CSAM 19-042

COG

Software Verification and Validation Plan (SVVP)

Revision 0

Document Status: Issued

Nuclear Criticality Safety Division

September 30, 2019

830 Software - Risk Level 3

Major Development Control

Auspices Statement

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Approvals

This plan is approved by the manager of each unit of the organization having responsibilities defined within this SVVP or their designated representatives.

Signatures on this page indicate agreement to support and/or follow this plan for the application development effort. Signatures also represent agreement that this plan lays out the necessary software engineering activities to meet software quality assurance requirements and that any justifications for non-applicability and/or tailoring are acceptable. Any modifications made to this document, other than for clarification purposes, will require a re-evaluation of the potential impact upon the development effort by the organizations represented on this signature page.

Prepared by: Chuck K Lee 9-30-2019

Chuck K. Lee
COG Software Developer

Date

Reviewed by: Edward M Lent 9-30-2019

Edward M. Lent
COG Software Developer

Date

Approved by: David P Heinrichs 9-30-2019

David P. Heinrichs
Division Leader, Nuclear Criticality Safety

Date

Revision History

Document Version	Revision Date	Originator(s)	Revision Description
0	09-30-2019	Chuck K. Lee	Initial Draft

Table of Contents

List of Tables

1 Purpose & Scope

This Software Verification and Validation Plan (SVVP) describes the activities the software effort performs in order to assure the software product satisfies its intended use and user needs. In addition, these activities assure that the software engineering processes described in the Software Quality Assurance and Software Configuration Management plans are followed. The verification process provides objective evidence that the software meets its requirements while the validation process provides the objective evidence that the software solves the intended problem. This plan applies to the full product lifecycle as described in the COG Software Quality Assurance Plan (SQAP).

2 References

The following are the applicable governing Federal Regulations, DOE Orders, and guidance documents:

- DOE O 414.1D Admin Chg 1, Quality Assurance
- 10 CFR 830, Subpart A, Nuclear Safety Management, Quality Assurance Requirements
- DOE O 200.1A, Information Technology
- DOE O 420.1C, Chg. 2, Facility Safety
- DOE-STD-3007-2017, Guidelines for Preparing Criticality Safety Evaluations at Department of Energy Non-Reactor Nuclear Facilities
- NAP-24A, Weapon Quality Policy

The following standards were used in the development of this document to flow down the requirements identified in RID-0116, 830 Software Quality Assurance Consensus Standards. Applicable requirements from these standards are addressed in this document. The IEEE standard was used as a guide; no claim to conformance to the entire standard is made. Only those parts cited in RID-0116 are required.

- ASME NQA-1-2008, Quality Assurance Requirements for Nuclear Facility Applications
- IEEE Std 1012™-2012, IEEE Standard for System and Software Verification and Validation

The following institutional documents are directly applicable to software, unless superseded by programs, standards, policies, procedures, or processes required by the organization:

- DES-0111, 830 Institutional Software Quality Assurance Program
- PRO-0107, Software Risk Grading
- PRO-0110, Identification, Documentation, Control, and Maintenance of the 830 Software Inventory

Additional documents referenced in this SVVP are:

- CSAM15-049, Rev. 1, COG Software Quality Assurance Plan (SQAP)
- CSAM15-046, Rev. 1, COG Software Configuration Management Plan (SCMP)
- CS-P-005, Safety Software Verification and Validation

- ANSI/ANS-8.24-2017, Validation of Neutron Transport Methods for Nuclear Criticality Safety Calculations

3 Definitions

Table 1, Acronyms and Definitions

Acronym	Definition
830 Software	Nuclear/Radiological Safety Software
Admin	Administrative
ANS	American Nuclear Society
ANSI	American National Standards Institute
ASME	American Society of Mechanical Engineers
CFR	Code of Federal Regulations
Chg	Change
CSAM	Criticality Safety Administrative Memorandum, LLNL
DC	Development Control
DOE	Department of Energy
DOE O	DOE Order
IEEE	Institute of Electrical and Electronics Engineers, Inc.
LLC	Limited Liability Company
LLNL	Lawrence Livermore National Laboratory
NAP	NNSA Policy
NNSA	National Nuclear Security Administration
NQA-1	Nuclear Quality Assurance, Quality Assurance Requirements for Nuclear Facility Applications
PRO	Procedure
Rev	Revision
RID	Requirements Interpretation Document
RL	Risk Level
SCMP	Software Configuration Management Plan
SQAP	Software Quality Assurance Plan
SRD	Software Requirements description
Std	Standard
SVVP	Software Verification and Validation Plan
V&V	Verification and Validation

