

정보시스템개발 프로젝트에서의 효과적인 요구사항추적 관리 방안에 관한 연구

정 천 수*, 김 승 렬*

A Study on Effective Requirement Traceability Management Method in Implementation Project of Information System

Cheon-Su Jeong *, Seung-Ryeol Kim*

요 약

소프트웨어 개발 시 요구사항들이 개발시스템에 반영되었는지 추적하는 일은 매우 중요하다. 하지만 기존 연구들은 대부분 요구사항추적표를 개발관리방법론에 따른 전체 SDLC에 모두 적용하고 있다. 그리고 추적항목이 많으며 추적방법이 복잡하여 실제 정보시스템 개발에서는 실질적인 관리가 거의 이루어지고 있지 않아 실용성에 한계가 있다.

따라서 본 연구에서는 요구사항추적표를 제안요청 및 제안단계부터 SDLC단계 중에서 효과적으로 관리할 수 있는 적용범위와 요구사항추적에 꼭 필요한 항목만을 구성한 개선된 추적표를 제안하고 이를 적용한 K사의 정보시스템 개발프로젝트의 6개 실제 적용 사례를 통하여 그 효과를 확인 하였다.

▶ Keyword : 요구사항관리, 요구사항추적표, 산출물, 정보시스템 개발, 소프트웨어개발 생명주기

Abstract

It is very important to trace whether all the requirements has been reflected in the developed system. However, most existing researches apply Requirement Traceability Matrix(RTM) to the whole SDLC according to the development methodology. RTM has no practical value because it has

• 제1저자 : 정천수 • 교신저자 : 김승렬

• 투고일 : 2012. 01. 16, 심사일 : 2012. 01. 20, 게재확정일 : 2012. 02. 04.

* 국민대학교 비즈니스 IT전문대학원(Graduate school of Business IT, Kookmin University)

※ 본 논문은 2012년도 국민대학교 교내연구비 지원으로 수행되었음.

not only too many pursuit items but also its tracking method is very complicated and the practical management in developing the information system is not accomplished nearly.

Therefore, in this study, we proposed an enhanced RTM which was composed of only necessary items and allow us to manage the applied area effectively from Request for Proposal(RFP) or the proposal stage to SDLC stage then confirmed the effect through six actual applicable cases of information system development project of "K" company.

▶ Keyword : Requirement Management, Requirement Traceability Matrix, Deliverable, Information System Development, SDLC

1. 서 론

최근 들어와서 국내에서는 정보시스템 구축 차세대 프로젝트가 급증하고 있는 실정이다. 이는 IT기술의 급속한 발전과 국내의 정보시스템 인프라가 발전함에 따라 고객의 요구사항이 증가하고 있고 기업의 인수 합병에 따른 비즈니스 통합 등 각 중 이슈가 기존 시스템으로서의 고객의 요구사항을 충족시킬 수 없기 때문에 나타나는 현상이라고 볼 수 있다.

하지만 정보시스템 구축 차세대 프로젝트가 의도한 만큼의 효과를 내지 못하고 있고 계획된 프로젝트 오픈 일정을 넘기는 사례가 많다. 이는 프로젝트에 대한 무리한 일정계획수립 및 고객의 요구사항을 충분히 이해하지 못한 상태에서 진행함에 따라 나타나는 문제점이라고 할 수 있다. 이는 요구사항을 낸 사용자가 처음부터 프로젝트의 종료 시까지 지속적으로 요구사항을 모니터 하지 못하고 테스트단계 또는 마지막 단계에서 사용자 인수테스트에 참여하여 만족하지 못한 요구사항을 발견함으로써 나타나고 있다. 대형 프로젝트의 경우 서로 다른 관점을 갖는 다양한 이해 당사자가 존재하기 때문에 이해관계의 상충이 비일비재하게 발생한다. 특히 고객의 요구사항은 곧 개발 비용으로 직결되기 때문에 프로젝트 수행 업체와 고객 간에는 크고 작은 대립 구조가 형성되는 것이 일반적이다[1].

현실적으로 사용자가 개발된 시스템을 좀 더 이해하게 됨에 따라 사용자의 요구사항이 변경되거나 새로운 요구사항이 추가되는 경우, 혹은 정보기술의 급속한 발전으로 인하여 개발기간 중 적용기술이나 서비스를 변경해야 하는 경우, 그리고 외부적인 시스템의 적용환경 변화에 따른 변경 요구 등 요구사항의 변경 요인이 끊임없이 발생한다.[2]

Standish 컨설팅 그룹의 보고서인 [표 1]에 따르면 2009년에 IT기업의 프로젝트 수행결과, 실패 및 변경된 프로젝트가 10년 전인 1998년 74%에서 68%로 낮아지기는 했으나

2006년에 비해서는 성공률이 35%에서 32%로 크게 개선되지 않고 있으며 프로젝트가 취소되거나 사용자가 사용하지 않는 실패 프로젝트들의 실패요인을 보면 여전히 요구사항관리 미흡으로 인한 실패사유가 주요요인을 차지하고 있다[3].

표 1. CHAOS 보고서
Table 1. The CHAOS Report

구분 \ 년도	2009	2006	2004	2002	2000	1998	1996	1994
Successful	32%	35%	29%	34%	28%	26%	27%	16%
Challenged	44%	19%	53%	15%	23%	28%	40%	31%
Failed	24%	46%	18%	51%	49%	46%	33%	53%

요구사항은 고객과의 계약사항 이행 여부를 확인할 수 있는 중요한 근거 자료이며, 또한 개발 전 과정을 통해 지속적으로 변경되는 자료이다. 따라서, 이의 체계적인 관리는 프로젝트의 성공을 판가름하는 중요한 요소 중의 하나이다. 요구사항을 체계적으로 관리하기 위해서는 요구사항을 잘 정리하여 저장 및 보관하는 일뿐만 아니라, 요구공학의 제반 작업인 추출, 분석, 검증, 명세 등과 개발 프로세스 상의 모든 작업인 설계, 구현, 테스트 등의 산출물을 효과적으로 연계하여 관리할 수 있어야 한다. 요구사항은 다양한 채널을 통해서 접수되며, 일반적으로 대면 의사소통 하는 경우가 대부분이지만 SNS를 사용이 일반화 되면서 비대면으로 의사소통하는 경우도 많이 있다[4]. 이렇게 접수된 요구사항을 효과적이고 체계적으로 관리하기 위해서는 고객이 프로젝트 전 단계에 참여할 수 있어야 하고 이를 모니터 하기 위한 정형화 되고 이해하기 쉬운 면서도 고객이 요구사항 반영 여부를 쉽게 확인할 수 있는 요구사항추적 도구를 사용하는 것이 효율적 이다.

