|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Laboratory-wide | | Template | |  | eCR Number: |  | | 619940 | |
|  |  | | | | | | | | |
| Document ID: PLN-5552  Revision ID: DRAFT  Effective Date: TBD | | | | | |
| Software Quality Assurance Plan (SQAP) | | | | | |
|  | | | |  | |
|  | | | |  | |
|  | | | | | |
|  | | | |  | |
| RAVEN and RAVEN Plug-ins Software Quality Assurance Plan | | | | | | | |

|  |  |  |
| --- | --- | --- |
| RAVEN and RAVEN Plug-ins Software Quality Assurance Plan | | |
| PLN-5552 | | |
|  | | |
| **Prepared by:** |  | |
| See eCR # |  |  |
| IT Project/M&O Manager |  | Date |
| **Reviewed by:** |  | |
| See eCR # |  |  |
| Independent Reviewer |  | Date |
| **Approved by:** |  | |
| See eCR # |  |  |
| IT Asset Owner |  | Date |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Applicability: | Plan |  | eCR Number: |  |
| Manual: | | | | |

CONTENTS

[1. PURPOSE 3](#_Toc524959716)

[1.1 Software Items Covered 3](#_Toc524959717)

[1.2 Software Lifecycle 3](#_Toc524959718)

[1.3 Assumption and Constraints 3](#_Toc524959719)

[1.4 Deviation Policy 4](#_Toc524959720)

[2. REFERENCES 5](#_Toc524959721)

[3. DEFINITIONS AND ACRONYMS 7](#_Toc524959722)

[3.1 Definitions 7](#_Toc524959723)

[3.2 Acronyms 12](#_Toc524959724)

[4. MANAGEMENT 13](#_Toc524959725)

[4.1 Organization 13](#_Toc524959726)

[4.2 Tasks 13](#_Toc524959727)

[4.3 Roles and Responsibilities 14](#_Toc524959728)

[5. DOCUMENTATION 19](#_Toc524959729)

[5.1 Minimum Documentation Requirements 19](#_Toc524959730)

[5.2 Other Documentation 20](#_Toc524959741)

[6. STANDARDS, PRACTICES, CONVENTIONS, AND METRICS 21](#_Toc524959742)

[6.1 Purpose 21](#_Toc524959743)

[6.2 Content 21](#_Toc524959744)

[6.2.1 Software Coding Standards 21](#_Toc524959745)

[6.2.2 Commentary Standards 21](#_Toc524959746)

[6.2.3 Testing Standards and Practices 22](#_Toc524959747)

[7. SOFTWARE REVIEWS 22](#_Toc524959756)

[7.1 Purpose 22](#_Toc524959757)

[7.2 Minimum Requirements 22](#_Toc524959758)

[7.2.1 Requirements Reviews 22](#_Toc524959759)

[7.2.2 Design Reviews 23](#_Toc524959760)

[7.2.3 Acceptance Review 23](#_Toc524959762)

[7.2.4 Change Request Approval Check List 23](#_Toc524959763)

[8. TEST 23](#_Toc524959767)

[9. PROBLEM REPORTING AND CORRECTIVE ACTION 24](#_Toc524959768)

[10. TOOLS, TECHNIQUES, AND METHODOLOGIES 24](#_Toc524959795)

[11. SUPPLIER CONTROL 26](#_Toc524959796)

[12. RECORDS COLLECTION, MAINTENANCE, AND RETENTION 27](#_Toc524959797)

[13. TRAINING 28](#_Toc524959798)

[14. RISK MANAGEMENT 28](#_Toc524959799)

[14.1 Safety Software Determination 28](#_Toc524959800)

[14.2 Quality Level Determination 29](#_Toc524959801)

[15. PLAN MAINTENANCE 29](#_Toc524959803)

# PURPOSE

The Risk Analysis and Virtual ENviroment (RAVEN) Software (see def.) was designed and has been developed to provide all the capabilities needed to perform Uncertainty Quantification, Probabilistic Risk Assessment, Data Analysis, Validation and Parameter Optimization.

This Software Quality Assurance Plan (SQAP) establishes the software Quality Assurance program for RAVEN and any RAVEN plug-ins (see def.). It covers the periods of software development, maintenance and operations (M&O), and retirement. It implements all the requirements as specified by LWP-13620, “Managing Information Technology Assets,” for a QL-3 software application.

* RAVEN is operational within multiple projects. Ongoing support of RAVEN is required for the purpose of adding functionality, correcting computational errors and improving the performance of the RAVEN software.
* RAVEN is maintained by a team of nuclear and software engineers, referred to herein as the RAVEN core team (see def.). RAVEN maintenance and operations, performed by the RAVEN core team, is an ongoing activity.
* This SQAP also addresses the organization, responsibilities, procedures, methods, and tools employed in maintaining, sustaining, and enhancing RAVEN. This plan describes the QA functions and the specific activities applicable to RAVEN. It covers activities performed by QA personnel and activities monitored by QA that are performed by other personnel.

