Conf-920428--3

WSRC-MS-91-389

SOFTWARE QUALITY ASSURANCE FOR SAFETY ANALYSIS AND RISK MANAGEMENT AT THE SAVANNAH RIVER SITE (U)

WSRC-MS--91-389

DE92 009403

by

M. J. Ades[1], H. Toffer[2], and R. D. Crowe[2]

[1] Westinghouse Savannah River Company
Savannah River Site
Aiken, SC 29808

[2] Westinghouse Hanford Company
Mail Stop HO-38
P. O. Box 1970
Richland, WA 99352

A paper proposed for presentation at the
1992 Simulation MultiConference - Visualization, Validation, and Verification of Computer Simulations
Orlando, FL
April 6-9, 1992

and for publication in the proceedings

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, 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.

This article was prepared in connection with work done under Contract No. DE-AC09-89SR18035 with the U. S. Department of Energy. By acceptance of this article, the publisher and/or recipient acknowledges the U. S. Government's right to retain a nonexclusive, royalty-free license in and to any copyright covering this article, along with the right to reproduce and to authorize others to reproduce all or part of the copyrighted article.

MASTER

MAR 10 1992

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

SOFTWARE QUALITY ASSURANCE FOR SAFETY ANALYSIS AND RISK MANAGEMENT AT THE SAVANNAH RIVER SITE*

by

M. J. Ades
Westinghouse Savannah River Company
Savannah River Site
Aiken, SC 29808

H. Toffer and R. D. Crowe
Westinghouse Hanford Company
P. O. Box 1970
Richland, WA 99352

ABSTRACT

As a part of its Reactor Operations Improvement Program at the Savannah River Site (SRS), Westinghouse Savannah River Company (WSRC), in cooperation with the Westinghouse Hanford Company, has developed and implemented quality assurance for safety-related software for technical programs essential to the safety and reliability of reactor operations. More specifically, the quality assurance process involved the development and implementation of quality standards and attendant procedures based on industry software quality standards.[1,2] These procedures were then applied to computer codes used in reactor safety and probabilistic risk assessment analyses.

This paper provides a review of the major aspects of the WSRC safety-related software quality assurance. In particular, quality assurance procedures are described for the different life cycle phases of the software that include the Requirements, Software Design and Implementation, Testing and Installation, Operation and Maintenance, and Retirement Phases. For each phase, specific provisions are made to categorize the range of activities, the level of responsibilities, and the documentation needed to assure the control of the software.

The software quality assurance procedures developed and implemented are evolutionary in nature, and thus, prone to further refinements. These procedures, nevertheless, represent an effective controlling tool for the development, production, and operation of safety-related software applicable to reactor safety and probabilistic risk assessment analyses.

REFERENCES

1. "Quality Assurance Requirements of Computer Software for Nuclear Applications", ANSI/ASME NQA-2 Part 2.7, American Society of Mechanical Engineers.

2. "Standards for Software Quality Assurance Plans", ANSI/IEEE Std-730-1984, Institute of Electrical and Electronics Engineers.

---

* The information in this article was developed during the course of work done under Contract No. DE-AC09-89SR18035 with the U. S. Department of Energy.

SOFTWARE QUALITY ASSURANCE FOR SAFETY ANALYSIS AND RISK MANAGEMENT AT THE SAVANNAH RIVER SITE

by

M. J. Ades[1], H. Toffer[2], and R. D. Crowe[2]

[1] Westinghouse Savannah River Company
Savannah River Site
Aiken, SC 29808

[2] Westinghouse Hanford Company
Mail Stop HO-38
P. O. Box 1970
Richland, WA 99352

INTRODUCTION

As a part of its Reactor Operations Improvement Program at the Savannah River Site (SRS), Westinghouse Savannah River Company (WSRC), in cooperation with the Westinghouse Hanford Company, has developed and provided quality assurance for safety-related software for technical programs essential to the safety and reliability of reactor operations. More specifically, quality assurance includes the use of quality standards and attendant procedures developed for, and applied to, computer codes used in safety and probabilistic risk assessment analyses.

