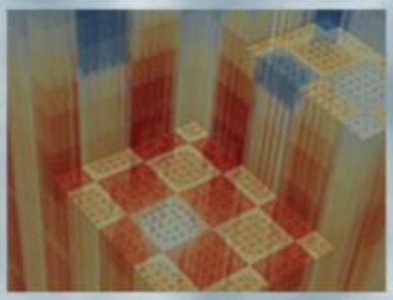
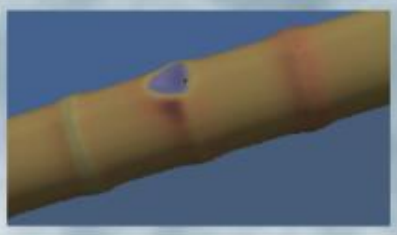
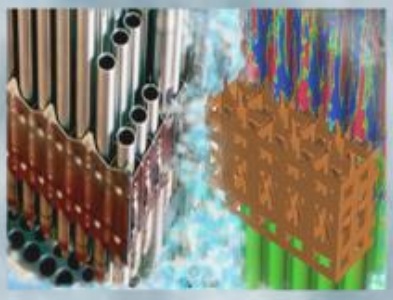
CASL-U-2019-1878-002







Consortium for Advanced Simulation of LWRs

VERAIO Software Management Plan

Scott Palmtag[1] and Erik Walker[2]

[1]North Carolina State University
[2]Oak Ridge National Laboratory

3/3/2021

VERA QA Document – This document was prepared under the ORNL VERA Quality Assurance Program in accordance with procedure VERA-QA-003. The OFFICIAL COPY of this document is the electronic version in the VERA Documents Repository. Before using a printed copy, verify that it is the most current version by checking the Revision ID against the electronic version.

U.S. DEPARTMENT OF ENERGY | Nuclear Energy

Revision Log

Revision	Date	Affected Pages	Revision Description
0	8/5/2019	All	Initial Release
1	11/8/2019	iii	Updated PSM
2	3/3/2021	2,4,10,11,13	Revised for VERA 4.2 release

Document pages that are:

Export Controlled:	None
IP/Proprietary/NDA Controlled:	None
Sensitive Controlled:	None
Unlimited:	All

VERAIO Software Management Plan

Approvals:

Erik Walker Digitally signed by Erik Walker Date: 2021.05.21 16:28:46 -04'00' 5/21/2021

Erik Walker, VERAIO Product Software Manager **Date**

Aaron Graham Digitally signed by Aaron Graham Date: 2021.05.21 16:30:49 -04'00' 5/21/2021

Aaron Graham, Independent Reviewer **Date**

BENJAMIN COLLINS (Affiliate) Digitally signed by BENJAMIN COLLINS (Affiliate) Date: 2021.05.24 18:16:33 -04'00' 5/24/2021

Benjamin Collins, Program Technical Leader **Date**

Tristan Grover Digitally signed by Tristan Grover Date: 2021.05.25 15:01:29 -04'00' 5/25/2021

Tristan Grover, VERA Software Quality Coordinator **Date**

Acronyms

CASL Consortium for the Advanced Simulation of Light Water Reactors

GUI graphical user interface

HPC high performance computing

ORNL Oak Ridge National Laboratory

PHI physics integration

PSM product software manager

QA quality assurance

QAPP Quality Assurance Program Plan

SBMS Standards Based Management System

SHA-1 Secure Hash Algorithm 1

SQA software quality assurance

SQAP software quality assurance program

VERA Virtual Environment for Reactor Applications

V&V verification and validation

Contents

1. Project Background and Objectives

VERAIO is a set of utility codes used to provide a common set of input and outputs to the Virtual Environment for Reactor Applications (VERA). VERA is a collection of several different computer codes that all have a common input and output. This prevents the need to manage input and output from each individual code, allowing for ease of use and reducing errors associated with code operability. The VERAIO utilities include VERAIn, VERAView, and VERARun. Each of these utilities is described below.

VERAIn is an input processor that reads an ASCII input file generated by users, parses the file, performs some error checking, and writes an XML file to be read by other VERA codes. The main purpose of VERAIn is to provide a common input to all of the VERA codes so users only need to learn one input. VERAIn is written in Perl and uses YAML configuration files to provide flexibility.

VERAView is a graphical user interface (GUI) that reads a VERA HDF output file and allows users to visualize results. VERAView is written in Python.

VERARun is a script that drives the VERA execution in a high performance computing (HPC) environment.

Work performed at the code level supports the *Quality Assurance Program Plan (QAPP)* (VERA-QA-001), and *VERA Software Quality Assurance Plan* (VERA-QA-002).

