. Similarly, unrecorded "emergency" changes could have affected numerous other components of the application system, and management would not be able to determine the correct configuration of the system.

Without library management or SCM tools, system managers would find it extremely difficult to identify which version of each of the numerous software components comprised a particular operable configuration. Considering the complexity of many of EPA's national systems, it could be costly and complicated to accurately identify and reinstate a prior software configuration. Instead, system managers would most likely be forced to live with the current faulty version of the system and correct the errors or deficiencies until the application was functionally correct.

In addition, constant revisions to a baseline configuration can have significant repercussions, since even a single modification will invariably impact numerous components of the baseline configuration and, in some cases, add new components to the configuration. Many of the application systems reviewed implemented thousands of modifications during a single calendar year. Therefore, manually performed impact analysis for proposed software changes may be insufficient to identify and evaluate the effect of each modification on other components of the baseline configuration. Without adequate impact analysis, implementation could produce unforeseen problems due to component interactions which were overlooked during software testing. Inadequate or incomplete impact analysis could lead to production failure in extreme cases.

BETTER FEDERAL CRITERIA AND EPA POLICIES, PROCEDURES, AND OVERALL MANAGEMENT PRACTICES WOULD HELP IMPROVE SOFTWARE CHANGE CONTROLS

Management Needs To View And Recognize The Significance Of System Operations And Maintenance Cost Information

As previously discussed in Chapter 3 of this report, EPA has not implemented a cost accumulation process for major information systems. In addition, refer to the "Management Did Not Recognize The Significance Of System Operations And Maintenance Cost Information" section in Chapter 2 for a detailed explanation of how this factor influences the adequacy of software change controls.

System Managers Overlook The Relevance Of Software Changes Which Consumed Fewer Budgetary Resources

System managers did not place sufficient importance on software modifications which required limited program office resources. The collective impact of successive minor changes was assigned limited significance, as compared to those proposed software modifications which would consume more budgeted funds. In fact, for several application systems, system managers chose to classify software changes solely on the level of effort involved with the maintenance activity. Due to this viewpoint, system management classified requests for change in a manner which would not facilitate overall analysis of change control processes. Additionally, in some cases, minor changes were not reviewed, controlled, or tracked in an effective or productive manner.

EPA's Policies And Procedures[35] Do Not Provide Adequate Direction For The Software Change Control Process

EPA's Operation and Maintenance Guidance defines a configuration management process which includes primarily quality assurance activities. Although testing standards and procedures are identified as important in the Agency's guidelines, the significance and extent of user acceptance testing, V&V testing, the detailed contents of test plans, and the analysis of test results are not adequately addressed. Likewise, the guidance regarding source code standards does not establish rules or measurement criteria to specify "how well" code must be written, organized, or formatted. Without such guidelines, it would be difficult to assess the maintainability of the application.

---

[35] EPA Directive 2100, Information Resources Management Policy Manual, Chapter 17 (August 1994)
EPA System Design and Development Guidance (June 1989)
EPA Operations and Maintenance Manual (April 1990)

Chapter 17 of EPA Directive 2100, issued in August 1994, identifies relevant Federal and Agency guidance which it states "should" be followed with regard to system life cycle management. The policy does not stipulate that the provisions of these FIPS publications *must* be followed. Neither does this policy provide additional guidance to managers regarding how these guidelines could best be implemented within their application systems.

Since many key areas are not clearly addressed in existing Agency policies, several program offices issued a number of supplemental policies and guidelines to govern their software maintenance activities. In many respects, these supplemental policies filled the gaps not specifically addressed in the Agency guidelines and, thereby, strengthened controls over software maintenance for those programs. However, not all application system managers had the resources or inclination to establish additional procedures and, therefore, many application systems rely solely on Agency guidance to provide the framework for their change control practices. Several of the prominent differences with regard to software maintenance practices are addressed in Appendix V.

Neither EPA Directive 2100 nor the OIRM O&M manual address the following key management issues which would establish controls over software maintenance:

- software configuration management and version control techniques to manage the maintenance process within an evolving and dynamic application system;
- identification of benefits of software configuration management tools or support the use of an automated tool to ensure an adequate audit trail of system modifications;
- descriptions of how change request and problem reporting data can be used to evaluate the adequacy of current maintenance practices, identify questionable trends in software maintenance, or evaluate application stability;
- identification of what types of maintenance data provide the most reliable and useful metrics information, how data should be measured, or how management can interpret measurements to improve their control of application software maintenance processing;
- definition of the quality assurance functions to manage the maintenance process;
- identification of the types of testing which should be mandatory for evaluating the anticipated performance of a software change; and
- specifics on the formulation of SVVPs and independent V&V testing for software maintenance activities, whether or not the initial application development products were subjected to V&V testing under existing Federal standards.

FIPS Publication 106 Does Not Define Software Maintenance Management Processes

FIPS Publication 106 recognizes the importance of management in the software maintenance process, stating that management is clearly one of the most important factors in improving the software maintenance process. It states that management must examine how the software is maintained, exercise control over the process, and ensure that effective software maintenance techniques and tools are employed. In addition, software maintenance managers are responsible for making decisions regarding the performance of software maintenance, assigning priorities to the requested work, estimating the level of effort for a task, tracking the progress of the work, and assuring adherence to system standards in all phases of maintenance.

However, FIPS 106 -- which is the only Federal guidance on software maintenance -- assumes that all managers will know how to fulfill these responsibilities. It does not provide guidance about the effective techniques and tools which managers must employ. It does not define techniques and tools which help control the software maintenance process, or which aid in estimating the level of effort for a task. Agencies are left to exercise these technical responsibilities without adequate guidance. This is an issue which we intend to address in our governmentwide report to the Federal oversight agencies (e.g., OMB, National Institute of Standards and Technology, GSA, etc.). Nevertheless, prudent business practices would still necessitate individual Federal agencies establishing their own guidance in the absence of Federal guidance.

System Managers Rely Heavily On Adequate Performance Of Supporting Contractors

Due to the absence of qualified full-time employees, most EPA application systems relied on contractor personnel to perform software maintenance activities. Our review disclosed that EPA management involvement focused on the initial review and approval of proposed changes, and chose to rely on contractor personnel to perform and oversee those reviews and controls which were built into the final stages of software modification.

The assigned program office review board, or similarly responsible management personnel, had little or no interaction with the modification process once the software change had been sent to the contractor for work. Contractor personnel performed the actual design, coding, and testing of software changes for most application systems reviewed. Although independent peer reviews and unit tests were often part of the contractor's procedures, in many cases the originating program office did not participate in an oversight capacity. Based on our discussions with system managers, we concluded that the extent and frequency of program office interface

during the final stages of change control processing was often minimal and, in some cases, limited to administratively routing the modified code to NDPD for implementation.

RECOMMENDATIONS

We recommend that the Assistant Administrator for Administration and Resources Management, in his role as the Designated Senior Official for IRM and, when appropriate, in conjunction with the Executive Steering Committee for IRM:

4-1. As a subset of Recommendation 2-1 (page 27), define Agency-wide measurable performance indicators which will enable management to:

- evaluate the efficiency and effectiveness of the change control process;
- assess overall stability of the application system;
- assist in allocating budgetary resources; and
- identify software maintenance trends and highlight instances of program rework or excessive corrective modifications.

4-2. Initiate actions to:

- use the software maintenance practices and policies of the SDC and revise them, where appropriate, to ensure that the individual controls and reviews outlined in this report are sufficiently and actively addressed;
- utilize the SDC to promote the use of the best practices in software maintenance activities, within the framework required under Chapter 17 of EPA Directive 2100, throughout the Agency; and
- emphasize the need for and importance of controlled software maintenance practices, through IRM Forums and other meetings regarding EPA's information system activities.

4-3. Modify existing Agency guidance, based on the performance indicators defined in our first recommendation, for managing the software maintenance process and products throughout the Agency. Require formal procedures for automated application systems to include, at a minimum:

a. That each application system establish a standardized form for initiating all requests for software changes, regardless of anticipated level of effort. The form should minimally include: (1) requestor name; (2) date; (3) priority; (4) problem description/justification; (5) type of change; (6) management approval; and (7) completion date.

b. A requirement that "Major Agency" application systems, which experience high availability requirements, develop and maintain a comprehensive and cohesive change tracking system which will track *all* software changes made to the application system, regardless of the type of proposed change or its anticipated level of effort. In addition to the data stipulated in paragraph 4-3.a. above, the tracking system should require data, such as: change request number, affected programs/modules, and comments field for referencing associated change requests.

c. A classification system for change requests which delineates the types of changes being made to the application system based on the nature of change (e.g., adaptive, corrective, perfective). Level of effort information, if desired, should be maintained separately.

d. A centralized review point, within each system or major subsystem, if applicable, for *all* software change requests, regardless of level of effort.

e. Specific thresholds for implementation of independent V&V testing which clearly define the level of effort or other criteria which will be used to determine which software changes are subject to such tests.

4-4. Modify the Operations and Maintenance Guidance and/or Chapter 17 of the IRM Policy Manual to include a requirement for test results of major application software modifications to be reviewed by either: (1) a designated panel of technically-knowledgeable reviewers; or (2) the steering committee which initially reviewed and approved the software change. At a minimum, the appointed reviewers should:

- review the results obtained from testing;
- compare test results with the initial request for change, detailed specifications related to the change, and the applicable test plan; and
- compare modified source code with latest production version of code to ensure that no additional unapproved changes were introduced by programmers during the coding process.

4-5. Revise Chapter 17, Section 8, of the IRM Policy Manual to state "Other relevant Federal and Agency guidance documents which must be followed are noted below:" In addition, Revise Chapter 17, Section 8, of the IRM Policy Manual to include FIPS Publication 132 as one of the referenced Federal guidance documents. The revised policy should stipulate thresholds for implementation of independent V&V testing and clearly define the level of effort or other criteria which will be used to determine which software changes are subject to testing.

4-6. Modify the Operations and Maintenance Guidance and/or Chapter 17 of the IRM Policy Manual to include a requirement for acceptance testing by the user community. User acceptance testing should take place prior to the implementation of a software modification and should be of sufficient duration as to adequately examine and evaluate application functionality. This requirement could reasonably be limited to those software changes which:

- represent a new program or module within the application;
- represent a major system enhancement to the application;
- represent a level of effort which is technically considered by management as a "development" project, rather than a routine or minor maintenance action item; or
- is comprised of a group of software changes which collectively represent a considerable change to the application's performance.

4-7. Through IRM Forums and/or other meetings with the IRM community, promote the benefits of periodic reviews of historical change control data as a valuable management tool. Illustrate to system managers how pending modifications and historical data can be used to detect and evaluate trends regarding the nature and frequency of processed software changes. Emphasize the usefulness of historical data to discern inadequacies in review and test procedures or inadequacies in the contractor's performance. Encourage system managers to make use of available historical data to judge the stability of their application systems, as well as the adequacy of their current change control practices.

4-8. Make an SCM tool, such as the ENDEVOR product already implemented in IFMS, available to EPA's program offices, and encourage system officials to implement its use on application systems which were classified as "Major Agency" systems due to their high availability requirements.

In preparation for adoption of this recommendation, IRM management should establish a definite implementation schedule for those "Major Agency" applications under its control. The schedule should be aimed at enforcing SCM implementation within a reasonably short period of time. Desirable features of a good SCM product are outlined in Appendix VII of this report.

AGENCY COMMENTS AND OIG EVALUATION

In their March 17, 1995, response to our draft report, OARM officials agreed with ten of our eleven *revised* recommendations and disagreed with one recommendation. OARM has initiated action on one recommendation, while NDPD is performing action to bring its customers into the already implemented change management system. Completion of the eight remaining recommendations are dependant on the implementation schedules of recommendations 2-1 and 4-1. OARM's proposed actions, however, do not fully meet the intent of four of our recommendations.

Although OARM agreed with recommendation 4-1, their response did not clearly indicate that specific measurable performance indicators would be defined or incorporated in Agency guidance as a means of managing software maintenance. OARM's response focused on performance of source code reviews and software testing, rather than recognizing that metrics would promote informative trend analysis or assist management in allocating budgetary resources for software maintenance activities.

In addition, OARM agreed with recommendation 4-3.b., but the corrective action did not specifically state that the revised O&M

Manual would require "Major Agency" application systems to develop and maintain a comprehensive and cohesive tracking system to track all software modifications. Rather, OARM's response only indicated that the revised manual would provide information about this type of tracking system. Similarly, it was not immediately clear from the response to recommendation 4-3.c., whether the revised document would contain a requirement for program offices to adopt industry's standard classification system for software changes. Instead, OARM's response emphasized that the currently suggested change request form already prompts the submitter for a "Category." The intent of recommendation 4-3.c. was to incorporate a more meaningful classification system in the tracking process by requiring a requester to define the nature of the software change in consistent terms (e.g., adaptive, corrective, perfective), rather than other less descriptive categories.

OARM also agreed with recommendation 4-6., but stated that the O&M Manual already contained requirements for "user" acceptance testing. Although one appendix form (EEI-7) made reference to "User Acceptance," the current manual text contains neither guidelines nor a requirement for user acceptance testing on software modifications.

OARM disagreed with our draft report recommendation to revise Chapter 17, Section 8, of the IRM Policy Manual to make Federal and Agency guidance documents mandatory and to include FIPS Publication 132 as one of the referenced documents. The OARM response stated that the policy was only recently issued and that it was intended to provide high level statements of principle and direction rather than procedural instructions. Considering NIST's recent announcement of their intention to rescind a number of FIPS publications, we have withdrawn recommendation 4-5 with respect to the IRM Policy Manual. However, the planned revisions of the O&M Manual should adequately address the topic of independent V&V testing and stipulate thresholds for implementation which would clearly define the level of effort and other criteria used to determine which software changes are subject to V&V testing.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX I

AGENCY COMMENTS

OA-0167

UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
WASHINGTON, D.C. 20460

MAR 17 1995

OFFICE OF
ADMINISTRATION
AND RESOURCES
MANAGEMENT

MEMORANDUM

SUBJECT: Response to Draft Report of Audit on
Management of Application Software Maintenance in EPA
(E1NMF3-15-0072-)

FROM: Jonathan Z. Cannon
Assistant Administrator

TO: Kenneth A. Konz,
Acting Deputy Inspector General

Thank you for the opportunity to respond to the above-referenced draft audit. My staff have enjoyed numerous and informative discussions with your staff on the topics raised in the draft audit. While we may not agree on all issues, the interactions and dialogue have been productive.

One indication of this, from our perspective, was your office's stated desire to revise certain recommendations presented in your draft report. In adherence to your staff's request, we are responding to the *revised versions* of your office's recommendations, rather than to the original versions presented in the February 13th draft report.

I also appreciate that the audit took a balanced view in pointing out a number of instances where OARM implemented or promoted good software maintenance practices.

There remain, however, a number of broad issues of concern in the draft report:

1) The large number of policy-related recommendations, and their focus on improving internal Agency processes, seems contrary to the mission-accomplishment orientation of the National Performance Review and the Government Performance and Results Act.

APPENDIX I

2) Because of the report's focus on efficiency of internal Agency processes, rather than effectiveness of outcomes, it is uncertain that carrying out the recommendations would really improve mission accomplishment in a cost-effective manner.

3) The many recommendations calling for additional internal Agency policy and procedures are also in direct conflict with the Agency's TRIM initiative, through which we are meeting an Executive Order requirement to reduce, rather than expand, our internal mandates.

