

추적테이블을 이용한 요구사항 변경관리 및 추적 효과 연구

김 주 영[†] · 류 성 열^{**} · 황 만 수^{***}

요 약

소프트웨어 프로젝트의 실패요인 중 54%가 요구사항관리 미흡에서 발생하며, 이중 22%는 요구사항 변경관리에서 기인한다. 따라서 실패요인을 줄이기 위해 요구사항 관리활동이 중요하며 이중 요구사항 변경을 위한 핵심활동으로 추적기법을 활용한다.

추적기법에 사용되는 방법인 추적테이블은 단순링크 방식으로 사용이 간편하고 가독성이 있으며 추적의 정확성이 높은 장점이 있다. 그러나 기존 추적테이블 연구는 변경관리 방법 및 추적의 효과를 구체적으로 제시하고 있지 않으며, 또한 변경영향 추정연구는 추정방법이 복잡하여 실용성에 한계가 있다.

이에 본 연구는 기존에 연구된 추적테이블을 이용하여 변경요구사항을 관리하는 방법을 제시하고 이로 인해 변경율 및 변경영향도를 기존 연구에 비해 용이하게 추정할 수 있는 기법을 제안한다. 또한 15개 프로젝트를 대상으로 추적테이블의 효과를 가설 검증하여 추적테이블이 프로젝트 성공에 영향을 미치며, 요구사항 관리미흡으로 인한 실패요인을 감소시킴을 확인한다.

키워드: 요구사항, 요구사항추적, 추적테이블, 변경관리

A Study of Requirement Change Management and Traceability Effect Using Traceability Table

Kim Ju-Young[†] · Rhew Sung-Yul^{**} · Hwang Man-Su^{***}

ABSTRACT

Insufficient requirement management accounts for 54% of unsuccessful software development projects and 22% of insufficient requirement management comes from requirement change management. Hence, requirement management activities are important to reduce failure rates and a tracing method is suggested as the major factor in requirements change management.

A traceability table is easy to use because of its legibility accurate tracing. However, traceability tables of existing studies have failed to concretely suggest method of change management and effect of traceability. Also, studies of methods to estimate change impact is complex.

Hence, this study suggests how to use a traceability table to manage changes in requirements. Together, in comparison to existing studies, this study suggests easier methods to estimate change rate and change impact. Also Fifteen projects were sampled to test the hypothesis that traceability table influences the success of projects and that it decreases the failure rate that comes from the insufficient requirements management.

Keywords: Requirement, Requirement Traceability, Trace Table, Requirement Change Management

1. 서 론

Stanish가 1994년, 2004년에 13,522개 IT기업의 40,000개 프로젝트 수행결과를 조사한 보고서를 보면 1994년에는 일정 지연, 비용초과, 범위누락, 프로젝트 취소 등의 이유로 실패

패한 프로젝트는 전체의 84%이다[1]. 실패요인을 분석한 결과 불완전한 요구사항, 요구사항의 변경관리 부족 등의 요구사항관리 미흡으로 인한 이유가 전체 실패이유 중 61%를 차지한다[1]. 한편 10년 후인 2004년의 결과를 보면 프로젝트 실패율은 전체의 66%로 1994년에 비해 현저하게 줄어든 반면, 실패 프로젝트들의 실패요인을 보면 요구사항관리 미흡으로 인한 실패이유가 1994년 보다 겨우 7% 감소하여 전체 실패요인의 54%를 차지한다[2].

특히 실패요인의 22%는 요구사항 변경으로 프로젝트 성공여부에 큰 영향을 미침에도 불구하고 여전히 변경관리 미흡

* 본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음(NIPA-2010-(C1090-0903-0004)).

† 정 회 원: 대림산업 정보시스템실 차장

** 종신회원: 숭실대학교 컴퓨터학부 교수

*** 정 회 원: 신홍대학 컴퓨터정보계열 교수

논문접수: 2010년 2월 9일

수정일: 1차 2010년 3월 29일

심사완료: 2010년 4월 2일

으로 인해 프로젝트가 잘 이루어지고 있지 않음을 시사한다.

이에 요구사항을 관리하기 위한 활동을 구체적으로 나열한 CMMI은 레벨3에서 추적성을 강조한 요구사항 검증(Verification) 및 확인(Validation)을 특정 목적 및 특정 프랙티스에 추가하고 있다[3]. 이는 요구사항 관리 활동 중 추적성의 중요성을 강조한 것인데, 정유신[4]은 CMMI 기반의 요구사항 추적성 관리 프로세스를 준수하였을 때 소프트웨어 결함이 감소한 효과를 보여주고 있다.

요구사항 추적은 V&V 모델(Verification & Validation Model)에 기반한 것으로 개발프로세스 생명주기 동안 진화하는 요구사항에 대해 각 단계별로 요구사항의 일관성을 검증하고 단위테스트, 통합테스트, 시스템테스트, 사용자테스트 등을 거쳐 요구사항의 완전성을 확인하는 것이다[5]. 따라서 요구사항을 추적하는 방법 및 기법에 대한 기존 연구가 많은데, 기존 추적연구는 Ramesh[6]의 추적모델처럼 구체적인 추적대상물을 정의하지 않거나 정보검색(Information Retrieval) 기법[7], 이벤트기반(Event-based) 기법[8], 시나리오기반(Scenario-based) 기법[9] 등과 같이 추적모형을 구현하고 유지보수하는데 비용이 많이 든다. 또한 추적의 전체인 전체 개발프로세스를 대상으로 하지 않으며, 대부분의 추적연구가 사용자의 합의된 기대를 유도하는 사전요구사항(Pre-requirement)을 추적하지 않는 단점이 있다.

이에 반해 단순링크 방식의 추적테이블을 개선한 김주영의 연구[10]는 추적의 정확성 및 가독성을 높이고 사용에 용이하다는 장점을 이용한 속성기반의 확장 추적테이블을 제시하여 추적의 효과를 높이고 있다. 그러나 김주영의 연구는 Stanrish 보고서의 실패요인 중 가장 많은 부분을 차지하는 요구사항 변경에 대한 관리방법 제시가 미흡하며, 추적테이블 사용 효과에 대한 제시가 없다.

이에 본 연구에서는 추적테이블을 이용한 변경관리 방법을 제시하는 한편, 추적테이블을

통해 요구사항관리 미흡으로 인한 프로젝트의 실패요인을 감소시키기는 효과를 제시하고자 한다.

연구의 범위 및 방법은 선행연구인 김주영의 추적테이블을 이용해 요구사항 변경시 변경요구사항을 추적테이블에 기재하여 관리하는 방법을 제시하고, 제시된 방법이 변경시 프로젝트에 미치는 성능 및 결함을 설명하는 변경을 및 변경영향도를 추정하는 과정에서 기존 연구에 비해 용이함을 설명한다. 또한 15개의 실제 프로젝트를 대상으로 설문 실시하고, 추적테이블이 프로젝트 실패요인을 줄인다는 가설을 설정, 가설검정기법을 이용하여 추적테이블 사용효과를 검증한다.

본 연구는 총 5장으로 구성되어 있다. 1장은 연구의 배경 및 범위를 설명한 서론이며 2장은 본 연구를 위한 선행연구 및 기존연구를 기술한다. 3장과 4장, 5장은 본론으로, 3장은 추적테이블을 이용한 변경관리 방법, 4장은 사례 적용, 5장은 추적테이블의 사용효과 검증을 제시한다. 마지막 6장은 결론 및 향후 연구과제를 제시하며, 연구내용에 쓰인 설문자료를 부록으로 첨부한다.