4 V&V overview

4.1 Organization

The organizational structure used to execute and support the software effort and the quality requirements of this SVVP is found in Section 3.1 Organization, of the COG Software Quality Assurance Plan (SQAP).

4.2 Master schedule

A resource-loaded master schedule is available in the Five-Year Execution Plan for the US DOE Nuclear Criticality Safety Program, which is the principal sponsor of this software. Additional details are provided in quarterly progress reports (QPRs) for analytical methods provided to the US DOE NCSP Manager, NNSA, NA-511.

4.3 Resources Summary

The resources required to implement this plan are:

- COG code developers
- COG software repository manager
- COG responsible manager
- COG code users
- NCSD computer system coordinator

The institution provides LLNL with a Division Leader for Nuclear Criticality Safety who is also the responsible manager for 830 Software including COG software. COG code developers and the software repository manager are primarily programmatically funded under the auspices of the DOE Nuclear Criticality Safety Program[1]. COG code users are criticality safety engineers who are programmatically funded by several programs including the E-Program, N-Program, and W-Program.

4.4 Responsibilities

Table 2, V&V Activities and Responsible Individuals, below identifies the major verification and validation activities and the responsible individual(s). Depending upon the availability of personnel, one individual may perform more than one activity, or one activity may require the actions of more than one person. Different individuals may perform these activities at different times as assigned by the Software Effort Leader.

Table 2, V&V Activities and Responsible Individuals

Activity	Responsible Individual(s)
Interface with other processes	Code developers
Acquisition System Requirements Review	Responsible manager
Acquired Software Evaluation	Code developers
Hazard Analysis	Responsible manager
Security Analysis	Responsible manager
Traceability Analysis	Not applicable
Software Requirements Evaluation	Responsible manager
Design Evaluation	Code developers
Interface Analysis	Code developers
Source Code & Source Code Documentation Evaluation	Code developers, repository manager, responsible manager

[1] https://ncsp.llnl.gov

Software Test Plan V&V	Code users, responsible manager
Software Test Design V&V	Code developer
Software Test Procedure V&V	Repository manager; responsible manager
Software Test Execution	Repository manager; code users
Software Test Execution V&V	Repository manager; computer system coordinator
Software Installation & Checkout V&V	Repository manager; computer system coordinator
Software Operation V&V	Code users
Software Maintenance Task Iteration	Repository manager

4.5 Tools, Techniques and Methods

This software effort uses various tools, techniques, and methods in the V&V effort throughout the software lifecycle. These tools, techniques, and methods are listed and described in Section 9, Tools, techniques, and methods, of the COG Software Quality Assurance Plan. Refer to this document for further details.

4.6 Computer Program Test Records

Test documentation is an important quality record as it is the evidence that the testing activities occurred, and that the execution met requirements. While the content of the test records can vary depending on the test type, purpose, and application, there is a minimum set of information that is expected.

Verification testing assures that the software product satisfies the requirements for the software lifecycle phase at which it is being tested. At a minimum, the following information is included in the test records:

- Computer program tested
- Computer hardware tested
- Test equipment and calibrations, where applicable
- Date of test
- Tester or data recorder
- Simulation models used, where applicable
- Test problems
- Results and applicability
- Action taken in connection with any deviations noted
- Person evaluating test results

This testing is typically performed by the code developer during the research and development cycle while establishing prototypes (see SCMP, Fig. 3). Additional verification testing of safety features (k-eff, alpha, flux) is done by comparison to analytic benchmarks with exact or high precision numerical solutions. See https://cog.llnl.gov/sqa.php, Table II, "Test, Verification, and Validation Suites."