4) The factual basis for Chapter 2 is very weak, and the conclusions drawn from this weak factual basis form the core of the draft audit.

5) The report overemphasizes the need for mandatory Agency implementation of Federal guidance that is actually optional.

Overall, we agree that EPA should place more attention and discipline on its software maintenance activities. Within the context of all information resources management (IRM) concerns, however, making improvements in software maintenance is probably not as important as, for example, correct upfront planning for major new investments. While there are benefits to following "best practices," the payoffs for applying them in software maintenance are not as great as in some other areas of IRM.

Please find attached two items that comprise our full response to the draft audit. The first is a summary matrix which provides an overview of our responses to the revised recommendations. The second attachment is a more detailed set of responses to particular sections of the draft audit report. This detailed response follows the same chapter order as the draft audit.

Should you or your staff have any questions or need additional information regarding this response, please contact Alvin M. Pesachowitz, Director of the Office of Information Resources Management, on (202) 260-4465.

Attachments

cc: Members, Executive Steering Committee for IRM
Senior IRM Officials
Kathryn S. Schmoll, Comptroller

APPENDIX I

Summary Matrix of Responses to Revised Recommendations from the Audit on Management of Application Software Maintenance in EPA (EINMF3 - 15-0072-)

Revised Recommendation	Response	Discussion
We recommend that the Assistant Administrator for Administration and Resources Management, in his capacity as Designated Senior Official for IRM and, when appropriate, in cooperation with the Executive Steering Committee for IRM: 2-1. Identify the measurements needed to support Agency-wide management of software maintenance. The measurements should include: o resource tracking - quantification in dollar amounts of internal and external resources used as the input for production of a service or product, (i.e., estimating and tracking resources use, tasks, deliverables, and milestones); o work product tracking - the number of units of the product or service provided to the customer; the level of service or product quality, both in terms of customer satisfaction (external quality) and of work performed to provide the service (internal process quality)(e.g., tracking and control of source code, test case, and document versions and changes); and measure of size and complexity (e.g., Halstead code measurements, function points, cyclomatic complexity, Kiviat diagrams); and o problem tracking - tracking and control of problems, defects, and open issues.	Agree	OARM agrees with the need to identify measurements for supporting Agency-wide management of software maintenance and will determine the most cost-effective way to accomplish this. As recommended, we will consider measurements such as: Cost measurements for resource tracking. Quality measurements for both external users and internal processes, and Problem tracking measures. Milestones: OIRM is currently discussing measurement options with other agencies in an effort to comply with GPRA requirements. Milestones will be tailored to conform with GPRA deadlines.

1

2-2. Based on the metrics defined in our first recommendation, require that OIRM modify its Operations and Maintenance Guidance to establish processes to: o define appropriate project status reporting and quality assurance tasks for software maintenance activities; o manage the software life cycle, maintenance process, and products within Agency programs in compliance with EPA Directive 2100; and o implement FIPS Pub 106 guidelines to examine how the software is maintained, exercise control over the process, and ensure the effective software maintenance techniques and tools are employed.	Agree	We agree to update the Operations and Maintenance Manual. It is important to note that the current document does reinforce the principles for managing software presented in FIPS PUB 106. We will continue to reinforce and strengthen those points in the revised document. Timeframe: Dependent on implementation schedule of recommendation 2-1.
2-3. Require that OIRM modify existing policies, procedures, and standards...		We understand that this recommendation will be removed entirely by OIG, except for bullet five, which is now bullet one in Rec. 2-2.
2-4. Evaluate commercial defect tracking software, and determine whether any available package should be included as an Agency standard for problem tracking and defect removal in Agency roadmap planning and hardware/software standards documents.	Agree	OIRM is evaluating commercially available problem management systems for its own and broader Agency use, as part of the Distributed Systems Management (DSM) program. We may propose a standard to the Agency, based on the results of the software evaluation. NDPD is considering creating a problem management service under the Working Capital Fund.

APPENDIX I

2-5.	Based on the metrics defined in our first recommendation, require OIRM to update EPA Directive 2115 to make the ADP Review a comprehensive review of the system and its support for Agency goals and missions. Include review requirements that would: • require quantitative measures of performance, and a user satisfaction survey of the system; • require that the program office demonstrate the extent to which the system supports Agency and program office strategic objectives; • require a periodic review of the effectiveness, accuracy, need, and economic justification for continued operation for each information system; and • ensure that operational systems use an optimum, least-cost mix of resources to meet user functional, data, and other systems' compatibility requirements.	Agree	We agree this document is in need of an update and OIRM intends to issue a revised document. EPA is currently revising the IRM Review Program to meet the requirements of the Paperwork Reduction Act (PRA) more comprehensively. Part of the revised program's infrastructure will consist of integrating IRM review activities into program review activities and developing evaluative tools to assist in the review of the IRM activities. Timeframe: Dependent on implementation schedule of recommendation 2-1.
3-1	We recommend that the Assistant Administrator for OARM, in his capacity as Designated Senior Official for IRM and, when appropriate, in cooperation with the Executive Steering Committee for IRM: Ensure that the Chief Financial Officer (CFO) project related to project cost accounting provides the ability to accumulate system level costs. Continue to coordinate these efforts with the working capital fund initiative.	Partially Agree	We disagree with the need for this recommendation. OARM already plans to install the Project Cost Accounting System (PCAS) module during July of 1995 to support the Working Capital Fund. PCAS will be available for consideration in FY96 for meeting other Agency cost accounting requirements. During FY96, further study will be conducted of IFMS account code structure along with the evaluation of PCAS usage. Corrective Action and (Target Date) - Complete evaluation of needs for additional systems' cost tracking. 06/30/96 - Implement policies, procedures and requirements for any additional tracking of system costs through the IFMS account code structure or PCAS. 10/31/96

3

APPENDIX I

3-2.	Incorporate requirements for the accumulation and capitalization of all new development costs and major enhancements which meet the $5,000 capitalization threshold into the project cost accounting project. Establish an interim process to accumulate major system costs to be capitalized. These costs should be incorporated into the financial statements as appropriate.	Partially Agree	We have not determined that the project accounting system module would be an appropriate tool to meet our overall objective for capitalizing software. However, we agree with the recommendation as it relates to the capitalization of software costs and their recognition in the Agency's financial statements. As part of OARM's plan to improve the Agency's accounting policies and procedures, we will address requirements for capitalizing system costs. Corrective Action and (Target Date) - Complete analysis of policy and procedural changes. (7/31/95) - Issue draft policy revisions. (10/31/95) - Issue interim revised policy. (12/31/95) - Issue final policy directive. (9/30/96)
3-3.	Change the OMB Circular A-11 40B report on financial system obligations to reflect system costs for telecommunications and timeshare, include CPS and EPAYS as financial systems in future reports.	Agree	We agree with the recommendation, providing that the Office of Inspector General (OIG) delete "include CPS and EPAYS as financial systems in future reports" from the recommendation. EPA's 40B report to OMB for FY 1995, containing information on FY 1994 through FY 1996, includes the Contracts Payment System and the Payroll system as financial systems. Currently, EPA already reports telecommunications and timeshare costs, in aggregate, to OMB in Exhibit 43 under Circular A-11. For the next OMB report, we propose using data accumulated at the National Computer Center (NCC) in support of the Working Capital Fund. Corrective Action and (Target Date) - Include Timeshare and Telecommunications Costs within FY 1995 Exhibit 40B on Financial Systems, using cost data provided by NCC. (10/15/95)

4

APPENDIX I

3-4.	Require the completion of a feasibility study for replacing or modifying the timeshare management system to provide accurate levels of workload accumulation for individual major systems for both NDPD capacity planning and system managers.	Agree	NDPD is in the process of making changes to the TSSMS which include a required field for the National ADP System Code. The purpose of this field is to enhance the Agency's ability to capture system utilization and the associated costs. Corrective Action and (Target Date) Feasibility Study (Completed) Design and Develop Enhancements (Completed) Test and Review Enhancements (4/95) Enhanced TSSMS Software Becomes Operational (4/95)
3-5.	Provide the capability within the system access and accounting systems to capture and accumulate resource utilization costs for the different life cycle phases of each information system (e.g., maintenance programming, operations, user access, etc.).	Partially agree	The changes planned for TSSMS include a required field for the National ADP System Code. The TSSMS system captures computer related charges for each ADP Account and specific utilization charges for each authorized user of that account. TSSMS cannot determine the purpose of the system utilization, (i.e., whether the account was being used to perform maintenance or for basic access). System owners have the capability of establishing separate account codes which can provide more detailed information on system maintenance costs.
3-6	Require the establishment of controls or edits in FIMAS to force FIMAS codes to be linked to the correct users, accounts, and organizations.		We understand that this recommendation will be deleted in its entirety.
3-7	We have been asked to respond to the following potential revision of this recommendation: Establish thresholds to enforce the requirement of cost-benefit analyses for all major changes to application systems, using the criteria for system classification outlined by HCFA Directive 2100, Chapter 17. The cost benefit analyses should provide managers, users, designers, and auditors with adequate cost and benefit information to analyze and evaluate alternative approaches. The cost benefit document should contain a summarization of the criteria used in the evaluation, as well as the estimated costs and benefits.	Partially agree	This will be addressed in the context of our broader work in promoting sound IRM planning for information systems. Benefits for use in cost-benefit analyses during the maintenance phase should be incremental (i.e., not inclusive of the existing benefits) in comparison to the costs of the improvements. It does not make sense to compare total system benefits with incremental changes. Criteria need to be developed to indicate when cost-benefit analyses are needed. The requirement for, and size of system-level cost benefit analyses should be geared to the size and mission-importance of the application.

5

APPENDIX I

We recommend that the Assistant Administrator for the Office of Administration and Resources Management, in his capacity as Designated Senior Official for IRM and, when appropriate, in cooperation with the Executive Steering Committee for IRM:		
4-1. As a subset of Recommendation 2-1, define Agency-wide measurable performance indicators which will enable management to: • evaluate the efficiency and effectiveness of the change control process; • assess overall stability of the application system; • assist in allocating budgetary resources; and • identify software maintenance trends and highlight instances of program rework or excessive corrective modifications.	Agree	These points can be addressed in the revision of the Operations and Maintenance Manual or in associated practice papers. Software testing, if done properly, determines whether the software performs as expected by the user(s). Source code reviews would therefore check for adherence to coding standards. Timeframe: The performance indicators would be projected for release by the end of FY96.

APPENDIX I

4-2.	Initiate actions to: • use the software maintenance practices and policies of the System Development Center (SDC) and revise them, where appropriate, to ensure that the individual controls and reviews outlined in this report are sufficiently and actively addressed; • utilize the SDC to promote the use of the best practices in software maintenance activities, within the framework required under Chapter 17 of EPA Directive 2100, throughout the Agency; and • emphasize the need for and importance of controlled software maintenance practices, through IRM Forums and other meetings regarding EPA's information system activities.	Agree	The software maintenance practices and policies at the SDC will be revised as needed to reflect the requirements in the revised O & M Manual. We agree that the "state of the practice" software maintenance activities at the SDC should be promoted throughout the Agency and will ensure that the SDC maintenance practices and policies are distributed and briefed to the IRM community. We also agree that the need for, and importance of, controlled software maintenance practices should be communicated in appropriate meetings of the IRM community. The SDC already presents briefings and brown bag seminars on various topics such as the SDC Product Development Process and the SDC Product Assurance Policy, and will include software maintenance practices as a topic. Delivery Order Project Officers are required to attend the SDC Product Development Process briefing and are routinely invited to attend the brown bag seminars. The next scheduled seminar on March 23, 1995 will address software metrics issues. The forthcoming OIRM reorganization will consolidate application systems expertise into an Enterprise Systems group, which will lead these varied actions. Timeframe: Dependent upon final determination of revisions to be made to the O & M Manual.
4-3.	Modify existing Agency guidance, based on the performance indicators defined in our first recommendation, for managing the software maintenance process and products throughout the Agency. Require formal procedures for automated application systems to include, at a minimum: a. That each application system establish a standardized form for initiating all requests for software changes, regardless of anticipated level of effort. The form should minimally include: (1) requestor name; (2) date; (3) priority; (4) problem description/justification; (5) type of change; (6) management approval; and (7) completion date.	Agree	We agree with recommendation 4-3 overall and will appropriately modify existing Agency guidance. We disagree with the aspects of point (a.) that are micromanaging and overly prescriptive. Per our response to recommendation 2-2, we have agreed to update the Operations and Maintenance Manual and will continue to include the requirement that there be a systematic approach to change requests. It should be noted, that in the current version of the O&M Manual, there is a requirement to document change requests, and Exhibit 3-2 on page 3-8 provides a model for the information which should be included in a change request form. We will include as additional guidance those items recommended in the audit which were not in the original model change request form.

APPENDIX I

b. A requirement that "Major Agency application systems, which experience high availability requirements, develop and maintain a comprehensive and cohesive change tracking system which will track all software changes made to the application system, regardless of the type of proposed change or its anticipated level of effort. In addition to the data stipulated in paragraph 3.a above, the tracking system should require data, such as: change request number, affected programs/modules, and comments field for referencing associated change requests.	Agree	We will provide information about this type of tracking system in the revised document.
c. A classification system for change requests which delineates the types of changes being made to the application system based on the nature of change (e.g., adaptive, corrective, perfective). Level of effort information, if desired, should be maintained separately.	Agree	This recommendation for classifying the change request has already been agreed to in our response to Section a. The change request form requires a classification of the type of change requested. This information is relevant to include in a tracking system.
d. A centralized review point, within each system or major subsystem, if applicable, for all software change requests, regardless of level of effort.	Agree	This requirement can be addressed in the revised guidance document. Overall timeframe for 4-3, a ,b, c, and d are dependent on implementation schedule of recommendations 2-1 and 4-1.

APPENDIX I

4-4.	Modify the Operations and Maintenance Manual and/or Chapter 17 of the IRM Policy Manual to include a requirement for test results of major application software modifications to be reviewed by either: (1) a designated panel of technically-knowledgeable reviewers; or (2) the steering committee which initially reviewed and approved the software change. At a minimum, the appointed reviewers should: • review the results obtained from testing; • compare test results with the initial request for change, detailed specifications related to the change, and the applicable test plan; and • compare modified source code with latest production version of code to ensure that no additional unapproved changes were introduced by programmers during the coding process.	Agree	In the update to the Operations and Maintenance Manual, we will reinforce the responsibilities of the reviewing parties. It should be noted that the current version of the O&M Manual describes the responsibilities of the Configuration Control Board in Exhibit 3-1 on page 3-4. We will, however, make sure the role of the reviewers, be they a formal Configuration Management Board or a comparably experienced group, is communicated clearly in the revised document. Timeframe: Dependent on implementation schedule of recommendation 2-1.