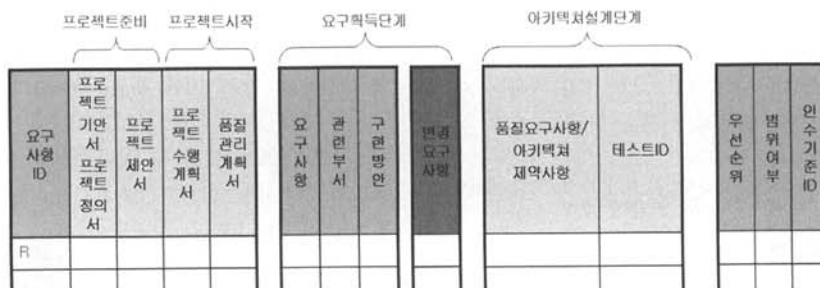
2. 선행 및 관련연구

2.1 추적테이블에 대한 선행연구

본 연구의 선행 연구인 김주영의 요구사항 추적테이블의 확장 및 정규화 방안 연구[10]에서 마르미방법론III-v4.0 전체 공정 산출물을 대상으로한 요구사항추적테이블의 개선연구가 (그림 1)과 (그림 2)와 같이 진행되었다.



(그림 1) 기능요구사항추적테이블



(그림 2) 비기능요구사항추적테이블

이 연구는 추적필드 추출과정과 추출된 추적필드로 구성된 추적테이블을 기능 및 비기능요구사항으로 나눠서 제시하고 추적테이블을 단계별로 분리하거나 합칠 수 있는 정규화 방안을 제시한다. 그러나 추적테이블의 중요한 기능 중 하나인 변경관리 방법에 대해서는 제시가 미흡하고 한계가 있다. 이에 본 연구는 추적테이블을 이용한 요구사항 변경관리 방안을 구체적으로 제시한다.

2.2 변경영향도 측정에 대한 관련연구

소프트웨어 프로젝트 성공에 큰 영향을 주는 요구사항 변경관리는 다양한 관점에서 연구되고 있다[11]. 특히 변경 요구사항 영향 연구들[12, 13]은 요구사항 변경으로 인한 영향을 분석하여 프로젝트에 미치는 성능과 결함에 대해 설명한다. 이는 요구사항 변경이 프로젝트 성과에 직접적인 영향이 있으므로 변경이 발생했을 경우 관련 요구사항 및 작업에 미치는 영향을 추정하여 프로젝트 위험이 발생할 확률을 줄이게 한다.

본 연구는 변경으로 인한 실패요인이 21% 차지하는 만큼 변경영향을 추정하는 연구를 상세히 설명하고 본 연구의 기회를 도출한다.

먼저 Luigi의 Enhancing Requirement and Change Management through Process Modeling and Measurement 연구[14]는 요구사항 추적의 중요성을 언급하면서 요구사항 변경시 요구사항 추적에 근거하여 변경영향도를 측정하는 모형을 제시하고 있다. Luigi의 변경영향도 측정모형은 다음과 같다.

- $O = f(I, R, A)$
- O:output, I:input, R:resource, A:activity

I는 변경작업에 들어가는 입력물이며, R은 변경작업에 소요되는 자원, A는 작업활동으로써 I, R, A의 함수로 변경영향도(O)를 추정한다. 특히, 변경영향도력을 비용으로 추정하는 모형을 함께 제시하는데 다음과 같다.

- $C = f(I, R, A, O)$
- C:Cost, I:input, R:resource, A:Activity, O:output

비용함수는 변경영향도에서 추정된 결과값을 더해서 추정된다.

Luigi 연구는 변경영향을 추정하는 개념모델을 제시하여 유사 연구에 많은 참조를 주며, 본 연구도 Luigi 연구를 참조하여 추적테이블에서 변경영향도를 보다 용이하게 추정하는 방법을 제시한다.

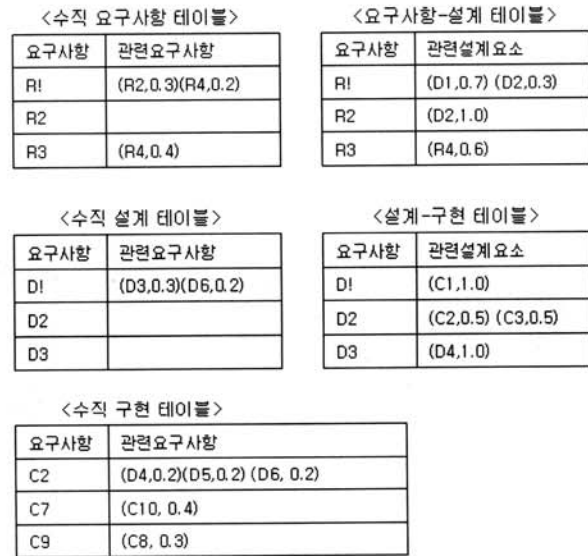
한편, 개념모델만을 제시한 Luigi 연구와는 달리 변경영향도를 추정하는 함수와 함께 구체적 방법을 제시한 정규장의 개발방법론의 요구사항 변경관리를 개선하기 위한 프로세스 모델 연구[15]는 변경영향도를 구하는 모형을 다음과 같이 제안한다.

- $R = f(d, s)$
- d: 제작업 거리, s: 영향도 범위

d는 요구사항 변경이 발생한 단계에서 요구사항이 최초 명세된 단계까지의 거리이다. 즉 변경으로 인해 제작업이 필요한 단계들을 의미하며, 요구사항이 명세된 시점에서 많은 단계가 지날수록 변경영향도는 커지게 된다. s는 변경으로 인해 영향을 받는 작업의 범위로서 요구사항을 추적하여 같이 변경되어야 하는 요소와 변경 요구사항과 관련된 요구사항들의 범위이다. 정규장 연구는 변경영향도 모형에 따라 변경영향도를 추정하기 위해 (그림 3)과 같이 요구사항과 관련 요구사항에 대한 수직-요구사항테이블을 생성한 다음, 이 결과를 이용하여 다시 요구사항-설계 테이블을 생성한다. 동일한 방법으로 변경이 일어난 단계까지 수직-수평상 관계테이블을 반복적으로 생성하여 영향도 범위를 구하고, 이를 토대로 변경영향도를 구할 수 있다.

정규장 연구는 변경영향도 모형에 따라 변경영향도를 구하는 구체적 방법을 위와 같이 제시하고 있으나, 변경이 발생할 때마다 수직-수평상관계테이블을 반복해서 생성해야 하는 불편함이 있으므로 연구에서도 실용성에 한계가 있음을 설명하고 있다.

이에 본 연구는 변경영향도 추정 모형을 참조하되 추적테이블을 이용하여 변경영향도를 보다 용이하게 추정하는 방법을 제시한다.



(그림 3) 정규장의 수직-수평상관계 테이블

3. 추적테이블을 통한 요구사항 변경관리

3.1 추적테이블의 요구사항 변경관리 규칙

요구사항 변경 발생시 변경된 요구사항을 추적하고 변경으로 인한 추적요소를 파악하여 변경으로 인한 영향도를 추정할 수 있어야 한다[13, 14]. 변경이 발생하지 않는다면 순방향 추적에 따라 요구사항의 일관성 및 완전성을 검증하기

용이하나 변경이 발생할 경우 추적관리는 역방향 추적과 함께 관련 요구사항과 그 추적요소까지 추적할 수 있어야 한다.