2. Plan Description

The VERAIO code team is responsible for implementing software engineering and software quality assurance (SQA) requirements under its control as defined in this plan. Compliance with this plan is required throughout the software lifecycle, which includes planning, requirements, acquisition, design, implementation, acceptance testing, maintenance and operations, and retirement.

The purpose of this document is to define the VERAIO processes applicable the VERA project. VERAIO is a set of stand-alone codes, and multiple organizations are associated with VERAIO development.

The scope of application of this plan is the VERAIO configuration items, which are discussed Chapter 8. Software that does not affect the performance of the VERA product suite is not within the scope of this plan.

2.1 Assumptions and Constraints

- Adequate funding, required hardware, and system software is available to complete planned VERAIO activities.
- Responsibilities can be delegated for a temporary period to alternate personnel; however, the personnel designated in this plan retain ultimate responsibility. The designees must have the appropriate knowledge, required training, and required independence as identified in the "Training" section of this plan.
- The hardware that serves VERAIO applications is managed by the Oak Ridge National Laboratory (ORNL) infrastructure team. Management of this hardware is outside the scope of this plan.

3. References

The following VERA requirements documents apply to development of this plan:

VERA-QA-001, VERA Quality Assurance Program Plan (QAPP)

VERA-QA-002, VERA Software Quality Assurance Plan (SQAP)

VERA-QA-003, VERA Management of Controlled Documents and Records

VERA-QA-004, VERA Software Configuration and Control

VERA-QA-005, VERA Users Manual and Theory Manual

VERA-QA-006, VERA Requirements and Design Process

VERA-QA-009, VERA Release Process

VERA-QA-010, VERA Problem Reporting and Corrective Action

4. Definitions

The following terminology is used in this plan. Acronyms are defined with their first use in this plan. Defined terms and document titles are denoted in this plan by italics font. Plan section or appendix titles are denoted by quotation marks.

Baseline: a specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for the use and further development, and that can be changed only by using an approved change control process. Source: ASME NQA-1a-2009

Branch: also referred to as *topic branch*. Refers to a branch in the VERAIO Git repository that is not the master branch. More information can be found at https://guides.github.com.

Code repository: also referred to as *VERAIO repository*, *VERAIO code repository*, and simply *repository*. It is the repository that stores all source code, tests, configuration instructions, and documentation. Version control is performed using the Git version control system (see https://guides.github.com).

Commit: a specific configuration of the code repository that is identified by a unique Secure Hash Algorithm 1 (SHA-1) number. See https://guides.github.com for more information.

Gitlab: internal ORNL and University of Michigan website for hosting the VERAIO code repository, running automated continuous integration tests, and submitting and performing merge request reviews. See https://about.gitlab.com for more information.

Ticket: a code development activity that is tracked through the stages of development, including planning, development, and review. After tickets are closed, they are archived indefinitely. Tickets are tracked on the external ORNL Gitlab website.

Merge request: a request to merge new code changes into the baseline version of the code. A merge request is submitted by the developer to an independent reviewer, who must approve changes before they can be merged. Additionally, the new changeset must pass all automated tests prior to their merge.

5. Quality Management

5.1 Organization

Each VERAIO code team member shall adhere to this plan. The VERAIO product software manager (PSM) retains overall responsibility for ensuring adherence to this plan. High-level VERA product suite design requirements are made at the VERA leadership level and flow down to the VERA PSMs in the form of individual code requirements. VERAIO code development, review, and testing activities are conducted by a small group of VERAIO code developers and independent reviewers under the supervision of the PSM. After the code is approved at the code level, additional reviews and tests are conducted at the VERA product suite level using processes defined in the VERA software quality assurance program (SQAP).

5.2 Roles and Responsibilities

Table 1 identifies the roles and responsibilities for VERAIO software development and management activities.

Table 1: Roles and responsibilities

Role	Responsibilities
VERA program technical leader	• Approve this plan. • Define high-level software design requirements for the VERA product suite and VERAIO. • Determine and/or approve the software quality level and software type. • Provide concurrence on initial software requirements documentation and all subsequent software requirement changes that are considered a major software change. • Ensure that code team members are properly qualified and trained per this plan. • Identify VERA product suite software risks and methods to mitigate those risks associated with the VERA product suite release.

