Status of SFR Codes and Methods QA Implementation

Nuclear Engineering Division

About Argonne National Laboratory

Argonne is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC under contract DE-AC02-06CH11357. The Laboratory's main facility is outside Chicago, at 9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne and its pioneering science and technology programs, see www.anl.gov.

DOCUMENT AVAILABILITY

Online Access: U.S. Department of Energy (DOE) reports produced after 1991 and a growing number of pre- 1991 documents are available free via DOE's SciTech Connect (<http://www.osti.gov/scitech/>)

Reports not in digital format may be purchased by the public from the National Technical Information Service (NTIS):

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Rd
Alexandria, VA 22312
www.ntis.gov
PHONE: (800) 553-NTIS (6847) or (703) 605-6000
Fax: (703) 605-6900
Email: **orders@ntis.gov**

Reports not in digital format are available to DOE and DOE contractors from the Office of Scientific and Technical Information (OSTI):

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
www.osti.gov
Phone: (865) 576-8401
Fax: (865) 576-5728
Email: **reports@osti.gov**

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 UChicago Argonne, LLC, nor any of their employees or officers, 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 document authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, Argonne National Laboratory, or UChicago Argonne, LLC.

Status of SFR Codes and Methods QA Implementation

Prepared by
Acacia J. Brunett, Laural L. Briggs, Thomas H. Fanning
Nuclear Engineering Division, Argonne National Laboratory

January 31, 2017

Acknowledgements

The authors would like to express their appreciation to the members of the Quality Assurance group of Argonne National Laboratory's Research and Test Reactor Department in the Nuclear Engineering Division for their guidance and insight on development of a software quality assurance program.

Keith Derstine

Benoit Dionne

Donna Ferguson

Erik Wilson

Executive Summary

A key component of reactor design commercialization in the U.S. is completion of a license application, which must ultimately be approved by the Nuclear Regulatory Commission (NRC). The NRC's approval of the license application is contingent on, among other things, satisfactory analysis of the design basis and response to transient and accident scenarios using accepted codes and models. Despite interactions with the NRC on various sodium fast reactor (SFR) designs throughout the 1970s, 1980s, and 1990s, the SFR has not received a high degree of regulatory exposure, as no SFR has been commercially licensed by the NRC (Fermi-1 was licensed by the Atomic Energy Commission, the NRC's predecessor, which is representative of a different regulatory environment).

In an effort to identify and resolve any existing safety and licensing barriers, Sandia National Laboratories (SNL) compiled a licensing research plan in which a comprehensive group of subject matter experts identified and characterized existing gaps in the safety case that would be used to satisfy the license application for an SFR. Of the five topical areas of focus in the study, the ability of existing codes and models to support a license application was identified as a high priority area that should be addressed in the near term. A subsequent gap analysis by Argonne National Laboratory reiterated the findings of the SNL study and indicated that resolution of this item had high regulatory significance with the potential for significant gain with minimal NRC input.

As the result of recommendations of these reports and the DOE Advanced Reactor Technology (ART) Program's Regulatory Technology Development Plan (RTDP), a task was launched to systematically develop SFR safety analysis codes and methods that can be utilized in a licensing framework. As part of the preliminary phase of this effort which was completed in FY16, Argonne identified the safety analysis codes and methods necessary to support an SFR license application and summarized the state of each code in terms of its potential use to support a license application. A parallel effort by Oak Ridge National Laboratory (ORNL) identified the software quality assurance (SQA) requirements necessary to develop and maintain a code of a pedigree acceptable for use in a domestic license application.

As a continuation of the RTDP effort, the current task seeks to establish and begin implementation of a provisional SQA program for a prioritized list of safety codes in accordance with the recommendations set forth in the ORNL study. Accordingly, a pilot SQA program is under development for the Argonne severe accident analysis systems code SAS4A/SASSYS-1, a code integral to the majority of required safety analyses. An ultimate objective of this effort would be to leverage the expertise and framework developed in this phase by applying it to additional safety codes once the SAS4A/SASSYS-1 SQA Program achieves sufficient maturity.

This report details development of the SAS4A/SASSYS-1 SQA Program and describes the initial stages of Program implementation planning. The provisional Program structure, which is largely focused on the establishment of compliant SQA documentation, is outlined in detail, and Program compliance with the appropriate SQA requirements is highlighted. Additional program activities, such as improvements to testing methods and Program surveillance, are also described in this report. Given that the programmatic resources currently granted to development of the SAS4A/SASSYS-1 SQA Program framework are not sufficient to adequately address all SQA

requirements (e.g. NQA-1, NUREG/BR-0167, etc.), this report also provides an overview of the gaps that remain the SQA program, and highlights recommendations on a path forward to resolution of these issues. One key finding of this effort is the identification of the need for an SQA program sustainable over multiple years within DOE annual R&D funding constraints.

Table of Contents

1	Introduction.....	1
2	Background.....	4
2.1	SAS4A/SASSYS-1	4
2.1.1	Project History	4
2.1.2	Users.....	5
2.1.3	Capabilities and Expected Role in Licensing.....	7
2.2	ORNL SQA Recommendations	7
2.2.1	Program Evaluation.....	9
2.2.2	NQA-1-2008/2009 Compliant Program	10
2.2.3	Commercial Grade Dedication	10
2.3	SQA Methodologies and Requirements	11
3	Provisional Program Structure	15
3.1	Software Quality Assurance Plan.....	16
3.2	Configuration Management Plan	17
3.3	Coding Standards.....	18
3.4	Procedures	18
3.4.1	Software Development and Modification.....	20
3.4.2	Error Reporting and Corrective Actions.....	21
3.4.3	Software Testing.....	21
3.4.4	Version Release	21
4	SQA Program Activities	22
4.1	Implementation Planning.....	22
4.2	Technical Activities.....	23
4.2.1	V&V Test Suite Expansion.....	23
4.2.2	Development and Implementation of Additional Testing Algorithms	23
4.2.3	Improved Compiler Support.....	24
5	Path Forward.....	26
5.1	Planned Activities.....	26
5.2	Near Term Required Program Development	28
5.3	Longer Term Gap Resolution	30
6	Works Cited.....	31
	Appendix A.....	33

List of Figures

Figure 1.1: Timeline of SAS4A/SASSYS-1 SQA Program Development Phases.....	2
Figure 2.1: Guidance and Requirements Documents per SQA Methodology	12
Figure 3.1: Graphical Representation of SQA Program Hierarchy	16
Figure 3.2: Ticket Workflow Defined for the SAS4A/SASSYS-1 Trac Environment	20
Figure 4.1: Preliminary SQA Program Implementation Stages.....	22
Figure 4.2: Diagram of Interactive Testing Process	24
Figure 5.1: SAS4A/SASSYS-1 SQA Program Development Needs	26

List of Tables

Table 2.1: SAS4A/SASSYS-1 Capability Description.....	7
Table 2.2: SQA Activities Common to NQA-1-2008/2009 and NUREG/BR-0167	14
Table 3.1: Contents of SAS4A/SASSYS-1 SQAP	17
Table 3.2: Contents of SAS4A/SASSYS-1 CMP	18
Table 5.1: Prioritized SAS4A/SASSYS-1 SQA Program Gaps	28
Table 5.2: Long Term SAS4A/SASSYS-1 SQA Program Gaps	30
Table A.1: Crosswalk of SQA Activities with Requirements and SAS4A/SASSYS-1 SQA Program.....	33
Table A.2: Crosswalk of SQAP Elements with Requirements.....	36
Table A.3: Crosswalk of CMP Elements with Requirements.....	38

Acronyms

ART	Advanced Reactor Technology
ASME	American Society of Mechanical Engineers
CM	Configuration Management
CMP	Configuration Management Plan
CNWG	Civil Nuclear Energy Research and Development Working Group
CRBR	Clinch River Breeder Reactor
CS	Coding Standards
CSAU	Code Scaling, Applicability, and Uncertainty
FFTF	Fast Flux Test Facility
GE	General Electric
IFR	Integral Fast Reactor
INL	Idaho National Laboratory
NRC	Nuclear Regulatory Commission
ORNL	Oak Ridge National Laboratory
PIRT	Phenomena Identification and Ranking Table
PNNL	Pacific Northwest National Laboratory
PRISM	Power Reactor Innovative Small Module
QA	Quality Assurance
RTDP	Regulatory Technology Development Plan
SAFR	Sodium Advanced Fast Reactor
SDD	Software Design Documentation
SFR	Sodium Fast Reactor
SNL	Sandia National Laboratories
SPP	Software Project Plan
SQA	Software Quality Assurance
SQAP	Software Quality Assurance Plan
SRS	Software Requirements Specifications
SVN	Subversion
V&V	Verification and Validation

1 Introduction

Deployment of a sodium fast reactor (SFR) beyond the research and development (R&D) space requires identification of a clear path forward to commercialization of the technology. In the U.S., a key component of commercialization is completion of a license application, which must ultimately be approved by the Nuclear Regulatory Commission (NRC). The NRC's approval of the license application is contingent on, among other things, satisfactory analysis of the design basis and response to transient and accident scenarios using accepted codes and models.¹

Despite interactions with the regulating body on the Fast Flux Test Facility (FFTF) [1], the Clinch River Breeder Reactor (CRBR) Project [2, 3], Rockwell International's Sodium Advanced Fast Reactor (SAFR) [4], and General Electric's (GE) Power Reactor Innovative Small Module (PRISM) [5], the SFR has not received a high degree of regulatory exposure, as no SFR has been commercially licensed by the NRC (Fermi-1 was licensed by the Atomic Energy Commission, the NRC's predecessor, which represents a different regulatory environment). In an attempt to determine and resolve any existing safety and licensing barriers, Sandia National Laboratories (SNL) completed a licensing research plan [6] in which existing gaps in the safety case that would be used to satisfy the license application for an SFR were identified and examined. Of the five topical areas of focus in [6], the ability of existing codes and models to support a license application was identified as a high priority area that should be addressed in the near term. A subsequent gap analysis by Argonne National Laboratory reiterated the findings of [6] and indicated that resolution of this item had high regulatory significance with the potential for significant gain with minimal NRC input.

As the result of recommendations of these reports and the ART Program's Regulatory Technology Development Plan (RTDP) [7], a task was launched to systematically develop SFR safety analysis codes and methods that can be utilized in a licensing framework. As a part of the preliminary phase of this effort, which was completed in FY16, Argonne identified the safety analysis codes and methods necessary to support an SFR license application and summarized the state of each code in terms of its potential use to support a license application [8]. A parallel effort by Oak Ridge National Laboratory (ORNL) [9] identified the software quality assurance (SQA) requirements necessary to develop and maintain a code of a pedigree acceptable for use in a domestic license application.