The need for software quality assurance stems from the diversity in the methods used in the production of software, enhanced by a rapidly changing environment in which significant software engineering expertise is being applied to a wider range of applications on a variety of computing systems. As a result, strict quality assurance procedures and guidelines based on industry software engineering standards[1,2] were developed to efficiently produce and control high-quality software that performs as expected and retains a high degree of portability.

Software quality assurance at WSRC is incorporating national standards (References 1 and 2) into a quality assurance manual serving the entire Site. Each department produces more specific procedures for their applications based on the sitewide manual. The procedures discussed in this paper are department specific and cover computer codes used for safety analyses and probabilistic risk assessment.

This paper provides a review of the major aspects of the WSRC safety-related software quality assurance. The importance of the software procedures

developed and their application to the different software life cycle phases is outlined and discussed.

SOFTWARE QUALITY ASSURANCE PROCEDURE

General

The development and operation of a computer code is governed by different phases. The software life cycles typically include a requirements phase, a design and implementation phase, a test and installation phase, an operation and maintenance phase, and a retirement phase. The number of phases and the relative emphasis placed on each phase depends on the nature and complexity of the software.

During the life cycle phases, the individuals or organization with the primary responsibility for the phase activities may be different. In general, during software development, the primary participants are the software Owner, Designer, and Maintainer. After the code development and testing has been completed, the primary responsibilities shift to the Code Proprietor and Software Configuration Management System Custodian (SCMS). Software Configuration Management for safety analysis and risk management is provided by a separate organization. Therefore, procedures developed have to be in concurrence with SCMS requirements.

The different life cycle phases and associated procedures and activities are reviewed below. Key software documentation associated with each phase is shown in Figure 1.

Software Requirements Phase

A Task Plan and a Software Development Plan (SDP) with a Software Requirements Specification are developed by the Task Leader prior to beginning software design. The SDP specifies the individuals and organizations responsible for each life cycle phase. The combination of the SDP and the Task Plan are equivalent to the Software Quality Assurance (SQA) Plan. During the requirements phase, the Owner/Task Leader determines the impact level of the software.

Software Design and Implementation Phase

During this phase, the Software Requirement Specification is the baseline that defines the desired software technical capabilities, and thus serves as the foundation for designing the software. The design, in turn, becomes the baseline for the implementation or coding of the software.

