

<http://dx.doi.org/10.7236/IIBC.2014.14.1.37>

IIBC 2014-1-6

소프트웨어 요구공학에 대한 처리 시스템 정립

An Establishment of the Process System for Software Requirements Engineering

이상운*, 최명복**

Sang-Un Lee*, Myeong-Bok Choi**

요약 본 논문은 지금까지 제안된 요구사항 공학과정을 체계적으로 분석하고 최적의 과정을 제안하였다. 세부적으로는 요구사항 추적관리와 기원을 살펴보고, 요구사항 명세서를 도출하는 다양한 출발점 (고객이 제시한 요구사항 명세서로부터, 요구사항 도출로부터, 문제분석으로부터, 비즈니스 모델링부터)을 고찰하였다. 또한, 이해 당사자 욕구 도출, 문제 분석, 비즈니스 모델링의 수행 우선순위에 대한 다양한 방법을 고찰하였다. 이들 결과로부터 "전산화 목표 파악, 비즈니스 프로세스 파악, 비즈니스 모델링, 문제분석, 이해당사자 욕구 도출, 시스템 특징 도출, 요구사항 도출, 요구사항 명세와 검증" 과정을 수행하는 것이 가장 체계적으로 요구사항 명세서를 도출하는 과정임을 제안하였다. 이 방법을 적용하면 개발단계에서 관리하면 개발단계에서의 요구사항 변경을 최소화 시키고 프로젝트 성공률을 향상시킬 수 있으며, 고객의 만족도도 향상시킬 수 있을 것이다.

Abstract This paper proposes a optimal process after analyzing the existing papers of software requirements engineering process. In analyzing it examines requirement pursuit management and origin, and considers a variety of starting points that draw requirement documents. Also, this paper considers carefully the multiple methods of stakeholder needs, problem analysis, processing priority of business modeling. From the analyzed result this paper proposes the best method that draws systematically the requirement documents. The proposed method can minimize the requirement changes, draw the higher project success ratio, and satisfy the customer.

Key Words : Software Requirements Engineering Process, Requirements Analysis, Requirements Management, Stakeholder Needs, Problem Domain, Business Modeling

1. 서 론

소프트웨어가 겪고 있는 복잡도 (Complexity), 고객의 기대치 (Expectations)와 요구사항 변경 (Requirements Creep) 등의 근본적인 원인으로 인해 개발된 소프트웨어는 예산 초과 (Over-budget), 일정 지연 (Over-time), 낮은 품질 (Low Quality), 고객 요구사항 불만족, 프로젝트

의 관리 불가능과 유지보수를 위한 코딩 어려움 결과를 나타내는 현상을 소프트웨어 위기 (Software Crisis)라 표현한다. 소프트웨어 위기는 1960년대 말에 Dijkstra^[1]가 제기한 이후 지금까지도 해소되지 않고 있다. 소프트웨어 성공의 조건은 일반적으로 비용 (Cost), 일정 (Time), 품질 (Quality)과 요구사항 (Requirements) 만족으로 평가되며, 소프트웨어 위기가 제기된 지 반세기가

*정회원, 강릉원주대학교 멀티미디어공학과

**중신회원, 강릉원주대학교 멀티미디어공학과

접수일자 2013년 8월 20일, 수정완료 2013년 12월 19일

게재확정일자 2014년 2월 7일

Received: 20 August, 2013 / Revised: 19 December, 2013

Accepted: 7 February, 2014

*Corresponding Author: cmb5859@gmail.com

Dept. of Multimedia Eng., Gangneung-Wonju National University, Korea

지난 현 시점에서도 해소되지 않고 있으며, 소프트웨어 프로젝트 성공률은 30% 이하이다.^[2]

Standish Group Report^[3]는 소프트웨어 프로젝트 실패의 가장 근본적인 원인을 “요구사항 불만족”에서 찾고 있다. 세부적으로는 사용자의 요구사항 제시 불충분 (13%), 불완전한 요구사항과 명세서 (12%)와 요구사항 변경 (12%) 등 프로젝트 실패 원인의 37%를 차지하고 있다. 소프트웨어 프로젝트가 성공하기 위해서는 요구사항 도출이 필수적이며, 이를 위해 사용자와 지속적인 대화를 통해 요구사항 문서화, 요구사항의 사용자 검증, 요구사항에 대한 변경관리와 추적성 관리가 요구된다.^[4]

현재 적용되고 있는 요구사항 도출 방법은 어떤 것들이 있기에 프로젝트 성공률이 매우 낮은 수준인가? 요구사항은 어디에서 기원 (Source)하며, 어떤 과정을 통해 체계적으로 도출될 수 있는가? 개발된 소프트웨어가 이해당사자 욕구 (Stakeholder Needs)를 만족시킬 수 있는 최선의 방법은 존재하는가?

