

웹 기반의 요구사항 관리를 위한 프로세스 모델 설계 (A Design of Process Model for Web-based Requirements Management)

황 만 수(Man-Soo Hwang)¹⁾ 민 태 홍(Tae-Hong Min)²⁾

요 약

소프트웨어 개발환경이 대형화, 복잡화됨에 따라 개발 생명주기의 각 단계에서 발생하는 요구사항은 기술적, 관리적 측면에서 많은 영향을 받게 되었다. 따라서 본 연구에서는 요구공학과 CMM에서 제시하는 활동을 기반으로 완전성과 일치성을 가진 안정적인 요구사항의 생성 및 관리 등을 포함하는 총체적인 프로세스와 활동에 대한 공학적 접근을 제시하여 전체 소프트웨어 개발비용과 위험부담을 경감시키며 품질향상을 이룰 수 있도록 한다.

ABSTRACT

As the Software development condition is changed into enlargement and complex, requirements generated in each step of Software development life cycle has important effects on the technique and management area.

In this thesis, We propose so engineering approach for effective and systematic process and activities including Requirements acquisition and management based on activities proposed with Requirements Engineering and CMM that my special emphasis is putted on Software quality improvement through the reduction of software development cost and danger.

1. 서론

소프트웨어 개발환경이 대형화, 복잡화됨에 따라 개발 생명주기의 각 단계에서 발생하는 요구사항은 기술적, 관리적 측면에서 많은 영향을 받게 되었다.

특히 소프트웨어 개발 생명주기의 초기단계에서 요구사항에 대한 잘못된 이해나 분석, 다른 경

험과 의미를 가지는 다양한 그룹의 영역에 대한 충분한 이해와 관리의 부재 및 사용자와 개발자에 의한 지속적인 변경요구는 소프트웨어 프로젝트의 성공에 대해 중요한 요소로 나타나고 있다. 이러한 결과는 전체적인 소프트웨어의 완전성과 성능 등에 커다란 영향을 주고 생명주기의 다른 단계에서 발생하는 에러에 비하여 유지보수에 많은 비용과 노력을 요구한다.

1) 정회원 : 신홍대학 컴퓨터정보계열 부교수

2) 정회원 : 인하공업전문대학 컴퓨터정보공학부 교수

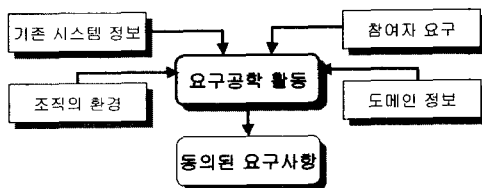
※ 본 연구는 인하공전 교내 연구비 지원에 의해 수행되었음.

따라서 본 연구에서는 이러한 문제점을 해결하기 위하여 요구사항 관리와 관련된 프로세스와 활동을 요구공학과 CMM을 기반으로 요구사항의 생성, 관리 및 유지 등을 포함하는 모든 총체적인 활동과 원칙에 대한 공학적 접근을 제시하여 전체 소프트웨어 개발비용과 위험부담을 경감시키며 품질향상을 이루고 소프트웨어 개발 프로세스의 이후 단계에 충실한 기초를 제공할 수 있도록 일관된 요구사항 관리를 위한 프로세스를 구현하는데 목표를 둔다.

2. 관련연구

2.1 요구공학

요구공학은 [그림 1]과 같이 요구사항 관리에 관한 모든 활동과 원칙에 대한 체계적이고 총괄적인 공학적 접근 방법이다. 즉, 참여자들이 제시하는 다양한 형태의 추상적인 관련 정보로부터 정확하고 의미있는 요구사항을 획득하고, 사용자 요구를 만족시키기 위하여 획득된 요구사항에 대한 완전성과 일관성에 대한 분석을 수행하며, 제안된 시스템 기능의 신뢰성 확보를 위하여 서비스와 기능 그리고 제약사항을 명세한 후, 명세 문서에 대한 일치성, 완전성, 현실성 등의 검증을 수행한다. 또한 요구사항 진화에 따른 변경관리와 추적관리 등을 위하여 수행되는 모든 생명주기 프로세스와 이를 지원하는 행위를 포함한다[1][2].