APPENDIX I

4-5.	Revise Chapter 17, Section 8, of the IRM Policy Manual to state "Other relevant Federal and Agency guidance documents which must be followed are noted below:" In addition, Revise Chapter 17, Section 8, of the IRM Policy Manual to include FIPS Publication 132 as one of the referenced Federal guidance documents. The revised policy should stipulate thresholds for implementation of independent V&V testing and clearly define the level of effort or other criteria which will be used to determine which software changes are subject to testing.	Disagree	We have committed to revising the O&M Manual and can cite relevant FIPS Pubs. However, we do not agree with the recommendation to revise the policy for the following reasons: • Introductory language in FIPS Pubs 106 specifically states that use of that Guideline is encouraged but not mandatory. It would be inappropriate for EPA be more prescriptive than what NIST presents in their direction to Agencies. Considering NIST's recent announcement of their intention to rescind a number of FIPS Pubs, it is better to have a statement of policy of commitment to the FIPS Pubs in a general sense rather than citing individual ones which may be rescinded. Please note that the Agency's Software Management Policy (Chapter 4 of Directive 2100) provides this global commitment. "EPA program officials will adhere to FIPS and guidelines as published or adapted for the Agency in developing, documenting, maintaining and using software applications." The policy is intended to provide high level statements of principle and direction rather than procedural instructions. For that reason, we will address more detailed procedural information in the revised Guidance document. • It was just recently enacted, receiving the concurrence of all organizations, including the Office of Inspector General. Considering, how long it took to get the initial policy issued, it does not seem cost-effective or prudent to reopen the green border process and invite additional changes, some of which may in fact weaken the existing policy.

10

APPENDIX I

4-6.	Modify the Operations and Maintenance Manual and/or Chapter 17 of the IRM Policy Manual to include a requirement for acceptance testing by the user community. User acceptance testing should take place prior to the implementation of a software modification and should be of sufficient duration as to adequately examine and evaluate application functionality. This requirement could reasonably be limited to those software changes which: o represent a new program or module within the application; o represent a major system enhancement to the application; o represent a level of effort which is technically considered by management as a "development" project, rather than a routine or minor maintenance action item; or o is comprised of a group of software changes which collectively represent a considerable change to the application's performance.	Agree	The current Operations and Maintenance Manual contains requirements for acceptance testing but we will reinforce and strengthen this point in the revised document. Timeframe: Dependent on implementation of schedule of recommendation 2-1.

11

4-7.	Through IRM Forums and/or other meetings with the IRM community, promote the benefits of periodic reviews of historical change control data as a valuable management tool. Illustrate to system managers how pending modifications and historical data can be used to detect and evaluate trends regarding the nature and frequency of processed software changes. Emphasize the usefulness of historical data to discern inadequacies in review and test procedures or inadequacies in the contractor's performance. Encourage system managers to make use of available historical data to judge the stability of their application systems, as well as the adequacy of their current change control practices.	Agree	NDPD has already initiated action to bring its customers into the already implemented change management process. On March 1, 1995 at the monthly SIRMO meeting the long range change management function was announced. The SIRMOs were informed that they would soon be asked to identify a system contact within their organization to communicate with NDPD about major/critical system changes. NDPD would then coordinate the customers' changes with planned NDPD changes. We intend to further discuss this at one of the IRM Branch Chiefs meetings, publicize it in the monthly newsletter, the CONNECTION and discuss it at the biannual Outreach teleconferences with regional office and program office personnel. OIRM's forthcoming reorganization will unite application systems expertise into one Enterprise Systems group which will promote the benefits of these types of periodic reviews. This message will be reinforced by the Policy and Oversight group through its emphasis on varied types of IRM reviews.
4-8.	Make a software configuration management (SCM) tool, such as the ENDEVOR product already implemented in IFMS, available to EPA's program offices, and encourage system officials to implement its use on application systems which were classified as "major agency" systems due to their high availability requirements. In preparation for adoption of this recommendation, IRM management should establish a definite implementation schedule for those "major agency" applications under its control. The schedule should be aimed at enforcing SCM implementation within a reasonably short period of time. Desirable features of a good SCM product are outlined in Appendix VII of this report.	Agree	NDPD is announcing the availability of ENDEVOR to the entire NCC user community through a User Memo. The memo should be published within three weeks and the ENDEVOR SCM product will be available on June 1, 1995. As was noted earlier ASD has already implemented IFMS under control of ENDEVOR. Work has begun on ENDEVOR for EPAYS with an expected implementation target date before the end of FY95. ASD will then focus on GICS with implementation expected in FY96.

APPENDIX I

Detailed Response to Draft Report of Audit on Management of Application Software Maintenance in EPA (E1NMF3-15-0072-)

OIG representatives conveyed to OARM staff on March 3, 1995, a number of specific, planned changes to the recommendations presented in the February 13th version of the draft audit report on Management of Application Software Maintenance in EPA.

OIG representatives asked OARM to respond to these planned, revised recommendations rather than to the original recommendations in the February 13th version of the draft audit report. To streamline the response and avoid the need to revisit the many discussions that led to the planned revisions, OARM representatives agreed to this approach.

Throughout the following detailed response, we have indicated these promised changes in the final wording of the audit recommendations by using **bold** text to show language to be added. There are also instances where OIG has indicated they intend to remove text or bullet points from individual recommendations. In these cases, we made the deletions in the wording of the recommendations, and are responding only to the text and bullets that we have been told will remain in the final report of audit.

OARM managers and staff have identified significant factual errors and interpretation errors in the text of the draft audit. The following pages provide specific comments pertaining to portions of the draft audit report that we believe require revision by the OIG. We cite, by page and paragraph, the parts of the report to which we take exception, and we provide detailed comments to support our view.

Explanations of these factual errors are included in the pages that follow, along with our responses to the revised recommendations. The detailed response is arranged in the same chapter order as the draft report of audit, and closes with comments concerning the appendices.

Executive Summary

Page ii, first paragraph -- This paragraph states, "However, we found that EPA managers do not really know how much this function costs, so effective decision-making is greatly hindered about things like what software changes to make, when to make them, and whether to replace old systems with new ones."

As written, this statement is incorrect. For IFMS, "EPA managers" know exactly how much Operation and Maintenance (O&M) costs are; we prepare budgets and monitor Spending Plans that distinguish between O&M and Development and that separately identify work termed adaptive and perfective maintenance in this report; and we prepare a Decision Paper for the Chief Financial Officer based on a cost-benefit study that included all cost components referred to in the draft audit report.

Page iv, second paragraph -- This paragraph states, "In most cases, management involvement was limited to the initial stages of review and approval, with EPA management relinquishing control over the final test and review stages to contractor personnel. Overall, software changes were not consistently or effectively controlled by EPA management."

This assertion is decidedly not true for IFMS. Elsewhere we describe extensive user acceptance testing as well as senior management approval for substantial changes to the system. These procedures have been in effect for several years.

Page iv, third paragraph -- It is an overstatement to say that continuity of system operations "cannot be guaranteed" because of the perceived problems in software change control and configuration management. The audit raises no instances of systems' continuity of operations actually being affected.

Page v, last paragraph -- Although we see the value in promoting a more consistent and structured approach to managing application software maintenance across the Agency, we do not agree that establishing "mandatory practices" is the best way to achieve this end. Overly prescriptive, mandatory practices can eliminate the flexibility needed to appropriately tailor good software maintenance principles to the needs of a particular application system.

Chapter 1

Page 3, paragraphs one to three -- These paragraphs state that, "In the spirit of the FMFIA process ... specifically, we concentrated on internal control improvements to offset potential adverse effects .... It is possible that some of the effects identified could be mitigated through the use of compensating management controls .... However, this claim could not be made for all of the application systems reviewed and, therefore, the effects identified depict real and potentially damaging situations which cannot be overlooked."

We appreciate the spirit of FMFIA, but in point of fact detections of real, damaging situations were out of the scope of the

OIG review. Nor did the OIG investigation disclose any examples of the risks alluded to. We believe the final report should more explicitly recognize this scope limitation. The draft report recognizes this point in the peripheral statement, "No other issues came to our attention which we believed were significant enough to warrant expanding the scope of this audit."

Chapter 2

OARM does not agree with most points raised in the sections starting with page 5 and carrying through page 13. These sections are derived from raw data found in MICS, a copy of which was made available to the OIG for their own information gathering. Our main concern is that the numbers given in the figures may not reflect what would normally be considered "Total CPU" or "Production Application Abend". The figures reveal some fundamental misunderstandings of the meaning of the MICS data. These misunderstandings may significantly impact the validity of the findings and recommendations throughout the report.

Page 8, second paragraph -- OIG refers to Dr. Bill Hetzel of the Software Practices Research Center, and states that "...tracking systems used by system managers do not meet Dr. Hetzel's definition of a problem tracking system". We believe that problems are tracked, although EPA does not follow Dr. Hetzel's model. We are unaware of any requirement that EPA follow this particular model.

Pages 8 and 9, Figures 1 and 2 -- These figures attempt to make the case that EPA systems exhibit error rates that are higher than expected for mature, production systems. It is essential to document how statistics for "production" systems were derived from statistics representing all jobs. This is critical because MICS provides statistics for all jobs, large or small. Most abends and JCL errors on the mainframe **are not** from programs/JCL that are in production. It is likely that user job abends tracked by MICS were counted among the "production" job failures, and they should not have been.

The number of ABENDS and JCL errors shown in Figure 1 can be misleading. Most of the time, these problems are caused by the users of the system not allowing enough CPU time, not enough lines of print, or enough size for large extraction of data. In these cases, the job fails with an ABEND, but these types of ABENDS are not really system problems. Therefore, these types of failures should not be included in the figure.

Often, new programs are intentionally forced to abend to verify that automated recovery procedures work correctly. It is unclear whether these new development and test failures were also included in

the error rates of production systems. Additionally, it is unclear whether hardware failures were also inappropriately included in the error rates of production systems.

There is potential for misunderstanding the MICS statistics for certain families of related systems. Some IFMS/EPAYS/GICS accounts are shared by other applications, so it is unclear whether error rates of non-production, as well as other, applications are factored into the job failure analysis for individual systems.

It is also necessary to be explicit about what was considered to be a "job" for the purpose of the analysis. Technically, only MICS JOBGROUPs < 199 are batch jobs. Job group 199 is a TSO online user. It is not clear whether users' manual efforts in native TSO/ISPF have been factored in as production failures. These unanswered questions raise significant concern about the validity of the analysis of MICS data, and about the conclusion that these EPA systems exhibit unusually high error rates.

Page 9, end of first paragraph -- This states that, "The graph for IFMS [Figure 1, Job Failures] reflects significant system difficulties..." All of the charts in Chapter 2 of the report are misleading for IFMS and probably for other systems as well. The charts depict job failures, ABENDs, Job Control Language failures, changes, Central Processing Unit time, and CPU usage.

In the case of IFMS, the OIG team used statistics on an IBM billing account, IFMS, which captures usage statistics for the core financial system (also termed 'IFMS') as well as the less mature Management and Accounting Reporting System and interfaces that carry data from mixed systems into IFMS. Furthermore, user reporting and all of our testing are billed to the same account.

A typical reader of the report will not envision the family of systems that are collected under this billing identifier. Those readers, rather, will attribute the statistics solely to the core financial system, which is incorrect and misleading.

The report should more clearly identify the basis of the statistics and describe in much more detail the kinds of activity captured within "MICS records." In particular, the report should discuss whether hardware failure or intentional testing, such as for disaster recovery, could be the source of some of the statistics in this chapter.

Page 10, Figure 3 -- The figure implies that many problems occur, and that few problems are appropriately logged. It is worth noting that there is more than one source of "problem logs". System problems are

gathered in INFO-Management at NCC. Application problems are often logged by IFMS, EPAYS, GICS hotlines.

Not all errors are application production problems. Thus, not all errors should necessarily be logged as application system problems. For example:

- IFMS sends warnings when no data is processed; these may be captured, by MICS, as user abends.
- User TSO sessions may be included in the error count which has auto-recovery. Users may never call with a TSO problem (and TSO is directly the application).
- Errors occur when testing new software.
- Errors occur when testing error recovery procedures.
- Users manually change their own reports and often find their own mistake and never call for help.
- Some features in IFMS online are not installed and not supported, but can abend if attempts are made to access these unsupported features. Access to these features is controlled through security tables managed by FMD. These abends are not usually considered software problems.
- IFMS may abend with a hardware error. Problem management would log this as a hardware problem, not an IFMS Production problem.

Page 11, paragraph one -- This states that, "System managers do not monitor and record software changes corresponding to environmental changes in laws and regulations, system software configuration, and hardware configuration as adaptive maintenance, and do not distinguish between corrective, adaptive and perfective maintenance."

System managers do, in fact, know the origin of the changes that they must install. For several years IFMS has had a Strategy and Master Work Plan, not mentioned anywhere within the report, that effectively separates system activity into the categories identified in the report. Budget formulation and execution for the system employ those categories as well. Sample categories within the Strategy include "Comply with Federal and External Requirements" and "Meet User Requirements."

Page 12, Figure 5 -- This figure argues that most system managers do not have a realistic view of the amount of CPU time used each month by their system, and reports that the IFMS System Manager overestimated IFMS CPU utilization by 100%. It is very likely that the MICS data captured for this chart reports "JOB CPU" and not JOB CPU plus ADABAS CPU. Approximately half of IFMS's total CPU usage is ADABAS processing. This simple error in interpreting the meaning of the MICS data accounts for the apparent discrepancy for IFMS.

Moreover, the chart shows a discrepancy between the actual and reported CPU utilization for CPS. At the time, we did not have NDPD utilization statistics to provide to the OIG staff, so we referred them to NDPD for the statistical information. Since we did not provide them any utilization statistics, it is unclear how OIG had a number to report as CPU hours reported by the system manager.

To adequately interpret technical information such as that found in MICS, it is necessary to clearly define the terms "Production", "Application", and "CPU". It is important to note that:

- Development programs often reference production programs.
- IFMS/GICS/EPAYS accounts are shared by other applications, as well as sharing same TSSMS information.
- CPU means different things on different machines. IFMS uses both the EPA2 and EPAG IBM mainframes, which have different CPUs.

Page 13, Figure 6 -- It is true that there are strong variances in IFMS usage from month to month, and there may be monthly or quarterly differences in resources worth exploring for cost recovery and service level agreements. However, applications which grow faster and use more of the data they retain will have larger CPU spreads that really do not vary from month to month, but rather just grow. IFMS, for example, has two known capacity issues:

1) Fiscal yearend requires many more resources than any other time of year.

2) Many IFMS subsystems start out empty at the beginning of the fiscal year, grow through yearend, and are archived/deleted at yearend. The amount of CPU grows as the amount of data reported on grows and the indexing of larger files is greater.

Capacity is not usually measured by CPU per month. There may be a notable CPU variance between the two months September/October and the month of November. It would help to better understand what factors were accounted for in the CPU spreads. For example, IFMS at yearend works 7 days per week, and at Christmas or Thanksgiving may only work 3 days per week. Some months have 31 days, others 28 days. These factors should be accounted for in any reasonable interpretation.

Much of the concentration on capacity planning is looked at in time windows in a day, and by factoring production and non-production CPU, as well as other resources and factors such as I/O, memory, software licensing, and online user response time.

APPENDIX I

We have additional questions surrounding the criteria used for developing Figure 6:

- CPU is for CICS, TSO, & Batch?
- Was JOBCPU used or JOBCPU plus ADABAS CPU?
- CPU utilization grows over time as amount of data grows. Were smaller months the earlier months in the chart?
- Were the measures of CPU based on account?

Page 15, paragraph one -- This states that, "System managers do not periodically review all software resources to determine and prevent obsolescence of software."

We believe that the Decision Paper, approved by the Chief Financial Officer, updating the feasibility and cost-benefit studies for IFMS should be noted in the final report. Another OIG audit team was instrumental in that activity; a cross-reference to that team's work would thus demonstrate appropriate coordination within the Inspector General's Office.

