Design of Quality Assurance System for APro Code Based on PAPiRUS

In-Young Kim [a*], Jung-Woo Kim [a]
[a]*Korea Atomic Energy Research Institute, (34057) 111, Daedeok-daero 989beon-gil, Yuseong-gu, Daejeon*
[*]*Corresponding author: iykim@kaeri.re.kr*

Keywords : APro Code, PAPiRUS, software development, quality assurance, total system performance assessment

1. Introduction

Korea Atomic Energy Research Institute (KAERI) is developing the Adaptive Process-based Total System Performance Assessment Framework (APro) as a process-level TSPA tool. To ensure the reliability of the APro code for its future application in disposal facility design, safety assessments, and licensing, a rigorous quality assurance (QA) framework must be established. This study aims to enhance the credibility of QA implementation and record management by developing a comprehensive QA system and integrating it into the PAPiRUS, Performance Assessment database management Platform integrating Research results of radioactive-waste disposal in Underground System, a comprehensive data management platform. To achieve this, the essential documentation requirements were analyzed and organized. Additionally, a review of existing system was conducted to identify key document management elements necessary for effective QA record management. Based on these findings, initial design of QA system tailored for integration into the PAPiRUS was configured.

2. Methods and Results

To design QA system integrated into the PAPiRUS, the documentation targets were first identified, and the types of documents requiring management were determined. Additionally, an analysis of the existing QA record management system was conducted to specify the necessary information for QA compliance documentation. Based on these findings, a page layout for PAPiRUS integration was designed.

2.1 Identification of Documentation Requirements

The APro development team reviewed various QA standards, including ISO, ASME, IEEE, ANSI/ANS, KEPIC, nuclear facility QA guidelines, and KAERI's internal QA framework, to establish software QA procedures. Given APro's status as a newly developed code, highly sophisticated frameworks like ISO and IEEE were deemed unsuitable, leading to the adoption of KAERI's internal QA framework based on ASME NQA-1 and KEPIC QAP. In KAERI's nuclear R&D, QA is managed through manuals and procedures such as RD-QAM, QAP-RD, QAP-NP, and SWQAM. To meet QA requirements, the necessary software development deliverables and detailed documentation items are summarized in Tables 1 and 2 [1-3]. Based on NP procedures and SWQAM, the APro SVVR and Individual Test Report Guidelines were developed as a comprehensive QA guide for APro [4].

2.2 Identification of Document Management Requirements

The required deliverables and information items for each stage of software development were identified and integrated into the SVVR and Individual Test Report Guidelines for APro code. To design a QA system for managing QA documents, the document management framework implemented in ANSIM [5], developed by the Korea Atomic Energy Research Institute (KAERI), was reviewed. The classification hierarchy for managed documents is summarized in Table 3. The document type, software name, version, document number, revision details, creation and modification dates, title, author, approval hierarchy, main file, and status need to be provided to ensure comprehensive document management.

2.3 Initial Design Layout of QA System in PAPiRUS

By modifying the data management page of the PAPiRUS program, the QA system can be integrated. The page includes a filter and metadata search (top left), a data list (bottom left), and a main page (right) that presents detailed information and utilization plans for each data group in a table format while providing access to corresponding files. The table functions like a folder containing explanations for data use, and users can open files in a new tab for processing and saving through a linked program. To incorporate QA, the left panel will list QA target items, while the right table will include software name and version, development stage deliverables, checklists, SVVR, In-Use Test Reports, User Lists, Error Management, Maintenance, and Decommissioning, with document details (e.g., document number, author, approval hierarchy) accessible upon selection. The initial layout is shown in Figure 1.