요구사항추적(Requirements traceability)이란, 고객의 요구사항으로부터 소프트웨어 개발과정에서 생산되는 산출물들 간의 관계를 정의한 것을 말한다[5]. 이러한 요구사항추적을 통하여 실제로 개발된 산출물들이 요구사항에 잘 부합하는

지 확인할 수 있고, 개발 과정에서 요구사항이 변경되었을 때, 그에 따라서 변경되어야 하는 산출물들이 어떠한 것이 있는지 파악할 수 있게 된다. 따라서 요구사항과 산출물의 일관성을 검증하는 추적기법이 다양하게 연구되어 왔다[6]. 그 중 많이 쓰이는 것이 요구사항추적표이다. 요구사항 추적표에서는 구성항목을 잘 정의 하여 프로젝트 전체의 각 단계별 산출물이 일관성 있게 연동되고 프로젝트 이해관계자가 쉽게 검증할 수 있도록 구성되어 요구사항의 추적을 효과적으로 관리 할 수 있어야 한다.

하지만 기존 연구들은 대부분 요구사항추적표를 사전준비 단계를 배제한 개발관리방법론의 전체 SDLC에만 중점을 두고 있어 요구사항의 최초 도출경로 관리가 안 되고 있으며 추적단계가 많아 실제 정보시스템 개발에서는 실질적인 관리가 거의 이루어지지 않고 있어 실용성에 한계가 있다.

이에 본 연구에서는 추적을 보다 용이하게 하고 요구사항의 도출경로를 명확하게 밝힐 수 있게 제안요청 및 제안단계부터 관리하고, SDLC전체 단계중에서 효율적으로 관리할 수 있는 적용범위를 제시하고 항목구성은 고객이 참여하여 검증할 수 있도록 하여 효과적으로 관리할 수 있는 개선된 추적표를 제안한다. 그리고 6개 프로젝트를 대상으로 제안된 추적표의 적용결과를 제시하고자 한다.

본 연구는 5장으로 구성되어 있으며 1장은 서론부분으로 연구의 배경 및 목적에 대하여 서술하고 있으며 2장은 이론적 배경으로 본 연구를 위한 선행연구 및 기존 연구에 대하여 기술하고 있다. 3장은 요구사항추적표의 항목을 도출하기 위한 산출물간의 추적성을 도식화하고 도출한 추적표의 항목을 이용하여 개선된 추적표를 작성하는 방법을 설명하고 있으며 본 연구를 실제 적용한 사례연구결과인 4장, 마지막으로 결론인 5장으로 구성되어 있다.

II. 이론적 배경

1. 소프트웨어 개발 생명주기 고찰

소프트웨어 위기라는 이야기가 나온 지 수십 년이 지났지만, 아직도 소프트웨어는 여전히 품질, 납기 문제와 내재된 결함을 갖고 있다. 이러한 소프트웨어가 갖는 문제점을 해결하기 위해 소프트웨어 개발 프로세스에 소프트웨어 개발 생명주기(SDLC Software Development Life Cycle)모델을 도입하게 된다. SDLC는 소프트웨어를 개발하기 위한 정의 과정, 개발 과정, 유지보수 과정, 폐기 과정까지를 하나의 연

속된 주기로 보고, 효과적으로 수행하기 위하여 방법론을 모델화 한다. 그리고 소프트웨어 공학을 실제 구현하기 위해 사용되는 프레임워크이다.

소프트웨어 위기로 인하여 소프트웨어 개발을 효과적으로 수행하기 위한 방안이 모색 되었다. 소프트웨어를 획득하는 과정에서 나타나는 소프트웨어 위기를 극복하기 위한 방안이 필요하게 되고 효과적으로 소프트웨어를 개발하기 위해 표준화된 수행 방법과 절차가 필요하게 되었으며, 고품질의 소프트웨어를 획득하는 데에도 일정 수준 이상의 생산성 확보가 필요함에 따라 체계적인 소프트웨어 개발이 필요하게 되었다. SDLC는 정보시스템 개발 과정을 모니터링하고, 필요한 경우 설계변경 내지 프로젝트 중단의 조치를 취할 수 있도록 개발 과정을 단계화하여 매 단계마다 산출되는 중간 산출물을 점검, 확인함으로써 성공적인 시스템을 개발하는 프로젝트 관리 기법으로 활용되고 있다.

SDLC절차는 프로젝트계획을 세우고 요구사항 정의(Requirement Definition)단계를 거쳐 타당성 검토시 선택된 시스템 개발에 대한 요구사항을 식별, 상세화 하는 작업을 진행하게 된다. 설계(Design)단계에서는 고객의 요구사항에 기초하여 프로그램 개발을 위한 사양 작성 및 개발 프로세스에 대한 새로운 요구사항을 관리하기 위한 공식적인 변경관리 프로세스를 정의한다. 개발(Development)단계에서는 프로그램밍을 착수하고, 시스템을 운영할 수 있는 형태를 갖추고 다양한 수준의 테스트를 통해 개발 시스템에 대한 검토와 검증이 수행된다. 개발의 마지막으로 이행(Implementation) 단계에서는 새로운 정보시스템에 대하여 운영환경을 구축하고 사용자 인수테스트를 수행하며 개발결과를 운영시스템으로 이관하여 운영하게 된다.

2. 요구사항 관리의 이해

2.1 요구사항 관리의 정의

요구사항이란 문제의 해결 또는 목적 달성을 위해서 시스템이 가져야 할 기능이나 시스템이 만족해야 할 조건이다. 이러한 요구사항을 프로젝트와 관련된 이해 관계자들로부터 추출, 구성 및 문서화하고 변경에 대한 동의를 설정해 관리하는 시스템적 활동을 요구사항 관리라고 한다. 그래서 요구사항 관리는 참여자들 사이에서 효과적인 의사소통 전략을 사용하여 충분한 합의와 협의를 통해 공통의 이해를 구축하고 개발 생명주기의 전체 기간 동안 지속적인 관리를 제공한다[7]. 이를 위해서는 요구사항의 도출부터 개발주기 마지막 단계까지 요구사항의 변화 및 추적을 관리할 수 있는 도구를 사용하여 효율적인 관리를 하게 된다.

2.2 요구사항 관리 프로세스

요구사항을 추출하고 관리하는 것이 요구사항 관리 프로세스이다.

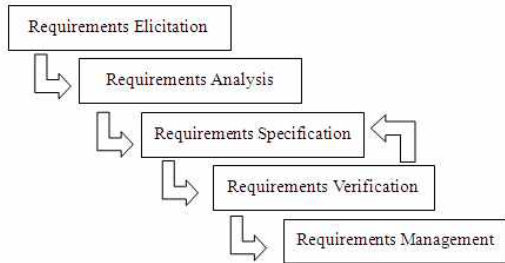


그림 1. 요구사항 프로세스
Fig. 1. The Requirements Process

Davis, Karl E. Weigers 등 많은 사람들이 요구사항 프로세스에 대한 모델을 [그림 1]과 같이 요구사항 도출 (Requirements Elicitation), 요구사항 분석(Requirements Analysis), 요구사항 명세(Requirements Specification), 요구사항 확인(Requirements Verification), 요구사항 관리(Requirements Management)의 5개의 프로세스로 구성된 요구사항 관리 프로세스 모델을 제시하였다[8,9].