Table 1 – *Continued from previous page*

Role	Responsibilities
VERA SQA coordinator	• Provide concurrence on this plan. • Provide concurrence on initial software requirements documentation and all subsequent software requirement changes that are considered a major software change. • Ensure the appropriate level of rigor for the SQA activities. • Provide oversight to ensure that identified software problem reporting and corrective actions are implemented per this plan. • Verify that actions have been completed to properly retire the software.
VERAIO PSM	• Perform system management and administration activities. • Manage code release through the VERA product suite release process. • Hold responsibility for administration and execution of this plan. • Provide task-specific training and mentoring for code team members per this plan. • Ensure that code software risks are mitigated as specified in this plan. • Address credible code software failures through verification and validation (V&V) activities. • Ensure that software configuration items are managed per this plan. • Acquire materials and services per this plan. • Assign access and authorities for VERAIO repositories. • Review and approve software requirements and design documentation materials generated by code developers. • Ensure proper handling of code change requests. • Define and delineate work activities for code development and testing. • Ensure that software problems are documented and dispositioned per procedure VERA-QA-010, *Problem Reporting and Corrective Action.* • Manage code software retirement when needed.

Table 1 – *Continued from previous page*

Role	Responsibilities
Independent technical reviewer	• Review this plan and changes to this plan. • Review requirements and design documentation. • Review and approve change requests (includes new feature implementation). • Review and approve test results. • Review and approve documentation changes associated with change requests. • Perform additional review activities as assigned by the PSM.
Software developer	• Perform and document design, coding, and unit testing of software code. • Perform additional development and testing activities as assigned by the PSM.
Records Management Officer	• Ensure that the code and documentation baseline is maintained. • Ensure that identified code records are scheduled and maintained. • Ensure that documents are numbered and controlled in accordance with Standards Based Management System (SBMS) requirements. • Maintain a documents repository to ensure easy retrieval for use and future revisions.

6. Documentation and Records

All documents and records generated through this plan are managed per VERA-QA-003, *VERA Document Control and Records*. All VERAIO documentation is retained in the VERAIO code repository and is updated as necessary when modifications are made to VERAIO software. VERAIO change request reviews include documentation review. The lifecycle documentation listed below will be managed and maintained as quality assurance (QA) records for VERAIO:

- VERAIO Software Management Plan
- VERAView User's Manual
- VERAView Software Test Plan, Requirements, and Test Report
- VERAView Programmer's Manual
- VERAIn User's Manual
- VERAIn Programmer's Manual
- VERAIn Software Test Plan, Requirements, and Test Report
- Completed software QA checklists
- Software problem reports

The VERAIn manuals also apply to VERARun.

7. Software Risk Management

The software risk management processes described in VERA-QA-002, *VERA Software Quality Assurance Plan*, have been reviewed. The stated risks are applicable to the VERAIO product, and no additional software risks have been identified. The software work activities implemented by the code team will ensure that the software meets its intended function before being released to the nuclear industry.

7.1 Software Quality Level

Because VERAIO will be used for supporting nuclear reactor safety analysis, it is designated as SQL2. This plan defines how the following software-specific work activities will be implemented for the VERAIO product:

1. Configuration management
2. Software acquisition
3. Software requirements
4. Software design
5. Software verification and validation
6. Software problem reporting and corrective action
7. Training

7.2 Risk Management

Software risk management, which includes software failure analysis, provides a disciplined environment for proactive decision making to continuously assess what can go wrong, determine what risks are important to address, and implement actions to address those risks. VERAIO is in the maintenance phase of the development lifecycle, which requires frequent changes to support the next VERA product suite software release. The PSM is responsible for managing identified code-level risks that would potentially impact the successful completion of the next VERA product suite software release:

Introduction of code defects: mitigating factors include software design and in-process code reviews, training and qualification of code developers, and PSM mentoring and oversight

Inadequate resources for completion of software requirements implementation: mitigating factors include establishment of technical and resource-loaded performance parameters and milestones that are tracked and correlated with cost and schedule data

8. Software Configuration Management

Software configuration management will be performed in accordance with VERA-QA-004, *VERA Software Configuration and Control*. The software configuration item list can be found in the VERAIO Software Requirements, Test Plan and Test Report. The list includes unique, identifying versions of VERAIO and its supporting libraries and software, the names of the testing machines, and testing results.

9. Software Acquisition

Acquired software (including system software, software tools, and third-party libraries) is obtained through the physics integration (PHI) Infrastructure Team and is controlled under the ORNL SBMS Acquisition Management System.

Inclusion of acquired third-party libraries to support the functionality of a VERA software product are covered through the software lifecycle activities, including requirements definition, design reviews, testing, and verification and validation.

10. Software Engineering Method

The tasks delineated in the following subsections encompass the software engineering activities performed throughout the lifecycle of a software product. VERAIO is already developed, has operational history, and is being maintained: enhancements and new features are added occasionally.