선행연구인 2.1 김주영의 추적테이블에서 변경 요구사항을 (그림 4)와 같이 변경요구사항 필드 하나에 변경이력을 한꺼번에 열거하여 관리한다면 다음 단계의 추적시 변경 요구사항에 대한 추적요소를 일렬로 한눈에 확인하기 어려워져 가독성이 떨어진다. 이는 추적테이블이 단순하고 가독성이 있다는 장점에 대립하는 것이다. 따라서 추적테이블을 장점을 그대로 이용한 요구사항 변경을 관리하는 규칙을 본장에서 제시한다.

요구사항이 지속적으로 변경하는 경우와 변경된 요구사항이 다시 기존의 요구사항으로 변경되는 경우를 나눠 설명한다. 먼저 지속적으로 요구사항이 변경되는 경우는 (그림 5)와 같다.

최초 정의된 요구사항ID가 R1이라고 가정하면 1차로 변

경된 요구사항은 R1-1로 식별하고 새로운 요구사항 행으로 추가한다. 다음 단계 추적노드 또한 R1-1에 해당하는 추적요소를 기재하는데 기존 요구사항R1과 동일할 경우 변경없이 같은 내용을 기재한다.

R1-1에 대해 다시 변경이 발생할 경우 요구사항ID는 R1-1-1로 식별한다. 즉, 요구사항 변경시 최초 요구사항ID를 참조하여 하이픈(-)으로 계속 이음으로써 마치 지수함수에서 지수가 올라가는 규칙을 따른다. 이렇게 변경요구사항을 (그림 4)처럼 별도 요구사항ID 없이 변경요구사항필드에 내용만 열거하는 것과는 달리 또 하나의 ID를 부여하되, 변경의 원천 요구사항을 알 수 있도록 기재함으로써, 변경요구사항의 근원을 인지하여 변경영향 추정을 용이하게 하고, 추적테이블에서 요구사항을 축으로 변경으로 인한 추적요소 또한 추적테이블의 규칙대로 일렬로 추적하는게 가능하다.

두 번째, 변경된 요구사항이 다시 원래의 요구사항으로

기능 요구사항 ID	요구사항 내용	변경요구사항 ID	관련요구사항 ID	버전	관련부서	구분방안	유스 케이스	클래스	화면 ID / 보고서 ID	외부인터페이스 ID	이력테저요구사항	가변성	공용성	임베디드	변태어지 ID	제사용 / 획득여부	엔터티	테스트 ID	단위테스트 ID / 결과	통합테스트 ID / 결과	인수테스트 ID / 결과	변경관리 ID	우선순위	범위여부	인수기준 ID
R1	A	B	X	1.0			uc 12	cll 46	ui 10						cp 55	wp 32		te 43				ch 2			ac 43
R2		C	X	0			uc 12																		
		D	Z				uc 13																		

(그림 4) 변경발생시 추적테이블의 문제점

기능 요구사항 ID	요구사항 내용	변경요구사항 ID	관련요구사항 ID	버전	유스 케이스명	...	변경관리ID	범위여부	인수기준ID
R1	A	B	X	1.0	UC101_급여제도관리	...		OUT	AC101_급여제도
R1-1					UC101_급여제도관리	...	CH201	OUT	AC101_급여제도
R1-1-1 (R ^{x3})					UC101_급여제도관리 UC301_직무급관리	...		IN	AC101_급여제도 AC301_직무급

변경관리 현황	변경관리ID	변경제목	...	요구사항ID
		CH201	급여제도를 직무급으로 변경	...

(그림 5) 지속적 변경발생시 추적테이블 변경관리 규칙

되돌아가는 경우인데 (그림 4)에서는 기재가 애매하거나 일렬 추적이 어려운 경우이다. 이에 본 연구는 지속적으로 변경되는 경우와 같이 요구사항ID를 식별하는 방법은 동일하되 특히 원래의 요구사항으로 원복되었다 하더라도 새로 변경된 요구사항으로 추가하도록 한다. (그림 6)과 같이 요구사항ID R2가 R2-1로 변경된 후 다시 R2의 요구사항내용으로 원복되었다면 마지막 요구사항ID는 R2-1-1로 한 행을 추가하고 R2의 추적요소를 그대로 또는 변경된 내용으로 기재한다.

추적테이블에서 변경 요구사항을 작성하는 규칙을 정리하면 다음과 같다.

- ① 변경전 요구사항의 ID에 RX-1로 기재한다.
 - 변경이 계속될 경우 RX-1, RX-1-1,.. 로 표시한다.
 - 변경요구사항에 대한 추적요소를 변경전 요구사항을 참조하여 기재한다.
- ② 변경요구사항이 다시 변경 이전의 요구사항으로 변경될 경우 추가 변경요구사항ID로 기재한다.
 - 추적요소는 변경 이전의 요구사항의 추적요소를 참조한다.
- ③ 변경요구사항이 발생하면 범위여부, 인수기준ID를 변경한다.
 - 변경전 요구사항에 대해 범위여부 [IN, OUT]를 정하고, 변경요구사항은 범위여부 [IN]을 기술한다.
 - 변경전 요구사항에 대해 인수기준ID를 확인/변경하고, 변경요구사항은 인수기준ID를 기재한다.
- ④ 변경관리현황에 요구사항 매핑 위한 요구사항ID 기재시 변경 이전 요구사항을 포함한다.
 - 변경 원인 및 영향에 대한 알림을 정확하게 한다.

3.2 요구사항 변경영향 추정

3.2.1 요구사항 변경을 산정

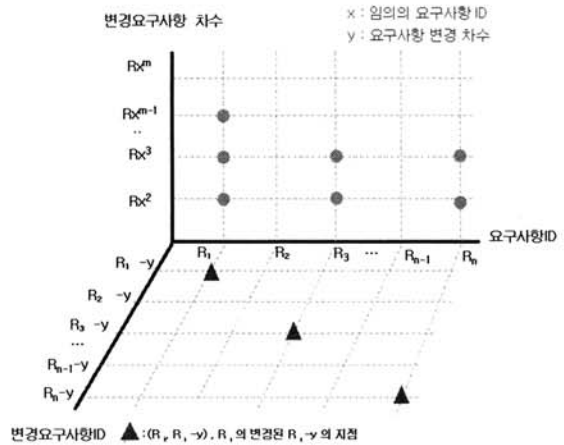
요구사항 변경시 프로젝트 위험을 측정하고 계획과 대비

한 실적을 예측하기 위해 요구사항 변경율을 산정한다[3]. 요구사항 변경이 많을수록 프로젝트는 계획에 차질이 생기고 위험도가 높아져 프로젝트의 실패요인이 되기 때문이다[2].

이에 CMMI[3, 4]에서는 요구사항 변경율을 <표 1>과 같이 정의하고 있다.