As a continuation of the RTDP effort, the current task seeks to establish and begin implementation of a provisional SQA program for a prioritized list of safety codes in accordance with the recommendations set forth in the ORNL study. In order to achieve the preliminary goals of this task, it was necessary to select a code suitable for application of the pilot program. The Argonne severe accident analysis systems code SAS4A/SASSYS-1 [10] was selected for this effort due to its prevalent use in a wide range of steady state and transient safety analyses, making it a key tool in a license application. Additionally, the SAS4A/SASSYS-1 development team is currently utilizing the majority of SQA best practices (e.g. configuration management, regression

¹ There is significant distinction between codes and models considered to be acceptable for use in a license application, versus a regulatory code that NRC staff or contractors may use to independently confirm safety analyses. This report highlights the alternative pathways for SQA compliance.

testing, etc.), making it an ideal candidate for efficient development of an SQA program under the RTDP effort.

This report details Phase 1 of SAS4A/SASSYS-1 SQA Program development spanning December 2015 through January 2017, where Phase 1 primarily entails development of the provisional SQA Program. Phase 1 follows the early FY16 effort in which SFR safety analysis codes and methods were identified and examined for their suitability to be used in a license application [8]. The remaining portion of this effort, Phase 2, will conclude in September 2017, and will entail finalization and implementation of the SQA Program, as well as completion of key technical activities. A timeline summarizing the SQA Program development stages is provided in Figure 1.1.

Figure 1.1: Timeline of SAS4A/SASSYS-1 SQA Program Development Phases

Much of the provisional SQA Program structure, which is largely focused on compliant SQA documentation, was developed in accordance with the recommendations outlined in [9], which prescribed alternative SQA methodologies dependent on intended code use. At a high level, a safety code can be utilized in a license application by either the licensee, requiring fulfillment of a commercial-grade dedication process, or as a confirmatory tool by the NRC, requiring compliance with NRC requirements set forth in various NUREG documents. In an effort to maximize the effectiveness of this effort, an SQA program capable of satisfying the key points of either code application has been developed, with the expectation that future efforts can be focused on fulfillment of the more rigorous, common requirements of either user (i.e. license applicant versus NRC). As such, the provisional SQA Program structure and its compliance with the appropriate SQA requirements are described in this report.

In addition to the SAS4A/SASSYS-1 SQA Program structure, program activities, such as Program implementation and improved testing methods are also described in this report. While implementation of a suitable SQA program framework is essential to maintaining a pedigreed code, SQA involves continuous program improvement. Activities are largely centered on improvement of key testing methods to improve code quality; long term addition of regression, unit, and validation test problems are required for a monolithic code such as SAS4A/SASSYS-1. Expansion of the existing user base will ultimately contribute to test suite expansion, as users will continuously exercise the code and discover new limitations or issues. Beyond technical use of the code, SQA requires regular program surveillance to assure compliance with the SQA program.

This report is structured as follows: **Section 2** provides background information on the SAS4A/SASSYS-1 code and the recommendations provided by ORNL in [9], and outlines the requirements of relevant SQA documents. **Section 3** describes the provisional SAS4A/SASSYS-1 SQA Program structure, including a description of all associated documentation and fulfillment of the various SQA requirements. **Section 4** provides details on the SQA Program implementation planning and its current status, as well as technical efforts completed up to this point. Lastly, **Section 5** outlines activities planned for Phase 2, of this effort and identifies known SQA Program gaps.

2 Background

This section provides background information on the SAS4A/SASSYS-1 code, including a summary of the project history (Section 2.1.1), a listing of active users (Section 2.1.2), and a review of the capabilities of the code and its expected role in licensing (Section 2.1.3). Background information on ORNL's recommended SQA development activities is provided in Section 2.2, and a review of the alternative SQA methodologies and corresponding requirements is included in Section 2.3.

2.1 SAS4A/SASSYS-1

2.1.1 Project History

Development of the SAS series of codes [11-14] began in the mid-1960s to model the initiating phases of hypothetical core disruptive accidents in oxide fueled sodium cooled fast reactors. SAS1A originated from a sodium-boiling model and included single- and two-phase coolant flow dynamics, fuel and cladding thermal expansion and deformation, molten fuel dynamics, and a point kinetics model with reactivity feedback. By 1974, SAS evolved to the SAS2A computer code [13] which included a detailed multiple slug and bubble coolant boiling model which greatly enhanced the ability to simulate the initiating phases of loss of flow and transient overpower accidents up to the point of cladding failure and fuel and cladding melting.

The SAS3A code [14] added mechanistic models of oxide fuel and cladding melting and relocation. This version of the code was used extensively for analysis of accidents in the licensing of the Fast Flux Test Facility. In anticipation of loss of flow and transient overpower analysis requirements for licensing of the Clinch River Breeder Reactor Plant, new fuel element deformation, disruption, and material relocation models were written for the SAS4A version of the code which saw extensive validation against TREAT M-Series test data. In addition, a variant of SAS4A, named SASSYS-1 [13] was developed with the capability to model ex-reactor coolant systems to permit the analysis of accident sequences involving or initiated by loss of heat removal or other coolant system events. This allows the simulation of whole-plant dynamics feedback for both shutdown and off-normal conditions.

Although SAS4A and SASSYS-1 are sometimes portrayed as two computer codes, they have always shared common code architectures, the same data management strategy, and the same core channel representation. Subsequently, the two code branches were merged into a single code referred to as SAS4A/SASSYS-1. Version 2 of the SAS4A/SASSYS-1 code was distributed to Germany, France, and Japan in the late 1980s, and it served as a common tool for international oxide fuel model developments.

Beyond the release of SAS4A/SASSYS-1 v 2.1, revisions to SAS4A/SASSYS-1 continued throughout the Integral Fast Reactor (IFR) program between 1984 and 1994 [15] culminating with the completion of SAS4A/SASSYS-1 v 3.0 in 1994. During this time, the modeling emphasis shifted towards metallic fuel and accident prevention by means of inherent safety mechanisms. This resulted in 1) addition of new models and modification of existing models to treat metallic fuel, its properties, behavior, and accident phenomena, and 2) addition and validation of new capabilities for calculating whole-plant design basis transients, with emphasis on the EBR-II reactor and plant [16], the IFR prototype. The whole-plant dynamics capability of

the SASSYS-1 component plays a vital role in predicting passive safety feedback. Without it, meaningful boundary conditions for the core channel models are not available, and accident progression is not reliably predicted.

By the mid 1990s, SAS4A/SASSYS-1 v 3.1 had been completed as a significant maintenance update, but it was not released until 2012 [10]. In the time since the development of Version 3, a variety of modeling additions and enhancements have been made to meet U.S. DOE programmatic needs. This collection of updates was released in 2012 as SAS4A/SASSYS-1 Version 5.0.

The DOE-NE Office of Advanced Reactor Technologies is the current sponsor for development and maintenance of the code. Argonne National Laboratory obtained permission from DOE to assert copyright over SAS4A/SASSYS-1 in May 2012. To obtain a license, prospective users need to contact the Technology Development and Commercialization Division of Argonne. Licenses may be obtained for executable only or for full source code access. Once a fully executed license has been approved, the Project Leader handles distribution to the licensee.

In addition to the full version of SAS4A/SASSYS-1, a limited version, *Mini SAS*, is also available for non-commercial use. Mini SAS is compiled from the same source as SAS4A/SASSYS-1, but it excludes all severe accident (sodium boiling, plus fuel melting and relocation) and steam balance of plant capabilities. It also limits a model to five core channels. Mini SAS does support primary and intermediate coolant system models and all reactivity feedback effects for passive safety studies. In 2015, Argonne also obtained permission from DOE to assert copyright over Mini SAS. The Argonne Legal department has approved a general license agreement for executable copies only. Because Mini SAS is intended for R&D activities and trial applications only, an independent SQA program does not exist for the code. However, because Mini SAS is compiled from the same source as SAS4A/SASSYS-1, Mini SAS is subject to the majority of activities performed under the SAS4A/SASSYS-1 SQA Program.

2.1.2 Users

SAS4A/SASSYS-1 is utilized by a variety of groups in the realms of R&D and academia. Active programs and collaborators that currently use SAS4A/SASSYS-1 include the following:

- **EBR-II IAEA Benchmark:** The DOE-NE Advanced Reactor Concepts program is supporting a high-profile Coordinated Research Project with the International Atomic Energy Agency based on the Shutdown Heat Removal Tests conducted at EBR-II. Both protected (SHRT-17) and unprotected (SHRT-45R) loss of flow tests are part of the benchmark activity. SAS4A/SASSYS-1 models of both tests have been developed to provide results under the CRP.
- **FFTF Benchmark:** Pacific Northwest National Laboratory (PNNL) and Argonne National Laboratory are preparing benchmark specifications for the Passive Safety Tests carried out at the Fast Flux Test Facility between 1984 and 1986. The most prominent tests were the loss of flow without scram. In collaboration with PNNL, Argonne National Laboratory is assessing the benchmark specifications and preparing SAS4A/SASSYS-1 models for verification and validation purposes.