본 논문은 이러한 의문을 해소하기 위해 기존의 요구사항의 기원부터 시작하여 최종적으로 고객이 만족하는 소프트웨어 요구사항 명세서 (Software Requirements Specification, SRS)를 도출하는 체계적인 방법을 제안한다.

2장에서는 소프트웨어 요구사항 공학과정 (Software Requirements Engineering Process, SREP), 요구사항 추적관리 방법과 요구사항의 기원을, 3장에서는 SRS를 도출하는 다양한 방법들과 문제점을 고찰해본다. 4장에서는 조직의 현 수행 업무 분석부터 시작하여 SRS를 도출하는 체계적인 방법을 제안한다.

II. 소프트웨어 요구사항 공학

1. 소프트웨어 요구사항 공학 수행 과정

소프트웨어 요구사항에 대해 IEEE Std 610.12-1990^[5]는 (1) 문제를 풀기 위해 또는 목적을 달성하기 위해 사용자가 필요로 하는 조건 (Condition) 또는 능력 (Capability)의 문서화된 표현; (2) 시스템 (또는 구성품)이 계약, 표준서, 명세서 또는 다른 문서에 일치하거나 갖추어야 할 조건 또는 능력의 문서화된 표현”으로, Sommerville와 Sawyer^[6]은 “구현하고자 하는 것의 명세”, Suen^[7]은 “시스템에 부과된 능력”, Hossenlopp^[8]는

“목적 달성을 위해 풀어야할 문제에 대한 사용자가 필요로 하는 능력”이라 정의하고 있다.

소프트웨어 요구사항 공학 (Software Requirements Engineering, SRE)에 대해 Wiegiers^[9]은 “개발을 착수하기 전에 무엇을 개발해야 하는지 이해하는 것”으로, Brackett^[10]은 “정의, 분석, 대안연구와 프로토타이핑 등의 반복적인 과정을 거치면서 시스템 요구사항을 소프트웨어 요구사항, 성능 변수와 소프트웨어 형상으로 기술하는 활동”으로 정의하고 있다.

SREP는 일반적으로 요구사항 분석 (Requirements Analysis, RA)과 요구사항 관리 (Requirements Management, RM)로 구분된다. 요구사항 분석은 “도출 (Elicitation) → 분석 (Analysis) → 명세 (Specification) → 검증 (Validation)” 단계를 거친다. 요구사항 관리는 “소프트웨어 개발 과정에서 발생하는 요구사항 변경 (Requirements Creep)에 대한 관리”이다.^[6,11] SREP에서 거론되고 있는 용어들을 살펴보면 표 1과 같이 하나의 개념에 대해서도 다양한 동의어들이 사용되고 있다.

표 1. 소프트웨어공학과정에서 사용되고 있는 동의어
Table 1. Synonym that are using software engineering process

용어	동의어
소프트웨어 요구사항 공학 과정 (Software Requirements Engineering Process)	· 요구사항 공학 영역 (Requirements Engineering Domain) · 요구사항 생성 프로세스 (Requirements Generation Process)
요구사항 관리 (Requirements Management)	· 요구사항 생성 프로세스 (Requirements Generation Process)
요구사항 분석 (Requirements Analysis)	· 요구사항 공학 (Requirements Engineering) · 요구사항 수집 (Requirements Gathering) · 요구사항 수집 (Requirements Capture) · 운영개념 문서화 (Operational Concept Documenting) · 시스템 분석 (System Analysis) · 요구사항 명세 (Requirements Specification) · 요구사항 개발 (Requirements Development) · 요구사항 생성 (Requirements Generation)
이해당사자 욕구 (Stakeholder Needs)	· 사업 목적 (Business Objectives)
문제 보고서 (Problem Statement)	· 비전 문서 (Vision Document) · 프로젝트 헌장 (Project Charter) · 프로젝트 요구사항 문서 (Project Requirements Document) · 마케팅 요구사항 문서 (Marketing Requirements Document)
시스템 요구사항 (System Requirement)	· 비즈니스 요구사항 (Business Requirement)
도출 (Elicitation)	· 발견 (Discovery) · 수집 (Capture, Gathering) · 획득 (Acquisition) · 이해당사자 욕구 이해 (Understanding stakeholder Needs)
명세 (Specification)	· 시스템 정의 정제 (Refine the system definition) · 기록 (Recording) · 명세/문서화 (Specification/Documentation)
검증 (Validation)	· 검증 (Verification & Validation)
정의 (Definition)	· 특징 (Features)
요구사항 관리 (Requirements Management)	· 진화 (Evolution)