Page 16, first paragraph -- This mentions that, "CPS managers indicate that a determination of obsolescence would be based on the subjective judgement of the division director based on his experience managing and developing administrative systems."

Although this statement is correct, it was not the only criterion discussed in the System Manager interview. We noted instances where we have totally rewritten major components of the original CPS system in an attempt to prolong the effective life of the application. In addition, budgetary implications, technology considerations, etc., were mentioned as factors that would be considered when determining obsolescence of the CPS application. In fact, the FY1994-FY1999 FMD Five Year Plan indicates an objective to "Downsize the CPS Application to Client/Server Technology", with a technical and cost benefit analysis to be conducted in FY1996. We believe that this statement should be either clarified or removed.

Page 19, second paragraph -- This states that,"Since ABENDS and JCL errors are not reported as problems, any attempt to remove defects or improve the quality of the software would start with incomplete information."

The statement is not true for IFMS. ASD staff supporting the financial system require "Problem-Cause-Solution" reports documenting system problems and recommending solutions. The draft audit report does not mention this compensating tool whatsoever. The reports are useful. We recently, for example, modified program return codes for the IFMS nightly cycle. By doing so, we downgraded return codes for

well-defined, specific, recurring data anomalies so that the operations staff could concentrate on truly significant job failures.

Page 19, last paragraph -- This states, "We believe that each of these systems, as shown in the chart [Table I, System Replacement Factors], exhibit one or more of the characteristics FIPS Pub 106 defines as the factors to consider in weighing a decision to maintain or redesign..."

Our earlier comment about the use of IBM Billing Accounts in place of system usage makes the chart meaningless for the IFMS family of systems and probably for others. The statistics in the chart include more than one large system (i.e., the financial system plus its reporting system, MARS, plus other usage). A similar comment applies to the vulnerability assessment presented in Figure 9 on page 22.

RECOMMENDATIONS:

We recommend that the Assistant Administrator for Administration and Resources Management, **in his capacity as Designated Senior Official for IRM and, when appropriate, in cooperation with the Executive Steering Committee for IRM:**

2-1. Identify the measurements needed to support Agency-wide management of software maintenance. The measurements should include:

- **resource tracking - quantification in dollar amounts of intramural and extramural resources used as the input for production of a service or product, (i.e., estimating and tracking resource use, tasks, deliverables, and milestones);**
- **work product tracking - the number of units of the product or service provided to the customer; the level of service or product quality, both in terms of customer satisfaction (external quality) and of work performed to provide the service (internal process quality)(e.g., tracking and control of source code, test case, and document versions and changes); and measures of size and complexity (e.g., Halstead code measurements, function points, cyclomatic complexity, Kiviat diagrams); and**
- **problem tracking - tracking and control of problems, defects, and open issues.**

Agree: OIRM agrees with the need to identify measurements for supporting Agency-wide management of software maintenance and will determine the most cost-effective way to accomplish this.

* We will develop cost measurements for resource tracking as recommended.
* We will develop quality measurements for both external users and internal processes.
* We will develop problem tracking measures.

Milestones: OIRM is currently discussing measurement options with other agencies in an effort to comply with GPRA requirements. Milestones will be tailored to conform with GPRA deadlines.

2-2. Based on the metrics defined in our first recommendation, require that OIRM modify its Operations and Maintenance Guidance to establish processes to:

- **define appropriate project status reporting and quality assurance tasks for software maintenance activities;**
- manage the software life cycle, maintenance process, and products within Agency programs in compliance with EPA Directive 2100; and
- implement FIPS Pub 106 guidelines to examine how the software is maintained, exercise control over the process, and ensure the effective software maintenance techniques and tools are employed.

Agree: We agree to update the Operations and Maintenance Manual. It is important to note that the current document does reinforce the principles for managing software presented in FIPS PUB 106. We will continue to reinforce and strengthen those points in the revised document.

Timeframe: Dependent on implementation schedule of recommendation 2-1.

2-3. We understand that this recommendation will be removed entirely by OIG, except for bullet five, which is now bullet one in Rec. 2-2.

2-4. Evaluate commercial defect tracking software, and determine whether any available package should be included as an Agency standard for problem tracking and defect removal in Agency roadmap planning and hardware/software standards documents.

Agree: OIRM is evaluating commercially available problem management systems for its use and broader Agency use, as part of the NDPD's Distributed Systems Management (DSM) program. Currently, NDPD is using a combination of the Information Management (InfoMan) system, commercially available from IBM Corporation, and the HEAT system,

also commercially available. Neither of those systems meets the long-term requirements of NDPD and the Agency, so the DSM program is evaluating alternatives for a replacement to both. After completion of the evaluation, OIRM may propose a standard for Agency approval.

NDPD's current plan is to implement a new problem management system for a limited number of users (approximately 30) in FY95, with the remainder of the approximately 400 InfoMan and HEAT users being migrated to the new system in FY96, assuming the availability of funds. Either the current InfoMan and HEAT systems or a new system would be capable of tracking software defects, if resources were dedicated to that use of the system.

Currently, NDPD is neither budgeted nor responsible to perform application software defect tracking. However, NDPD is considering creating a problem management service under the Working Capital Fund. This service would consist, at a minimum, of access to NDPD's new problem management system for direct use by NDPD customers in tracking their own application system problems, including software defects. The service offering may also include problem record data entry, problem information maintenance, management reporting services, etc., performed by NDPD on a reimbursable basis.

2-5. **Based on the metrics defined in our first recommendation,** require OIRM to update EPA Directive 2115 to make the ADP Review a **comprehensive** review of the system and its support for Agency goals and missions. Include review requirements that would:

- **require quantitative measures of performance, and a user satisfaction survey of the system;**
- require that the program office demonstrate the extent to which the system supports Agency and program office strategic objectives;
- require a periodic review of the effectiveness, accuracy, need, **and economic justification for continued operation for each information system;** and
- ensure that operational systems use an optimum, least-cost mix of resources to meet user functional, data, and other systems' compatibility requirements.

Agree: We agree this document is in need of an update and OIRM intends to issue a revised document. It is important, however, to note that there is specific language in the current document which does address several points made in this recommendation, including the need to conduct regular reviews to determine whether systems are continuing to satisfy Agency requirements, are operating efficiently, effectively and in compliance with standards, operating procedures

and policies. User satisfaction is not overlooked in the document. It is addressed regularly throughout the document.

EPA is currently revising the IRM Review Program to meet the requirements of the Paperwork Reduction Act (PRA) more comprehensively. Part of the revised program's infrastructure will consist of integrating IRM review activities into program review activities and developing evaluative tools to assist in the review of the IRM activities.

EPA Directive 2115 addresses only ADP reviews. The scope, currency, and usefulness of this directive will be evaluated in the process of analyzing the broader requirements of the review program including the development of a more comprehensive tool set to support reviews of the full range of IRM activities (e.g., records management, information security, FIP acquisition management, information systems, etc.).

It appears some of the key review-related issues include the need for better documentation of decisions, documenting the quantitative basis for those decisions, and continuing management attention throughout the system life-cycle. These issues are addressed in the recently revised system lifecycle management policy. Additionally, benefits for use in cost-benefit analyses during the maintenance phase should be incremental (i.e., not inclusive of the existing benefits) in comparison to the costs of the improvements. System reviews should include evaluation of the implementation and use of software quality metrics and productivity, scheduling and business measures. Reviews should also include identification of successes and best practices in these areas.

The requirement "bullets" should be made more generic, for consideration in system reviews, rather than presented as concrete review requirements. The recommendations as stated appear to extend well beyond software maintenance-related areas, beyond the scope of the audit, and into a higher level of program and system management.

The intent of bullet 1 is not clear regarding specific software maintenance issues. For example, the recommendation to include review requirements that would "require quantitative measures of performance" is somewhat ambiguous. One assumption might be that the quantitative measures of performance are associated directly with the metrics developed above (2-1) as opposed to program performance. It is unclear how a user satisfaction survey would relate in this regard. The references in recommendation 2-1 relating to customer satisfaction discuss delivery of software work products, which is hard to equate with any traditional view of "user satisfaction".

APPENDIX I

The word "demonstrate" is not appropriate in bullet 2. Perhaps the terms "evaluate" (context of reviews) or "document" (context of what is needed to review) would be more appropriate. OMB Circular A-130 policy states that post-implementation reviews of information systems should "validate estimated benefits and document effective management practices for broader use" [emphasis added]. The emphasis is on evaluation of the "anticipated benefits" of a system. The benefits would be those that are derived from program office strategic objectives or derived from the performance measurement development process.

The analysis section of A-130 also states, "agencies should seek to quantify the improvements in agency performance results through the measurement of program outputs." These program "outputs" may be aided by automated information systems to differing degrees. This may make direct correlations to systems problematic, and correlations to software maintenance activities even more problematic and indirect.

The criteria in bullet 3 go beyond the scope of software maintenance in the review process. Federal policies discuss review for a number of issues including economy, efficiency, effectiveness along with meeting mission needs. In addition the FIRMR (201-20.202) discusses selecting the alternative that is "most advantageous" to the government. What is most advantageous may not/does not have to be the lowest cost. This parallels the "best value" approach being implemented in acquisitions. In addition, statutory requirements may override economic justifications in some circumstances for continued system operation. Software maintenance issues are just one aspect for considerations of continued system viability.

Bullet 4 raises a complex issue. In system design, selection of hardware and software should involve consideration of likely resource costs in the maintenance phase. Once a system becomes operational (maintenance phase) and the investments are made, certain aspects of the cost mix may be "set" for the component's useful life (equipment life, software life, technology life, etc.). In addition, the FIRMR (201-20.202) discusses selection of the "most advantageous" alternative to the government. Because of the myriad of other considerations, the selection may not be the least cost "mix". Yet it still may be economically justified. More costly items may lead to greater benefits, more than justifying the expenditure. On the other hand, realized costs may be higher or lower than anticipated based on changing or unforeseen circumstances introducing an element of risk. Thus, what was economical may become less "justifiable" in hindsight. The mix of factors and cost-benefit related decisions, and hence review factors and objectives, may be very different for

program information systems versus administrative systems, especially under the GPRA requirements for major programs.

Timeframe: Dependent on implementation schedule of recommendation 2-1.

Chapter 3

We agree in principle with this chapter's recommendations, with two exceptions. Our first exception centers on our belief that the project accounting system module is not an appropriate tool for the capitalization of software. Our second exception is that the Contracts Payment System (CPS) and the EPA Payroll System (EPAYS) do not have to be included as financial systems in future reports because they have already been included in our financial system inventory reported to the Office of Management and Budget (OMB).

Page 36, third paragraph -- This states that, "However, only 'significant' changes were subjected to these [cost-benefit] requirements. Neither simple cost estimates nor cost-benefit analysis were required for those *in*significant IFMS change requests which were directly handed off to the contractor through FMD's Action Request Tracking System (ARTS)."

The statement is incorrect and should be removed from the report. No changes are "directly handed off" through ARTS. We receive "simple cost estimates" for small changes; they are documented within the Change Management System in place for IFMS. The report should also refer to the 1994 cost-benefit analysis for the full IFMS system, which another OIG team reviewed in detail.

Page 41, final paragraph -- This states that, "Even though timeshare and telecommunications charges for the systems reviewed total more than $28 million; none of the information system officials interviewed in this audit expressed interest in tracking or controlling costs associated with timeshare or telecommunications costs, unless these costs directly affected their budget."

The IFMS Executive Management Group, at its December, 1994, meeting, approved an action item explicitly recognizing timeshare and telecommunications costs as an "integral part of the full IFMS Life Cycle costs and requiring that future project plans take such costs into account when presented to the group." Further, the Chief Financial Officer's response to another OIG team's audit of IFMS management agreed with a similar finding and embarked on a mechanism for dealing with such costs.

The final audit report should acknowledge this activity and the OIG should also modify its assertion about "none" of the officials interviewed.

RECOMMENDATIONS:

We recommend that the Assistant Administrator for OARM, **in his capacity as Designated Senior Official for IRM and, when appropriate, in cooperation with the Executive Steering Committee for IRM:**

3-1. Ensure that the Chief Financial Officer (CFO) project related to project cost accounting provides the ability to accumulate system level costs. Continue to coordinate these efforts with the working capital fund initiative.

Partially Agree: Because of our prior actions, we disagree with the need for this recommendation. One of our key objectives in implementing universal usage of the IFMS six-field (41 character total capacity) account code structure in all EPA systems was to respond to various offices' needs to track project level costs at a greater level of detail, and by additional attributes, than was possible with the single-field 10 digit account code. Universal usage of the IFMS account code structure began in October 1994. We also plan to install the Project Cost Accounting System (PCAS) module during July of 1995 to support the Working Capital Fund. PCAS will be available for consideration in FY96 for meeting other Agency cost accounting requirements.

The Agency issues annual guidance on the use of the account code structure, and the guidance that will be issued later this fiscal year will address uses of the IFMS account code fields for funds control and Superfund site project codes, among a limited number of other uses.

As noted elsewhere in this response, EPA already has a significant amount of information on key systems' costs. Whether use of the remaining capacity in the IFMS account code structure for additional identification and tracking of systems' costs is appropriate, necessary and otherwise cost-effective will be evaluated during FY96 along with the evaluation of PCAS usage.

Corrective Action	**Target Date**
- Complete evaluation of needs for additional systems' cost tracking.	06/30/96
- Implement policies, procedures and requirements for any additional tracking of	10/31/96

system costs through the IFMS account code structure or PCAS.

3-2. Incorporate requirements for the accumulation and capitalization of all new development costs and major enhancements which meet the $5,000 capitalization threshold into the project cost accounting project. Establish an interim process to accumulate major system costs to be capitalized. These costs should be incorporated into the financial statements as appropriate. (FMD)

Partially Agree: We do not fully agree with this recommendation. We have not determined that the project accounting system module would be an appropriate tool to meet our overall objective for capitalizing software. However, we agree with the recommendation as it relates to the capitalization of software costs and their recognition in the Agency's financial statements.

An action plan has already been developed to implement these requirements. A Quality Action Team (QAT) was formed to develop a plan to improve the Agency's accounting policies and procedures for all capital assets. Our plan, which was submitted to the Agency's Senior Resource Committee in June 1994, is to: (1) perform a comprehensive analysis of accounting policy and procedural requirements for capitalizing assets, and (2) issue revised policies and procedures by December 1995. We are also evaluating the feasibility of implementing a new Integrated Fixed Asset System by July 1996. As part of this initiative, we will address requirements for capitalizing system costs.

We believe that our target milestone dates are reasonable and will ensure that any policy and procedural changes implemented will effectively address this recommendation. Moreover, the dates take into consideration the fact that the Financial Accounting Standards Advisory Board (FASAB) is scheduled to issue revised standards for federal property and managerial cost accounting. The new FASAB standards could establish major changes in current federal accounting principles and practices.

Corrective Action	Target Date
- Complete analysis of policy and procedural changes.	07/31/95
- Issue draft policy revisions.	10/31/95
- Issue interim revised policy.	12/31/95
- Issue final policy directive.	09/30/96

3-3. Change the OMB Circular A-11 40B report on financial system obligations to reflect system costs for telecommunications and timeshare, include CPS and EPAYS as financial systems in future reports. (IFMS-PMS, FMD, WCF, NDPD)

Agree: We agree with the recommendation, providing that the Office of Inspector General delete "include CPS and EPAYS as financial systems in future reports" from the recommendation. EPA's 40B report to OMB for FY 1995, containing information on FY 1994 through FY 1996, includes the Contracts Payment System and the Payroll System as financial systems. The recommendation need not refer to CPS and EPAYS. We have provided a copy of the formal transmittal to the OIG staff.