- **GAIN Voucher Program:** Oklo, Inc. is one of the recipients of a Gateway for Accelerated Innovation in Nuclear award. Under the award, Oklo is gaining access to knowledge of metallic alloy fuel. They are also investigating the available tools for fast reactor analysis and have acquired a license for Mini SAS version 5.1.
- **NEUP Projects:** Several Nuclear Energy University Program awards are utilizing SAS4A/SASSYS-1 or Mini SAS as part of their scope of work. The University of California at Berkeley is using Mini SAS to evaluate safety benefits that might be achieved with autonomous reactivity control devices. Other universities are preparing experiments that can improve the modeling of thermal stratification in SFRs, with the goal that the improved models would be incorporated into SAS4A/SASSYS-1.
- **GEH:** GE-Hitachi Nuclear Energy Americas, LLC has a government-use license of SAS4A/SASSYS-1 to carry out development of modern probabilistic risk assessment evaluations of the PRISM reactor concept. GEH and Argonne National Laboratory are collaborating on this work under the 2014 Advanced Reactor Research and Development funding opportunity (DE-FOA-0001163).
- **ASTRID Collaboration with CEA:** An implementation agreement has been established between the U.S. DOE and the Commissariat à l'énergie atomique et aux énergies alternatives of France for cooperation in low carbon energy technologies. One purpose of the agreement is to evaluate the safety performance of the ASTRID reactor design. DOE uses the SAS4A/SASSYS-1 safety analysis code for simulations performed under the agreement.
- **CIAE Bilateral Collaboration:** The DOE-NE Office of International Nuclear Energy Policy and Cooperation has established the U.S.-China Bilateral Civil Nuclear Energy Cooperative Action Plan with the China Institute of Atomic Energy. Joint activities under the action plan include model development and safety analyses of the China Experimental Fast Reactor using SAS4A/SASSYS-1.
- **TerraPower TWR Reactor Concept:** TerraPower, LLC has licensed the SAS4A/SASSYS-1 source code to perform safety analysis studies for their Traveling Wave Reactor concept. TerraPower also funds code development activities that improve the modeling capabilities of SAS4A/SASSYS-1.
- **KAERI PG-SFR:** The Korean Atomic Energy Research Institute acquired a license for SAS4A/SASSYS-1 to perform safety analysis and model development for the Prototype Generation-IV Sodium Fast Reactor. KAERI is supporting metallic fuel severe accident model developments that will be incorporated into a future version of SAS4A/SASSYS-1.
- **KINS:** The Korea Institute of Nuclear Safety is an independent regulatory expert organization that supports the Nuclear Safety and Security Commission in Korea. KINS recently acquired a license for SAS4A/SASSYS-1 to support the regulatory obligations over the PG-SFR project.
- **KTH ELECTRA LFR Concept:** The Royal Institute of Technology (Kungliga Tekniska Högskolan) in Stockholm Sweden has a license for SAS4A/SASSYS-1 to perform natural circulation design performance studies of their ELECTRA lead-cooled fast reactor concept.

- **JAEA Bilateral Collaboration:** The Civil Nuclear Energy Research and Development Working Group (CNWG) was established by the U.S.-Japan Bilateral Commission on Civil Nuclear Cooperation in 2012 to enhance coordination of joint civil nuclear research and development efforts. The Japan Atomic Energy Agency and Argonne National Laboratory plan to collaborate under the CNWG to improve the oxide fuel severe accident modeling capabilities in SAS4A/SASSYS-1.
- **NRA:** The Nuclear Regulatory Authority of Japan has acquired a license for SAS4A/SASSYS-1. The NRA plans to use SAS4A/SASSYS-1 to support the potential relicensing evaluation of Monju.

2.1.3 Capabilities and Expected Role in Licensing

As an integrated systems analysis tool, SAS4A/SASSYS-1 possesses the capability to model the majority of steady state and transient phenomena expected to occur in an SFR. A result of the initial FY16 Argonne effort was the identification of high level functional areas that characterize the behavior of a nuclear power plant. Table 2.1 lists the functional areas identified in the FY16 Argonne effort that must be analyzed in a license application and identifies the areas for which SAS4A/SASSYS-1 has some level of capability. Additional details on each functional area with regard to the specific behavior or phenomena can be found in [8]. Details on the maturity of the various models in SAS4A/SASSYS-1 can be found in [6]. As indicated by Table 2.1, SAS4A/SASSYS-1 treats, or provides supporting calculations to, nearly all functional areas. As no other systems analysis tool exists with such a wide range of capabilities and maturity, it is expected that SAS4A/SASSYS-1 would be utilized as a primary tool in safety analyses supporting a license application.

Table 2.1: SAS4A/SASSYS-1 Capability Description

Functional Area	SAS4A/SASSYS-1 Capability*
<i>Steady State Core-Wide Thermal Hydraulics</i>	Primary
<i>Fission Gas Behavior</i>	Primary
<i>Fuel and Clad Motion</i>	Primary
<i>Primary/Intermediate System Heat Transport</i>	Primary
<i>Structural Response</i>	Primary
<i>Inherent Reactivity Feedback</i>	Primary
<i>Passive Heat Removal</i>	Primary
<i>Source Term</i>	Supporting: Determines timing, magnitude, and location of fuel failure.

*Primary capability indicates results are produced directly by SAS4A/SASSYS-1, while supporting indicates SAS4A/SASSYS-1 results are utilized in subsequent calculations by a separate tool.

2.2 ORNL SQA Recommendations

During the initial stages of the RTDP code qualification project, an ORNL effort [9] was completed that identified relevant guidance documents and standards, highlighted SQA best

practices of qualified² codes, and provided recommendations on a path forward for SAS4A/SASSYS-1 SQA. The report characterized the alternative SQA methodologies that are dependent on code use (i.e. by the licensee within the license application or by the regulator as a confirmatory tool), and described activities key to satisfaction of both pathways. Key findings of [9] that are directly relevant to the SAS4A/SASSYS-1 SQA effort are summarized here. Additional details on SQA pathways and the corresponding requirements can be found in Section 2.3.

The insights and recommendations provided in [9] form the primary basis of the SAS4A/SASSYS-1 SQA Program framework. A high level overview of SQA best practices common to most code SQA programs and guidance/requirements documents is provided in [9]. All of these activities are included in, or are planned to be incorporated into, the SAS4A/SASSYS-1 SQA Program. These activities include:

- Requirements must be clearly defined for both functional performance and application readiness;
- All processes (e.g. project management, design, implementation, configuration management, validation, verification, noncompliance monitoring) must be documented and utilized;
- A configuration management strategy must be established and utilized for all source code, test suites, test results, and documentation;
- Source code must be verified, either manually or automatically, to confirm that models are implemented correctly and that source code is written correctly and operates correctly on all supported platforms;
- Simulations of supported applications must be validated against experimental data;
- Noncompliance and resulting actions must be documented; and
- The entire quality assurance (QA) program must be routinely audited to confirm compliance.

As discussed in [9], the NRC formally endorses the requirements of the American Society of Mechanical Engineers (ASME) NQA-1-2008 and the NQA-1a-2009 Addenda (collectively referred to as NQA-1-2008/2009)³ [17, 18] in Regulatory Guide 1.28, “Quality Assurance Program Criteria (Design and Construction)” [19] as being acceptable for compliance with 10 CFR 50 Appendix B, where the CFR delineates QA requirements for nuclear power and fuel reprocessing plants. For software not compliant with the NQA-1-2008/2009 standard, it is required that the software be dedicated via a process compliant with the requirements of Part II,

² In this context, “qualified” indicates that the code is accepted as qualified by the relevant regulating body. Codes discussed in [9] include those used as either NRC or DOE confirmatory tools; the codes discussed in [9] have not undergone the commercial-grade dedication process and are therefore not considered to be “qualified” for use by a license applicant.

³ Requirements specific to software are provided in Part II, Subpart 2.7, “Quality Assurance Requirements for Computer Software for Nuclear Facility Applications” of the standard.

Subpart 2.14, “Quality Assurance Requirements for Commercial Grade Items and Services” of NQA-1-2008/2009.⁴

In light of the NRC-accepted methods and requirements, and given the development history of SAS4A/SASSYS-1 which classifies it as a legacy code, [9] recommended three key objectives in development of a qualified program for SAS4A/SASSYS-1. These objectives are listed below and described in the following subsections.

1. Evaluate the QA applied to the code during its development, the suitability of existing documentation, and identify limitations, vulnerabilities, and areas for improvement and the resources needed to address them.
2. Establish an NQA-1-2008/2009 compliant SQA program for future maintenance and development.
3. Develop the documentation base needed to support commercial grade dedication.

In addition to the items listed above, [9] also emphasized the need for an increased user base and open user community. An expanded user base assists with quality improvement, as bug identification and reporting are increased due to additional users exercising the code in an expanded operating envelope. An open user community facilitates information exchange and also encourages growth in the user base.

While not a formal finding of [9], one additional conclusion can be derived: there is no singular framework or blueprint for satisfying SQA requirements. Of all SQA programs detailed in [9], the exact structures of the SQA framework were not identical. However, all programs addressed the key SQA activities discussed above.

2.2.1 Program Evaluation

The objective of evaluation is to provide documented evidence of limitations and vulnerabilities of the code and its SQA program and to identify and prioritize these gaps. This process requires review of the supporting SQA documentation to determine its compliance with an accepted standard (e.g. IEEE, ASME, etc.), as well as review of any documented software requirements to determine code acceptability for its intended use. As per DOE-EH-4.2.1.2-Criteria⁵ [21], the evaluation should occur as follows:

1. Establish SQA criteria for the code;
2. Provide requested documentation, and estimate resources needed to address deficiencies;
3. Review SQA documentation and evaluate the software against the SQA criteria;

⁴ Detailed guidance for licensees and suppliers on commercial dedication is provided in The Electric Power Research Institute (EPRI) Technical Report 3002002289, “Plant Engineering: Guidelines for the Acceptance of Commercial-Grade Design and Analysis Computer Programs Used in Nuclear Safety-Related Applications” [20].

⁵ DOE-EH-4.2.1.2-Criteria, “Software Quality Assurance Plan and Criteria for the Safety Analysis Toolbox Codes,” released in 2003, provides a plan for qualification of prominent safety analysis codes, such as MACCS2, MELCOR, and CFAST, which had an uncertain SQA pedigree. While this document does not address the NRC licensing process, it provides a useful evaluation framework for SAS4A/SASSYS-1 SQA Program review.

4. Document code review in gap analysis reports;
5. Determine minimum required actions to be taken before software will meet SQA criteria;
6. Update SQA documentation and software to address deficiencies or make improvements;
7. Identify upgraded computer code version for DOE users to central registry; and
8. Provide configuration management and control of qualified software.

In development of the SAS4A/SASSYS-1 SQA Program, elements of the above evaluation process were adopted as possible. Relevant SQA criteria, discussed in Section 2.3 and Section 3, were first identified, and a framework satisfying these criteria was developed. Given that SAS4A/SASSYS-1 development efforts historically did not utilize a formal SQA program, evaluation of the acceptability of the SQA Program effectively occurred concurrently with Program development. In an effort to produce a meaningful and effective SQA Program within existing programmatic limits, some elements of the relevant SQA criteria were omitted. These omitted items or activities are documented in Sections 5.2 and 5.3, and resolution of these issues will occur on a prioritized basis.

