개발방법론의 요구사항 관리를 개선하기 위한 요구사항 관리 프로세스

신 종 철† · 구 연 설††

요 약

정보시스템 개발과정에서 발생하는 절반 이상의 결점이나 오류가 요구사항과 관련 있는 것으로 알려져 있다[14]. 그러나 개발방법론에서는 요구사항의 도출과 분석작업을 분석단계의 활동으로 정의하고 있어, 실제 프로젝트에서 계획단계부터 부분적으로 시작되는 요구사항의 수집 및 도출작업을 충분하게 지원하기 어렵다. 그리고 개발 초기단계의 요구사항들은 통상 불완전하고, 개발기간 동안 문제에 대한 개발자의 이해는 지속적으로 변화하며, 사용자에게 시스템이 가시화되거나 환경이 변하면 새로운 요구사항이 생겨나게 되므로, 요구사항의 변경을 지원할 수 있는 방안이 필요하다. 이 논문에서는 이러한 문제점을 해결하기 위하여 개발방법론에 통합 적용이 가능한 요구사항 관리 프로세스를 제안하여, 요구사항을 점진적으로 상세화하고, 요구사항의 관리범위를 전체 개발생명주기로 확대하며, 설계단계 이후에도 요구사항의 변경을 지원할 수 있는 방안을 제시한다. 그리고 제안 요구사항 관리 프로세스를 기존의 요구사항 관리방법들과 비교 평가하고, 대표적인 개발방법론의 하나인 마르미방법론에 대한 적용 검토를 통하여 개발작업 단계별 개선효과를 분석한다.

A Requirements Management Process to improve the Requirements Management of Development Methodologies

Jong Cheol Shin† · Yeon Seol Koo††

ABSTRACT

More than 50% of defects and errors in the software development projects are related to the requirements[14]. But the development methodologies can not provide enough supports to the requirements management and changes. In the real world, the requirements keep changing as the requirements at the early development phase are usually incomplete, the developer's understanding of the problem will be constantly changed during the software process, and new requirements may emerge with the organizational and environmental changes. In this paper, a requirements management process that can be integrated into the development methodologies is proposed to extend the requirements management over the whole development life-cycle and to support the requirements changes at the design/implementation/test phases. To evaluate the proposed process, it was compared to the existing requirements management processes and models, and examined through a integration test with a widely used development methodology 'MaRMI' to show the expected improvements at each development phase.

키워드 : 요구사항(Requirements), 요구공학(Requirements Engineering), 요구사항 관리(Requirements Management), 개발생명주기(Development Life-cycle), 개발방법론(Development Methodologies), 프로세스 평가(Process Assessment)

1. 서 론

폭포수 모형을 기반으로 하는 전통적 개발방법론에서는 시스템 요구사항과 소프트웨어 요구사항의 도출과 분석작업이 요구분석 단계에서 실시되고, 산출물은 다음의 설계작업을 위한 중간과정으로 사용된다. 그러나 실제 프로젝트의 수행과정에서는 제안요청서 단계에서부터 요구사항의 수집이 시작되고, 설계단계 이후에도 요구사항은 지속적으로 변화할 수 있다. 개발 초기단계의 요구사항들은 통상 불완전하거나 불명확하며, 개발기간 동안 문제에 대한 개발자의 이해는 지속적으로 변화하고, 개발이 진행되는 도중에도 정보기술의 발전이나 시장 환경의 변화로 요구사항의 변경이 불가피한 경우가 발생하기 때문이다. 따라서 요구사항의 관리범위를 요구분석 단계로 한정하지 않고 전체 개발 생명주기로 확대하며 지속적으로 요구사항의 변경을 지원하는 방안이 필요하다.

대표적인 개발방법론으로는 국제적 표준으로 사용되는 ISO/IEC 12207 소프트웨어 생명주기 공정표준(Software Life Cycle Process Standard)과 미 국방성의 정보시스템 개발 표

† 정 회 원 : 송우아이엔티(주) 기술연구소장
†† 정 회 원 : 충북대학교 컴퓨터과학과 교수
논문접수 : 2001년 7월 10일, 심사완료 : 2001년 12월 5일

준으로 제정된 MIL-STD-498, 그리고 국내에서는 한국전산원의 '관리기법/1', 한국전자통신연구원의 '마르미', 정보공학방법론 등이 있다. 그리고 개발된 시스템의 품질이나 개발 프로세스의 성숙도를 평가하기 위한 기준으로 한국전산원의 '관리기법/1'을 바탕으로 한 감리지침[6]이나, CMM(Capability Maturity Model), SPICE(Software Process Improvement and capability dEtermination) 참조모형[9] 등이 널리 사용되고 있다. 그러나 이들 개발방법론과 평가기준에서는 요구사항 관련 작업과 활동이 분석단계로 한정되어 전체 개발생명주기에 걸친 요구사항의 관리와 변경을 지원하지 못하는 문제점이 있다.

그리고 요구사항 관리모델은 요구사항 자체의 관리와 품질 향상을 주요 목적으로 하기 때문에 개발방법론과의 통합이 미흡하고 프로젝트의 특성이나 규모에 따라 적용범위를 조정하기 어렵다. 사실 CASE 도구들이 소프트웨어 프로세스 개선에는 중요한 역할을 하였지만 생산성이나 품질의 향상에는 기대보다 별로 도움이 되지 못하였으며, 요구공학은 소프트웨어 설계보다 훨씬 더 가변적이기 때문에 규칙과 지침을 이용하여 정형화하는데는 한계가 있다[10]. 따라서 요구 공학적 관리기법이나 자동화된 요구사항 관리도구를 실제 프로젝트에 효율적으로 활용하기 위해서는 개발방법론 측면에서 필요한 활동과 산출물을 먼저 정의하는 것이 바람직하다.

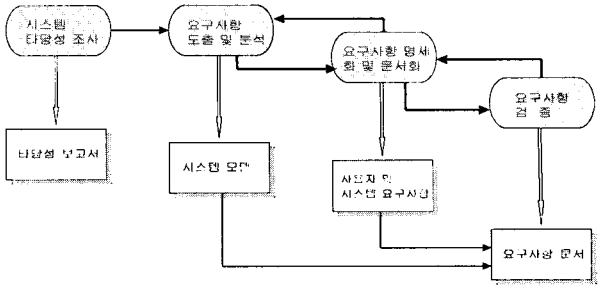
이 논문에서는 개발방법론에서 미흡한 요구사항 관리를 개선하기 위하여 요구사항의 관리범위를 전체 개발생명주기로 확대하고 지속적인 요구사항의 변경을 지원하며 개발방법론에 통합 적용이 가능한 요구사항 관리 프로세스를 제시하고, 그 개선 효과를 분석한다.

이 논문의 제2장에서는 요구공학 프로세스와 개발방법론에서의 요구사항 관리방법을 분석하고, 제3장에서는 개발방법론에 통합 적용할 수 있는 요구사항 관리 프로세스를 제시한다. 그리고 제4장에서는 제안 요구사항 관리 프로세스를 기존의 요구사항 관리방법들과 비교 평가하고, 마르미방법론에 대한 적용 검토를 통하여 개발작업 단계별 적용효과를 분석한다.