In-use testing is done after a software product is installed on a different platform or when there are significant changes in the operating system. At a minimum, the following information is included in the test records:

- Computer program tested
- Computer hardware tested
- Test equipment and calibrations, where applicable
- Date of test
- Tester or data recorder
- Acceptability

The requirements for platform-specific in-use installation and verification testing, and validation testing, and their documentation requirements, are specified in CS-P-005, "Safety Software Verification and Validation."

5 V&V Processes

5.1 Common V&V Processes

The activities noted in this section occur throughout the software lifecycle. The processes support other V&V activities by assuring that data exchanges between and within organizations are defined and processes are defined to assure acquired software has complete requirements and is properly evaluated prior to use.

5.1.1 Interface with Other Processes

V&V activities occur throughout the software lifecycle and need to be coordinated with the development, test, operations, maintenance, and other appropriate activities. When multiple organizations are involved, interactions between these organizations are coordinated. The data to be exchanged is identified and the exchange process addresses the review, approval, release, distribution, and revision of documents involving design interfaces.

The COG team participates in the US DOE NCSP Analytical Methods Working Group, which includes the MCNP (LANL) team and SCALE (ORNL) team in addition to other ad hoc participants, which have included the MORET (IRSN) team and TRIPOLI (CEA) team. This working group provides an auspices of information exchange including sharing lessons-learned and facilitating joint inter-comparison testing.

The Radiation Safety Information Computational Center (RSICC) at ORNL provides an interface between the COG code developers and all research and 830 (safety) users in addition to direct contact between the users and developers via https://cog.llnl.gov and cog@llnl.gov.

The major interface is to external databases such as neutron and photon cross-sections. The COG team interfaces with the nuclear data community during the development phases at activities such as Nuclear Data Week convened each November at Brookhaven National Laboratory. Nuclear data library databases are available from multiple sources including:

- LLNL Nuclear Data Group – ENDL, GNDS
- BNL Cross Section Evaluation Working Group – ENDF/B
- IAEA Nuclear Data Section – POINT
- OECD NEA Nuclear Databank – JEFF
- Japan Atomic Energy Agency – JENDL

5.1.2 Acquisition System Requirements Review

The acquisition system requirements are reviewed to

- Verify consistency with user needs
- Validate that the defined technologies, methods, and algorithms are capable of satisfying the requirements
- Verify that objective information that can be demonstrated by testing is provided

In addition, any other requirements, such as deliverable definitions, compliance standards, and regulations are reviewed for completeness, consistency, and accuracy.

For research software prototypes and baselines used in research, computer platform and infrastructure software is supplied as part of the LLNL institutional image. Development tools, such as compilers, IDEs, and code repositories, are supplied by the LLNL IT Division. These institutional divisions perform the required reviewed and approved for use process at that level. Acquisition system requirements beyond those supplied by the LLNL institutional image and the IT Division are specified in the Software Requirements Document for each baseline.

For 830 software baselines, the NCSD maintains hardware platforms and infrastructure software under configuration management as described in the SCMP and as part of the LFO approved Criticality Safety Program. Additional details are available in:

- CSAM15-046, Rev. 1, COG Software Configuration Management Plan (or latest revision)
- CS-P-000, Criticality Safety Program Description Document
- CS-P-005, Safety Software Verification and Validation
- CS-P-009, Archiving and Back-Up of Computer Data

5.1.3 Acquired Software Evaluation

Software that is acquired (e.g., freeware, shareware, procured commercial off-the-shelf, or otherwise acquired software) is evaluated and qualified for use. If the software meets the definition of 830 Software, then the qualification is performed per PRO-0110, Identification, Documentation, Control, and Maintenance of the 830 Software Inventory. This qualification evaluation includes, as a minimum:

- Identification of capabilities and limitations for intended use

- Test plans and tests required to demonstrate the capabilities within the limitations
- Instructions for use within the limits of capabilities

Very limited acquisition activities are planned for the COG software effort other than platform and development environment tools acquired or supplied by the LLNL institutional and IT divisions (see Section 5.1.2 of this SVVP). Therefore, this section does not typically apply. Any plans to acquire software are specified in the version specific Software Requirements Document together with a description of the intended use and plans for testing.

Instructions for use within the limits of capabilities will be published in the Manual or Manual Supplement.

