

# 가치분석을 통한 휘처 기반의 요구사항 변경 관리

## Feature-Oriented Requirements Change Management with Value Analysis

안상임(Sangim Ahn)\*, 정기원(Kiwon Chong)\*\*

### 초 록

소프트웨어 개발 초기에 모든 요구사항을 정의하는 것은 불가능 하기 때문에 요구사항은 소프트웨어 개발이 진행되는 동안에 지속적으로 변경된다. 이러한 요구사항 변경은 개발자가 소프트웨어 구조나 행위를 완벽하게 이해하지 못하거나 변경에 따라 영향을 받는 모든 부분을 식별할 수 없을 경우 많은 오류를 야기시킨다. 그러므로, 조직의 비즈니스에 공헌하면서 비용·효과적으로 적절히 처리되기 위하여 요구사항은 관리되고 평가되어야 한다. 본 논문은 가치분석을 통하여 생성된 휘처 기반의 요구사항 추적 링크를 근간으로 하는 요구사항 변경 관리 기법을 제안한다. 이는 사용자 요구사항과 산출물간의 연결을 분석하기 위하여 휘처를 중간 매개체로 활용한 추적 링크를 이용한다. 그리고, 요구사항 변경 요청을 휘처 단위로 상세화하기 위한 변경 트리 모델을 정의하고 변경 관리가 수행되는 전체적인 프로세스를 제시한다. 또한, 요구사항 변경 관리 기법을 자산관리포털시스템에 적용한 사례의 결과를 기술한다.

### ABSTRACT

The requirements have been changed during development progresses, since it is impossible to define all of software requirements. These requirements change leads to mistakes because the developers cannot completely understand the software's structure and behavior, or they cannot discover all parts affected by a change. Requirement changes have to be managed and assessed to ensure that they are feasible, make economic sense and contribute to the business needs of the customer organization. We propose a feature-oriented requirements change management method to manage requirements change with value analysis and feature-oriented traceability links including intermediate catalysis using features. Our approach offers two contributions to the study of requirements change: (1)We define requirements change tree to make user requirements change request generalize by feature level. (2)We provide overall process such as change request normalization, change impact analysis, solution dealing with change request, change request implementation, change request evaluation. In addition, we especially present the results of a case study which is carried out in asset management portal system in details.

키워드 : 요구사항 추적성, 휘처 모델링, 휘처기반 요구사항 변경관리 프로세스, 요구사항 변경 트리  
Requirements Traceability, Feature Modeling, Feature-Oriented Requirements  
Change Management Process, Requirement Change Tree

---

본 연구는 숭실대학교 교내연구비 지원으로 이루어졌음.

\* 숭실대학교 대학원 컴퓨터공학과

\*\* 숭실대학교 대학원 컴퓨터공학과 교수

## 1. 서 론

대부분의 요구사항은 소프트웨어 개발 초기에 정의되고 시스템과 관련된 이해당사자, 고객 조직, 운영환경 등의 요구가 변경되는 것을 반영하기 위하여 소프트웨어 개발 생명주기 동안에 지속적으로 진화한다. 이러한 변경은 시스템, 다른 요구사항, 조직의 목표에 영향을 미치게 된다. 따라서, 요구사항 변경은 조직의 비즈니스에 공헌하면서 비용·효과적으로 적절히 처리되기 위하여 관리 및 평가되어야 한다. 또한, 최근 모든 요구공학 분야에서 고객 및 사용자의 만족을 넘어서는 가치 창출에 대한 중요성이 대두됨에 따라 가치분석(Value analysis), 창의적 요구공학 프로세스(Creativity requirements engineering process), 비즈니스 기반 요구사항 분석(Business-based requirements analysis), 비즈니스 전략(Business strategy)등과 접목하여 요구사항의 가치를 평가하고 새로운 고객 가치를 창출하는 요구공학 기법이 필요하다[1, 2].

본 논문에서는 비용·효과적으로 요구사항 변경을 처리하기 위한 휘처 기반의 요구사항 변경 관리 기법(Feature-Oriented Change Management Method, FoCM)을 제안한다. 이는 노력, 비용, 위험 등을 고려한 가치분석을 근간으로 생성된 휘처 기반의 요구사항 추적 링크를 이용한다. 추적 링크는 요구사항과 아키텍처 설계서, 소스 코드, 테스트 케이

스 등과 같은 시스템 산출물들 사이의 연결을 의미하는데, 요구사항 변경 영향 분석, 요구사항 충돌 분석, 요구사항 일관성 점검에 가장 중요한 요소이다. 휘처 기반의 요구사항 변경관리를 위해서 우선 일반적인 요구사항 변경 요청을 휘처 단위로 상세화하기 위한 변경 트리 모델을 정의하고, 변경 관리가 수행되는 과정을 설명하기 위하여 전체적인 프로세스를 제시한다. 변경 트리 모델은 변경 요청, 요구사항, 휘처와의 관계를 형식화한 것이다. 휘처 기반의 요구사항 변경 관리 프로세스는 변경 요청 정규화, 변경 범위 분석, 대처방안 생성, 변경요청 구현, 변경 요청 사후 처리로 구성된다. 또한, 본 논문에서 제안한 휘처 기반의 요구사항 변경 관리 기법의 타당성을 보여주기 위해서 경매 및 공고를 위한 자산관리포털시스템에 적용한 사례 결과를 기술한다.