2.2.2 NQA-1-2008/2009 Compliant Program

Included in [9] is the recommendation to create an NQA-1-2008/2009 compliant program for all future development and maintenance activities. Accordingly, initial SAS4A/SASSYS-1 SQA Program planning was performed with the goal of meeting Part II, Subpart 2.7 requirements; a crosswalk mapping elements of the SAS4A/SASSYS-1 SQA Program with Part II, Subpart 2.7 requirements can be found in Table A.1. Key elements of an NQA-1-2008/2009 program, as described by [9], include:

- Creation of standardized documentation including: SQA plan, configuration management plan, and various procedures;
- Implementation of a modernized configuration management and bug tracking system;
- Development of software requirements and software design documentation, including definition and assessment of functional needs and performance requirements; and
- Development of a comprehensive verification and validation (V&V) test suite which includes unit, integral, regression, and validation test problems.

It is important to note however that while an attempt was made to address all elements of Part II, Subpart 2.7, gaps in the SAS4A/SASSYS-1 SQA Program are known to exist; these gaps are discussed in Sections 5.2 and 5.3. Additionally, a program is not considered to be NQA-1-2008/2009 compliant until it is audited and approved by a certified reviewer. As such, resources within the DOE R&D framework will need to be identified such that SQA Program gap closure and an NQA-1 audit (and the support required for such an activity) can be completed.

2.2.3 Commercial Grade Dedication

Supporting commercial grade dedication requires the development of an extensive documentation base that addresses the quality of critical characteristics and product attributes. Detailed guidance on the commercial grade dedication of design and analysis software is provided in EPRI TR 3002002289 [20]. As these critical characteristics and attributes are measures of the acceptability

of models or features specific to each application or reactor design, the scope of required documentation and testing can generally be reduced to manageable segments. The entirety of capabilities and features of the code need not be documented to complete the commercial grade dedication process.

However, given the expected role of SAS4A/SASSYS-1 as a key safety analysis tool for a wide range of phenomena (see Table 2.1), it is anticipated that an initial commercial dedication process would require the documentation and qualification of a significant portion of critical characteristics. Efforts to address the requirements of EPRI TR 3002002289 have not been completed at this time, and it is expected that they would not be initiated until after provisional Program implementation is completed. Identification and prioritization of key critical characteristics anticipated to be of significance to a license applicant would assist in completion of this objective. Typically this process involves development of a Phenomena Identification and Ranking Table (PIRT) which identifies key phenomena and models required for successful simulation of a specific scenario. In this case, identification of a candidate code user and plant design would be required to complete this step.

2.3 SQA Methodologies and Requirements

This section delineates the various SQA methodologies and provides high level information on the requirements of each. In the case of a commercial operating license, a license applicant will perform a range of safety analyses for inclusion in the application. The NRC will then review the safety analyses presented by the licensee to confirm the validity of the results. In most cases, the NRC and the license applicant will use independent tools or methods.⁶ As such, there are typically two distinct roles a safety analysis tool may fulfill: a license application tool for use by the applicant, or a confirmatory analysis tool for use by the NRC. Accordingly, the SQA requirements vary per use of the software; Figure 2.1 highlights the alternative uses and the relevant standards and guidance documents. Summary information on the contents of these guidance and requirements documents and their relationship to a SAS4A/SASSYS-1 SQA program can be found in [9].

⁶ Selected tools may be used by both the license applicant and the NRC in certain circumstances.

Figure 2.1: Guidance and Requirements Documents per SQA Methodology

In the case of a license application tool, it is required that software be developed under an ASME NQA-1-2008/2009 framework in accordance with Part II, Subpart 2.7. If a tool does not have an NQA-1-2008/2009 pedigree, then commercial-grade dedication of the tool must occur as per NQA-1-2008/2009, Part II, Subpart 2.14. Guidance for licensees on the commercial dedication process can be found in EPRI TR 3002002289; supporting guidance on this process can also be found in numerous IEEE standards. As SAS4A/SASSYS-1 is not currently NQA-1-2008/2009 compliant, it is likely that a license applicant would utilize this commercial-grade dedication process.

If the software is utilized by the NRC as a confirmatory tool, then NRC-issued guidance governs the SQA process. NUREG/BR-0167, “Software Quality Assurance Program and Guidelines” [22], provides guidance to NRC organizations and contractors on the development and maintenance of software used by NRC staff and acts as an overarching document on NRC SQA requirements. Guidance on SQA of NRC thermal hydraulic tools is found in NUREG-1737, “Software Quality Assurance Procedures for NRC Thermal Hydraulic Codes” [23]. This document defines the best practices in satisfying the life cycle activities described in NUREG/BR-0167 and provides recommendations on preparation of SQA plans and checklists used to ensure compliance with NRC guidelines. Lastly, specific guidance on validation activities is provided in NUREG/CR-5249, “Quantifying Reactor Safety Margins” [24]; this document describes the code scaling, applicability, and uncertainty (CSAU) methodology that is to be applied to analyses of specific events and phenomena as a validation activity.

It should be noted that DOE-licensed facilities are not required to meet NRC guidelines. Instead, DOE-issued guidance in DOE-EH-4.2.1.2-Criteria [21], DOE Order 414.D [25], and DOE Guide 414.1-4 [26] should be utilized if a DOE license is sought. In an effort to provide preliminary support to use of SAS4A/SASSYS-1 in a DOE license application, requirements of DOE G 414.1-4 have been considered when developing the SQA Program framework; however,

establishing support for use of SAS4A/SASSYS-1 in a commercial license application is the primary objective of this effort.

Despite the existence of unique requirements and guidance documents, both SQA methodologies utilize a common set of SQA best practices, including: configuration control and identification; documentation supporting traceability of activities; rigorous verification testing; and regular surveillance to ensure compliance with respective SQA requirements. Validation activities are also very similar between methodologies; models and features relevant to the analysis must undergo a series of documented reviews. A listing of SQA activities common to both methodologies is provided in Table 2.2. This listing is meant to represent high-level topics that are crosscutting, and does not indicate minor deviations in the satisfaction of each requirement. In an effort to maximize the efficiency and broaden the applicability of the SAS4A/SASSYS-1 SQA Program, the majority of elements listed in Table 2.2 were addressed when developing the Program framework, as these elements are common to both methodologies.

Table 2.2: SQA Activities Common to NQA-1-2008/2009 and NUREG/BR-0167

Activity	NQA-1-2008/2009 Part II, Subpart 2.7	NUREG/BR-0167
<i>Software Project Management</i>	Part I, Requirement 1: Organization	Section 5: Project Management, Section 4.2: Software Project Plan
<i>QA Program and Plan</i>	Part I, Requirement 2: QA Program	Section 8: Quality Assessment and Improvement
<i>Configuration Management</i>	Section 203: Software Configuration Management	Section 6: Configuration Management
<i>Software Requirements Specification</i>	Section 401: Software Design Requirements	Section 2.2: Requirements Definition
<i>Software Design Documentation</i>	Section 402: Software Design	Section 2.3: Design
<i>Implementation</i>	Section 403: Implementation	Section 2.4: Implementation
<i>Testing</i>	Section 404: Acceptance Testing	Section 2.5: Qualifications Testing
<i>Installation and Acceptance</i>	Section 404: Acceptance Testing	Section 2.6: Installation and Acceptance
<i>Operation and Maintenance</i>	Section 405: Operation, Section 406: Maintenance	Section 2.7: Operations and Sustaining Engineering
<i>Verification and Validation</i>	Section 402.1: Software Design Verification, Section 404: Acceptance Testing	Section 3: Verification and Validation
<i>Audit and Assessment</i>	Part I, Requirement 18: Audits	Section 8: Quality Assessment and Improvement
<i>Retirement</i>	Section 407: Retirements	Section 2.8: Retirement and Archiving
<i>Problem Reporting and Corrective Action</i>	Section 204: Problem Reporting and Corrective Action	Section 7: Nonconformance Reporting and Corrective Action

3 Provisional Program Structure

The provisional SAS4A/SASSYS-1 SQA Program structure has been developed with the goal of satisfying the requirements highlighted in Section 2.3. Because it is expected that this effort will occur over a period of years, a phased approach to SQA Program development is being executed to leverage the resources available at each phase and to reduce any obstacles to efficient SQA Program sustainment. As a result, Phase 1 of SQA Program development is focused on satisfying the key, common points of the two primary SQA methodologies outlined in Section 2.3, and highlighted in Table 2.2. Accordingly, it can be expected that the provisional SQA Program will contain several known gaps when measured against either SQA methodology. Future efforts will focus on resolving these gaps.

Because SQA is heavily reliant on implementation and adherence to a series of documented plans and procedures, the remainder of this section will discuss the structure of the SQA Program, with focus on Program documentation. A hierarchical document structure has been adopted for the SAS4A/SASSYS-1 SQA Program, with a central SQA Plan (SQAP) comprising the top level of the Program. A brief description of each document is included below, and a graphical hierarchy is provided in Figure 3.1. Additional details on each document can be found in the following subsections.

- **Software Quality Assurance Plan (SQAP):** Acts as an entry point to the SQA Program and directs developers, managers, and SQA coordinators to the appropriate sublevel document. Provides an overview of all Program activities and requirements.
- **Configuration Management Plan (CMP):** A second-tier document. Describes configuration management (CM), the process of identifying, managing, and controlling the status and revision of software items. These activities are inherent in all procedures.
- **Coding Standards:** A second-tier document. Contains requirements for programming practices and conventions. Identifies relevant standards. Inherent to all SQA activities.
- **Procedures:** Third-tier documents. The following procedures are used in the SAS4A/SASSYS-1 SQA Program:
 - **Software Testing:** Prescribes steps for evaluating whether software adequately performs all intended functions.
 - **Error Reporting and Corrective Actions:** Prescribes steps for identifying, reporting, and correcting errors.
 - **Software Development and Modification:** Prescribes steps for the design and implementation of new features or models.
 - **Version Release:** Prescribes steps for the formal release of software and delivery of products.

Figure 3.1: Graphical Representation of SQA Program Hierarchy

The SQA activities highlighted in Table 1 of [9] were explicitly considered when developing the SAS4A/SASSYS-1 SQA Program framework. It should be noted that while the activities listed in Table 1 of [9] represent a full spectrum of SQA activities, this list is not exhaustive when comparing all existing SQA requirements set forth by all agencies (e.g. ASME, NRC, DOE, IEEE, etc.). However, this list is considered sufficient for Phase 1 of SQA Program development. A crosswalk outlining compliance of the SAS4A/SASSYS-1 SQA Program structure with NQA-1-2008/2009, NUREG/BR-0167, and DOE G. 414.1-4 requirements for key SQA activities is provided in Table A.1.

3.1 Software Quality Assurance Plan

The Program’s approach to SQA and an overview of all SQA Program activities and requirements are provided in the SQAP. This top-tier document acts as an entry point to the SQA Program and directs developers, managers/leaders, and SQA coordinators to the appropriate sublevel document for all Program activities. The primary purpose of the SQAP is to delineate the SAS4A/SASSYS-1 SQA Program framework by describing Program activities, organization, and documentation such that the interconnection of all Program items and activities is clearly defined.

