

## 요구사항 명세서에 첨부하는 요구사항 추적표 작성 양식 제안

김대승\*

SNS ENG, 시스템엔지니어링 연구본부 이사

## A Suggestion on a Better Template for Requirements Traceability Matrix of a Requirements Specification

DaeSeung Kim\*

*Director, Systems Engineering R&D Department, SNS ENG Co., Ltd.*

**Abstract** : Most of systems engineers make a traceability matrix and attach it to their technical documents as a result of systems engineering activities. I have been working in the field of systems engineering for many years and have been watching traceability matrices created by systems engineers or developers from various companies. I have been thinking that some of them are not suitable in terms of purposes of traceability matrix. In this paper, I would like to suggest a right template for the traceability matrix in conformance to traceability purposes. The key is that traceability matrix should be created from higher level of requirements to current level of requirements.

**Key Words** : Requirements traceability matrix, Requirements Specification, Bi-directional traceability, Traceability matrix template

---

**Received:** November 11, 2015 / **Revised:** June 11, 2016 / **Accepted:** June 15, 2016

\* 교신저자 : DaeSeung Kim, [daeseung.kim@snseng.co.kr](mailto:daeseung.kim@snseng.co.kr)

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1. 서론

어떠한 시스템 개발 방법론을 적용한다 하여도 빠지지 않고 수행하겠다고 선언하는 작업이 요구사항 정의 및 분석활동이다. 대부분의 시스템 엔지니어 또는 요구사항 관리자들은 요구사항 분석활동의 중요성을 잘 알고 있지만 프로젝트의 일정 및 비용 절감 등의 이유로 가장 먼저 생략하고 바로 설계 단계로 넘어가는 것이 현실인 것 또한 사실이다.

Figure 1에서 보는바와 같이 프로젝트 시작단계에서 충분한 요구사항 분석을 거치지 않는다면 많은 재 작업들이 시스템 개발 마지막 단계에서 발생하게 되어 납기 지연 및 과도한 비용 증가로 이어지고, 궁극적으로 개발 시스템 품질에도 영향을 미치게 된다. 요구사항 분석 활동 결과의 중요한 산출물 중 하나이며, 요구사항 분석결과 검증과 누락된 요구사항을 식별하는 도구로 활용할 수 있는 요구사항 추적표 작성은 매우 중요한 활동이다. CMMI에서나 요구사항 공학에서는 모든 요구사항이 나오게 된 근거에서부터, 설계 및 시험에 이르기까지 관련된 모든 정보의 양 방향(Bi-directional traceability) 추적성 확보를 해야 한다고 언급하고 있다.

본 논문에서는 현재 많은 회사들이 사용하고 있는 추적표 양식의 문제점들을 지적하고, 추적표 작성의 목적을 충실히 달성하기 위한 올바른 양식을

**요구사항관련 이슈는 과도한 재작업, 납기 지연, 낮은 품질과 프로젝트의 실패를 야기.**



[Figure 1] Requirements Errors vs. Correction Cost [3]

제안함으로써 시스템 엔지니어링 산출물의 품질향상에 기여하고자 한다. 논문의 주요 구성은 시스템 엔지니어링 및 요구사항 관리에 있어 가장 핵심 프로세스라고 할 수 있는 추적성 관리의 개념과 중요성을 간단히 살펴보고 요구사항 추적 관리 결과물인 요구사항 추적표(Requirements Traceability Matrix)의 작성 방법에 대한 구체적인 방법과 양식을 제안한다.

## 2. 요구사항 추적관리

요구사항의 추적관리는 요구사항 문서 내에 존재하는 개별 요구사항 단위로 연결관계를 만드는 것을 의미하며, 이에 대한 개념도는 아래 Figure 2와 같다.

요구사항의 추적관리 당위성은 다음과 같이 정성적인 내용으로 정리할 수 있다.

- 1) 공급자(Supplier)가 고객이 원하는 제품을 개발하고 있다는 확신을 가질 수 있는 수단
- 2) 고객이 요구사항을 변경 요청하였을 때, 이에 따른 영향분석을 수행
- 3) 고객의 요구사항이 어떻게 설계되고, 구현되고, 테스트 되었는지를 확인(프로젝트의 진척관리)
- 4) 요구사항 변경이 승인되었을 때, 이와 관련된 요구사항들의 일관된 변경관리

또한 요구사항 간 추적성을 확보할 때 2가지의 원칙을 2014년 추계학회에서 발표한 “적용 가능한 실용적인 양방향 추적성 확보 방안제시”에서 정의한 바 있다.