제 2장에서는 가치기반 소프트웨어 공학 및 요구사항 변경관리와 관련된 기존 연구들을 소개하며, 제 3장에서는 휘처 기반의 요구사항 변경 관리를 위한 요구사항 추적 모델, 요구사항 변경 트리 모델, 변경관리 프로세스에 대하여 제시한다. 제 4장에서는 자산관리포털 시스템에서의 사례연구를 통하여 제안된 기법의 실용성을 검증한다. 제 5장에서는 본 연구에서 제안된 기법과 관련된 기법들을 여러 기준 항목으로 비교한다. 마지막으로 제 6장에서는 결론을 기술하고 향후 연구를 제시하며 본 논문을 마무리 한다.

## 2. 관련 연구

### 2.1 가치 기반 소프트웨어 공학

많은 기업들은 요구사항 추적, 결함 추적, 형상관리 등과 같은 소프트웨어 공학 실행을 가치-중립(Value-neutral) 방식으로 수행하고 있으며, 이러한 실행들은 자주 비즈니스와 연관된 가치의 이해가 없는 상태로 진행된다. 그러나, IT 환경이 급변하는 오늘날 기업이 성공하기 위해서 투자는 보다 현명하게 이루어져야 하며 더 이상 가치가 고립된 상태로 소프트웨어 공학이나 정보기술이 수행될 수 없다.

이를 해결하기 위해서, Boehm[3]은 기존 또는 새로 만들어지는 소프트웨어 공학 원리 및 실행과 이것들이 상호 조화롭게 강화되기 위한 전반적인 프레임워크를 개발하는데 가치 중요성을 통합하기 위하여 가치 기반 소프트웨어 공학(Value-based software engineering, VBSE)을 제안한다. VBSE는 가치 기반 환경하에서 소프트웨어 공학 행위들이 기본으로 사용하게 되는 Benefits Realization Analysis, Stakeholder Value Proposition Elicitation and Reconciliation, Business Case Analysis, Continuous Risk and Opportunity Management, Concurrent System and Software Engineering, Value-Based Monitoring and Control, Change as Opportunity과 같은 일곱 가지 원리를 포함한다. 또한, Boehm은 시스템 개발 초기에 비즈니스 가치와 연관되어야 할 몇몇 소프트웨어 공학 실행들에 대하여 기술하고 있다. 그 중에서 하나가 가치 기반

요구공학(Value-base Requirement Engineering, VBRE)이다. 가치 기반 요구공학은 중요한 이해당사자 식별, 이해당사자의 요구를 시스템의 기능에 지속적으로 접목, 상호 목적에 부합하는 시스템 개발과 관련된 원리 및 실행에 대하여 기술하고 있다. 요구사항이나 이해당사자의 목적은 개발중인 시스템이 고객에게 가치를 제공하고 언제나 고객의 요구를 정확하게 반영한다는 확신을 주기 위하여 관리되고 추적될 필요가 있는 요소이다. 요구사항 추적성은 프로젝트 생명주기 또는 이 외에서 요구사항을 추적 및 관리를 가능하게 하는 메커니즘이다. 요구사항 추적의 가치는 프로젝트 실패의 주요 요인이 소프트웨어 개발 생명주기 동안 요구사항과 관련된 문제이기 때문에 매우 중요하다. 따라서, 요구사항 추적이 소프트웨어 공학의 주요 행위로 이해되기 위하여 행위 자체로서가 아니라 프로젝트에서 가치를 제공하여야 한다. 즉, 가치를 이해하기 위해서 다양한 요구사항들을 가지고 있는 여러 기업들에게 유용하도록 융통성 있는 방식을 제공해야 한다.

Zemont[4]는 이러한 요구사항 추적의 가치를 보여주기 위하여 VBSE의 일곱 가지 원리를 근간으로 다음과 같은 가치 기반 요구사항 추적 프레임워크(Value-based Requirements Traceability Framework)을 제안한다. 이때, VBSE의 원리는 프로세스로 이용된다.

- Stakeholder Value Proposition Elicitation and Reconciliation: 기업은 요구사항 추적 프로세스에 관련된 이해당사자를 식별하고, 그들의 요구 및 요구들의 충돌을 평가

- Benefits Realization Analysis: 요구사항 추적으로부터 획득된 가치를 이해하고 실현된 목표에 맵핑
- Business Case Analysis: 목표에 대한 검증 및 재정적인 배대 형성
- System and Software Engineering: 요구사항 추적을 실무적으로 이용
- Value-Based Monitoring and Control: System and Software Engineering 프로세스를 실행하는 동안 변경 및 위험에 대한 측정, 관찰, 대안 실행
- Continuous Risk and Opportunity Management: 위험 발생에 따른 조치
- Change as Opportunity: 변경 요청 처리

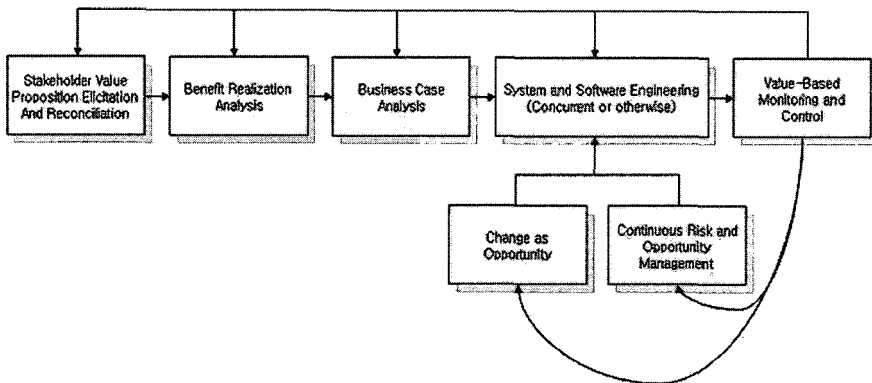
Continuous Risk and Opportunity Management 와 Change as Opportunity 프로세스에서 형성된 대안들은 다시 System and Software Engineering 프로세스로 반영된다. 또한, Value-Based Monitoring and Control 프로세스는 조정이 필요할 경우 내용을 나머지 프로세스에게 전달한다. <그림 1>은 VBRT내 프

로세스들의 연관관계를 보여주는 모델이다. Zemont[4]는 요구사항 추적에 대한 가치를 정량적으로 관리하기 위한 프레임워크를 수립하였지만 요구사항 추적 및 Change as Opportunity 등 일부 프로세스에 대한 기술적인 방법에 대해서는 미흡하다. 따라서 본 논문에서는 가치를 근간으로 하는 요구사항 추적 및 변경 요청에 대한 처리를 기술적으로 처리하기 위한 방법을 제안하고자 한다.