Currently, EPA already reports telecommunications and timeshare costs, in aggregate, to OMB in Exhibit 43 under Circular A-11. For the next OMB report, we propose using data accumulated at the National Computer Center (NCC) in support of the Working Capital Fund.

Corrective Action	Target Date
- Include Timeshare and Telecommunications Costs within FY 1996 Exhibit 40B on Financial Systems, using cost data provided by NCC.	10/15/95

3-4. Require the completion of a feasibility study for replacing or modifying the timeshare management system to provide accurate levels of workload accumulation for individual major systems for both NDPD capacity planning and system managers. (NDPD, WCF)

Agree: NDPD, as a result of the Working Capital Fund Mainframe Account Code Clean-up Team, is in the process of making changes to the TSSMS which include a required field for the National ADP System Code. The purpose of this field is to enhance the Agency's ability to capture system utilization and the associated costs.

APPENDIX I

The WCF Team sent Account Code Clean-up worksheets for all current EPA mainframe accounts to the appropriate SBOs. The SBOs were asked to coordinate the review of each account with the appropriate ADP System Administrator to ensure the account is still necessary and to add information for the new fields, one of which is the National ADP System Code.

Milestones:
-- Feasibility Study - Completed
-- Design and Develop Enhancements - Completed
-- Test and Review Enhancements - Due date 4/95
-- Enhanced TSSMS Software Becomes Operational - Due date 4/95

3-5. **Provide the capability within the system access and accounting systems to capture and accumulate resource utilization costs for the different life cycle phases of each information system (e.g., maintenance programming, operations, user access, etc.). (NDPD, WCF)**

Partially Agree: As described above, NDPD is in the process of making changes to TSSMS which include a required field for the National ADP System Code. The TSSMS system captures computer related charges (e.g., data storage) for each ADP Account and specific utilization charges for each authorized user of that account. TSSMS captures system utilization, but it does not have any way of distinguishing for what *purpose* the system was being utilized. That is to say, TSSMS cannot determine whether the account was being used to perform maintenance or for basic access to the system. System owners have the capability to establish separate account codes that they use for specific purposes, such as maintenance, if they should wish to account for their system utilization in that way. It is worth noting that TSSMS is designed as an on-line utilization system. It is not designed to capture or record total labor and contract costs for designing and developing system enhancements or system maintenance. The costs it tracks will always be only a component of total system maintenance costs.

3-6 We understand that this recommendation will be deleted in entirety.

3-7 We have been asked to respond to the following potential revision of this recommendation:

Establish thresholds to enforce the requirement of cost-benefit analyses for all major changes to application systems, **using the criteria for system classification outlined by EPA Directive 2100, Chapter 17.** The cost benefit analyses should provide managers, users, designers, and auditors with adequate cost and

benefit information to analyze and evaluate alternative approaches. The cost benefit document should contain a summarization of the criteria used in the evaluation, as well as the estimated costs and benefits. (IRMPG, OCSS)

Partially Agree: It is important to note that benefits for use in cost-benefit analyses during the maintenance phase should be incremental (i.e., not inclusive of the existing benefits) in comparison to the costs of the improvements. It doesn't make sense to compare total system benefits with incremental changes. On the other hand, if the system doesn't have a cost-benefit analysis, as a minimum, a baseline benefits analysis should be performed from which incremental benefits would be determined. Certain changes to administrative systems, such as payroll (where tax law changes or withholding rates need frequent changes), should not require cost-benefits analyses. Criteria need to be developed to indicate when cost-benefit analyses are needed (i.e., when a major upgrade or version change is contemplated, or after so many years, an obsolescence review should be undertaken).

The requirement for, and size of, system-level cost-benefit analyses should be geared to the size and mission-importance of the application. A full blown cost-benefit analysis for smaller systems may not be cost-effective. Guidance should be written to advise programs on the level of detail needed for cost-benefit analyses based on system size and importance. The smallest of PC-based systems may require no cost-benefit analysis, or perhaps a very minimal one, especially if there is a time-critical, intra-office need.

Chapter 4

Page 49, last paragraph -- The System Development Center's award fee process is customer-driven, and the customers are well aware of the services they have received. The award fees received by contractors are just as likely to be deflated as inflated, based on performance.

Page 50, last paragraph -- The emphasis on a particular change control request form is not as important as an emphasis on meeting software maintenance objectives. Both GICS and FINDS will come under more standard configuration management controls due to the OIRM reorganization, which consolidates all OIRM application systems work. Many EPAYS changes are analogous to what would be simple table changes in a more modern system (i.e., they reflect requirements to change data rather than to change software). The apparently high number of EPAYS modifications should be viewed in this context.

APPENDIX I

Page 61, second paragraph -- This states that, "CPS was a unique case since maintenance effort was not contracted out, but instead performed in-house by full-time EPA employees. Again, the coding standards cited by CPS management did not establish "rules" for writing source code. Rather, these standards pertained to existing Job Control Language (JCL), naming standards, and screen standards."

Although this statement is mostly correct, we question the usefulness of trying to impose "rules" on our FTE programmers. Every programmer has their own unique style and creativity that would not be drastically altered or improved by imposing a strict set of rules. Even if rules were to be imposed, there would be undue overhead associated with their enforcement. Instead, we have provided "standards" for screen layouts, program naming conventions, use of common sub-programs, on-error conditions, etc. We believe that these are the guidelines that programmers need, rather than a set of "rules" that must be followed. We believe that the wording in this paragraph should be rephrased.

Page 64, second paragraph -- This states that, "The functional user community was not given the opportunity to perform user acceptance testing in three (CPS, FINDS, and GICS) of the ten application systems reviewed."

This statement is incorrect. For all CPS modifications where there are functional changes (i.e., except for corrective maintenance), CPS systems staff conduct a "Train the Trainer" session with a user representative from the affected functional area. This user representative is then responsible for training all other members of that functional area. For large-scale development efforts, a separate QA environment is established specifically for users to conduct acceptance testing. CPS software changes are not moved into the production environment until the user community has ensured that the changes meet their operational requirements, and they have been adequately trained. CPS should be removed from this list of three applications where user acceptance testing is not performed.

Page 65, second paragraph -- This states that, "Also, uncertainty existed regarding whether 'user acceptance testing' was performed on software changes to IFMS which originated from Agency components outside of OARM/ASD. Our review of test documentation related to recent IFMS software modifications did not disclose confirmation of user acceptance testing."

APPENDIX I

For IFMS, we dispute the intimation that there was no user acceptance testing of changes. The "uncertainty" within the OIG team may be a result of a flawed survey instrument, which was not adequate to disclose substantial activity in user acceptance testing. More generally, the draft report overlooks significant activity by "client" organizations outside of OIRM. We know this comment applies to IFMS and EPAYS, and believe it applies to other systems as well.

The Financial Management and Budget Divisions each have a Branch whose specific task is to gather and interpret requirements for the financial system. Those branches perform user acceptance testing. They function as clients to ASD. They are responsible for managing requirements gathered from users of the system outside OARM. Our change control boards (the System Management Group and Executive Management Group) have active non-OARM users to help OARM ensure that those users' needs are being met.

Page 66, third paragraph -- This states that, "Furthermore, CPS managers, who were responsible for their own software maintenance, did not perform sufficient independent tests for specific types of software modifications."

This entire paragraph is incorrect. The Test and Acceptance Unit of the Financial Systems Section was responsible for both production support (i.e., critical corrective maintenance) and Quality Assurance testing. This unit is staffed with computer specialists who do programming or testing. Established procedures ensure that program changes for critical corrective maintenance are independently reviewed and tested. Although this was done by members of the same unit, there is adequate separation of duties/responsibilities. We believe this entire paragraph should be deleted.

Page 67, third paragraph -- This states that, "However, no CPS or AIRS modifications, regardless of their level of significance, were subject to independent V&V testing prior to implementation in the production environment."

This statement is totally incorrect. CPS maintenance activities, with the exception of critical corrective maintenance, are done by the System Development Unit (SDU) of the Financial Systems Section. Upon completion of their unit testing, SDU personnel pass software on to members of the Test and Acceptance Unit for formal testing and preparation for moving into the production environment. Although the development and testing activities are done by FTEs within the same section, their duties and responsibilities are adequately separated to ensure independence. We believe that the reference to CPS should be removed.

APPENDIX I

Page 71, second paragraph -- This states that, "Since the commencement of this audit, OIRM initiated action to research and select a suitable SCM tool..." In actuality, OIRM began this research long before commencement of this audit.

RECOMMENDATIONS:

We recommend that the Assistant Administrator for the Office of Administration and Resources Management, **in his capacity as Designated Senior Official for IRM and, when appropriate, in cooperation with the Executive Steering Committee for IRM:**

4-1. **As a subset of Recommendation 2-1,** define Agency-wide measurable performance indicators which will enable management to:

- evaluate the **efficiency and** effectiveness of the change control process;
- assess overall stability of the application system;
- assist in allocating budgetary resources; and
- identify software maintenance trends and highlight instances of program rework or excessive corrective modifications.

Agree: These points can be addressed in the revision of the Operations and Maintenance Manual or in associated practice papers. Software testing, if done properly, determines whether the software performs as expected by the user(s). Source code reviews would therefore check for adherence to coding standards. Many of the applications now being developed are generated from CASE tools and are regenerated from the models maintained by the tools when there are changes. Obviously, review of automatically generated source code is of limited value. Other source code is programmed in a variety of languages and detailed standards may not be available. Such code could, however, be checked for adherence to standards in a Verification and Validation process based upon the high-level EPA coding standards.

Timeframe: The performance indicators would be projected for release by the end of FY96.

4-2. Initiate actions to:

- **use** the software maintenance practices and policies of the **System** Development Center (SDC) **and revise them, where appropriate,** to ensure that the individual controls and

reviews outlined in this report are sufficiently and actively addressed;

- **utilize the SDC to** promote the use of **the best practices in software maintenance activities, within the framework required under Chapter 17 of EPA Directive 2100,** throughout the Agency; and

- emphasize the need for and importance of controlled software maintenance practices, through IRM Forums and other meetings regarding EPA's information system activities.

Agree: The software maintenance practices and policies at the SDC will be revised as needed to reflect the requirements in the revised O & M Manual. We agree that the "state of the practice" software maintenance activities at the SDC should be promoted throughout the Agency and will ensure that the SDC maintenance practices and policies are distributed and briefed to the IRM community. We also agree that the need for, and importance of, controlled software maintenance practices should be communicated in appropriate meetings of the IRM community. The SDC already presents briefings and brown bag seminars on various topics such as the SDC Product Development Process and the SDC Product Assurance Policy, and will include software maintenance practices as a topic. Delivery Order Project Officers are required to attend the SDC Product Development Process briefing and are routinely invited to attend the brown bag seminars. For example, Dr. Louis Blazy from USDA is returning to the SDC on March 23 to present the more detailed portion of his software metrics program briefing. SDC DMMG staff and management will attend, along with SDC clients.

Timeframe: Dependent upon final determination of revisions to be made to the O & M Manual.

4-3. Modify existing Agency guidance, based on the performance indicators defined in our first recommendation, for managing the software maintenance process and products throughout the Agency. Require formal procedures for automated application systems to include, at a minimum:

a. That each application system establish a standardized form for initiating all requests for software changes, regardless of anticipated level of effort. The form should minimally include: (1) requestor name; (2) date; (3) priority; (4) problem description/justification; (5) type of change; (6) management approval; and (7) completion date.

APPENDIX I

Agree: Per our response to recommendation 2-2, we have agreed to update the Operations and Maintenance Manual and will continue to include the requirement that there be a systematic approach to change requests. It should be noted, that in the current version of the O&M Manual, there is a requirement to document change requests, and Exhibit 3-2 on page 3-8 provides a model for the information which should be included in a change request form. We will include those items recommended in the audit which were not in the original model change request form.

b. A requirement that "Major Agency" application systems, which experience high availability requirements, develop and maintain a comprehensive and cohesive change tracking system which will track *all* software changes made to the application system, regardless of the type of proposed change or its anticipated level of effort. In addition to the data stipulated in paragraph 3.a above, the tracking system should require data, such as: change request number, affected programs/modules, and comments field for referencing associated change requests.

Agree: We will provide information about this type of tracking system in the revised document.

c. A classification system for change requests which delineates the types of changes being made to the application system based on the nature of change (**e.g.**, adaptive, corrective, perfective). Level of effort information, if desired, should be maintained separately.

Agree: This recommendation for classifying the type of change request has already been agreed to in our response to Section a. The change request form requires a classification of the type of change requested. This information is relevant to include in a tracking system.

d. A centralized review point, within each system or major subsystem, if applicable, for *all* software change requests, regardless of level of effort.

Agree: This requirement can be addressed in the revised guidance document.

Timeframe for 4-3 a, b, c, and d: Dependent on implementation schedule of recommendations 2-1 and 4-1.

4-4. Modify the Operations and Maintenance Manual and/or Chapter 17 of the IRM Policy Manual to include a requirement for test results **of major application software modifications** to be reviewed by either: (1) a designated panel of technically-knowledgeable reviewers; or (2) the steering committee which initially reviewed and approved the software change. At a minimum, the appointed reviewers should:

- review the results obtained from testing;
- compare test results with the initial request for change, detailed specifications related to the change, and the applicable test plan; and
- compare modified source code with latest production version of code to ensure that no additional unapproved changes were introduced by programmers during the coding process.

Agree: In the update to the Operations and Maintenance Manual, we will reinforce the responsibilities of the reviewing parties. It should be noted that the current version of the O&M Manual describes the responsibilities of the Configuration Control Board in Exhibit 3-1 on page 3-4. We will, however, make sure the role of the reviewers, be they a formal Configuration Management Board or a comparably experienced group, is communicated clearly in the revised document.

Timeframe: Dependent on implementation schedule of recommendations 2-1 and 4-1.

4-5. Revise Chapter 17, Section 8, of the IRM Policy Manual to state "Other relevant Federal and Agency guidance documents which **must** be followed are noted below:" In addition, Revise Chapter 17, Section 8, of the IRM Policy Manual to include FIPS Publication 132 as one of the referenced Federal guidance documents. The revised policy should stipulate thresholds for implementation of independent V&V testing and clearly define the level of effort or other criteria which will be used to determine which software changes are subject to testing.

Disagree: We have committed to revising the O&M Manual and can cite relevant FIPS Pubs. However, we do not agree with the recommendation to revise the policy for the following reasons:

- Introductory language in FIPS Pubs 106 specifically states that use of that Guideline is encouraged but not mandatory. It would be inappropriate for EPA be more prescriptive than what NIST presents in their direction to Agencies.

. Considering NIST's recent announcement of their intention to rescind a number of FIPS Pubs, it is better to have a statement of policy of commitment to the FIPS Pubs in aggregate rather than citing individual publications which may soon be rescinded. Please note that the Agency's Software Management Policy (Chapter 4 of Directive 2100) provides this global commitment..."EPA program officials will adhere to FIPS and guidelines as published or adapted for the Agency in developing, documenting, maintaining and using software applications."