[Figure 2] Schematic Traceability between Requirements

**[추적성 원칙 1]** 모든 상위요구사항은 반드시 1 개 이상의 하위 요구사항과 연결관계를 가져야 한다.

이는 고객의 요구사항들이 빠짐없이 시스템 요구 사항에 반영되었음을 확인하는 것으로 일반적으로 하나의 상위요구사항은 다수의 하위요구사항으로 도출되기 때문에 “1:n”의 추적관계성을 가지게 된다.

**[추적성 원칙 2]** 상위요구사항과 연결관계를 가지지 못하는 하위요구사항에 대해 필요성을 반드시 검토해야 한다.

추적성 원칙 2는 상위요구사항과 연결관계를 가지지 못하는 요구사항이 불필요한 요구사항(Gold plating)이 아닌지를 반드시 확인해야 함을 강조한 것이다. 요구사항은 단순히 문서상에 존재하는 텍스트가 아니라 이를 시스템의 기능으로 구현하고 테스트하기 위해 많은 공수와 비용이 발생하는 것임을 항상 인식해야 한다. 다시 말해 모든 요구사항은 존재하는 확실한 이유가 있어야 한다.

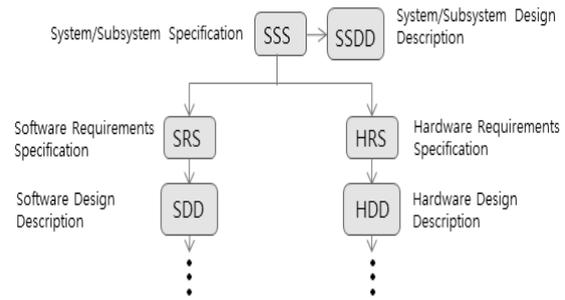
### 3. 문서 트리 및 요구사항 문서템플릿

대상 시스템의 체계개발 단계에서 프로젝트 수명 주기 동안 산출하는 산출 문서들과 문서들 간의 관계를 표시한 것을 본 논문에서는 문서 트리라고 부르기로 한다. 또한 각 산출 문서 내에 어떤 정보를 기술해야 하는 지를 표시한 문서 구조를 문서템플릿이라고 정의한다.

#### 3.1 시스템 공학 문서 산출물 트리

현재 무기체계 체계개발 사업 시 산출하는 전체 문서 중 일부문서들과 연관관계를 표시한 문서트리를 Figure 3에 나타내었으며, 이 연관관계 구조를 활용하여 요구사항 추적표 템플릿을 설명할 것이다.

소요군의 작전운용성능(Required Operational Capability)을 만족하는 체계요구사항을 기술한 문서인 SSS(System/Subsystem Specification)를 작성하고, 체계요구사항의 구현을 위한 설계 결정사항을 기술한 문서가 SSDD(System/Subsystem Design Description)이다. 또한 체계 요구사항의 기능 및



[Figure 3] Documents Tree as a result of systems engineering process

성능 요구사항들을 소프트웨어로 구현할 것인지 하드웨어적으로 구현할 것인지를 결정해서 이를 소프트웨어 요구사항 명세서(SRS, Software Requirements Specification)와 하드웨어 요구사항 명세서(HRS, Hardware Requirements Specification)로 풀어 작성하게 된다.

Figure 3에서 나타낸 것과 같이 반드시 순차적으로 작성되는 것은 아니나, 일반적으로 위에서 아래 방향 시간 순으로 체계공학결과 문서들이 작성되며, 화살표의 방향은 개별 요구사항들간의 추적성을 나타낸다. 이러한 추적성 연결의 이유/근거를 표시하기 위해 연결관계(Relationship type) 이름을 “Is Satisfied By”라고 부여하기도 하며, 사용자에 따라 자유롭게 표현하여 사용할 수 있다.

#### 3.2 요구사항 문서템플릿 예시

일반적으로 체계공학 산출물 작성 시 해당 문서의 템플릿은 국제 표준 템플릿, 방위사업청의 산출물 템플릿, 또는 회사의 표준 산출물 템플릿을 참고하여 해당 프로젝트의 특성에 맞게 선택하여 사용한다. 모든 체계공학 필수 산출문서에는 추적표를 반드시 첨부하게 되어 있다. Figure 4는 방위사업청에서 표준템플릿으로 제공하고 있는 소프트웨어 요구사항명세서의 예시이며, 5장의 “요구사항 추적성”장에 요구사항 추적표를 첨부하게 되어 있음을 확인할 수 있다.