## 2.2 기존 요구사항 변경 관리 연구들

기존 요구사항 변경관리 연구들은 요구사항의 변경 요청을 처리하기 위한 프로세스에 초점이 맞추어져 진행되어 왔다.

Wieggers[1]에서 제시하는 변경 관리 프로세스는 템플릿 기반으로 프로세스를 실행하기 전에 충족되어야 하는 조건인 시작 기준(Entry criteria), 프로세스에 포함된 다양한 작업(Tasks), 작업이 제대로 완료되었는지 검증(Verify)하는 단계들, 프로세스가 성공적으로 완료된 것을 알려주는 종료 조건



<그림 1> Value-based Requirements Traceability Model

(Exit criteria)으로 구성된다. Kotonya[5]에서 제시하는 변경관리 프로세스는 첫째, 문제 분석 및 변경 작성(Problem analysis and change specification)단계에서는 새로운 고객의 요구나 시스템 동작 문제 등으로부터 요구사항 문제가 발견되면, 이를 분석하여 변경 요청을 수락한다. 둘째, 변경 분석 및 비용 산정(Change analysis and costing)단계에서는 변경 요청이 영향을 미치는 요구사항 범위를 분석하고, 변경에 수반되는 비용과 시간을 산정한다. 셋째, 변경 구현(Change implementation)단계에서는 새로운 요구사항이 구현된다.

Lock[6, 7, 8]에서 제시하는 변경관리 프로세스는 6단계들로 구성된다. 첫째, 변경 요청 생성(Change request creation)단계에서는 변경 요청서를 작성한다. 둘째, 영향 분석(Impact analysis)단계에서는 변경 요청에 의해 영향을 받는 범위를 분석하고 결과를 문서화 한다. 셋째, 시스템 배포 계획(System release planning)단계에서는 비용과 우선순위가 결정된다. 넷째, 변경 구현(Implementation changes)단계에서는 제안된 변경 요청을 구현한다. 다섯째, 변경 시험 및 검증(Test and verify changes) 단계에서는 개발된 컴포넌트가 정확하게 잘 동작하는지 시험 및 검증한다. 여섯째, 시스템 배포 및 통합(System release and integration) 단계에서는 고객에게 완성된 시스템을 배달하고, 동작 환경에 설치한다. 박[9]은 소프트웨어 프로젝트 라인 공학에서 일관성 있는 요구사항 변경 관리를 위해 변경관리 프로세스를 제시한다. 이는 추상화된 변경 요청을 개발자가 처리할 수 있는 단위로 재정의해서 상세화하는

변경 요청 처리 단계와 요청된 변경이 다른 요구사항에 미치는 영향의 범위를 분석해서 도식화하는 변경 분석 단계, 컴포넌트 수준에서 변경 처리를 할 수 있도록 대처방안을 제시하는 변경 대처방안 설계 단계, 마지막으로 명세서로 제시되는 변경 대처방안을 구현하는 변경 처리 단계로 구성된다. 이렇듯, 기존 요구사항 변경관리 연구들은 프로세스 및 일부 기법에 대해서는 방안을 제시하고 있지만 이해당사자들의 요구에 대한 가치를 이해하거나 평가하는 부분의 처리가 부족하다. 따라서, 본 논문에서는 요구사항 변경을 효과적으로 처리하기 위하여 변경 요청에 대한 가치분석 및 이를 일관되게 처리하기 위한 프로세스를 제시하고자 한다.

### 3. 휘처 기반의 요구사항 변경 관리

휘처 기반의 요구사항 변경 관리의 목적은 비용, 노력, 위험 등과 같은 가치를 고려하여 생성된 요구사항 추적 링크를 이용하여 새로운 변경 요청을 효율적으로 처리하는 것이다. 제 3장에서는 휘처 기반의 요구사항 추적에 대한 모델, 변경 요청을 세분화한 변경 트리 모델, 그리고 요구사항 변경관리를 위한 전체적인 프로세스와 활동에 대하여 기술한다.

#### 3.1 휘처 기반의 요구사항 추적

휘처는 제품의 주요 특성을 의미한다. 휘처 기반의 요구사항 추적은 사용자 요구

사항을 공통성과 가변성에 의해 조직화된 위치를 중간 매개체로 하여 비용, 노력, 위험에 의해서 우선순위화된 요구사항을 근간으로 가치 있는 추적 링크를 식별하는 것이다[10]. <그림 2>는 [10]에서 제시한 위치 기반 추적링크를 근간으로 위치 기반의 추적성을 가시적으로 표현하기 위하여 메타 모델이다. 위치 기반의 요구사항 추적 관련 메타 모델은 우선 사용자 요구사항이 입력되면, 이를 위치로 그룹핑한 후 위치와 구현 산출물들을 연결하기 위한 추적 링크를 생성하는 과정을 표현하고 있다[10].