<표 1>의 산식대로 요구사항 변경율을 구하기 위해서는 통상적으로 매번 모든 요구사항 수를 계산하고 변경된 요구사항의 수를 확인하여야 한다. 반면 추적테이블에서 요구사항 변경율을 산정하는 방법은 (그림 7)과 같이 일일이 건수를 세지 않아도 전체 테이블에서 요구사항 수와 변경이력을 가진 요구사항 수를 즉시 계산할 수 있다. (그림 7)에서 가로 X축은 기준선 시점의 요구사항ID인 Rn을 의미하며, 세로 Y축은 X축의 요구사항ID별로 변경된 차수들로 Rxⁿ을 의미한다. 나머지 Z축은 X축과 Y축의 관계에서 한번이라도 변경으로 인해 좌표가 표시된 요구사항ID를 나타낸다. 즉, Z축이 나타내는 (Ri, Ri-y)의 건수가 기준선 시점의 요구사항에 대해 한번이라도 변경이 일어난 건수가 되게 된다.

따라서 추적테이블에서 요구사항 변경율을 구하는 산식을



(그림 7) 추적테이블에서 변경을 산정

요구사항ID	요구사항 내용	변경 요구사항	관련요구사항ID	버전	유스케이스명	...	변경관리ID	범위여부	인수기준ID
R2	일반직 업적평가감	일반직+전문직 업적평가감 (0000년 0월0일)	R11	1.0	UC401.업적평가관리	...	CH401	OUT	AC401.업적평가관리
R2-1 (Rx ²)	일반직+전문직 업적평가감	일반직만 업적평가감 (0000년 0월0일)	R11, R12	2.0	UC401.(일반직)업적평가관리 UC411.전문직업적평가관리	...		OUT	AC401... AC411...
R2-1-1 (Rx ³)	일반직만 업적평가감		R11	3.0	UC401.업적평가관리	...		IN	AC401.업적평가관리

변경관리 현황	변경관리ID	변경제목	...	요구사항ID
	CH401	업적평가에 전문직 추가	...	R2, R2-1

(그림 6) 이전 요구사항으로 변경시 추적테이블 변경관리 규칙

<표 1> 요구사항 변경율 측정

측정항목	측정내용(산식)	측정주기	측정방법	측정담당자
요구사항 변경율	(Σ요구사항변경건수 / Σ요구사항건수) X 100	월간	요구사항 베이스라인 설정 후 변경된 개수를 측정함	요구사항 관리담당자

정의하면 다음과 같다.

$$\bullet \text{ 요구사항 변경율} = \frac{\sum(R_i, R_i - y)}{\sum(R_n)}$$

3.2.2 요구사항 변경영향도 추정

요구사항 변경율과 함께 요구사항 변경에 대한 영향 정도를 추정하는 것 또한 현재 프로젝트 현황을 파악하고 위험을 대비함으로써 프로젝트 실패를 줄이는 활동이다. 이에 2.2에서 본 바와 같이 요구사항 변경영향도를 추정하는 모형을 인용하여 추적테이블에서 변경영향도를 추정하는 방안을 제시한다.

변경영향도를 추정하는 모형은 [14, 15]연구처럼 다음과 같이 정의한다.

- $R = f(d, s)$
- R : 소요자원(인력, 일정, 비용)
- d : 재작업 거리, s : 영향을 받는 요구사항의 범위

변경영향도 R은 변경을 처리하는데 소요되는 자원을 의미하며 인력과 일정과 비용으로 환산된다. 재작업 거리 d는 변경이 일어난 현재시점의 개발단계에서 요구사항의 기준선이 정의된 개발단계까지의 거리로써, 거리가 멀수록 재작업해야 하는 활동이 많으므로 변경영향도는 커진다[16].

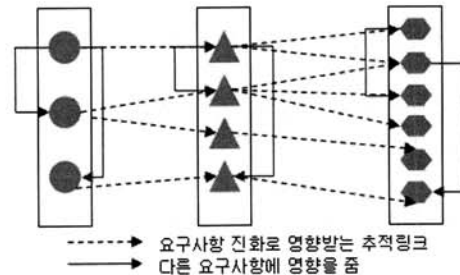
이에 Karl[16]은 요구사항 변경이 요구분석 이후 후행단계의 활동에서 발생할수록 변경에 따른 문제해결 노력이 많이 소요된다고 설명하고 이 노력을 정량적으로 <표 2>와 같이 정의한다.

또한, 정규장 연구[15]에서 범위 s는 변경으로 인해 영향을 받는 변경범위로서 관련요구사항에 대한 범위와 변경요구사항의 추적요소에 대한 범위를 합한 값이다. 이를 그림으로 표현하면 (그림 8)과 같다[15].

즉, 변경범위를 산정할 때는 점선으로 표시한 변경 요구사항의 추적요소 뿐만 아니라 실선으로 표시된 관련 요구사항과 그에 따른 추적요소까지 함께 포함해야 한다. 범위값은 0.0에서 1.0사이의 값으로, 하나의 요구사항이 다른 요구

<표 2> Karl의 변경시 단계별 재작업 정도

단계	문제해결노력	단계	문제해결노력
요구분석	1	테스트	15-40
설계	3-6	검수	30-70
구현	10	운영	40-100



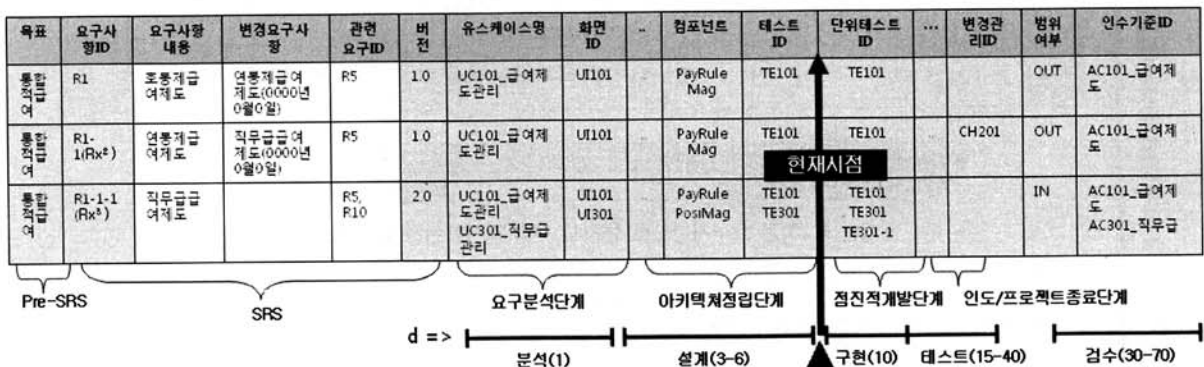
(그림 8) 변경범위 추정 모형

사항에 의하여 유도된 것이거나 완전히 종속적인 경우에는 최고의 가중치(1.0)를 부여한다.

정규장 연구는 변경범위를 (그림 8)의 추적 모형에 의거 산정한다. 구체적인 산정 위해 2.2의 (그림 3)과 같이 변경시마다 수직 및 수평 상관관계를 넓이-깊이 우선검색 알고리즘을 통해 테이블로 생성해야 하는데 이는 노력이 많이 들어 실용성에 한계가 있다[15]. 또한 변경관리는 빈번한 변경으로 인한 신속한 대처가 필요한 활동인데 이처럼 노력이 많이 소요되는 관리방안은 변경관리 성능을 저하시킨다.

이에 본 연구는 추적테이블의 변경관리 방안을 이용하여 기존연구에 대비하여 용이한 변경영향도를 추정 방법을 제시한다. 재작업거리를 산정하기 위해서 (그림 9)와 같이 추적테이블에서 추적노드의 단계와 Karl[16]이 제시한 단계를 매핑한다. 요구분석단계는 분석단계로, 아키텍처정립단계 설계단계로 일대일 매핑되며, 점진적개발단계는 구현 및 테스트단계로 나눠 매핑한다. 인도/프로젝트종료단계는 검수단계로 매핑한다.