## Software Items Covered

This plan covers the maintenance of all existing and future components of RAVEN. This includes, but is not limited to, servers, server software, user workstations, RAVEN software, and control documents. Changes to this document will be completed through the Electronic Change Request (eCR) process.

## Software Lifecycle

This SQAP covers all work performed on RAVEN, beginning with maintenance and lasting through retirement. All changes to the system will take place according to the guidance given in PLN-5553, “RAVEN Configuration Management Plan.”

## Assumption and Constraints

* All software management practices must be in compliance with LWP- 13620, “Managing Information Technology Assets”, including all software management activities performed for Quality Level (QL)-1, QL- 2, and QL-3 application software and associated support software (see def.). This plan applies directly to custom-developed software (see def.) and acquired applications and support software managed by the modeling and simulation team.
* The RAVEN core team will adhere to LWP-1303, “Management of Unclassified Cyber Security Information Systems” and LWP-1401, “Preparing and Releasing Scientific and Technical Information Products,” where applicable.
* 29 USC 794d, Section 508 of the Workforce Investment Act of 1998 considerations will be made for the ability of disabled individuals to access the information or service provided by the software.
* Roles and responsibilities cited in this plan can be reassigned as needed by the project manager or personnel designated by the information technology (IT) asset owner.
* INL will manage the software with support from vendors (for *acquired software* [see def.]) until the software is retired.
* Software vendor support agreements are maintained.
* For firmware, changes to acquired software including software updates and security patches will be implemented by the product vendor.
* All changes to safety software (see def.) including security patches will be controlled through the change control board (see def., CCB). For external release, the project or maintenance and operations (M&O) team will adhere to LWP-1401, “Preparing and Releasing Scientific and Technical Information Products.”
* The scope of this document only covers the RAVEN software and projects that utilize the RAVEN framework.
* The hardware that serves RAVEN is managed by the High-Performance Computing Group. The hardware is considered a configuration item (see def.) for the RAVEN IT asset, and changes impacting the RAVEN framework must be reviewed by the RAVEN technical lead or designee; however, the management of the hardware is outside the scope of this plan.
* Software that is not within the scope of this plan include static webpages and other tools used for administrative purposes.
* All software packages that utilize the RAVEN software are covered by this plan in case they are managed at the same Quality Level assigned to RAVEN or lower

## Deviation Policy

All deviations from this plan require management approval. Whether planned or unplanned, if any deviation from this plan is necessary, the following components will be determined:

* Identification of task affected.
* Reasons for deviation defined.
* Effects on the quality of the project.
* Time and resource constraints affected.

A deviation report will be generated, and authorization will be required. Deviations that violate requirements must be documented within the relevant issue.

# REFERENCES

The following source documents apply to this SQAP:

* DOE Order 414.1D, “Quality Assurance”
* Form 562.29, “Software Product Review Report and Checklist”
* Form 562.33, “INL SQA Assessment Checklist”
* GDE-XXXX, “RAVEN User Documentation”
* ISO/IEC/IEEE 24765:2010(E), “Systems and software engineering — Vocabulary,” 1st Edition, December 15, 2010.
* LWP‑1201, “Document Management”
* LWP‑1202, “Records Management”
* LWP-13620, “Managing Information Technology Assets”
* LWP-13840, “Issues Management”
* NQA-1 2008 with 1a-2009 Addenda, “ASME Quality Assurance Requirements for Nuclear Facility Applications and Addenda”
* OINL1631, “Introduction to IT Asset Management” Training
* PDD-12005, “INL Training Program”
* PDD-13610, “Software Quality Assurance Program”
* PLN-4653, “INL Records Management Plan”
* PLN-5553, “RAVEN Configuration Management Plan (CMP)”
* SPC-2366, “RAVEN Software Requirements Specification (SRS)”
* SDD-513, “RAVEN Software Design Description (SDD)”
* PLN-5554, “RAVEN Software Test Plan (STP)”
* PLN-XXXX, “RAVEN IT Asset Maintenance Plan”
* PLN-5555, “RAVEN Verification & Validation”
* TEM-142, “Software Quality Assurance Plan Template”
* PLN-4005, “SQAP for MOOSE and MOOSE-Based Applications”

# DEFINITIONS AND ACRONYMS

This section defines, or provides the definition of, all terms and acronyms required to properly understand this plan.

## Definitions

*Acquired software.* Software generally supplied through basic procurements, two- party agreements, or other contractual arrangements. Acquired software includes commercial off-the-shelf software, support software such as operating systems, database management systems, compilers, software development tools, and commercial calculational software and spreadsheet tools (e.g. Microsoft’s Excel). Downloadable software that is available at no cost to the user (referred to as freeware) is also considered acquired software. Firmware is acquired software. Firmware is usually provided by a hardware supplier through the procurement process and cannot be modified after receipt.

