ornl

ORNL/TM-11327

OAK RIDGE
NATIONAL
LABORATORY

MARTIN MARIETTA

A Study of Auditing Methodologies for the Energy Information Administration Data Collection and Processing System Quality Assessment Program

L. F. Truett
T. G. Yow

OAK RIDGE NATIONAL LABORATORY
CENTRAL RESEARCH LIBRARY
CIRCULATION SECTION
4500N ROOM 175
LIBRARY LOAN COPY
DO NOT TRANSFER TO ANOTHER PERSON
If you wish someone else to see this report, send in name with report and the library will arrange a loan.

OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

This report [illegible]

Available [illegible] and Technical Information Service, [illegible]
5285 Port Royal Road, [illegible]

Available [illegible]
Department of [illegible]

This report was [illegible] the United States [illegible] agency [illegible] nor any [illegible] employees, [illegible] makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL/TM-11327

Energy Division

A STUDY OF AUDITING METHODOLOGIES FOR THE ENERGY INFORMATION ADMINISTRATION DATA COLLECTION AND PROCESSING SYSTEM QUALITY ASSESSMENT PROGRAM

L. F. Truett
T. G. Yow

Date Published -- November 1989

Prepared for the
ENERGY INFORMATION ADMINISTRATION
U.S. DEPARTMENT OF ENERGY

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831
operated by
MARTIN MARIETTA ENERGY SYSTEMS, INC.
for the
U.S. DEPARTMENT OF ENERGY
under Contract No. DE-AC05-84OR21400

MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES

3 4456 0328594 1

CONTENTS

LIST OF FIGURES v

ABSTRACT vii

1. INTRODUCTION 1
   1.1 BACKGROUND 1
   1.2 PURPOSE AND SCOPE OF THIS REPORT 2
   1.3 PHILOSOPHY OF AUDITING 3
      1.3.1 Overview 3
      1.3.2 System Quality Assurance Program 4
      1.3.3 The Auditing Team 4
      1.3.4 Types of Audits 5
      1.3.5 Steps in the Audit Cycle 6
      1.3.6 Software Auditing 7

2. QUALITY EVALUATION 9
   2.1 AUDITING METHODS 10
      2.1.1 Interview, Walkthrough, and Manual Spot-Checks 10
      2.1.2 Checklist 11
      2.1.3 Software Metrics 12
      2.1.4 Data User Survey 17
   2.2 AUDIT APPROACH 17
      2.2.1 System Perspective 18
      2.2.2 Data Perspective 22
      2.2.3 Data User Perspective 26
   2.3 AUDIT MATRIX 27

3. RECOMMENDATIONS 31

4. SUMMARY 41

REFERENCES 43

LIST OF FIGURES

Figure 1. Audit Matrix v

ABSTRACT

Since FY83 the Oak Ridge National Laboratory (ORNL) has assisted the Department of Energy's (DOE's) Energy Information Administration (EIA) Quality Assurance Division, Office of Statistical Standards (OSS), with assessments of the quality of EIA's automated data collection and processing systems. These assessments have had three objectives: (1) to determine whether the survey systems collect, process, and publish accurate and reliable information; (2) to ensure that the system documentation is current and adequate; and (3) to evaluate the systems' compliance with applicable EIA standards. By the end of CY89 over 40 different EIA data survey systems will have been audited. This will complete the first round of EIA auditing.

As part of this EIA OSS quality audit tasking, ORNL was asked to review some current auditing methodologies and to develop recommendations for alternative techniques that might be used in the second round of EIA quality audits. Based on a survey of current quality assurance literature, interviews with personnel involved in auditing, and personal experiences with the auditing process, ORNL (1) presents a brief overview of the world of quality auditing; (2) briefly discusses some standard, proven audit methodologies; (3) identifies some new approaches to the auditing of data collection systems, with an emphasis on the area of software metrics; (4) identifies areas of EIA survey operations that were not thoroughly examined in the first audit cycle and that may deserve additional attention during the second round of auditing; and (5) makes specific recommendations for improvements to some techniques that were used in the first audit cycle that will be retained for the second round of audits. ORNL also presents an audit matrix to be used as a tool for evaluating which audit method(s) would provide the best results for each area to be covered in an audit.

1. INTRODUCTION

1.1 BACKGROUND

Since FY83 the Oak Ridge National Laboratory (ORNL) has assisted the Department of Energy's (DOE's) Energy Information Administration (EIA) Quality Assurance Division, Office of Statistical Standards (OSS), with assessments of the quality of EIA's automated data collection and processing systems. These assessments, part of a long-term program conducted by ORNL for EIA, have had three objectives: (1) to determine whether the survey systems collect, process, and publish accurate and reliable information; (2) to ensure that the system documentation is current and adequate; and (3) to evaluate the systems' compliance with applicable EIA standards.

The approach to these system audits has been hierarchical. Auditing has been conducted on three levels: (1) survey work-flow identification and analysis; (2) detailed evaluation of survey operations and EIA standards compliance; and (3) review of computational techniques and performance statistics and analysis of data security, confidentiality, and archival procedures. Both automated and manual procedures have been assessed. Checklists have been employed for evaluation of the system documentation, compliance of the systems with EIA standards, and quality control (QC) of general system procedures. In addition, a survey software matrix has been used to document the results of software testing. Finally, interviews have been conducted with the personnel actually operating the survey systems. A report of the results has been produced for each audit, and the EIA OSS has made recommendations for improvements to the systems based on the results documented in these reports. A log has been maintained by the EIA OSS of the recommendations and follow-up actions.

By the end of CY89, over 40 different EIA data survey systems will have been audited using this methodology. This will complete the first round of auditing.

1.2 PURPOSE AND SCOPE OF THIS REPORT

This report is an outgrowth of the EIA OSS quality audit tasking. ORNL was asked to review some current auditing methodologies and to develop recommendations for alternative techniques that might be used in the second round of EIA quality audits. As part of this tasking ORNL conducted a limited survey of quality assurance (QA) literature (i.e., selected references that address auditing methodologies and techniques appropriate to the EIA quality assessment program) and also drew on the experiences of ORNL personnel involved in auditing programs. Based on this information, ORNL has identified some new approaches to the auditing of data collection systems. ORNL has also identified areas of EIA survey operations that were not thoroughly examined in the first audit cycle and that may deserve additional attention during the second audit cycle. Finally, ORNL has made recommendations for improvements to some techniques that were used in the first round of audits that will be retained for the second round of audits.

EIA data collection systems include both manual and automated processes. Thus, the auditing methodologies examined must address both. However, it should be noted that the field of software evaluation, in particular, is a broad and sometimes controversial discipline that is growing rapidly. Therefore, the scope of this report has necessarily been limited to include only those methodologies that seemed appropriate to the EIA requirements.

The remainder of this section will briefly discuss a general philosophy of auditing. Section 2 discusses the areas that should be examined in an audit and reviews some accepted auditing methodologies. Section 3 of this report contains specific

recommendations for changes to the EIA audit program, and Sect. 4 provides a report summary.

