INEL-94/0211
Revision 2

February 1996

RECEIVED
APR 17 1996
QSTI

**Waste Management Facilities
Cost Information:
System Cost Model
Software Quality Assurance Plan**

**B. L. Peterson
A. S. Lundeen**

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Lockheed
Idaho Technologies Company

MASTER

**INEL-94/0211
Revision 2**

**Waste Management Facilities
Cost Information:
System Cost Model
Software Quality Assurance Plan**

**Barbara L. Peterson
Allen S. Lundeen**

Published February 1996

**Idaho National Engineering Laboratory
Lockheed Idaho Technologies Company
Idaho Falls, Idaho 83415**

**Prepared for the
U.S. Department of Energy
Under DOE Operations Office, Idaho
Contract DE-AC07-94ID13223**

System Cost Model Software Quality Assurance Plan

Prepared By: B. L. Peterson
B. L. Peterson, Advisory Engineer

Date: 2-28-96

Prepared By: A. S. Lundeen
A. S. Lundeen, SCM Technical Lead

Date: 2/29/96

Reviewed By: David E. Shropshire for
K. M. Hsu, SCM Work Package Manager

Date: 2/29/96

Approved By: David E. Shropshire
D. E. Shropshire, SCM Project Manager

Date: 2/29/96

Contents

Acronyms.....	vi
1. Purpose.....	1
1.1 Background.....	2
2. Reference Documents.....	3
3. Management	5
3.1 Organization	5
3.2 Tasks	6
3.3 Responsibilities.....	10
4. Documentation.....	13
4.1 Purpose	13
4.2 Minimum Documentation Requirements	13
5. Standards, Practices, Conventions, Metrics.....	17
6. Reviews and Audits	19
6.1 Software Requirements Review	19
6.2 Preliminary Design Review.....	19
6.3 Critical Design Review.....	19
6.4 Software Verification and Validation Plan Review	19
6.5 Function Audit.....	20
6.6 Physical Audit.....	20
6.7 In process Audit.....	20
6.8 Managerial Audit.....	20
6.9 Software Configuration Management Audit	20
6.10 Post-Mortem Review	20
7. Test	21
8. Problem Reporting and Corrective Action	23
9. Tools, Techniques, and Methodologies.....	25

10. Code Control.....	27
11. Media Control.....	29
12. Supplier Control.....	31
13. Record Collection, Maintenance, and Retention	33
14. Training.....	35
15. Risk Management	37
Appendix A - System Cost Model Risk Evaluation	A-1

Table

1. Responsibility Summary.....	11
--------------------------------	----

Acronyms

AEDL	Applied Engineering Development Laboratory, a LITCO branch
BEMR	Baseline Environmental Management Report
COTS	commercial-off-the-shelf
DOE	Department of Energy
DR	design review
FTP	file transfer protocol
HF	human factors
IEEE	Institute of Electrical and Electronic Engineers
INEL	Idaho National Engineering Laboratory
LLW	low level waste
LITCO	Lockheed Idaho Technologies Company
MCP	LITCO management control procedure
MLLW	mixed low level waste
MTRU	mixed transuranic waste
PPD	LITCO quality assurance program description
QA	quality assurance
QP	quality plan
SCM	system cost model
SDD	software design description
SQA	software quality assurance
SQAP	software quality assurance plan
TBA	task baseline agreement
TRU	transuranic waste
TSD	treatment, storage, and disposal
V&V	verification and validation

System Cost Model

Software Quality Assurance Plan

1. PURPOSE

In May of 1994, Lockheed Idaho Technologies Company (LITCO) in Idaho Falls, Idaho and subcontractors developed the System Cost Model (SCM) application. The SCM estimates life-cycle costs of the entire U.S. Department of Energy (DOE) complex for designing; constructing; operating; and decommissioning treatment, storage, and disposal (TSD) facilities for mixed low-level, low-level, transuranic, and mixed transuranic waste. The SCM uses parametric cost functions to estimate life-cycle costs for various treatment, storage, and disposal modules which reflect planned and existing facilities at DOE installations. In addition, SCM can model new facilities based on capacity needs over the program life cycle. The SCM also provides transportation costs for DOE wastes. Transportation costs are provided for truck and rail and include transport of contact-handled, remote-handled, and alpha (transuranic) wastes. The user can provide input data (default data is included in the SCM) including the volume and nature of waste to be managed, the time period over which the waste is to be managed, and the configuration of the waste management complex (i.e., where each installation's generated waste will be treated, stored, and disposed). Then the SCM uses parametric cost equations to estimate the costs of pre-operations (designing), construction costs, operation management, and decommissioning these waste management facilities.

For the product to be effective and useful the SCM users must have a high level of confidence in the data generated by the software model. The SCM Software Quality Assurance Plan (SQAP) is part of the overall SCM project management effort to ensure that the SCM is maintained as a quality product and can be relied on to produce viable planning data. This document defines tasks and deliverables to ensure continued product integrity, provide increased confidence in the accuracy of the data generated, and meet the LITCO's quality standards during the software maintenance phase. The activities for the SCM's retirement are defined to ensure that the product will not be used beyond its effective life cycle.

The INEL Quality Assurance Program Description (PDD-1, July 1995) defined the company's quality program. The *Software Management Procedure*, (MCP-550, Draft January 1996) details the implementation of the software portion of the Quality Assurance Program Document. MCP-550 requires that the *Institute of Electrical and Electronic Engineers Standard for Software Quality Assurance Plans* (IEEE Std. 730-1989) be followed for medium risk products, unless an alternate SQAP standard has been agreed upon with the customer and project management. This document is based on the IEEE standard and complies with existing and currently planned company standards for a quality level 2 product. The current analysis of the SCM reflects the product to be of medium risk. Appendix A documents the medium risk

determination for the SCM software product. The primary driver for this risk level is the potential impact that incorrect or inaccurate data could have to key planning processes.