*Anomaly.* Anything observed in the documentation or operation of software that deviates from expectations based on previously verified software products or reference documents.

*Change control.* An element of configuration management, consisting of the evaluation, coordination, approval or disapproval, and implementation of changes to configuration items (CIs see def.) after formal establishment of their configuration identification. [ISO/IEC/IEEE 24765:2010(E)]

*Change control board (CCB).* The group by which a change is proposed, evaluated, approved or rejected, scheduled, and tracked. This board is also responsible for evaluating and approving or disapproving proposed changes to configuration items (CIs) and implementation of approved changes when required.

*Change requests (CRs).* CRs can be initiated by anyone, including off site users, and can be used for maintenance (fine-tuning and problem resolving), new development, and enhancements, or can be used to report program errors and problems.

*Change request log.* A log that provides a listing of all the change requests and the change request status used for application software, system software, and hardware configuration control.

*Continuous Integration System (CIS).* A system, linked to a central version control repository, such as GITHUB and GITLAB (see def.), aimed to automatically build and test a targeted software. Examples are CIVET, Jenkis, and GitLab Continuous Integration.

*Configuration identification.* An element of configuration management, consisting of selecting the configuration items (see def.) for a system and recording their functional and physical characteristics in technical documentation.

*Configuration item (CI).* An item or aggregation of hardware or software (including documentation) or both that is designed to be managed as a single entity (ISO/IEC/IEEE 24765:2010(E) edited).

*Configuration management.* A discipline applying technical and administrative direction and surveillance to identify and document the functional and physical characteristics of a configuration item (see def.), control changes to those characteristics, record and report change processing and implementation status, and verify compliance with specified requirements (ISO/IEC/IEEE 24765:2010[E]).

*Custom-built IT assets.* Information technology (IT) assets designed, developed, or modified internally or by a qualified subcontractor through the procurement process. Examples include custom-developed (see def.) or customized software, spreadsheet, and calculation and analysis applications (e.g., computer models), the implementation of a new network infrastructure or IT technology (e.g., Gmail, Internet Protocol Version 6, Internet Explorer 9). [Developed for internal laboratory use]

*Defect.* An error, fault or failure in a computer program or system that causes it to produce an incorrect or unexpected result, or to behave in unintended ways.

*Doxygen*. Standard tool for generating documentation from annotated C, C++, Fortran and Python sources.

*Electronic Document Management System (EDMS).* System approved for long- term storage, management, and maintenance of electronic and hardcopy records.

*Enterprise Architecture (EA) Repository.* An Oracle database that houses information about software applications and servers and is the source for the INL data dictionary. The applications are related to the management system business functions it supports or implements. EA is the repository for the technology  
(e.g., software/hardware) used to construct and implement software applications. EA contains links to the software documentation stored in EDMS (see def.) and includes a list of software owners.

*GitHub.* A web-based revision control hosting service for software development and code sharing. GitHub provides additional tools such as documentation generation, issue tracking, Wikis, nested task-lists within files, etc.

*GitLab.* A web-based revision control hosting service for software development and code sharing similar to GitHub. GitLab is used for the applications/extensions/plugins built/developed on the RAVEN software. The CIS (see def.) connects to both the external and internal GitHub/GitLab to perform software builds.

*Issue.* Issues can be initiated by anyone, including off site users, and are used for maintenance (fine-tuning and problem resolving), new development, enhancements, or can be used to report program errors and problems.

*Issue (GitHub).* As defined for the GitHub environment, issues are suggested improvements, tasks, or questions related to the repository. Issues can be created by anyone (for public repositories) and are moderated by repository collaborators. Each issue contains its own discussion forum and can be labeled and assigned to a user/developer.

*Method.* A reasonably complete set of rules and criteria that establish a precise and repeatable way of performing a task and arriving at a desired result. [The Configuration Management Manual Guideline for Improving the Software Process, Carnegie Mellon University Software Engineering Institute, 1995]

*RAVEN core team.* INL personnel whose job description includes the development of the RAVEN software or software applications/extensions/plugins that are based on the RAVEN framework.

*RAVEN Software.* Open source software that resides in a public repository (GitHub) that provides all the capabilities needed to perform Uncertainty Quantification, Probabilistic Risk Assessment, Data Analysis, Validation and Parameter Optimization.

*Open source.* Denoting software for which the original source code is made freely available and may be redistributed and modified.

*Pull requests.* Pull requests can be initiated by anyone, including off-site users, and are used for maintenance (fine-tuning and problem resolving), new development, enhancements, or can be used to report program errors and problems. Pull requests let you tell others about changes you have pushed to a repository on GitHub. Once a pull request is sent, interested parties can review the set of changes, discuss potential modifications, and even push follow-up commits if necessary.

*Regression testing.* Selective retesting of a system or component to verify that modifications have not caused unintended effects and that the system or component still complies with its specified requirements.