1.3 PHILOSOPHY OF AUDITING

1.3.1 Overview

Any review of auditing methodologies must begin with a discussion of the general philosophy of auditing, a look at the underlying assumptions that guide the process of quality assessment. First, a system audit must evaluate both the adequacy of the system and its efficiency and effectiveness.[1] Although the auditing process can point out inadequacies or inaccuracies, the audit, by itself, cannot ensure product quality: "Quality cannot be inspected or tested into a product at the end of its development. Quality must be incorporated into a product from its inception, through its design, to its final creation as a tangible object" (de Jong and Trauth, p. 546). That is, product quality depends on a "quality-conscious" attitude on the part of all members of a system, each of whom must feel a personal responsibility for the quality of the final product (de Jong and Trauth, p. 549).

In the true quality-conscious environment, everyone understands quality concepts and their relationship to specific job tasks. This quality-conscious attitude also must include management's dedication and commitment to appropriate attitudes and to the pursuit of excellence. Given this understanding of quality assurance, the real purpose of the effective quality audit should be to improve, rather than to impose, product quality (de Jong and Trauth, pp. 549-551).

---

[1]In general, the word "system" denotes whatever unit is being audited.

1.3.2 System Quality Assurance Program

Before members of a system can dedicate themselves to the pursuit of quality, they must, of course, know the goal toward which they are working. Quality cannot be achieved unless the quality goals have been clearly defined and made known. According to the American National Standards Institute's Quality Assurance Program Requirements for Nuclear Facilities, the objectives of an audit revolve around a system's QA program. The audit must verify that the system has a QA plan in operation, assess its effectiveness, identify deficiencies and nonconformances, and establish follow-up procedures (Willborn, p. 7). A QA program "must emphasize a planned and systematic approach to achieve . . . quality rather than the random efforts performed by a few conscientious individuals" (Smith, pp. 6-7). This plan must ensure that there is a clear functional responsibility for quality assurance and must clearly outline the responsibilities of individual members of the system. It is the results of this systematic approach to the achievement of well-defined goals that the evaluator examines in the process of the quality assessment. If a system has no QA plan, then the audit standards must be imposed by an external source. The audit process then may result in a negative rather than a positive response from system personnel.

1.3.3 The Auditing Team

The composition of the auditing team is critical to the success of an audit. Obviously, an auditor of a system that manufactures mechanical products should feel comfortable in the machine shop. Similarly, the auditor of a software system should have a computer science background (Stewart, p. 552). For an audit of a multifaceted system (automated and manual procedures, mathematical and statistical computations, data handling, etc.), the team members must be chosen to provide related skills and understanding in all areas being reviewed. The audit team must maintain an independent, impartial attitude toward the system being audited in order to produce an audit report based on objective evidence rather than on subjective opinions (Willborn, pp. 10-14).

1.3.4 Types of Audits

If we define a "system" more exactly as a "set of related operations directed toward a central goal or objective and controlled by an integrated body of policies and procedures" (GA Technologies, p. 23), then it would seem that certain steps would be common to all system audits, whether the systems be financial, nuclear, product manufacturing, or data collection/processing. However, this is not entirely true. The specific focus of each system audit and the boundaries of that audit must be clearly defined prior to initiation of the assessment (GA Technologies, p. 27). In general, the different types of audits can be grouped into three major categories (GA Technologies, p. 25-26):

(1) The compliance audit - this type of audit addresses both system adequacy and system effectiveness and determines whether the applicable requirements are being met. Sometimes referred to as a functional audit, it includes an examination of the physical processes of a system.

(2) The trouble-shooting audit - this type of audit is designed to find specific errors in a system that is not operating properly. The audit is conducted following the identification of a problem area, which may be general or localized.

(3) The product audit - this type of audit examines a completed item that was produced and declared final by a system. In the case of a data collection/processing system, the product audit would consist of an examination of the data along the entire life cycle of the data. The audit would examine the requirements of the data item to ensure that the requirements are satisfied.

Just as the focus of a system audit must be clearly defined, the boundaries and limitations of the assessment must also be carefully determined prior to audit initialization. The audit must test clearly defined hypotheses. If the audit registers failure of the item being tested, the auditors do not investigate further to determine the full extent of the problem; that investigation must remain a separate task. It is also not the responsibility of the audit team to offer recommendations for correcting deficiencies, although the team may offer suggestions if so requested (GA Technologies, p. 27).

1.3.5 Steps in the Audit Cycle

Regardless of the type of audit being performed, there are a number of standard steps that are generally part of the audit process. These steps are as follows (GA Technologies, pp. 29-38):

(1) Prepare and plan
- define audit purpose
- define audit scope
- define hierarchical approach
- gather background information
- identify audit activities
- specify evidence to be examined
- prepare list of documentation to be examined
- prepare sampling plan
- prepare audit schedule
- determine audit checklist
- provide written audit plan

(2) Provide written and oral audit notification

(3) Conduct preaudit conference

(4) Conduct audit
- conduct formal interviews
- sample the evidence

(5) Analyze the results
- complete checklist
- organize material
- evaluate significance of findings
- formulate findings and observations

(6) Prepare formal report

(7) Conduct exit briefing

(8) Conduct follow-up activities

Of course, the steps to be performed in a particular audit should be tailored to the specific needs of that audit. Some steps may be omitted or performed in an order different from that listed above. Those involved in the initiation and performance of the audit should use their own expertise and experience to determine the steps to be performed. These steps must, of course, be made known to all involved in the audit.

1.3.6 Software Auditing

Software auditing is a relatively new and very specialized discipline in the area of QC assessments. The OSS is auditing EIA systems that use automated methods to collect large amounts of data. Therefore, software quality assurance has played a significant role in the auditing methodology employed by EIA for the past several years; it will continue to do so in the audit rounds to come. Because software QA procedures are a critical part of software development activities, much of the literature describes software audits in terms of checkpoint reviews, client acceptance audits, and benchmark audits. However, software quality assurance is also a critical part of the maintenance of a system that is already operational. When an operational software system is being assessed, the audit team should take advantage of the many automated assessment tools that are now available for software system auditing. These tools, which use objective techniques to determine such characteristics as the complexity, reliability, and maintainability of a software system, are described and evaluated in Sect. 2 of this report.

2. QUALITY EVALUATION

Most examinations of the science of auditing describe the philosophy of and the standards for evaluation, define the scope of auditing roles and responsibilities, and list the general steps in the audit process. Section 1 of this report briefly describes all of these topics with respect to the EIA's specific needs.

In this section we lay the groundwork for the specific recommendations for conducting the second round of EIA audits. These recommendations are described in Sect. 3. In Sect. 2.1 we briefly review specific auditing methodologies. Some of these are standard techniques that have already been proven effective in the world of auditing and are currently being used by EIA in its quality assessments. Others are new methodologies that may prove useful in the next round of EIA auditing.

In Sect. 2.2 we have attempted to step back from the system audits currently under way at EIA to examine the bigger picture and to gain a new perspective on the way auditing is generally performed. Of course, conducting an audit from a system perspective, as is now being done by EIA for its data collection/processing systems, is a standard auditing methodology that is universally accepted and well documented in the literature. Auditing by individual unit or system provides a wealth of information on that particular system. However, this approach does not allow a bigger picture to develop, a larger perspective that perhaps cuts across several discrete units/systems to reveal common patterns, or common problems. We believe that in addition to an audit conducted from a system perspective, there are important benefits to be gained from conducting audits across several systems or discrete units. In particular, we believe that audits conducted from the data perspective or the data user's perspective would provide valuable information to EIA on the working of all of its data collection/processing systems. Because these two

audit perspectives are not well-documented or typical approaches, this section serves to "introduce" these new audit perspectives for consideration and to provide a framework for some of the recommendations contained in Sect. 3.