Table 1: Key deliverables of SW development/utilization [1~3]

Category (level 1)	Subcategory (level 2)
Software Development Stage	- Software Verification and Validation Checklist - Software Verification and Validation Report - Software Certification
Software Utilization Stage	- Software User List - Error Management List (notification, resolution) - Maintenance, Decommissioning List

Table 2: Required documentation items for each stage of software development [1~3]

1. Design Requirement Stage Security Functional Requirements, Performance Requirements (e.g., speed, recovery time, response time), Attribute Requirements (e.g., software portability, user access management, acceptance criteria), External Interface Requirements (interactions with users, hardware, and other software), Design Inputs and Softwared Design Constraints, Technical Design Requirements, Software Engineering Requirements, Operating System Requirements, Testing, Inspection, and Acceptance Criteria, Installation Considerations, Security Requirements
↓
2. Design Stage Computational Sequence Definition (including mathematical models, physical models, control flow diagrams, control logic, data flow diagrams, process flow diagrams, data structures, process structures, and data-process integration)
↓
3. Implementation Stage Source Code, Update Files, and Identification Numbers, Compilation and Execution Requirements (including necessary auxiliary and library files), Execution Methods and Implementation Details, List of Computer Programs, User Manual for Computer Programs
↓
4. Integration Stage Software Component Interconnection Aspects
↓
5. Verification & Validation Test Stage Software Verification and Validation Test
↓
6. Installation/Inspection and Approval Test Stage Software Guidelines, Input/Output Criteria, Input/Output Formats, System Limitations, Error Messaging, Maintenance and Support Information
↓
7. Issuance of the Software Verification and Validation Report (SVVR) SVVR Contents: Cover Page, Revision History, Table of Contents, Software Design Requirements Documentation, Software Design Documentation, Software Implementation Documentation, Software Integration Documentation, Software Verification Test Documentation (including test plans and results), Software Installation/Inspection and Approval Test Documentation, User Manual and Other User Guidelines, List of Source Code Files
↓
8. Software Certification Issuance
↓
9. Configuration Management Configuration Identification: Software Certification (including software name, version, executable file name, usage domain, quality assurance procedures, usage restrictions, and special hardware considerations), SVVR, Software User List (including users, usage domains, and distribution dates), In-Use Test Reports and Related Documents, Computer Programs (including source code, object code, and backup files), Software Error Correction Documents, Other Supporting Software

Table 3: Document Management System Applied to ANSIM [5]

Category (level 1)	Subcategory (level 2)
Quality Assurance Manual	-
Software Development	Software Registration Status
	Independent Reviewer
	Software Development Documents (SVVR, Checklist, V&V Test Report)
	Software Certification
	In-Use Test Report
	Error Management
Commercial/General/Other Software	Software Registration Status
	In-Use Test Report
Forms	-
Procurement List	-

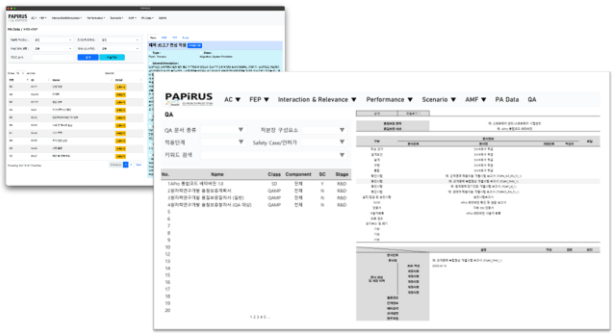

Fig. 1. PAPiRUS DB Management Page (upper) and QA System Design Layout (lower)

3. Conclusions

This study identifies the key deliverables and documentation items required for the quality assurance (QA) implementation of the APro code. It also analyzes the data managed in documentation systems such as KAERI ANSIM to determine the essential information that should be handled within the QA system. Based on these findings, the system layout was designed, categorizing the information to be managed in each section and deriving the key functionalities. The system development will proceed based on this design.

REFERENCES

[1] KAERI, Nuclear R&D Quality Assurance Manual (RD-QAM) (Internal document)
[2] KAERI, Nuclear R&D Quality Assurance Procedure (QAP-NP) (Internal document)
[3] KAERI, Software Quality Assurance Manual (Internal document)
[4] In-Young Kim, Jung-Woo Kim, Development of Quality Assurance Procedures and Guidelines for the Development, Verification, and Validation of the APro Code, 2024 Fall Conference of Korean Radioactive-waste Society, 2024
[5] KAERI ANSIM System (Internal System)