[그림 1] 요구공학 활동

[Fig. 1] Requirements Engineering Activities

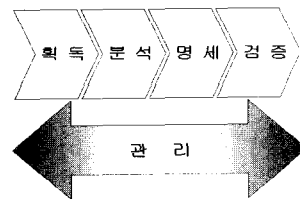
2.2 CMM의 요구사항 관리

CMM의 성숙도 레벨 2에서는 첫 번째 단계로 요구사항 관리에 대한 평가를 수행한다. CMM에서 제안하는 요구사항 관리를 위한 활동은 다음과 같다[3][6].

- ① 소프트웨어 요구사항은 문서화되어야 한다.
- ② 소프트웨어 요구사항의 명세는 프로젝트의 다른 요소들과의 관계를 정의한다.
- ③ 기능적 요구사항, 비기능적 요구사항 및 인증 기준을 정의하고 문서화한다.
- ④ 요구사항 관리를 위하여 적절한 자원과 비용이 요구된다.
- ⑤ 소프트웨어 요구사항은 개발과 관리를 위한 기준선 설정과 제어가 필요하다.
- ⑥ 요구사항에 대한 변경은 검토되고 프로젝트 계획에 포함되어야 한다.
- ⑦ 각각의 요구사항에 대한 상태를 포함하여 측정이 수행되어야 한다.

3. 요구사항 관리 프로세스

요구사항 관리 프로세스는 전체 5단계로 나누어져 있으며, 각 단계는 아래의 [그림 2]와 같다 [2][14][4].



[그림 2] 요구사항 관리 프로세스

[Fig. 2] Requirements Management process

3.1 획득 단계

고객의 문제가 무엇인지를 파악하여, 문제로부터 기인되는 고객의 요구사항들을 추출하는 단계이다. 또한 요구사항의 추출 이전에 현재 시작하고자 하는 시스템이 과연 현실적으로 구축 가능한지 혹은 현재의 문제점을 해결할 수 있는 솔루션이 될 수 있는지에 대한 검토가 필요하며, 이를 토대로 요구사항을 어떤 식으로 관리할 지에 대한 계획이 수립되어야 한다. 이러한 사전 준비가 완료되고 나면 실제로 여러 가지 추출기법들을 프로젝트의 성격에 맞게 운영함으로써, 정확한 고객의 요구사항을 추출할 수 있다. 이 단계에서의 활동과 수행 타스크는 <표 1>과 같다.

<표 1> 획득단계의 활동
(Table 1) Elicitation Activity

단계	액티비티	태스크
문제분석	문제분석	현 시스템의 문제점들을 파악한다. 프로젝트 관련자와 최종사용자를 식별한다.
	시스템구현 가능성분석	시스템의 목적을 정확히 설정한다. 시스템구축 가능성, 효과를 분석한다.
획득	요구사항 관리계획 수립	요구사항 타입을 정의한다. 요구사항 타입별 추적성을 설정한다. 요구사항 타입별 특성을 정의한다. 정의된 요구사항 관리체계를 도구에 매핑한다.
	시스템 환경정의	시스템 사용자조직의 특성을 파악 시스템구축 관련 제약사항을 식별
	공통업무 용어정의	업무 용어사전을 작성한다.
초기 요구사항 추출	요구사항을 추출할 대상을 결정한다. 초기 요구사항을 수집한다.	

3.2 분석단계

두 번째 분석 단계에서는 요구사항 획득 단계를 통해 수집된 초기 요구사항들을 분류하여 이미 정의되어 있는 요구사항 타입 중 어디에 해당되는지를 파악하여 할당하고, 각각에 알맞은 특성들을 부여함으로써 서로간의 연관관계를 파악한다. 이런 작업을 통하여 구축해야 할 시스템의 모습을 유스케이스 모델링 방법을 통하여 표현한다. 유스

케이스 모델링은 시스템의 기능적인 요구사항을 처리하며, 나머지 비기능적인 요구사항들은 별도의 문서에 선언적인 문장들로 명세한다. 또한 고객과의 원활한 의사소통과 시스템간의 효율적인 인터페이스를 위해 고객과 시스템, 시스템과 타시스템간의 상호작용을 담당하는 영역클래스(Boundary Class)에 대한 간략한 명세서를 작성하여 이해를 돕도록 한다.