The quality level determination is based on the software product's end use. The SCM quality level was established at the start of the project in accordance with the company procedures in place as of October 1994. LITCO's Quality Levels are defined as 1, 2, and 3, with 1 being the highest level, and requiring the most rigor. The company definitions are derived from DOE guidelines that have not changed since the initial determination. A quality level 2 indicates that:

- The end use of the software product and the data generated during run-time has no direct or indirect impact to human safety.
- The software end product is **not** a prototype. Note: A rapid prototype development methodology may be used for non-prototype software end products.
- A software failure **could** have a negative impact to the INEL or their customer (i.e. financial, technical credibility, future work, etc.).

The quality level 2 designation applies to all products, including software and documentation, that are used to develop or produce a product.

1.1 Background

The SCM version 1.1B software product adhered to *the Waste Management Facilities Cost Information: System Cost Model Software Quality Plan*, INEL-94/0211, November 1994. The SCM product version 1.1B was evaluated and a summary documentation *System Cost Model (SCM) Version 1.1B Evaluation*, R. K. Fink, B. L. Peterson, D. G. Wilson, March 1995. Enhancements and new features were incorporated into the software since the initial evaluation. The software changes and new product releases were controlled and documented by the LITCO SCM technical lead.

In August 1995, the SCM technical lead requested a review of the software quality assurance activities. The work was fully funded by the SCM project and was performed by the Software Process Engineering organization of the Systems Engineering Directorate. This organization has been chartered to direct the software quality activities for LITCO's Applied Engineering Development Laboratory (AEDL). The evaluation was conducted by an engineer independent of the software development and implementation effort. The evaluation determined that the SQA efforts were adequate for the software. Recommendations for improvements to the quality process were documented in an informal publication provided to the project manager and technical lead *SCM Quality Evaluation*, October 1995. One of the recommendations documented in this informal report was to revise the quality plan to reflect the current company procedures and the SCM software life cycle phase.

2. REFERENCE DOCUMENTS

IEEE Standard for Software Quality Assurance Plans, IEEE 730-1989, January 1990.

Lockheed Idaho Technologies Company Quality Assurance Program Description, PDD-1, July 1995.

Lockheed Idaho Technologies Company Management Control Procedure, Software Management, MCP-550, January, 1996 (Draft).

System Cost Model User's Manual, Version 1.3, D. Shropshire, INEL-95/0273, Revision 1, February 1996.

System Cost Model (SCM) Version 1.1B Evaluation, R. K. Fink, B. L. Peterson, and D. G. Wilson, March 1995.

Waste Management Facilities Cost Information: System Cost Model Product Description, R. K. Fink, B. L. Peterson, D. E. Shropshire, EGG-WM-11352, Rev 1.

Waste Management Facilities Cost Information: System Cost Model Quality Evaluation, B. L. Peterson, October 1995.

Waste Management Facilities Cost Information: System Cost Model Software Quality Plan, B. L. Peterson, INEL-94/0211, November 1994.

3. MANAGEMENT

The development of the SCM software product adhered to the quality plan developed in 1994 and used for the initial SCM 1.1B release. Under this plan the software was effectively developed, integrated, tested, and released. The SCM software product is in a maintenance phase. Listed below are the SQA management tasks for the maintenance phase along with detailed activities necessary to close out the maintenance phase and retire the software.

The software maintenance efforts include:

- Corrections to existing software
- Enhancements to existing features
- Development and integration of new features
- Upgrade to software platform
- Design changes to enhance performance

The Software Quality Assurance tasks for software in a maintenance phase include:

- Requirements Management
- Software Design Control
- Software methodology
- Software Change Control
- Documentation Change Control
- Configuration Management
- Testing
- Reviews

The Software Product Manager is responsible to ensure that the software maintenance adheres to the product planning documentation. The technical lead will ensure that the software maintenance and SQA tasks are implemented. A software quality expert has assisted with the development of this plan and identification of the tasks to be implemented.

3.1 Organization

The INEL's prime contractor, LITCO, has established a matrix organization, the AEDL, to provide engineering support for the diversified products and services at the INEL. The SCM project is managed by the Environmental Operations branch of LITCO. The technical lead, software support personnel, and SQA support has been matrixed to the project. The technical lead, support personnel and the SQA support are from separate company organizations and report to separate department managers and directorates. Unresolved quality issues will be brought up through management for resolution.

3.2 Tasks

The Software Quality Assurance, for the SCM product, will continue to be effectively managed by adherence to the following defined tasks.

TASK 1 *Planning:* Effective planning will maintain the quality of the software product changes and enhancements. The planning documentation may be a short statement of work, Task Baseline Agreement (TBA), subcontract, work package documentation, memo, or formal documentation based on IEEE Std. 1058.1-1987 for Software Project Management Plans. The detail and rigor of planning documentation should reflect the extent and risks associated with the software changes being implemented. It is anticipated that the standard planning documentation for software changes (corrections and enhancements) will be contracts, TBAs, or work package documentation.

Software Management Planning is developed by the project manager, work package manager, or technical lead. All plans need to be reviewed and approved by the project or work package manager. The software management plan must document or reference documentation for:

- Deliverables (documentation and software products)
- Key decision and milestones dates
- Product development methodology
- Integrated schedule information
- Cost estimate
- Change control process

TASK 2 *Requirements Management:* Well defined and documented requirements are the key to quality assurance. Requirements management is the responsibility of the technical lead. New or modified SCM requirements must be documented and under change control.

The technical lead will establish a requirements management system capable of documenting and tracking the requirements to closure. The technical lead determines the most effective tracking methodology. The minimum documentation requirements are:

- Requirement statement that is unambiguous, complete, verifiable, consistent, modifiable, and traceable.
- Source of the requirement - software change request, legacy feature, or a new feature.
- The software release number which contain the requirement implementation.
- Method of requirement verification (reviews, inspections, testing, analysis).

TASK 3

Design: If a new software requirement changes the existing software design then the design changes should be documented. The extent of this design documentation required must be defined in the planning documentation. The technical lead is responsible to ensure that design documentation is adequate for the software modifications and meets the level of documentation defined in the planning documentation.