In Sect. 2.3 we present a matrix developed at ORNL that identifies the areas to be covered in an audit and identifies the methodologies or techniques that can be used with each area.

2.1 AUDITING METHODS

The survey of the literature and interviews with ORNL auditing personnel have revealed that most of the techniques regarded as standard auditing procedures (e.g., the use of a standardized checklist, the audit interview) have already been carefully evaluated and put into use by EIA. Because the thrust of this report is to identify different approaches to auditing that might be useful in the second round of EIA audits, these methods will not be examined in great detail here. However, because ORNL does have some recommendations for changes in the way EIA is currently using these standard auditing techniques, these methods will be discussed briefly. New auditing methodologies that have come to light in this study are discussed in more detail. Specific recommendations for using these new methodologies or recommendations for changing the way EIA is currently using standard auditing techniques are discussed in Sect. 3.

2.1.1 Interview, Walkthrough, and Manual Spot-Checks

Because the interview, walkthrough, and manual spot-checks methodology has been successfully used during the first round of EIA quality assessments and because this is a fairly standard auditing methodology, it will not be described in this report. Only a few reminders will be noted. For example, the number of random samples that are to be spot-checked during the walkthrough should be determined in advance. If possible, a definite sample for examination should be chosen in advance (e.g., the folder for a

specific respondent for a particular time period). Although an escort is desirable throughout the walkthrough, the audit team should be free to request additional materials and to interview individuals not directly involved in the audit discussion. (See Recommendation 7 in Sect. 3.)

2.1.2 Checklist

From the Code of Federal Regulations to the American National Standards Institute, from the American Society of Mechanical Engineers to ORNL and EIA, most agencies agree that the checklist is a valuable tool in performing quality evaluations. A checklist is efficient and detailed and provides for continuity among audits of similar systems.

The audit checklist is a standard tool for objectively reporting the presence of certain properties that represent acceptable quality in the unit being audited (Martin and McClure, p. 47), whatever the definition of that unit (see Sect. 2.2 for different approaches to defining the audit unit). Of course, one criticism of the checklist is that this approach can be subjective; that is, its validity depends on the knowledge of a human evaluator who must determine whether or not the unit being audited complies or fails to comply with specified standards or whether the unit possesses or lacks certain crucial characteristics. The value of the checklist, when it is properly designed and formatted, is that it promotes consistency in the type of coverage and the depth to which the audit goes. The checklist also provides a means for ensuring continuity from one audit of a particular activity to the next round of auditing.

Problems in the use of checklists arise when there are inconsistencies or omissions in the design or format of the checklist itself, a situation that places an additional burden on the auditor. Our recent experiences in using the EIA checklist and our research into the design of standard auditing checklists have led us to formulate recommendations for changes to the design and format of the EIA checklist. Because checklists have been developed and implemented throughout this audit program, the checklists have received thorough testing and are ready for revision. (See Recommendation 1.)

2.1.3 Software Metrics

The ability to evaluate the quality of the software that drives a system is quite obviously a critical part of software QC auditing. According to Martin and McClure,

> Software developers and users have different notions of software quality . . . The inability of users, managers, and programmers to understand one another's concept of software quality and the inability to define and measure quality are two major contributors to user disappointments and the high cost of software maintenance (p. 43).

But how does an auditor objectively measure software quality? The auditor may, of course, rely on the standard quality checklist, which should contain a list of questions that test for the presence of certain program properties considered essential in a high-quality software system (Martin and McClure, p. 47). However, for large programs this type of checklist evaluation may prove too time-consuming. In addition, as Martin and McClure note, "A major criticism of the checklist approach is that it is subjective and its validity depends on the knowledge of a human evaluator. To solve this problem more objective measures have been proposed" (p. 51).

Thus, today's auditor has at his or her disposal a unique set of tools by which to judge selectively the quality of the software that is at the heart of the system. These tools are designed to measure as objectively as possible the maintainability, usability, reliability, and understandability of software programs. Known collectively as "software metrics," these tools, four of which are described individually in the following subsections, represent the new and rapidly growing discipline of software quality assurance.

Specific recommendations for the use of software metrics in the EIA auditing process are covered in Sect. 3 of this report.

2.1.3.1 Halstead's theory

Called "software science," this promising software metrics theory formulated by Halstead has been used to measure program complexity and the overall quality of existing software, to measure program reliability, to predict program length, and to estimate programming effort. The theory is based on a count of program operators (i.e., arithmetic operators, logical operators, key words, and delimiters) and operands (i.e., constants and variables) that can be computed automatically during program compilation. Software science metrics for any program can be derived from four basic counts:

n_1 = number of distinct operators in a program,

n_2 = number of distinct operands in a program,

N_1 = total number of operators in a program, and

N_2 = total number of operands in a program.

Several simple complexity algorithms have been developed to relate these counts to program properties, such as length, volume, and languages. For example, the length, N, of a program is computed by

$$N = N_1 + N_2 \ ,$$

and the vocabulary of a program is computed by

$$n = n_1 + n_2 \ .$$

Thus, N is a simple measure of program size. The larger the value of N, the more difficult the program is to understand and the more difficult the effort required to maintain it.

According to Halstead's algorithms, the simplest useful program has one input and one output, or $n_2 = 2$. The next simplest useful program has one input and two outputs or two inputs and one output, or $n_2 = 3$. Therefore, the value of n_2 can be used as a measure of program complexity. Experiments have shown that as n_2 increases, programming difficulty also increases. Results reported thus far by those who have used Halstead's theory indicate that program complexity, understandability, and maintainability can be assessed by a simple examination of a few elementary program factors such as those proposed by Halstead.

The benefits of using Halstead's software science are as follows (Martin and McClure, p. 57).

1. Software science metrics are easy to calculate and do not require an examination of statement nesting or detailed flow analysis.

2. Software science metrics are applicable to any programming language.

3. Many different statistical studies of programs from industry demonstrate the validity of software metrics as predictors of programming effort and mean number of bugs in a program.

4. When the length of large programs prohibits a line-by-line examination of source code, software science metrics can be used to pinpoint the most complex and error-prone sections of a program. Instead of evaluating an entire program, auditors can define a subset of critical or error-prone sections of a software program and concentrate on these as a representative subset of the system.

2.1.3.2 McCabe's cyclomatic number

McCabe's cyclomatic number, a software metrics theory already being used to some extent by EIA, is similar to Halstead's software science. Both theories are designed to measure program complexity, understandability, and maintainability. McCabe's strategy is to measure program complexity by computing the number of linearly independent paths through a program. This number McCabe calls the cyclomatic number.

McCabe applies this theory at the program module level, using the cyclomatic number to control the size of a program and improve its understandability by limiting the cyclomatic number (complexity) of each module in the program to a maximum of 10. Having found modules with cyclomatic numbers greater than 10 to be generally more troublesome and less reliable than those with lower cyclomatic numbers, McCabe suggests that modules with cyclomatic numbers greater than 10 be redesigned and perhaps subdivided into a group of modules.