<표 2> 분석단계의 활동
(Table 2) Analysis Activity

단계	액티비티	태스크
요구사항 분류 및 우선순위 식별	요구사항 분류 및 우선순위 식별	액터를 식별한다. 식별된 초기 요구사항별로 각 특성을 식별한다. 초기요구사항을 특성 기반으로 분류하고 정제해 나간다. 기능적인 요구사항들은 유스케이스 모델로 매핑한다.
	분석	유스케이스와 비기능적 요구사항 들의 우선순위를 식별한다. 개략적인 유스케이스 명세서를 작성한다.
요구사항 분석 모델 확립	요구사항 분석 모델 확립	유스케이스를 재구성한다. 정제된 유스케이스를 상세화한다. 비기능적 요구사항들을 상세화한다. 요구사항 베이스라인을 설정한다.
	인터페이스 분석	유스케이스 명세서를 기반으로 영역 클래스를 식별한다. 영역클래스 명세서를 작성한다.

3.3 명세 단계

세 번째 명세 단계에서는 고객으로부터 추출하여 정형화된 형태로 분석한 요구사항들을 일관성 있게 문서화하는 단계로, 본 논문에서는 요구사항 자체에 대한 명세서로 소프트웨어 요구사항 명세(SRS : Software Requirement Specification)를 작성한다.

소프트웨어 요구사항 명세문서는 프로젝트 전체에 걸쳐서 시스템의 구축을 제어할 수 있는 문서로 모든 프로젝트 개발 팀과 관련 있는 문서이다. 각각 참여자별 관점에서 본 소프트웨어 요구사항 명세문서의 사용 목적은 아래와 같다.

- 요구사항 분석가 : 실제 소프트웨어 요구사항 명세 문서를 작성하는 역할을 한다.
- 설계자 : 클래스의 역할과 오퍼레이션, 속성 등을 정의할 때나 클래스를 구현 환경으로 적용시킬 때 소프트웨어 요구사항 명세 문서를 근거로 작업하게 된다.
- 운영자 : 클래스 구현때 참고자료로 사용한다.
- 프로젝트 관리자 : 반복 계획 시의 참고자료로 사용한다.
- 테스트 시험자 : 시스템의 성능 검증시의 참고 자료로 사용한다.

〈표 3〉 명세단계의 활동
(Table 3) Specification Activities

단계	액티비티	태스크
명세	요구사항 분류 및 우선순위 식별	초기 요구사항 명세서를 작성한다. 정제된 요구사항들을 수집한다. 소프트웨어 요구사항명세서를 작성한다.

〈표 4〉 검증 단계의 활동
(Table 4) Validation Activities

단계	액티비티	태스크
검증	요구사항 검토팀 구성	요구사항 검토팀을 구성한다.
	정제된 요구사항 검증	정제된 요구사항의 타당성을 검증한다. 요구사항간의 추적성을 검토한다. 요구사항의 결함을 보완한다. 검토/보완된 요구사항에 대해 고객의 동의를 획득한다. 업무용어 사전을 검증한다.
	요구사항 관련 문서 검증	초기 요구사항 문서를 검증한다. 유스 케이스 모델을 검증한다. 비가능적 요구사항 명세서를 검증한다. 소프트웨어 요구사항 명세서를 검증한다.

3.4 명세 단계

네 번째 검증 단계에서는 명세된 요구사항이 적절한 소스로부터 추출되어 정확하게 분석되었는지를 검증함과 동시에 이러한 요구사항이 형식화된 요구사항 정의서에서 제대로 잘 표현되고 있는지를 검증한다. 또한 요구사항을 획득, 분석, 명세하고 관리하는 일련의 과정상에 문제점이 없는지를 검증하는 단계이다[9].