*Software.* Computer programs and associated documentation and data pertaining to the operation of a computer system and includes application software and support software.

*Software life cycle.* The activities that comprise evolution of software from conception to retirement. The software life cycle typically includes the activities associated with requirements, design, implementation, test, installation, operation, maintenance, and retirement.

*Software quality assurance.* All actions that provide adequate confidence that software quality is achieved.

*Software tool*. A computer program used in development, testing, analysis, or maintenance of a program or its documentation. Examples include comparators, cross-reference generators, compilers, computer-aided software-engineering tools, configuration and code management software, flowcharters, monitor test case generators, and timing analyzers.

*Support software.* Software tools (see def.) and system software (see def.).

*System software.* Software designed to facilitate operation and maintenance of a computer system and its associated programs (e.g., operating systems and utilities).

*System testing.* Testing conducted on a complete, integrated system to evaluate the system’s compliance with its specified requirements.

*Task (GitHub).* A suggested improvement or feature enhancement.

*Test case. (1)* A set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement. (2) Documentation specifying inputs, predicted results, and a set of execution conditions for a test item.

*Test driven development.* A method of software development in which unit testing is repeatedly conducted on source code. After each test, refactoring is done and the same or a similar test is performed again. The process is iterated until the unit functions in accordance with the specifications.

*User documentation.* Instructions for use describing the capabilities and intended use of the software within specified limits. May also include a theory manual, when relevant.

*Validation.* Confirmation, through the provision of objective evidence (e.g., acceptance test), that the requirements for a specific intended use or application have been fulfilled. [ISO/IEC/IEEE 24765:2010(E) edited]. As described in SDD-513, “RAVEN Software Design Description”, RAVEN does not own Physical models and the Validation is performed verifying the algorithms/methods with analytical solutions (if applicable).

*Verification.* (1) The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. (2) Formal proof of program correctness (e.g., requirements, design, implementation reviews, system tests).  
[ISO/IEC/IEEE 24765:2010(E) edited]

*Custom‑developed software.* Software built specifically for an INL application. It may be developed by INL or contracted with a qualified software company through the procurement process. Examples of custom-developed software include material inventory and tracking database applications, accident consequence applications, control system applications, and embedded custom developed software that controls a hardware device. [DOE G 414.1‑4 edited]

*Configuration control.* An element of configuration management, consisting of the evaluation, coordination, approval or disapproval, and implementation of changes to configuration items after formal establishment of their configuration identification. [ISO/IEC/IEEE 24765:2010(E)]

*Configuration management.* A discipline applying technical and administrative direction and surveillance to: identify and document the functional and physical characteristics of a *configuration item* (see def.), control changes to those characteristics, record and report change processing and implementation status, and verify compliance with specified requirements. [ISO/IEC/IEEE 24765:2010(E)]

*Quality grade.* The grade applied to the level of quality activities to be applied to the specific task or activity. Current quality grades are Nuclear Use QL and Commercial Use Quality Levels (QLs) High, Medium, and Low.

*Retirement.* Permanent removal of an IT asset (e.g., system or component) and associated support from its operational environment. [ISO/IEC/IEEE Std 24765‑2010 edited]

*Agile development.* Agile development is an approach to software development under which requirements and solutions evolve through the collaborative effort of self-organizing and cross-functional teams and their customer(s)/end user(s). It prescribes adaptive planning, continuous development, early delivery, and continual improvement, and it encourages rapid and flexible response to change.

## Acronyms

ASME American Society of Mechanical Engineers

CCB Change Control Board

CI Configuration Item

CIS Continuous Integration System

CR Change Request

CM Configuration Management

CMP Configuration Management Plan

DOE Department of Energy

EA Enterprise Architecture

EDMS Electronic Document Management System

IAS Integrated Assessment System

INL Idaho National Laboratory

ISMS Integrated Safety Management System

ISO International Organization for Standardization

IT Information Technology

M&O Maintenance and Operations

NQA Nuclear Quality Assurance

QA Quality Assurance

QLD Quality Level Determination

QLD Quality Level Determination

RTM Requirement Traceability Matrix

RAVEN Risk Analysis and Virtual ENviroment

SRS Software Requirements Specification

SSD Safety Software Determination

SQA Software Quality Assurance

SQAP Software Quality Assurance Plan

SSD Safety Software Determination

V&V Verification and Validation

# MANAGEMENT

## Organization

The RAVEN core team is responsible for the project activities of the RAVEN software (see def.).

During the initial configuration of RAVEN, the RAVEN core team/management team, with each of their respective responsibilities, are defined in PLN-XXXX, Project Management Plan.

The RAVEN core team/management team will be defined in PLN-XXXX, “RAVEN IT Asset Maintenance Plan”. For changes to RAVEN, a configuration team, as outlined in PLN-5553, RAVEN Configuration Management Plan’, will be used to perform all configuration activities.