When applicable the software modifications should take into consideration and document the following:

- Design methodology.
- Platform (hardware and software) that will be impacted by the modification(s).
- Software dependencies that could be impacted by the software modification.
- Software interfaces to existing software or other software modules under development.
- Display requirements (screen or hardcopy):
 - Will the software modifications impact current displays?
 - What impacts to existing software will need to be considered to change a display?
- Identify the software components that will be impacted and the new components that are being developed. Review the components for impact to existing software and for design maintainability.
- Design impacts of the software modifications to the existing data dictionary.

TASK 4

Software Development: The technical lead is responsible for the engineering practices to be used during development. The following must be addressed by the technical lead and be part of the planning process.

- Internal and external software documentation format
- Software configuration management process
- Software change control
- Software peer reviews process
- Software integration and release process (testing)

TASK 5

Software Configuration Management: Configuration management procedures ensure that documentation accurately reflects the product prior to release. Change control ensures the software delivered is the same software that was evaluated during integration testing and that the software tested was the same software that was provided by the developers.

The technical lead must establish a baseline software repository to control software integration and development. The software system build process is documented and implemented by the technical lead to ensure that the **current** and **correct** software is used for testing and product release. Software change control and documentation review procedures must be included as part of the Software Configuration Management.

The technical lead is responsible for software release and version control. Guidelines for version control are established to ensure consistency and understanding of software releases. The SCM software product is released with a version control identification, SCM X.xa, where X and x are numeric and a is an alpha. A new version of the SCM software product is indicated by X being assigned the current value +1 and xx going to zero, for example SCM 1.0, SCM 2.0. are version releases. The revision to the current software is indicated by .x being incremented by +.1, (e.g. 3.1, 3.2 3.3, etc.). The alpha indicator is used for an interim release usually associated with alpha or beta copies provided to users for product comment and evaluation.

The SCM documentation revisions are tracked using LITCO's document numbering system in place at the time of product release. The SCM project management in conjunction with the technical lead will review and approve the software product's associated documentation for completeness and accuracy.

TASK 6

Verification: Verification includes the activities determined by the technical lead to ensure that the requirements are met throughout the maintenance process. The technical lead will be responsible to assure the software and related software products are verified. Methods of verification may include:

- Documentation review and approval - the documentation is reviewed to verify its accuracy and completeness.
- Requirement review - concurrence that the software changes or enhancements are correct and reflect the needs of the customer.
- Design reviews - for large software modifications the design should be reviewed by technically cognizant individuals to assure that the design is an effective implementation of the requirements.
- Code reviews - the software code to be implemented is looked at by an individual other than the developer for completeness, error identification, and adherence to software standards.
- Peer review of work completed - a peer review to check for completeness and accuracy.
- Black-box testing - a non-invasive testing approach, which demonstrates that the output generated for a specific input correctly meets the requirement(s).

The rigor and extent of software and software product verification must be part of the planning process for software modifications.

TASK 7

Software Testing: The SCM software release (new or revision) must be tested against the documented requirements. A baseline test procedure is established and tracked. The test documentation is identified by the SCM software release and revision numbers. Test results are documented and maintained as part of the project records. Any deficiencies noted during testing must be documented and tracked to closure.

TASK 8

Software Release: The software distributed will be tracked and users notified of SCM software upgrades. The user will be given a copy of the SCM software and related documentation. The distribution of software and documentation released will be tracked by the project.

The **retirement phase** is initiated when the determination not to continue software maintenance is made. To ensure that the impact of discontinuing the SCM product and user support is minimal the following tasks must be completed.

Step 1 - The justification for the decision to discontinue the software support must be documented.

Step 2 - A thirty-day notification is sent to all users explaining the discontinuation of the product and product support.

Step 3 - The technical lead captures the current and at least 1 prior version of the SCM software on a magnetic media (DISK) for storage.

Step 4 - Project manager initiates record storage process in accordance with the LITCO requirements. The backups and user documentation is included in record storage for a minimum of 1 year after the software product support is terminated.

3.3 Responsibilities

The responsibilities for project tasks are assigned by the project manager. A Task Baseline Agreement (TBA) and subcontracts are used to document the assigned work and to identify the person(s) responsible to complete the task. A summary of the high level tasks is shown in Table 2. The Work Package Manager for the SCM project provides the day to day project oversight. The activities identified in Section 3.2 indicate the individual responsible for the task.

Summary of SCM Responsibilities	
Project Title	Task Summary
Project Manager	Organizational and project interfaces Long-term planning Customer point of contact Documentation approval Work authorization Project baseline development and approval Budget approval
Work Package Manager	Work and task direction and oversight Work and task status Product implementation and support
Technical Lead	Software Product design, development, implementation, release, and distribution Software Configuration Management Change control Version control and release documentation Software reviews and analysis Software testing
SQA Analyst	Software Quality Assurance Plan Software Quality Assurance reviews and evaluation SQA and Test Reports

TABLE 1. Responsibility Summary

4. DOCUMENTATION

4.1 Purpose

The IEEE standard for Software Quality Assurance Plans (IEEE Std 730-1989), that was used as a basis for this document, defines the documentation that is considered necessary to ensure that a software product meets the requirements. The SCM software product is currently in the maintenance phase and retrofitting documentation for the product would not be cost effective or provide added value. The following sections address how the documentation required by IEEE Std. 730-1989 has been met, is planned for, or the reason for non-compliance to the standard. The method for documentation review to ensure completeness, accuracy, and adequacy is part of the project's SQA tasks.

4.2 Minimum Documentation Requirements

4.2.1 Software Project Management Plan

Software Project Management documentation will be included as part of the overall project management documentation, in compliance with the current LITCO requirements. A document will be developed to identify the work planned for each new release of the SCM product. This will be used as baseline for the work to be performed. The minimum acceptable documentation is a memo or TBA. These documents will cover the scope of the planned software modifications, the associated documentation to be modified or developed, necessary resources, and an integrated schedule. The documentation may reference existing project documentation. Planning documents will be reviewed by the project manager, work package manager, and technical lead. The project manager must approve the plan for implementation.