위치들은 각각 고유한 이름을 가지고 있으며 성능, 운영환경, 도메인 기술, 구현 기술과 같은 위치 카테고리에 의해서 분류된다. Kang[11]에서는 위치 모델 개념이 기술되어 있고 Lee[12]에서는 실용적으로 위치를 모델링하는 가이드라인을 제시하고 있다. 각 위치 카테고리는 해당되는 요소들로 다시 세분화되며, 세분화된 위치들은 구성, 일반화, 특별화, 구현, 옵션, 대안 관계에 의해서 연결된다. 이렇게 식별된 위치들을 소프트웨어 개발 단계에서 생성된 산출물과 추적 링크를 통해서 연결된다. 이때, 각 위치들은 가치, 위험, 노력 등의

기준에 의해 평가된 결과인 우선순위를 가지고 있는데, 우선순위 레벨에 따라 낮음, 중간, 높음으로 세분화가 되고 이는 가치 있는 추적 링크를 생성하기 위한 기준으로 사용된다. 즉, 추적링크는 이 우선순위를 참조하여 생성된다.

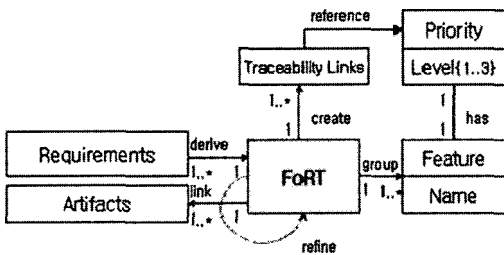
<표 1>은 요구사항 추적 링크를 생성하기 위한 우선순위 레벨을 보여준다. 레벨이 올라갈수록 높은 세분화가 이루어진다. 위치 우선순위화를 위하여 모든 이해 당사자는 각각 요구사항에 대하여 가치, 위험, 노력과 관련된 순위를 추정한다. 이때, 각 위치들은 추정된 결과에 근거하여 분류된다. 중요도가 높음으로 분류된 요구사항의 경우 메서드 레벨까지 추적링크를 생성한다.

<표 1> 우선순위 레벨 및 산출물 분류

우선순위	세분화	산출물 분류
1	낮음	Component
2	중간	Class
3	높음	Method

### 3.2 위치 기반의 요구사항 변경 트리

추상화된 변경 요청을 처리하기 위해서는 시스템의 변경 단위로 재정의가 필요하다. 일반적으로 변경 요청은 여러 개의 요구사항으로 세분화 될 수 있다. <그림 3>은 이러한 일반적인 요구사항 변경 요청이 위치 기반의 변경 관리로 처리되기 위하여 상세



<그림 2> 위치 기반의 요구사항 추적 메타 모델

화되는 과정으로 보여주는 모델이다.

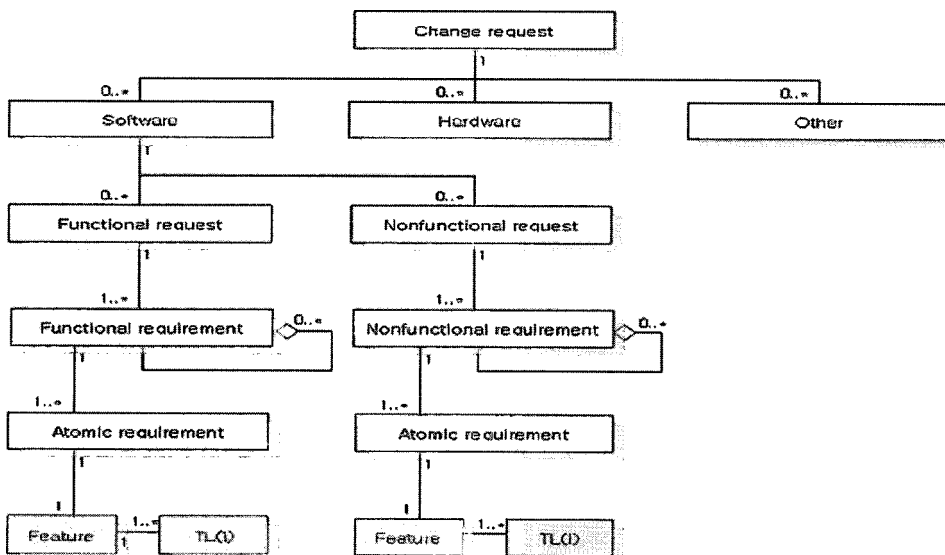
우선, 변경 요청은 소프트웨어, 하드웨어, 이외 다른 컴퓨팅 환경으로 나눌 수 있다. 본 논문에서는 현재 사용중인 소프트웨어에서 변경을 처리해야 할 경우를 대상으로 한다. 현재 사용중인 소프트웨어를 변경하기 위해서 변경 요청은 다시 기능적 변경 요청인지, 비기능적 변경 요청인지로 나누어진다. 기능적인 변경 요청은 다양한 기능적 요구사항들로 처리될 수 있고 비기능적인 변경 요청은 다시 내부에 비기능적인 요구사항으로 구성된다.

기능적 및 비기능적 요구사항은 여러 의미의 요구사항을 포함하므로 다시 단일 요구사항 수준으로 상세화된다. 이러한 단일 요구사항은 공통성과 가변성에 의해 조직화된 휘처와 연결된다. 즉, 기능적 요

구사항은 내부에서 다시 단일 요구사항으로 나누어지며 최종적으로 휘처와 매핑된다. 비기능적인 요구사항은 하나 이상의 기능적 요구사항과 관련되어 있으므로 이는 다시 단일 요구사항으로 나누어 질 수 있다. 또한, 휘처는 변경 요청을 처리하기 위한 범위 분석을 위하여 개발과정에서 생성된 다양한 산출물들과 추적 링크를 통하여 연결된다.