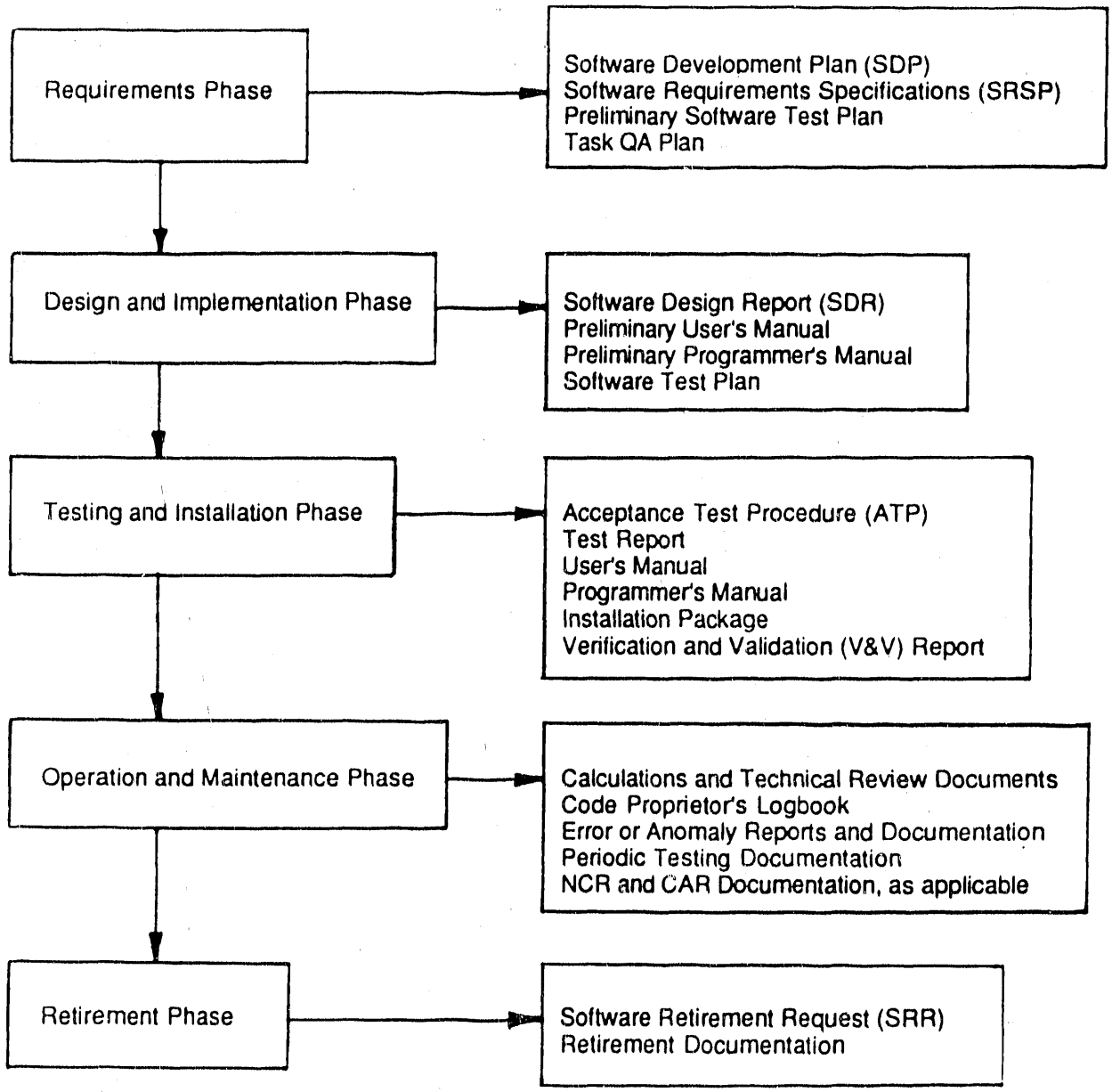

Figure1. Software Life Cycle Phases and Associated Quality Assurance Documentation.

The Designer has the primary responsibility for the following activities:

- Preparing the software design and documenting the design.
- Ensuring that the design is consistent with, and satisfies, the requirements identified in the Software Requirement Specification.
- Preparing User, Programmer, Software Test Plan, and Test Procedure documentation, as required.
- Ensuring that design and implementation documents are reviewed by an independent reviewer.

The Owner reviews and approves the software design and implementation documentation to ensure that the technical objectives of the Software Requirement Specification have been met.

Similarly, the Maintainer reviews and approves the software design and implementation documentation.

The Software Configuration Management System (SCMS) is the central system for managing and controlling computer software. The SCMS Custodian is responsible for the operation of the SCMS, which includes control and maintenance of all computer software.

One of the key documents produced during the Design and Implementation phase is the Software Design Report (SDR), prepared by the Designer. The SDR addresses the requirements identified in the Software Requirement Specification and links the design to the requirement specification. The SDR covers the following:

- Functional capabilities, mathematical models, and algorithms.
- The structure of the code including major models.
- Inputs.
- Outputs.
- Interface definitions.

The Software Test Plan is developed by the Designer to address, as applicable, the evaluation of software elements for:

- Compliance with the design requirements.
- Assessment of timing, sizing, and accuracy.
- Interfaces under normal and under error conditions.
- Assessment of software reliability and maintainability.
- Plan documentation of test tasks and results.

Functional testing is performed during software development to eliminate coding errors at the earliest opportunity and to demonstrate that the code is operational. Functional tests of modules and integration tests are also performed. This phase produces some important documentation, such as the preliminary User's Manual and Programmer's Manual.

Software Testing and Installation Phase

During this phase, the Owner:

- Approves the Acceptance Test Procedures.
- Reviews and approves documentation and changes produced during the testing phase.
- Prepares a final Verification and Validation (V&V) Report summarizing all the V&V activities carried out, describing the overall results, and presenting conclusions and recommendations concerning acceptance and utilization of the program.

The Designer:

- Prepares detailed Acceptance Test Procedures (ATP).
- Executes the test cases in stages, as specified in the Final Software Test Plan, and prepares a Test Report.
- Verifies the test results by reviewing the Test Report and outputs to ensure that the testing has been conducted accurately and is in accordance with the Test Plan.