Cyclomatic complexity can be calculated simply by counting the number of compares:

$$\text{cyclomatic complexity} = \text{compares} + 1 \quad .$$

Because the calculation is so simple, an automated tool is not necessary for evaluating the complexity of software programs. Although McCabe's theory was designed primarily to be used with FORTRAN programs, the cyclomatic complexity can be computed for programs written in any language (McCabe, pp. 308-320).

2.1.3.3 McClure's control variable complexity

McClure's tool for measuring program complexity was designed primarily to be used with COBOL programs. Although similar to McCabe's theory, McClure's approach is a more accurate measure of program complexity because it counts not only the number of compares in a program but also the number of different variables used in those compares. McClure's method for calculating the complexity of a program has three steps.

1. Compute the complexity value for each control variable (i.e., a variable whose value is used to direct path selection in a program).

2. Compute the complexity value for each module in a program.

3. Compute the complexity value for a program by summing the complexity values for all of its modules.

Thus, the complexity, $C(m)$, of a module, m, is calculated as follows:

$$C(m) = C + V \ ,$$

where C is the number of compares in the module and V is the number of control variables referenced in the module.

McClure's method can be used during the design phase or during program testing. During the design phase, the theory can be used to produce the best possible modularization scheme for a program. Such a modularization scheme should aim at increasing program understandability, testability, and maintainability by attempting to distribute complexity evenly throughout a program and avoiding what McClure calls "pockets" of complexity. McClure suggests computing an average complexity value for the modules in a program. Modules whose complexity is high or low relative to this average are then examined and perhaps redesigned. McClure also suggests using this method during program testing. Modules and variables that are the most complex are tested the most thoroughly since these portions of the program are more error-prone than other sections. Quite obviously, using this technique during software auditing can be very beneficial to the evaluator, especially when the program is too long to be examined line by line.

2.1.3.4 Bebugging

Gilb proposes a method for quantifying program reliability, a method that is based on the assumption that the number of errors removed from a program is related to the reliability of the program. "Bebugging" involves first inserting into a program a known number and type of artificial error. The program is seeded by someone other than the program testers at the rate of four errors per every 100 lines of instruction. Then

during testing, both real and seeded errors are found, and the number of seeded errors found is used to estimate the number of real errors remaining in the program. The percentage of real errors remaining in the program indicates program reliability (Gilb, pp. 26-49).

2.1.4 Data User Survey

The end user of the data published by EIA is an important part of the data collection activity. In normal software QC reviews, the user survey is an important tool. It identifies areas that are deficient, areas that are difficult to use, and unnecessary areas. However, in a software activity in which the data are the product, the user survey is not as widely employed. If a subscription list of the EIA documents were obtained, a small survey could determine the usefulness of the published data.

2.2 AUDIT APPROACH[2]

The term "quality" assessment implies more than a simple evaluation of compliance to standards and procedures; true quality is a measure of how well the unit being assessed is performing and how effectively and efficiently it is operating (Smith, p. 6-2). In defining a new perspective for assessing the quality of the EIA data collection and processing systems, the "unit" being assessed must first be defined. The most obvious choice, which was used for the first round of audits, is by survey form. This type of assessment is from a system perspective. A different viewpoint is an audit from a data perspective. A third viewpoint is an assessment of the quality from the data user's perspective.

---

[2]The issues discussed in this subsection are very closely related and are often difficult, if not impossible, to separate into distinct and independent areas of discussion. For the purposes of this report we have attempted to group these interrelated issues as logically as possible.

We believe an understanding of these new perspectives is important as EIA begins its second round of audits. A specific recommendation on audit perspectives is contained in Sect. 3. (See Recommendation 17.)

2.2.1 System Perspective

Because EIA is auditing data-dependent systems, the following questions must be answered: Is the computer system maintainable? Is configuration control maintained? Is the information that is being produced reliable? Does the system comply with national policies and regulations? Does it accomplish the established objectives? Are all manual activities appropriate? (Willborn, p. 31). We address these questions concerning system maintainability and reliability in Sect. 2.2.1.4, "Program Source Code." The question concerning regulatory compliance is addressed in Sect. 2.2.1.3, "Standards Compliance." The other questions are addressed in Sects. 2.2.1.1 and 2.2.1.2.

2.2.1.1 System requirements

To determine whether the system accomplishes the established objective, system requirements must be examined. After the original system requirements plus any changes to these requirements have been identified, the audit team must determine, using objective criteria, whether these requirements are being met. This area was not sufficiently stressed in the first round of EIA audits because much of the design and requirements documentation for the data collection surveys was unavailable to the audit team. A new evaluation of the system requirements could include data user surveys and a review of all requirements information, including system design specifications and documentation of configuration management changes. (See Recommendations 2 and 5 in Sect. 3.)

Although manual procedures (e.g., maintaining logs, microfilming) designed to meet system requirements were carefully examined during the first round of EIA auditing, one area was not sufficiently stressed to enable the audit team to determine whether the

systems being audited fulfilled the necessary requirements. The checklists employed for the EIA audits do not include an adequate number of questions concerning the physical archiving and integrity of magnetic media used for storage of data and programs. (See Recommendation 8.)

All changes that were recommended during the first round of audits should be reviewed. If the recommendations were accepted and incorporated into a system, then that system should be checked for compliance with these recommendations during the second round of audits. If the recommendations were not accepted, this fact should be noted in the system documentation with reasons for nonacceptance. (See Recommendation 14.)

2.2.1.2 System documentation

Any document that affects the performance of the system being audited should be examined by the audit team. Distribution of these documents should be regulated to ensure that those individuals responsible for particular activities receive the most up-to-date versions (GA Technologies, p. 13). The QC program should include a mechanism for ensuring that obsolete documents are destroyed.

Every system needs a written QA plan that has been approved by management. This document should be made available to all personnel operating the system and to all members of the auditing team. There should be a means of ensuring that the QA plan has been implemented and is operational. (See Recommendation 3.)

Every system needs a written training plan that has been approved by management. This document can be very brief, but the steps in the training plan should be verifiable. This plan should also be made available to the audit team. (See Recommendation 4.)

Every system needs a configuration management plan, which is used by the QA group to control and document changes to the source code, data base design, and documentation.

This written plan, which may be very brief, should be made available to the auditing team. (See Recommendation 2.)

System documentation should be readable. Use of the EIA established and standardized format is acceptable; however, duplication of information is unnecessary and misleading. Documentation should strive to be less voluminous. It is especially critical that the QA plan be readable. (See Recommendation 5.)

2.2.1.3 Standards compliance

Three types of standards exist: organization standards (e.g., the Energy Information Administration Standards Manual of April 1989), international software standards (e.g., IEEE standards), and other government standards (e.g., MIL-Specs). Conformance to established organizational and national rules and regulations was tested during the first round of audits through use of the EIA Standards Compliance Checklist. With the use of this checklist, conformance to EIA orders and standards and to Federal Information Processing Standards was determined. The Institute for Electrical and Electronic Engineers (IEEE) is in the process of publishing 17 software standards. One of these, IEEE Standard for Software Quality Assurance Plans (ANSI/IEEE Std 730-1984), is especially important for the quality auditor (Ackerman and Buckley, pp. 259-262). Two others, ANSI/IEEE Std 828-1983 and ANSI/IEEE Std 829-1983, are also relevant to the EIA QC program. Other standards should also be carefully examined, and, as appropriate, items from these standards should be covered in the EIA audit checklists. (See Recommendation 6.)