매핑결과에 따라 변경된 현재시점에서 요구분석단계까지의 거리를 산정하면 재작업 거리(d)가 되는데, (그림 9)의 예처럼 변경요구사항 발생시점이 아키텍처정립단계라면 재작업 거리는 다음과 같다.



(그림 9) 추적테이블의 재작업 거리 산정

- 제작업 거리(d) = 아키텍처 정립단계 - 요구분석단계

변경영향을 받는 범위(s)를 추정하기 위해서는 정규장 연구처럼 추적요소간의 수직, 수평상관관계를 구하는 것이 아니라, 변경요구사항과 관련요구사항의 범위를 <표 3> <표 4>과 같이 추적테이블의 추적요소를 그대로 일렬로 표현하여 영향범위의 값을 정의한다. 괄호안 수치는 (그림 8)에서 설명한 바와 같이 영향범위에 대한 가중치로, UC101은 변경요구사항으로 인한 영향범위가 0.3이며, UC301은 변경요구사항에 완전히 종속되어 영향받은 범위이므로 1.0의 값을 가진다.

추적테이블은 이미 변경요구사항 별로 추적요소를 일렬로 나열하고 있으며, 관련요구사항 또한 추적노드로 포함하고 있기 때문에 (그림 3)과 같은 별도의 알고리즘이나 다른 생성절차 없이 추적테이블만으로 변경범위를 추정할 수 있게 된다.

따라서 추적테이블은 변경 요구사항에 대한 변경영향도를 기존 연구에 비해 용이하게 추정한다.

<표 3> 변경요구사항의 추적요소

요구사항	유스케이스명	화면ID	..	컴포넌트ID	테스트ID
R1-1-1	UC101(0.3), UC301(1.0)	UH101(0.3), UI301(1.0)	PayRule(0.3), PosiMag(1.0)	TE101(0.3), TE301(0.5), TE301-1(0.5)

<표 4> 변경요구사항의 관련요구사항

요구사항	관련요구사항
R1-1-1	(R5, 0.3), (R10, 0.5)

4. 추적테이블의 변경관리 방안 적용 사례

연구의 추적테이블 변경관리 방안을 실제 프로젝트에 적용한 사례가 (그림 10)이다. 사례 프로젝트는 D사의 인사시스템으로 1)인사기획, 2)조직 및 인사기본, 3) 경력/발령, 4) 근태/복무, 5)급여보상, 6)복리후생, 7)인사평가, 8)퇴직정산, 9)연말/중도정산, 10)사회보험 등이

주요 요구사항 범위이며 3,540FP(Function Point) 규모이다.

(그림 10)에서 보는 바와 같이 선행연구의 추적테이블을 토대로 기능요구사항 추적테이블을 작성하였다. 아키텍처정립단계에서 FN306 요구사항이 변경되었고, 변경된 요구사항을 FN306-1로 정의하고 다음 행에 추가하였다. 최초 요구사항인 FN306은 테스트ID 다음단계에서 추적요소가 더 이상 없으며, FN306-1은 FN306의 추적요소가 그대로 이어지거나 새로운 추적요소가 추가되었다.

(그림 11)은 (그림 10)에서 변경요구사항을 기재한 부분을 보다 확대하여 보여준다.

요구사항ID/명	분석/설명 요구사항	시스템 구현 요구사항	변경요구사항	유스케이스	화면 메뉴	UI	인터페이스 아키텍처 요소	패턴
FN300, 급여보상 기능 요구사항	기존 급여보상 시스템은 월급 지급 시, 월급 지급액에 각종 공제(연금, 건강보험, 국민연금, 퇴직금 등)를 차감하여 지급한다. 변경된 요구사항은 퇴직금 공제에 대한 처리를 변경한다. 퇴직금 공제액이 월급 지급액보다 클 경우, 퇴직금 공제액의 초과분을 월급 지급액에서 차감한다.	퇴직금 공제액에 대한 처리를 변경한다. 퇴직금 공제액이 월급 지급액보다 클 경우, 퇴직금 공제액의 초과분을 월급 지급액에서 차감한다.	FN306-1, 퇴직금 공제액에 대한 처리를 변경한다. 퇴직금 공제액이 월급 지급액보다 클 경우, 퇴직금 공제액의 초과분을 월급 지급액에서 차감한다.	FN306, 퇴직금 공제액에 대한 처리를 변경한다. 퇴직금 공제액이 월급 지급액보다 클 경우, 퇴직금 공제액의 초과분을 월급 지급액에서 차감한다.	FN306, 퇴직금 공제액에 대한 처리를 변경한다. 퇴직금 공제액이 월급 지급액보다 클 경우, 퇴직금 공제액의 초과분을 월급 지급액에서 차감한다.	FN306, 퇴직금 공제액에 대한 처리를 변경한다. 퇴직금 공제액이 월급 지급액보다 클 경우, 퇴직금 공제액의 초과분을 월급 지급액에서 차감한다.	FN306, 퇴직금 공제액에 대한 처리를 변경한다. 퇴직금 공제액이 월급 지급액보다 클 경우, 퇴직금 공제액의 초과분을 월급 지급액에서 차감한다.	MVC

(그림 10) 추적테이블 변경관리 방안 적용 사례

요구사항ID/명	분석/설명 요구사항	시스템 구현 요구사항	변경요구사항	유스케이스	화면 메뉴	UI	인터페이스 아키텍처 요소	패턴
FN300, 급여보상 기능 요구사항	현재직 급여에 산정시 표준 데이터를 적용하여 경력과 업무별 월급을 급여 지급 기능 구현.	현재직 급여에 산정시 표준 데이터를 적용하여 경력과 업무별 월급을 급여 지급 기능 구현.	FN306, 호봉테이블 관리	UC3103, 급여할목설정 관리	급여기준정보>급여테이블	UI44	자바개발관 경인 PROWORKS 과 RSA 연동하여 설계와 코드 동기화	MVC
모든 직원 급여시스템 통합	급여시스템 통합을 위한 모든 가능한 시스템 구현	급여시스템 통합을 위한 모든 가능한 시스템 구현	FN306-1, 직무급 테이블 관리	UC3103, 급여할목설정 관리 UC3910, 직무급관리	급여기준정보>급여테이블	UI44 UI200	자바개발관 경인 PROWORKS 과 RSA 연동하여 설계와 코드 동기화	MVC

(그림 11) 변경요구사항 적용 사례부분

5. 추적테이블 효과 검증

추적테이블이 요구사항관리 미흡으로 인한 프로젝트 실패 요인을 감소시키는데 대한 효과를 검증하기 위해 15개의 표본 프로젝트 대상으로 가설검정을 한다. 이때 표본 프로젝트는 선행연구인 김주영 추적테이블에 본 연구의 변경관

리 방안을 적용한 사례로, 선행연구에 이어 변경관리를 포함한 추적테이블의 효과 검증을 수행한다.

표본 프로젝트는 D사에서 최근 5년간 수행한 내·외부 프로젝트로써 규모 20M/M 이상인 프로젝트를 대상으로 한다[17]. 표본 프로젝트는 <표 5> <표 6>이며 프로젝트 관리자를 대상으로 설문을 진행한다. 상세한 설문내용은 부록을 참조한다.