The specific contents of the SAS4A/SASSYS-1 SQAP have been derived to address the requirements of the licensee and confirmatory tool SQA approaches. The requirements in IEEE Std 730-2002, “IEEE Std for Software Quality Assurance Plans,” which is recommended in [20], were referenced in order to support the commercial dedication process that may be pursued by a license applicant. When considering the requirements of a regulatory confirmatory tool, the NRC’s recommendations for a Software Project Plan [22] were utilized when developing the SQAP. In an effort to provide preliminary support to pursuit of a DOE license, elements of [26] were also adopted for the SQAP. An outline of the contents of the SQAP is provided in Table 3.1.

Table 3.1: Contents of SAS4A/SASSYS-1 SQAP

1.	<i>Introduction</i>
1.1.	<i>Project Background</i>
1.2.	<i>Purpose and Scope of Application</i>
1.3.	<i>Definitions</i>
1.4.	<i>Documentation</i>
1.5.	<i>Plan Maintenance</i>
2.	<i>Roles and Responsibilities</i>
3.	<i>Standards, Practices, and Conventions</i>
4.	<i>Software Work Activities</i>
4.1.	<i>Software Project Management and Quality Planning</i>
4.2.	<i>Software Risk Management</i>
4.3.	<i>Software Configuration Management</i>
4.4.	<i>Procurement and Supplier Management</i>
4.5.	<i>Software Requirements Identification and Management</i>
4.6.	<i>Software Design and Implementation</i>
4.7.	<i>Software Verification, Validation, and Testing</i>
4.8.	<i>Error Reporting and Corrective Actions</i>
4.9.	<i>Training</i>

A crosswalk outlining congruence of the SAS4A/SASSYS-1 SQAP with IEEE Std 730-2002, NUREG/BR-0167, and DOE G 414.1-4 requirements is provided in Table A.2. Note that in the case of NUREG/BR-0167, compliance with the Software Project Plan (SPP) is almost exclusively considered, as it is most representative of an SQAP. Elements of NUREG/BR-0167 beyond the SPP, (e.g. V&V, error reporting and corrective actions, documentation, training, etc.) are also included in the SAS4A/SASSYS-1 SQAP.

3.2 Configuration Management Plan

The CMP describes the processes required for control of all configuration items in the SAS4A/SASSYS-1 SQA Program. This second-tier document provides details on the various CM activities that are inherent in all procedures, including revisioning of configuration items, repository access permissions, comprehensive status accounting, configuration reviews, identification of configuration items, and formal release and delivery of products.

Similar to the SQAP, the contents of the CMP have been developed with the goal of addressing the licensee and confirmatory tool SQA requirements. The requirements in IEEE Std 828-2005, “IEEE Standard for Software Configuration Management Plans” [27] were referenced to provide support to the commercial dedication process. When considering satisfaction of the NRC SQA guidelines, the NRC’s recommendations regarding configuration management [22] were utilized. An outline of the contents of the CMP is provided in Table 3.2. A crosswalk outlining congruence of the SAS4A/SASSYS-1 CMP with the requirements and guidelines in IEEE Std 828-2005 and NUREG/BR-0167 is provided in Table A.3.

Table 3.2: Contents of SAS4A/SASSYS-1 CMP

1.	<i>Introduction</i>
1.1.	<i>Purpose</i>
1.2.	<i>Scope of Application</i>
1.3.	<i>Definitions</i>
1.4.	<i>Reference Documents</i>
1.5.	<i>Plan Maintenance</i>
2.	<i>Roles and Responsibilities</i>
3.	<i>SCM Activities</i>
3.1.	<i>Configuration Identification</i>
3.2.	<i>Configuration Control</i>
3.3.	<i>Configuration Status Accounting</i>
3.4.	<i>Configuration Evaluation and Review</i>
3.5.	<i>Interface Control</i>
3.6.	<i>Release Management and Delivery</i>

3.3 Coding Standards

The Coding Standards (CS) document outlines the coding standards and programming practices which developers are required to follow. These requirements apply to all development and modification activities; changes to an established baseline cannot occur unless they adhere to this document.

As the SAS4A/SASSYS-1 source code and its accompanying utilities contain both Fortran and C/C++, the appropriate ISO/IEC standards for these languages are identified in this document. The CS document also contains an abbreviated list of required programming practices. These requirements, which are largely associated with programming style, are imposed primarily to improve code readability and reduce coding errors.

3.4 Procedures

Procedures establish the processes, or workflows, that must be followed to meet SQA requirements. A fully developed SQA plan may include numerous procedures. The goal in the present work is to define high-priority procedures that are required for specific software development activities. These activities include the following:

- Software Testing
- Error Reporting and Corrective Actions
- Software Development and Modification
- Version Release

The procedures for the above activities are summarized in the sub-sections that follow. Procedures for additional activities will be developed as required.

While the purpose of procedures is to define the workflow for software development, testing, and release, the main outputs of procedures are the QA records that document the completion of tasks

in the procedure workflow and that provide traceability for future review or audits. SAS4A/SASSYS-1 is maintained in a Subversion source code repository, which tracks all changes to the source code. Likewise, a Trac *environment*⁷ is used to provide a ticket system that documents the completion of procedure tasks. Although each procedure is unique, a generalized Trac workflow has been defined that can be applied to each procedure. The workflow is shown in Figure 3.2.

The Trac ticket system has been configured with four key review *states* that a ticket may be in during the execution of a procedure's workflow. Review states, along with the two outcomes of a review, are highlighted in blue in Figure 3.2 and cover the following areas:

- Software Requirements Specifications (SRS)
- Software Design Documentation (SDD)
- Implementation (e.g. of code or documentation)
- Quality Assurance

Not all procedures will require a review in all of the above areas. For example, the *Error Reporting and Corrective Actions* procedure is intended to apply to faults identified in the implementation (of code or documentation) that deviates from the SRS and/or SDD. For that procedure, a review would be required only for the implementation of the correction followed by a QA review to ensure all documentation in the ticket is complete and provides a traceable record.

In addition to ticket states, the Trac ticket system has been configured with different ticket *types* that correspond to the procedure being invoked. Ticket types can be changed in the course of following a procedure. For example, if an "Error Report" ticket is created and the evaluation of the error reveals that there are deficiencies in the SRS or SDD, then the ticket can be changed to a "Development" type that corresponds to the *Software Development and Modification* procedure, which requires reviews of the SRS and SDD.

⁷ <https://trac.edgewall.org/wiki/1.1/TracEnvironment>

Figure 3.2: Ticket Workflow Defined for the SAS4A/SASSYS-1 Trac Environment

3.4.1 Software Development and Modification

The *Software Development and Modification* procedure establishes the process for developing and implementing new functional capabilities or modifying or removing existing functional capabilities within the SAS4A/SASSYS-1 safety analysis software. Functional capabilities are defined by project planning activities that take place outside the scope of this procedure and are documented according to DOE project management practices or in sponsored research contracts with external organizations.

Key activities in this procedure include writing software requirements, specifying software design, implementing code and documentation, testing, and integration. All four review states in Figure 3.2 must be completed and documented in the Trac ticket.

This procedure does not apply to the treatment of errors that are reported in existing software releases and related documentation.

3.4.2 Error Reporting and Corrective Actions

The *Error Reporting and Corrective Actions* procedure establishes the process for reporting errors associated with the SAS4A/SASSYS-1 safety analysis software and for taking corrective actions to resolve those errors. Errors associated with the software may include documentation errors or numerical, logical, functional, or structural flaws in the source code or generated output files. Corrective actions are those activities carried out to reduce or eliminate, to the extent practical, these errors.

A key activity in this procedure is verifying the reproducibility of the error and identifying if it is a code, documentation, or possibly user error. Corrections can then be made in the code or documentation to reduce or eliminate the error. Technical and QA reviews must be completed and documented in the Trac ticket. Affected users must be notified of the error, and mitigation strategies for avoidance of the problem prior to issue resolution must also be provided to the affected group.

This procedure does not apply to the development of new functional capabilities or to the extension or removal of existing functional capabilities.

3.4.3 Software Testing

The *Software Testing* procedure establishes a process for evaluating whether the SAS4A/SASSYS-1 software adequately performs all intended functions. This includes testing new software units and performing regression testing on previously tested coding.

Key activities in this procedure include: identifying the software features and associated software units to be tested; identifying test cases and acceptance criteria to be used; verifying the input data; and performing unit and regression testing using the interactive testing tool (described in Section 4.2). An independent, cognizant staff member must complete a technical review of the testing, and document the findings in the Trac ticket.

3.4.4 Version Release

The *Version Release* procedure establishes a process for formal release of versions of the SAS4A/SASSYS-1 software and related utilities.

Key activities in this procedure include: initiating a Release ticket (via Trac) for the version of the code to be released; performing release verification testing; creating a release package; authorizing release of the code version; and distributing the code. An independent staff member must complete a Technical review of the release verification testing and document the results in the Trac ticket.

4 SQA Program Activities

In addition to the creation of a provisional SQA Program framework and its supporting documentation, development of the SAS4A/SASSYS-1 SQA Program entails a series of planning and technical activities. The current status of SQA Program implementation planning is reviewed in Section 4.1. Technical activities completed during 2016 are highlighted in Section 4.2.

4.1 Implementation Planning

A staged approach to SQA Program implementation is required to enable a smooth transition to Program compliance. This staged approach, highlighted in Figure 4.1, entails four phases, where the final phase is expected to be an ongoing process. All of these activities will occur within Phase 2 of the project, meaning they will be completed during the current FY. Additional details on these stages are provided in the remainder of this subsection.

Figure 4.1: Preliminary SQA Program Implementation Stages

The initial stage, implementation planning, primarily entails the development of an implementation plan which delineates specific tasks, associated personnel, and the anticipated completion date. The implementation plan describes the progression through the final three stages of implementation. Currently, the SAS4A/SASSYS-1 SQA Program is in the implementation planning period.

Upon completion of the implementation plan, the SQA Program enters a pre-implementation period. A key activity within this period is the training of personnel on the use of procedures, the SVN configuration management system, and the Trac ticket system. Training materials to support this activity will need to be developed. A set of documentation, separate from the material described in Section 3, will need to be developed to support this stage. This includes various checklists and inventories that help with identification of personnel involved with development, responsible personnel and associated timeframes for tasks, and software baseline inventories.