In addition, standards may include an evaluation of the economy of resource usage. Are computer resources efficiently used for both programs and data? What are the costs for contractor effort in collecting/processing the data? These issues should be carefully examined in the audit checklists or noted for inclusion during the interview. (See Recommendation 7.)

2.2.1.4 Program source code

The EIA data collection systems vary with respect to size and with respect to programming languages. Most, but not all, of the systems are located at the EIA central computing facility.

Source code style was checked during the first round of audits of the EIA data collection systems. In particular, a software compliance matrix was used to ensure that programs contained appropriate comments, used standardized naming conventions, and were consistently and appropriately formatted. Although these features were checked in the previous audit, some problems with this technique were observed. For example, the software matrix for compliance with EIA standards for programming allows for a decision of "complies" or "does not comply." In a very large program, the source code could, with only a few exceptions, comply and receive the same rating as a system that failed to comply in many areas. (See Recommendation 1.)

Maintainability implies that a system will experience changes -- changes to correct errors, to adapt to user requests or to a changing environment, and to improve efficiency. To test a system for maintainability, certain characteristics must be examined. For the EIA software systems, the most important of these are that the system be testable, modifiable, and understandable. If a system is testable, then correctness of software changes can be easily verified. Modifiability is the ease with which the code can be changed. Understandability is the ease with which we can read the software code and documentation (Martin and McClure, pp. 44-45).

To aid in determining whether the system being audited has these desirable features, software metrics can be employed. (See Recommendation 11.)

2.2.1.5 Security

Regardless of how well a system is designed or how efficiently it operates, the ultimate success of a system depends upon computer security. Computer security includes all procedures that safeguard a data processing system (hardware, software, and data) from accidental or deliberate modification or destruction. Backup and recovery processes are part of a good security policy. The first round of audits addressed both physical (e.g., locked storage and work areas) and automated (e.g., user access permissions) processes at the contractor sites. Computer security at the EIA central computer facility is controlled by the ACF2 system. However, several problems were identified during the first round of audits. For example, access to EIA-64A and EIA-23 surveys is controlled by the same access permission, and use of ADABAS and the Natural language is not controlled by ACF2. (See Recommendations 9 and 10.)

2.2.2 Data Perspective

To conduct an audit from a data perspective is to begin the audit at the end -- that is, at the level of the printed data. Thus, the areas to audit are the areas in which these data are collected, manipulated, or documented.

2.2.2.1 Determination of data requirements

EIA data are used for the development, implementation, and evaluation of legislation and of public-sector and private-sector energy policy. Because confidence in the accuracy of the data is important and because EIA is not the only government organization collecting and publishing data, comparisons of the various data series occur. Usually the requirements of the data collections are not identical. If the data requirements for two surveys are identical or result in collection of the same data, then one of the surveys should be eliminated. (See Recommendation 13.)

EIA publishes its data in several reports relating to specific topics (e.g., energy-use statistics, petroleum supply, electricity consumption). In addition, assessments of the data quality of selected EIA series are published on a less-frequent basis. These assessments state the potential for error in any data collection process and stress the goal of making data users aware of the quality of the data being published. Both types of reports (i.e., data reports and data quality assessments) are necessary. There is no assurance, however, that the data that are provided are sufficient. Perhaps other permutations or different aggregations of the data would be helpful to policy makers. (See Recommendation 12.)

2.2.2.2 Data collection

EIA collects its data through responses to data collection survey forms. Most of the forms are printed. Some of them are quite lengthy, and respondents complain about the burden of reporting. And, although the respondents are legally required to complete some of the forms, the chance of error in reporting has not been eliminated. Reporting errors could occur at two main points: while the data are being collected and transcribed onto the form by the respondent and when they are being keyed from the hardcopy form into an automated version useful to the EIA.

One way to eliminate the potential for keying errors is to provide a means for the respondent to key and transmit the data via an automated system. The EIA-23 form uses a PC-compatible floppy diskette to allow respondents to enter data. Another system, the Petroleum Electronic Data Reporting Option (PEDRO), allows the respondent to enter/exit, edit, import data from another system, and transmit data to the EIA central computer. Both of these advances eliminate a step in production of the data, thereby shortening the process, lessening the chance of error, and decreasing costs. (See Recommendation 12.)

2.2.2.3 Data processing

The EIA central computer system is an IBM mainframe shop. Data base management systems include ADABAS and DB2. Also, structures created and maintained by third-generation languages are used for data management. Several different programming languages are used in the data collection systems. It is generally true that on large data bases, a relational data base with a high-level programming language is more flexible. For example, many of the EIA programs are written in COBOL, which is not interactive and does not access a relational data base. Although ADABAS allows interactive screens, it is very slow. In addition, the security permission for using the Natural language (used with ADABAS) is not the typical ACF2 security system. (See Recommendation 10.)

Some software maintainability characteristics that define software quality are understandability, reliability, testability, modifiability, portability, efficiency, and usability. All of these characteristics must be tested when an audit from a data quality perspective is conducted. Testing can be accomplished by comparison with a benchmark or with programming standards, by running test cases, or by using software metrics. Testing for data quality includes several procedures. Every computation should be tested to ensure that data have not been mathematically altered. All statistical manipulations and sampling procedures should be verified. The software audit should review all data validation checks. If it is determined that the EIA data collection and processing systems should undergo a massive redesign, then a data dictionary, data flow diagrams, and structure charts would be helpful. Because several survey forms collect similar information, it is possible that a shared-data environment could be cost-effective. (See Recommendations 11-13 and 15.)

Configuration management includes identification of configuration items that define baseline behavior, control of these items, auditing items for compliance with the baseline, and configuration status accounting, which is maintaining a record of changes to configuration items (Shere, p. 242). An audit determines whether written plans have

been developed for configuration control. However, written plans do not ensure that a software program is under control. Many systems, particularly older systems, have evolved over the years, receiving patches and "fixes." The audit must provide enough information for management to be able to determine whether it is time to redesign and rebuild any of the systems. (See Recommendation 15.)

2.2.2.4 Published data

Because the published statistics contain only aggregated data, all programs used for the aggregation and the procedures for printing the aggregated data should be audited. Periodically, the data that have been published over time should be reviewed to ensure that the information is still appropriate. That is, let us suppose that data items A, B, and C have been published for the past ten years. Because of policy change, however, EIA believes that A is no longer needed. A user survey could determine whether or not this is true. In addition, let us say that data items D and E have never before been collected but that it is possible these two data items could be very important to current data users. A user survey could also determine whether or not this is the case. (See Recommendation 12.)

An audit of the published data could identify turning points. Depending on the scope of the audit, the auditor could test the data at the turning points to determine whether the extremes were caused by policy changes, program changes, or data collection changes or are the result of an error. The audit could also test the data in order to identify the areas in which errors may be occurring.