A Software Project Management Plan would only be required if the need was determined by the project manager. If there are questions, a software quality expert should be consulted. If a software modification is such that a separate Software Project Management Plan is necessary, the plan should adhere to the IEEE standards.

4.2.2 Software Requirements Specification

The SCM Version 1.1 requirements were derived from legacy software products, customer input, and management direction. The SCM requirements were documented in the *Waste Management Facilities Cost Information: System Cost Model Product Description*, R. K. Fink, B. L. Peterson, D. E. Shropshire, EGG-WM-11352, Rev 1.

The software change log documents the software maintenance requirements. The planning documentation should identify those log items covered by a new SCM release. If the technical lead determines that additional requirements documentation is necessary a separate document may be

generated, but is not required, as long as the change log conforms to the requirements documentation defined in Section 3.2. Requirements documentation should be reviewed by the requester or customer for concurrence, management to ensure the requirement is within the overall project scope, and the technical lead to ensure feasibility.

4.2.3 *Software Design Description*

A formal software design document was not developed for SCM Version 1.1. The software documentation included a data dictionary and the imbedded software comments. The data dictionary and on-line software comments were reviewed as part of the product evaluation.

The requirements for a design document have been detailed in Section 3.2. The software design should be part of the documentation for each new release. The design document must be reviewed and concurred with by the technical lead. The review and concurrence will be documented by a signature or memo. If the software modification is large, complex, or high risk the technical lead may determine that a design review is necessary. If a design review is conducted, the review will be held in accordance with LITCO procedures.

4.2.4 *Software Verification and Validation Plan*

A Software Verification and Validation (V&V) plan is not in place. The software quality analyst determined that the quality plan was adequate for the SCM software product. The SCM software products tests and evaluations were completed and documented as stated in the *Waste Management Facilities Cost Information: System Cost Model Software Quality Plan*, B. L. Peterson, INEL-94/0211, November 1994.

A verification and validation plan is **not** considered necessary for the SCM software maintenance phase. This plan will be sufficient to ensure the product's integrity. The technical lead will include verification and testing activities in the planning documentation for a given software release. If there are concerns that software modifications could impact the product's quality, a software quality expert will review the document for completeness and applicability.

4.2.5 *Software Verification and Validation Report*

The determination not to develop a V&V plan for this product also applies to a V&V Report. However, past product reviews, tests, and evaluations of the software were documented and include:

- The SCM Version 1.1B release was tested prior to implementation. The product tests, engineering review, and the documentation reviews were summarized in *the System Cost*

Model (SCM) Version 1.1B Evaluation, R. K. Fink, B. L. Peterson, D. G. Wilson, March 1995.

- An evaluation of the ongoing SQA activities for the SCM product was conducted during August 1995. This review was summarized in the Waste Management Facilities Cost Information: System Cost Model Quality Evaluation, B. L. Peterson, October 1995.

Product evaluations and test reports must be provided for all new releases and major revisions to the software product. For minor software revisions a test summary is prepared and included in the project files. The technical lead determines the extent of testing and reporting applicable to the software product.

4.2.6 User Documentation

The SCM user guide is part of the SCM product deliverables. The user guide is reviewed for completeness and accuracy at the time of the product release. The *Waste Management Facilities Cost Information: System Cost Model User Guide* is maintained and upgraded to accurately reflect the product. The user guide is also available as part of the on-line help. The documentation review is the responsibility of the technical lead and may be delegated.

4.2.7 Software Configuration Management Plan

Appendix A of the *System Cost Model (SCM) Version 1.1B Software Quality Plan*, March 1995 documented the SCM Software Configuration Management procedures. Software change control and configuration management practices are in place and were summarized in the *Waste Management Facilities Cost Information: System Cost Model Quality Evaluation*.

5. STANDARDS, PRACTICES, CONVENTIONS AND METRICS

The LITCO company and branch procedures for software products are adhered to for the SCM software maintenance. Metrics are not required and no formal metrics are maintained by the project for the software or the development process.

6. REVIEWS AND AUDITS

Several product reviews have been defined by the IEEE standard. The SCM is in a software maintenance effort and not a software engineering task. It was the joint opinion of the software analyst and the project technical lead that these reviews would not provide sufficient added value to prove them cost effective. Each review and audit identified as required by the IEEE standard is addressed.

6.1 *Software Requirements Review*

The requirements for software modifications to be included in the SCM software release will be identified and documented in the change log, clarified for implementation by the technical lead, and reviewed by project management. Software change requests are initially reviewed by the technical lead to ensure clarity prior to entry into the SCM log. The status of software changes (correction or modifications) must be tracked. A formal requirement review is not required. If the SCM project management feels that it is necessary a review will be conducted according to LITCO's policies and procedures.

6.2 *Preliminary Design Review*

The designs for software changes will be reviewed by the technical lead or their delegate. The SCM technical lead is responsible to ensure the changes comply with the software standards. New or corrected functionality will be included as part of the integrated product test. The technical lead may choose to use peer reviews for software design or implementation reviews. The review process will be documented by a memo to the project file or by a signature page documentation. If a software design is considered complex or high risk a formal design review may be held. All design reviews will comply with LITCO's policies and procedures.

6.3 *Critical Design Review*

A critical design review will be held only if the project manager and technical lead feel that it is required to ensure product integrity. A critical design review will adhere to LITCO's policies and procedures.

6.4 *Software Verification and Validation Plan Review*

This section is not applicable. An software verification and validation plan is not required for SCM software maintenance.

6.5 *Functional Audit*

The Black Box testing recommended for the software product will be sufficient for the functional audit of the software modifications. This is part of the overall product evaluation done prior to a product release. The black box tests must demonstrate that the software performs the required functions.

