SAND99-2642 J

Software Quality Assurance in the 1996 Performance Assessment for the Waste Isolation Pilot Plant

G. K. Froehlich[1], H. C. Ogden[2], and K. A. Byle[3]

[1]*Regulatory Compliance Department, Sandia National Laboratories, P.O. Box 5800, Albuquerque, NM 87185, USA*
[2]*Technical Integration Department, Sandia National Laboratories, P.O. Box 5800, Albuquerque, NM 87185, USA*
[3]*Quality Assurance Department, Sandia National Laboratories, P.O. Box 5800, Albuquerque, NM 87185, USA*

Abstract

The US Department of Energy (DOE) Waste Isolation Pilot Plant (WIPP), located in southeast New Mexico, is a deep geologic repository for the permanent disposal of transuranic waste generated by DOE defense-related activities. Sandia National Laboratories (SNL), in its role as scientific advisor to the DOE, is responsible for evaluating the long-term performance of the WIPP. This risk-based Performance Assessment (PA) is accomplished in part through the use of numerous scientific modeling codes, which rely for some of their inputs on data gathered during characterization of the site. The PA is subject to formal requirements set forth in federal regulations. In particular, the components of the calculation fall under the configuration management and software quality assurance aegis of the American Society of Mechanical Engineers (ASME) Nuclear Quality Assurance (NQA) requirements. This paper describes SNL's implementation of the NQA requirements regarding software quality assurance (SQA). The description of the implementation of SQA for a PA calculation addresses not only the interpretation of the NQA requirements, it also discusses roles, deliverables, and the resources necessary for effective implementation. Finally, examples are given which illustrate the effectiveness of SNL's SQA program, followed by a detailed discussion of lessons learned.

Keywords: Software quality assurance; Software QA; Software testing; Software verification; Software validation; Software life-cycle; Quality assurance; Performance assessment; Waste Isolation Pilot Plant; Transuranic waste; Radioactive waste

DISCLAIMER

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

1. Introduction

The Waste Isolation Pilot Plant (WIPP) is a deep geologic repository located in southeast New Mexico, which has been licensed for the permanent disposal of transuranic waste generated by US Department of Energy (DOE) defense-related activities [1]. The scientific advisor to DOE, Sandia National Laboratories (SNL), is responsible for evaluating the long-term (10,000-year) performance of the WIPP. This risk-based Performance Assessment (PA) is accomplished in part through the use of numerous scientific modeling codes, which rely for some of their inputs on data gathered during characterization of the site. These calculations depend in large part on computer codes used to simulate processes within the repository system, as well as transport and retardation of radionuclides from the repository through surrounding hydrogeologic formations, and simulation of releases due to possible future human intrusion into the repository. Probabilistic modeling and analysis codes are also used to characterize both the uncertainty of physical parameters and the unpredictability of future events. In such a regulatory environment as nuclear-waste disposal, SNL's work must be held to high standards of accountability. For SNL, this means that the PA codes must comply with rigorous software quality-assurance (SQA) requirements.

The origins of the formal SQA requirements are given, and the status of SQA at the time the requirements were imposed is described. Interpretation of the requirements and their applicability to WIPP PA is discussed. This is followed by a detailed description of SNL's implementation of SQA, including a description of procedures and roles. Examples are cited which demonstrate the effectiveness of our implementation of SQA. Finally, there is a discussion of lessons learned.

2. Origin of SQA Requirements

In 1992, the Land Withdrawal Act [2] named the US Environmental Protection Agency (EPA) as the regulator for WIPP. As such, EPA became responsible for developing disposal regulations, and for certifying the long-term safety of the repository. In late 1993, federal regulation 40CFR191 [3] set forth the disposal regulations and release limits. In effect, this regulation outlined *what* needed to be done to demonstrate compliance with the release limits, without specifying *how* to do it. Then, in early 1996, 40CFR194 [4] established criteria for demonstrating compliance with 40CFR191, in effect specifying *how* to do so. This latter regulation invoked the American Society of Mechanical Engineers' (ASME) Nuclear Quality Assurance (NQA) Standards, *i.e.*, ASME NQA-1-1989 edition, NQA-2a-1990 addenda (Part 2.7) to ASME NQA-2-1989 edition, and ASME NQA-3-1989 edition [5].