2.2.2.5 Security

Several issues must be addressed. The security plan at the EIA central computing facility is critical to the security of the programs and data for all of the data collection systems. Therefore, a general audit of the facility, with particular emphasis on security features such as password protection, the ACF2, permissions for particular accounts, and

data encryption would be appropriate. In addition, the physical security of the various contractor sites, the EIA central computer facility, and EIA agencies not located at the Forrestal Building should be verified. Finally, an assessment of the methods used for transmittal of information (e.g., phone lines and the postal service) should be reassessed to ensure that the potential for loss or modification of data is minimized. (See Recommendation 10.)

2.2.3 Data User Perspective

An audit from the data user's perspective examines two criteria: first, does the "product" (i.e., the published data) meet the user's needs, and, second, does the user have a reasonable degree of confidence in the data? According to the IEEE Standard for Software Quality Assurance Plans, the user should be involved during software development (p. A-3). Because EIA data users are from a wide community, encompassing both the public and private sectors, the individual participants may change over the years that any particular data collection effort is being conducted. Thus, if an audit were to be conducted from the data user's perspective, the first step would be to determine the users of a particular set of energy-related data (e.g., the community of petroleum producers, storers, transporters, or users).

The next step would be to determine how many users exist in each community. At this point, a comparison of the costs and benefits of production of the data with number of users is appropriate.

After identification of the user community, a survey could ascertain whether the users could get the data in another manner (and, if so, how) and whether the data are timely enough. The survey could also determine how difficult the data are to acquire and if the user would like to be able to log into a data base rather than depending on printed material as a source. The survey format should be a checklist, to make it easy to complete. There could be both generic and specific questions. The generic questions would result in a evaluation from the user's perspective of all EIA published data; the

specific questions would relate to the data that are pertinent to a community of users. (See Recommendation 12.)

Currently OSS publishes reports on the state of the data for selected EIA data series. These reports assess the quality of the data. If the reports were more timely, the data users might find them more useful. (See Recommendation 12.)

2.3 AUDIT MATRIX

Figure 1 is a matrix of auditing areas and methodologies. It identifies all methodologies that can be used in a QC evaluation of the units defined under each audit approach. Note that the matrix does not include audit perspectives because the specific areas audited (e.g., standards compliance) and the auditing techniques used (e.g., checklist) are essentially the same for all of the auditing perspectives examined.

FIG. 1. AUDIT MATRIX

AUDIT AREAS	METHODOLOGIES			
	CHECKLIST	INTERVIEW/ WALKTHROUGH/ SPOT CHECKS	DATA USER SURVEY	SOFTWARE METRICS
Archiving	x	x		
Backup/Recovery	x	x		
Computer Resource Usage	x	x		
Computer Security	x	x		
Configuration Management	x	x		
Data Collection Procedures	x	x		
Data Publication	x	x	x	
Data Quality	x	x	x	x
Data User Needs	x	x	x	
Data Validation	x	x	x	
Performance Statistics	x	x		x
QA Plan	x	x		
Software Maintainability	x			x

FIG. 1. AUDIT MATRIX

AUDIT AREAS	*METHODOLOGIES*			
	CHECKLIST	*INTERVIEW/ WALKTHROUGH/ SPOT CHECKS*	*DATA USER SURVEY*	*SOFTWARE METRICS*
Software Reliability	x			x
Standards Compliance	x	x		x
System Documentation	x	x		
System Requirements	x		x	
Training	x	x		

3. RECOMMENDATIONS

Based on our survey of the literature, our interviews with personnel involved in auditing, and our own experiences with the auditing process, we have formulated a number of specific recommendations for changes in or additions to the auditing process as conducted by EIA. The preceding sections of this report form the framework for these recommendations. Therefore, wherever appropriate, we have cross-referenced earlier sections of this report.

The recommendations we present cover a wide range of topics and issues. We have, as before, attempted to organize related topics into the most convenient and logical groupings possible. However, it should be remembered that these issues are very closely related and that some overlapping will occur.

RECOMMENDATION 1: CHECKLIST DESIGN AND FORMAT

Because the use of checklists requires a judgment call on the part of the auditor, every attempt should be made to provide a checklist that allows the auditor to make fine distinctions regarding compliance or noncompliance with a particular standard or characteristic. The EIA checklists are inconsistent in this regard. For example, the EIA Standards Compliance Checklist offers the auditor the option of designating "complete compliance" (rating = 1), "partial compliance" (rating = 2), "no compliance" (rating = 3), or "not applicable" (N/A). The Documentation Checklist, on the other hand, allows the designations "yes" (rating = Y), "no" (rating = N), "partially present" (rating = PP), or "not applicable" (rating = N/A). More important, the EIA Quality Audit Checklist does not allow the auditor to designate partial compliance, the options being only "yes" (rating = Y), "no" (rating = N), "not applicable" (rating = N/A), and "undetermined"

(rating = U). As discussed in Sect. 2.2.1.4, the software matrix for compliance with EIA standards for programming also limits the possible rating options to "complies" (rating = +), "does not comply" (rating = -), and "not applicable" (rating = N/A). This situation basically forces the auditor to reject a program or module that complies overall but includes some minor instances of noncompliance and to give this program or module essentially the same rating as a program or module that contains many instances of noncompliance.

Because the auditing checklist is one of the most useful tools available to the auditor, we feel that it should be designed and formatted with the goal of making the auditor's task as easy as possible. The ratings possibilities available should not be unrealistically limited. Therefore, we recommend that all checklists be modified to allow the rating of "partial compliance."

We also recommend that space be added on all EIA checklists to allow the auditor to make comments to explain a particular rating. This space for comments would be particularly useful to the auditor who wishes to express a qualified opinion or to include a disclaimer in his audit findings.

Finally, we recommend that EIA examine the possibility of changing the references to page numbers in the Documentation Checklist and the Quality Audit Checklist to references to report section numbers. Although references to page numbers are useful, the numerous text revisions that occur from the draft to the final version of a document cause page numbers to change frequently. When this occurs, the individual preparing the report must edit all checklists for possible changes in page numbers. This is a time-consuming and cost-ineffective task. On the other hand, report section numbers do not change as frequently as report page numbers, even when the document is still in draft stage, and section references are just as useful as page references in locating the material desired, especially in reports the size of those produced for EIA audits.

RECOMMENDATION 2: CONFIGURATION MANAGEMENT

As discussed in Sects. 2.2.1.1 and 2.2.1.2, a critical element in the integrity of a system is configuration management, which is used to control and document changes to the source code, data base design, and documentation. To ensure that there is a carefully defined configuration management plan and that the plan is functioning properly, questions directly related to configuration management should be added to the EIA Quality Audit Checklist. In addition, the configuration management document should be added to the Documentation Checklist used by EIA and made available for review by the audit team.

RECOMMENDATION 3: QUALITY ASSURANCE PLAN

Sections 1.3.2 and 2.2.1.2 of this report both stress the importance of a written QA plan. If there is no QA plan, then there are really no guidelines by which to judge the quality of a system. Therefore, we recommend that the EIA Quality Audit Checklist be revised to reflect questions specifically related to the existence of a QA plan and to the quality-conscious environment in the unit being audited. A written QA plan should also be made available to members of the audit team, and questions about a written QA plan should be added to the EIA Documentation Checklist.