10.1 Reviews

Reviews are conducted in accordance with VERA-QA-009, *VERA Release Process*.

10.2 Software Requirements

The functional requirements for VERAIn and VERARun are included in the *VERAIn Software Requirements, Test Plan, and Test Report*.

The functional requirements for VERAView are included in the *VERAView Software Requirements, Test Plan, and Test Report*.

10.2.1 Modeling Assumptions and Constraints

Code modeling assumptions and constraints are discussed in the respective VERAIO programmer's manuals and user's manuals.

10.2.2 Requirements Traceability

A custom-developed requirements traceability tool is used to link the unique requirements identifiers to tests that assess those requirements. This allows for both backwards and forwards traceability.

10.3 Software Design

Software design follows *VERA-QA-006, VERA Requirements and Design Process*. VERAIO does not have a single design document. Code design for code changes is discussed in Gitlab tickets. Gitlab tickets must be reviewed by an independent reviewer before they can be closed. Closed Gitlab tickets are archived

indefinitely. See the "Software Design Review" section of this plan for additional information on design reviews. A high-level overview of code organization is given in the programmer's manuals.

Design traceability is maintained by assigning a Gitlab ticket number to code changes and merge requests to describe the motivation, design, and testing related to the code changes.

10.4 Software Implementation and Control of Libraries

VERAIO software is implemented as outlined in the "Configuration Change Control" section of this plan using designated VERAIO coding practices. Third-party code (functional libraries, such as solvers) and data (non-functional) libraries are utilized by VERAIO and are obtained from the VERA development environment. Regardless of type, all libraries are managed and controlled by the PHI Infrastructure Team. VERAIO libraries are included on the "Support Software and Libraries Software Configuration Item List." See the "Coding Standards, Practices, and Conventions" section of this plan for additional information.

10.5 Software Testing

Testing is planned and conducted in accordance with *VERA-QA-006, VERA Requirements and Design Process* and VERA-QA-009, *VERA Release Process*.

10.6 Code Availability for VERA Product Suite Integration

The master branch of the VERAIO Gitlab site is always available for VERA product suite integration. The master branch is synced to the casl-dev repository nightly using an automated sync script. The sync script performs VERA product suite–level automated casl-dev system testing prior to syncing the VERAIO Gitlab site with the casl-dev repository. VERA product suite releases are made from the casl-dev repositories.

10.7 Security, Backup, and Recovery

Code access is controlled through the VERAIO Gitlab site. Access must be explicitly granted on a per-user basis by the PSM. Code backup and recovery services are managed by ORNL Information Technology Services.

10.8 Software Retirement

VERAIO is an active software product with ongoing support. If VERAIO is removed from distribution and support, then this section will be updated.

11. Verification and Validation

Verification and validation activities are planned and executed in accordance with *VERA-QA-006, VERA Requirements and Design Process* and VERA-QA-009, *VERA Release Process*.

12. Coding Standards, Practices, and Conventions

Coding standards, practices, and conventions are selected to meet code requirements, to facilitate software lifecycle activities, and to improve coding consistency, functionality, performance, and testability. VERAIO coding standards are documented in VERAIO programmer's manuals. The VERAIO PSM is responsible for managing the VERAIO programmer's manuals.

13. Software Tools and Support Software

Support software consists of software tools and system software. All VERAIO support software is managed and controlled by the PHI Infrastructure Team. The PSM determines the need for VERAIO support software changes and communicates the need to the PHI Infrastructure Team. The PSM assesses the potential impact on the code and establishes the required level of review and retesting. Changes to VERAIO support software are controlled under the VERA-level configuration management provisions.

VERAIO support software is included on the "Support Software and Libraries Software Configuration Item List."

14. Problem Reporting and Corrective Action

Problem reporting and corrective action follow VERA-QA-002, *VERA Software Quality Assurance Plan*, and VERA-QA-004, *VERA Software Configuration and Control*.

15. Training

Personnel performing or managing activities contained in this plan shall receive training for their job responsibilities. All roles identified in the "Roles and Responsibilities" section of this plan shall be qualified and trained as directed by the PSM. New software developers must review the VERA QAPP and associated procedures, this software management document, and the *VERAIO Style Guide*. The user support and training materials listed below will be developed and supplied as part of the VERAIO product release:

- VERAIO Software Management Plan
- VERAView User's Manual
- VERAView Software Test Plan, Requirements, and Test Report
- VERAView Programmer's Manual
- VERAIn User's Manual
- VERAIn Programmer's Manual
- VERAIn Software Test Plan, Requirements, and Test Report