2. 요구사항 관리모델과 프로세스

2.1 요구공학 프로세스

요구공학은 개발될 시스템에 대한 요구사항 문서를 생성하고 유지하는 모든 활동을 포함하는 프로세스를 의미하며, 시스템 타당성 조사, 요구사항의 도출과 분석, 요구사항의 명세화 및 문서화, 그리고 요구사항 검증 등의 프로세스 활동으로 구성된다[11]. Ian Sommerville이 정의한 요구공학 프로세스의 흐름과 각 단계별 산출물은 (그림 1)과 같다[11].



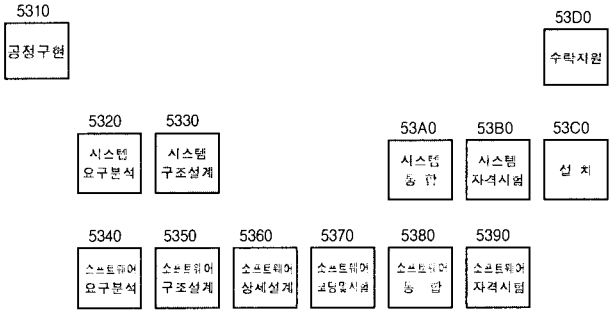
(그림 1) 요구공학 프로세스 흐름 및 산출물

(그림 1)에 나타난 요구공학 활동들은 요구사항의 도출, 문서화 및 검증과 관련되며, 지속적으로 변화하는 요구사항의 변경을 지원하기 위하여 요구사항 관리활동이 추가될 수 있다. 여기서 요구사항 관리활동은 요구사항의 변경을 이해하고 통제하는 프로세스를 의미한다.

요구사항 관리를 지원하기 위하여 몇 가지 자동화된 관리도구들이 상용화되어 있는데, 일반적으로 요구사항 관리도구는 요구사항의 저장, 변경관리, 추적성 관리기능을 제공한다[18]. 대부분의 상용화된 요구사항 관리도구들은 주로 요구사항 관리의 정보관리 측면, 즉 추적성과 조직화에 중점을 두고 있으며, 추적성은 하나의 요구사항과 다른 시스템 프로세스 요소(설계 컴포넌트, 사양서 등)와의 관계를 의미하고, 조직화는 기능 요구사항들의 그룹화, 키워드 할당, 요구사항의 속성, 관련 요구사항의 탐색 등을 포함한다. 이 외에도 보고서의 출력이나 다른 시스템 공학 도구와의 인터페이스 등도 포함될 수 있다[18].

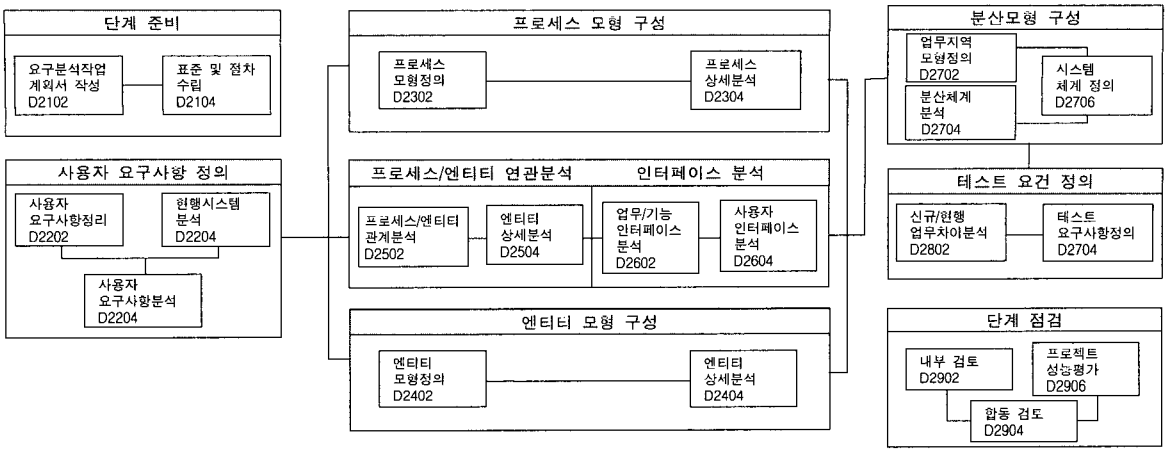
2.2 개발방법론의 요구사항 관리

ISO 12207에서는 소프트웨어의 생명주기동안 수행되는 여러 가지 활동들을 5개의 기본 공정(Primary Process)과 8개의 지원 공정(Supporting Process), 그리고 4개의 조직 공정(Organizational Process)로 구분하는데, 5개의 기본공정 중 개발공정에서 소프트웨어의 개발과 관련된 13개의 주요 활동들을 (그림 2)와 같이 정의한다[5].



(그림 2) ISO/IEC 12207 개발공정의 주요 활동

여기서 요구사항 분석활동은 시스템 요구분석(5320)과 소프트웨어 요구분석(5340)으로 나누어지고, 시스템 요구분석

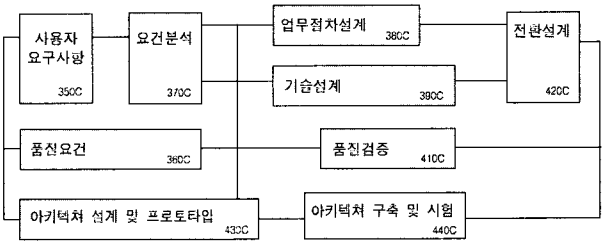


(그림 3) 마르미방법론의 요구분석단계 프로세스 흐름도

은 시스템 요구명세 작성(5321)과 시스템 요구사항 평가(5322)로 세분화되며 소프트웨어 요구분석은 소프트웨어 요구명세 작성(5341), 소프트웨어 요구사항 평가(5342), 합동검토 및 베이스라인 설정(5343)으로 세분화된다. 그리고 시스템 요구분석 활동은 시스템 구조설계 활동(5330)으로, 소프트웨어 요구분석 활동은 소프트웨어 구조설계 활동(5350)으로 변환된다.

마르미방법론에서는 요구분석 단계를 단계 준비, 사용자 요구사항 정의, 프로세스 모형 구성, 프로세스/엔티티 연관분석, 인터페이스 분석, 엔티티 모형구성, 분산 모형구성, 테스트 요건 정의, 단계 점검의 9개 활동으로 구분하는데, 이들 중에서 사용자의 요구사항을 도출하고 분석하는 작업은 사용자 요구사항 정의활동 단계에서 주로 이루어진다. 이 사용자 요구사항 정의활동은 사용자 요구사항 정리(D2202), 현행 시스템 분석(D2204), 사용자 요구사항 분석(D2206)의 3개 세부업무로 구분된다. 그리고 사용자 요구사항 정리(D2202)작업과 현행 시스템 분석(D2204)작업은 사용자 요구사항 분석(D2206)작업으로 통합되며, 사용자 요구사항 정의활동은 (그림 3)과 같이 프로세스 모형 구성, 프로세스/엔티티 연관 분석, 엔티티 모형 구성으로 변환된다[5].