<표 5> 추적테이블 효과검증 표본프로젝트-I

표본No	기간	규모	방법론/적용기법/도구	결과 (계획대비 실적초과치)	본 연구 적용 여부	성공/실패 요인
1	6M	50 M/M	D사 개발방법론 v2.0(마르미III 유사) J2SE/RSA/Requisit pro	-비용준수 -일정지연(1M) -품질불만족	미사용	사용자 기대이상
2	7M	54 M/M	D사 개발방법론 v2.0 J2SE/RSA	-비용준수 -일정준수 -품질불만족	사용	요구사항 불충분
3	4M	20 M/M	D사 개발방법론 v2.0 J2SE/RSA/Requisit pro	-비용초과 -일정지연(1M) -품질불만족	미사용	테스트시 요구사항 변경 사용자 참여/합의부족
4	3M	20 M/M	D사 개발방법론 v2.0 J2SE/RSA	-비용초과(6M/M) -일정지연(2M) -품질불만족	미사용	잡은 요구사항 변경 사용자 참여/합의부족
5	18M	128 M/M	마르미III J2SE	-비용준수 -일정준수 -품질만족	사용	
6	9M	60M	객체방법론+정보공학J2SE	-비용준수 -일정준수 -품질만족	사용	
7	9M	54 M/M	객체방법론+정보공학 J2SE	-비용준수 -일정준수 -품질만족	사용	
8	16M	215 M/M	D사 개발방법론v2.0 J2SE/RSA/Requisit pro	-비용초과(10M/M) -일정지연(4M) -품질불만족	사용	데이터전환오류
9	15M	180 M/M	D사 개발방법론v1.0 VB.Net	-비용준수 -일정준수 -품질만족	미사용	
10	6M	30 M/M	D사 개발방법론 v2.0 J2SE/RSA/Requisit pro	-비용준수 -일정준수 -품질만족	사용	

<표 6> 추적테이블 효과검증 표본프로젝트-II

표본No	기간	규모	방법론/적용기법/도구	결과 (계획대비 실적초과치)	본 연구 적용여부	성공/실패 요인
11	5M	20 M/M	D사 개발방법론v1.0 VB.Net	-비용준수 -일정준수 -품질만족	미사용	
12	12M	154 M/M	D사 개발방법론v1.0 VB.Net	-비용초과(16M/M) -일정지연(4M) -품질불만족	미사용	요구사항 불충분 사용자 참여/합의부족 과한 기대
13	6M	120 M/M	마르미III .Net/EAI/SOA	-비용준수 -일정준수 -품질불만족	사용	시스템성능(속도 등) 불만족
14	8M	20 M/M	D사 개발방법론v1.0 ASP.Net	-비용준수 -일정준수 -품질만족	미사용	운영시 추가/변경요구사항 반영
15	11M	54 M/M	D사 개발방법론v1.0 J2SE	-비용준수 -일정지연(2M) -품질만족	미사용	오픈 위한 물리/행정조치 지연

먼저 추적테이블의 효과에 대한 가설을 다음과 같이 수립한다.

- 1) 1차 가설
 - 귀무가설 : 추적테이블 사용여부가 프로젝트 성공에 영향이 없다
 - 대립가설 : 추적테이블 사용여부가 프로젝트 성공에 영향이 있다
- 2) 2차 가설
 - 귀무가설 : 추적테이블은 요구사항관리 미흡으로 인한 프로젝트 실패요인을 감소시키는데 영향이 없다.
 - 대립가설 : 추적테이블은 요구사항관리 미흡으로 인한 프로젝트 실패요인을 감소시키는데 영향이 있다.

1차 가설에서 프로젝트 성공에 대한 정의는 PMBOK[18]의 프로젝트 통합관리 영역인 품질, 일정, 비용 영역에 대해 계획준수율로 하며, 설문에서 3가지 각 항목에 대해 점수를 매긴다. 이중 품질영역은 나머지 두 영역에 비해 사용자 측면의 가치를 포함하므로 가중치를 1.4로 부여한다. 성공점수의 범위는 1에서 100사이의 값으로 100에 가까울수록 성공한 프로젝트로 가정한다.

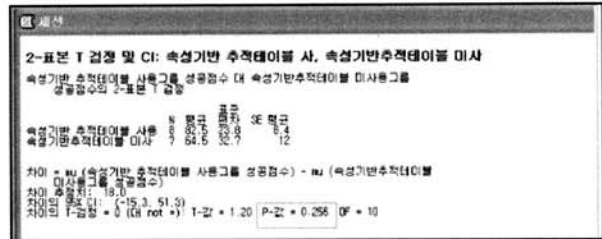
1차 가설에 대해 진행한 설문을 정리하면 <표 7>과 같다.

1차 가설을 검증하는 방법으로 2-표본 t 검정을 사용한다. 2-표본 t 검정은 두 모집단 평균의 차이가 유의한지 여부를 결정하기 위한 가설 검증이다. 두 모집단 평균의 차이가 가정된 값과 같다는 귀무가설($H_0: \mu_1 - \mu_2 = 0$)을 사용하며 좌측($\mu_1 - \mu_2 < \mu_0$), 우측($\mu_1 - \mu_2 > \mu_0$) 또는 양측($\mu_1 - \mu_2 \neq \mu_0$) 대립가설에 대해 귀무가설을 검증한다 [19]. 사용된 도구는 MINITAB15의 통계분석>기초통계학>2-표본 t 검정 분석이다.

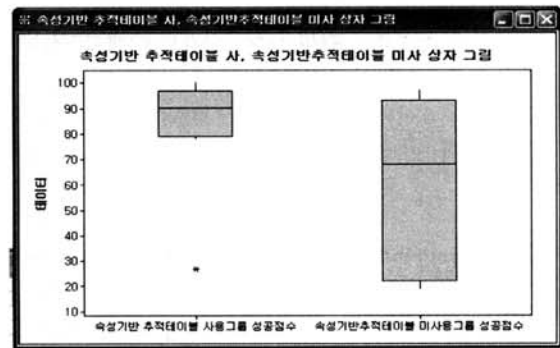
<표 7>의 데이터를 추적테이블 사용그룹과 미사용 그룹으로 나뉘고 통계도구에 각각 성공점수를 입력하여 나온 결과가 (그림 12)이다. 검정결과에서 확률 P값이 0.256이며, 이는 신뢰구간 0.005를 벗어나므로 귀무가설은 기각하고 대립가설을 채택한다.

따라서 추적테이블 사용은 프로젝트 성공에 영향을 미치며, 사용할수록 성공점수가 올라가는 영향이 있다.

표본의 분포 및 중심위치, 변동성을 보여주는 상자그림 그래프로 표본의 분포도를 나타내면 (그림 13)과 같다. 추적테이블 사용그룹이 높은 성공점수에 분포하고 있음을 알 수 있다.