5.2 Software V&V Processes

The hazard, security, and traceability analyses, described in the first three subsections, can be done during any phase of the software lifecycle. The processes described in the remainder of this section cover each phase of the software lifecycle, requirements through maintenance. For each process the V&V approach is defined, relevant documents to be reviewed are noted, the V&V process is performed, the results of the V&V effort are documented, and anomalies are recorded. The personnel performing the V&V effort, in consultation with the appropriate development personnel, will review anomalies to determine their severity. If it is a problem, the problem reporting process referenced in the SQAP and detailed in the SCMP is followed. For those anomalies that are change requests, the change request process described in the SCMP is followed. Any other anomaly is dispositioned as agreed to by those reviewing it. Such actions may include, but are not limited to, updating documentation, editing source code to follow defined standards, or clarifying the process and/or expected results.

NAP-24A additional considerations:

- Test requirements for new features are specified in the Software Requirements Document
- Test requirements for 830 (safety) features are specified in the Software Requirements Document
- Someone other than the developer (e.g., RSICC at Oak Ridge National Laboratory) performs the testing of that developer's software
- In the case of automated testing, the test scripts are designed by someone other than the developer (e.g., the ADVANCE Project at Brookhaven National Laboratory)

Software nonconformances (anomalies) may occur in the software quality engineering processes or in the product. Software V&V processes (including testing) are used to identify these nonconformances. CS-P-005, Appendix B, provides a Software Error/Corrective Action Report Form.

5.2.1 Hazard Analysis

An initial hazard analysis helps ensure safety requirements are defined and included in the overall set of software requirements. It also provides a baseline that is used for comparison purposes for future analyses.

Throughout the lifecycle, the goal of hazard analysis is to verify that the software is correctly implemented, and no new hazards have been introduced since the last analysis was done. At a minimum, it is important that the analysis is done after software modifications have been made to assure they are correctly implemented and do not introduce any new hazards. The tasks for this activity are to:

- Verify software modifications are correctly implemented and do not introduce new hazards
- Assess identified mitigation strategies to verify that the hazards are prevented, controlled, or minimized and any unmitigated hazards are documented and addressed as part of software operations

Prior to use of COG software in hazard analysis (i.e., criticality safety evaluations) installation and verification, and validation reports are prepared in accordance with CS-P-005, Safety Software Verification and Validation:

- Section 7.1, Verification, and Section 9.0, Records, Subsection 9.1
- Section 7.2, Validation, and Section 9.0, Records, Subsection 9.2

Software errors or anomalies may be reported using the Software Error/Corrective Action Report Form provided as CS-P-005, Appendix B.

Risk grading and risk management is described in SQAP, Section 14.

5.2.2 Security Analysis

The goal of security analysis is to understand the system owner's definition of an acceptable security risk and to evaluate the system for potential security risks. This analysis is usually done when the system is first defined, at the concept stage. It can be repeated throughout the lifecycle as appropriate.

Relevant documents, at a minimum the system requirements documents, are reviewed to gain an understanding of the owner's definition of acceptable security risk. With that understanding, the system concept is analyzed from a security perspective to assure that potential risks have been identified with respect to:

- Confidentiality – disclosure of sensitive information/data
- Integrity – modification of information/data
- Availability – withholding of information/services
- Accountability – attributing action to an individual/process

An assessment of the sensitivity of the information/data to be processed is included in the system requirements documentation. In addition, the risks introduced by the system itself and those associated with the environment with which the system interfaces are analyzed. The sensitivity of COG software is formally assessed prior to external release through Information Management. To date, COG software has been determined to be Export Control Information (ECI) and external distribution is provided through the Radiation Safety Information Computational Center (RSICC) at Oak Ridge National Laboratory, or under license agreement through the LLNL Industrial Partnerships Office (IPO).