관리기법/1에서는 프로젝트의 규모, 기술적 범위, 업무절차 등을 고려하여 정보계획 수립, 클라이언트/서버 시스템 개발, 호스트 시스템 개발, 패키지 시스템 개발, 소규모 프로젝트 개발, 고속 개발 등으로 개발경로를 구분하는데, 이 논문에서는 클라이언트/서버 개발단계가 대규모 프로젝트를 기준으로 전체 경로를 포함하고 있으므로 이를 기준으로 사용한다. 이 경우 분석, 설계 및 개발단계에 해당되는 세그먼트중에서 요구사항과 관련되는 활동은 사용자 요구사항(350C) 세그먼트에서 이루어지고, 이 사용자 요구사항 세그먼트는 작업흐름 및 조직 확인, 사용자 요구사항 파악, 현행설계 복구의 3개 타스크로 구분된다. 그리고 사용자 요구사항 세그먼트는 (그림 4)와 같이 요건분석(370C) 세그먼트로 변환된다[5].

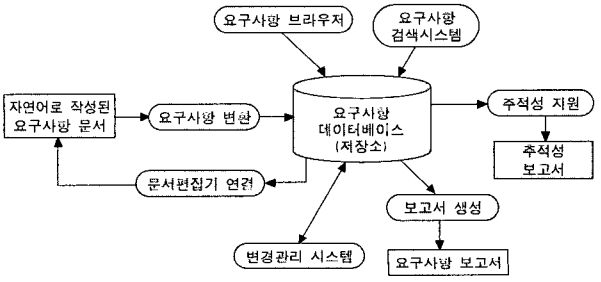


(그림 4) 관리기법/1의 분석/개발/설계 단계 프로세스 흐름도

정보공학방법론에서는 구조적 기법을 하나의 프로젝트 단위에 적용하지 않고 기업 전체 혹은 핵심 영역에 적용하기 때문에 다른 개발방법론과는 달리 요구분석활동이 별도로 구분되지 않고, 업무영역 분석과 시스템 설계 및 구현 단계에서 광범위하게 이루어진다[5]. 대부분의 요구사항 관련 활동들은 업무영역 분석과 시스템 설계 및 구현 단계에서 이루어지게 되는데, 업무영역 분석의 세부업무에는 예비 데이터 모델작성과 지속적인 정보 정련이, 그리고 시스템 구현 및 설계의 세부 업무에는 JRP(Joint Requirements Planning) 워크샵 실시 등이 있다.

2.3 요구사항 관리모델

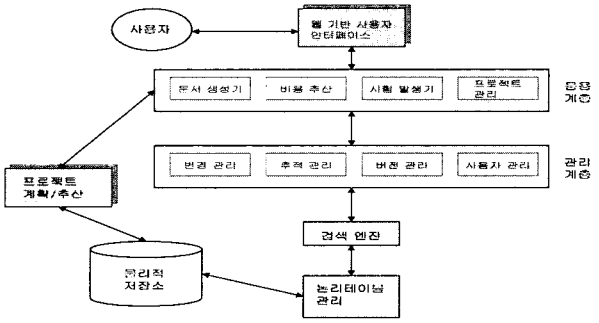
Gerald Kotonya와 Ian Sommerville이 제시한 요구사항 관리모델은 (그림 5)와 같이 구성되어 있는데[10], 자연어로 작성된 요구사항을 데이터베이스로 변환하고 문서 편집기



(그림 5) 요구사항 관리모델

와의 인터페이스를 제공하며, 저장소나 데이터베이스에 시스템 요구사항을 모으고, 이들 요구사항에 대한 정보를 검색하는 기능을 제공한다. 이 모델은 요구사항의 관리에 초점이 맞추어져 있으며, 특정한 요구공학 프로세스 모델에 국한되지 않는 일반적인 흐름을 나타내고, 특히 요구사항의 추출은 매우 제한적인 범위에서만 지원한다.

요구사항 관리모델을 웹 기반으로 확장한 '웹 기반 요구사항 관리모델'은 (그림 6)과 같이 사용자 인터페이스, 응용계층, 관리 계층, 저장소 및 프로젝트 관리부분으로 구성된다[1]. 이 모델은 요구사항의 수집 및 식별화, 시스템 요소의 추출 및 요구사항의 흐름 분석, 요구사항의 명세화, 요구사항 변경관리 및 변경 추적과 영향 분석, 버전 관리, 사용자 인터페이스 관리, 시스템 환경 분석 및 다른 도구와의 인터페이스, 표준화, 유지보수 및 문서화 기능 등을 포함하는데, 이 모델도 요구공학 프로세스 전반을 대상으로 하기보다는 분석단계에서 산출한 요구사항들을 효율적으로 관리하기 위한 관리적 측면의 모델이라 할 수 있다.



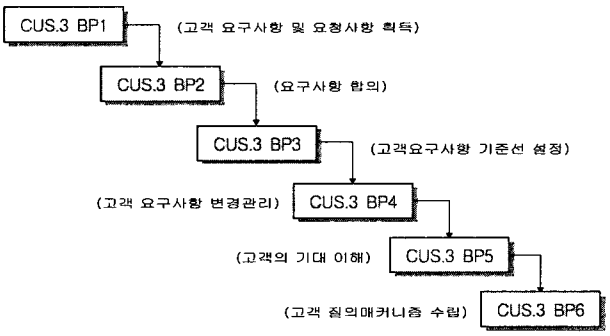
(그림 6) 웹 기반 요구사항 관리모델

2.4 SPICE 참조모형의 요구사항 관련 프로세스

SPICE(Software Process Improvement and capability dEtermination)로 불리는 ISO/IEC TR 2 15504는 9개의 표준으로 구성되어 있으며, 소프트웨어의 프로세스에 대한 계획, 관리, 감시, 통제, 개선을 위한 능력 심사와 프로세스 개선을 목적으로 한다[3]. 프로세스 심사의 기준이 되는 참조모형은 프로세스와 프로세스 능력의 2개 차원으로 구성되며, 프로세스 차원은 소프트웨어 프로세스를 평가하기 위해 프로세스 범주를 고객-공급자 프로세스(CUS), 공학 프로세스(ENG), 지원 프로세스(SUP), 관리 프로세스(MAN), 조직 프로세스(ORG) 등 5개로 구분되고, 각 프로세스 범주는 다시 세부적인 프로세스들로 세분화된다. 그리고 프로세스의 능력 차원은 Level 0~Level 5의 6단계 프로세스 능력 수준과 9개의 PA(Process Attribute)로 구성되어 있다[4].

세부 프로세스 중에서 요구사항과 직접 관련되는 프로세스는 CUS.3(요구사항 도출), ENG.1.1(시스템 요구사항 분석 및 설계), ENG.1.2(소프트웨어 요구사항 분석) 등이 있는데, ENG.1.1과 ENG.1.2는 공학 프로세스의 개발공정에 속하며, 개발방법론의 시스템 요구분석과 소프트웨어 요구분석 단계에 대응하여 주로 개발자에 대한 프로세스 평가를 위한 기본활동들을 정의한다.

CUS.3(요구사항 도출)은 계속적으로 발전하는 사용자의 필요와 요구를 수집, 처리, 추적하는 프로세스로, (그림 7)과 같이 고객 요구사항 및 요청사항 획득, 요구사항 합의, 고객 요구사항 기준선 설정, 고객 요구사항 변경관리, 고객의 기대 이해, 고객 질의 매커니즘 수립 등의 6가지 기본활동으로 구성되며, 요구사항의 변경에 대한 추적은 SUP.2(형상관리 프로세스)에, 그리고 고객과의 합동회의나 검토를 위한 의사소통은 SUP.6(합동검토 프로세스)에 포함된다. 이 프로세스는 전체 생명주기에 걸쳐 실시하며, 프로세스의 결과로 고객과의 지속적인 의사소통이 수립되고 합의된 고객 요구사항이 정의된다[7].