(그림 12) 2-표본 t 검정 결과



(그림 13) 추적테이블 사용/미사용그룹의 성공점수 분포도

<표 7> 표본프로젝트의 성공점수 설문데이터

표본번호	본 연구 적용여부	표본 수행 결과						결과합 (항목*가중치)
		비용		일정		품질		
		1-29 계획상이->계획만족	가중치	1-29 계획상이->계획만족	가중치	1-30 계획상이-> 계획만족	가중치	
1	미사용	25	1	15	1	20	1.4	68.0
2	사용	29	1	29	1	20	1.4	86.0
3	미사용	18	1	18	1	19	1.4	62.6
4	미사용	10	1	5	1	5	1.4	22.0
5	사용	29	1	29	1	30	1.4	100.0
6	사용	29	1	29	1	27	1.4	95.8
7	사용	29	1	29	1	28	1.4	97.2
8	사용	15	1	5	1	5	1.4	27.0
9	미사용	29	1	29	1	25	1.4	93.0
10	사용	29	1	29	1	26	1.4	94.4
11	미사용	29	1	29	1	28	1.4	97.2
12	미사용	4	1	8	1	5	1.4	19.0
13	사용	25	1	29	1	20	1.4	82.0
14	미사용	26	1	29	1	25	1.4	90.0
15	미사용	25	1	18	1	25	1.4	78.0

2차 가설을 검증하기 위해서는 추적테이블 사용여부와 프로젝트 실패요인간의 연관관계를 분석하는 피어슨 상관관계 분석(Pearson Correlation Analysis)기법을 사용한다[19]. 피어슨 상관계수(Pearson correlation coefficient)는 두 변수간의 관련성을 구하기 위해 보편적으로 이용된다. 이때 값의 크기가 0.2~0.4는 약한 상관관계, 0.5~0.6 중간정도, 0.7이상이면 강한 상관관계를 나타낸다[19].

2차 가설 데이터는 표본 프로젝트에서 비용, 일정, 품질 중 한가지라도 만족하지 않은 프로젝트를 (그림 14)와 같이 실패그룹의 표본으로 정한다. 실패그룹에 속한 표본에서 추적테이블을 사용한 표본 4개와 사용하지 않은 표본 4개에 대해 실패요인을 설문한 데이터를 정리하면 <표 8>과 같다.

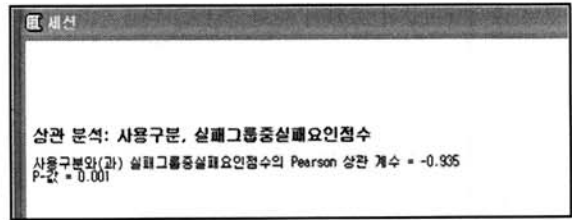
결과에서 피어슨 상관계수가 -0.935 로 강한 음의 값이다. 이는 강한 두 변수간에 강한 상관관계가 있음을 나타내며, 음의 값이므로 두 변수는 서로 반대방향으로 움직인다는 의미이다. 즉, 추적테이블 사용여부와 프로젝트 실패요인 점수는 상호 강한 연관을 가지고 영향을 미친다. 또한 추적테이블을 사용하면 실패요인점수는 감소영향이 있고 그렇지 않으면 증가하는 영향이 있음을 말한다. 이를 산점도 그래프로 보면 (그림 16)과 같다.

따라서 2차 가설에서 귀무가설이 기각되고 대립가설이 채택되어 추적테이블은 실패요인을 감소시키는데 영향이 있다.

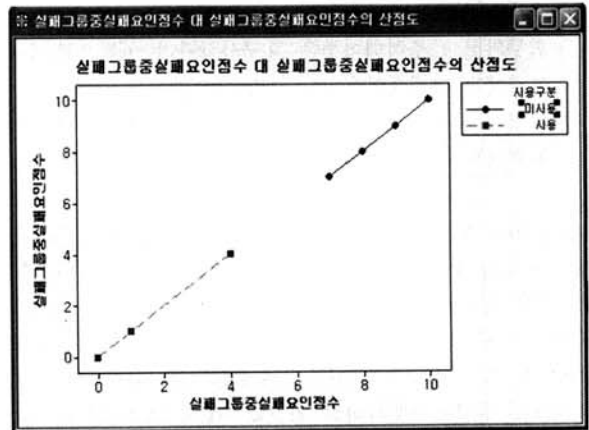
<표 8>의 데이터를 통계도구인 MINITAB15의 통계분석>기초통계학>P 상관관계에 입력하면 결과는 (그림 15)와 같다.

실패	표본 8 실패그룹 I	표본 15 표본 13	표본 12 표본 4 표본 3
	표본 2	표본 1	
성공	표본 10 성공그룹 I	표본 7 표본 6 표본 5	표본 14 표본 11 표본 9

(그림 14) 표본의 성공/실패 및 추적테이블 사용/미사용 분류



(그림 15) 피어슨 상관관계분석 결과



(그림 16) 추적테이블 사용여부와 실패요인의 산점도

1차, 2차 두 가설검정을 통해 추적테이블은 프로젝트 성공에 영향이 있으며, 요구사항관리미흡으로 인한 프로젝트 실패요인을 감소시키는 효과가 있다고 검증한다.

6. 결론 및 향후 연구 계획

6.1 결론

본 연구는 추적테이블을 이용한 요구사항 변경관리가 기존 연구에 비해 변경을 및 변경영향을 용이하게 추정할 수 있음을 제시하였다. 이로써 소프트웨어 프로젝트 이해 당사자인 사용자, 프로젝트관리자, 개발자가 변경내용 및 영향도를 추적테이블에서 상호 공유하고 쉽게 이해함으로써 요구사항 변경을 통제할 수 있게 한다. 즉 소프트웨어 프로젝트 이해당사자들이 요구사항의 진화과정 및 변경이력을

<표 8> 실패그룹 표본의 실패요인점수 설문데이터

표본번호	사용여부	실패요인						실패요인 점수
		불완전한 요구사항	사용자 참여부족	비현실적기대	요구사항 변경	필요성 인식부족	기타	
2	사용	3	0	0	1	0		4
8	사용	0	0	0	1	0	데이터전환	1
13	사용	0	0	0	0	0	시스템성능(속도등)불만족	0
15	사용	0	0	0	1	0	오픈 위한 물리/행정조치 지연	1
1	미사용	3	0	1	0	3		7
3	미사용	2	3	0	0	3		8
4	미사용	3	1	0	3	3		10
12	미사용	2	3	0	3	1		9

찰하고 통제함으로써 사용자의 참여가 유도된다. 이로써 합의된 요구사항을 명세하고 승인하여 비현실적인 요구 및 기대 등의 실패요인을 감소시킬 수 있을 것이다.

한편 연구의 변경관리를 포함하고 있는 추적테이블의 사용이 프로젝트 실패요인을 감소시키는 효과가 있음을 확인하였다. 이로써 추적테이블의 활발한 사용 및 다양한 활용을 권장하면 성공적인 소프트웨어 개발 및 관리를 할 수 있다.

6.2 향후 연구 계획

본 연구는 추적테이블을 이용한 수작업방식인데 산출물의 모든 속성이 데이터베이스화 되어 있다면 변경관리를 비롯한 추적테이블의 활용범위가 더 넓어질 것이다. 특히 형상관리, 테스트관리, 변경관리 등의 도구와 연계하여 상호 데이터를 교환하여 연계하는 것도 가능할 것이다. 따라서 추적테이블을 데이터베이스 형태의 시스템화하는 연구를 진행하고자 한다.

또한 추적테이블 효과연구에 있어 본 연구는 변경관리 방안을 포함하고 있는 추적테이블의 사용과 추적테이블을 아예 사용하지 않은 경우에 대해 가설검정하고 있다. 향후에는 이보다 더 나아가 추적테이블 개선내용에 따라 개선효과만을 보다 명확하게 검증할 수 있도록 연구하고자 한다.

한편, 추적테이블 외에 프로젝트 실패요인을 감소시킬 수 있는 방안에 대해 좀더 확대하여 연구하여 프로젝트 실패요인별로 구체적인 감소방안을 제시하고자 한다.