The NQA standards were originally developed by the ASME at the request of the American National Standards Institute (ANSI). ASME formed a committee on Nuclear Quality Assurance in 1975, which developed NQA-1 and NQA-2 from the ANSI/ASME N45.2 series of standards and initially issued them in 1979. The NQA standards define QA program requirements for siting, design, construction, operation, and

decommissioning of nuclear facilities. The standards also define QA requirements for planning and executing tasks during fabrication, construction, modification, repair, maintenance, and testing of the systems, components, and structures of nuclear facilities. Part 2.7 defines quality assurance requirements for the development, procurement, maintenance, and use of computer codes. The NQA standard is recognized as the *de facto* standard for the nuclear industry, largely because it is maintained by an active organization that periodically updates the standard to reflect both state-of-the-art technologies and real world experience. This fact, and the maturity of the standard, led EPA to select it over other standards that were considered.

3. Applicability of NQA to WIPP

As described above, the NQA standards were developed for application to nuclear facilities. While WIPP qualifies as a nuclear facility in a certain sense, the PA software clearly has no real-time performance requirements. In a nuclear power plant, there are time-critical issues for software, such as restart capability following a software "crash". Another example is unintended functionality that, by itself or in combination with other unintended functionality, could degrade the entire system (possibly with serious consequences). PA codes, on the other hand, do not share these concerns. If a code "crashes", the cause is found, and the job is merely resubmitted. Unintended functionality, while undesirable, at most results in an incorrect result, which will be discovered in a subsequent review; there is no real-time safety issue. Using this rationale, we interpreted the requirements of Part 2.7 for application to WIPP PA codes. It is important to note that our interpretation was not unilateral — concurrence was obtained from both the customer (DOE) and the regulator (EPA), who were engaged early in the interpretation process.

The majority of the WIPP software that required qualification to Part 2.7 standards was already either partially or completely written before promulgation of 40CFR194. These codes had undergone various degrees of earlier SQA, but in all cases, the earlier QA did not fulfill Part 2.7 requirements (primarily in the area of documentation). SNL applied Part 2.7, Section 10.2, "Software Developed Not Using This Standard", to address this condition. Section 10.2 essentially permits the necessary software requirements, testing, and user documentation to be created after most of the development phases have been completed. Furthermore, Section 10.2 requires that once Part 2.7 is implemented, configuration management and change control per Part 2.7 be implemented as well [6]. Evaluation of our earlier SQA activities showed that it would be more effective to re-do (versus supplement) qualification for previously qualified computer codes, to ensure consistency and to enable uniform application of testing tools and methods.

4. Establishment of Existing SQA Program

Promulgation of 40CFR191 led to development of the SNL Quality Assurance Program Document, Rev. R, 7/31/95, which was the basis for the SNL QA program that was in

place for the Compliance Certification Application (CCA) submitted to EPA [7]. Prior to that time, the SNL SQA program was based on good scientific practice rather than regulatory requirements. Based on the nature of the software and its intended use, a three-level approach for software qualification was developed [8]. The levels were progressive, and were defined as **X** (eXperimental), **C** (Candidate), and **A** (Adjudicated). An **X** level code was one that was still in the developmental stages. At the **X** level, conceptual models were being implemented for evaluation. Testing was conducted, but not formally documented. The testing was conducted and reviewed by the code developer. At this stage the code team consisted of the code sponsor, responsible for guiding the code through the defined QA process, and the code consultant, who was responsible for the theoretical basis (conceptual models, physics, etc.) of the code. In cases where the software was a utility code, this was the same person.

At the point where the conceptual model(s) represented by a code were determined to be applicable for WIPP, the code moved to the next level of QA (level **C**). At this stage the code was a candidate for full quality assurance. A review team (one or more individuals, depending on the complexity of the software being reviewed) was assigned to the code, and a code-qualification package was assembled for review. When the code was determined to be stable and ready for final qualification, the code and its accompanying documentation (test cases, user's manual, theoretical manual) were assembled for consideration as an **A** level package. Rather than following specific criteria, as the codes were of many different types (utility, modeling, etc.), code sponsors and reviewers were given the following guidance — the documentation must be sufficiently complete that a competent expert has assurance the code results are correct.