### 3.3 요구사항 변경 관리 프로세스 및 활동

휘처 기반의 요구사항 변경 관리 프로세스는 변경 요청 정규화, 변경 범위 분석, 대처방안 생성, 변경 요청 구현, 변경 요청 사후 처리의 5단계로 구성된다.



〈그림 3〉 휘처 기반의 변경 요청 트리 모델

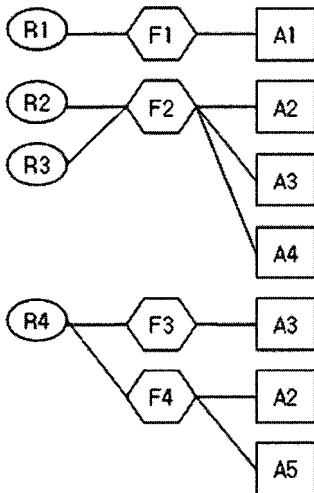
1. 변경 요청자	2. 변경 요청 부서	
3. 변경 근거	5. 서버명	
4. 업무명	8. 변경 내용	
6. 변경 구분 (신규, 변경, 폐기)		
7. 변경 사유		
9. 변경 승인 여부	11. 적용 예정일	12. 적용일
10. 변경 요청일	14. 변경 성공 여부	
13. 가치분석 결과		

〈그림 4〉 변경관리 정보 템플릿

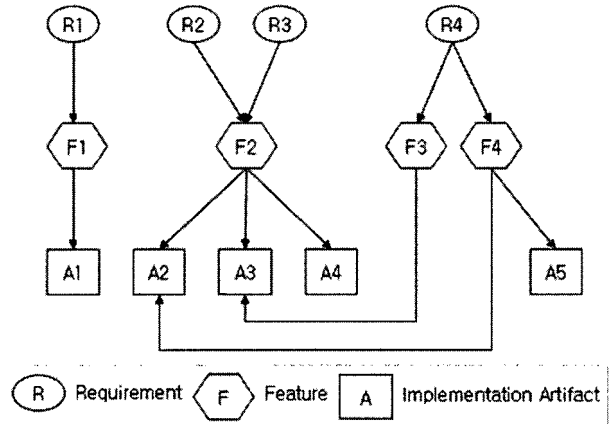
3.3.1 변경 요청 정규화

변경 요청 정규화 단계에서는 사용자로부터 전달된 변경 요청을 시스템에 반영하기 위하여 분석하고 정규화한다. 대부분의 사용자 변경 요청은 애매모호하게 작성되거나 추상적인 경우가 많다. 따라서, 이 단계에서는 첫째, 사용자 변경 요청서를 검토한다. 둘째, 변경 요청서를 입력으로 변경관리 정

보 템플릿을 작성한다. 〈그림 4〉는 변경 관리 정보 템플릿에 포함되어야 할 주요 항목을 보여준다. 셋째, 〈그림 3〉의 변경 요청 트리와 같이 기능적·비기능적 요구사항을 식별한 후 최종적으로 단일 요구사항을 식별한다. 그리고, 모든 변경 요청이 단일 요구사항 사항으로 분류되면, 모든 단일 요구사항에 식별자를 할당한다. 이렇게 식별된 단일 요구사항을 근거로 변경 요청을 재정의



〈그림 5〉 개별 추적링크



〈그림 6〉 변경 영향 구조도



한다. 이 단계의 결과물은 변경관리 정보 및 재정의된 변경 요청서이다.

### 3.3.2 변경 범위 분석

변경 범위 분석 단계에서는 각 변경 요청에서 식별된 단일 요구사항이 영향을 미치는 범위를 분석하고 이를 통하여 개발 비용 및 가치분석을 수행 후 변경 범위를 결정한다. 이를 위하여, 첫째 재정의된 변경요청서의 단일 변경 요구사항과 관련된 휘처와 추적 링크를 추출한다. 그리고, 추적 링크를 통하여 각 산출물간 의존성을 분석함으로써 변경 범위를 식별한다. 둘째, 식별된 개별 추적링크의 상호관계를 분석하여 변경 범위를 도식화한다. <그림 6>은 <그림 5>의 개별 추적 링크의 관계를 분석한 변경 영향 구조를 보여준다. 셋째, 변경 영향 분석 결과를 바탕으로 개발 비용 및 필요 자원을 추정한다. 넷째, 각 단일 변경 요청에 대하여 관계된 프로젝트관리자, 시스템분석가, 개발자, 품질관리자 등 모든 이해당사자가 비용, 노력, 위험 등을 고려한 가치 분석을 실시한다. 이때, 변경요청은 각 항목별로 높음, 중간, 낮음(레벨 3~1)으로 중요도가 결정된다. 그리고, 항목별로 평균치를 계산하여 오름차순으로 정리하여 우선순위를 확정한다. 다섯째, 가치 분석 결과를 근간으로 최종 변경 범위를 결정한다. 마지막으로 변경 승인을 획득한다. 변경 범위 분석 단계의 결과물은 개별 추적 링크 리스트, 변경 영향 구조도, 가치 분석 결과등이다.

### 3.3.3 대처 방안 생성

대처 방안 생성 단계에서는 결정된 변경 범위에 속한 요구사항, 휘처, 산출물에 대한 상세 자료를 제공하여 효과적인 변경 처리를 위한 대처 방안을 제시한다. 그리고, 대처 방안에 제시된 산출물들에 대한 추적을 실시한다. 이 단계의 결과물은 대처 방안 및 추적 리스트이다.