The Maintainer:

- Approves the Software Test Plan and Acceptance Test Procedures.
- Reviews and approves all corrections of V&V reports produced during the testing phase, including the Test Report.
- Reviews, approves, and maintains an installation package, including procedures for installation, verification of installation, and a final installation phase V&V report.

An independent review, performed by individuals different from the Designer, is performed to verify the test results by evaluating the Test Report and the test outputs to ensure that the testing has been conducted in accordance with the Test Plan, and that the Test Report is an accurate description of the results obtained.

Error control during the Testing and Installation Phase (developmental computations) is the responsibility of the Designer. The Designer generates a detailed Acceptance Test Procedure (ATP) to implement the Test Plan.

The Designer is responsible for updating the User's Manual. Test analysts refer to the User's Manual during the test program and report to the Designer any deficiencies or discrepancies.

The testing is executed in three stages, as applicable:

- Integration Test Execution, which includes Integration testing performed in accordance with the Acceptance Test Procedure.
- System Test Execution, which includes a system testing performed in accordance with the Acceptance Test Procedure.
- Acceptance Test Execution, which includes an acceptance testing performed in accordance with the Acceptance Test Procedure under formal configuration control.

The Designer also generated an installation package consisting of installation procedures, installation medium (e.g., magnetic tape) containing all files necessary to install the program, selected test case data for use in verifying installation, and expected output from the test case. Once verified, this package may be used for backup. The installation package is reviewed and approved by the Owner and Maintainer.

The Maintainer then installs the program by following the installation procedure, and executes selected test cases with the installed program. The output produced by these test cases is checked against the expected output supplied by the installation package to ensure that the program is successfully installed and that it will produce the same results as the program operating in the development environment.

At the conclusion of the Test and Installation Phase, the Owner prepares the Final V&V Report. This report summarizes the V&V activities, describes the overall results, and presents conclusions and recommendations concerning acceptance and utilization of the program.

Software Operation and Maintenance Phase

In this phase the code is exposed to the user environment. Generally an individual is designated as the Proprietor for the code. He or she may be the Designer or Maintainer from the preceding phases.

Key procedural items developed for, and applied to, the Operation and Maintenance phase include:

- Use of the software within its validation range as documented in the User's manual.

- Operation of the code by authorized users only. To this effect, the Code Proprietor (or Owner) maintains and controls a list of authorized users.

- Software and system changes, where procedures are used for implementation of major software changes that alter the original code requirements and design specifications. In this situation, based on the determination of the Owner or Code Proprietor, the preceding life cycle phases may have to be repeated.

In particular, if the changes are minor and do not represent alterations to the software specifications, the corrections are implemented under the software configuration control plan. Testing is performed to determine whether the changes adversely impacted the output. The testing process and results are documented in the Code Proprietor's Logbook.

In other situations, changes to the operating environment, such as upgrades in system software or hardware, may adversely impact the output of software. In this case, the modifications in the operating environment must be checked to ensure that the performance of subroutines dependent on compilers and other system software features has not been adversely impacted. The SRSP must be verified to ensure that the program has not been affected by word size, storage allocation, or other system dependent features and requirements.

The Code Proprietor (or Owner) validates the code following changes made to the code or the operating environment. The test problems are used to test both code funtionality and validation range. If the software is sensitive to hardware performance and stability, such tests may be required on a periodic basis. Test results are documented in the Code Proprietor's Logbook.

More generally, the Code Proprietor plays a key role for software changes and control of such changes. Changes suggested by Users are directed to the Code Proprietor and implemented, as needed, by him or her. Configuration controlled source codes are made accessible only to the Code Proprietor.

A key item during the Operation and Maintenance phase consists of Error Control (Production Computations). Procedures exist for reporting errors/deviations of software problems and taking appropriate corrective action. Problems identified during the operations phase are promptly reported by the Users to the Code Proprietor and by the Code Proprietor to other Users.

Nonconformance Reports (NCRs) are filed for discrepancies between the software and its requirements specifications or for errors detected in the software during the Operation and Maintenance phase. The NCR is written either by the individual identifying the discrepancy or by the Code Proprietor. A copy of the NCR is provided to the Code Proprietor so that the software can be removed from use pending resolution of the NCR. The Code Proprietor notifies all authorized Users that an error has been discovered, the time period that the code has been affected by the error, and the effect of the error on the functionality of the code.