The review process consisted of providing the reviewers with the code documentation, and then holding an initial review meeting in which the reviewers, code sponsor, code consultant, QA representative, and scribe met to discuss and document any reviewer concerns. After comment resolution was complete, and reviewers were satisfied that the code was working as described, the code was raised to level **A**. The main emphasis of this multi-level approach was the reliance on the reviewers' expertise and technical judgement. This was quite reasonable, since the primary WIPP modeling codes were not amenable to "traditional" model-validation methods; *i.e.*, one cannot hope to validate predictions of performance for a geological system over a 10,000-year period [9]. In comparison to the imposed regulatory NQA requirements, the main features that were lacking were consistency of documentation and consistency in degree of testing. Also, the SQA process was not universally applied. With respect to providing objective evidence of quality and confidence in the modeling codes to both the regulator and the public, this was not acceptable.

5. Implementation of New SQA Program

Once NQA Part 2.7 was identified as the software quality assurance standard for WIPP project participants, SNL began implementation. The standard was evaluated for applicability to WIPP software, and the existing SQA process was compared to the

applicable sections of the standard. The existing SNL software procedures needed revision, after which SNL WIPP staff needed to be trained to them. The NQA Part 2.7 standard represented a different approach to software quality than the multi-level **X**, **C**, and **A** approach. Rather than the software moving through successive quality levels, it would henceforth pass through successive life-cycle phases.

Part 2.7 endorses a systematic, life-cycle approach to software development and, although it does not require the use of a particular life cycle, it uses the ANSI/IEEE 1012 model [10] to illustrate its major components. These are:

(a) the *requirements phase*, in which the functionality, performance, and interface requirements of the code are defined;
(b) the *design phase*, in which overall input, output, and problem-solution strategy are defined;
(c) the *implementation phase*, in which actual computer code is written to implement the design;
(d) the *testing phase*, in which plans for testing the operation and technical correctness of the code are developed and executed; and
(e) the *installation and checkout phase*, in which the code is deployed and tested to demonstrate proper operation on the actual production system(s), and is released for production use. In addition, Part 2.7 addresses an
(f) *operation and maintenance phase*, in which corrective, functional, and/or adaptive modifications are implemented, including updating of activities and information documented in previous phases (a-e); and
(g) a *retirement phase*, in which use of the software is prevented.

These phases are just an example; they are illustrated in Figure 1. The number of phases and the relative emphasis placed on each phase of software development depends on the nature and complexity of the software.

After considering the standard, and with DOE approval, SNL adopted the following life-cycle phases for its software:

(a) the *requirements phase*, in which the functionality, acceptance criteria, and test cases for the code are documented;
(b) the *design phase*, in which the theoretical basis, embodied mathematical models, control flow, control logic, and data structures of the code are documented;
(c) the *implementation phase*, in which the source code is produced, the process of executable generation is documented, the user's manual (which contains instructions that describe the user's interactions with the software) is produced, and the tested functionality requirements are documented;
(d) the *verification and validation phase*, in which test cases are executed and the outputs are evaluated for demonstration that the software produces valid results (for inputs within the range of permitted usage);
(e) the *installation and checkout phase*, in which the qualified executable is installed on the production computer (and test cases are repeated if the production platform is not the same as the platform on which the software was qualified);
(f) the *operation and maintenance phase*, in which software modifications are approved, documented, and controlled; and

(g) the *retirement phase*, in which the use of the code is discontinued.

Recognizing the importance of independent technical review to our software, as well as the standard's identification of a testing phase, we combined these two activities (testing and independent review) into one phase, called verification and validation. In this context, validation is defined as the demonstration that software requirements have been correctly implemented, *not* that the underlying conceptual model has been validated. From Part 2.7, software verification is defined as the process of determining whether or not the product of a given phase of the software development cycle fulfills the requirements imposed by the previous phase. Due to the complexity of the simulation software used by SNL to evaluate WIPP, complete removal of bugs is infeasible. It has long been established in the practice of software engineering that complete removal of bugs by testing, even for relatively simple codes, is impossible from a practical standpoint [11].