RECOMMENDATION 4: TRAINING PLAN

As noted in Sect. 2.2.1.2, every system should have a written training plan that can be examined by the members of the audit team. The questions on the EIA Quality Audit Checklist are not sufficient to ensure that staff training is being carried out properly. Members of the audit team should be able to examine physical evidence of training techniques, including a copy of the training plan itself, training session attendance lists, training sessions, or any training handouts used by staff members in the performance of their jobs.

RECOMMENDATION 5: OTHER DOCUMENTATION ISSUES

The Documentation Checklist currently used by EIA covers many topics that must be addressed during the course of a successful audit. However, many other issues that are equally important are not addressed on this checklist. The Documentation Checklist should be revised to cover these other important topics, which are discussed in Sect. 2.2.1.2. The checklist should include questions that check to see that documentation is readable and up to date. We have already noted that a configuration management plan, a QA plan, and a training plan should be added to the Documentation Checklist and made available for review by the audit team. We also recommend that documentation relating to early system development and design specifications be required (not just recommended) documentation. It is impossible to determine whether or not a system is operating efficiently and effectively unless one knows exactly what the system was designed to do. In the first round of EIA audits, early system requirements/development documents were not made available for review by the audit team. For the most part this omission was handled by marking "undetermined" on the Documentation Checklist. This is not an adequate solution to this problem. Enforcing the requirement is the solution.

RECOMMENDATION 6: STANDARDS COMPLIANCE

Other standards as discussed in Sect. 2.2.1.3 should be examined to see if compliance with these standards and directives should be included as part of the next round of EIA audits.

RECOMMENDATION 7: INTERVIEW/WALKTHROUGH

The interview/walkthrough techniques used in the first round of EIA audits were very useful to members of the audit team. However, often the most insightful issues in an audit are those that surface during informal conversations. Therefore, we recommend that the concept of the audit interviews/walkthrough be redefined to allow members of the audit team to speak casually with personnel in the system being audited. At the same time, these personnel should be encouraged to speak honestly with auditors and to be ready to explain details about their work if they are questioned by the audit team. Auditors should be prepared to ask questions like "Who is in charge of quality assurance in your organization?" or "Are you regularly given instructions or information on quality assurance procedures?"

RECOMMENDATION 8: ARCHIVING

Many of the questions on the EIA checklists and questions posed during the formal EIA audit interviews/walkthroughs focus on archiving procedures. These questions should, of course, remain part of the next round of EIA audits. However, additional questions about archiving techniques need to be added in the next round of EIA audits. For example, during the first round of audits, archiving techniques were carefully explained to members of the audit team. However, in addition to asking the questions currently on the EIA checklists, members of the audit team should be allowed to request that any archived magnetic media be taken out of storage and run against the original source code to verify the integrity of the archived data. Auditors should ask questions like "What kind of precautions do custodians take to protect magnetic media?" or "What procedures do you take to ensure that tapes are not erased by housekeeping procedures?" These kinds of practical questions verify that data are being archived correctly and stored safely. These questions should be added to the EIA checklists to ensure that archiving procedures are adequate for the system being audited.

RECOMMENDATION 9: SYSTEM BACKUP/RECOVERY

The questions included in the EIA checklists on system backup and recovery are not adequate to ensure that the systems being audited could recover from a natural or man-made disaster, like a fire or flood or major system failure. Such questions need to be asked during interviews and walkthroughs and added to the EIA checklists (for example, "What would be the recovery procedure if the EIA computer facility were to be extensively damaged by fire?"). Because these recovery and backup procedures should be fully described and explained in the system documentation, these questions also need to be included in the EIA Documentation Checklist.

RECOMMENDATION 10: SECURITY

As noted in Sects. 2.2.1.5 and 2.2.2.5, the EIA standard procedures to ensure data security (physical and automated) should be reevaluated. Alternatives to the current system should be proposed. The security of the EIA central computer facility, EIA agencies not located at the Forrestal Building, contractor sites, the DOE mail room at the Forrestal Building, and sites used for archiving magnetic media should be studied, as should the use of public phone lines and postal services for data transmission.

RECOMMENDATION 11: SOFTWARE METRICS

Section 2.1.3 of this report contains a description of several software metrics theories that we believe would improve the EIA audits. These techniques provide a more objective way of determining the maintainability, reliability, and usability of software programs than is possible using only a standard software checklist and the standard practice of running sample data through the programs in an attempt to find programming errors. In particular, these techniques should be used when very large

systems are being audited. When large systems are audited, the size of the source code prohibits a careful line-by-line examination of the code. The alternative is to use software metrics to select a subset of modules to be examined. The sample should include the most complex of the modules in a program since these are the most error-prone. After the modules to be examined are selected, then the software matrix checklist, as revised per Recommendation 1, would be used to document the tests that are run on the selected modules.

RECOMMENDATION 12: DATA USER SURVEY

Many of the issues and concerns raised in Sect. 2 of this report concern the perspective of the data user. In particular, we have raised questions about whether the EIA systems are supplying data that remain useful to the entire community of EIA users (see Sect. 2.2.2.4). Also, we have examined the idea of surveying the data users to see if they would benefit from more automation in the reporting process (see Sect. 2.2.2.2) and to see if they would like to be able to query a data base directly to obtain nonproprietary data in new and different combinations (see Sect. 2.2.3). We recommend that EIA undertake an examination of the needs of the user community as a whole and develop a survey to be sent to members of the user community. We also suggest that users be surveyed for their opinions on the timeliness of the assessment reports of the quality of selected EIA data series. The results of this survey might indicate that changes in the way data are collected or published are in order across all EIA systems. After determining usage levels of the data, a study should examine the cost of producing the data against the value of the data to the users.

RECOMMENDATION 13: ACROSS-SYSTEM EXAMINATION

Another examination that needs to be conducted by EIA covers all EIA systems being audited. First, the checklists currently being used include a few questions about whether two survey systems are producing basically the same data. However, these questions are not specific enough to determine whether or not a particular system should be eliminated because of duplication of effort. We recommend that EIA examine this issue in the course of the next round of audits. We also recommend that EIA carefully examine the results of all of the audits conducted during the first round of auditing to see if there are common problems experienced by one or more systems. The insight gained from such an examination could help EIA in determining the focus of the next round of audits.

RECOMMENDATION 14: AUDIT FOLLOW-UP

A standard step in the audit procedure is follow-up actions (see Sect. 1.3.5). If deficiencies are found, procedures must be developed to both correct the deficiencies and check during the next round of audits to make sure that changes/recommendations have been implemented.

RECOMMENDATION 15: ROUND TWO

The next round of EIA auditing will begin soon. Before EIA begins this process, some major issues need to be examined. EIA needs to determine whether to audit exactly as before (all systems by survey form) or to choose a random sampling of systems for a more in-depth audit. EIA also needs to determine which follow-up procedures to incorporate into the second round of audits. EIA needs to examine the need for and cost-benefit of a major redesign of systems before the second round of auditing. EIA

also needs to note whether audited systems share common problems that should be examined before the next round of auditing begins.

RECOMMENDATION 16: AUDIT MATRIX

In Sect. 2.3 of this report we introduced an audit matrix developed at ORNL. This matrix represents a convenient and efficient tool for making a preliminary determination of which auditing methodologies or techniques should be used in auditing various areas. We recommend the use of this tool, which is still under development at ORNL. Note that this tool can be used regardless of the auditing perspective selected.