The objective of security analysis in an operational environment is to assure that security threats and vulnerabilities, such as malicious code, are identified and mitigation strategies (prevent, control, or mitigate) are defined. Any unmitigated threats and/or vulnerabilities are documented and addressed as part of the operational procedures. The tasks for this activity in an operational environment are to:

- Verify that each identified threat has one or more mitigation strategies
- Verify that unmitigated threats are documented and addressed in the documentation for software operations

DOE is accountable for determining in DOE-STD-3007-2017 that the following code systems are accepted programs for use in nuclear criticality safety applications when used in accordance with a site-specific software quality assurance program for classifying and controlling software:

- COG
- MCNP
- SCALE

Having three independently developed codes qualified and available as 830 Software mitigates the loss of availability or integrity of any one code, which could occur due to loss of funding or discovery of a significant error requiring disaster recovery. Additional (foreign) codes with similar (safety) pedigree may be acquired including:

- MONK (Wood, United Kingdom)
- MORET (IRSN, France)
- TRIPOLI (CEA, France)

5.2.3 Traceability Analysis

Traceability analysis is done at the end of each lifecycle phase to assure that each component of the previous phase is mapped to at least one component of the just completed phase and vice versa. The relationships are analyzed for correctness, completeness, and consistency. The focus of the traceability analyses is on safety related requirements, as well as any critical requirements as identified by the software effort team. The types of tracing include but are not limited to:

- Requirements – software requirements are traced to the system requirements and the system to the software to assure that they are mapped to each other as appropriate
- Design – design elements are traced to requirements and requirements to design elements

- Construction – source code components are traced to corresponding design specifications and design specification(s) are traced to source code components
- Test – analyzes the relationships in the V&V software test plans, designs, cases, and procedures for correctness and completeness in testing requirements, design elements, and source code components

Traceability is maintained to the extent practicable with downward compatibility of software input specifications so that 830 (safety) Software application validation tests are easily repeated with minimal modification expediting inter-comparison of 830 Software baselines. These validation tests rely heavily on exact "analytic benchmarks" for software verification of algorithms and physical benchmarks from the International Criticality Safety Benchmark Evaluation Project (ICSBEP) enabling reference critical values derived from experiment to be compared with simulated values (e.g., k-eff, flux, reaction-rates, etc.).

5.2.4 Software Requirements Evaluation

The objective of this activity is to validate the adequacy of the requirements. For design analysis software, and other software as appropriate, the physical phenomena that are modeled are reviewed to assure it conforms to the system accuracy requirements and physical laws. The description of what is included in the software requirements is in Section 4.2.1 - Software requirements description (SRD) of the SQAP. Each requirement is uniquely identified and defined such that its achievement is capable of being objectively verified and validated.

Additional details are provided in the Software Requirements Document (SRD). The SRD is reviewed by the Change Control Board, which includes code developers and users, and is approved by the Responsible Manager.

5.2.5 Software Design V&V

Software design is the transformation of software requirements into software components. A high level or architectural design illustrates how the software interfaces to external hardware and/or users. The detailed design defines the internals of each component, including databases and other data elements, to a level from which it can be coded. Each component of the design is traceable to the original software requirements. The documentation for detailed design may be in the source code.

High level documentation of the software design is provided in the Software Architecture Design Description Document, which includes software interfaces to databases and other data elements. Baseline specific requirements are provided in the Software Requirements Document. The detailed design documentation is provided in the source code itself.

The objective of this activity is to demonstrate that the design is a correct, accurate, and complete transformation of the software requirements and that no unintended features are introduced.

5.2.5.1 Design Evaluation

The design evaluation task evaluates the adequacy and acceptability of design elements (e.g., design documents) for correctness, consistency, completeness, accuracy, readability, and testability. This task is completed before the code is approved for use. This evaluation considers both architectural and detailed design components. The constraints and tasks for this activity include but are not limited to:

- The responsible design organization is included in the evaluation activity
- Verify and validate the design satisfies the requirements
- Documentation is legible and unambiguous to intended audience
- Objective acceptance criteria exist for validating each design element
- Each design element is testable to the acceptance criteria

Documentation that the design is correct and provides accurate results is provided in documentation of the code results to analytical (verification) benchmarks, physical (validation) benchmarks, and code intercomparison studies.