Implementation of Part 2.7 required both the development and approval of procedures compliant with Part 2.7, and the documented training of staff to those procedures. This process was complicated for at least two reasons. First, a wide range of computer software was involved, ranging from complex, SNL-written analysis codes to vendor-provided utility codes. Second, many of the codes requiring qualification implemented theoretical models whose results cannot be explicitly proven (*e.g.* predicting the performance of the deep geological repository for 10,000 years into the future). SNL's SQA approach dealt with the theoretical-model issue by splitting software and model verification and validation. SNL's SQA process demonstrated the proper implementation (*i.e.*, coding) of theoretical models in the codes while initially assuming that the theoretical models were appropriate and reasonably representative of the WIPP. Validation of the theoretical models themselves and the appropriateness of their use in modeling repository behavior was deferred to SNL's subsequent analysis-review process.

SNL developed a Software QA Procedure [12] to implement the requirements of Part 2.7. This procedure includes a software classification matrix, wherein documentation and testing requirements for each class of code (ranging from complex, SNL-written analysis codes to vendor-provided utility codes) are explicitly defined. Software verification and validation is implemented in the procedure by requiring development of documentation for each phase, plus a documented independent review of each phase for compliance with its requirements. Reviewer's responsibilities are delineated in review forms for each phase. In addition to formal SQA procedures, informal practices were agreed upon to promote completeness and use of consistent methods.

The new SQA procedure developed for the CCA incorporated detailed review checklists for each of the life-cycle phases. The procedure also established a role identified as the Software Configuration Management (SCM) Coordinator. The duties of the SCM Coordinator included reviewing all SQA documentation for compliance with the procedure, and ensuring that configuration management requirements were followed. For the CCA, this was a full-time position. The SCM Coordinator had final signature approval on all SQA documentation. The resulting configuration management of the

codes and associated documentation was very complete, as indicated by the fact that, despite many audits by DOE, a noncompliance was never issued on activities under the responsibility of the SCM coordinator.

Training software staff to the life-cycle process was accomplished in a number of ways. Group training sessions were provided, guidance memos were issued, and two members from the QA staff were assigned as full-time SQA Consultants. These SQA Consultants were available to answer questions, defend the software process and products during audits, and respond to any software nonconformances. Removing the burden of responding to audits thus freed analysts and code sponsors to pursue their primary tasks.

Prior to acceptance of a software item for use in the CCA, the software item and its associated baseline documentation had to be approved by an independent technical review. From NQA Part 2.7, supplement 3S-1, documentation had to be reviewed to the standard " ... that a person technically qualified in the subject can review and understand ... the adequacy of the results ... without recourse to the originator".

The scope of the software-testing program implemented by SNL included analytical and functional test cases, as required by Part 2.7. In addition, automated analysis of a code's structure (static testing) and run-time behavior (dynamic testing) was performed, based on good industry practice. Test teams were formed to address each code having a direct bearing on the WIPP CCA. The test teams consisted of the code sponsor (or author), code-testing staff whose responsibility was to plan and execute a comprehensive set of code tests, and independent reviewers whose responsibility was to confirm adequate testing and documentation for each code. The way in which SNL organized the test teams was a departure from classical independent software testing, where code developers play no role in the code testing process. However, owing to the unique technical nature of the WIPP codes, the only practical way to develop analytical test cases was to have the subject-matter experts (generally the code authors) develop the test cases and, where possible, define corresponding acceptance criteria. The required "independence" of testing was provided by an independent review of the scope of the testing and of the documented results of the tests.

6. Effectiveness

The SQA and testing effort for the CCA began in earnest early in 1995 with the creation of the new SQA procedure, which fully complied with the requirements of NQA Part 2.7. All appropriate WIPP personnel were trained to this procedure in July of that year. Since it was clear that SNL did not have the personnel resources necessary to complete the qualification and testing tasks on time, qualified personnel were identified and contracted through several existing contracting organizations. In all, about 20 persons were added to the existing personnel to form a total team of about 60 persons working on the SQA effort. There were 21 code sponsors, 10 software testers, 19 technical reviewers, 2 SQA Consultants, 15 documentation support persons (consisting of both technical writers and word-processing support), and 4 management coordinators. Some individuals performed

multiple roles. Each code to be qualified was assigned a team consisting of one code sponsor, one tester, one technical reviewer, one SQA Consultant, and up to four documentation support personnel (typically one or two technical writers, plus word-processing support). Weekly status meetings were held to track the progress of every deliverable, identify and solve problems, and demonstrate the strong management support for the effort.