RECOMMENDATION 17: A NEW PERSPECTIVE

Finally, we recommend that in addition to considering the specific recommendations just described, the EIA begin now to address the possibility of conducting future audits from the new perspectives discussed in Sect. 2.2. Obviously, if one particular system audited in the first round of audits has significant problems, EIA might choose to audit that system again from a system perspective. However, we feel that much could be gained by looking at all of the EIA data collection and processing systems from a data or data user perspective in order to take full advantage of all the possibilities associated with the process of auditing.

4. SUMMARY

This report provides a brief overview of the world of quality auditing, briefly discusses auditing methodologies with the emphasis on software metrics, and provides an audit matrix as a method for evaluating which audit method(s) would provide the best results for each area to be audited.

The previous audits conducted by OSS for EIA have resulted in an evaluation of over 40 of EIA's data collection systems. The results of this first round of audits should be analyzed to determine which types of problems occurred most often. Follow-up activities should provide reassurance that the problems have been corrected. ORNL recommends that before the second round of audits begins, follow-up procedures be used to ensure that all corrections to problems identified during the first round have been implemented or, at least, initiated. The summary analysis of the audits and documentation of audit results should be produced in a report. ORNL also suggests that the recommendations discussed in Sect. 3 be carefully examined prior to the next round of auditing.

ORNL has also identified three auditing perspectives: a system perspective, a data perspective, and a data user perspective. Although the system perspective is the most common approach to auditing, each of the three approaches has certain advantages. ORNL recommends that future auditing of the EIA data collection systems include one or more of these approaches in the following order.

First, audits should be conducted from a data user perspective. Through a survey, the users of the EIA data can help EIA determine the value of its various data series. From the planning of the program, through development of the survey instrument, to compilation of the results, it is estimated that this program should take about six months.

An analysis of results could cause a major change in policy -- or no change at all, depending on the responses.

Second, audits should be conducted from a data perspective. Implications here are more far-ranging. This process includes an audit of the EIA central computing facility, its data storage processes and procedures, data quality, and security. It also involves a study of the cost-effectiveness of redesign of hardware and software architectures. A data dictionary for the data elements of any particular community of users would be desirable. The potential for a shared-data environment should be examined. It involves a study of the potential use of fourth-generation programming languages. This audit procedure would probably take at least one year to complete.

Finally, after these audits and the recommendations resulting from these audits are completed, another cycle of audits from a system perspective is in order. Perhaps some of the older systems are in need of major redesign, and the audit could help with recommendations. The system perspective methodology used in the past was efficient and effective. There is some logic to continuing to use this perspective, with more emphasis placed on those areas determined to have the most recurring problems. This round of audits would take about three months per system audit.

REFERENCES

Ackerman, A. Frank, and Buckley, Fletcher J. October 1983. "Software Standards Take Shape," pp. 259-262 in Datamation.

American National Standards Institute. 1983. IEEE Standard for Software Configuration Management Plans. ANSI/IEEE Std 828-1983. The Institute of Electrical and Electronics Engineers, Inc., New York.

American National Standards Institute. 1983. IEEE Standard for Software Test Documentation. ANSI/IEEE Std 829-1983. The Institute of Electrical and Electronics Engineers, Inc., New York.

American National Standards Institute. 1984. IEEE Standard for Software Quality Assurance Plans. ANSI/IEEE Std 730-1984. The Institute of Electrical and Electronics Engineers, Inc., New York.

De Jong, Kent, and Trauth, Sharon. May 1989. "Culture Shock: Improving Software Quality," pp. 546-563 in 43rd Annual Quality Congress. American Society for Quality Control.

GA Technologies. 1985. Quality Assurance System Audit. GA-A14460, GA Technologies, Inc., San Diego, Calif.

Gilb, T. 1977. Software Metrics. Winthrop Pub., Cambridge, Mass.

Martin, James, and McClure, Carma. 1983. Software Maintenance, The Problem and Its Solutions. Prentice-Hall, Englewood Cliffs, N.J.

Martin Marietta Energy Systems, Inc. 1986. "Plant QA Programs: Y-12 Plant Specific Application of the Martin Marietta Energy Systems Software Quality Assurance Procedure," 23 pp. in Oak Ridge Y-12 Plant Procedures.

Martin Marietta Energy Systems, Inc. 1986. "Plant QA Programs: Software Quality Audit Program," 6 pp. in Oak Ridge Y-12 Plant Procedures.

McCabe, T. December 1976. "A Complexity Measure," pp. 308-320 in IEEE Trans. Software Engineering. Vol. SE-2, No. 4.

McClure, C. 1978. Reducing COBOL Complexity Through Structured Programming. Van Nostrand Reinhold, New York.

Oak Ridge National Laboratory. 1987. "Software Quality Assurance," 17 pp. in Quality Assurance Program.

Shere, Kenneth D. 1988. Software Engineering and Management. Prentice Hall, Englewood Cliffs, N.J.

Smith, Michael W. 1988. Software Quality Assurance Audits. Software Quality International, Upland, Calif.

Stewart, Nick P. May 1989. "Software Quality Audits," pp. 552-556 in 43rd Annual Quality Congress. American Society for Quality Control.

Willborn, Walter. 1987. Audit Standards, A Comparative Analysis. American Society for Quality Control. Milwaukee.

INTERNAL DISTRIBUTION LIST

1. D.T. Bain
2. P.W. Baird
3. F.P. Baxter
4. D.J. Bjornstadt
5. R.A. Cantor
6. E.L. Dagley
7. R.G. Edwards
8. W. Fulkerson
9. C.W. Hagan
10. F.D. Hammerling
11. E.L. Hillsman
12. R.B. Honea
13. Patrick Hughes
14. Pat Hu
15. H.L. Hwang
16. R.S. Loffman
17. D.E. Reichle
18. Eleanor Rogers
19. R.M. Rush
20. R.B. Shelton
21. M.Z. Smith
22-26. L.F. Truett
27-31. T.G. Yow
32. Central Research Library
33. Laboratory Records-RC
34-36. Laboratory Records
37. ORNL Patent Office

EXTERNAL DISTRIBUTION LIST

38. Office of Assistant Manager for Energy Research and Development, Department of Energy, Oak Ridge Operations Office, Oak Ridge, TN 37831.

39-48. Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831.

49. R.L. Perrine, Professor, Engineering & Applied Sciences, Civil Engineering Department, Engineering I, Room 2006, University of California, Los Angeles, CA 90024.

50. D.E. Morrison, Professor of Sociology, Michigan State University, 201 Berkey Hall, East Lansing, MI 48824-1111.

51. B.G. Buchanan, Department of Computer Science, University of Pittsburgh, Room 318, Alumni Hall, Pittsburgh, PA 15260.

52. J.J. Cuttica, Vice President of Research & Development, Gas Research and Development, Mawr Avenue, Chicago, IL 60631.

53. Pete Armendarez, 8233 Foxall Circle, Knoxville, TN 37923.

54-58. Ramesh Dandekar, EIA/OSS, EI-74, U.S. Department of Energy, Forrestal Building, Room IE-268, 1000 Independence Avenue, SW, Washington, DC 20585.

59. M. Williams, Professor, Department of Economics, Northern Illinois University, DaKalb, IL 60115.