NAP-24A additional consideration: Verify the adequacy of designs prior to final software acceptance, including:

- Applicable standards are used (i.e., DOE-STD-3007-2017)
- Mathematical models in simulation codes are adequately verified and validated (as specified in the Software Requirements Document)
- Qualification methods are adequate (in compliance with national consensus standards; e.g., ANSI/ANS-8.24)

5.2.5.2 Interface Analysis

The interface analysis task verifies and validates that all interfaces to the software product are correct, consistent, complete, accurate, and testable. This applies, but is not limited to interfaces with: hardware, users, operators, software components, processes, other systems, and facilities. This analysis may be performed concurrently with the design evaluation activity. The tasks for this activity include but are not limited to:

- Internal and external interface design meets requirements
- Each interface is identified in the COG Software Architecture Design Description Document
- Each interface provides information with the required accuracy
- Objective acceptance criteria exist to validate the interface design

Validation reports as required by CS-P-004, "Criticality Safety Evaluations," and CS-P-005, "Safety Software Verification and Validation," provide the required interface analysis by quantitative performance testing of the complete software package including interfaces (e.g., cross-section data libraries) on computational (hardware and software) platforms under formal configuration management. Analysis includes quantitative analysis of accuracy in terms of bias and bias uncertainty including confidence limits with a margin of safety as required by ANSI/ANS-8.14-2017 and DOE-STD-3007-2017 for development of an upper subcritical limit for a defined area of applicability.

5.2.6 Software Construction V&V

The objective of this activity is to verify and validate that the transformation of design into code is correct, accurate, and complete.

5.2.6.1 Source Code & Source Code Documentation Evaluation

This task involves review of the actual source code and its related documentation for correctness, consistency, completeness, accuracy, readability, and testability. The tasks for this activity include but are not limited to:

- Verify correct implementation of algorithms, equations, mathematical formulations or expressions
- Verify source code components are identified in the Software Architecture Design Description Document
- Verify that source code components comply with standards, regulations, policies, physical laws, and business rules
- To the extent practicable, validate there are no unnecessary, unintended, or deleterious consequences from design element interactions
- Assess appropriateness of coding methods and standards are being used
- All terms and code concepts are consistently documented
- Documentation satisfies specified coding standards
- Logic, computational, and interface precision is accurate in the system environment
- Modeled physical phenomena conform to accuracy requirements and physical laws
- Documentation is legible, understandable, and unambiguous to intended audience
- All acronyms, abbreviations, and terms are defined

This is done through feature testing. The results of feature tests are included the software repository and results are documented in a publication describing a specific 830 (software) baseline.

5.2.7 Software Test V&V

The objective of software qualification testing is to test the functional, environmental, and reliability performance of an integrated software product to assure that it satisfies its *requirements* prior to being released to the customer.

The objective of software acceptance testing is to assure that the software satisfies its *acceptance criteria* and *to enable the customer to determine whether or not to accept the integrated software product.*

In practice, qualification and acceptance testing may be combined into a set of tests that assure that both testing objectives are met. The following sections present the qualification and acceptance testing as combined. These testing tasks may be repeated during the software lifecycle with each instance having a specific purpose. These testing tasks are used to satisfy the qualification requirements of PRO-0110, Identification, Documentation, Control, and Maintenance of the 830 Software Inventory.

The test cases and related documentation may be created during the code development phase or immediately prior to beginning the testing phase. Prior to starting the test V&V activities, test plans are written to satisfy the requirements of the specific V&V activity.

The Software Requirements Document specifies requirements for software test V&V. The code developers provide instructions and a limited set of (regression) test V&V cases with supported platform-specific results so that users may verify proper code installation and performance. The instructions request providing any noted anomalies to the developers.

Prior to distribution to external users, RSICC performs an independent test that the instructions for installation and V&V testing can be actually performed successfully by a knowledgeable user.

As a result of the software test activity, all test plans, cases, and results are documented, reviewed, and approved prior to using the software. As appropriate, testing V&V tasks may be combined, such as performing the test design and test procedure V&V concurrently.

For internal users, the required documentation for installation and verification testing are specified in CS-P-005, "Safety Software Verification and Validation." This documentation is further attached and submitted to the LLNL SQA Manager using FRM-3128, "Installation Verification Summary." When approved for 830 (safety) usage, the software (baseline) version will be listed as "Status: ACTIVE" together with the platform "Host Name/Identifier" in the LLNL 830 Software Inventory.