6.6 *Physical Audit*

A physical audit documents that the released software, hardware, and associated documentation is current and under control. The SCM technical lead performs this audit to ensure that the software configuration management has been implemented and is effective. A summary of the audit and results is documented. An audit is conducted a minimum of once a year and may be done in conjunction with a software release. If deficiencies are found the corrective action and closure will be documented.

6.7 *In Process Audits*

The SCM technical lead will perform interim assessments of the software maintenance process to ensure conformance to the planning documentation, company procedures, design documentation, and overall product integrity. These are not formal audits. It is recommended, but not required, that a memo of these assessments are included in the project file.

6.8 *Managerial Review*

The SCM management ensures a qualified third party SQA expert to review the activities for compliance with this SQAP. A review should be conducted on the maintenance activities prior to new SCM releases. A review of the SQA activities should be conducted once a year to ensure compliance. The project manager or their delegate is responsible to ensure that the work performed adheres to LITCO's policies and procedures and track the status and quality of work.

6.9 *Software Configuration Management Review*

Software Configuration Management reviews are part of the physical audit which is reviewed for adequacy and consistent implementation.

6.10 *Post-Mortem Review*

A review is held at the end of the project to assess the development efforts and activities implemented. The lesson's learned and corrective measures for deficiencies identified will provide added value to future software efforts. This is an informal review and documentation is minimal. The key project participants should provide input into the review. A summary may be provided to interested personnel.

7. TEST

Testing provides insurance that the software changes meet the new requirements without degrading the existing system. Task 6 of Section 3.2 identifies testing requirements for the maintenance of the SCM. The technical lead is responsible for defining the extent and rigor of the SCM product testing. Software testing recommended for the SCM product releases include:

- The developer should test the software changes for **correctness** prior to including the changes in the software baseline for testing. The software deficiencies found during testing are corrected and the new software is re-tested by the developer. The technical lead will then incorporate the changes into the software baseline. This baseline contains the pre-released software for testing.
- Black box testing to demonstrate product functionality.

The developer's testing is not formal and is not required to be documented. It is recommended that the developers testing be recorded as a memo to a file or as part of a developer's log. The test documentation for black box testing conducted prior to product release include:

- Test procedures which are reviewed against the requirements for the current software product. The technical lead is responsible to modify test procedures to accurately test the current software releases.
- Test logs.
- Identification of requirements that are not verified by the test procedures and resolved.
- Deficiency summary report identifying corrective actions.
- Test report to be included with the product evaluation document.

These tasks are repeated for each software release. The rigor of the testing is based on the extent of the software changes which can range from a isolated software modification requiring minimal analysis to re-engineering requiring complete test coverage.

8. PROBLEM REPORTING AND CORRECTIVE ACTION

The software problem reports are documented in several forms including trip reports, meeting minutes, user requests, and management requests. A FoxPro database is in place to log the software problems and change requests. The SCM technical lead is responsible for the implementation and maintenance of the software tracking system. For clarity the technical lead may add additional information to the initial problem or change request. The tracking system should document:

- Description of software modification
- Requester of the modification
- Type of log entry (e.g. error, request, feature)
- Version and release effected by the change
- Test method used to verify modification
- Status of software modification

If a document is incorrect or deficient the technical lead is notified in writing. A problem report is generated that initiates the change control process for the project. A signature sheet is used to document concurrence and approval of new or revised documents.

9. TOOLS, TECHNIQUES, AND METHODOLOGIES

The SCM project does not anticipate the use of any special tools, techniques, or methods. The evaluation methods include reviews by inspection and black-box testing.

10. CODE CONTROL

The SCM software product is a PC based application. Software releases are tracked by version control, as defined in Task 5 of section 2.3. The software version release is controlled by the technical lead.

The development software is to be maintained in a separate directory from the testing baseline and product releases. Backups of the development software are the responsibility of the developer. The testing and released software will be controlled by the technical lead.

11. MEDIA CONTROL

The SCM software is designed for an IBM PC hardware platform. The recommended minimum hardware configuration is specified in the *Waste Management Facilities Cost Information: System Cost Model Product Description* and *System Cost Model User's Manual*. The SCM product is intended for a wide user group. To provide SCM to this user community the software is ported by File Transfer Protocol (FTP), electronic mail attachments, or distribution of PC disks. A windows load function is provided with the software package for loading SCM on the user's PC. Changes to the hardware platform, for the software distribution media, must be included in the software product's documentation.

12. SUPPLIER CONTROL

The SCM project uses commercial off the shelf (COTS) products from industry recognized providers. New or unproven products may not be incorporated with the SCM system, without a risk evaluation being completed. All COTS software is tested as part of the SCM black-box testing.

13. RECORDS COLLECTION, MAINTENANCE, AND RETENTION

All SCM software documentation developed in support of the SCM software is to be incorporated with the project files. The SCM tracking system is backed up and a hardcopy is included with the project. The content of this system is audited once every year for completeness, consistency, and accuracy. This review is the responsibility of the technical lead.

14. TRAINING

No additional training is required for the implementation of this SQAP.

15. RISK MANAGEMENT

A risk evaluation was conducted at the start of the SCM development effort and was repeated in August of 1995. The evaluation conducted August 1995 is included as Appendix A of this document. The evaluation identified areas requiring added rigor for effective SQA. The risk areas are to be re-evaluated at project key decision points and milestones, to ensure ongoing applicability. The risks identified for SCM software product are:

- The configuration management and control
- Software modifications causing inaccurate data to be generated

This SQAP is part of the project's risk management effort. The technical lead is tasked with identifying new areas of risks as they occur and take corrective actions.

Appendix A

System Cost Model
Risk Evaluation

Appendix A

System Cost Model Risk Evaluation

September 1995

A-1. Introduction

A risk evaluation of the SCM software product was performed to aid in the identification of the software quality assurance tasks recommended for implementation to mitigate the product risks. The risk evaluation, as documented in the SCM Software Quality Plan, was applicable to the software development phase. The criteria was based on EG&G's QP-21 Draft implementing procedures, May 1993. Since the Lockheed Idaho Technologies Company (LITCO's) revised software procedures have not been approved for implementation, as of September 1995, the EG&G QP-21 draft document was used for the risk evaluation.