요구사항의 세부 절차가 있는 소프트웨어 요구사항프로세스인 SRP(Software Requirements Process)에 대하여는 [그림 2]와 같이 요구사항 개발단계와 요구사항 관리단계로 구성되며 요구사항은 요구사항들이 구현되어 제품화 될 때까지 변경 사항을 유지 관리하는 절차라고 제시하고 있다[10].

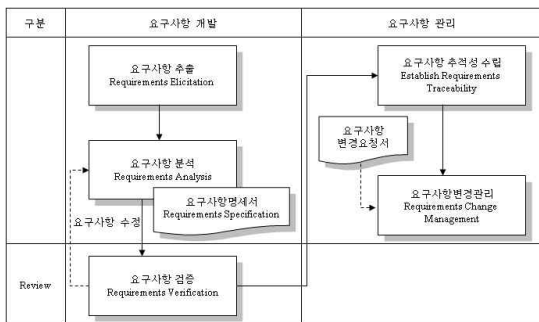


그림 2 소프트웨어 요구사항 프로세스
Fig. 2. SRP(Software Requirements Process)

요구사항관리에서는 요구사항 추적표를 작성하여 요구사항을 관리한다. 그리고 요구사항 변경 관리의 요구사항 변경이 요구되었을 때 소프트웨어의 개발 범위와 구현 가능성을 고려하여, 요구사항을 변경하여 개발 되어 지고 있는 소프트웨어의 변경된 요구사항이 잘 반영되도록 하는 작업이다[10].

3. 요구사항추적 표 고찰

3.1 요구사항 추적 모델

소프트웨어 개발과정에서의 관리, 유지보수의 중요성이 점차 증가함에 따라서 요구사항추적에 대한 중요성도 증가하고 있다. 여기서 요구사항추적이란, 고객의 요구사항으로부터 소프트웨어 개발과정에서 나오는 산출물들의 관계를 추적관계를 통해 기술하는 것을 말한다[11].

Lee M. Williams의 연구에서는 요구사항을 추적관리하기 위한 정보시스템의 개념을 제안 하고 있는데 특히, 이 모델에서는 설계 및 개발명세 뿐만 아니라 위험 및 기술성능 측정 명세 산출물과 요구사항과의 추적도 제시하고 있다. 요구 관리정보시스템은 요구사항을 데이터베이스로 저장하면서 다른 산출물간의 일관성과 완전성을 검증하기 위해 각 산출물마다 항목 단위별로 식별자를 두고, 그 식별자를 각 산출물에서 참조하도록 설계하였는데, 산출물마다 포함된 요구사항 항목의 식별자를 통해 요구사항을 추적하는 개념이다[12].

Grant의 추적모델 연구에서는 [그림 3]과 같이 요구사항을 요구사항 명세 이전인 사전요구사항(Pre-RS : Pre-Requirement Specification)과 요구사항명세 이후인 사후요구사항(Post-RS : Post-Requirement Specification)으로 나누고 추적 시 사전요구사항과 사후요구사항을 통합하여 추적하는 것을 설명하고 있다[13,14].

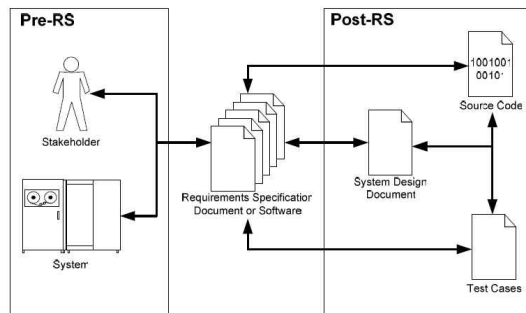


그림 3. Pre-RS 및 Post-RS의 추적성
Fig. 3. The Pre-RS and Post-RS Traceability

이는 비즈니스 요구사항에 해당되는 사전요구사항을 요구사항 명세서(SRS : Software Requirements Specification)의 진원지로 인식하고 사용자가 참여하여 요구사항을 확인하고 합의 하는데 필요하다. 또한 개발완료 후 사용자가 비현실적 기대를 하지 않고 합의된 품질에 승인할 수 있는 근거가 되기도 한다[15]. 하지만 기존 추적연구들의 대부분이 사후 요구사항(Post-RS)에 치우쳐 있다. 따라서 본 연구에서는

사전요구사항을 포함한 추적기법을 사용하고자 한다.

3.2 요구사항 추적 방식

추적기법의 종류는 [표 2]와 같이 수작업방식 (Manual Techniques), 시스템을 이용한 동적방식(Dynamic Techniques), 자동화된 도구(Automated Tool)로 구분할 수가 있다[13,14].

동적방식(Dynamic Techniques)은 많은 시간과 노력이 소요되며 비용이 많이 드는 단점이 있으며 소프트웨어 개발 초기 단계에서 사용하기에는 한계가 있다[8]. 그리고 자동화된 도구(Automated Tool)는 추적에 관련된 기본항목 및 프로세스가 도구에 반영되어 있어 관리의 효율성 및 자료추적에 도움이 되지만 도입 비용이 발생하고 도구를 충분히 익히고 사용해야 한다.

표 2. 요구사항 추적기법의 종류
Table 2. The type of Requirements Traceability Techniques

구분	Pre-RS	Post-RS
수작업(Manual Techniques)		요구사항추적 매트릭스(Requirement Traceability Matrix)
		요구사항추적표(Requirement Traceability Table)
동적(Dynamic Techniques)		정보검색(Information Retrieval)추적기법 이벤트기반추적(Event-Based Traceability) 추적기법 시나리오기반추적(Scenario-Based Traceability)기법
자동화된 도구(Automated Tool)	[IBM] Rational Requirements Composer, Rational Doors, Requisite Pro [Boland] Caliber RM	

이렇듯 추적기법마다 장 단점이 있어 어떤 것이 좋고 나쁘다고 논하기는 어렵다. 본 연구에서는 적용이 쉽고 도입비용이 적으며 사용이 쉬우면서 개발초기 단계에서부터 최종단계까지 지속적으로 추적 가능한 수작업기법의 요구사항추적표를 대상으로 한다.

III. 요구사항추적표를 통한 요구사항 관리

1. 추적대상 산출물 및 추적 항목

본 연구는 추적표를 구성하는 산출물 대상을 정의하기 위해 1990년대 중반까지 가장 많이 사용했던 소프트웨어 개발

방법론인 정보공학방법론과 1990년 중반 이후 새로이 사용이 증가하고 있는 객체지향 및 컴포넌트 기반 방법론에 따른 SDLC의 주요 산출물을 참조하고 있다.

정보공학은 아직까지도 시스템구축 방법의 주를 이루고 있는 안정된 방법론 중의 하나이다. 새로운 방법론들이 나오고는 있지만 기업을 대상으로 하는 정보시스템의 개발에서는 정보공학 방법론에 익숙해져 있는 기업의 이해관계자가 많아 새로운 방법론에 대한 이해가 부족한 현실이다. 그러기 때문에 정보공학의 기본 SDLC를 벗어나기가 어려운 환경에 있다.