## Tasks

Table , Software quality assurance tasks, identifies the software quality assurance tasks to be performed. The tasks are listed in the order they are performed during procurement and configuration. The “schedule” column identifies when the tasks are performed, and the entrance and exit criteria for each stage are also established.

Table 1. Software quality assurance tasks.

| **Task** | **Schedule** | **Entry Criteria** | **Exit Criteria** |
| --- | --- | --- | --- |
| Risk Analysis | Per major release | Business requirements | Approved safety software determination (SSD) and quality level determination (QLD) |
| Management Plan review and approval | As needed | Draft management plan | Approved by IT Asset Owner. |
| Requirements review and approval | Per major release | Draft design description | Approved by IT Asset Owner. |
| Design review and approval | Per major release | Draft design description. | Approved by IT Asset Owner. |
| Implementation review | Per release | Baselined software prior to system test. | Documented code walk‑through to ensure consistency of software and supporting documentation including traceability of requirements through the life cycle. |
| System Test | As required by SCMP. | Completed implementation review. | Approved system test by test case personnel. |
| Acceptance Test | Per release | Completed system test. | IT asset approved by IT Asset Owner or Technical Leader. |
| Problem Resolution | As needed | Problem report submitted. | Closed problem report. |
| In-process QA inspection | Biennially | Initiated by Assurance Portfolio and Integrated Assessment System (IAS) scheduled start date. | Approved assessment report. |

## Roles and Responsibilities

[Identify the roles that will perform the SQA tasks identified in Section 4.2, such as the independent verifier, and describe their responsibilities. For V&V activities, personnel independent from the developers is required. Include status reporting to management and problem resolution responsibilities so that management may take action to mitigate risk. If this information is adequately documented in the IT project management plan or IT asset maintenance plan, that plan may be referenced.]

Table 1 identifies the roles and responsibilities for the SQA and RAVEN project management activities.

The general roles and responsibilities for RAVEN core team are outlined in PLN-XXXX. Table 1 identifies any additional roles and responsibilities for the SQA tasks.

**NOTE:** The roles discussed below are in terms of RAVEN personnel. LWP-13620 identifies the same roles, but with different names. The table below shows the corresponding role in LWP-13620.

Table 1. Roles and responsibilities.

| **Role** | **Responsibilities** | **Corresponding Role Per LWP-13620** |
| --- | --- | --- |
| Management | * Provide funding and stuffing for RAVEN and RAVEN plugins/extension software activities * Assign personnel and ensure they are properly qualified and trained to perform SQA tasks defined in Section 4.2. Refer to Training Section of this plan for further detail. * Ensure fund availability for at-need QA inspection by external assessing team. * Ensure corrective actions are implemented as needed * Ensure assessment are performed as specified in this plan | IT Asset Owner/  Management |
| IT Asset Owner | * Identify and document the safety software determination SSD as part of a criticality/risk analysis. * Acquire and dedicate IT materials and services in accordance with INL acquisition policy and this plan. * Responsible for administration and execution of this plan. * Ensure the completion of the QLD as part of a criticality/risk analysis. * Participate as necessary on the change control board (see def., CCB) as needed and act as final authority when necessary. * Act as the final authority for the approval/disapproval of change requests (see def.). * Review acceptance test and approve IT asset for deployment. | IT Project Manager/ M&O Manager |
| Project Manager | * Interface for all the internal and external RAVEN contacts/customers * Create/update IT asset portfolio including total expected life-cycle cost information. Review and approve management plan documentation. * Create/revise management plan documentation. Acquire and dedicate IT materials and services in accordance with INL acquisition policy. * Coordinate execution of implementation review. Ensure at-need QA inspection by external assessing team is executed and issues are properly tracked as part of the INL Issues and Corrective Action Management System. * Manage and resolve problems per this plan. Create/revise management plan documentation. * Provide status reports to management per communication plan. Coordinate independent review of management plan, requirements, and design documentation. * Act as the chair of the CCB. * Approve/disapprove and status all change requests. |  |
| Technical Lead | * Document test procedures and instructions for use. * Coordinate execution of implementation review. * Maintain requirements/design baseline (see def.). * Final approval on design reviews. * Oversight of design implementation and integration testing activities when applicable. * Assign system administrator duties as needed. * Resource Allocation dispatching in conjunction with the Project Manager * Conduct requirements, design, and implementation reviews. * Approve testing results for release of a new software version. * Identify and manage configuration items (see def.). * Ensure implementation and verification (see def.) of change and document as required by this plan. * Participate on the CCB. * Place IT assets under version control. * Establish baseline (see def.) of the IT asset prior to acceptance test. * Evaluate issues (see def.) and anomalies. * Initiate component and integration tests prior to system test. * Ensure implementation and verification activities are complete and document as required. * Coordinate execution of configuration audits. |  |
| QL Analyst | * Identify and document the QLD. * Manage and resolve problems per this plan. |  |
| Independent Reviewer | * Review management plan and participate in requirements, design, and implementation reviews. * Provide status reports to management per communication plan. |  |
| Change Control Board (CCB) | * Review and approve change requests. * Evaluate test results as part of the approved changes. |  |
| Software Developer | * Perform design, implementation and testing of the software code. * Adhere to this plan. |  |



# DOCUMENTATION

The purpose of this section is to define the minimum documentation required to properly implement the SQA requirements.