The initial risk evaluation was reviewed for applicability to the current maintenance and support phase. The changes to the initial risk evaluation were noted in bold text for each criteria. The results for this re-evaluation lowered the risk for the SCM software product from high to medium.

The risk drivers identified as high were an aggressive schedule and informal integration testing. The software change control management has been shown to be effective in mitigating the impact of risk to the operational software, and still meeting the schedule. The risks that were identified during the development phase were effectively managed, and a quality product was implemented. The risk drivers identified for the SCM product have been revised for the current life cycle phase as follows.

- Configuration management and change control.
- Inaccurate data generated due to deficiency in software modifications and enhancements.

A-2. Risk Assessment

A-2.1 Risk Exposure Determination

Risk Level: The probability that the risk driver will impact the system.

Ratings:

- Low probability.
- Medium probability.
- High probability.

Medium - Medium risk level assessed as these areas were identified early and being managed at both the project and technical levels.

Impact Level: The consequences to the project should a risk materialize.

Ratings:

- I - Causes inconvenience to user.
- D - Partially disable system but can be worked around.
- S - Seriously disables the system.

Disable - This rating has been changed to disable as the software is being modified not re-designed. Effective Software Configuration management would allow the system to be re-installed at a previous release if a software change introduced inaccuracies.

Risk Exposure: The expected impact of a risk. Based on combining risk and impact levels.

Ratings:

- Negligible- Somewhat threatening.
- Marginal - Threatening.
- Severe - Very threatening.

Marginal - The results of medium risk and a disabling impact.

A-3. Risk Evaluation

A-3.1 Software Product Risk

Complexity: The system complexity relative to staff experience.

Ratings: Low - Significantly Less complex than average.
 Medium- Average complexity.
 High - Significantly more complex than average.

Medium - **No change.** The technical complexity of the system development does not rely on unproven technology and therefore does fall into a high rating.

Test Coverage: The extent of planned test coverage

Ratings: Low - Full and formal test coverage at unit, integration, and system levels
 Medium- Informal testing at unit and integration levels. Formal testing at system level.
 High - No formal testing

High - **Changed from medium.** The determination of "necessary and sufficient" test coverage will be the responsibility of the LITCO Technical Lead. This determination will be based on the extent and complexity of the software change. A record of the testing will be maintained.

Size: Size of the software system relative to the staff experience

Ratings: Low - Significantly smaller
 Medium- Average
 Large - Significantly larger majority of staff experienced with.

Medium - **No change.** The developers are experienced with software development projects of this magnitude.

Language: The language's complexity level and staff experience

Ratings: Low - Staff experienced with the high level language or language is appropriate to the application.
 Medium- Staff is in experienced with high level language or language is inappropriate to the application.
 High - Extensive assembly code required.

Low - **No change.** The staff has extensive experience with the several high level languages.

A-3.2 Software Operating Environment Risk Drivers

Development hardware

Ratings: Low - Excessive computational resources are available.
 Medium- Generous computational resources are available.
 High - Computational resources are tight or inadequate.

Low - **No change.** Resources are available to the project and staff.

Operating System

Ratings: Low - Staff is proficient with the operating system.
 Medium- Staff is familiar with the operating system.
 High - Staff has no knowledge of operating system.

Low - **No change.** The staff at LMIT and MK Environmental Services are experienced with the development environments.

Hardware Reliability

Ratings: Low - Hardware is robust.
 Medium- Hardware is somewhat robust.
 High - Hardware is unreliable or robustness unknown (new).

Low - **No change.** The current hardware identified is IBM-PC based which has an extensive history of reliability.

Hardware Accessibility

Ratings: Low - Staff has ready access to the hardware.
 Medium- Staff has periodic access to the hardware.
 High - Staff has limited or no access to the hardware.

Low - **No change.** The development personnel has access to the hardware resources needed for this development effort.

A-3.3 Software Project Risk Drivers

Development Methodology

- Ratings:
- Low - Documented methodology is followed.
 - Medium- Documented methodology is usually followed.
 - High - No documented methodology is followed or is unavailable.

Low - changed from medium. The software product has been in use for several months. The software product was developed using rapid prototype techniques. While the project did not document the methodology, the process met industry standards. The software product is now in the operational phase. Further development is being managed by change requests and integration of software product.

Team Capacity

- Ratings:
- Low - Team is experienced in the application domain.
 - Medium- Team is experienced in similar application domains.
 - High - Team has no experience in similar domains.

Low - No change. The development personnel have experience with the defined application domain (based on stand-alone PC-based environment)

Budget

- Ratings:
- Low - Budget is in excess.
 - Medium- Budget is generous.
 - High - Budget is tight or inadequate.

Medium - No change. Budgeting is adequate for the development effort.

Schedule

- Ratings:
- Low - Schedule is in excess.
 - Medium- Schedule is generous.
 - High - Schedule is tight or inadequate.

High - No change. The SCM Schedule is very aggressive.

A-3.4 Software Application Risk Driver

Clarity of Application Requirements

Ratings: Low - Requirements are complete, clear, and measurable.
 Medium- Requirements are mostly defined, clear, and measurable.
 High - Requirements are not defined, incomplete, unclear or not measurable.

Medium - **Changed from high.** Requirements were dynamic during the development process. The current environment is documenting new requirements through change control, as part of managing the technical product.

Application Interfaces

Ratings: Low - Interfaces are well defined.
 Medium- Some are not defined.
 High - None of the interfaces are defined.

Low - **Changed from medium.** The interfaces associated with the SCM software product have all been defined.

Complexity of Application: Software application complexity relative to staff experience.

Ratings: Low - complexity.
 Medium- complexity.
 High - complexity.

Medium - **No change.** The application is typical of the software applications developed by the project support personnel.

User Experience of the Application

Ratings: Low - System targeted for sophisticated computer users.
 Medium- The system is targeted for users familiar with computers, but not this type of application.
 High - Users have no experience with computers.