(그림 7) CUS.3(요구사항 도출) 기본활동 순서

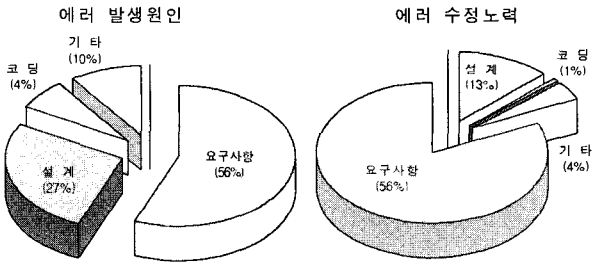
ENG.1.1(시스템 요구 분석 및 설계 프로세스)는 시스템에 대한 요구사항과 아키텍처를 정립하는 프로세스로, 시스템 요구사항 식별, 시스템 요구사항 분석, 시스템 아키텍처 기술, 요구사항 할당, 공표(Release) 전략의 개발, 시스템 요구사항 전달, 추적성 수립 등의 7가지 기본활동으로 구성되며, 프로세스의 결과로 문제 영역에 대한 우선 순위가 결정되고, 그 결과가 승인되며 개발과정에서 갱신된다. 그리고 ENG.1.2(소프트웨어 요구분석)는 소프트웨어 구성요소들에 대하여 사용자 요구사항을 수립하기 위한 프로세스로, 소프트웨어 요구사항 명세화, 운영환경의 영향 결정, 고객과 함께 요구사항 평가 및 확인, 소프트웨어 확인기준 개발, 공표(Release) 전략의 개발, 요구사항의 갱신, 소프트웨어 요구사항 전달, 추적성 수립 등의 8가지 기본활동으로 구성된다. 프로세스의 결과로 요구사항이 소프트웨어 구성요소와 구성요소간 인터페이스에 할당되며 정의된다.

2.5 문제점 및 개선의 필요성

현재 널리 적용되고 있는 전통적 개발방법론에서는 요구사항의 추출 및 명세화가 분석단계에서 이루어지고, 다음 단계인 설계 및 구현을 위한 기준으로 사용되므로, 한번 명세화된 요구사항들은 적어도 개발단계 내에는 변경되지 않

을 것이라는 가정을 바탕으로 한다. 그러나 현실적으로는 개발기간 중 요구사항이 끊임없이 변화하고 개발기간이 상대적으로 긴 프로젝트의 경우에는 요구사항 분석작업이 정확하게 수행되더라도 개발이 완료된 시점에서의 시스템은 이미 사용자의 요구를 벗어나는 사례가 발생하게 된다. 그리고 요구사항 관리모델과 프로세스들은 요구사항의 관리적 측면만 제시하거나 개발생명주기의 일부와 관련된 프로세스만 제시하여, 소프트웨어 개발생명주기 전체를 관리하는 개발방법론과의 통합 적용이 어렵다.

그런데 코딩단계에서 하나의 결함을 검출하고 수정하는데 소요되는 노력을 1 이라고 할 때, 요구사항단계의 노력은 0.1~0.2, 그리고 유지보수단계의 노력은 20 정도가 소요되는 것으로 알려져 있다[13]. 그리고 James Martin에 의하면 (그림 8)과 같이 결함의 발생원인 중에서 요구사항 결함이 전체의 56%나 되고 결함 수정노력의 82%가 요구사항 결함을 수정하는데 사용되며[15], Bohm이나 Dion에 의하면 전체 프로젝트 비용의 40%~50%가 이러한 결함 수정을 위한 재작업에 투입되고 있다. 따라서 전체 프로젝트 비용의 30%가 재작업에 사용되고, 재작업 비용의 70%가 요구사항에 기인한다고 가정하더라도, 요구사항 에러를 10%만 감소시키면 요구사항 관리에 투입되는 비용의 153%, 20%를 감소시키면 407%의 투자대비 효과를 기대할 수 있다[13].



(그림 8) 에러 발생 및 수정 노력 분포

따라서 개발방법론에서 미흡한 요구사항 관리를 보완하며 프로세스 심사과정에서 사용자의 필요와 요구를 수집, 처리, 추적하는 CUS.3(요구사항 도출) 프로세스를 충족시키고 개발 프로젝트의 전체적인 요구사항 결함을 감소시킬 수 있는 방안이 필요하며, 이러한 목적을 위하여 개발방법론에 통합 적용이 가능한 요구사항 관리 프로세스를 설정하여 요구사항 관리범위를 전체 개발생명주기로 확대하고, 개발 생명주기의 진행에 따라 요구사항의 점진적 상세화를 가능하게 하며, 설계단계 이후에도 지속적으로 요구사항의 변경을 지원할 수 있는 방안을 제시하고자 한다.

3. 제안 요구사항 관리프로세스

3.1 범위 및 전제조건

일반적으로 프로세스는 입력을 출력으로 변환하는 상호연관된 활동들의 집합을 의미하며[10], 절차와 방법뿐만 아니라 사람과 도구까지 포함하기도 한다[12]. 따라서 간단한 프로세스의 경우에는 상세한 절차를 규정하기도 하지만, 좀 더 복잡한 프로세스의 경우에는 이를 적용하는 사람의 개인적 배경이나 기술 수준, 적용 환경에 따라 적용 방법이나 결과가 달라질 수 있으므로 상위 수준의 절차나 결과물만 규정한다. 특히 개발방법론에서는 프로젝트의 규모와 성격에 따라 프로세스의 조정(customization)을 전제로 하기 때문에[5], 이 논문에서는 요구사항 관리 프로세스를 개략적인 단계별 절차와 활동, 그리고 산출물을 통하여 정의하고, 각 산출물의 세부적인 내용이나 작성방법 등은 포함하지 않는다.

요구사항의 관리는 개발 단계뿐만 아니라 향후 운영 및 유지보수 단계까지 시스템의 전체 생명주기와 연관된다. 그러나 이 논문에서는 적용 대상 개발방법론을 폭포수 모형을 기반으로 하는 전통적 개발방법론으로 한정하고, 요구사항의 관리 범위를 개발방법론을 적용하는 개발생명주기 즉, 개발계획 단계에서부터 설치 및 인도 단계로 한정하며, 개발계획 단계는 외주 개발의 경우를 감안하여 제안요청서 준비단계와 제안서 작성 단계로 세분화한다.

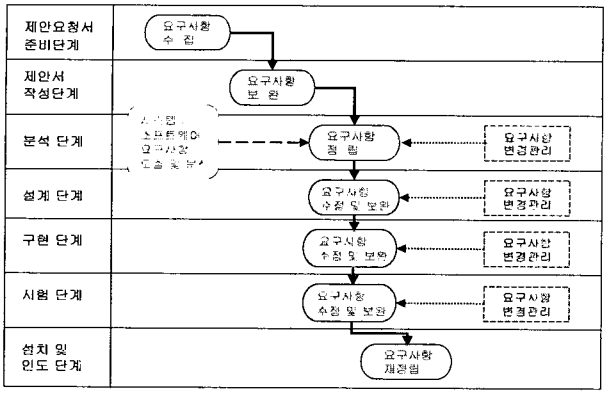
요구사항의 변경으로 인한 영향 범위의 분석이나 구현에 필요한 소요 자원(기간, 인력, 비용 등)의 산정 등을 포함한 요구사항의 변경관리 절차는 프로젝트 관리나 일반적인 변경관리(Change Management) 절차에 따라 이루어지는 것으로 가정한다. 따라서 이 논문에서는 이미 분석 및 평가 단계를 거쳐 승인이 이루어진 요구사항의 변경 요청을 기준으로 요구사항의 관리적 측면을 위한 변경관리로 프로세스의 범위를 한정한다.