추적 대상 산출물은 정보공학방법론 및 CBD방법론의 필수 산출물중 요구사항을 요청한 이해관계자들이 참여하여 확인해야 하는 산출물을 대상으로 [표 3]과 같이 제시 하고자 한다[16]. 구성필드를 보면 SDLC에 해당하는 ‘단계’, 해당 단계의 ‘산출물’, 산출물을 구성하고 있는 항목들을 도출 하여 ‘구성항목’에 표시 하였으며, 이 중에서 요구사항 관련 항목을 추출하여 ‘요구사항관련항목’으로 표시하였는데, 이 항목은 이후 요구사항추적표에 사용될 추적필드중의 하나가 될 것이다.

표 3. SDLC의 주요 산출물 구성 항목
Table 3. The configuration items of SDLC deliverables

단계	산출물	구성항목	요구사항관련항목
사전준비	제안요청서	배경, 목표, 범위, 역할, 고려사항, 일반사항	범위(단위시스템, 요청기능)
	제안서	제안목표, 범위, 역할, 구축방법론, 시급성, 제안가액	범위(단위시스템, 제안기능)
기획	프로젝트수명계획서	구축범위, WBS, 조직, 인력계획, 일정, 예산, 총괄	구축범위(단위시스템, 상세기능)
분석	요구사항정의서	요구사항ID, 요구사항명, 유형, 관련요구사항, 요구사항내용, 전제조건, 관련부서, 담당자	요구사항ID, 요구사항명, 유형, 관련요구사항ID, 요구사항내용, 관련부서, 담당자
	Use Case 모형기술서	Use Case Diagram(역사, Use Case 명, 설명)	Use Case 명, 설명
	Concept 모형기술서	Class Diagram(이름, 개수, 속성, 설명)	Class 명
	이벤트 기반 추적기법	이벤트ID, 포고서ID	이벤트ID, 포고서ID
설계	외연설계서	외연ID	외연ID
	내연설계서	내연ID	내연ID
구현	프로그램 소스 코드	프로그램명, 클래스명, 메소드명, 변수명, 리턴값, 리터럴값	프로그램ID
	단위테스트 시나리오	단위 Test Case ID, 테스트결과	단위 Test Case ID
테스트	통합테스트 시나리오	통합 Test Case ID, 테스트결과	통합 Test Case ID
	사용자 인수테스트 시나리오	인수 Test Case ID, 테스트결과	인수 Test Case ID

주요이탈릭체 : 객체지향 및 CBD 산출물

요구사항과 구체적인 관련 여부를 판단하는 방법은 해당 필드가 목적, 기능, 업무, 범위 등과 같이 행위를 나타내면서 업무와 기능을 담은 항목이거나 주체, 부서, 사용자 등과 같이 행위주체를 담은 항목일 경우관련이 있는 것으로 선정한다 [17,18].

사전준비 단계의 산출물은 프로젝트가 내부에서 기안되고 프로젝트 발의가 되면 실질적인 활동으로 제안요청서를 공식하게 되는데, 이 제안요청서는 소프트웨어를 개발 할 주체가 접하게 되는 최초의 요구사항임을 알 수 있다. 제안요청서에는 제안요청 배경 및 목표, 요구하는 범위 등이 상위수준으로 기술되어 있다. 여기에서 제안요청 범위를 기반으로 제안서의 수행범위를 작성 하게 되고, 상위의 요청범위에 대하여 단위

시스템 및 상위 기능별로 구축방안을 제시한다.

착수단계에서는 제안서에 기술한 구축범위를 수행하기 위한 프로젝트 수행계획서를 작성하게 되며, 제안서에 포함된 상위기능을 세부기능으로 나누고 수행할 주체에 대한 R&R을 정의한다.

분석 또는 요구분석 단계 산출물은 사전준비 단계 및 프로젝트 수행계획서를 통해 이미 상위수준의 요구사항에 대한 식별이 된 상태이기 때문에 소프트웨어 개발을 위한 상세화가 필요하다. 상세화는 [그림 2] SRP의 요구사항 개발과정을 거쳐 요구사항을 명세화하고 검증한다. 요구사항 명세의 경우 고객이 제공하는 시스템 요구사항 명세가 될 수 있으며, 일반적인 소프트웨어 프로젝트에서는 계약서, RFP, 사업수행계획서와 같은 자료를 통한 요구사항 수집도 이에 해당된다고 볼 수 있다[19]. 요구사항 명세서 또는 Use Case모형기술서에서는 요구사항명과 기능 또는 비 기능 구분을 하고 요구사항을 달성하기 위한 전제조건과 관련부서 및 관련담당자를 항목으로 두어 SDLC의 각 단계가 끝날 때 마다 관련담당자가 단계말 검토에 참여하여 요구사항에 대한 반영여부를 확인할 수 있도록 하고, 향후 요구사항의 변경이 발생할 때 영향도 파악 등을 위하여 관련요구사항 항목을 두어 관리하도록 본 연구에서는 필수적으로 작성할 것을 제안한다.

설계 또는 아키텍처 정립단계에서는 요구사항에 대한 설계 산출물인 화면설계서 및 Class모형 기술서등 각 종 산출물을 통하여 소프트웨어 개발을 위한 요구사항이 구체화된다. 구체화된 설계서들을 기반으로 구현 및 테스트를 또는 점진적인 개발단계를 거쳐 요구사항에 대한 소프트웨어를 생산해 내고, 이 단계에서는 소프트웨어의 실제인 프로그램과 구현 Class들이 존재하며 단위테스트 및 통합테스트에 대한 테스트 케이스, 시나리오, 결과에 대한 산출물을 작성하게 된다.

마지막으로 이행 또는 인도단계에서는 사용자의 요구사항이 제대로 반영되었는지 확인 할 수 있게 요구사항을 직접 제기한 사용자가 참여하여 인수테스트를 거쳐 최종 요구사항의 반영여부를 확인하게 된다.

[표 3]에서 추출한 산출물 및 추적항목을 가지고 추적하기 위한 항목을 정하기 위하여 [그림 4]와 같이 산출물 및 산출물을 구성하는 항목간에 추적성을 도식화 하였다.

단계	산출물	요구사항 추적 항목			
사전준비 단계	제안요청서	타위시스템	요구기능		
	제안서	타위시스템	제안기능		
착수단계	프로젝트수행계획서	타위시스템	요구기능	상세기능	
분석 단계	요구사항명세서	요구사항ID	요구사항명	유형	관련부서 담당자
	UseCase모형기술서	UseCase ID	UseCase 설명		관련요구사항ID
	Concept모형기술서	Class명	Class설명		
이행 단계	화면설계서	화면ID	화면설명		
	화면설계서 보고서요청서	보고서ID			
	컴포넌트명세서, 설계서	컴포넌트ID			
구현 단계	Program Source Code	Program ID	ERTest Case ID		
	단위테스트시나리오	테스트 Case ID			
테스트 단계	통합테스트시나리오	통합Test Case ID			
이행 단계	사용자인수테스트 시나리오	인수Test Case ID			

그림 4. 산출물간 요구사항 추적
Fig. 4. The Requirements Traceability between deliverables

요구사항관련 필드가 있는 산출물만을 추출하여 상위 산출물과 하위 산출물을 순방향으로 추적한 것이다. 점선 부분은 하나의 산출물 내에서 서로 연관관계가 있는 경우의 추적을 나타낸다. 그리고 분석단계에서 추적할 요구사항명을 상세하게 기록하고 이후 단계의 추적은 산출물을 식별하기 위한 유일한 ID체계로 연관성을 추적할 수 있게 항목을 구성한다.