Medium - **No change.** Knowledgeable with IBM-PCs.

Application Reliability Requirements

Ratings: Low - Clearly documented or achievable.
 Medium- Mostly documented or achievable.
 High - No reliability documentation or not achievable.

Not Applicable - **No change.** There are no reliability requirement defined at this time

A-3.5 Software Performance Risk Factors

Complexity of Performance Requirements

Ratings: Low - Simple or easily to allocate.
 Medium- Moderate or can be allocated.
 High - Significant or difficult to allocate.

Not applicable - **No change.** Detailed performance requirements have not been established.

Size of Performance Requirements

Ratings: Low - Small or easily broken down into units.
 Medium- Medium or can be broken down into units.
 High - Large and can't be broken into units.

Not Applicable - **No change.** Detailed performance requirements have not been established.

Stability of Performance Requirements

Ratings: Low - Little or no change to established baseline.
 Medium- Some change in the baseline is expected.
 High - Rapidly changing or no baseline.

Low - **Changed from medium.** Changes to the performance baseline are not expected at this point.

Computer Resources impact to Performance

Ratings: Low - Mature, flexible, and growth capacity allowed within the design.
 Medium- Available with some growth capacity.
 High - New development, no growth capacity, or inflexible.

Low - **Changed from medium.** The SCM product has been used for several months and the computer resources have been adequate.

Personnel

Ratings: Low - Available and in place, experienced, and stable.
 Medium- Available but not in place, some experience.
 High - High turnover, little or not experience or not available.

Medium - **No change.** Personnel are in place, and experienced, but availability is a concern.

Standards

Ratings: Low - Appropriately tailored for the application.
 Medium- Some tailoring, but not all are reviewed for applicability.
 High - No tailoring is considered.

Low - **No Change.** Standards will be applied as appropriate under the direction of the Technical Lead.

Environment

Ratings: Low - Little or no impact on design.
 Medium- Some impact on design.
 High - Major impact on design.

Low - **No change.** System will be used in office environments only. No impact is anticipated on development.

Performance Envelopes

Ratings: Low - Operation well within boundaries.
 Medium- Occasional operation at boundaries.
 High - Continuous operation at boundaries.

Low - **Changed from Medium.** It was anticipated that the full system would push the operational boundaries of older PC environments. This was handled administratively by defining system hardware baseline for PCs running the SCM and was not a problem with product release.

A-3.6 Technology Risk Drivers

Language

Ratings: Low - Mature high order language.
 Medium- High order language.
 High - Significant assembly language used.

Low - No Change. A mature software development environment will be used for the SCM.

Hardware

Ratings: Low - Mature and available.
 Medium- Some development, or available.
 High - All new development.

Low - No Change. All hardware products included in the system will continue to be mature and readily available.

Tools

Ratings: Low - Documented, validated, and in place.
 Medium- Available, validated but some development required.
 High - Not validated, proprietary, or major development.

Low - No change. All software development tools are available and have complete documentation included with them.

Data Rights

Ratings: Low - Fully compatible.
 Medium- Minor incompatibilities.
 High - Incompatible.

Not applicable

Experience

Ratings: Low - Greater than four (4) years.
 Medium- Less than four (4) Years.
 High - None.

Low - No Change. Technology being used for this development effort has been available for over four years.

A-3.7 Development Approach Risk Drivers

Prototypes and reuse

Ratings: Low - Used, documented sufficiently for use.
 Medium- some use and documentation.
 High - No use and no documentation.

Low - **Changed from medium.** The SCM product has been in use and the documentation has been sufficient.

Documentation

Ratings: Low - Correct and available.
 Medium- Some deficiencies, but available.
 High - Non existent

Low - **Changed from Medium.** The documentation is available and correct.

Management approach

Ratings: Low - Existing products and process control
 Medium- Product and process controls need enhancement
 High - Weak or nonexistent

Low - **No Change.** Management is in place and LITCO project control process is being applied at the project management level.

Integration Controls

Ratings: Low - Internal and external controls are defined and in place.
 Medium- Internal or external controls are defined but not in place.
 High - Weak or non existent control.

Low - **No Change.** Integration controls have been maintained along with Software Configuration Management. This continues to be the responsibility of the Technical Lead.

A-3.8 Maintenance Software Design Risk Drivers

Design Complexity

Ratings: Low - Structural maintainable.
 Medium- Certain aspects are difficult.
 High - Extremely difficult to maintain.

Medium - While some aspects of the software model development were considered difficult the software product has been determined to be maintainable.

Documentation

Ratings: Low - Adequate.
 Medium- Some deficiencies.
 High - Inadequate.

Low - documentation has been shown to be effective.

Data Rights

Ratings: Low - Unlimited.
 Medium- Limited.
 High - Restricted.

Medium - Data generated is not intended as public information. However, there are no security or restrictions with the data at this time.

Configuration Management

Ratings: Low - Sufficient and in place.
 Medium- Some short falls.
 High - Insufficient.

Low - Configuration Management is currently in place and sufficient.

Stability

Ratings: Low - Little or no stability.
 Medium- Moderate, controlled change.
 High - Rapid or uncontrolled change.

Medium - Changes to the software product and future releases are anticipated. Controlled releases with limited distribution group and associated release notes maintained by Technical Lead mitigates this risk.

A-3.9 Maintenance Responsibility Assignments Risk Drivers

Management

Ratings: Low - Defined, assigned responsibilities.
 Medium- Some roles and missions issues.
 High - Undefined or unassigned.

Low - The project has clearly defined management roles for the release and maintenance phase of the software product.

Configuration Management

Ratings: Low - Single point control.
 Medium- Defined control points.
 High - Multiple control points.

Medium - A Control point has not been defined for the software product.

Software identification

Ratings: Low - Consistent with support organization control system.
 Medium- Some inconsistencies.
 High - Inconsistent.

Low - Software identification is being established and will continue to be maintained by the development and maintenance team. The version release control for the product has been maintained effectively by the Technical Lead.