. The policy is intended to provide high level statements of principle and direction rather than procedural instructions. For that reason, we will address more detailed procedural information, such as thresholds for implementation of independent V&V testing, in the revised Guidance document.

. It was just recently enacted, receiving the concurrence of all organizations, including the Office of Inspector General. Considering how long it took to get the initial policy established, it does not seem cost-effective or prudent to reopen the green border process and invite additional changes, some of which may in fact weaken the existing policy.

4-6. Modify the Operations and Maintenance Manual and/or Chapter 17 of the IRM Policy Manual to include a requirement for acceptance testing by the user community. User acceptance testing should take place prior to the implementation of a software modification and should be of sufficient duration as to adequately examine and evaluate application functionality: This requirement could reasonably be limited to those software changes which:

- represent a new program or module within the application;
- represent a major system enhancement to the application;
- represent a level of effort which is technically considered by management as a "development" project, rather than a routine or minor maintenance action item; or

- is comprised of a group of software changes which collectively represent a considerable change to the application's performance.

Agree: The current Operations and Maintenance Manual contains requirements for acceptance testing but we will reinforce and strengthen this point in the revised document.

Timeframe: Dependent on implementation schedule of recommendation 2-1.

4-7. Through IRM Forums and/or other meetings with the IRM community, promote the benefits of periodic reviews of historical change control data as a valuable management tool. Illustrate to system managers how pending modifications and historical data can be used to detect and evaluate trends regarding the nature and frequency of processed software changes. Emphasize the usefulness of historical data to discern inadequacies in review and test procedures or inadequacies in the contractor's performance. Encourage system managers to make use of available historical data to judge the stability of their application systems, as well as the adequacy of their current change control practices.

Agree: NDPD has already initiated action to bring its customers into the already implemented change management process. On March 1, 1995 at the monthly SIRMO meeting the long range change management function was announced. The SIRMOs were informed that they would be asked in a memo in late March or early April to identify a system contact within their organization to work with NDPD to inform NDPD of major/critical system changes. NDPD would then coordinate the customers changes with planned NDPD changes. All long and short range changes will be summarized on a listserver.

We intend to further discuss this at one of the IRM Branch Chiefs meetings, publicize it in the monthly newsletter, the CONNECTION and discuss it at the biannual Outreach teleconferences with regional office and program office personnel.

4-8. Make a software configuration management (SCM) tool, such as the ENDEVOR product already implemented in IFMS, available to EPA's program offices, and encourage system officials to implement its use on application systems which were classified as "major agency" systems due to their high availability requirements.

In preparation for adoption of this recommendation, IRM management should establish a definite implementation schedule for those "major agency" applications under its control. The

schedule should be aimed at enforcing SCM implementation within a reasonably short period of time. Desirable features of a good SCM product are outlined in Appendix VII of this report.

Agree: NDPD is announcing the availability of ENDEVOR to the entire NCC user community through a User Memo. The memo should be published within three weeks and the ENDEVOR SCM product will be available on June 1, 1995.

As noted earlier, ASD has already implemented IFMS under control of ENDEVOR. Work has begun on ENDEVOR for EPAYS, with an expected implementation target date before the end of FY95. ASD will then focus on GICS, with implementation expected in FY96.

Appendices

Appendix I, Page 85, second paragraph -- This states that, "IFMS runs 3,360 programs at a fiscal 1993 cost of almost $16 million."

The cost values are for the entire IFMS family of systems, which includes the commercial product, the Management and Accounting Reporting System, the predecessor financial management system, and several small interface programs that accept data from other mixed systems such as GICS. This definitional point is important because the text in Chapter 2 of the draft audit report refers to IFMS as if it were a single system, when all of the performance and ABEND data refer, rather, to several substantial computer systems.

Appendix V, Pages 106 and 107 -- We do not agree with a number of specific judgments about IFMS. The chart uses 'Y' for INADEQUATE controls so that the "Condition exists"; 'P' for PARTIALLY INADEQUATE; and 'N' for controls that are ADEQUATE.

- *Lack of Measurement Performance Indicators*
  We create a Spending Plan, reviewed by steering groups (change control boards) for the system, that categorizes Operations & Maintenance in a way similar to the categories proposed in the draft audit report. Further, the Problem-Cause-Solution form creates a record of system problems. Change the condition from 'Y' to 'P'.

- *Inadequate Coding Standards and Review*
  We incorporate proper standards within requirements of each Delivery Order. ENDEVOR is also in place. Furthermore, we observe that as Commercial Off The Shelf software, warranted to comply with GAO and JFMIP standards, there is an issue whether coding standards for the commercial component of the IFMS family

of systems should apply at all. Change the Condition from 'Y' to either 'P' or 'N/A'.

- *Lack of Sufficient Testing and Acceptance by Functional Users* We believe the audit team overlooked substantial efforts within the Financial Management and Budget Divisions to perform this function. Change the Condition from 'P' to 'N'.

APPENDIX II

SYNOPSIS OF APPLICATION SYSTEMS REVIEWED

AEROMETRIC INFORMATION RETRIEVAL SYSTEM (AIRS)

AIRS was implemented in 1987 at a cost of $8 million. AIRS stores air quality, point source emissions, and area/mobile source data required by Federal regulations from the 50 States. Monitoring is required for the criteria pollutants based on population, pollutant sources, geographical area, etc. Point sources emitting more than 100 tons per year of any criteria pollutant (except 5 tons per year for lead and 1,000 tons per year for carbon monoxide) must report actual or estimated annual emissions data. The Office of Air and Radiation is responsible for the operation and maintenance of AIRS. AIRS runs 5,379 computer programs with a fiscal 1993 operating cost of over $11 million.

COMPREHENSIVE ENVIRONMENTAL RESPONSE, COMPENSATION, AND LIABILITY INFORMATION SYSTEM (CERCLIS)

CERCLIS was implemented in 1987 at a cost of $3.9 million. Version 2.0 supports EPA Headquarters and regions for the management and oversight of the Superfund program. It has two purposes: (1) maintain an automated inventory of abandoned, inactive, or uncontrolled hazardous waste sites; and (2) act as a vehicle for Regions to report to Headquarters the status of major stages of site clean-up. A hotline supports CERCLIS version 2.0 operations at Headquarters and Regional offices. The system provides a decentralized national system where each region controls and enters its respective data on regional subsystems. OSWER is responsible for CERCLIS operations. CERCLIS runs 449 computer programs at a fiscal 1993 operating cost of almost $4 million.

CONTRACT PAYMENT SYSTEM (CPS)

CPS was implemented in 1987 at a cost of $1.2 million. CPS, which is maintained by OARM, provides a comprehensive financial database for the more than 3,200 Agency contracts. CPS is a major sub-system to IFMS, and provides detail and summary level information on contract award and invoice data. User-friendly menus enable finance personnel and external users to examine information via the on-line query capability. Other benefits include warehousing invoices to meet the Prompt Payment Act, generation of the invoice approval form and use of electronic approval by project officers and contracting officers, and generation of reports to accommodate the Superfund legislation. CPS runs 649 computer programs at a fiscal 1993 operating cost of over $1.5 million.

APPENDIX II

EPA PAYROLL SYSTEM (EPAYS)

EPAYS was implemented at EPA in 1984. EPAYS was obtained from the Department of the Interior and, as such, EPA did not incur development costs. EPAYS features a standardized nationwide data entry system for Time and Attendance, Payroll and Personnel data (i.e., the TAPP system). The system also contains a labor distribution function for Agency payroll accounting and biweekly production of Agency payroll requirements. The system has the ability to distribute personnel management information to meet management and regulatory reporting requirements. OARM maintains EPAYS, which runs 413 programs at a fiscal 1993 operating cost of over $2.5 million.

FACILITY INDEX SYSTEM (FINDS)

FINDS was installed in 1981 and details on its installation cost were not available. FINDS is a computerized inventory of facilities regulated or tracked by EPA. OARM is responsible for FINDS operations. All facilities are assigned unique Facility Identification numbers which serve as cross-reference numbers to facility information residing in the EPA program system. This function supports cross-media data integration by tracking facility locations across EPA program offices. It is used to assist in integrated enforcement analysis, "hot spot" determination, risk analysis, etc. FINDS is a data base system that points to other EPA application systems. FINDS management could not identify its operating budget for fiscal 1993.

GRANT INFORMATION AND CONTROL SYSTEM (GICS)

GICS was first implemented in 1972, and subsequently updated in 1986. The system development cost and the cost associated with the conversion in 1986 are unavailable. GICS, which is maintained by OARM, is the Agency's management information system for all grant programs. This national system is used by Headquarters, Regions, and States to administer and monitor grants. Report menus are available for batch or on-line reporting. On-line data entry systems for the construction and non-construction programs have been customized to provide for updating and tracking of the grant process. GICS runs 5,802 computer programs at an annual cost of over $1 million.

INTEGRATED FINANCIAL MANAGEMENT SYSTEM (IFMS)

IFMS was purchased in 1987 for $510,000 and installed in 1989, but will not be fully implemented until 1998. IFMS records do not differentiate development and enhancement costs; as such, development costs can not be quantified. IFMS was designed expressly for

government financial accounting and supports GAO Title 2 requirements[36], OMB internal control requirements, and OMB's A-127 initiatives[37]. IFMS performs funds control from commitments through payment; updates all ledgers and tables as transactions are processed; provides a standard means of data entry, edit, and inquiry; and provides a single set of reference and control files. IFMS has table driven editing, posting, and reporting capabilities. It supports on-line inquiries as well as standard and ad hoc reporting. OARM is responsible for IFMS, which runs 3,360 programs at a fiscal 1993 cost of almost $16 million.

IFMS includes several subsystems as part of the "Core Financial System." These include: General Ledger, Budget Execution/Funds Control, Budget Preparation, Accounts Payable/Disbursements, Accounts Receivable/Collections, Travel, Purchasing, and Standards Reporting. FMS/SPUR and MARS are systems which provide additional reporting capabilities. In addition, there are several "Mixed Systems" which are part financial, part programmatic. These mixed systems include EPAYS, CPARS, ADCR, RMIS, ADPS/CIS, CIS, GICS, and PPAS. Several of these systems have direct interfaces with IFMS while others require data to be re-entered into IFMS.

PERMIT COMPLIANCE SYSTEM (PCS)

PCS was first installed in 1975, and updated 10 years later. The original development cost is unavailable; the approximate cost of the update was $885,000. PCS is a computerized management information system for tracking permit, compliance, and enforcement status for the NPDES program under the Clean Water Act. PCS contains information on more than 63,000 active water discharge permits issued to facilities throughout the nation. EPA's Office of Enforcement and Compliance Assurance (OECA) is responsible for the operation and maintenance of PCS. EPA Regional and State users of PCS are responsible for the entry and the quality of data in the system. The system components are: (1) on-line and batch data entry; (2) batch update; and (3) batch and on-line retrieval packages. PCS runs 951 programs at a fiscal 1993 cost of almost $4.8 million.

[36] GAO Title 2 states accounting systems must conform to the accounting principles, standards, and related requirements and internal control standards prescribed by the Comptroller General.

[37] OMB Circular A-127 states agencies shall establish and maintain a single, integrated financial management system, which may be supplemented by subsidiary systems.

APPENDIX II

RESOURCE CONSERVATION AND RECOVERY INFORMATION SYSTEM (RCRIS)

RCRIS was installed in 1991 at a cost of over $18 million. It replaces the permanently archived Hazardous Waste Data Management System as the major system supporting the RCRA program. RCRIS accommodates new data as required by the 1984 Hazardous and Solid Waste Amendments. It provides interactive, on-line data edit checking; offers additional facilities for processing and reporting; and allows the use of inexpensive personal computers for most tasks. It is used interactively on a day-to-day basis at the State and Regional level, and is updated via batch uploads and merges on a monthly basis to the national oversight database. OSWER is responsible for RCRIS operations. RCRIS runs 3,403 programs at a fiscal 1993 operating cost of almost $7.5 million.

STORAGE AND RETRIEVAL OF WATER QUALITY INFORMATION (STORET)

STORET was installed in 1965 at an approximate cost of $1 million. The STORET system, which is maintained by the Office of Water, assists State and EPA officials in making pollution control decisions by providing a capability to store, retrieve and analyze water quality information. Current emphasis of control decisions are: issuing water quality based NPDES permits; including toxic pollutants in water quality standards; evaluating water quality impacts of control programs; and assessing levels of toxic pollutants, including dioxin and other bioaccumulative pollutants in the aquatic biological data, hydrologic data, stream reach data, ground-water data, and other related information. The system is used by State and EPA analysts to assemble and analyze data to support each of the above types of decisions. As noted in Chapters 2 and 3, 405 programs are run on STORET at a fiscal 1993 operations and maintenance cost of over $ 1.3 million.

TOXIC CHEMICAL RELEASE INVENTORY SYSTEM (TRIS)

TRIS was installed in 1988 at a cost of $285,000. The EPA internal system for TRIS contains all non-trade secret data submitted to EPA for chemicals and chemical categories listed by the Agency. Data include chemical identity, amount of on-site users, releases and off-site transfers (including publicly owned treatment works), on-site treatment, and minimization and prevention actions. The Office of Prevention, Pesticides and Toxic Substances maintains the TRIS system. 276 programs are run on TRIS at a fiscal 1993 cost of $9.5 million.

APPENDIX III

AUDIT METHODOLOGY

We initiated this audit with two survey instruments: a management questionnaire directed to senior Agency managers, and a vulnerability questionnaire directed at system managers. The management survey questionnaire included questions which impact the maintenance of the software, such as the need for computer support within the program office, success factors for computer-related support, mission-based planning, system development methodologies, software maintenance goals and objectives, policies, procedures and standards, staffing, cost records, change approval, and record keeping. The survey questionnaire was sent to nine Assistant Administrators, ten Regional Administrators, and twelve Laboratory Directors. We distributed a summary of the responses obtained from the management questionnaire to participating management on October 18, 1993.

We judgementally selected ten application systems for a vulnerability assessment and more detailed review. The ten systems were identified in our 1991 Special Review of EPA's Major Information Systems, report number E1RMG1-15-0041-1400061, as high risk or very high risk systems. These systems were all national in scope, and eight of the ten were identified by the Agency as major systems requiring security plans in 1989. These systems were from a cross-section of the Agency, including administrative, enforcement, compliance and scientific data. Survey questionnaires were followed by detailed interviews with the system managers and some senior program managers.

The vulnerability questionnaires included questions which impact the maintenance of the software, such as system interfaces; number and size of programs; age of the software; age of documentation; programming languages and data base management; frequency of modification; processing type and frequency; and record and file sizes. A standard GSA risk model was modified by incorporating maintenance-related questions. The questions and risk model were reviewed and modified by OIRM, and then reviewed by the National Institute for Standards and Technology Computer Systems Laboratory. The responses to the vulnerability questionnaires were put into the risk model and ranked.

With regard to our review of cost management, the initial objective was to determine the true cost of software maintenance for the ten major systems in fiscal 1993. All of these systems were defined as major information systems by the Agency. Seven of the systems -- AIRS, IFMS, TRIS, CERCLIS, RCRIS, PCS, and STORET -- were reported to OMB under Circular A-11 as major information systems meeting reporting thresholds.