3.5 관리 단계

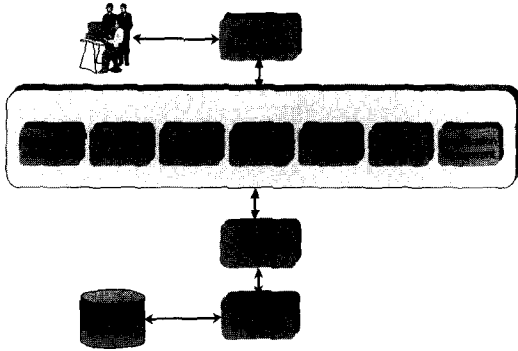
마지막으로 관리 단계에서는 프로젝트 생명주기를 통하여 개발 관련자로부터 제시되는 변경요구나 개발 중에 발생하는 문제보고에 의해 시스템 요구사항의 변경요소를 식별 및 정의하고, 변경된 요구사항을 추적하여 영향분석을 하며, 또한 변경된 형상과 상태를 레파지토리에 저장하고 관리하는 활동과 태스크를 정의함으로써, 시스템 요소의 완전성과 정확성을 증명하는 관리적, 기술적 절차를 수행한다[10].

〈표 5〉 관리 단계의 활동
(Table 5) Management Activities

단계	액티비티	태스크
관리	요구사항 변경관리	변경관리 정책을 정의한다. 각 요구사항에 유일한 식별자를 할당한다.
	요구사항 추적관리	전역 시스템 요구사항을 식별한다. 휘발성 요구사항을 관리한다. 요구사항 추적 정책을 정의한다. 요구사항 추적 매뉴얼을 작성한다.
	요구사항 형상관리	요구사항 변경 영향 분석을 수행한다. 요구사항 데이터베이스를 구축한다. 변경내용을 데이터베이스에 저장한다. 거부된 요구사항을 기록한다.

4. 웹기반의 요구사항 관리 도구

웹 기반의 요구사항의 획득 및 관리를 위한 WRMS(Web-Based Requirements Management System)는 요구사항 및 관련된 여러 가지 정보를 저장하기 위한 DBMS(Database Management System), 웹상에서 동적으로 입력되는 내용들을 처리하기 위한 어플리케이션 서버(Application Server), 실제로 요구사항을 관리하게 될 코어 어플리케이션(Core Application), 사용자와 상호작용 하게 되는 웹 서버(Web server)로 구성된다. 여기에 제시되는 WRMS의 개요적 구조(Overall Architecture)는 [그림 3]과 같다.



[그림 3] WRMS 체계도
[Fig. 3] WRMS System Architecture

사용자들은 웹 기반의 사용자 인터페이스를 통하여 요구사항들을 입력하게 되고 입력된 요구사항으로 본 도구는 다음과 같은 기능을 수행하게 된다.

(1) 웹기반 환경에서의 요구사항 추출

사용자가 원하는 시스템 요구를 충족시키기 위한 요구사항의 속성을 추출하고 각각의 추출된 요구사항들에 번호를 부과하고 계층적 입출력을 하여, 효과적인 요구사항 관리를 지원한다.

(2) 추출된 요구사항을 재구성

개개의 추출된 요구사항들을 형식기준에 맞게 구성한다. 분석가는 사용자가 입력한 요구사항들을 비교, 분석하여 반복, 충돌, 불일치 등을 제거하고, 요구사항들을 주제별로 분류한다.

(3) 요구사항들 간의 연계성 지원

주제별로 분류된 요구사항들 간의 유사도 계산을 통한 연계성을 유지한다.

(4) 요구사항들의 변경관리

새로운 요구사항의 추가, 기존 요구사항의 변경 및 삭제로 인해 파급되는 영향을 분석하고 변경되는 사항에 대한 모든 기록을 보관, 관리한다.

(5) 사용자 보안관리

요구사항 문서들에 관한 보안 유지를 위해 각 사용자들에 대한 접근 권한을 유지 관리한다.

(6) 명세화 관리

기술된 요구사항들을 표준형식에 근거하여 작성, 관리한다.

(7) 버전 관리

요구사항들의 문서버전을 관리한다.

(8) 변경영향 분석기능

요구사항의 변경으로 인한 영향 분석을 한다.