At all times during the life cycle of RAVEN, the following documents will be maintained as part of the IT Asset Portfolio.

## Minimum Documentation Requirements

As a minimum, the following documentation is required for the RAVEN software and supported RAVEN plug-ins. These documents are managed as records in accordance with Section 12, “RECORDS COLLECTION, MAINTENANCE, AND RETENTION”.

The following documentation is required as a minimum:

|  |  |  |
| --- | --- | --- |
| **Document** | **Record Location** | **ID** |
| IT Asset Maintenance Plan | Electronic Document Management System (EDMS) | PLN-XXXX |
| Configuration Management Plan | Electronic Document Management System (EDMS) | PLN-5553 |
| Standards Document | Electronic Document Management System (EDMS) | PLN-XXXX |
| Verification & Validation | Electronic Document Management System (EDMS) | PLN-5555 |
| Software Test Plan | GITHUB/GITLAB | PLN-5554 |
| Software Requirements Specification | GITHUB/GITLAB | SPC-2366 |
| Software Design Description | GITHUB/GITLAB | SDD-513 |
| Requirements Traceability Matrix | GITHUB/GITLAB | SPC-XXXX |
| User Documentation (see def.) | GITHUB/GITLAB | INL/EXT-15-34123, INL/EXT-18-44465, INL/EXT-16-38178 |
|  |  |  |

## Other Documentation

In addition to the above documents, the following are created during the procurement and baselining of the project. These may be used in support of Change Control Request implementation and M&O activities.

* SSD-000649, “Risk Analysis and Virtual Environment”
* QLD, “RAVEN Quality Level Determination”
* RAVEN Enterprise Architecture Entry #331229

A description of each of these documents is included in PLN-XXXX. It is anticipated that the QLD and SSD will not require review for every change to the system. However, for major changes, the sQLD and SSD may be revisited.

The RAVEN Software Test Plan, PLN-5554, will be used for implementation reviews and acceptance testing. The use of these documents is described in PLN-XXXX.

All documents will be managed according to LWP-1201, “Document Management.”

All records generated as part of this plan will be processed and managed according to LWP-1202, “Records Management.”

# STANDARDS, PRACTICES, CONVENTIONS, AND METRICS

## Purpose

Implementing the software quality assurance activities described in this plan will ensure that the RAVEN and RAVEN plugin-is software will conform to requirements of DOE Order 414.1D and NQA-1 2008, with 2009 Addenda, as implemented in [LWP-13620](https://inl-edms/pls/inl_docs/doc_3?f_doc=LWP-13620), “Managing Information Technology Assets.”

## Content

The standards for RAVEN and RAVEN supported plugins are maintained/recorded in the RAVEN GitHub repository (Wiki section). Any developer of the RAVEN software and RAVEN supported plug-ins need to be aware of the standards and to follow the development guidelines.

The RAVEN standards evolve around the following macro-areas:

* Software Coding Standards
* Commentary Standards
* Testing Standards and Practices

### Software Coding Standards

The RAVEN software imposes a coding standard on all source code within the repository. This standard is publicly maintained on the RAVEN GitHub repository wiki website (https://github.com/idaholab/raven/wiki/RAVEN-Software-Coding-Standard) and enforced through the continuous integration testing system.

### Commentary Standards

The RAVEN software imposes a commentary standard on all source code within the repository. The standard is aimed to fully describe any module/method in the source code, guaranteeing the automatic generation of software documentation via doxygen (see def.). This standard is publicly maintained on the RAVEN GitHub repository wiki website (https://github.com/idaholab/raven/wiki/RAVEN-Software-Commentary-Standard) and enforced through the continuous integration testing system.

### Testing Standards and Practices

The RAVEN software imposes a testing standard and practices on all the capabilities/methods of the RAVEN software. This standard is publicly maintained on the RAVEN GitHub repository wiki website (https://github.com/idaholab/raven/wiki/RAVEN-Testing-Standards-and-Practices) and enforced through the review process by a member of the CCB.

# SOFTWARE REVIEWS

## Purpose

Since RAVEN is a custom-developed software, it is required to have reviews to ensure compliance with NQA-1 standards.

Software reviews serve to assure appropriate progression to the next phase in the software lifecycle. Software reviews provide an independent perspective from those directly involved in the software development process. The independent reviewer will participate in the process of these reviews on a risk-based graded approach based on the RAVEN subject matter expert (SME) assessment to assure the requirements of LWP-13620 are being properly met. Any CCB member is considered independent reviewer for the documentation associated with the CR.