본 논문에서는 많은 체계종합 회사들이 사용하고 있는 추적표 양식은 방위사업청에서 배포한 “소프트

목 차	
1 개요	1
1.1 적용범위	1
1.2 체계개요	2
1.2.1 체계 개발 경위	2
1.2.2 관련 기관	2
1.2.3 운용 사이트	2
1.3 문서개요	2
1.3.1 문서목적	2
1.3.2 문서 내용의 요약	3
1.3.3 보안 및 주의사항	3
1.3.4 식별자 부여 규칙	3
2 관련 문서	5
2.1 경부분서	5
2.2 비경부분서	5
3 요구사항	7
3.1 상태(state)와 모드(mode)	7
3.2 소프트웨어 현상항목 요구사항	8
...	...
5 요구사항 추적성	20
6 참고사항	21
6.1 용어/약어 설명	21
6.2 기타 참고자료	21
부록A	22

[Figure 4] Software Requirements Specification Template Provided by Defense Acquisition Program Administration

트웨어 기술문서 작성표준서식”을 기준으로 작성하고 있는데 이에 대한 기존의 문제점을 살펴보고, 이에 대한 보다 나은 해결 대안을 제시한다.

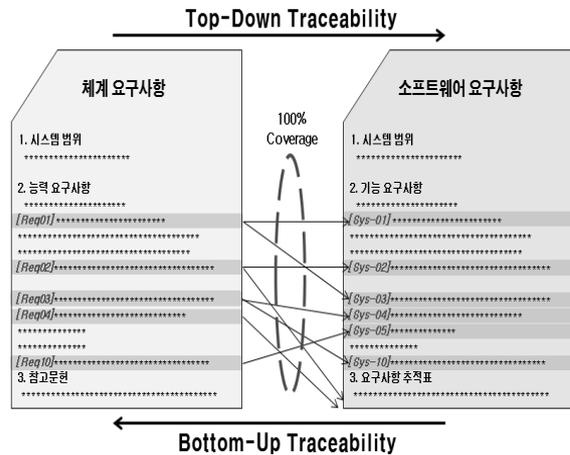
### 3.3 기존 요구사항 추적표 양식 문제점

기존의 추적표 양식의 문제점을 살펴보기 전에 먼저 요구사항 추적표를 누가 무슨 목적으로 만드는지를 생각해 볼 필요가 있다. 여기서는 소프트웨어 요구사항 명세서를 예시로 설명한다.

요구사항 추적표의 주요 목적이 모든 상위 요구사항(SSS, 체계요구사항 명세서)들 중 소프트웨어로 구현하기로 할당된 요구사항들이 빠짐없이 현재의 문서(SRS, 소프트웨어 요구사항 명세서)에 반영되었는지 확인하는 것이다.(Coverage 분석)

다시 말해 소프트웨어 개발팀이 체계 종합팀에서 할당해준 체계요구사항을 빠짐없이 100% SW 요구사항으로 변환하였는지를 감사(audit)하기 위한 것이다.

이런 관점에서 보면, 소프트웨어 요구사항 명세서에 첨부된 요구사항 추적표는 누구를 위한 것일까? 2장에서 언급한 [추적성 원칙 1]은 반드시 지



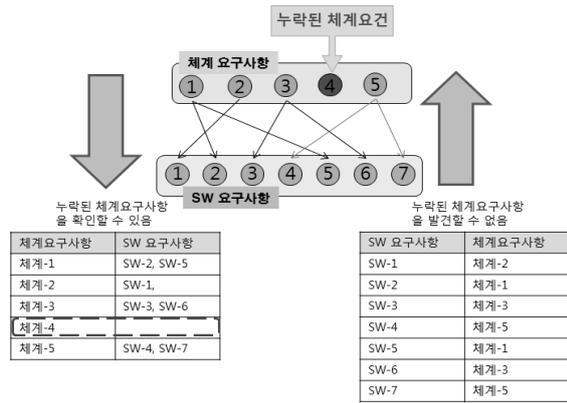
[Figure 5] Coverage Analysis through Traceability

켜져야 하는 원칙이며, 추적표 작성은 추적성 원칙 1을 반드시 검증할 수 있도록 작성되어야 한다. 결론적으로, 추적표 작성은 상위요구사항에서 시작하는 추적표 양식을 작성해야 한다. 아래 양식이 현재 방위사업청에서 권고하는 소프트웨어 요구사항 명세서의 추적표 양식이다. 이 양식은 저자가 제안하는 양식과 완전히 반대방식으로, 현재 현재요구사항(하위 요구사항)에서 시작해서 상위요구사항을 표현한 양식이다.