Over the six-month period from July 1995 to January 1996, a total of 64 codes (comprising about half-a-million source lines of code) were qualified. Of these, there were 15 scientific modeling codes, 19 utility codes, 10 pre- and post-processor codes, 5 subroutine libraries, and 15 Data Acquisition System (DAS) codes. The resulting documentation set for each qualified code consisted of a Requirements Document (RD), a Verification and Validation Plan (VVP), an Implementation Document (ID), a Validation Document (VD), and a Users Manual (UM). Because all of theses codes were "pre-existing" codes, that is, they were developed prior to implementation of Part 2.7, they were exempted from the requirement for a Design Document (DD) by Section 10.2 of NQA Part 2.7, as previously described. All documentation and test results, including test scripts, input files, and output files, were stored in the Software Configuration Management (SCM) system [6].

The EPA was involved in our SQA effort through observation of formal DOE audits and less-formal technical interaction sessions. The EPA was especially interested in the adequacy of the testing for our main scientific modeling codes. They reviewed the test cases and test results, interacted with our technical reviewers, and even brought contracted technical experts of their own to review our work. In many cases these interactions resulted in the addition of new test cases, the modification of existing test cases, and, in a few cases, the modification of the codes. In the end, they approved and accepted all deliverables from the SQA effort. Engaging the regulator in our processes from the beginning proved to be very beneficial to the success of this effort.

7. What Worked Best

Several practices that we adopted proved to be critical to the success of the SQA effort. One of these was weekly status meetings, which were handled in a manner that minimized the impact on the schedules of the code teams (*i.e.*, the code sponsors, testers, technical reviewers, and documentation support staff), but required tremendous time commitments on the part of the management-support team. These status meetings allowed us to efficiently track the progress of every deliverable, and to identify problems early so that they could be addressed before they had much impact. The meetings also provided powerful motivation to the code teams to expeditiously perform the required work, due to the continual presence of management at the meetings.

Another practice that proved to be invaluable was the involvement of professional documentation-support personnel. Each code had up to four such persons assigned to it. This enabled the code sponsors and the testers to work more efficiently, because they did

not need to be concerned with the details of developing professional-level documentation. They merely had to supply the correct information to the documentation-support staff, who would then build the documents. This expedited the overall process greatly, and was a major factor in our overall success.

Also, we involved the regulator (EPA) early in the development of our SQA processes. This proved to be very valuable to our overall success. By viewing the regulator as an ally rather than an opponent, we were able to modify our processes to better suit their concerns, and in so doing, we improved our deliverables and our processes. At the same time, the regulators were able to become more familiar with our codes and the testing that was performed on them. This gave everyone assurance that we were on the right track, and that the level of documentation and testing which we intended to provide in the CCA would indeed be adequate.

Over the period of time that this SQA effort was underway, we experienced several audits and surveillances by our customer, DOE, which were observed by the regulator, EPA. Although these audits were difficult at the time because of schedule pressures, they proved to be extremely useful. First, they gave both our customer and our regulator direct input into what we were doing. It also provided us the opportunity to identify and correct weaknesses in our processes before they were propagated, or included in the CCA. The reviews also improved the overall quality of our deliverables. These formal interactions were also instrumental in obtaining the strong support of our management and the complete cooperation of our code sponsors. These reviews produced a large number of Corrective Action Reports (CARs), all of which had to be addressed and resolved. This was a painful, but beneficial process!

8. Lessons Learned

Probably the most important lesson that we learned from this effort is that it is more efficient and less costly to build quality in, as a routine part of your daily work, than to try and add it on at the end. This is difficult to accomplish in a scientific research and development organization because it requires a complete change of the culture, but in a regulatory environment it is absolutely necessary.

Our QA and testing, due to the nature of our codes, relied heavily on the quality of the technical reviews that were performed. In some cases, the quality of technical reviews was inadequate, and this was determined early in our process, during audits. Reviewer training is essential to convey what constitutes a thorough review. In addition, most technical reviewers must be reminded that their task is *not* to suggest alternatives to the way the work was done, but rather to evaluate whether the way it *was* done was adequate for its intended use. It is also important that the reviewer have adequate time to perform the review, free of any pressure to produce a predetermined result.