3.2 요구사항 관리 절차

제 2.5 절에서 살펴 본 개발방법론에서의 요구사항 관리의 문제점을 개선하기 위하여, 먼저 제안 요구사항 관리프로세스가 충족시켜야 할 기본 요건들을 다음과 같이 정의한다.

- 제안요청서 단계에서부터 설치 및 인도까지 전체 개발생명주기 지원
- 개발작업의 진행에 따라 요구사항의 점진적인 상세화 지원
- 현재 널리 사용되고 있는 전통적 개발방법론과의 통합 적용 용이
- 특정한 개발방법론이나 CASE 도구, 혹은 요구사항 관리도구에 의존적이지 않음
- 프로젝트의 규모와 특성에 따라 적용범위의 조정 가능
- 항상 현재 시점에서 유효한 요구사항 목록 유지 지원
- 설계단계 이후의 요구사항 변경에 대한 체계적 지원
- 요구사항 변경에 따른 영향분석 지원

이러한 요건들을 만족시키기 위해서는 요구사항 관리 프로세스가 요구사항에 대한 전체 개발생명주기에 걸친 지속적인 관리와 요구사항의 변경을 체계적으로 지원하는 변경관리의 두 가지 측면을 지원해야 하므로, 먼저 앞 절의 범위 및 전제조건을 바탕으로 요구사항 관리 절차를 (그림 9)와 같이 정의한다.



(그림 9) 제안 요구사항 관리 절차

요구사항의 관리는 요구사항의 수집활동을 제안요청서 단계에서부터 시작하고 개발작업 단계가 진행됨에 따라 점진적으로 요구사항을 상세화할 수 있도록 지원하며, 요구사항의 변경을 반영하여 기존의 요구사항에 대한 보완이 가능하도록 관리 절차를 설정한다. 제안요청서 준비단계에서는 발주자가 사용(예정)자들의 의견을 수집하여 목표 시스템의 요구사항을 개략적으로 기술하고, 제안서 작성단계에서는 개발자가 제안요청서에 대한 이해를 바탕으로 구현될 목표 시스템의 요구사항을 좀 더 구체적으로 보완한다. 여기서 명세화된 요구사항들은 향후 발주자와 개발자 사이의 계약을 위한 기준으로 사용될 수 있다. 그리고 분석단계에서는 제안서 단계의 요구사항을 포함하여 목표 시스템이 만족시켜야 할 모든 요구사항들을 구체적으로 도출하고 분석하여 명세화하는데, 이렇게 정립된 요구사항들은 시스템 개발의 전체 생명주기에서 요구사항의 기준선으로 사용된다. 설계/구현/시험 단계에서는 요구사항의 변경이 발생하면 이를 반영하여 요구사항을 보완하고, 설치 및 인도 단계에서는 지금까지 발생한 모든 요구사항 변경을 반영하여 요구사항을 재정립하고 이를 발주자에게 인도한다.

요구사항의 변경관리는 분석단계에서부터 시작되며, 요구사항의 변경이력 관리와 변경으로 인한 영향 분석으로 구성된다. 분석단계에서는 기술적이나 비용문제 혹은 개발기간 등의 제약 등으로 제안서 단계의 요구사항에 대한 변경이 필요한 경우, 요구사항 변경관리를 실시하고 이를 요구사항 관리에 반영한다. 그리고 설계/구현/시험 단계에서는 각 단계의 진행과정에서 발생하는 요구사항의 변경요구를 관리하기 위하여 요구사항 변경관리를 실시하며, 이를 다시 요구사항 관리에 반영한다.

특히 개발작업의 한 단계가 완료된 후 다음 단계로 진행하기 전에 실시하는 검증활동이나 합동검토회의에서는 반드시 요구사항 목록을 점검하고 변경내역을 확인하여, 변경이 발생한 경우 이에 대한 영향을 분석하도록 한다. 이렇게 함으로써 개발자와 발주자가 요구사항 문서를 바탕으로 개발될 목표 시스템에 대한 공통의 이해를 향상시키고 변경의 범위가 커질 경우 이에 따른 프로젝트 변경을 결정하기 위한 기준으로 활용할 수 있다.

3.3 단계별 활동 및 산출물

앞 절에서 제안한 요구사항 관리 절차를 개발방법론에 적용하기 위한 단계별 활동과 산출물을 정의한다. 활동은 요구사항의 수집, 도출, 분석, 명세화, 보완 및 재정립 등으로 구분하며, 작성할 산출물은 요구사항 기술서, 요구사항 변경 이력서, 그리고 요구사항 변경영향 분석서로 구분하여 개발작업 단계별로 <표 1>과 같이 정의한다.

〈표 1〉 개발작업 단계별 활동 및 산출물

개발 단계	활 동	산 출 물	비 고
제안요청서 준비	요구사항 수집	요구사항 기술서	
제안서 작성	요구사항 보완	요구사항 기술서	
분 석	요구사항 도출, 분석, 명세화	요구사항 기술서 요구사항 변경 이력서 요구사항 변경영향 분석서	요구사항 기준선 설정
설 계	요구사항 변경관리	요구사항 기술서 요구사항 변경 이력서 요구사항 변경영향 분석서	
구 현	요구사항 변경관리	요구사항 기술서 요구사항 변경 이력서 요구사항 변경영향 분석서	
시 험	요구사항 변경관리	요구사항 기술서 요구사항 변경 이력서 요구사항 변경영향 분석서	
설치 및 인도	요구사항 재정립	요구사항 기술서	요구사항 기준선 재설정

요구사항 기술서는 하나의 요구사항을 단위로 작성하며, 모든 관련 담당자가 쉽게 이해할 수 있도록 자연어로 작성하고, 각각의 요구사항을 하나의 관리 단위로 설정한다. 그리고 요구사항 변경이력서는 변경 요구가 발생한 요구사항에 대하여 이를 체계적으로 관리할 수 있도록 추가, 수정 삭제로 구분하여 변경이력을 작성하며, 요구사항 변경영향 분석서는 추가, 변경, 삭제된 각각의 요구사항에 대하여 관련되는 요구사항들을 열거하고 변경으로 인하여 발생할 수 있는 영향범위를 기술적 측면과 관리적 측면으로 나누어 기술한다.

제안요청서 준비단계에는 사용(예정)자의 요구사항을 수집하여 요구사항 기술서를 작성한다. 이 단계에는 개발될 목표 시스템이 구체화되지 않은 상태이므로 요구사항들을 개략적으로 기술하되, 개발자가 제안서를 작성할 때 기준으

로 사용되므로 목표 시스템이 만족시켜야 할 기본적인 요구사항들은 반드시 포함되어야 한다.