Table 1에서 보여주는 추적표는 하위문서(소프트웨어 요구사항 명세서)에서 도출된 요구사항이 상위요구사항인 체계요구사항과 어떻게 연결되었고, 다시 소프트웨어 요구사항보다 하위문서인 소프트웨어 설계기술서와 추적되는 상황을 나타내고 있다. 이 추적표의 문제는 상위요구사항인 체계요구사항이 소프트웨어 요구사항으로 모두 변환/반영되었는지를 확인할 수 없다는데 있다. 왜냐하면 소프트웨어 엔지니어가 도출한 요구사항들만을 기준으로 상위요구사항과 연결한 관점의 추적표를 작성하기 때문에 추적표

<Table 1> Requirements Traceability Matrix Provided by Defense Acquisition Program Administration

SW 요구사항 명세서 식별자	체계 요구사항 명세서 식별자	SW 설계 기술서 식별자	SW 시험 절차서 식별자
SRS-AAA-001	SSS-AAA-003	"추후결정"	"추후결정"



[Figure 6] Traceability Matrices to find a Missing Requirement of Upper Level System Requirements

에 누락된 상위요구사항을 표현할 수 없다.

### 3.4 요구사항 추적표 양식 제안

Figure 6에 나타난 바와 같이 체계 요구사항 4번이 소프트웨어 요구사항으로 연결을 가지고 있지 않음에도 불구하고, 방위사업청에서 권고하는 추적표 양식(Figure 6의 하단 오른쪽 추적표)은 체계요구사항 4번(체계-4)을 찾을 수 없음을 볼 수 있다. 그러나 하위요구사항에서 상위요구사항 관점의 추적표가 불필요하다는 의미가 아니다. 이 관점의 추적표는 [추적성 원칙 2]에서 말하는 불필요한 요구사항(Gold Plating)을 식별하는 데 사용할 수 있으며, 요구사항 규격서에 첨부할 추적표로 적절하지 못하다는 것이다.

시스템 엔지니어링 모든 산출문서에 포함되어야 하는 요구사항 추적표는 해당 문서의 상위문서의 요구사항 관점에서 추적표를 작성해야 한다고 제안한다. 이 개념을 본 논문에서 예로 든 소프트웨어 요구사항 명세서의 추적표인 경우 Table 2과 같이 적용할 수 있다.

## 4. 결론

요구사항 추적표는 상위요구사항을 빠짐없이 하위요구사항으로 반영했는지를 확인하고, 누락된 요구사항을 발견/보안하여 작성하는 문서의 품질을

<Table 2> Suggested Traceability Matrix for Software Requirements Specification

체계 요구사항 명세서 식별자	SW 요구사항 명세서 식별자	SW 설계 기술서 식별자	SW 시험 절차서 식별자
SSS-AAA-003	SRS-AAA-001 SRS-AAA-003	"추후결정"	"추후결정"

확보하는데 있다. 그러나 저자는 지금까지 많은 요구사항 추적표가 이러한 역할을 하지 못하는 양식으로 작성되는 경우를 보아 왔다.

이러한 목적을 달성하기 위해서는 각 단계별 시스템 엔지니어링 산출 문서에 포함해야 하는 추적표작성 관점을 작성 문서의 상위문서 요구사항에서 현재 작성중인 문서의 요구사항과의 연결성을 보여주는 추적표로 작성하기를 강력히 제안한다.

또한 추적성 확보라는 시스템 엔지니어링의 기본 활동을 올바르게 수행하기 위한 검증수단으로 추적표를 활용해야 하며, 이를 위해 추적표양식의 개선이 절실하다.

## References

1. DaeSeung Kim, Two Principles to Secure Practical Bi-directional Traceability, The Korea Society of Systems Engineering, 2014.
2. Technical Document Templates as Software Development Artifacts, Defense Acquisition Program Administration, 2013.
3. Leffingwell & Widrig, "Managing Software Requirements," Addison Wesley, 1999.
4. CMMI Institute, "Introduction to CMMI for Development", 2015.
5. Klasus Pohl, Chris Rupp, "Requirements Engineering Fundamentals", Rockynook, 2011.
6. INCOSE Systems Engineering Handbook, Fourth Edition, 2015.
7. Charles Wasson, "System Engineering Analysis, Design, and Development", Second Edition, Wiley, 2016.