At this point, the SQA Program enters a brief provisional implementation period, where the ultimate goal of this stage is to move toward self-assessment and SQA Program revision as per the assessment. Additional training, checklists or forms, or revisions to the SQA documentation in Section 3 may be required. It is in this final stage, which is expected to be cyclical and ongoing, that long term surveillance and self-assessment schedules, programmatic priorities and needs, and critical characteristics will be identified. The latter two items, programmatic priorities and critical characteristics, will be categorized to assist with planning of future SQA Program goals and needs as improvements are warranted.

4.2 Technical Activities

Throughout Phase 1, quality enhancing technical activities have been completed consistent with the recommendations in [9] to increase the size of the user base. The focus in Phase 1 included expansion of the V&V test suite, development and implementation of automated and interactive testing algorithms, and successful compilation and analysis with an alternative compiler. A brief summary of the technical activities is provided below; select details on these tasks are provided in the remainder of this subsection.

- Addition of 6 test cases to the approved V&V test suite. These test cases address previous gaps in the suite, which were primarily in measured control signals.
- Development and implementation of automated testing via a Buildbot server. This automated functionality tests code repository functionality and integrity by performing a series of checkout, build, and simulation activities which are used to report any errors in the repository. Currently, only OS X is supported.
- Development and implementation of an interactive testing algorithm that facilitates development and maintenance activities. Primarily used as a developer tool, the algorithm will execute a series of user-defined cases for various executables and identify any errors or discrepancies in the output. Currently, only OS X is supported.
- Successful build and testing of the source code with an alternative compiler (NAG). In this process, several areas of the source code were improved by the addition of more robust arithmetic error checks, removal of nonstandard functions, and correction of errant type definitions.

4.2.1 V&V Test Suite Expansion

A new set of test cases has been created to extend the existing SAS4A/SASSYS-1 V&V test suite. The new cases focus on the control system, which allows users to track a large variety of plant variables during a simulation, process those parameters in mathematical blocks, and implement a response in the simulation. The first set of problems examined the 22 block signals, which includes simple operations such as summation and multiplication, as well as more complicated operations such as the JK flip-flop and velocity limiters. Another set of recently completed test cases focuses on measured signals for heat transport system parameters. A similar set of cases is currently being developed for a recently added feature that allows access to measured signals for core channel parameters. The control signals, which impose responses such as reactivity insertions or changes in pump torque, will be tested in upcoming cases.

4.2.2 Development and Implementation of Additional Testing Algorithms

The automated and interactive testing algorithms were developed to support regression and V&V testing activities that are required in the development and maintenance processes. Both testing algorithms utilize a base Python script (`run_Tests.py`) which enables execution and comparison of a series of test cases completed using various compilers. In the case of the automated algorithm, a Buildbot server was configured to execute the Python script on a nightly basis. Buildbot will checkout the main development branch in the Subversion (SVN) SAS4A/SASSYS-1 source code repository, compile the source code, and call the Python testing script. Results of the Buildbot process are provided via a web portal in the form of a standardized summary page. The Buildbot

web portal also enables users to obtain additional information on the status and output of each step (e.g. checkout, compile, etc.), if requested. The interactive testing algorithm utilizes a bash shell script rather than Buildbot to execute the base Python script. The shell script allows for additional, developer-specific customizations to be implemented using command line arguments. A diagram depicting the flow and options of the interactive testing process is shown in Figure 4.2.

Figure 4.2: Diagram of Interactive Testing Process

4.2.3 Improved Compiler Support

Recent SAS4A/SASSYS-1 development activities have primarily supported the Intel Fortran compiler when building the source code. Successful compilation with an alternative compiler aids in quality improvement as errors in the source code, or even weaknesses in compilers, may be discovered as alternative build algorithms are utilized. Additionally, different compilers impose differing requirements on adherence to language standards, meaning alternative compilers can uncover outdated and problematic coding structures. As such, SAS4A/SASSYS-1 was compiled with the NAG Fortran compiler, which is maintained by the Numerical Algorithms Group. In order to accomplish this, various modifications to the source code were required, where the majority of issues were related to use of nonstandard functions or errors in type declarations.

Following successful build with the NAG compiler, a series of tests were completed, with the goal of comparing and addressing any discrepancies in output results. The SAS4A/SASSYS-1 V&V test suite was executed using source code compiled with both Intel and NAG compilers. This effort resulted in the improvement of various subroutines due to the identification of the need for additional arithmetic error checking and refinement of variable type declarations. In

general, however, comparison of numerical output for the two compilers found that most discrepancies in results were negligible.

5 Path Forward

This section provides additional information on activities planned for the remainder of this FY (Section 5.1) and the longer term gaps that will still need to be addressed (Sections 5.2 and 5.3) within the SAS4A/SASSYS-1 SQA Program. While not specifically stated in the content that follows, continued SQA Program maintenance and compliance are assumed to be performed for the lifetime of the code. While the SAS4A/SASSYS-1 SQA Program aims to avoid being overly burdensome and hinder maintenance and development activities, additional resources and lead times will be required to comply with the SQA Program. While SAS4A/SASSYS-1 previously utilized an informal design and review process, configuration management protocols, and some documentation requirements, planning for future development activities will require resources beyond what was historically available.

Maturation of the SAS4A/SASSYS-1 SQA Program will require significant additional Program development in the near term. A graphical depiction of the SAS4A/SASSYS-1 SQA Program development needs is provided in Figure 5.1. This report summarizes the work completed in Phase 1 of the figure. Section 5.1 provides an overview of activities planned in Phase 2; the implementation discussed in Section 4.1 will also occur during Phase 2. As indicated in the figure, Section 5.2 discusses the SQA Program development phase which is required to achieve compliance with a suitable SQA pathway. Section 5.3 describes how the effort to develop the SQA Program can be systematically reduced during a gap resolution phase. Following this phase, some level of resources is required for long-term SQA Program maintenance.

Figure 5.1: SAS4A/SASSYS-1 SQA Program Development Needs

5.1 Planned Activities

Several key program and technical gaps will be addressed during Phase 2 of this effort, where the objective is to develop a stable and sustainable SAS4A/SASSYS-1 SQA Program that can be exercised for other SFR safety analysis codes. While completion of these activities will enhance the SAS4A/SASSYS-1 SQA Program, it is important to note that these tasks represent a fraction of the activities required to develop a fully functional and regulator-accepted SQA program. A listing of planned activities is provided below, with additional details on these activities included in the remainder of the subsection:

- Finalize and implement the provisional SQA Program;
- Obtain feedback from and interface with other safety software managers regarding development of an SQA program for additional Argonne-developed safety codes;
- Complete internal SAS4A/SASSYS-1 SQA Program audit to identify and prioritize gaps; and
- Perform additional technical activities, as resources allow: improve test suite, expand platform support, etc.

Finalization and implementation of the provisional SQA Program is a pivotal task in Phase 2 of the project. Accordingly, final review and refinement of the provisional Program structure is a priority task. This activity primarily entails review of SQA Program documentation to ensure consistency among documents and compliance with the proposed SQA framework. At this point, further configuration of supporting SQA tools (e.g. the SVN source code repository and Trac ticket system) may be necessary to support steps in SQA Program procedures. Upon finalization of the SQA Program structure and support tools, implementation of the Program can begin. A staged implementation is planned to accommodate an effective transition to a compliant SQA Program. Implementation of the provisional SQA Program (which is relatively simple and minimal) is anticipated to require approximately 3 to 4 months, where stages are focused on documentation finalization, training, trial application, and informal audits. Additional details on implementation can be found in Section 4.1.

Following successful SQA Program implementation, the remaining planned activities will be pursued in parallel. To facilitate application of the SAS4A/SASSYS-1 SQA Program framework to additional safety analysis tools (e.g. neutronic codes such as MC²-3, DIF3D-Variant, PERSENT, GAMSOR, etc.), a series of planning and review activities will be held with the managers of these codes, with consideration of input from end-user stakeholders. The objectives of these activities are to obtain additional information on the structure of current development and maintenance activities of these codes and obtain feedback from code managers on the suitability and acceptability of an SQA framework adapted from SAS4A/SASSYS-1. This interface and feedback activity will be limited to a prioritized list of safety codes, and will likely require future resources to support SFR development and deployment.

Concurrent with the previous interface activity, an informal audit of the SAS4A/SASSYS-1 SQA Program will be performed by an internal resource. Due to the recent establishment of numerous NQA-1 compliant programs in the Nuclear Engineering Division, several Division staff members certified as NQA-1 auditors may be available to provide an informal review of the provisional SAS4A/SASSYS-1 SQA Program. It is important to note that in this activity, this would not entail a full NQA-1 audit, and therefore the SAS4A/SASSYS-1 SQA Program would not become NQA-1 compliant following this review. Instead, the objective of this activity would be to identify significant gaps in the SAS4A/SASSYS-1 SQA Program that may result from a lapse in satisfaction of NQA-1 or regulator-derived requirements such that these gaps may be prioritized and addressed in the near term.

Beyond the completion of the previously discussed program activities, several key technical activities will be pursued. These activities include the improvement of the approved V&V test suite and addition of improved platform support. While it is expected that update of the V&V test

suite will be an ongoing activity that continues throughout the lifetime of the SAS4A/SASSYS-1 project, near term update of the test suite will provide increased support of regression and acceptance testing. Unit, integral, and validation tests for prioritized models (e.g. thermal hydraulics, reactivity feedback, etc.) will be developed. Coverage (a measure of the number of lines or modules exercised by a series of input decks) of the test suite will also be increased by the addition of test cases, where significant coverage allows for more robust testing.

Increased platform support will be completed via expansion of the automated and interactive testing algorithms to additional platforms, such as Windows and/or Linux with the x86 architecture. Support for platform-specific testing assists with quality improvement as platform-specific errors can be detected more easily. Research aides or student interns will be utilized to complete these technical activities as possible in an effort to continue to grow the SAS4A/SASSYS-1 user and developer base.

5.2 Near Term Required Program Development

This section provides an overview of prioritized gaps in the SAS4A/SASSYS-1 SQA Program that are expected to remain at the end of FY17. These activities, listed in Table 5.1, represent tasks that are of relatively high importance with regard to required program development, and therefore, should be completed in the near term in order to build an acceptable SQA program. Table 5.1 provides a summary of key SQA Program gaps, and identifies the relative importance and expected lead time to resolution. Additional details are provided in the remainder of this subsection.