2. 추적표 작성

2.1 추적표 항목 구성

III장 1절에서 추출한 산출물과 추적항목을 기반으로 작성한 요구사항추적표는 [그림 5]와 같이 구성하였으며, 객체지향 및 CBD 산출물은 각 단계별로 정보공학방법론 산출물과 대치하여 사용할 수 있으므로 추적표 작성 표기에서는 제외하고 설명 하고자 한다.

사전요구사항(Pre-RS)이 도출되는 사전준비단계에는 요구사항의 발생자인 발주자의 제안요청서부터 요구사항추적표를 작성하게 되는데 단위시스템 또는 어플리케이션명과 제안요청 기능명을 기록한다. 그리고 공급자가 작성하는 제안 기능명에는 제안서에 기록된 내용을 바탕으로 실질적으로 구축하여 서비스를 제공할 기능명을 기록하게 되며, 발주자와 공급자간의 요구사항을 상호 검증 및 추적할 수 있는 첫 단계라고 할 수 있다. 요구사항명과 단위시스템 또는 어플리케이션명은 제안요청서에서 기록된 용어 및 명칭을 준용하여 혼선을 최소화할 수 있게 작성해야 한다.

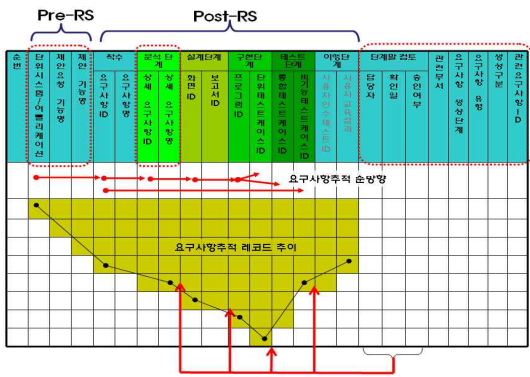


그림 5. 요구사항 추적표
Fig. 5. The Requirements Traceability Table

사후요구사항(Post-RS)은 제안된 내용을 토대로 SDLC 각 단계별로 상세화되고 요구사항이 각 산출물간 추적될 수 있도록 작성하게 된다. 착수단계에서는 프로젝트 수행계획서에 구축할 요구사항의 기능범위를 작성하여 요구사항의 추적 연결고리가 되는 요구사항ID를 식별하여 작성하게 된다.

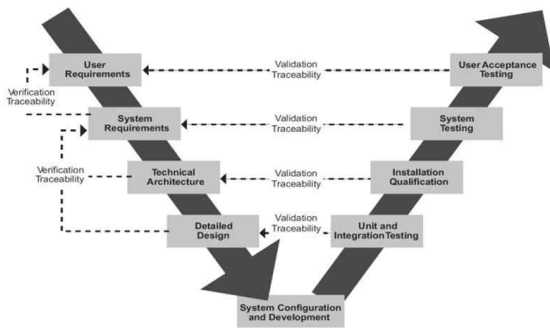


그림 6. V 모델
Fig. 6. V Model

분석단계를 거쳐 요구사항이 상세화 되면서 추적항목으로 상세요구사항ID를 도출하여 설계, 구현 및 테스트 단계에 걸쳐 진행되는 요구사항추적의 기본 단위로 결정 한다. 테스트는 [그림 6]의 V모델에서 보여 주듯이 SDLC의 각 단계와 그에 상응하는 소프트웨어 테스트 관계를 가진다[20].

이행단계에서는 요구사항을 제기한 사용자가 요구사항에 대하여 제대로 구현되었는지 확인하기 위하여 사용자 인스테스트와 교육을 받게 된다. 사용자 인스테스트는 V모델에서 알 수 있듯이 사용자 요구사항정의 단계에서 도출된 내용을 검증하는 하므로 추적표에서도 추적을 하기 위한 항목관계는 요구사항도출 단계의 추적항목과 연결이 된다. 따라서 [그림 5]의 요구사항 추이에서 보듯이 요구사항의 추적관계 레코드가 줄어들고 테스트단계의 추적항목과 연속성을 가진 다기 보다는 별도로 관리된다. 따라서 본 연구에서는 요구사항추적표

관리의 효율성과 편의성을 위하여 이행단계의 추적표 작성을 사용자 인스테스트 결과서 및 사용자 교육결과서로 대체하여 관리하는 것을 제시하고자 한다.

추가적으로 단계말 검토 항목을 두어 사용자가 SDLC의 각 단계마다 참여하여 요구사항 반영에 대한 검토 및 추적을 할 수 있게 함으로써 프로젝트 막바지에 요구사항이 변경되어 프로젝트 전체의 납기 및 품질에 영향을 줄 수 있는 위험요인을 줄일 수 있도록 추적항목을 구성하며, 요구사항의 변경 및 구축 시 관계되는 요구사항을 파악할 수 있도록 하여 각각의 요구사항이 가지는 영향도를 쉽게 파악 할 수 있도록 관련요구사항ID 항목을 둔다. 기타항목으로는 요구사항의 생성시기를 파악할 수 있게 생성단계를 두고, 요구사항의 기능유형을 구분할 수 있는 요구사항유형과 요구사항의 변경사유를 알아 볼 수 있게 생성유형 항목을 둔다.

2.2 추적표 항목 구성

본 연구에 제안하는 요구사항추적표는 Rre-RS에서 제시된 산출물에서 식별된 요구사항이 Post-RS의 산출물에 연관성을 가지고 요구사항을 쉽게 추적할 수 있도록 하며, 단계말에 사용자가 참여하여 요구사항에 대하여 검토하며, Post-RS에서의 요구사항 변경사항에 대하여 그 발생시기와 근거를 기록하게 하여 요구사항추적을 용이하게 한다. 이에 따라 다음과 같은 작성규칙을 제시하고자 한다.

2.2.1 Pre-RS 요구사항 추적표 작성

- 요구사항은 반드시 하나의 단위시스템 또는 어플리케이션에 속해 있어야 한다.
- 요구사항의 도출 경로가 명확하도록 제안 기능은 제안 요청 기능과 맵핑될 수 있게 작성되어야 하며 기능명칭에는 식별자를 사용하지 않고 요구기능에 해당하는 내용을 작성한다.