제안서 작성단계에는 개발될 목표 시스템의 개념 모델을 구체화하기 위하여 제안요청서에서 발주자가 제시한 요구사항들을 보완하고 구체화하여 요구사항 기술서를 보완한다. 일반적으로 제안서는 발주자와 개발자간의 계약 체결의 기준으로 사용되므로, 개발자는 구현될 목표 시스템을 구체화할 수 있는 수준까지 요구사항 기술서를 보완한다.

분석단계에는 개발자가 요구분석 과정을 통하여 시스템이 만족시켜야 할 모든 요구사항들을 구체적으로 도출하고 분석하여 요구사항 기술서를 보완한다. 이 문서는 향후 개발 생명주기에서 요구사항의 기준선으로 사용되며, 다음 설계단계의 검증을 위한 기준이 된다. 그리고 요구사항 자체의 에러나 요구사항간의 상충, 설계나 구현상의 문제 등으로 인하여 제안서 작성단계의 요구사항에 대한 변경이 필요하거나 새로운 요구사항이 발생하게 되면 요구사항 변경 이력서와 이로 인하여 발생할 수 있는 영향을 분석하여 요구사항 변경영향 분석서를 작성한다.

설계단계에는 설계작업의 진행 도중에 발견되는 요구사항의 문제점이나 사용자의 변경요구 등을 반영하여, 필요한 경우 요구사항 변경 이력서와 요구사항 변경영향 분석서를 작성하고 이에 따라 요구사항 기술서를 보완한다.

구현단계에서도 설계단계와 마찬가지로 진행과정에서 발견되는 요구사항의 문제점이나 사용자의 변경요구 등을 반영하여, 필요한 경우 요구사항 변경 이력서와 요구사항 변경영향 분석서를 작성하고 이에 따라 요구사항 기술서를 보완한다.

시험단계에는 요구사항 기술서를 테스트를 위한 검증 기준으로 활용하게 되는데, 테스트 과정에서 문제점이나 보완사항이 발견되면 요구사항 기술서를 보완하고 요구사항 변경 이력서와 요구사항 변경영향 분석서를 작성한다. 특히 검수시험 시에는 지금까지 변경된 모든 요구사항을 반영하여 유효한 요구사항들을 모두 만족시키는지 확인함으로써 요구사항을 검수 기준으로 활용할 수 있다.

설치 및 인도단계에서 최종적으로 재정립되는 요구사항 목록은 개발된 시스템이 발주자에게 인도되어 운영 및 유지보수단계로 전환되는 시점을 기준으로 한 새로운 요구사항 기준선으로 사용된다.

4. 평가 및 검토

4.1 기존 요구사항 관리방법과의 비교 분석

이 논문에서 제안하는 요구사항 관리 프로세스를 평가하기 위하여 우선 요구사항의 관리범위 확대, 요구사항의 변경관리 지원, 그리고 요구사항 관리의 효율성 등 세 가지 측면에 대하여 기존의 요구사항 관리모델 및 개발방법론에서의 요구사항 처리절차와 비교 평가한다. 그런데 ISO 12207을 비롯한 기존의 개발방법론은 요구분석 단계에서 요구사항을 도출하고 분석하는 작업을 수행하지만 SPICE 참조모형은 시스템 및 소프트웨어 요구분석뿐만 아니라 고객-공급자 사이의 요구사항 관리까지 포함하고 있으므로 개발방법론 측면은 SPICE 참조모형을 포함한 확장된 개념을 사용한다.

여기서 요구사항의 관리범위 확대는 요구사항의 수집, 도출, 분석 및 보완을 프로젝트 계획단계에서부터 유지보수단계까지 전체 개발 생명주기에 걸쳐 실시할 수 있는지를 평가하며, 요구사항의 변경관리는 설계단계 이후에 발생하는 요구사항의 변경에 대한 체계적 지원 여부를 평가한다. 그리고 요구사항 관리의 효율성은 요구사항 자체의 품질향상과 다른 프로젝트 관리활동과의 연계에 대한 지원 여부를 평가한다. 이러한 세 가지 분야를 상세 항목으로 다시 세분화하여 제안 요구사항 관리 프로세스를 개발방법론 및 요구사항 관리모델과 비교 평가한 결과는 <표 2>와 같다.

〈표 2〉 개발방법론 및 요구사항 관리모델과의 비교 평가

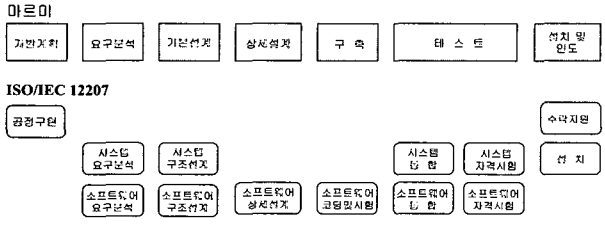
구 분		제안 요구사항 관리프로세스	개발 방법론	요구사항 관리모델
요구사항 관리 범위 확대	요구사항의 점진적 상세화 지원	○	×	○
	전체 생명주기에 걸친 요구사항 관리	○	△ (CUS.3)	○
	항상 현재 유효한 요구사항 파악 용이	○	×	○
	개발방법론과의 통합 지원	○	○	×
	구체적 절차 및 활동 명시	○	×	×
요구사항 변경 관리	설계단계 이후의 요구사항 변경 지원	○	×	○
	요구사항 변경관리 절차 제공	○	×	○
	요구사항 변경영향 분석 지원	○	×	×
요구사항 관리의 효율성	요구사항의 범주화 및 일관성 검증 지원	×	×	○
	요구사항의 집중 관리	○ (요구사항 기술서)	×	○ (요구사항 저장소)
	자동화 도구의 활용 지원	×	×	○
	프로젝트 관리 및 형상관리와의 연계	×	△ (SUP.2)	×
특 징		개발방법론에 요구사항 관리 통합 지원	SPICE 참조 모형 포함	개발방법론과 통합 지원 미흡

<표 2>에 의하면 이 논문에서 제안하는 요구사항 관리 프로세스는 개발방법론에서 미흡한 요구사항의 관리범위 확대와 요구사항 변경관리를 개선할 수 있으나, 요구사항 관리의 효율성 측면에 대한 개선효과는 크지 않은 것으로 평가할 수 있다. 그리고 기존의 요구사항 관리모델에 비하면 개발방법론과의 통합 지원과 요구사항 변경영향 분석에는 유리하지만, 요구사항 자체의 품질향상과 요구사항의 효

율적인 관리를 위한 지원방안이 미흡하여 실무 적용단계에서는 자동화 된 요구사항 관리도구를 사용하여 이를 보완하는 것이 바람직할 것으로 판단된다.

4.2 마르미방법론 적용 검토

제3장에서 제안한 요구사항 관리 프로세스를 대표적인 개발방법론의 하나인 마르미방법론에 적용하여 그 적용성을 평가한다. 마르미방법론에는 정보시스템 개발방법론인 마르미-D, 정보계획 수립방법론인 마르미-P, 객체지향 개발방법론인 마르미-Ⅱ, 컴포넌트기반 개발방법론인 마르미-Ⅲ 등이 있다. 그리고 적용 검토의 대상으로 선정된 마르미-D는 일반적인 경영정보시스템(MIS)을 주 대상으로 하는 전통적 개발방법론으로 (그림 10)과 같이 7단계로 구성되며, 이 7단계는 ISO/IEC 12207의 개발공정 13가지 활동(Activity)를 토대로 도출되었다[5].