Table 5.1: Prioritized SAS4A/SASSYS-1 SQA Program Gaps

Gap	Importance	Lead Time	Comment
<i>Interaction with regulator</i>	High	Ongoing	Supports NRC's Near-Term Implementation Action Plans [28].
<i>Requirements and design documentation</i>	High	Long	Key critical characteristics can be addressed in the near term.
<i>Code verification and model validation documentation</i>	High	Medium	Key critical characteristics can be addressed in the near term.
<i>Validation and verification test suite</i>	Medium	Lifetime Activity	Test suite improvement expected to occur throughout lifetime of project.
<i>Development of documentation supporting commercial grade dedication</i>	Medium	Ongoing	Supported by resolution of gaps in requirements, design, and V&V documentation and test suite.

In light of the NRC's recently released Non-LWR Near-Term Implementation Action Plans [28] which highlights actionable technical objectives for achieving regulatory readiness, interaction with the regulator on the acceptability of the SAS4A/SASSYS-1 SQA Program is identified as a high priority task. Establishing an agreed upon code qualification strategy with the regulator,

which would directly support Strategy 2 of [28], is considered to be an ongoing process, as stages of a qualification strategy may be developed or updated as license applicants begin to move forward with their respective designs. However, it is expected that a significant portion of the qualification path can be established and resolved early in this process if key critical characteristics common to most domestic SFR designs are addressed early on. Completion of these industry support and regulatory readiness efforts also directly supports the objectives of DOE's "Vision and Strategy for the Development and Deployment of Advanced Reactors" [29].

Development of requirements and design documentation is an essential component to both SQA methodologies. Software requirements specifications are intended to comprehensively describe the functional and performance requirements (e.g. the code must model a core with hexcan geometry, or the code must complete the simulation in approximately real-time) of the software to be developed. An SRS is typically completed prior to software design and development, and is meant to be a contractual agreement between a contractor and sponsor on the expected capabilities of the code. In the case of SAS4A/SASSYS-1, an SRS must be developed that encompasses the entirety of the existing code as no software requirements are formally defined for the code. Software design documentation provides an overall description of the software architecture to be developed that complies with the SRS. The design of data structures, architecture, interface, and procedures (e.g. program structure) must all be included in the SDD. The SAS4A/SASSYS-1 User's Guide loosely satisfies some requirements of an SDD, as the Guide describes the phenomenological models implemented in the code. Despite the large volume of information available in the User's Guide, significant gaps in the design description of SAS4A/SASSYS-1 still exist (e.g. documentation of acceptable input ranges for critical characteristics, updated documentation of code architecture, etc.). For a large, integrated systems tool like SAS4A/SASSYS-1, identification of a facility design can assist with documentation generation and gap reduction. Given a specific plant, critical software characteristics and models can be identified and prioritized for inclusion in requirements and design documents. In the absence of a candidate design, key critical characteristics common to most domestic SFR designs can be addressed, which would significantly reduce this gap.

Code verification and validation documentation, like the requirements and design documentation, is critical to both SQA pathways. Adequate V&V documentation provides evidence that the coding and models were implemented properly. Similar to requirements and design documentation, identification of a facility design for licensing consideration will assist with reducing the scope of this task, but significant progress on gap closure can be accomplished in lieu of a design by focusing on key critical characteristics common to most SFR designs.

As discussed in Section 5.1, addition of verification and validation test problems is expected to occur throughout the lifetime of the software project. Introduction of new standardized problems supports acceptance testing requirements as new input is developed to confirm the codes and models implemented via maintenance and modernization activities. While the addition of test problems that exercise key models is expected to be performed in the near term, it is likely that some gaps may still remain in the test suite's coverage of prevalent subroutines and functions. Near term efforts should seek to address these gaps first, while tests that exercise models utilized less frequently (e.g. oxide fuel failure models) should then be developed.

Resolution of the final gap in Table 5.1, documentation supporting commercial-grade dedication, can be facilitated by closure of the previously discussed documentation and V&V test suite gaps. As highlighted in Tables 6-2 through 6-5 of [20], the documentation supporting commercial dedication focuses on key functional attributes and critical characteristics (e.g. platform portability, accuracy, applicability of models, etc.), and is fairly similar to software requirements, design, and V&V documentation. Similar to interaction with the regulator, this activity can be expected to be an ongoing process as industry members approach and engage the regulator on the acceptability of their safety analysis tools.

5.3 Longer Term Gap Resolution

This section provides a high level overview of long term SAS4A/SASSYS-1 SQA Program gaps that are expected to remain following completion of the items described in Section 5.2. These gaps are identified as being in addition to, or a subcomponent of, those discussed in Section 5.2. These gaps are considered to be of a lower priority relative to those discussed in Section 5.2. A brief listing of these long term gaps is provided in Table 5.2, with additional discussion in the remainder of this subsection.

Table 5.2: Long Term SAS4A/SASSYS-1 SQA Program Gaps

Gap	Importance	Lead Time	Comment
<i>Completion of requirements and design documentation for all models</i>	Medium	Medium	Continuation of effort discussed in Section 5.2.
<i>Completion of code verification and model validation documentation for all models</i>	Medium	Medium	Continuation of effort discussed in Section 5.2.
<i>Continued V&V test suite development for all models</i>	Low	Medium	Continuation of effort discussed in Section 5.2.

The gaps identified in Table 5.2 relate to the resolution of documentation gaps discussed in Section 5.2. At this stage in the SQA program development, it is assumed that documentation supporting the majority of critical characteristics has been developed, meaning documentation gaps remain only in low priority models that are used less frequently in a safety analysis. Completion of the SRS, SDD, and V&V documentation for legacy code and models should require a relatively reduced lead time as the scope would be greatly reduced. Development of V&V test problems for these legacy models would also be continued over the long term.

6 Works Cited

- [1] U.S. Nuclear Regulatory Commission, "Safety Evaluation Report Related to Operation of Fast Flux Test Facility," NUREG-0358, 1979.
- [2] Project Management Corporation, "Clinch River Breeder Reactor Project - Preliminary Safety Analysis Report," 1976.
- [3] U.S. Nuclear Regulatory Commission, "Safety Evaluation Report Related to the Construction of the Clinch River Breeder Reactor Plant," NUREG-0968, 1983.
- [4] U.S. Nuclear Regulatory Commission, "Preapplication Safety Evaluation Report for the Sodium Advanced Fast Reactor (SAFR) Liquid-Metal Reactor," NUREG-1369, 1991.
- [5] U.S. Nuclear Regulatory Commission, "Preapplication Safety Evaluation Report for the Power Reactor Innovative Small Module (PRISM) Liquid-Metal Reactor," NUREG-1368, 1994.
- [6] M. Denman, J. LaChance, T. Sofu, *et al.*, "Sodium Fast Reactor Safety and Licensing Research Plan - Volume I," SAND2012-4260, 2012.
- [7] Idaho National Laboratory, "Advanced Reactor Technology - Reactor Technology Development Plan (RTDP)," INL/EXT-14-32837, 2015.
- [8] A. J. Brunett and L. L. Briggs, Unpublished Information, 2015.
- [9] N. R. Brown, W. D. Pointer, M. T. Sieger, *et al.*, "Qualification of Simulation Software for Safety Assessment of Sodium-Cooled Fast Reactors: Requirements and Recommendations," Oak Ridge National Laboratory, ORNL/TM-2016/80, 2016.
- [10] Editor: Fanning, T. H., "The SAS4A/SASSYS-1 Safety Analysis Code System: User's Guide," Argonne National Laboratory, ANL/NE-12/4, 2012.
- [11] J. E. Cahalan, A. M. Tentner, and E. E. Morris, "Advanced LMR Safety Analysis Capabilities in the SASSYS-1 and SAS4A Computer Codes," in *Proceedings of the International Topical Meeting on Advanced Reactor Safety, Vol. 2*, Pittsburgh, PA, 1994, pp. 1038-1045.
- [12] J. C. Carter *et al.*, "SAS1A, A Computer Code for the Analysis of Fast-Reactor Power and Flow Transients," Argonne National Laboratory, ANL-7607, 1970.
- [13] F. E. Dunn *et al.*, "The SAS2A LMFBR Accident-Analysis Computer Code," Argonne National Laboratory, ANL-8138, 1974.
- [14] M. G. Stevenson *et al.*, "Current Status and Experimental Basis of the SAS LMFBR Accident Analysis Code System," in *Proceedings of the Fast Reactor Safety Meeting*, Beverly Hills, CA, 1974, pp. 1303-1321.
- [15] C. E. Till, "Fast Reactor Directions: The LMR Integral Fast Reactor Program at Argonne," in *Proceedings of the International Fast Reactor Safety Meeting*, Snowbird, UT, 1990.
- [16] J. Sackett *et al.*, "EBR-II Test Programs," in *Proceedings of the International Fast Reactor Safety Meeting*, Snowbird, UT, 1990.
- [17] American Society of Mechanical Engineers, "Quality Assurance Requirements for Nuclear Facility Applications," ASME NQA-1-2008, 2008.
- [18] American Society of Mechanical Engineers, "Addenda to ASME NQA-1-2008: Quality Assurance Requirements for Nuclear Facility Applications," NQA-1a-2009, 2009.
- [19] U.S. Nuclear Regulatory Commission, "Quality Assurance Program Criteria (Design and Construction)," Regulatory Guide 1.28, Rev. 4, 2010.
- [20] Electric Power Research Institute, "Plant Engineering: Guideline for the Acceptance of Commercial-Grade Design and Analysis Computer Programs Used in Nuclear Safety-Related Applications: Revision 1 of 1025243," EPRI TR 3002002289, 2013.

- [21] U.S. Department of Energy, "Software Quality Assurance Plan and Criteria for the Safety Analysis Toolbox Does," DOE-EH-4.2.1.2-Criteria, 2003.
- [22] U.S. Nuclear Regulatory Commission, "Software Quality Assurance Program and Guidelines," NUREG/BR-0167, 1993.
- [23] F. Odar, "Software Quality Assurance Procedures for NRC Thermal Hydraulic Codes," U.S. Nuclear Regulatory Commission, NUREG-1737, 2000.
- [24] B. Boyack, R. Duffey, P. Griffith, *et al.*, "Quantifying Reactor Safety Margins: Application of Code Scaling, Applicability, and Uncertainty Evaluation Methodology to a Large-Break, Loss-of-Coolant Accident," U.S. Nuclear Regulatory Commission, NUREG/CR-5249, EGG-2552, R4, 1989.
- [25] U.S. Department of Energy, "Quality Assurance," DOE O 414.1D, 2011.
- [26] U.S. Department of Energy, "Safety Software Guide for Use with 10 CFR 830 Subpart A, Quality Assurance Requirements, and DOE O 414.1C, Quality Assurance," DOE G 414.1-4, 2005.
- [27] Institute of Electrical and Electronics Engineers, "IEEE Standard for Software Configuration Management Plans," Software Engineering Standards Committee of the IEEE Computer Society, IEEE Std 828-2005, 2005.
- [28] U.S. Nuclear Regulatory Commission, "NRC Non-Light Water Reactor (Non-LWR) Vision and Strategy - Staff Report: Near-Term Implementation Action Plans," 2016.
- [29] U.S. Department of Energy, "Vision and Strategy for the Development and Deployment of Advanced Reactors," DOE/NE-0147, 2017.