2.2.2 Post-RS 요구사항 추적표 작성

- 분석단계까지의 요구사항관련 추적항목에는 유일한 식별자(UID ; Unique identification number)와 요구사항명을 작성하여 추적표의 기준으로 삼고, 설계단계부터는 추적이 용이성을 위해 UID로 작성하며 구체적인 내용확인 식별자와 연결된 해당 산출물을 통해서 검증한다.
- 설계단계의 추적항목인 화면ID와 보고서 ID는 해당사항이 있는 경우에 작성을 하고 해당사항이 없는 배치 프로그램등은 화면ID 항목을 이용하여 작성하고 추가로 추적항목을 두어 의미를 파악할 수 있게 한다.
- 구현단계의 프로그램ID는 요구사항을 처리하는 주 프로그램을 기록하고 분기 프로그램 및 공통적으로 사용

하는 프로그램은 기록하지 않는다.

2.2.3 관리 항목

- 단계말 검토항목은 반드시 요구사항을 제기한 담당자가 작성하고 승인여부항목은 승인, 조건부승인, 미승인중 선택하여 작성하며 승인 이외의 항목을 선택 시에는 사유를 기록하도록 한다.
- 요구사항의 변경이 발생하였을 때를 관리하기 위해 생성단계 항목에는 변경발생 시점의 SDLC단계를 기록하고 생성구분항목에 요구사항의 추가, 수정, 삭제 유형 중에 하나를 선택하여 작성한다. 삭제의 경우에도 요구사항을 지우지 않고 관리 될 수 있도록 한다.

요구사항의 유형에는 비기능, 기능중 선택하여 기록하며 다른 요구사항과 관련이 있는 경우에는 관련 요구사항ID에는 착수 또는 분석단계에 식별된 요구사항ID를 작성하도록 한다.

IV. 요구사항추적표 적용사례

1. 추적표의 적용사례

제안하고 있는 요구사항추적표를 프로젝트에 실제 적용한 사례를 [그림 7]과 [그림 8]에 제시하고 있으며, 사례는 K그룹의 계열사에서 수행한 프로젝트로 2곳의 계열회사에 적용한 추적표 이다.

[그림 7]은 K은행은 차세대 프로젝트중 하나로 프로젝트 관리, 품질보증, 테스트관리, 변경관리, 형성관리, 방법론관리 업무로 구성되어 있으며 차세대 프로젝트를 통합적으로 관리하기 위한 품질관리시스템을 구축하는 프로젝트이다. 본 프로젝트는 제안서요청기능보다 제안서내용과 프로젝트계획서 내용이 상세화되어 있어 별도 추적필드로 관리하고, 이행단계의 추적은 사용자 인수테스트 결과서로 대체하기로 프로젝트 착수시 테일러링을 하였다. 기타항목을 추가로 두어 요구사항의 변경되었을 때 사유를 작성하며 변경요청서가 있는 경우에는 변경요청서 ID를 기록하게 하였다. 이처럼 본 연구에서 제시하는 요구사항추적표를 기반으로 차세대 각 프로젝트의 상황에 맞게 유연하게 적용하였다.

[그림 8]은 K증권의 국제회계기준(IFRS ; International Financial Reporting Standards) 시스템 구축 프로젝트로서 비즈니스 요구기능이 사전 업무분석 컨설팅을 통하여 도출이 되었고 분석단계와 설계단계는 시스템 구현을 위한 시스템 위주의 단계였다.

따라서 컨설팅 결과에 따라 작성된 요구기능 부분에 대해서는 제안요청서, 제안서내용이 동일하여 착수단계의 요구기능으로 대체하기로 하였다. 추가적인 항목으로 시스템화 여부 두어 비기능 요구사항중 프로세스정립 같은 시스템화 되지 않는 부분을 쉽게 구별하여 추적 관리할 수 있게 하였으며 나머지는 본 연구에서 제시한 추적표를 사용 하였다.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	
1 요구사항추적표(테스트단계)																					
NO	단위시스템	제안요청기능	계안기능	작성단계	요구사항명	분석단계	설계단계	구현단계	테스트단계	추가항목											
		요구사항명	요구사항명	요구사항명	요구사항명	요구사항명	요구사항명	요구사항명	요구사항명	요구사항명	요구사항명	요구사항명	요구사항명	요구사항명	요구사항명	요구사항명	요구사항명	요구사항명	요구사항명	요구사항명	
3		인도문자	인도문자	Req_sam_1	인도문자	인도문자	인도문자	인도문자	인도문자												
4	비밀보유	인도문자	인도문자	Req_sam_1	인도문자	인도문자	인도문자	인도문자	인도문자												
5	비밀보유	인도문자	인도문자	Req_sam_1	인도문자	인도문자	인도문자	인도문자	인도문자												
6	비밀보유	인도문자	인도문자	Req_sam_1	인도문자	인도문자	인도문자	인도문자	인도문자												
7	비밀보유	인도문자	인도문자	Req_sam_1	인도문자	인도문자	인도문자	인도문자	인도문자												
8	비밀보유	인도문자	인도문자	Req_sam_1	인도문자	인도문자	인도문자	인도문자	인도문자												
9	비밀보유	인도문자	인도문자	Req_sam_1	인도문자	인도문자	인도문자	인도문자	인도문자												
10	비밀보유	인도문자	인도문자	Req_sam_1	인도문자	인도문자	인도문자	인도문자	인도문자												
11	비밀보유	인도문자	인도문자	Req_sam_1	인도문자	인도문자	인도문자	인도문자	인도문자												
12	비밀보유	인도문자	인도문자	Req_sam_1	인도문자	인도문자	인도문자	인도문자	인도문자												
13	비밀보유	인도문자	인도문자	Req_sam_1	인도문자	인도문자	인도문자	인도문자	인도문자												
14	비밀보유	인도문자	인도문자	Req_sam_1	인도문자	인도문자	인도문자	인도문자	인도문자												
15	비밀보유	인도문자	인도문자	Req_sam_1	인도문자	인도문자	인도문자	인도문자	인도문자												
16	비밀보유	인도문자	인도문자	Req_sam_1	인도문자	인도문자	인도문자	인도문자	인도문자												
17	비밀보유	인도문자	인도문자	Req_sam_1	인도문자	인도문자	인도문자	인도문자	인도문자												
18	비밀보유	인도문자	인도문자	Req_sam_1	인도문자	인도문자	인도문자	인도문자	인도문자												
19	비밀보유	인도문자	인도문자	Req_sam_1	인도문자	인도문자	인도문자	인도문자	인도문자												
20	비밀보유	인도문자	인도문자	Req_sam_1	인도문자	인도문자	인도문자	인도문자	인도문자												
21	비밀보유	인도문자	인도문자	Req_sam_1	인도문자	인도문자	인도문자	인도문자	인도문자												
22	비밀보유	인도문자	인도문자	Req_sam_1	인도문자	인도문자	인도문자	인도문자	인도문자												

그림 7. K은행 품질관리시스템 구축 프로젝트사례
Fig. 7. K Bank Quality Management System Project Case