(그림 10) 마르미방법론의 프로세스 단계

개발계획 단계는 단계준비, 기회분석 및 상위 요구사항 정의, 정보시스템 해결방안 설정 및 평가, 프로젝트활동 정의, 개발계획서 작성 및 승인, 단계점검의 6개 활동과 15개 작업으로 구성되어 있다. 이 단계에는 요구사항 관리를 위하여 기회분석 및 상위 요구사항 정의(D1200)활동의 사용자 업무파악(D1202)작업에서 사용(예정)자의 요구사항을 수집하고 상위 요구사항 정의(D1206)작업에서 상위 수준의 요구사항 명세서를 작성한다. 따라서 제안 프로세스의 적용을 위한 별도의 활동이나 작업은 추가하지 않고, 기존의 상위수준 요구사항 명세서를 보완하여 요구사항 기술서(개략)로 사용한다.

요구분석 단계는 단계준비, 사용자 요구사항 정의, 프로세스 모형 구성, 엔티티 모형 구성, 프로세스/엔티티 연관분석, 인터페이스 분석, 분산 모형 구성, 테스트 요건 정의, 단계점검의 9개 활동과 21개 작업으로 구성되어 있는데, 사용자 요구사항 정의(D2200)활동에서 요구사항을 도출, 분석, 명세화 하고 테스트 요건 정의(D2800)활동의 테스트 요구사항 정의(D2804)작업에서 테스트 요구사항을 정의하게 된다. 따라서 이 단계에서도 프로세스 적용을 위한 별도의 활동이나 작업은 추가하지 않고 기존의 요구사항 정의서와 테스트 요구사항 정의서를 보완하여 요구사항 기술서(보완)로 사용한다. 그러나 외주 개발의 경우에는 제안요청서 및 제안서 작성단계에서 요구사항이 이미 정리되고 개발자와 발주자간의 계약의 기준으로 사용되었으므로 요구사항 변경관리절차를 적용할 수도 있다.

기본설계 단계는 단계준비, 응용시스템 설계, 예비 분산설계, 분할 및 할당설계, 분산 어플리케이션 할당 설계, 아키텍처 설계, 업무 설계, 단계점검의 8개 활동과 24개 작업으로 구성되어 있으나, 마르미방법론에는 요구사항 관련 활동이나 작업이 정의되어 있지 않다. 따라서 이 단계에서는 요구사항 변경관리 활동과 산출물로 요구사항 기술서 (보완), 요구사항 변경 이력서, 요구사항 변경영향 분석서 등을 추가한다.

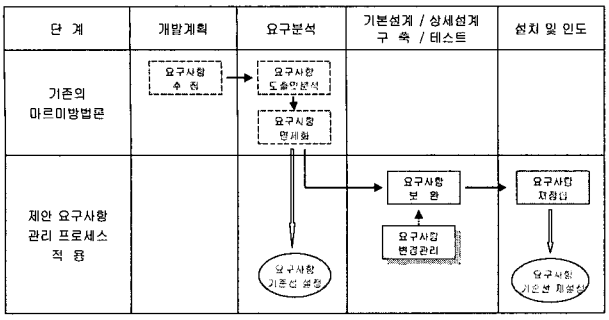
상세설계 단계는 단계준비, 데이터베이스 상세 설계, 프로그램 상세 설계, 테스트 및 이행 설계, 단계점검의 5개 활동과 11개 작업으로 구성되어 있으나, 마르미방법론에는 요구사항 관련 활동이나 작업이 정의되어 있지 않다. 따라서 이 단계에서는 요구사항 변경관리 활동과 산출물로 요구사항 기술서 (보완), 요구사항 변경 이력서, 요구사항 변경영향 분석서 등을 추가한다.

구축 단계는 단계준비, 프로그래밍, 단위 테스트, 지침서 및 교재 개발, 단계점검의 5개 활동과 14개 작업으로 구성되어 있으나, 마르미방법론에는 요구사항 관련 활동이나 작업이 정의되어 있지 않다. 따라서 이 단계에서는 요구사항 변경관리 활동과 산출물로 요구사항 기술서 (보완), 요구사항 변경 이력서, 요구사항 변경영향 분석서 등을 추가한다.

테스트 단계는 단계준비, 통합 테스트, 시스템 테스트, 사용자 승인 테스트, 단계점검의 5개 활동과 15개 작업으로 구성되어 있으나, 마르미방법론에는 요구사항 관련 활동이나 작업이 정의되어 있지 않다. 따라서 이 단계에서는 요구사항 변경관리 활동과 산출물로 요구사항 기술서 (보완), 요구사항 변경 이력서, 요구사항 변경영향 분석서 등을 추가한다.

설치 및 인도 단계는 단계준비, 사용자 교육 실시, 시스템 설치, 설치 후 관리, 단계점검의 5개 활동과 13개 작업으로 구성되어 있으나, 마르미방법론에는 요구사항 관련 활동이나 작업이 정의되어 있지 않다. 따라서 설치 후 관리(D7400)활동의 전달물 검토 및 갱신(D7404)작업에서 향후 유지보수를 위하여 지금까지의 모든 요구사항 변경을 정리하여 요구사항 기술서를 재정립하고 요구사항 기준선을 다시 설정하는 업무를 추가하고, 산출물로 요구사항 기술서 (재정립)를 추가한다.

지금까지 살펴 본 마르미방법론에 대한 제안 요구사항 관리 프로세스의 적용 결과는 제3.3절에서 제시한 단계별 활동 및 산출물의 적용과 거의 일치하며, 기존의 각 단계별 활동을 변경하지 않고 요구사항과 관련된 일부 활동과 산출물을 추가하거나 보완함으로써 쉽게 적용할 수 있음을 보여준다. 그리고 제안 요구사항 관리 프로세스를 적용한 마르미방법론의 요구사항 처리 절차는 (그림 11)과 같다.



(그림 11) 개선된 마르미방법론의 요구사항 처리 절차

4.3 개발작업 단계별 개선효과

일반적인 요구사항 관리 프로세스의 적용효과는 TBI 보고서[16]에 잘 나타나 있는데, 제 4.3 절의 마르미방법론에 대한 적용 검토를 바탕으로 이 논문에서 제안하는 요구사항 관리 프로세스의 기존 개발방법론에 대한 요구사항 관리의 개선효과를 개발작업 단계별로 분석하면 <표 3>과 같다.

〈표 3〉 개발작업 단계별 개선효과

단 계	개 선 효 과	비 고
제안요청서 준비단계	• 모든 관련자의 참여 유도 • 개발될 시스템에 대한 사용자의 이해 향상 • 발주자와 개발자간의 의사전달 향상	
제안서 작성단계	• 목표 시스템에 대한 개발자의 이해를 향상시켜, 보다 정확한 제안서 작성 가능 • 업무 범위의 명확화 • 발주자와 개발자간 계약의 기준 제공	
요구분석단계	• 이미 수집, 도출된 요구사항의 활용 • 사용자의 시스템에 대한 사전 이해 향상 • 개발될 목표 시스템의 요구사항 기준선 설정	
기본설계 및 상세설계 단계	• 설계 검증을 위한 기준 제공 • 테스트 사례 설정을 위한 기초 제공 • 요구사항 변경에 대한 개발자와 발주자 상호간 이해와 합의를 위한 바탕 제공	
구축 단계	• 구축 검증을 위한 기준 제공 • 요구사항 변경에 대한 개발자와 발주자 상호간 이해와 합의를 위한 바탕 제공	
테스트 단계	• 테스트를 위한 기준 제공 • 인수를 위한 검수 기준으로 활용	
설치 및 인도 단계	• 개발이 완료된 시스템의 요구사항 목록을 재정립하여 향후 유지보수를 위한 요구사항의 새로운 기준선 제공	

제안 요구사항 관리 프로세스는 기존의 개발방법론에서 미흡한 요구사항 관리를 보완함으로써 제안요청서 준비단계에서부터 설치 및 인도단계에 이르기까지 모든 관련 당사자의 참여를 유도하고, 각 개발단계의 완료시점에서 합동 검토나 검증을 위한 기준을 제공하며, 개발될 목표 시스템에 대한 이해를 증진시키고, 발주자와 개발자 사이의 원활한 의사소통을 위한 기반을 제공할 수 있다.