5.2.7.1 Software Test Plan V&V

The software test plan describes the scope, approach (task list), resources, and schedule for testing a software product to assure that it satisfies its requirements and customer expectations when deployed to their environment. Various documents are used to prepare the test plan. These may include: system requirements documentation, software requirements documentation, interface requirements documentation, and user documentation. The information in the plan includes, as applicable, but is not limited to the following to assure that the software:

- Conforms to all system requirements
- Conforms to acceptance requirements in the operational environment
- Has adequate user documentation
- Uses appropriate test methods and standards
- Operational and maintenance requirements are feasible and testable
- System requirements are traceable to the software test designs, cases, procedures, and results
- Correctly implements the system and software requirements in an operational environment
- Has the capability to be operated and maintained in accordance with user needs
- Meets the test requirements and acceptance criteria provided and/or approved by the responsible design organization
- Incorporates methods identified in Section 4.5 of the SVVP

The Software Test Plan is an element of the Software Requirements Document.

5.2.7.2 Software Test Design V&V

The software test design document includes details of the task list and approach to testing the software as described in the software test plan, as well as the method for performing the testing. Tests cases are designed to assure the software product meets its intended performance requirements, as defined in the requirements, design, and/or user documents, in its intended environment. The test cases have specific inputs, outputs, and expected results documented based on the stated performance requirements. The design document and test cases are reviewed to assure that they include, at a minimum, but are not limited to:

- Set of inputs are a reasonable sample of all possible inputs; boundary condition, rarely used, and invalid values are included
- Performance under stress conditions and at data and interface boundaries is defined
- Has test coverage (extent of software exercised) for all system requirements
- If used in design activities, testing provides for assuring the software produces correct results
- If used for operational control, tests exist to demonstrate required performance over the range of operation of the controlled function or process
- If appropriate, ways of comparing test results with output of alternative methods such as hand calculations, use of comparable proven programs, empirical data, and information from technical literature are identified

An extensive set of ICSBEP criticality benchmark validation tests represent a reasonable sample of all possible inputs, boundary conditions, and rarely used features. Due to fiscal limitations, testing of invalid values is limited to user errors.

Regression tests are intended to provide significant test coverage. The other test, verification, and validation suites specified in Table II of https://cog.llnl.gov provide additional performance testing at data and interface boundaries and provides additional test coverage of 830 (safety) features.

Feature tests provide assurance that specific software features produce correct results.

COG is not used for operational control.

The ICSBEP Handbook provides test results from a wide variety of comparable national and international proven programs (e.g., MCNP (LANL), MONK (Wood, UK), SCALE (ORNL), TRIPOLI (CEA, France).

5.2.7.3 Software Test Procedure V&V

A software test procedure defines the steps needed to execute the test plan, including any set up needed within the operating environment. The procedure is verified and validated to ensure that each of the elements of the plan is addressed. The plan elements to be included are in Section 5.2.7.2, Software Test Plan V&V, of this document.

The Software Test Plan V&V is an element of the Software Requirements Document, which is reviewed and approved by the Change Control Board including the code developers and responsible manager.

5.2.7.4 Software Test Execution

The software tests will be executed per the software test procedure. Design authority approvals will be obtained for any temporary facility changes before tests are run. The test results will be documented, including anomalies found, and re-run if indicated once corrections to the code have been made.

Significant anomalies or failures will result in abandonment of the baseline with development of new prototypes until a new, corrected, baseline can be finalized and tested.

Once tests have been successfully passed and approved, software will be qualified per PRO-0110, Identification, Documentation, Control, and Maintenance of the 830 Software Inventory, prior to official use in an 830 capacity.

The baseline may also be provided to RSICC for external distribution in accordance with Information Management.