### 3.3.4 변경 요청 구현

변경 요청 구현 단계에서는 실제 변경 요청에 대한 구현이 이루어지고 구현된 소프트웨어에 대한 시험 및 검증을 실시한다. 그리고, 시스템에 통합되거나 고객에게 배포되어 진다. 이 단계의 결과물은 구현된 산출물과 시험 결과 등이다.

### 3.3.5 변경 요청 사후처리

변경 요청 사후처리 단계에서는 변경에 대한 평가를 실시하여 변경 관리 정보 템플릿을 업데이트하고 가치분석 결과인 중요도에 따라 추적링크를 시스템에 반영한다.

## 4. 사례 연구

이 장에서는 제안된 휘처 기반의 요구사항 변경 관리 프로세스와 결과물을 경매 및 공고를 위한 자산관리포털 시스템에 적용한

사례를 보여준다. 단, 금번 사례에서는 기능적 요구사항에 대한 변경만을 고려하였으며, 개발 비용 및 자원추정 절차는 배제하였다.

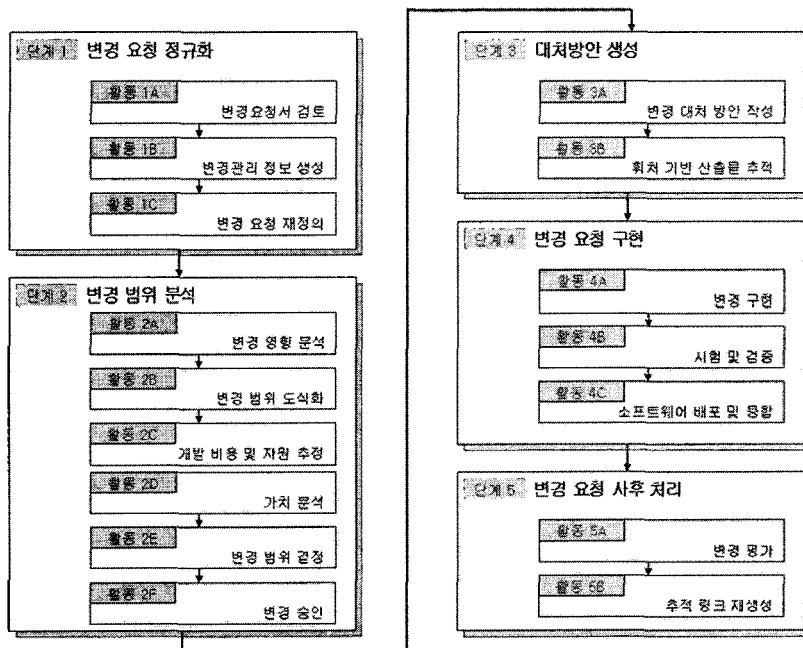
### 4.1 시스템 및 휘처 기반 요구사항 추적 링크 요약

자산관리포탈 시스템은 인터넷을 통하여 판매대상인 부동산 또는 자동차 등의 정보를 시스템에 온라인으로 등록하고 여러 사람이 동시에 해당 물건에 대한 구매를 수행하며, 관심물건, 장바구니, 커뮤니티 구성, 웹메일 제공 기능에 대한 개인화된 페이지가 제공되는 웹 정보시스템이다. 동 사례연구를 위하여 사용자용으로 60개, 관리자용으로 32개의 기능적 요구사항을 분석하였다. 비용,

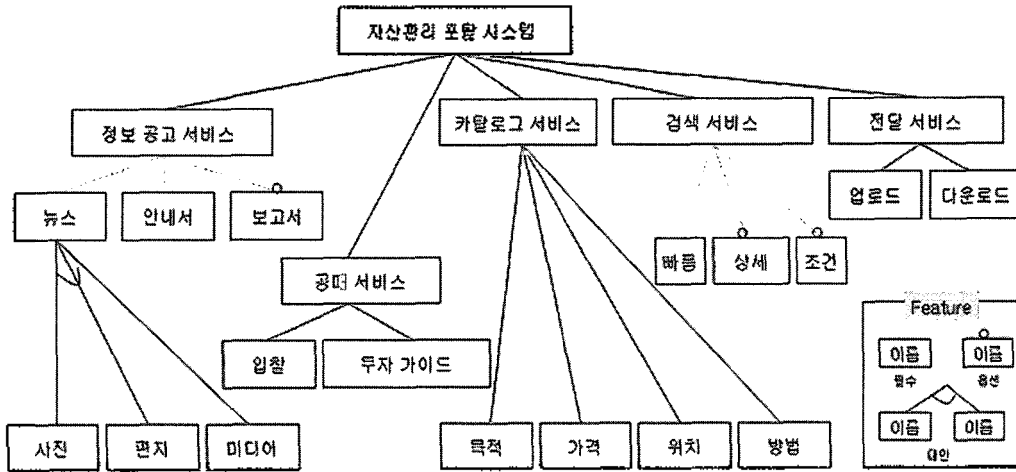
노력, 위험을 고려한 가치분석을 통하여 요구사항에 대한 21% 높음, 45% 중간, 34% 낮음의 우선순위 결과를 얻었다. 그리고, 105개의 휘처를 생성하였으며 총 152,580개(휘처 기반 추적링크를 적용하지 않았을 경우 726,570개)의 추적링크를 생성하였다. (그림 8)은 광고, 카탈로그, 검색 및 전달과 관련된 기능적 요구사항에 대한 휘처 다이어그램을 보여준다.

### 4.2 변경 요청 정규화

변경 요청 정규화의 주요 목적은 사용자 변경요청서를 분석하여 단일 요구사항을 식별한 후 변경 요청서를 재정의하는 것이다. 이를 위하여, 사용자로부터 포탈 시스템의



〈그림 7〉 휘처 기반의 요구사항 변경관리 프로세스



〈그림 8〉 자산관리포탈 시스템의 휘저 다이어그램(일부)

웹에서 검색 조건에 “최저 입찰가”를 추가해 달라는 변경 요청서를 접수하고 변경요청 트리를 이용하여 변경요청서를 분석한 결과 1개의 단일 요구사항이 식별되었고 여기에 CR-012 식별자를 할당하였다.