Appendix A

Table A.1: Crosswalk of SQA Activities with Requirements and SAS4A/SASSYS-1 SQA Program

SQA Activity	NQA-1-2008/2009 Part II, Subpart 2.7	NUREG/BR-0167	DOE G 414.1-4	SAS4A/SASSYS-1 SQA Program
<i>Software Project Management</i>	Part I, Requirement 1: Organization	Section 5: Project Management, Section 4.2: Software Project Plan	Activity 1: Software Project Management and Quality Planning	SQAP: Section 4.1
<i>QA Program & Plan</i>	Par I, Requirement 2: QA Program	Section 8: Quality Assessment and Improvement		SQAP
<i>Software Project Risk Management</i>		Section 5.8: Risk Management	Activity 2: Software Risk Management	SQAP: Section 4.2
<i>Configuration Management</i>	Section 203: Software Configuration Management	Section 6: Configuration Management	Activity 3: Software Configuration Management	SQAP: Section 4.3, CMP
<i>Procurement & Supplier Management</i>	Section 300: Software Acquisition		Activity 4: Procurement and Supplier Management	SQAP: Section 4.4
<i>Requirements</i>	Section 401: Software Design Requirements	Section 2.2: Requirements Definition	Activity 5: Software Requirements Identification and Management	SQAP: Section 4.5, SAS4A/SASSYS-1 User's Guide: Chapters 2-16
<i>Design</i>	Section 402: Software Design	Section 2.3: Design, Section 4.4: Design Documentation	Activity 6: Software Design and Implementation	SQAP: Section 4.6, SAS4A/SASSYS-1 User's Guide: Chapters 2-16
<i>Implementation</i>	Section 403: Implementation	Section 2.4: Implementation, Section 2.5: Implementation Documentation	Activity 6: Software Design and Implementation	SQAP: Section 4.6, SAS4A/SASSYS-1 User's Guide: Chapters 2-16

Table A.1: Crosswalk of SQA Activities with Requirements and SAS4A/SASSYS-1 SQA Program cont'd

SQA Activity	NQA-1-2008/2009 Part II, Subpart 2.7	NUREG/BR-0167	DOE G 414.1-4	SAS4A/SASSYS-1 SQA Program
<i>Support Software</i>	Section 600: Support Software			CMP: Section 3.5
<i>Testing</i>	Section 404: Acceptance Testing	Section 2.5: Qualification Testing	Activity 8: Verification and Validation	SQAP: Section 4.7, Software Testing Procedure
<i>Reviews</i>	Section 202: Review			Feature Change Procedure
<i>Installation & Acceptance</i>	Section 404: Acceptance Testing	Section 2.6: Installation and Acceptance	Activity 3: Software Configuration Management	Software Testing Procedure, Feature Change Procedure
<i>Operation & Maintenance</i>	Section 405: Operation, Section 406: Maintenance	Section 2.7: Operations and Sustaining Engineering	Activity 3: Software Configuration Management, Activity 9: Problem Reporting and Corrective Action	Error Reporting and Corrective Actions Procedure
<i>Verification & Validation</i>	Section 402.1: Software Design Verification, Section 404: Acceptance Testing	Section 3: V&V, Section 4.6: V&V Documentation	Activity 8: Verification and Validation	SQAP: Section 4.7
<i>Problem Reporting & Corrective Action</i>	Section 204: Problem Reporting and Corrective Action	Section 7: Nonconformance Reporting and Corrective Action	Activity 9: Problem Reporting and Corrective Action	SQAP: Section 4.8, Error Reporting and Corrective Actions Procedure

Table A.1: Crosswalk of SQA Activities with Requirements and SAS4A/SASSYS-1 SQA Program cont'd

SQA Activity	NQA-1-2008/2009 Part II, Subpart 2.7	NUREG/BR-0167	DOE G 414.1-4	SAS4A/SASSYS-1 SQA Program
<i>Training</i>			Activity 10: Training of Personnel in the Design, Development, Use, and Evaluation of Safety Software	SQAP: Section 4.9
<i>Documentation</i>	Section 201: Documentation	Section 3.1.1, 3.1.2, 3.2.1, 3.5.1, 3.6.3, 3.6.5, 5.1, 6.0, Appendix A, Appendix B	Activities 1-10	SQAP, CMP, and all supporting procedures
<i>Audit & Assessment</i>	Part I, Requirement 18: Audits			CMP: Section 3.4
<i>Retirement</i>	Section 407: Retirement			N/A

Table A.2: Crosswalk of SQAP Elements with Requirements

IEEE Std 730-2002	NUREG/BR-0167: Software Project Plan (SPP)	DOE G 414.1-4	SAS4A/SASSYS-1 SQAP
Section 4.1: Purpose	SPP Section 1.2: Plan Scope and Organization		Section 1.2: Purpose and Scope of Application
Section 4.2: Reference Documents			Section 1.4: Documentation
Section 4.3: Management	SPP Section 2.3: Organization, Tasks, and Responsibilities	Activity 1: Software Project Management and Quality Planning	Section 2: Roles and Responsibilities
Section 4.4: Documentation	SPP Sections 2.6, 3.2, 3.3, 3.4	Activities 3, 5, 6, 8, 9	Section 1.4: Documentation
Section 4.5: Standards, Practices, Conventions, and Metrics	SPP Section 3.4: Quality Assessment and Improvement Approach, NUREG Section 5.2.3: Methodology, Standards, and Procedures		Section 3: Standards, Practices, and Conventions
Section 4.6: Software Reviews	SPP Section 3.4: Quality Assessment and Improvement Approach	Activity 8: Verification and Validation	Section 4.7: Software Verification, Validation, and Testing
Section 4.7: Test	SPP Section 3.2: Verification and Validation Approach, NUREG Section 3: Verification and Validation	Activity 8: Verification and Validation	Section 4.7: Software Verification, Validation, and Testing
Section 4.8: Problem Reporting and Corrective Action	SPP Section 3.3: Nonconformance Reporting and Corrective Action, NUREG Section 7: Nonconformance Reporting and Corrective Action	Activity 9: Problem Reporting and Corrective Action	Section 4.8: Error Reporting and Corrective Actions
Section 4.9: Tools, Techniques, and Methodologies	NUREG Sections 3.3, 4.10, 5.9, 6.8, 7.4		Section 3: Standards, Practices, and Conventions, Section 4.3: Software Configuration Management

Table A.2: Crosswalk of SQAP Elements with Requirements cont'd

IEEE Std 730-2002	NUREG/BR-0167: Software Project Plan (SPP)	DOE G 414.1-4	SAS4A/SASSYS-1 SQAP
Section 4.10: Media Control	NUREG Section 4: Documentation and Deliverables		Section 1.4: Documentation, Section 4.3: Software Configuration Management
Section 4.11: Supplier Control	NUREG Section 5.4: Supplier Control	Activity 4: Procurement and Supplier Management	Section 4.4: Procurement and Supplier Management
Section 4.12: Records Collection, Maintenance, and Retention	NUREG Section 8.5: Quality Records Collection, Maintenance, and Retention		Section 1.4: Documentation, Section 4.3: Software Configuration Management
Section 4.13: Training	NUREG Section 5.7: Training	Activity 10: Training Personnel in the Design, Development, Use, and Evaluation of Safety Software	Section 4.9: Training
Section 4.14: Risk Management	NUREG Section 5.8: Risk Management	Activity 2: Software Risk Management	Section 4.2: Software Risk Management
Section 4.15: Glossary			Section 1.3: Definitions
Section 4.16: SQAP Change Procedure and History	SPP Section 1.3: Plan Maintenance		Section 1.5: Plan Maintenance

Table A.3: Crosswalk of CMP Elements with Requirements

IEEE Std 828-2005	NUREG/BR-0167	SAS4A/SASSYS-1 CMP
Section 3.1: Introduction		Section 1: Introduction
Section 3.2.1: Organization	Section 5.2: Project Planning and Organization	Section 2: Roles and Responsibilities
Section 3.2.2: SCM Responsibilities	Section 5.2: Project Planning and Organization	Section 2: Roles and Responsibilities
Section 3.2.3: Applicable Policies, Directive, and Procedures		Section 1.4: Reference Documents
Section 3.2.4: Management of the SCM Process	Section 5: Project Management	Section 2: Roles and Responsibilities
Section 3.3.1: Configuration Identification	Section 6.2: Baselines	Section 3.1: Configuration Identification
Section 3.3.2: Configuration Control	Section 6.3: Change Control, Section 6.6: Software, Access, and Media Control	Section 3.2: Configuration Control
Section 3.3.3: Configuration Status Accounting	Section 6.4: Status of Baselines and Changes	Section 3.3: Configuration Status Accounting
Section 3.3.4: Configuration Evaluation and Reviews	Section 6.7: Configuration Audits	Section 3.4: Configuration Evaluation and Review

Table A.3: Crosswalk of CMP Elements with Requirements cont'd

IEEE Std 828-2005	NUREG/BR-0167	SAS4A/SASSYS-1 CMP
Section 3.3.5: Interface Control		Section 3.5: Interface Control
Section 3.3.6: Subcontractor/Vendor Control	Section 5.4: Supplier Control	See Section 4.4 of SQAP
Section 3.3.7: Release Management and Delivery		Section 3.6: Release Management and Delivery
Section 3.4: SCM Schedules		Sections 1.5 and 3.4
Section 3.5: SCM Resources		
Section 3.6: SCM Plan Maintenance		Section 1.5: Plan Maintenance

Nuclear Engineering Division

Argonne National Laboratory
9700 South Cass Avenue, Bldg. 208
Argonne, IL 60439-4842

www.anl.gov

**U.S. DEPARTMENT OF
ENERGY**

Argonne National Laboratory is a U.S. Department of Energy
laboratory managed by UChicago Argonne, LLC