The implementation of SQA on a day-to-day basis required a change of culture, as mentioned above. Such a change can only be accomplished with the strong, consistent,

and visible support of the management. This support must include the necessary resources to complete the testing and documentation. The development costs of properly tested and documented software are largely incurred at the beginning of the process. The savings that result from continuous, ongoing software QA are accrued mainly in the operation and maintenance phase, in which software spends 80% of its lifetime. It is imperative that management recognize, accept, and support this. They must provide the QA staff, the coordinators, the software testers, the technical reviewers, and the documentation support staff. There must also be adequate support for the ongoing training of all staff. Also, management must allow enough time in schedules for adequate testing and documentation. These latter tasks typically require a level of resources equivalent to, and sometimes greater than, that required for the actual design and implementation of the software.

Research is by nature a hypothesis-verification, "prototyping" process that generally does not lend itself to long-term pre-planning. Documentation tends not to be the prime focus or interest of the researcher. However, the overall success of a regulated activity depends equally on the technical quality of work and development of comprehensive documentation, *i.e.*, the production of "objective evidence". To succeed at both scientific research and compliance with regulations, an organization must make a complete philosophical and working-level commitment to both. The otherwise high technical quality of scientific research can, on regulated projects, be brought into serious question during technical review or litigation processes if documentation is incomplete, appears to be inconsistent, or is of poor quality. This is increasingly true when the theoretical solutions that support project conclusions have no demonstrable method of validation. During the early portion of the WIPP PA SQA effort, SNL staff struggled with this dichotomy until the two (research and documentation of objective evidence) came into symbiosis. Our eventual success was a result of supportive management, and the recognition and acceptance by SNL staff of the importance of both requirements, which evolved over time.

The practical need for documentation in highly theoretical research became most apparent when it came time to execute and document the testing phase. The answer to the question "what should be tested?" lay in the functional requirements, which initially could be found only in the minds of the developers/researchers. The need to link software requirements with the definition of the testing scope and the definition of acceptance criteria for test results required considerable effort (and perhaps some re-work) that could have been simplified (or re-work avoided) if the objectives of the software-development effort for each code were spelled out in advance of coding, and maintained as changes were made throughout the development effort.

Importantly, sheer volume of documentation is not the solution. Optimal documentation, focused exclusively at fulfilling regulatory requirements and demonstrating technical adequacy, is. At the outset of regulated QA work, organizations must carefully identify what documentation is required by procedures, regulations, or other instructions, as opposed to documentation that is merely recommended or desired. Complete, consistent, reviewed, and controlled documentation must be provided to address SQA

documentation requirements. However, other documentation, which does not derive from specific SQA requirements, and thus may not be fully reviewed or maintained, should be minimized. Potential inconsistencies between SQA and non-SQA documentation can undermine the quality of all documentation, both in the eyes of well-intentioned regulators as well as intervenors who are planning litigation. The fallout from this confusion can be very expensive and difficult to correct after the fact.

9. Conclusions

The SNL SQA effort for the WIPP CCA proved to be highly successful in that the work was accepted and approved by the EPA and, ultimately, the CCA itself was approved [13]. While much effort (and even pain) was involved, we learned much and ended up with a solid SQA program that should serve us well for the future. It is impossible to be successful in a regulatory environment without a solid SQA program that is rigorously followed.

Acknowledgments

Work performed for Sandia National Laboratories (SNL), which is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000. Review provided at SNL by J. Helton, M. Shortencarrier, and A. Kane.

References

[1] Rechard, R.P., Historical Background on Performance Assessment for the Waste Isolation Pilot Plant, *Reliability Engineering and System Safety* (in this issue).

[2] Waste Isolation Pilot Plant Land Withdrawal Act, Public Law 102-579, (106 Stat. 4777), 1992.

[3] U.S. Environmental Protection Agency, 40 CFR Part 191: Environmental Radiation Protection Standards for the Management and Disposal of Spent Nuclear Fuel, High-Level and Transuranic Radioactive Wastes; Final Rule, *Federal Register*, 1993, **58**(242), 66398-66416.