테스트단계 요구사항 추적표											
번호	어플리케이션 (단위시스템)	생성단계	요구사항ID	KB 요구사항명	분석단계	기능구분	시스템화 여부	설계단계	구현단계		
		분석			상세요구사항ID	상세요구사항명		화면ID	보고서ID	프로그램	
524	금융상품 - 기존 시스템 변경	분석	REQ_10_001	15.2.유가증권 및 파생상품 관련 공시	REQ_10_001_01	정형화된 거래의 회	비가능	N/A			N/A
525		분석			REQ_10_001_02	공정가치측정의 지	가능	N	N/A		N/A
526		분석			REQ_10_001_03	금융상품의 분류 변	가능	N	N/A		N/A
527							가능				SE150800.scr
528							가능				SE101300.scr
529							가능				SE191300.scr
530							가능				SE191800.scr
531							가능		SE191800.rep		SE191800
532							가능				SE191000.scr
533							가능				SE191400.scr
534						가능				SE190300.scr	
535						가능				SE100800.scr	
536						가능	Y			SE100900.scr	
537						가능				SE101000.scr	
538						가능				SE110200.scr	
539						가능				SE110300.scr	
540						가능				SE110400.scr	
541						가능				SE111600.scr	
542						가능				SE111910.scr	
543						가능				SE119900.scr	
544						가능				SE160300.scr	
545						가능				SE160400.scr	
545		분석	REQ_10_003	11.3.지분증권 손상 자산	REQ_10_003_01	금융상품의 손상	가능	N	N/A		N/A

그림 8. K증권 IFRS 시스템 구축 프로젝트 사례
Fig. 8. K Securities IFRS System Project Case

2. 추적표의 효과검증

본 연구의 추적표가 비효율적인 요구사항관리로 인한 프로젝트의 실패를 줄이고 고객만족 및 품질향상에 효과가 있는지에 대한 연구결과를 도출하기 위하여 수행 환경이 다른 두 회

사에 실제 적용한 6개 프로젝트를 대상으로 하였다.

적용 프로젝트의 효과는 각 프로젝트의 단계별 완료보고서에 보고된 요구사항관련 측정자료 및 종료단계의 교훈자료(Lessonned learn)를 분석하여 정리하였으며 고객 평가전인 프로젝트는 본 연구 시점까지의 결과를 반영 하였다.

표 4. 추적표 적용 프로젝트
Table 4. The Requirements Traceability Table apply Projects

PJT No	프로젝트명	관리방법론	개발방법론	기간	상태	투입인력
1	통합품질관리시스템 구축	K은행 관리방법론	K사 정보공학	10개월	종료	460MM
2	경영정보시스템 구축	K은행 관리방법론	K은행 객체지향	10개월	종료	1,700MM
3	IFRS 구축	K증권 관리방법론	L사 정보공학	8개월	종료	130MM
4	HR시스템 구축	K증권 관리방법론	W사 패키지	4개월	종료	30MM
5	은행연계시스템 구축	K증권 관리방법론	K증권 정보공학	8개월	종료	40MM
6	카드시스템 분리구축	K은행 관리방법론	K사 정보공학	6.5개월	종료	1,424MM

표본 프로젝트는 정보시스템 구축 수행사인 K사에서 최근 4년간 수행한 프로젝트 중에서 본 연구자가 참여하여 추적표 관리에 대하여 가이드를 실시하고 관리한 프로젝트를 대상으로 하였으며 [표 4]와 같다.

본 연구의 선행 연구에서 제시한 프로젝트 실패율과 주요 실패 요인인 요구사항관리 미흡에 대하여 개선된 것을 확인한다. 프로젝트의 성공이라 함은, 고객의 기대에 잘 부응하는 것으로 정의 되었었고, 고객의 범위는 내·외부를 가리지 않았다. 더 나아가 프로젝트 성공은 주어진 시간, 비용, 그리고 품질에 딱 맞추어 수행하는 것을 의미하기도 하였다. 시간, 비용, 품질간의 상호 조절 없이 '성공'이라는 점에 딱 맞게 다른 프로젝트는 거의 없다[9]. 따라서 본 연구에서는 [표 6]과 같이 프로젝트 성공에 대한 검증으로 요구사항 변경율을 측정하여 본 연구에서 제시한 요구사항추적표의 적용효과와 프로젝트의 수행평가를 위하여 일정, 품질, 고객만족점수 항목을 두고 1에서 100까지의 값으로 측정하여 100에 가까울수록 성공한 프로젝트로 가정하였다.

비용, 일정은 계약서에 기록된 계약금액 및 프로젝트일정을 기준으로 삼았으며 품질은 각 단계별 산출물 및 품질검토서의 검토결과 항목을 기준으로 하였다.

Karl은 요구사항 변경이 분석단계 이후 활동에서 발생할수록 변경에 따른 노력이 많이 소요되며 특히 테스트단계 이후부터 발생시 문제해결노력이 많이 든다고 [표 5]와 같이 설명하고 있다[9].

표 5. 단계별 재작업 정도
Table 5. The rework rate for phase

단계	문제해결 노력	단계	문제해결 노력
분석	1	테스트	15~40
설계	3~6	검수	30~70
구현	10	운영	40~100

요구사항 변경 건은 SDLC 각 단계말에 요구사항추적표의 생성단계 및 생성구분 항목을 가지고 구하며 생성구분 항목에 추가, 삭제, 변경으로 기록된 항목의 합계가 변경건수(C) 이고 해당 변경건의 변경단계는 생성단계를 가지고 파악 하여 프로젝트 성공에 기인하는지 여부를 알아본다. 각 단계별 요구사항 변경율은 각 단계별 기준 요구사항건수(R) 즉, 직전 단계에 확정된 요구사항 건수를 기준으로 해당 단계 진행중 변경된 요구사항건수를 측정하며 구하며 산식은 아래와 같다.

$$\text{요구사항 변경율} = \frac{\sum (Ci, Cn)}{\sum (Rn)} \times 100$$

프로젝트수행결과인 [표 6]을 보면, 비용 및 일정은 6개 대상 모두 계약내용 범위에서 프로젝트를 진행하였고 품질 또한 모두 '적정' 평가를 받았다. 고객만족 점수는 평가전인 프로젝트를 제외한 5개 프로젝트에서 평균95.8점의 점수를 받아 본 연구에서 제안한 추적표 사용 프로젝트에서는 모두 성공한 프로젝트임을 확인할 수 있다. 특히, 요구사항 변경율을 보면 요구사항을 요청한 사용자가 각 단계별로 참여 하여 확인함으로써 문제해결 노력이 많이 드는 테스트단계에 변경이 없어 요구사항관리 미흡으로 인한 프로젝트 실패요인을 감소시키는 효과가 있고, 고객 만족 및 프로젝트 성공에 영향을 주었음을 알 수 가 있으며 전체적인 요구사항 변경에 대한 노력이 적게 들었다.