5.2.7.5 Software Test Execution V&V

This activity reviews the test documents and code to assure that requirements are met, and results satisfy the specified criteria. The test documents include information for documenting and dispositioning unexpected or unintended test results prior to approving the results. The test documents include any tests performed in support of reviews that complement the actual tests. The complementary tests are subject to the same criteria as the actual tests and do not substitute for performing the final comprehensive tests. Testing is performed before the code is approved for use. The tasks for this activity include, as applicable, but are not limited to:

- Review test related document(s)
- Analyze test results to validate the software satisfies system requirements
- Results are documented as specified in the V&V test plan
- Document discrepancies between actual and expected results of this review
- Unexpected or unintended results are documented and dispositioned before test results are approved
- Use the results of this review to validate that the software satisfies acceptance criteria
- Validate the test results trace to test criteria established by test traceability in the V&V software test planning documents
- Ensure the responsible design organization is involved in evaluating test results for design related tests
- Results are evaluated by a responsible authority to ensure requirements have been satisfied
- Ensure that knowledgeable personnel are involved in evaluating test results
- Review test results to assure there is sufficient accuracy to evaluate and accept the results
- Verify that tests done in support of reviews are documented and subjected to the same criteria as the actual acceptance tests
- Test results demonstrate, as appropriate, that the software:
    - Properly handles abnormal conditions and events as well as credible failures
    - Does not perform adverse unintended functions
    - Does not degrade the system either by itself or in combination with other functions or configuration items

This is described in CS-P-004, "Safety Software Verification and Validation."

5.2.8 Software Installation & Checkout V&V

The objective of this activity is to verify and validate the correctness of the software installation in the target environment. The tasks for this activity include but are not limited to:

- Verify by analysis and/or tests that the installed software corresponds to that subjected to V&V
- Verify that the software code and databases (if any) initialize, execute, and terminate as specified
- Verify that a comprehensive acceptance test was performed in the operating environment (The degree to which testing is comprehensive is related to the complexity and significance of the software application.)

Since COG is used to support nuclear facility operations, FRM-3128, Installation Verification Summary, will be completed and submitted to the MAS SQA office for inclusion on the LLNL 830 Software Inventory wiki as per PRO-0110, "Identification, Documentation, Control, and Maintenance of the 830 Software Inventory," whenever a new baseline version of the application is released into production for each installation of the software that will be used in an 830 capacity. The new version will not be used in an 830 capacity on that installation until the version has been fully qualified as per Section 5.2.7.4 or Section 5.2.10.1 and the form has been submitted and approved.

5.2.9 Software Operation V&V

This activity evaluates the ongoing suitability of the software for use.

The following periodic in-use manual or automatic self-check in-use tests are utilized for those computer programs in which computer program errors, data errors, computer hardware failures, or instrument drift can affect required performance:

This is an ongoing activity. COG is in continuous usage and CS-P-004 requires reporting of any anomalous results.

5.2.10 Software Maintenance V&V

This activity is performed when there has been a change to the software or related documentation. The changes may be for bug fixes, operating environment updates, or feature enhancements.

Changes for bug fixes and feature enhancements pertain to the development of either new prototype or baseline versions. In other words, baselines that need to be fixed are abandoned in preference to new, corrected, baselines.

5.2.10.1 Task Iteration, Updates, and Maintenance

Whenever a change is made to the software product, its operating environment (including operating parameters), or any related documentation, it is critical that the extent of V&V analyses and tests to be repeated is determined and completed prior to releasing the products to the user community. The V&V activities that are undertaken help assure that the following are performed:

- Planned changes are implemented correctly
- Documentation is complete and current
- Changes do not cause unacceptable or unintended system behaviors
- Assess nature of change to determine potential ripple or side effects and impacts on other aspects of the system
- Re-run relevant test cases based on changes, error corrections, and impact assessment to detect errors or unintended adverse effects resulting from the changes
- The new version has been fully qualified, approved by the MAS SQA office, and added to the LLNL 830 Software Inventory wiki per PRO-0110, Identification, Documentation, Control, and Maintenance of the 830 Software Inventory, prior to official use in an 830 capacity.

This section is redundant as all processes, activities, and reviews required for initial development and deployment apply to subsequent baselines. However, research prototypes are subject only to testing of features under development.

6 SVV Plan Maintenance

This SVVP may evolve over time and/or may be incrementally completed. Some work products may be separate documents or eventually incorporated into this document. Changes to this plan are managed using the change control procedures specified in the COG Software Configuration Management Plan.

The revision history is maintained and/or referenced at the beginning of this document.