[4] U.S. Environmental Protection Agency, 40 CFR Part 194: Criteria for the Certification and Re-Certification of the Waste Isolation Pilot Plant's Compliance With the 40 CFR Part 191 Disposal Regulations; Final Rule, *Federal Register*, 1996, **61**(28), 5224-5245.

[5] American Society of Mechanical Engineers (ASME) Nuclear Quality Assurance (NQA) Standards, ASME NQA-1-1989 edition, *Quality Assurance Program*

Requirements for Nuclear Facilities; ASME NQA-2a-1990 addenda, part 2.7, to ASME NQA-2-1989 edition, *Quality Assurance Requirements for Nuclear Facility Applications*; and ASME NQA-3-1989 edition, *Quality Assurance Program Requirements for the Collection of Scientific and Technical Information for Site Characterization of High-Level Nuclear Waste Repositories*; American Society of Mechanical Engineers, Fairfield, NJ, 1989-1992.

[6] Froehlich, G.K., Williamson, C.M., and Ogden, H.C., Computational Environment and Software Configuration Management of the 1996 Performance Assessment for the Waste Isolation Pilot Plant, *Reliability Engineering and System Safety* (in this issue).

[7] U.S. Department of Energy, *Title 40 CFR Part 191 Compliance Certification Application for the Waste Isolation Pilot Plant*, DOE/CAO-1996-2184, Volumes I-XXI, U.S. Department of Energy, Waste Isolation Pilot Plant,Carlsbad Area Office, Carlsbad, NM, 1996.

[8] Rechard, R.P., Roache, P.J., Blaine, R.L., Gilkey, A.P., and Rudeen, D.K., *Quality Assurance Procedures for Computer Software Supporting Performance Assessments of the Waste Isolation Pilot Plant*, SAND90-1240, Sandia National Laboratories, Albuquerque, NM, April 1991.

[9] Oreskes, N., Shrader-Frechette, K., and Belitz, K., Verification, Validation, and Confirmation of Numerical Models in the Earth Sciences, *Science*, February 1994, **263**(5147), 641-646.

[10] Institute of Electrical and Electronics Engineers, IEEE Standard for Software Verification and Validation Plans, *IEEE Standard 1012-1986*, New York, NY: IEEE Press, 1986.

[11] Myers, G.J., *The Art of Software Testing*, New York: Wiley-Interscience, 1979.

[12] Spinney, K., Waste Isolation Pilot Plant Quality Assurance Procedure (QAP), *QAP 19-1, WIPP Computer Software Requirements*, rev. 2, Sandia WIPP Central Files, WPO#37202, Sandia National Laboratories, Albuquerque, NM, November 1995.

[13] U.S. Environmental Protection Agency, 40 CFR Part 194: Criteria for the Certification and Re-Certification of the Waste Isolation Pilot Plant's Compliance With the Disposal Regulations: Certification Decision; Final Rule, *Federal Register*, 1998, **63**(95), 27354-27406.

List of Figures

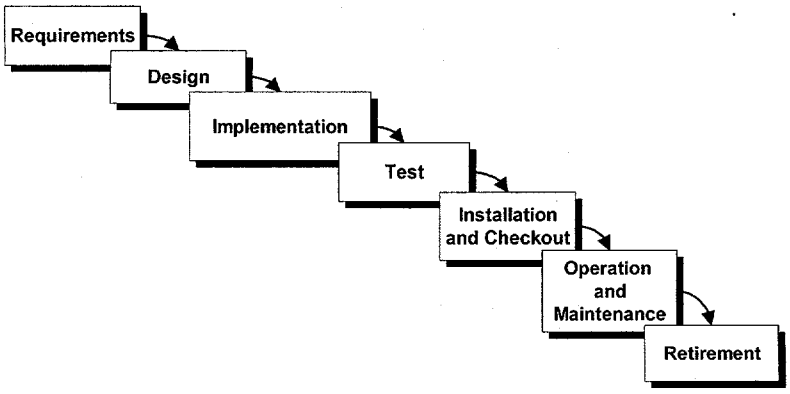

Figure 1. The software life cycle (after NQA part 2.7).