APPENDIX III

We solicited software maintenance cost data from the ten system managers. The records received varied greatly from system to system. Based on the inconsistency and incompleteness of these records, it proved impossible to separate maintenance costs from operations costs. Because system managers could not separate software maintenance costs from operations costs, these costs could not be grouped into specific categories identified in FIPS Publication No. 106 and the EPA Operations and Maintenance Manual.

At the time of our fieldwork, nine of these systems were in a maintenance phase and not scheduled for any major revision. All costs reported on an annual basis fell into the categories of maintenance (perfective, adaptive and corrective), and operations. STORET was the one exception, because STORET was undergoing a major revision. Therefore, we subtracted the STORET redevelopment costs reported to OMB from the total reported obligations for 1993.

For all systems, NDPD provided timeshare cost information. Gathering the cost information was more difficult than anticipated. The costs provided in Chapter 3 may not be entirely accurate, but were the best they could do. Telecommunications costs were computed using NDPD's algorithm, which is based on timeshare costs.

We attempted to determine system costs recorded in the accounting system, however, the Agency does not capitalize costs associated with software. This eliminated the possibility of using net present value or other accounting techniques in this analysis. The costs reflected in Table 1 of Chapter 3, are the closest possible cash basis figures, given the data that was available. The column titled "Fiscal 93 Program Costs" contains information which was obtained from the Agency's report to OMB under Circular A-11, Section 43A. For those systems not reported to OMB (CPS, EPAYS, and GICS), costs were accumulated based on system manager records. This is complicated by the fact that each system has multiple TSSMS account codes. This factor hinders the system managers' ability to accumulate and control total usage-based costs, such as timeshare and telecommunications.

We did not independently verify the information received from the NDPD, taken from the report to OMB, and stated by the system managers. Therefore, this information does not meet the requirements set forth in GAO's Government Auditing Standards. However, GAO requirements were met in all other areas of the cost-related findings.

The ten application systems chosen for our review of change control and configuration management practices differed slightly from the ten systems evaluated under other aspects of the software maintenance audit. Fieldwork in this area began later in the audit cycle and we

substituted FINDS for STORET. The following factors contributed to the decision to substitute FINDS for STORET in this section of the audit:

- Management of STORET was transferred from OIRM's Program Systems Division to the Office of Water during mid-October 1993, the same month this aspect of audit was initiated.

- STORET was identified as part of the Office of Water modernization initiative and was in the process of undergoing a complete overhaul due to: (1) the extreme age of STORET, (2) its deteriorating condition and the lack of knowledgeable maintainers, (3) significant problems with connectivity which limited the accessibility of data, and (4) a cooperative move to accommodate the overhaul of the US Geological Survey system.

System managers of the ten application systems were asked to provide information related to configuration management and software maintenance. Copies of policies and procedures established for the purpose of documenting, evaluating, and controlling proposed system changes were obtained, if available. In addition, problem logs and change logs were requested for each system. However, problem and change logs were not available for all systems.

We reviewed applicable Agency guidance, individual Program Office change control policies, applicable procedures governing contractor performance of maintenance activities, and related NDPD procedures to: (1) determine if software change control procedures were standardized for the application; (2) identify individual application system change control processes; and (3) assess the level of control present within the individual application's change control process. System managers were interviewed for additional information. If available, problem and change control logs were evaluated and analyzed for content, format, and usefulness to system managers.

Only four application systems were chosen for the review of specific software modifications and applicable test documentation: AIRS, TRIS, IFMS, and RCRIS. We chose these systems because they demonstrated a high percentage of software changes during fiscal 1992, as compared to the number of computer programs or modules used to operate the application. For this phase of the audit, we statistically selected a sample of software changes from available historical logs, and reviewed test plans, related test analysis, and other pertinent documentation which demonstrated the review and approval process for testing, evaluating, accepting and implementing software changes into the production environment.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX IV

FEDERAL AND INDUSTRY CRITERIA AND GUIDANCE

This appendix briefly discussed the Federal requirements and guidance we used to conduct our audit. Federal guidelines, and a number of industry publications were used to form a framework of sensible, stable business practices and, therefore, served as a means to evaluate software maintenance activities.

Paperwork Reduction Reauthorization Act Of 1986

The Paperwork Reduction Reauthorization Act of 1986 requires that Federal agencies periodically evaluate and, as needed, improve the accuracy, completeness, and reliability of data and records contained in Federal information systems.

Paperwork Reduction Act Of 1980

The Paperwork Reduction Act of 1980 defines agency responsibilities for managing information resources. Each agency shall be responsible for carrying out its information management activities in an efficient, effective, and economical manner, and for complying with the information policies, principles, standards, and guidelines prescribed by the OMB Director. Each Agency shall systematically inventory its major information systems and periodically review its information management activities, including planning, budgeting, organizing, directing, training, promoting, controlling, and other managerial activities involving the collection, use, and dissemination of information.

Government Performance And Results Act Of 1993

Purposes of the Government Performance and Results Act of 1993 include improving Federal program effectiveness and public accountability by promoting a new focus on results, service quality, and customer satisfaction; and helping Federal managers improve service delivery by requiring that they plan for meeting program objectives and by providing them with information about program results and service quality.

By September 30, 1997, the head of each Agency must submit to the OMB Director a strategic plan for program activities. The plan must contain: goals and objectives, including outcome-related goals and objectives, for the major functions and operations of the Agency; a description of the operational processes, skills and technology, and

the human, capital, information, and other resources required to meet these goals and objectives; and a description of how the performance goals are related to the general goals and objectives.

Each agency must prepare an annual performance plan which shall: establish objective, quantifiable, and measurable performance goals to be achieved by a program activity; establish performance indicators to be used in measuring or assessing the relevant outputs, service levels, and outcomes of each program activity; provide a means to be used to verify and validate measured values; and provide a basis for comparing actual program results with established performance goals.

OMB Circular A-11

OMB Circular A-11, dated August 1993, requires reporting of major information system initiatives which will require obligations that exceed $25 million over a system's lifecycle or $10 million in any one fiscal year. It also requires reporting on financial management systems, which include all core financial and mixed systems critical to effective agency wide financial management, reporting, and control. In addition, it requires reporting any financial and mixed systems appearing on the high risk list in the most recent president's budget. Agencies that obligate more than $2 million in a year must also prepare a report of obligations for systems activities including telecommunications, planning, cost-benefit, installation, operations, maintenance and support. Further, it requires system and application software that exceeds $25,000 to be treated as a capital investment. Paragraph 43 of the circular provides the format and requirements of this report. Finally, it requires Agencies to prepare benefit-cost analyses following OMB Circular A-94 for all proposed investments.

OMB Circular A-109

OMB Circular A-109 requires that each agency acquiring major systems maintain a capability to: (1) predict, review, assess, negotiate and monitor life cycle costs[38]; (2) assess acquisition cost, schedule and performance experience against predictions, and provide such assessments for consideration by the agency head at key decision points; (3) make new assessments where significant cost, schedule or performance variances occur; (4) estimate life cycle costs during system design, concept evaluation and selection, full-scale

[38] This circular defines life cycle cost as the sum total of the direct, indirect, recurring, nonrecurring, and other related costs incurred, or estimated to be incurred, in the design, development, production, operation, maintenance and support of a major system over its anticipated useful life span.

development, facility conversion, and production, to ensure appropriate trade-offs among investment costs, ownership costs, schedules, and performance; and (5) use independent cost estimates, where feasible, for comparison purposes.

OMB Circular A-130

OMB Circular A-130 mandates that agencies shall use FIPS and Telecommunications Standards except where it can be demonstrated that the costs of using a standard exceed the benefits or the standard will impede the agency in accomplishing its mission. Agencies may waive the use of Federal standards under certain conditions and pursuant to certain procedures.

This circular also states that an agency official who administers a program supported by an information system is responsible and accountable for the management of that information system throughout its lifecycle. Under this circular, agencies are required to account for the full costs of operating information processing services organizations (IPSOs). When the obligations for such organizations exceed $3 million annually, agencies shall implement a system to distribute and recover the obligations incurred for providing services to all users that: (1) prices each service provided by the IPSO to each user on an equitable basis commensurate with the resources required to provide that service and the priority of service provided; (2) directly distributes the full costs of dedicated services to users; (3) provides for the periodic submission of statements to all users, itemizing the costs of services provided; and (4) provides for the preparation of a report that documents the past year's obligations for operating the IPSO at the close of each fiscal year.

OMB Circular A-132

OMB Circular A-132 mandates that each agency will implement an active agency-wide productivity and quality improvement process. Inherent in a quality design and production process is avoidance of any rework or returns due to errors, unclear procedures, or any other cause. Resources saved by "doing the right thing right the first time" translates into improved productivity.

Measurement systems will be established that are straightforward, easy for managers and employees to understand, and of maximum utility in targeting areas for improvement in all program functions. Measurement systems provide:

(1) quantification in dollar amounts of resources used as the input for production of a service or product;

(2) the number of units (weighted, if applicable) of the product or service provided to the customer;
(3) the total amount of time consumed in providing the service or product to the customer; and
(4) the level of service or product quality, both in terms of customer satisfaction (external quality) and of work performed to provide the service (internal quality).

Once a baseline is established, standards are set by program managers that state what ought to be the level of work accomplished, its quality and its timeliness in order to meet customer requirements. The goals set in the productivity improvement plans, together with the performance standards for each program function should be made part of the Senior Executive Service and merit pay contracts and employee performance standards.

FIPS Publication 106

FIPS Publication 106, Guideline on Software Maintenance, presents information on techniques, procedures, and methodologies to employ in controlling and improving software maintenance. Software maintenance is the performance of those activities required to keep a software system operational and responsive after it is accepted and placed into production. The goal of software maintenance management is to keep systems functioning, and to respond to user requests in a timely and satisfactory manner.

Management is clearly one of the most important factors in improving the software maintenance process. Management must examine how the software is maintained, exercise control over the process, and ensure that effective software maintenance techniques and tools are employed. Software maintenance managers are responsible for making decisions regarding the performance of software maintenance, assigning priorities to the requested work, estimating the level of effort for a task, tracking the progress of the work, and assuring adherence to system standards in all phases of maintenance.

FIPS Publication 38

FIPS Publication 38, Guideline for Documentation of Computer Programs and Automated Data Systems, provides a basis for determining the content and extent of documentation for computer programs and automated data systems. Its intent is to serve as a reference and a checklist for general use throughout the Federal government to plan and evaluate documentation practices.

APPENDIX IV

FIPS Publication 132

FIPS Publication 132, Guideline for Software Verification and Validation Plans, announces the adoption of the Standard for Software Verification and Validation Plans (ANSI/IEEE Std. 1012-1986) as a FIPS Publication Guideline. This standard: (1) provides, for both critical and noncritical software, uniform and minimum requirements for the format and content of SVVPs; (2) defines, for critical software, specific minimum V&V tasks and their required inputs and outputs that shall be included in SVVPs; and (3) suggests optional V&V tasks to be used to tailor SVVPs as appropriate for the particular V&V effort.

NBS Special Publication 500-129

National Bureau of Standards (NBS) Special Publication, Software Maintenance Management, focuses on the management and maintenance of software and provides guidance to Federal government personnel to assist them in performing and controlling software maintenance. It presents techniques and procedures to assist management in controlling the activities performed by maintenance personnel (i.e., problem reporting, software quality assurance, code walkthroughs, software configuration management, and test plans and procedures).

GAO Title 2

GAO "Title 2: GAO Policy and Procedures Manual for Guidance of Federal Agencies" requires ADP software valued at $5,000 or more, with a useful life of two years or greater, to be capitalized as property, plant, and equipment.

GAO Executive Guide

A 1994 GAO Executive Guide entitled "Improving Mission Performance Through Strategic Information Management and Technology: Learning From Leading Organizations" lists 11 key practices to improve IRM activities within any organization. The seventh of these practices is to manage information systems as investments. The specific attributes of this practice include: (1) linking information systems decisions tightly to program budget decisions and focusing them on mission improvement; (2) using a disciplined process -- based on explicit decision criteria and quantifiable measures assessing mission benefits, risk and cost -- to select, control, and evaluate information systems projects using post-implementation reviews; and (3) balancing the proportion of maintenance expenditure versus strategic investment.

APPENDIX IV

GSA Guide For Acquiring Software Development Services, Chapter 16, Software Operation and Maintenance

Software operation and support involve activities that allow the software and its users to perform intended functions acceptably. Performance monitoring and configuration management have particular significance. The agency must measure whether the software is using more than the expected amount of resources. This requires setting performance standards and then taking measurements to compare against them. When the software deviates from the established standards, the agency knows a problem exists. When establishing performance standards, agencies should consider the following:

- Number of input/output reads and writes;
- Memory paging activity;
- CPU cycles used;
- Response time to the user;
- Transaction processing elapsed time;
- Communications characters transmitted and received; and
- Communications channel utilization.

FASAB Managerial Cost Accounting Standards

The Federal Accounting Standards Advisory Board (FASAB) issued an Exposure Draft on Managerial Cost Accounting Standards for the Federal Government, dated October 7, 1994. The Exposure Draft contains seven standards, each of which addresses a topic of managerial cost accounting in the Federal government.

EPA Directive 2100

EPA Directive 2100, Information Resources Policy Manual, establishes a framework for the IRM program in EPA. Chapter 4 establishes the principles and requirements that govern the planning, acquisition, development, maintenance and use of Agency software resources.

The Policy Manual assigns responsibility to Assistant Administrators, Associate Administrators, Regional Administrators, etc., for managing the software life cycle process and products within their programs. Chapter 4 states that the EPA software management program is needed to manage and protect EPA information as a valuable national resource, as well as improve the quality, uniformity and maintenance of software products.

The Policy Manual requires that program officials adhere to FIPS and guidelines as published or adapted for the Agency in developing, documenting, maintaining and using software applications. In addition, Chapter 17 of the policy establishes the life cycle

requirements for EPA's automated information application systems. The chapter: (1) recognizes "maintenance" as a major stage in the life cycle; (2) identifies key documents which should be produced during the cycle; and (3) identifies many FIPS publications which should be followed in order to establish standards and procedures for maintenance activities.

Appendix B of the Manual reinforces the fact that FIPS are mandatory for each Federal agency, and identifies FIPS 106 as one of the authorities on which the policy is based. EPA Directive 2100 also requires that EPA program officials periodically review all software resources to determine and prevent obsolescence of software.

EPA System Design And Development Guidance, Volume B

Chapter 3 of the "EPA System Design & Development Guidance, Volume B," dated June 1989, requires a life cycle benefit-cost analysis. This document was formally issued as a temporary EPA directive in April 1993. Costs to be included in the cost estimate include: (1) *non-recurring* costs such as site modifications, equipment, data communications, software purchase, database development, software development, studies, data conversion, procurement, training, system tests, and management overhead; (2) *recurring costs* such as personnel, maintenance and lease of equipment, space occupancy, supplies and utilities, timesharing, communications, software maintenance, training and security; and (3) *qualitative costs* such as operational disruptions, reduced employee morale and degraded organizational image. This chapter also states that "cost estimates must be supported by a reasonably accurate projection of workload and capacity requirements. Specific workload data and associated capacity requirements for each year in the process life must be provided".