As the review process can be part of the overall configuration control, PLN-5553, explains the process of identifying, requesting, approving, implementing, and documenting changes.

## Minimum Requirements

At a minimum, the following reviews will be conducted. All comments/change histories are retained in the *GitHub* system. Any design aspects that are not resolved at the time of the review are addressed in follow-up *issues* (GitHub) (see def.).

At a minimum, the following reviews will be conducted (if applicable):

### Requirements Reviews

In case of CR that modifies and/or add new requirements for the RAVEN software, the CCB chair or designee shall perform a requirement review during the initial stages of planning, before procurement and configuration of the IT asset. The results will be documented in PLN-XXXX, “RAVEN Requirements Traceability Matrix.”. An additional record of the review will be retained in GitHub (see def.).

### Design Reviews

One or more independent design reviews are required for all design changes to evaluate the technical adequacy of the design approach and ensure internal completeness, consistency, clarity, and correctness of the software design. In addition, it is required to demonstrate that software design is traceable to the software requirements. These reviews will include review of test results and be recorded, with identification of the reviewer, within the CR. This is implemented as a series of comments and date fields used by the RAVEN core team (being part of the CCB) to record the required reviews and approvals prior to acceptance for use. Any changes to source code, comments, or documentation within the code repository trigger automated testing.

### Acceptance Review

Review is performed by the Software Technical Leader or Project Manager to ensure compliance with the approved software requirements. Automated *regression testing* (see def.) system results are considered to be part of the acceptance review.

### Change Request Approval Check List

In order to guide the review process of any CR in the RAVEN software, a PR (see def.) check list needs to be satisfied. If any of the required checks are not satisfied the CR developer and the reviewer(s) need to document the reason why a certain check is not applicable. The PR check list is publicly maintained on the RAVEN GitHub repository wiki website (https://github.com/idaholab/raven/wiki/development-checklists#peer-review-checklist-for-merge-requests).

# TEST

The software *verification* (see def) and *validation* (see def) will be outlined in PLN-5555, “RAVEN Verification & Validation.” This plan will set forth the testing, reporting, and documentation standards for software testing.

The original software baseline will be tested according to PLN-5554, “RAVEN Software Test Plan.” This plan documents the test scripts and reporting procedures to validate the software baseline.

# PROBLEM REPORTING AND CORRECTIVE ACTION

Problems and corrective actions within the scope of the RAVEN software and RAVEN plug-ins are tracked using the CR process to report and resolve issues. For issues discovered during the verification process, resolution will be tracked on the original CR related to the software modification.

Any system user, internal or external, can report an issue in GitHub (for RAVEN software) or GitLab (for RAVEN internal plug-ins). The issue is evaluated and categorized in the GitHub system as either a “Defect” - bug or “Improvement” - a non-critical task such as a feature enhancement; or the issue is closed without category if the issue was created accidentally or due to a not perfect knowledge of the software by the initiator of the Issue. Any internal or external user can choose to work on a particular issue simply by associating the issue with a pull request in GitHub. A listing of all active issues is available through GitHub for all users to review. A similar system exists internally for the RAVEN plugin-ins through GitLab.

If a problem relates to RAVEN and violates a requirement, the issue will be documented and managed in the INL Issues Management System, per LWP-13840, “Issues Management.”.

Any Issue categorized as “Defect” will require a notification to the users (email or group notification through the Github/Gitlab software) upon resolution.

The process related to address problems/defects and consequential corrective actions are fully documented in PLN-5553, “RAVEN and RAVEN plug-ins Configuration Management Plan”.



# TOOLS, TECHNIQUES, AND METHODOLOGIES

The RAVEN software and its supported plug-ins are in continuous evolution, via an Agile development process (see def.), since new expansions and capabilities are needed by the different projects/programs. In order to guarantee the SQA standards identified by this plan, an articulated set of tools, techniques and methodologies are required:

Methods:

* + Test and user-need driven development
  + Pull requests
  + Continuous integration.

Techniques:

* + Code coverage analysis
  + Regression testing
  + Expected error testing
  + Unit testing (when applicable)
  + Cascading builds
  + Agile development
  + Peer reviews
  + Performance testing
  + Shared repository (GitHub)
  + Decentralization.

Tools:

* + GitHub (Git) software code repository used outside the INL software network.
  + Python software development language
  + C++/C language
  + Bash scripting
  + Wiki – RAVEN documentation
  + Doxygen – Software framework documentation generator
  + Enterprise Architecture (EA)
  + EDMS Safety Software Determination system
  + Quality Level Determination (QLD) system
  + Form 562.29, “Software Product Review Report and Checklist”
  + Form 562.33, “INL SQA Assessment Checklist”
  + Form 562.38, “Safety Software Training Documentation Form.”