A Corrective Action Report (CAR) is issued to identify the nature of the discrepancy, the actions to be taken, the reviews to be performed, and the closeout requirements. The Code Proprietor ensures that the necessary changes to software are made and that supporting documentation, such as user's manuals, programmer's manual, test plans, and installation instructions, is updated, if necessary. The Code Proprietor is also responsible for ensuring that appropriate testing and benchmarking is completed.

Upon completion of the code modification and testing, and following the closeout of the CAR, the Code Proprietor notifies all the authorized Users that the software is ready to be used.

The Users identify and rerun the appropriate problems that have been affected by the error. By comparing the old and new computer outputs, the Users can estimate the magnitude of the effect of the error. This information is sent to each User's Manager and to the Code Proprietor for evaluation and review of the correction process.

The Users notify any individuals or groups whose analysis results could have been impacted by the error in the software. Any high-impact calculations performed using results from the error-containing code are reviewed to determine if these calculations should be recomputed and revisions issued to any affected reports. The recalculations are made with the same level of Quality Assurance as the original calculations, and the revised report will require the same technical review and management approval as the original report.

Another key item during the Operation and Maintenance phase includes Computer Output Control. Computer Output Control stipulates that:

- All computer output must be uniquely identified (i.e., banner page) with the computer program name and revision, time and date, run number, and inputs.

- It is the responsibility of the User to assign a unique Run Number to each final computer run used for high impact calculations, and to record this number in his Engineering Notebook. The entry to the Engineering Notebook includes a description of the numbered computer run, including purpose, the reason for termination (run completed, aborted, etc.), a listing of all outputs generated, and an overall calculational status (finished, rerun, etc.).

- The Software Development Plan for high-impact analyses must specify the computer output controls. Different options are available, such as archiving paper output or microfiche of the final computer runs in a controlled permanent storage place, or preserving the input files, configuration controlled operating system, and the code version. In addition, depending on the type of calculations, copies of the output can be kept as records.

Software Retirement Phase

When software is retired, a retrievable record of the software and its latest configurations and associated libraries are documented, stored, and controlled. All Users of the software are informed that the software has been retired and is no longer available for unrestricted use.

A request for software retirement is made by the Code Proprietor who fills out a Software Retirement Request (SRR). Upon completion of the SRR, the Code Proprietor obtains the necessary approvals of the responsible Manager, and any additional approvals required by the SCMS Custodian. The Proprietor then forwards the request to the SCMS Custodian.

The SCMS Custodian signs and dates the SRR and forwards it to the Code Proprietor, acknowledging that the software has been "retired" within the SCMS.

The Code Proprietor is responsible for collecting the following documentation and sending it to Records Management:

- A copy, in a compatible format, of the software or data set to be retired, and supporting benchmark and installation test files.

- Copies of applicable software documents, such as installation instructions and tests, verification and validation reports, and operating system specifications in use at the time.

CONCLUSION

As a part of its Reactor Operations Improvement Program at SRS, Westinghouse Savannah River Company, in cooperation with the Westinghouse Hanford Company, has developed and provided quality assurance procedures for the development and control of computer software. Such detailed procedures cover the entire software life cycle including the Requirements, the Software Design and Implementation, Testing and Installation, Operation and Maintenance, and Retirement phases. The current procedures are evolutionary in nature, and thus prone to further refinements. Nevertheless they represent an effective controlling tool for development, production, and operation of safety-related software to be directly applied to reactor safety and probabilistic risk assessment analyses.

ACKNOWLEDGEMENT

The information contained in this article was developed during the course of work done under Contract No. DE-AC09-89SR18035 with the U. S. Department of Energy.

REFERENCES

1. "Quality Assurance Requirement of Computer Software for Nuclear Applications", ANSI/ASME NQA-2 Part 2.7, American Society of Mechanical Engineers.

2. "Standard for Software Quality Assurance Plans", ANSI/IEEE Std-730-1984, Institute of Electrical and Electronics Engineers.

DATE
FILMED
4/24/92