본 도구는 UML의 방식에 의해 분석, 설계되어, 시스템 설계도도 UML에서 정의하는 Use Case Diagram, Sequence Diagram, Class Diagram 등의 다이어그램으로 구성된다. Use Case Diagram은 시스템에서 제공하는 기능들과 외부와의 상호 작용을 나타낸다. Sequence Diagram에서는 세부적인 기능들 간의 메시지 전달과정을 나타낸다. 또한 Class Diagram에서는 클래스와 클래스들 간의 관계를 보여준다. 이들 세 다이어그램들로부터 상위 단계의 기본 개념에서부터, 하위 단계의 세부적인 처리부분까지 알 수 있다[13].

5. 결론

본 연구에서는 기존의 요구사항 관리가 분석과 문서의 생성에 관련되어 전체 개발 공정에 대한 해결책이 미흡하였던 문제를 해결하기 위하여, 소프트웨어 개발비용과 위험부담을 경감시키며 품질 향상을 이룰 수 있도록 요구공학 관리 프로세스와 CMM에서 제시하는 활동을 기반으로 안정적인 요구사항의 생성 및 변경에 대한 활동과 태스크를 제시하였다.

본 논문에서 제시하는 프로세스를 통하여 체계적인 요구사항 관리에 의한 소프트웨어 개발 생산성의 향상, 안정성있는 요구사항 관리에 대한 기술 습득과 같은 기대효과를 얻을 수 있다.

향후에는 제시된 요구사항관리 프로세스의 실

제 업무 적용과 사례분석을 통한 평가에 의하여, 각 업무 도메인과 개발 환경에 따라 적절하게 적용할 수 있도록 프로세스와 활동을 커스터마이징 및 컴포넌트화하는 연구가 계속될 것이다.

※ 참고문헌

[1] G. Kotonya, I. Sommerville, "Requirements Engineering Process and techniques", John Wiley & Sons Ltd, 1998.

[2] I. Sommerville, P. Sawyer, "Requirements Engineering - A Good Practice Guide", John Wiley & Sons Ltd, 1997.

[3] CMU SEI, "The Capability Maturity Model - Guidelines for Improving the Software Process", Addison Wesley, 1995.

[4] INCOSE, "Requirement Management," <http://incose.org/workgrps/rwg>, 1996.

[5] Enrique Garca Alczar, Antonio Monzn, "A Process Framework for Requirements Analysis and Specification," IEEE 4th International Conference on Requirement Engineering, June, 2000.

[6] D. A. Leffingwell, "A Field Guide to Effective Requirements Management Under SEI's Capability Maturity Model", Rational Software Corporation, 1998.

[7] A. I. Anton, "Goal-based requirements analysis," Proc. of the 2nd IEEE Int'l Conference on Requirements Engineering, 1996.

[8] Holbrook, H., "A Scenario-based Methodology for Conducting Requirement Elicitation", ACM Software Engineering Note, Vol.15, No.1, 1990.

[9] Sooyong Park, Harksoo Kim, Youngjoong Ko and Jungyun Seo. Implementation of an efficient requirements analysis supporting system using similarity measure techniques. , Infor-

mation and Software Technology, Vol. 42, Issue 6, April 2000, pp. 429-438.

[10] Dean Leffingwell, Don Widrig, Managing Software Requirements A Unified Approach, Addison-Wesley, 2000.

[11] Sooyong Park, Hoh Peter In, Soonhwang Choi, Automated Support to Quality Requirements Classification for Supporting WinWin, 2nd International Conference on Computer and Information Science, pages. 43-58,2002.8

황 만 수



1984 중앙대학교 전자계산학과 (이학사)
 1986 중앙대학교 전자계산학과 (이학석사)
 2001 숭실대학교 컴퓨터학과 (공학박사)
 1987~1993 LG 소프트웨어 연구원
 1993 ~ 현재 신홍대학 컴퓨터 정보계열 부교수
 관심분야 : 요구공학, SW 프로세스 개선

민 태 홍



1981년 중앙대학교 전자계산학과 졸업(이학사)
 1983년 중앙대학교 대학원 전자계산학과(이학석사)
 1992년 중앙대학교 대학원 컴퓨터공학과(공학박사)
 1984년-현재 인하공업전문대학 컴퓨터정보공학부 교수
 관심분야 : 분산운영체제, SW 프로세스 개선, 무선인터넷