# SUPPLIER CONTROL

The IT asset owner, with support from the INL Procurement organization, will acquire materials and services that are necessary to support the RAVEN software and its supported plug-ins. These acquisitions include otherwise acquired software (i.e., software that has not been previously approved under a program consistent with the INL Quality Assurance program including freeware, shareware, and firmware).

Subcontractor/vendor control activities incorporate items developed outside the control of this plan into the CIs. Included are IT assets developed by contract and IT assets acquired in their finished form. Special attention is directed to these CM activities due to the added organizational and legal relationships.

Periodic changes and improvements will be made by the vendor to incorporate lessons learned from similar installations. These changes will be managed per Software Change Control, documented in PLN-5553, “RAVEN and RAVEN plug-ins Configuration Management Plan”.

Subcontractors might be used to implement CRs. When this occurs, the project manager will be responsible for the procurement of their services. The Project manager will be responsible for determining which tasks of a CR should be subcontracted out and funding the tasks/improvements. When subcontractors are used to implement changes to CIs, the project manager or technical leader will deliver the CI to the subcontractor. The subcontractor will perform the changes and return the modified CI to the project manager or technical leader. After the modified CI has been tested or reviewed and approved by the CCB, the technical leader will baseline the modified CI and place it under configuration management.

Software updates and revisions will be available from the vendor through a maintenance support agreement with the vendor.

The selection and management of subcontractors is performed by the associated project manager. When it is determined necessary to involve subcontractors, the project manager will be responsible for the procurement of their services and for determining what tasks they will perform.

Acquisitions must be handled in accordance with the following procedures:

* LWP-4001, “Acquisition of Materials and Services”
* LWP-1305, “Acquisition of Computer Hardware/Software Resources.”

In addition, Quality Level-1 or Quality Level-2, acquisitions must be handled in accordance with the following procedures:

* LWP-4501, “Preparation and Control of Procurement Documents”
* LWP-4503, “Supplier Evaluation and Qualification.”

Procurement documents shall identify requirements for Supplier’s reporting of software errors to the RAVEN core team and as appropriate, the team’s reporting of software errors to the Supplier.

When acquiring application software (including upgrades), the following documentation is required:

* Business requirements describing the capabilities and limitations.
* Test plans and test cases that will be used to validate the capability of the system for its specific application.
* Instructions for use.

The resulting documentation and associated software will establish the current baseline.

# RECORDS COLLECTION, MAINTENANCE, AND RETENTION

The primary quality assurance (QA) records include:

* Management plans
* Business and technical requirements
* Design documentation
* Independent review documentation
* Verification and validation results
* Change control and configuration management documentation
* Assessment reports.

All RAVEN and RAVEN plug-ins QA records are managed, and retention periods set per PLN-4653, “INL Records Management Plan.” Retention periods for electronic records (e.g. GitHub) are identified and tracked in the IT Asset Portfolio per LWP‑1202, “Records Management.”

# TRAINING

Project manager is responsible for ensuring implementation of the required SQA and training.

The RAVEN core team personnel have been selected based on the expertise required for RAVEN software development.

Personnel assigned to any of the roles supporting the RAVEN software shall be assigned Training Records and Information Network (TRAIN) job code XXXXXX, IT Asset Management, on their employee training plan.

Training includes the following activities, all of which will be documented on the employee’s individual training plan within the TRAIN system:

* Complete laboratory IT Asset Management training course 0INL1631, “Introduction to IT Asset Management,” as required by LWP-13620, “Managing Information Assets.”
* Orientation for this plan will be given to every staff member by the RAVEN core team’s project manager or designee.
* Team members will also be trained in the use of Git, GitHub, GitLab, coding, and commentary standards through study of the RAVEN Developer Guide webpage (https://github.com/idaholab/raven/wiki/Developer\_Information).

Required training shall be implemented as described in PDD-13610, “Software Quality Assurance Program.”

# RISK MANAGEMENT

The RAVEN software and RAVEN supported plug-ins must be managed in accordance with the requirements outlined in LWP-13620. The risk analysis for each application is documented on the safety software determination (SSD) and quality level determination (QLD). The SSD and QLD are identified in the EA repository for each individual application. Risks associated with the RAVEN software and RAVEN supported plug-ins are controlled via the rigor implemented in requirements identification, testing, verification and validation, and change control processes.

## Safety Software Determination

The SSD documents the decision basis as to why a software application is or is not safety software. The record copy is maintained within the company approved electronic document management system in accordance with LWP-13014, “Determining Quality Levels.”.

The RAVEN software and RAVEN supported plug-ins will be required to have a documented SSD. The SSD for RAVEN supported plug-ins will be re-evaluated in case of moving to operation.

## Quality Level Determination

The QLD documents the risk analysis in accordance with LWP‐13014, “Determining Quality Levels”, based on the end use of the RAVEN software and supported plug-ins. The QLD for RAVEN supported plug-ins will be re-evaluated in case of moving to operation.

# PLAN MAINTENANCE