EPA Operations And Maintenance Manual

The EPA Operations and Maintenance Manual, dated April 1990, states that "proposed system modifications are subject to the life cycle benefit-cost analysis techniques described in the EPA System Design and Development Guidance, Volume B. These documents were formally issued as a temporary EPA directive in April 1993. Functional maintenance changes in particular must be thoroughly analyzed because they are optional in the sense that failure to implement them will not adversely affect system performance, as with corrective and adaptive maintenance changes. Attention should be paid to assessing the benefits of functional changes, since these benefits may be either small or large in relation to the cost of implementation. Because corrective and adaptive maintenance are not optional, benefit-cost analysis is most appropriately used to determine the

best option for applying required changes. The depth and formality of the benefit-cost analysis should be determined by the size of the system and the complexity of the proposed modifications."

OARM Memorandum

In 1991, OARM's Financial Reports and Analysis Branch issued a memorandum addressed to the Financial Management Officers entitled "Reconciliation and Verification of Capitalized Equipment with Property Management Officers and Accountable Officers." This memorandum includes guidelines on capitalization of equipment and states that "this policy also applies to ADP software (programs, routines or subroutines) valued at $5,000 or more, with a useful life of two years or greater." This was a re-issuance of a 1989 memorandum stating the same requirements. These memoranda were issued in response to a 1988 OIG audit report (#P5EH8-11-0030-81917), entitled "Obligations and Disbursements of the Hazardous Substance Superfund for the fiscal year Ended September 30, 1987" which recommended that OARM establish an Agency policy for capitalizing software.

Industry Practice Regarding Software Measurement

Measurement is the basis for management, and the basis for quality improvements. Software measurement has matured dramatically, since the concept was introduced as a management tool during the 1970s. In recent years, throughout the U.S., Europe, and the Far East, numerous special projects have focused on the development and use of better software measurements and metrics. In addition, world-wide interest in TQM and customer satisfaction programs has heightened management's interest in measurements in general and software measurements in particular.

Recognized industry publications on the topic of software measurement emphasize the fact that useful measures support effective analysis and decision making processes. At the 1993 Conference on Software Quality and Productivity, quality problems were attributed to the lack of measurement.

> *Organizations that experience quality problems usually do not have a reliable system of quality measurement. If they do have such a system, it's the first thing to be sacrificed when the pressure mounts. The lack of measurement defends poor management. It's easier to deny the existence of poor quality when no measurements are made of the quality of the work in progress.*

APPENDIX IV

> *With quantifiable measures of quality, it will be possible to assess whether a project is realistically on time and within cost, since measurements of the software can be taken at various points within the development process. The post-implementation software support community is the source of maintenance cost and time data, but especially the data necessary to evaluate both quality and process models. Realistic measures of software quality are not determined at test, but during the first six months to a year of operation.*[39]

Attendees at the 1991 Applications of Software Measurement Conference were surveyed on the use of 65 commonly cited measurements.[40] The measures in the survey included anything appearing in the literature as a recommended or suggested measurement. Respondents were asked to indicate the usage and perceived value for each of the 65 measurements. The most valuable measurements for software maintenance were reported as:

- Customer or user satisfaction;
- Cost to maintain;
- Number of defects found after release; and
- Operational reliability.

Measurements help provide insight about a monitored activity; and from that knowledge, goals and targets can be set and the monitored process can be improved or changed. Software metrics are particularly easy to collect, since measurements can be automatically collected as developed or changed software code modules pass through configuration management control points during their migration to production libraries.

Measurements must be relevant and readily used by managers in order to improve their understanding of change control problem areas and facilitate the decision-making process.

---

[39] Lloyd K. Moseman, II, Deputy assistant Secretary of the Air Force (Communications, Computers, and Logistics). Proceedings, Ninth Annual Joint Conference and Tutorial of Software Quality and Productivity, March 2, 1993

[40] Hetzel, Bill. Making Software Measurement Work, QED Publishing Group, 1993

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX V

AGENCY CRITERIA AND APPLICABLE SOFTWARE MAINTENANCE SERVICES

Each of the application systems reviewed were subject to the overall provisions of OARM's Operations and Maintenance Manual which is part of the OIRM System Design and Development Guidance. The extent of additional guidance used to regulate individual program office procedures for processing software changes varied significantly between the ten application systems reviewed.

Nine of the eleven application systems reviewed relied on contractor personnel to perform software maintenance. Of these systems, CPS and STORET were the only application systems which relied on EPA employees to perform software maintenance. Change control practices for STORET were not reviewed, as stated in Appendix III. Software maintenance for five of the application systems was handled by Science Applications International Corporation (SAIC), which processed and tested software changes according to established EPA SDC policies and guidelines. The SDC has implemented numerous formal policies and procedures which are based on "state of the art" industry practices to control software maintenance activities. Most of the program offices responsible for the application systems serviced by SAIC, relied on the contractor's performance of these formal SDC procedures to provide adequate controls and oversight to the change control process.

TRIS was managed by Computer Based Systems, Inc. (CBSI), and subjected to a separate set of policies on software configuration management, quality assurance, and test and evaluation development. Likewise, PCS relied on its contractor, VIAR, to perform software maintenance following procedures outlined in the PCS Test and Acceptance Procedures Guide.

Both the AIRS and IFMS used more than one contractor to perform software maintenance. The four subsystems under AIRS used a combination of services from Martin Marietta and TRC Environmental. Software maintenance activities were governed solely by procedures outlined in the contractual scopes-of-work and administered by in-house subsystem personnel. However, no supplemental written procedures addressed change control processes prior to release of a proposed change to the contractor.

Software maintenance for IFMS was performed by both American Management Systems, Inc. (AMS) and Computer Science Corporation (CSC), with service dependent on the particular software modules. In addition to the SDC procedures, IFMS management depended on FMD and

APPENDIX V

ASD Software Development Life Cycle (SDLC) policies to define acceptable software maintenance processes. Also, IFMS relied on its prototype CMS to control software maintenance processing.

Several cognizant program offices developed their own application policies and procedures to supplement the general guidance outlined in the O&M Manual or the applicable contractor's software development procedures. These additional policies and procedures provided those system managers with some level of assurance that changes to software source code were processed in a consistent fashion. The supplemental change control policies developed for the CERCLIS, RCRIS, and CPS followed very structured processes for reviewing and implementing software changes.

Diversity was also present with regard to how software changes were tracked and managed in the ten application systems. AIRS demonstrated a completely manual change control process which was comprised of four separate subsystems. Although some portions of the AIRS change control process were centralized, the initial recording and screening of change requests was decentralized. The other nine systems used centralized control boards, with varying degrees of management involvement, to review and approve requests for change.

In addition, the ten systems reviewed used different standards for performing tests on software maintenance changes. Depending on the system, different types and levels of testing were required. Also, requirements for test documentation were diverse and the review and approval of test results varied from system to system. Some application systems would test software changes against the entire system, while others only tested the functionality of the changed portions of the code.

Another aspect of inconsistency was the type and degree of change control information tracked by each system. These methods ranged from a manual system to a combination of automated systems. Two application systems, FINDS and GICS, did not possess a separate means for identifying, recording, and tracking software maintenance requests.

APPENDIX VI

SYNOPSIS OF AUDIT FINDINGS BY APPLICATION SYSTEM

The following charts provide a synopsis of the audit findings which relate to change control and configuration management control topics. Each row corresponds directly to a condition noted in Chapter 4 of the report and, therefore, is stated as a control deficiency. The charts provide a quick reference for system managers, in order to pinpoint areas where controls may be weak, thereby facilitating their review of these software maintenance activities.

To provide this simplified overview, a subjective classification was made regarding the level of controls displayed in each audit area. As stated in Chapter 4, our review disclosed that the type and extent of controls varied greatly among the ten application systems. For each condition, only those applications which clearly met either the industry and Federal guidelines used for audit evaluation purposes, or Agency policies or the formal policies established for the individual application system, were denoted as having adequate controls. Adequate controls were represented by an "N."

In many instances, an application would demonstrate some degree of control, although various control features would not be present. In cases where a substantial measure of control was displayed, a "P" was used to denote partially adequate controls. For example, Agency or individual policies and procedures may have substantially defined a particular management control, but our fieldwork may have determined that the formal procedures were not followed.

A "Y" was used to designate a serious lack of controls. In these cases, either: (1) no management controls existed; (2) the degree of control was minimal; or (3) actual change control procedures did not follow the formal standards and policies which had been established for that application system by program office management or by the Agency.

APPENDIX VI

SYNOPSIS OF AUDIT FINDINGS BY APPLICATION SYSTEM

CONDITIONS:	AIRS	CERCLIS	CPS	EPAYS	FINDS	GICS	IFMS	PCS	RCRIS	TRIS
Lack Of Measurement Performance Indicators	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Absence Of Standardized Change Control Request Form	N	N	N	P	Y	Y	N	P	P	N
Absence Of Standard Classification Of Requests For Change	P	Y	P	P	Y	Y	P	P	P	P
Absence Of Centralized Change Control Review	P	N	N	P	Y	Y	P	P	N	P
Inadequate Historical Review Process	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Inadequate Coding Standards And Reviews	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Y = Controls Are INADEQUATE - Condition Exists
N = Controls are ADEQUATE
P = Partially INADEQUATE Controls - Condition Exists

APPENDIX VI

SYNOPSIS OF AUDIT FINDINGS BY APPLICATION SYSTEM

CONDITIONS:	AIRS	CERCLIS	CPS	EPAYS	FINDS	GICS	IFMS	PCS	RCRIS	TRIS
Inadequate Test Plans And Analysis Of Test Results	P	N/A	N/A	N/A	N/A	N/A	P	N/A	P	P
Lack Of Sufficient Testing And Acceptance By Functional Users	P	P	Y	P	Y	Y	P	N	N	P
Lack Of Verification And Validation Testing	Y	P	Y	P	P	P	P	P	P	P
Lack of Scheduled Maintenance Plans	Y	N	N	Y	Y	Y	P	N	N	P
Lack Of Version Control In Software Configuration	Y	Y	Y	Y	Y	Y	N	Y	Y	Y

Y = Controls Are INADEQUATE - Condition Exists
N = Controls are ADEQUATE
P = Partially INADEQUATE Controls - Condition Exists

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX VII

BENEFITS OF SOFTWARE CONFIGURATION MANAGEMENT PACKAGES

A quality automated SCM product should possess many desirable features. Some of the principle properties of a suitable SCM product are:

- ABILITY TO WORK WITHIN PRE-EXISTING ENVIRONMENTS

  An SCM tool must:

  - be able to interface with library management systems and job scheduling systems, thereby, allowing pre-existing control methodologies to remain in place if they have been performing well;
  - enforce the segregation of development, testing, QA, and production environments;
  - interface with standard security packages

- EASE OF USE

  To facilitate ease of use:

  - inventory and change information should be entered only one time and be easily performed
  - relationships between system components should be easily established and maintained
  - on-line query abilities should be able to:

    1) determine component package relationships
    2) determine which versions of each component were in production at any given point in time

- RETAIN AND ENHANCE PACKAGE OR COMPONENT INTEGRITY

  An SCM tool must be able to document all steps of the migration process for each component of the application system, as well as:

- allow for and yet control concurrent development of multiple versions of the same software component. This capability should include warnings to developers of these concurrent changes and limit programmers to emergency changes when appropriate;

- include an approval process as part of package implementation which includes the ability to "freeze" a package and its components, thereby preventing them from being modified again while awaiting approval;

- possess the means for identifying any failed batch migration processes and backing out of a failed package

- possess an override or separate approval process for emergency fixes, and

- ensure that software components, once modified and approved, are impervious to additional unwarranted changes prior to implementation.

APPENDIX VIII

GLOSSARY[41]

ABEND	-	Abnormal Job End
ADP	-	Automated Data Processing
ANSI	-	American National Standards Institute
APMB	-	Architectural Planning and Management Branch
ASD	-	Administrative Systems Division
BAA	-	Business Area Analysis
CFO	-	Chief Financial Officer
CMS	-	Change Management System
CPMS	-	Centralized Problem Management System
CSIP	-	Computer Systems Integrity Project
DSO	-	Designated Senior Official
ETADS	-	Emergency Technical Assistance Document
FASAB	-	Federal Accounting Standards Advisory Board
FIMAS	-	Facility Impact Monitoring and Analysis System
FIPS	-	Federal Information Processing Standards
FMD	-	Financial Management Division
FRAB	-	Financial Reports and Analysis Branch
FSB	-	Financial Systems Branch
FSS	-	Financial Systems Section
FTE	-	Full Time Equivalent
GPRA	-	Government Performance and Results Act

[41] This glossary includes all acronyms defined in the body of the report, except those associated with EPA Program Offices and Major Information Systems.

APPENDIX VIII

IEEE	-	Institute of Electrical and Electronic Engineers
IPSO	-	Information Processing Service Organization
IRM	-	Information Resources Management
ISP	-	Integrated Strategic Plan
JCL	-	Job Control Language
KLOC	-	Thousand Lines of Code
MICS	-	MVS Integrated Control System
NBS	-	National Bureau of Standards
NDPD	-	National Data Processing Division
NPDES	-	National Pollution Discharge Elimination System
NPR	-	National Performance Review
OMB	-	Office of Management and Budget
OSA	-	On-Site Assistance
PCIE	-	President's Council on Integrity and Efficiency
QA	-	Quality Assurance
QI	-	Quality Improvement
RACF	-	Resource Access Control Facility
RTP	-	Research Triangle Park
SCM	-	Software Configuration Management
SDC	-	System Development Center
SDLC	-	System Development Life Cycle
SSB	-	System Support Branch
SVVP	-	Software Verification and Validation Plan
TC	-	Technical Consultant

APPENDIX VIII

TIR	-	Test Incident Report
TQM	-	Total Quality Management
TSSMS	-	Time Share Services Management System
USGS	-	United States Geological Survey
WAM	-	Work Assignment Manager
WCF	-	Working Capital Fund

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX IX

REPORT DISTRIBUTION

Office of Inspector General

Inspector General (2410)

Deputy Inspector General (2410)

EPA Headquarters

Assistant Administrator for Administration and Resources Management (3101)

Assistant Administrator for Policy, Planning, and Evaluation (2111)

Assistant Administrator for Enforcement and Compliance Assurance (2221)

Office of General Counsel (2310)

Assistant Administrator for Water (4101)

Assistant Administrator for Solid Waste and Emergency Response (5101)

Assistant Administrator for Air and Radiation (6101)

Assistant Administrator for Prevention, Pesticides and Toxic Substances (7101)

Assistant Administrator for Research and Development (8101)

Associate Administrator for Regional Operations & State/Local Relations (1501)

Associate Administrator for Congressional and Legislative Affairs (1301)

Associate Administrator for Communications, Education and Public Affairs (1701)

Comptroller, Office of the Comptroller (3301)

APPENDIX IX

Director, Office of Information Resources Management (3401)

Director, Financial Management Division (3303F)

Agency Followup Official (3101)
Attn: Assistant Administrator for Administration and Resources Management

Agency Followup Coordinator (3304)
Attn: Director, Resources Management Division

Audit Followup Coordinator (3102)
Attn: Program & Policy Coordination Office

EPA Headquarters Library

Regional Offices

Regional Administrator, Region 1

Regional Administrator, Region 2

Regional Administrator, Region 3

Regional Administrator, Region 4

Regional Administrator, Region 5

Regional Administrator, Region 6

Regional Administrator, Region 7

Regional Administrator, Region 8

Regional Administrator, Region 9

Regional Administrator, Region 10

Research Triangle Park, North Carolina

Director, Office of Administration and Resources Management (MD-20)

Director, National Data Processing Division/OARM (MD-34)