Technical management

Ratings: Low - Consistent with operational needs.
 Medium- Some inconsistency.
 High - Inconsistent.

Low - Technical management is expected to be consistent with operational needs of the product.

Change implementation

Ratings: Low - Responsive to user needs.
 Medium- Acceptable delays.
 High - Non responsive to user needs.

Medium - It is anticipated that the software product will be maintained with acceptable delays for software changes and distribution to other DOE sites.

A-3.10 Maintenance Tools and Environment Risk Drivers

Facilities

Ratings: Low - In place and requiring little change.
 Medium- In place but requiring some modifications.
 High - Non existent or require extensive modification.

Low - There are no risks defined with the facilities for operations and support.

Software tools

Ratings: Low - Delivered, certified, and sufficient.
 Medium- Some resolvable concerns.
 High - Not delivered, certified, or sufficient.

Medium - Software tools for maintenance are in place and sufficient for the identified distribution set.

Computer hardware

Ratings: Low - Compatible with operational system.
 Medium- Minor incompatibilities.
 High - Major incompatibilities.

Low - The operational systems targeted for the software product are compatible with the development environment.

Production

Ratings: Low - Sufficient for fielded applications.
 Medium- Some capacity questions.
 High - Insufficient.

Not applicable for this software product.

Distribution

Ratings: Low - Controlled and responsive.
 Medium- Minor response concerns.
 High - Uncontrolled or non-responsive.

Low - The product distribution for this system will be controlled. The software product user group and releases will be maintained by the project.

A-3.11 Maintenance Software Supportability Risk Drivers

Changes

Ratings: Low - Within projections.
 Medium- Slight deviations.
 High - Major deviations.

Medium - The overall change requests have been within the SCM project's expectations. Adequate funding and scheduling was available.

Operational Interfaces

Ratings: Low - Defined and controlled.
 Medium- Some hidden linkages.
 High - Major deviations.

Low - Operational interfaces have been defined and are under control for the SCM project.

Personnel

Ratings: Low - In place, sufficient, and experienced.
 Medium- Minor discipline mix concerns.
 High - Significant discipline mix concerns.

Low - Experienced personnel are in place and sufficient for the maintenance of the software product.

Release cycle

Ratings: Low - Responsive to user requirements.
 Medium- Minor incompatibilities.
 High - Non responsive to user requirements.

Low - The release meet the user requirements without impact to the quality of the software product.

Procedures

Ratings: Low - In place and adequate.
 Medium- Some concerns.
 High - Non existent or inadequate.

Low - The procedure for software change and support is adequate.

A-4. Software Cost Risk Drivers

The budget is adequate for the work defined. Cost risk is not viewed as having an impact to the project at this time. Factors evaluated included requirements, personnel experience, and usage of the existing software. This list may need to be re-evaluated at a future date, as the project merges from a prototype environment to a larger development effort. **No impact to maintenance phase noted at this time.**

A-4.1 Tools and Environment

Adequate facilities and tools are in place for the SCM development effort. A detailed analysis is not required at this point in time. If this changes areas of risk would include facility, facility availability, accessibility, and meeting technical requirements. **No impact to maintenance phase noted at this time.**

A-4.2 Software Schedule Risk Drivers

Personnel

Ratings: Low - Good discipline mix and available.
 Medium- Some disciplines not available.
 High - Questionable skill mix or not available.

Medium - **No change.** A good discipline mix is in place. However, there is some risk that resources may not be available when needed, this risk needs to be managed throughout the software product's life cycle.

Facilities

Ratings: Low - Exists with little or no modification and is available.
 Medium- Exists with some modifications and is available.
 High - Non-existent, needs major modifications, or is not available.

Low - **No change.** Adequate facilities are available.

Financial

Ratings: Low - Sufficient budget allocated.
 Medium- Some questionable allocations.
 High - Budget allocations in doubt.

Low - **No change.** The SCM project has a sufficient budget for the tasks identified.

Threats to Need Dates

Ratings: Low - Verified delivery projections.
 Medium- Some unstable aspects.
 High - Dates are rapidly changing.

Medium - **No change.** There are some unstable aspects to the product need dates, however the majority of these dates are stable.

Political Sensitivity to Need Dates

Ratings: Low - Little projected sensitivity.
 Medium- Limited sensitivity.
 High - Extremely sensitive.

Medium - The risk generated by the sensitivity of the need dates is due to the anticipated end user being outside of the INEL.

Tools Impact to Need Dates

Ratings: Low - In place, available.
 Medium- Some deliverables in question.
 High - Uncertain deliverable dates.

Low - **No change.** Tool are in place and available for the current development effort. The environment is not expected to change for future development phases.

Technology Availability

Ratings: Low - In place.
 Medium- Some aspects in development.
 High - All aspects in development.

Low - **No change.** No new technology has been identified with the software development.

Technology Maturity

Ratings: Low - Application verified
 Medium- Some application verified
 High - Non verified

Low - **No change.** Only proven technologies are to be used for the software development effort at this time.

Experience with Technology

Ratings: Low - Extensive experience with technology being used.
 Medium- Some experience.
 High - No experience.

Low - **No change.** Project contributors are experienced with the technology used for the SCM Project.

Requirements Definition

Ratings: Low - Definition known and a baseline.
 Medium- Baseline in place with some unknowns.
 High - Unknown and no baseline.

Medium - **No change.** A requirement definition has been established on the prototype system and subsequent release. New or modified requirements are being tracked and logged by the Technical Lead.

Requirements Stability

Ratings: Low - Little or no changes projected.
 Medium- Controllable change projected.
 High - Rapid or uncontrollable changes projected.

Medium - **No Change.** Changes are considered to be controllable at this time and no impact to schedule is expected.

Complexity of Requirements

Ratings: Low - Compatible with existing technology.
 Medium- Some dependency on new technology.
 High - Incompatible with existing technology.

Low - **No change.** The requirements as defined are compatible with current technology.