표 6. 추적표 적용 효과 검증
Table 6. The Requirements Traceability Table verification results

PJT No	프로젝트 성공 검증				요구사항 변경율			
	비용	일정	품질	고객만족	분석	설계	구현	테스트
1	100%	100%	적정	97점	0%	5.3%	0%	0%
2	100%	100%	적정	97점	11.8%	1.2%	4.7%	0%
3	100%	100%	적정	94점	36.1%	4.3%	10%	0%
4	100%	100%	적정	96점	3%	0%	10%	0%
5	100%	100%	적정	96점	0%	0%	5%	0%
6	100%	100%	적정	평가전	7%	0%	0%	0%
평균				95.8점	9.7%	1.8%	4.9%	0%

추적표의 이행단계에 대한 것은 사용자 인수테스트 결과로 대체함으로써 추적관리의 효율성을 높일 수 있었으며 추적표의 연결 및 추적성에도 문제가 없음을 확인 하였다. 또한 추가적인 항목관리를 통해 단계별 요구사항 변경을 쉽게 파악 하고 관리 할 수 가 있다는 것을 확인 하였다.

V. 결론

본 연구의 요구사항추적표가 요구사항의 발생단계인 Pre-RS부터 추적하여 쉽게 이해하고 작성할 수 있으며 사용이 용이하다는 점을 볼 때 본 연구에서 제시한 요구사항 추적표를 활용하는 것이 산출물간의 요구사항을 추적하는데 효과적인 방안이 될 것이다.

요구사항 도출의 사전 단계인 Pre-RS부터 사후 요구사항인 Post-RS까지의 추적성 및 연결성을 유지하면서 추적관리 활동에 효율적인 관리 모델을 제시함으로써 Pre-RS이후부터 일부 단계에만 적용하거나 또는 Post-RS의 마지막까지 추적

관리하는 기존 연구에 비해 효율성을 향상시킨다.

특히, 각 단계말에 요구사항 진행 검토를 위해 사용자가 참여함으로써 테스트단계 이후의 요구사항 변경발생을 최소화하여 프로젝트 막바지에 발생할 수 있는 위험을 줄임으로써 프로젝트의 실패요인을 감소시킨다.

향후 연구과제로는 본 연구가 연구자가 참여한 2개 회사의 프로젝트에 대해서만 검증을 함으로써 자료조사의 한계성이 있었다. 따라서 더욱 객관적인 효과를 나타내기 위하여 본 연구에서 제안하는 방법을 2개 이외의 타사 또는 더 많은 서로 다른 고객 프로젝트에 적용하여 효과를 측정하고 분석하는 것이 필요할 것이다.

또한 본 연구 결과를 토대로 프로젝트에 참여하는 프로젝트관리자, 품질관리자 및 감리자, PMO, 사용자 등 프로젝트 이해관계자간에 요구사항추적관리에 대한 의사소통 기준으로 활용 할 수 있을 것이다.

앞으로 본 연구에서 제시한 요구사항추적표의 수작업에 따른 불편함을 감소시키고 복잡하지 않으면서도 사용이 용이한 지원도구를 개발하여 사용한다면 추적표의 활용성 및 사용효과를 높여 보다 추적관리가 쉬워질 것이다.

참고문헌

- [1] C.S. Jeong, S.R. Kim, N.G. Kim, "Design and Application of PMO-based Project Management Systems", *Journal of Information System Research*, Vol. 20, No. 4, pp.119-143, Nov. 2011.
- [2] J.C. Shin, "Improving Requirements Specification to extend Requirements Management over the Development Life Cycle", *Journal of The Korea Society of Computer and Information*, Vol.6, No.4, pp.30-37, 2001.
- [3] Standish Group, "CHAOS Reports", Apr. 2009.
- [4] C.S. Jeong, K.H. Shin, W.C. Han "Communication Network Analysis of General Corporate IT Staff : A Case Study of Company "K"", *Journal of The Korea Society of Computer and Information*, Vol.16. No.11., pp.219-232, Dec. 2011.
- [5] O. Gotel, and A. Finkelstein, "An Analysis of the Requirements Traceability Problem, Proceedings of the First International Conference on Requirements Engineering", Colorado Springs, Colo., pp.94-101, 1994.
- [6] H.D. Rombach, "Software Specification: A Framework. Curriculum Module" SEI-CM-11-2.1, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa., 1990.
- [7] S.Y. Park, M.S. Hong, "Requirement Management Systematic Approach Method", *IT TREND*, pp.49, Jan.-Feb. 2007.
- [8] A.M. Davis, et.al, "A Strategy for Comparing Alternative Software Development Life Cycle Models", *IEEE Trans, Software Eng.* pp.1453-1461, Oct. 1988.
- [9] Karl E. Wiegers, "Software Requirements", Microsoft Press, 1999.
- [10] J.E. Choi, S.K. Choi, S.A. Lee, "The Case Study of Software Requirement Management", *The Korean Institute of Information Scientists and Engineers, Spring Conference*, Vol.29. No.1., pp.445-447, Apr. 2002.
- [11] J.K. Lee, H.K. Cho, I.Y. Ko, "A Method for Requirements Traceability for Reuse of Artifacts using Requirements-Ontology-based Semantic Tagging", *The Korean Institute of Information Scientists and Engineers*, Vol.35. No.6., pp.358, 2008.
- [12] Lee M. Williams, *The Elements of Technical Requirements Management*, INCOSE Atlanta Chapter, Feb. 2006.
- [13] J.Y. Kim, "Requirement Tracing Method Using CBD-based Traceability Table", *Doctoral Thesis of Soongsil University*, 2009.
- [14] Grant Zemnt, "Recovering Towards Value Based Requirements Traceability", *DePaul University Chicago Illinois*, Mar. 2005.
- [15] Y.B. Lee, H.J. Lee, M.S. Hwang, *A Study for Design and Development of a Standard Guidance for Software Requirement Specification*, Korea

- Conference on Software Engineering, Feb. 2009.
- [16] Quality Management Dept., "Software Development Methodology", KB Data Systems Co.Ltd., 2008.
- [17] Martin Fowler, and Kendall Scott, "UML Distilled: Applyin the Standard Object Modeling Language", Addison Wesley, Inc., 1997.
- [18] Soren Lauesen, "Software Requirements Styles and Techniques", Addison Wesley, 2002.
- [19] J.Y. Eom, N.H. Kim, "A Quantitative ERRC Analysis Method via Requirements Cost Estimation", Journal of Korea Society of IT Service, Vol.8. No.1., pp.116, 2009.
- [20] Amir Ghahrai, <http://www.testingexcellence.com/v-model/>, 2008.

저 자 소 개



정 천 수

2002 : 고려대학교 컴퓨터공학 석사
현 재 : 국민대학교 비즈니스 IT전문
대학원(박사과정)
관심분야 : Mega Project
Management & PMO,
Project Communication,
IT Governance
Email : paripal@kookmin.ac.kr



김 승 렬

1976 : 서울대학교 산업공학 석사
1983 : 미국 Iowa State University
산업공학과 공학박사
현 재 : 국민대학교 경영정보학부 교수
관심분야 : 소프트웨어 엔지니어링, 프
로젝트 관리, 시스템 분석
설계
Email : srkim@kookmin.ac.kr