### 4.3 변경 범위 분석

변경 범위 분석의 주요 목적은 각 변경 요청에서 식별된 단일 요구사항이 영향을 미치는 범위를 분석하고 가치분석을 통하여 변경 범위를 결정한다. 이를 위하여, 입력된 단일 변경 요청과 기존 요구사항들을 비교한 결과 기존 요구사항 식별자 RF-008 『주어진 조건을 이용하여 검색이 가능하여야 한다』와의 연관성을 확인하였

〈표 2〉 변경 요청과 관련된 추적 링크(구현 산출물)

추적링크	요구사항 식별자	휘저 식별자	구현 산출물		
			Package	Class	Method
T131	RF_008	F023	Search	searchContent.class	validDate()
T132	RF_008	F023	Search	searchContent.class	makeQuery()
T133	RF_008	F023	Search	searchContent.class	changeFormContent()
T134	RF_008	F023	Search	searchItem.class	makeQuery()
.....	.....	.....	.....	.....	.....
T137	RF_008	F023	Search	searchNotice.class	makeQuery()
.....	.....	.....	.....	.....	.....

〈표 3〉 변경 요청 CR-012에 대한 대처 방안

변경 요청		CR-012			
변경 범위		변경 타입	시스템 존재 유무	중요도	취치 범위
F21	빠른 검색	변경	○	상	Method
F22	상세 검색	변경	○	중	Class
F23	조건 검색	변경	○	중	Class

다. 이것은 취치 식별자는 F023『조건』과 연결되어 있으며, 관련된 구현 산출물에 대한 추적링크는 〈표 2〉와 같다. 그리고 추적 링크를 통하여 각 산출물간 의존성을 분석하였다. 식별된 개별 추적링크의 상호관계를 분석하여 〈그림 9〉와 같이 변경 범위를 도식화하였다.

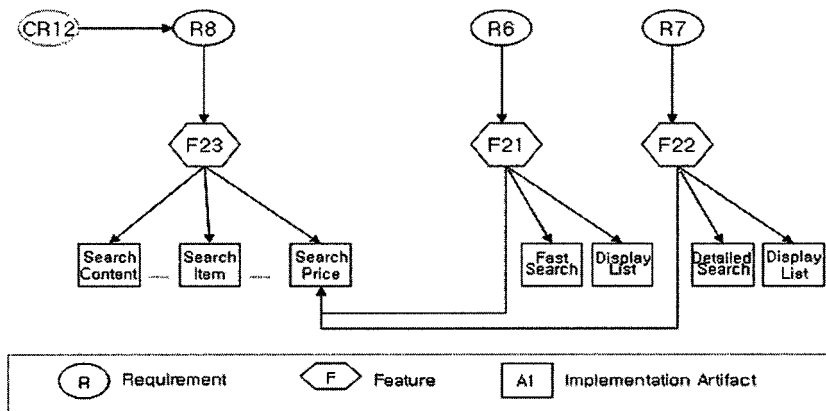
#### 4.4 대처 방안 생성

대처 방안 생성의 주요 목적은 결정된 변경 범위에 속한 요구사항, 취치, 산출물에 대한 상세 자료를 제공하여 효과적인 변경

처리를 위한 대처 방안을 제시하는 것이다. 변경 범위 분석 결과를 근간으로 〈표 3〉과 같은 대처방안을 제시한다.

### 5. 평 가

관련연구에서 기술한 요구사항 변경 관리와 관련된 기존 연구와 본 논문에서 제안하는 취치 기반 요구사항 변경 관리 기법과의 비교를 위하여 「Guide to the Software Engineering Body of Knowledge, SWEBOK」에서 제시하고 있는 요구사항 관리와 관



〈그림 9〉 변경 요청 CR-012에 대한 변경 범위 도식화(일부)

〈표 4〉 기존 연구들과 비교 평가

평가항목 \ 기법	Wieggers[1]	Kotonya[5]	Lock [6,7,8]	박[9]	본 논문의 기법
잘 정의된 프로세스	○	○	○	○	○
가치분석 제공	X	X	X	X	○
변경 범위의 도식화	X	X	○	○	○
휘처 모델링	X	X	X	X	○
가변성 고려	X	X	X	○	○
노력 절감	low	low	High	Medium	High
툴 지원	X	X	○	X	X

련된 주요항목과 효율적인 변경 관리를 위해 필요한 항목을 평가항목으로 선정하였다.

- 잘 정의된 프로세스: 요구사항 변경 관리를 위한 프로세스가 잘 정의되었는지에 대한 평가 기준
- 가치분석 제공, 노력 절감: 변경 영향 분석에서 사용되는 산출물 추적을 위한 링크를 생성하거나 유지보수하기 위하여 비용이나 노력을 절감하기 위한 가치분석을 고려하였는지에 대한 평가 기준
- 변경 범위의 도식화: 변경 영향 분석에 대한 결과를 도식화하였는지에 대한 평가
- 휘처 모델링, 가변성 고려: 향후 요구사항 변경에 대한 대처를 위하여 요구사항에 대한 가변성을 평가하였는지에 대한 평가 기준
- 툴 제공: 요구사항 변경 관리를 위한 자동화를 제공하는지에 대한 평가 기준

〈표 4〉와 같이 휘처 기반의 요구사항 변경 관리 기법은 휘처 모델링을 통하여 향후 발생하는 요구사항 변경을 효과적으로 처리할 수 있도록 가변성을 제공하고 있으며, 가치분석을 통한 추적 링크를 이용하고 유지보수하는 필요한 비용이나 노력을 절감할 수 있다. 그러나, 변경 관리를 효율적으로 관리하기 위한 자동화가 미흡하다. 이는 향후 연구과제로 개선이 필요하다.

