

# The Method of Requirements Traceability Analysis for Third-Party Auditors

Jang-Yeol Kim <sup>a\*</sup>, Jong-Gyun Choi <sup>a</sup>

<sup>a</sup>Advanced Instrumentation and Control Lab., Korea Atomic Energy Research, 111, Daedeok-daero, 989beon-gil, Yuseong-gu, Daejeon, 34057, Korea

\*Corresponding author: jykim@kaeri.re.kr

**\*Keywords :** Tracability Analysis, Process, Thread Path, Requirement, Functional, Non-Functional

## 1. Introduction

Traceability analysis refers to the systematic activity of tracking and verifying relationships between deliverables produced during the software development process (e.g., requirements specifications, design specifications, code specifications, and test documents) using requirement IDs. Specifically, it involves tracing a requirement forward, backward, and bidirectionally throughout the software development lifecycle of its associated design, code, and test artifacts. This process creates a traceability matrix to identify omissions, duplications, inconsistencies, and unnecessary requirements, missing requirements thereby ensuring quality and consistency. This activity is indispensable for third-party auditors and quality auditors. This paper proposes a method for creating a Traceability Matrix from the Request For Proposal(RFP) phase through the requirements phase to the integration testing phase. Based on this matrix, it outlines how to ensure each requirement ID is tracked without omission. Additionally, it presents a Naming Convention for assigning requirement IDs per project. The proposed traceability analysis method is practical and applicable to real project execution, not merely formalistic.

## 2. Methods and Results

This paper suggests a method for creating RFPs, requirement summaries, requirement definition tables, and requirement traceability matrices as traceability methods of requirements, and the traceability matrix were established for third-party auditors. Figure 1 shows the initial requirement traceability process, and Tables 1 through 5 are sequential matrices for execution the traceability process.

### 2.1 RFP(Request for Proposal)

An RFP (Request for Proposal) is an official document issued by an organization to proposals from external vendors for a specific project or service. The typical contents of an RFP include a project overview, requirements, proposal writing guidelines, evaluation criteria, and contract terms. To track requirements, an RFP template like Table 1 is necessary . This serves as a crucial starting point for requirement traceability.[1]

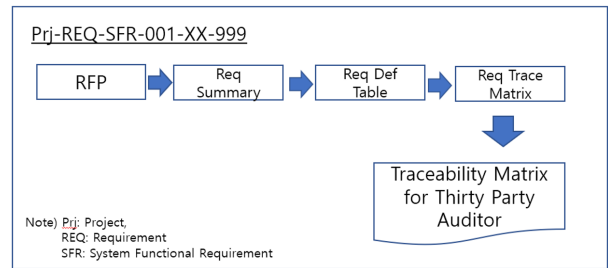


Fig. 1. Requirements Traceability Process

Table 1. RFP Template

Requirement ID			
Requirement Name			
Requirement Class		Acceptance Level	
Requirement Description			
Output Information			

The next process in the RFP is to compile a list of interviews with project stakeholders and schedule them to organize the requirements summary table.

### 2.2 Requirements Summary Table

Before creating the requirements summary table, define the requirements and assign identifiers. Requirements are categorized into functional requirements, performance requirements, interface requirements, data requirements, test requirements, security requirements, quality requirements, constraints, and project management requirements. Requirements are broadly divided into functional, non-functional, and interface. Detailed requirement classifications are shown in Table 2.

When the project scale is small, grouping requirements into two broad categories as follows can simplify, standardize, and specialize the requirements specification.

- o Technical and Functional Requirements
  - Functional Requirements, Security Requirements, Data Requirements, System Operation Requirements
- o Performance and Quality
  - Performance requirements, quality requirements, interface requirements

The next process in the requirements summary table is to create the requirements definition table as shown in Table 3.