5. 결 론

기존의 개발방법론에서는 요구사항의 관리 범위가 요구분석 단계로 한정되고, 도출된 요구사항은 다음의 설계작업을 위한 중간 산출물로 정의되며, 설계 단계 이후에 발생하는 요구사항의 변경을 체계적으로 지원하지 못하는 문제점이 있다. 그리고 요구사항 관리모델은 요구사항 자체를 관리대상으로 하기 때문에 요구사항 관리의 정보관리 측면, 즉 추적성과 조직화 기능은 효율적으로 제공하지만[18], 소프트웨어 개발 생명주기를 지원하기 위한 개발방법론과의 통합 적용을 위한 지원이 부족하다. 따라서 이 논문에서는 요구사항의 관리범위를 전체 개발생명주기로 확대하고 설계단계 이후에 발생하는 요구사항의 변경을 지원할 수 있도록 개발방법론에 통합 적용이 가능한 요구사항 관리 프로세스를 제안하였다.

제안된 요구사항 관리 프로세스에 대하여 요구사항 관리 모델 및 개발방법론(SPICE 참조모형 포함)과 요구사항의 관리범위 확대, 요구사항의 변경관리, 요구사항 관리의 효율성 측면을 기준으로 비교 평가하고, 대표적인 개발방법론의 하나인 마르미방법론에 대한 적용 검토를 통하여 개발작업의 각 단계별 개선효과를 분석하였다. 이 논문에서 제안한 요구사항 관리 프로세스는 요구사항의 관리를 개발방법론에 통합 적용이 가능하여, 목표 시스템이 구체화됨에 따라 요구사항을 점진적으로 상세화하고, 요구사항의 관리범위를 전체 개발생명주기로 확대하며, 설계단계 이후에도 요구사항의 변경이 발생하면 이를 체계적으로 관리하고 영향을 분석할 수 있는 장점이 있다.

그러나 이 논문의 연구범위에는 요구사항의 범주화와 요구사항 상호간의 상충이나 일관성 검증 등 요구사항 자체의 품질을 향상시킬 수 있는 방안이 포함되지 않았으며, 요구사항 변경으로 인한 영향범위 분석 및 소요 자원의 산정 등 요구사항 변경관리기법, 프로젝트 관리 및 형상관리 등 기존의 프로젝트 관리활동과의 연계방안 등에 대한 연구와, 최근 프로젝트에 적용이 확산되고 있는 객체지향 개발방법론이나 컴포넌트 기반 개발방법론과 관련된 연구가 미흡하여 이들에 대한 추가적인 연구가 필요하다.

참 고 문 헌

[1] 이원우, 황만수, 박수용, 류성열, "*요구사항 관리를 위한 웹 기반 모델 설계*", 소프트공학회지, 제11권 제4호, pp.35-45, 1998.

[2] 이원우, 박수용, 류성열, "*객체지향 어플리케이션 개발을 위한 UML기반의 요구공학 프로세스*", 제1회 한국소프트웨어공학 학술대회, pp.165-172, 1999.

[3] 정호원, 황선명, "*소프트웨어 프로세스 심사의 이해 : SPICE를 중심으로*", 정보과학회지, 제17권 제1호, pp.6-12, 1999.

[4] 정창신, 송정범, 이종무, "*SPICE 심사사례*", 정보과학회지, 제17권 제1호, pp.13-22, 1999.

[5] NCA II-AUER-97095, *시스템 개발방법론 적용기준에 관한 연구*, 한국전산원, 1997.

[6] NCA II-RER-97081, *정보시스템 분석 감리지침 연구*, 한국전산원, 1997.
[7] 이덕형, 배경열, 이남용, "*소프트웨어 프로세스*", 도서출판 라메르정보기술(주), 서울, 2000.
[8] 정기원, 윤창섭, 김태현, "*소프트웨어 프로세스와 품질*", 홍릉과학출판사, 서울, 1997.
[9] ISO/IEC TR 2 15504-2 : 1998(E), *Information Technology -Software Process Assessment-Part 2 : A Reference Model for Processes and Process Capability*, ISO/IEC JTC 1/SC 7/WG 10, Jan. 1998.
[10] Gerald Kotonya and Ian Sommerville, "*Requirements Engineering Process and Techniques*," John Wiley & Sons, 1998.
[11] Ian Sommerville, "*Software Engineering*," 6th Edition, Addison-Wesley, 2001.
[12] Ralph R. Young, "*Effective Requirements Practices*," Addison-Wesley, 2001.
[13] Dean Leffingwell, "*Calculating Your Return on Investment from More Effective Require-ments Management*," Rational Software Corporation, http : //www.rational.com/products/reqpro/whitepapers.jsp, 1997.
[14] Larry Boldt, "*Managing Requirements at the Object Level*," Technology Builders Inc, http : //www.tbi.com/products/asq/.
[15] Gary Mogyorodi, "*What is Requirements-based Testing?*," Technology Builder Inc, http : //www.tbi.com/products/asq/.
[16] Doc. # : 01 WP0598, "*Requirements Management : The Foundation for Solving Common Development Challenges*," Technology Builder Inc., http : //www.tbi.com/products/asq/.
[17] Don Young, Sr., "*Integrating Requirements Management and Project Management*," QSS Inc, http : //www.telelogic.com/doors/.
[18] INCOSE Tools Survey : *Requirements Management (RM) Tools*, http : //www.incose.org/tools/tooltax.htm, 2001.

신 종 철

e-Mail : sis@songwoo.co.kr
1976년 서울대학교 전기과 졸업
1994년 연세대학교 산업대학원 (전자계산 전공) 졸업
2000년 충북대학교 전자계산학과 박사과정 수료
1978년~1992년 현대엔지니어링 전산실/정보사업부
1992년~2000년 (주)송우정보 대표이사
2000년~현재 송우아이엔티(주) 기술연구소장
관심분야 : 개발방법론, 요구공학, 정보시스템 감리

구 연 설

e-mail : yskoo@cbucc.chungbuk.ac.kr
1964년 청주대학교 졸업
1975년 성균관대학교 경영행정대학원 전자자료처리 전공(경영학석사)
1981년 동국대학교 대학원 통계학 전공 (이학석사)
1988년 광운대학교 대학원 전자계산학 전공(이학박사)
1994년~1995년 한국정보과학회 부회장
1979년~현재 충북대학교 컴퓨터과학과 교수
관심분야 : 객체지향 테스팅, 품질관리, 정보 검색, 전자상거래.