참 고 문 헌

- [1] Standish Group, CHAOS Reports(1994), 1995.
- [2] Standish Group, CHAOS Reports(2004), 2004. 01.
- [3] CMMI Product Team, Capability Maturity Mode Integration (CMMI), Software Engineering Institute(SEI), 2002.
- [4] 정유신, CMMI 기반 요구사항 추적성관리 프로세스 준수의 결합감소 효과 분석 연구, 서강대학교 석사학위 논문, 2007. 12.
- [5] M. Elizabeth C. Hull, Introduction to SW Verification and Validation, Requirements Engineering, 2005.
- [6] Balasubramaniam Ramesh, Toward Reference Models for Requirements Traceability, IEEE Transactions on software Engineering Vol.27 No.1, 2001. 01.
- [7] G. Anoniol, G. Canfora, G. Gasazza, A. De Lucia, "Recovering Traceability Links between Code and Documentation," IEEE Transaction on Software Engineering, Vol.28 No.10, pp.970-983, 2002. 10.
- [8] J. Cleland-Huang, D. Schmelzer, Dynamically Tracing Non-Functional Requirements through Design Pattern Invariants, Workshop on Traceability in Emerging Forms of Software Engineering, in conjunction with IEEE International Conference on Automated Software Engineering, 2003. 10.
- [9] A. Egyed, "A Scenario-Driven Approach to Traceability", Proceedings of The 23rd International Conference on Software Engineering, pp.123-132, 2000. 06.
- [10] 김주영, 류성열, 요구사항 추적테이블의 확장 및 정규화 방안, 한국정보처리학회 논문지 제16권 제2호, 2009. 04.
- [11] Joost de Wit, Maria Laura Ponisio, Looking for Reasons behind Success in Dealing with Requirements Change, Technical Report TR-CTIT-08-07, Centre for Telematics and Information Technology, University of Twente, Enschede, 2008. 08.
- [12] L. Goldin and A. Finkelstein, "Abstraction-based requirements management," Proceedings of The 2006 ACM International workshop on Role of abstraction in software engineering (ROA '06), New York, NY, USA, pp.3-10, 2006. 05.
- [13] D. Zowghi and N. Nurmuliani, "A study of the impact of requirements volatility on software project performance," Proceedings of The Ninth Asia-Pacific Software Engineering Conference(APSEC '02), Washington, DC, USA, pp.3-13, 2002. 12.
- [14] Luigi Lavazza, Giuseppe Valetto, "Enhancing Requirement and Change Management through Process Modeling and Measurement," Proceedings of The 4th International Conference on Requirements Engineering (ICRE'00), pp. 106-118, 2000. 01.
- [15] 정규장, "개발방법론의 요구사항 변경관리를 개선하기 위한 프로세스 모델", 정보과학회 논문지 소프트웨어 및 응용 제 30권 제6호, pp 504-514, 2003. 06.
- [16] K. Weigers, Software Requirements, Redmond:Microsoft Press, 2003.
- [17] Project Closed Report, 대림아이엔에스, 2008. 12.
- [18] Project Management Body Of Knowledge 2nd Edition, PMI, 2000.
- [19] 김계수, 인과분석 연구방법론, 청람, 2006

부 록

A. 추적테이블 효과 가설검정 설문결과

설문I

프로젝트에 대해 계획대비 실적을 비용, 일정, 품질 각 항목에 대해 평가하시오

계약상이 -----계약만족

1 ----- 29(30)

표본 No.	추적 테이블 사용 여부	사례 수행 결과						
		비용	가중치	일정	가중치	품질	가중치	결과 합(항목*가중치)
		1 - 29 계획상이->계획만족	1	1 - 29 계획상이->계획만족	1	1 - 30 계획상이->계획만족	1.4	
1	미사용	25	1	15	1	20	1.4	68.0
2	사용	29	1	29	1	20	1.4	86.0
3	미사용	18	1	18	1	19	1.4	62.6
4	미사용	10	1	5	1	5	1.4	22.0
5	사용	29	1	29	1	30	1.4	100.0
6	사용	29	1	29	1	27	1.4	95.8
7	사용	29	1	29	1	28	1.4	97.2
8	사용	15	1	5	1	5	1.4	27.0
9	미사용	29	1	29	1	25	1.4	93.0
10	사용	29	1	29	1	26	1.4	94.4
11	미사용	29	1	29	1	28	1.4	97.2
12	미사용	4	1	8	1	5	1.4	19.0
13	사용	25	1	29	1	20	1.4	82.0
14	미사용	26	1	29	1	25	1.4	90.0
15	미사용	25	1	18	1	25	1.4	78.0

B. 추적테이블 효과 가설검정 설문결과

설문II	
실패그룹 프로젝트만, 요구사항관리미흡 실패요인 각 항목에 대해 프로젝트의 실패요인에 기여한 정도를 평가하시오	
3 : 전적으로 그렇다	2 : 50%정도 그렇다
1 : 10~20% 이내에 일부 그렇다	0 : 거의 영향이 없다

표본 No.	성공 실패 그룹	추적 테이블 사용 여부	실패요인					기타점수
			불완전한 요구 사항	사용자 참여부족	비현실적기대	요구사항변경	필요성인식부족	
2	실패 그룹	사용	3	0	0	1	0	4
8	실패 그룹	사용	0	0	0	1	0	데이터전환 1
13	실패 그룹	사용	0	0	0	0	0	시스템성능(속도 등) 불만족0
15	실패 그룹	사용	0	0	0	1	0	오픈 위한 물리/행정 조치 지연1
1	실패 그룹	미사용	3	0	1	0	3	7
3	실패 그룹	미사용	2	3	0	0	3	8
4	실패 그룹	미사용	3	1	0	3	3	10
12	실패 그룹	미사용	2	3	0	3	1	9



김 주 영

e-mail : gogumacake@hanafos.com

1995년 동아대학교 컴퓨터공학과(학사)

2004년 숭실대학교 정보과학대학원 소프트웨어공학과(공학석사)

2009년 숭실대학교 컴퓨터학과(공학박사)

2002년~현 재 대림산업 정보시스템실 차장

관심분야: 소프트웨어 요구공학, 소프트웨어 품질관리



류 성 열

e-mail : syrhw@ssu.ac.kr

1981년~현 재 숭실대학교 컴퓨터학부 교수

1982년~1995년 숭실대학교 전자계산원 원장 및 중앙전자계산소 소장

1997년~1998년 George Mason University 객원 교수

1998년~2001년 숭실대학교 정보과학대학원 원장

2004년~현 재 한국품질재단 운영위원회 위원장

2006년~현 재 공정거래위원회, 기획재정부, 보건복지부 정보화 위원

2008년~2009년 정보통신연구진흥원 비상임이사

관심분야: 소프트웨어 요구공학, 소프트웨어 유지보수, 오픈소스 소프트웨어



황 만 수

e-mail : mshwang@shc.ac.kr

1984년 중앙대학교 전자계산학과(이학사)

1986년 중앙대학교 전자계산학과(이학석사)

2001년 숭실대학교 컴퓨터공학과(공학박사)

1987년~1993년 LG 소프트웨어 연구원

1993년~현 재 신홍대학 컴퓨터정보계열 교수

관심분야: 소프트웨어 요구공학, 소프트웨어 품질관리, 소프트웨어 프로세스 개선