Table 2 Requirement Categories and Identifiers

Requirement Category	Identifier	Requirement Definition	Number of Requirements
Function Requirement	SFR-Serial No.	SFR-Sub Req	
Performance Requirement	PER-Serial No.	PER--Sub Req	
Interface	SIR-Serial No.	SIR-Sub Req	
Data	DAR-Serial No.	DAR-Sub Req	
Test	TER-Serial No.	TER-Sub Req	
Security	SER-Serial No.	SER-Sub Req	
Quality	QUR-Serial No.	QUR-Sub Req	
Constraint	COR-Serial No.	COR-Sub Req	
Project Management	PMR-Serial No.	PMR-Sub Req	

Note) Abbreviations: SFR: System Functional, Requirement, PER: Performance Requirement, SIR: System Interface Requirement, DAR: Data Requirement, TER: Test Requirement, SER: Security Requirement, QUR: Quality Requirement, COR: Constraint Requirement, PMR: Project Management Requirement

### 2.3 Requirements Definition Table

The Requirements Definition Table is a document or form used to systematically organize and manage the requirements necessary for a project or system development. The Requirements Definition Table is created during the initial planning phase of a project, continuously updated throughout the development process, and uses as the baseline document for change management and impact analysis. Table 3 is provided to enhance traceability with the RFP.

Table 3 Requirements Definition Table

RFP	Req. Type	Req ID	Addition or Deletion	Req Name	Request/Change History	Design Output	Accept (Y/N)	Functional/Non-Functional	Source (RFP)	Notes	Audit Review

When creating the Requirements Definition Table, requirements IDs must be assigned uniquely as follows.

Pj-REQ-SFR-001-XX-999

① ② ③ ④ ⑤

① Project Name

② REQ: Requirement Identifier

③ SFR-001: RFP Requirement ID

④ XX: Detailed Task Classification (2 or 3 uppercase letters)

⑤ 999: Serial Number (001-999, assigned per task); when adding omitted numbers, append a lowercase English letter after 999 (e.g., 999a, 999b, etc.)

The next process in the requirements definition table is to create a requirements traceability matrix as shown in Table 4.

### 2.4 Requirements Traceability Matrix

A Requirements Traceability Matrix(RTM) is a tabular document that maps the relationship between requirements and project deliverables (design, code,

tests, etc.) to enable traceability. Traceability must be possible in three directions: forward traceability (Requirements → Design → Code → Test), backward traceability (Test → Code → Design → Requirements), and bi-directional traceability (covering both forward and backward). To satisfy this, the Requirements Traceability Matrix must be structured as shown in Table 4 and Table 5 respectively.

Table 4-1 Requirements Traceability Matrix (Continued)

No.	RFP	No.	Req ID	Added Yes/No	Works ID	Req Name (RFP)	Req Details/Change History	Implementation Plan	Acceptance (Yes/No)	Type	Infor Source	Source Evidence

Table 4-2 Requirement Traceability Matrix

The Person in Charge	Process ID	Window ID	Program ID	Interface ID	Unit Test ID	Integration Test ID	Inspection Item	Inspection Criteria	Inspection t Pass/Fail	Remarks

Table 5. Building a Traceability Matrix for Auditors

No	RFP	Task ID	Req ID	Req Definition Table	CM No	Traceability Analysis Table	Architect Design Document	Process ID	Window ID	Window Design/Menu Structure	I/F Window	Interface

Notes: CM: Configuration Management

## 3. Conclusion

Requirement traceability analysis is essential in the safety-critical software, being indispensable for enhancing quality assurance, continuing change management, improving project visibility, and facilitating communication: all crucial for regulatory compliance and thirty party audits. However, it also exists significant time and cost expenditures, increased complexity, dependency on requirement management tools (if any), risks of formalism, and maintenance difficulties. In this paper, the method of Requirements Traceability Analysis is expected to serve as countermeasures that enhance quality and reduce risks when properly implemented

## REFERENCES

- [1] [https://www.larksuite.com/ko\\_kr/blog/rfp-template](https://www.larksuite.com/ko_kr/blog/rfp-template)
- [2] Jangyeol Kim, Method of Safety-Related Software Development and Its Verification & Validation for Small Digital Devices, (DOI:10.20944/preprints202406.1038.v1) 19 June, 2024