## 6. 결론 및 향후 연구

요구사항의 변경은 계획되지 않은 것이 많으므로 이를 반영할 수 있는 관리체계가 만들어져야 한다. 그러나, 기존의 변경관리 기법은 구현 산출물에 대한 변경 처리에 초점을 맞추거나 비용·효과적인 변경 관리를 위한 가치분석에 대한 적용이 미흡하다. 본 논문에서는 가치분석을 통하여 생성된 휘처 기반의 요구사항 추적 링크를 근간으로 하는 요구사항 변경 관리 기법을 제

안하였다. 이는 노력, 비용, 위험 등을 고려한 가치분석을 근간으로 중간 매개체인 휘처를 활용한 요구사항 추적 링크를 이용한다. 그리고, 휘처 기반의 요구사항 변경관리를 위해서 일반적인 요구사항 변경요청을 휘처 단위로 상세화하기 위하여 변경 요청, 요구사항, 휘처와의 관계를 형식화한 변경 트리 모델을 정의하였고 변경 요청 정규화, 변경 범위 분석, 대처방안 생성, 변경 요청 구현, 변경 요청 사후 처리로 구성된 변경 관리 프로세스를 제시하였다. 또한, 경매 및 공고를 위한 자산관리포탈 시스템에 적용한 사례를 통하여 제안한 휘처 기반의 요구사항 변경관리 기법의 타당성을 검증하였다. 등 결과는 대량의 요구사항 추적 링크가 생성되는 복잡한 시스템의 경우에 효과가 더욱 증가된다. 향후 본 연구는 자동화와 같은 시스템적인 접근으로 확장이 필요하며, 대량의 산출물이 작성되는 복잡한 소프트웨어 개발 프로젝트의 요구사항 변경이나 일관성 검증에 활용하여 심도 깊은 실무적인 분석이 수행되어야 한다.

---

### 참 고 문 헌

---

- [1] Karl E. Wiegers, Software Requirements 2/E, Microsoft Press, 2003.
- [2] 김상수, 인호, 이병정, 박수용, "미래의 소프트웨어 공학 기술", 정보과학회지, 제24권, 제12호, pp. 6-16, 2006.
- [3] Biff, S., Aurum, A., Boehm, B. W., et al, Value-based Software Engineering, Springer Verlag, 2005.
- [4] Grant Zemont, Towards Value-Based Requirements Traceability, thesis, DePaul University, Chicago Illinois, 2005.
- [5] Kotonya, G. and Somerville, I., "Requirements Engineering Process and Techniques, John Wiley & Sons, 1998.
- [6] Lock, S. and Kotonya, G., "Requirement Level Change Management and Impact Analysis", Cooperative System Engineering Group, Technical Report Ref: CSEG /21/1998, October, 1998.
- [7] Lock, S. and Kotonya, G., "An Integrated Framework for Change Management Impact Analysis", Proceedings for the 4th Australian Conference on Requirements Engineering, Sydney, Australia, September pp. 29-42, 1999.
- [8] Lock, S. and Kotonya, G., "Abstract: An Integrated Framework for Change Management Impact Analysis", Re-quireonautics Quarterly: The Newsletter of the Requirements Engineering Specialist Group of British Computer Society, Issue 18, January 2000.
- [9] 박지현, "소프트웨어 프로젝트 라인 공학에서 체계적인 요구사항 변경관리 방법", 학위논문, 부산대학교, 2005.
- [10] Sangim Ahn, Kiwon Chong, "A Feature-Oriented Requirements Tracing Method: A Study of Cost-benefit Analysis", The

International Conference on Hybrid Information Technology, pp. 611-616, 2006.

[11] Kang, K., Kim, S., Lee, J., Kim, K., Shin E, Huh, M., "FORM: A Feature-Oriented Reuse Method with Domain-Specific Reference Architectures", Annals of Software Engineering, pp. 143-163, 1998.

[12] Kwanwoo Lee, Kyo C. Kang, and JaeJoon Lee, "Concepts and Guidelines of Feature Modeling for Product Line Software Engineering", Proceedings of the Seventh Reuse Conference (ICSR7), Austin, U.S.A., pp. 15-19, 2002, Springer Lecture Notes in Computer Science Vol. 2319, pp. 62-77, 2002.

## 저자 소개



안상임

(E-Mail : siahn69@ssu.ac.kr)

송실대학교 대학원 컴퓨터학과

(Department of Computing, Soongsil University)

1992.

송실대학교 전자계산학과 졸업

2004.

연세대학교 공학대학원 컴퓨터공학 공학석사

2005 ~ 현재

송실대학교 대학원 컴퓨터공학과 박사과정 재학중

관심분야

요구공학, 소프트웨어개발방법론, 품질보증



정기원

(E-Mail : chong@ssu.ac.kr)

송실대학교 컴퓨터학부

(School of Computing, Soongsil University)

1967.

서울대학교 공과대학 전기공학과 졸업

1981.

미국 알라바마 주립대학(헨츠빌) 전산학과 공학석사

1983.

미국 텍사스 주립대학(일링턴) 전산학과 공학박사

1990 ~ 현재

송실대학교 컴퓨터학부 정교수

관심분야

소프트웨어 개발 프로세스 방법론, 모델링, 실시간 응용, 전자거래, 정보시스템 개발 및 평가