요구사항 도출과 요구사항 분석단계에서의 “요구사항”은 우리가 일반적으로 알고 있는 소프트웨어 요구사항이 아니며 “이해당사자 욕구 또는 시스템 요구사항”이다. 용어의 혼란을 피하기 위해 이를 “이해당사자 욕구”로, “요구사항 분석”을 “요구사항 생성”으로 이해하자.

순차적 개발 방법인 폭포수 모델 (Waterfall Model)을 포함한 기존의 대부분의 소프트웨어 개발 프로세스들은 소프트웨어 개발을 요구사항 분석 (RA) 단계부터 수행하여 설계, 코딩, 시험과 납품 과정을 거친다. 이 방법론은 소프트웨어 개발을 시작하기 전에 SRS가 완전히 이해되고 정의되어 있으며 변경이 거의 발생하지 않는다는 가정에 기반하고 있다. 그러나 사용자는 개발 초기에 요구사항을 정확히 기술하지 못한다. 이러한 불완전한 요구사항에 기반하여 소프트웨어를 개발하면 결국은 고객을 만족시킬 수 없는 결과를 초래한다.

반면에 반복적이고 점진적 개발 방법인 단일화된 프로세스 (Unified Process, UP)는 소프트웨어가 배치될 사업 환경 또는 사업 절차 (Business Process)를 이해해야만 보다 명확한 소프트웨어를 개발할 수 있다고 본다. 따라서 비즈니스 모델링 (Business Modeling)과 요구사항 단계를 수행한 후 분석 및 설계, 구현, 시험과 납품 과정을 반복적으로 수행한다.

객체지향 분석은 시스템 분석분야의 요구사항 구조 (Structuring) 수준에서는 실제적인 혁신을 가져왔으나 기본적인 요구사항 결정 (Determination) 방법은 거론하지 못하였다. 객체지향 프로그래밍을 적용하기 위해 DFD (Data Flow Diagram), 구조화된 영어 문장 등과 같은 전통적인 요구사항 구조화 도구들과는 완전히 다른 유스케이스 다이어그램 (Use-case Diagram), 클래스 다이어그램 (Class Diagram) 등과 같은 객체지향분석 요구사항 구조화 도구들이 개발되었다. 객체지향 분석방법이 전통적인 구조적 분석방법보다 좋은지는 아직도 의문사항으로 남아있다.

구조적이나 객체지향 개념에 상관없이 결국 요구사항 명세서 (SRS) 도출은 고객 만족을 판단하는 대표적인 속성이며, 소프트웨어 성공의 조건인 일정, 비용과 품질을 설정하는 기준 (Baseline)이 되는 동시에 소프트웨어 개발 과정의 기준을 제시한다. 따라서 프로젝트를 성공하기 위해서는 SRS를 보다 정확히 도출하는 것이 필요하다.

2. 요구사항 추적 관리와 요구사항 기원

소프트웨어 개발 성공률을 높이기 위해서는 요구사항 추적성이 절실히 요구된다. 요구사항 추적성은 그림 1과 같이 사전-요구사항 추적성 (요구사항의 기원부터 요구사항 명세서 까지)과 사후-요구사항 추적성 (요구사항 명세서부터 구현까지)으로 구분될 수 있다.^[11] 사전-요구사항 추적성은 Sommerville^[6]과 Sawyer^[6]과 Matthews^[11]가 정의한 소프트웨어 요구사항 공학 과정의 “요구사항 분석” 과정으로, 사후-요구사항 추적성은 “요구사항 관리”에 대응된다고 할 수 있다.

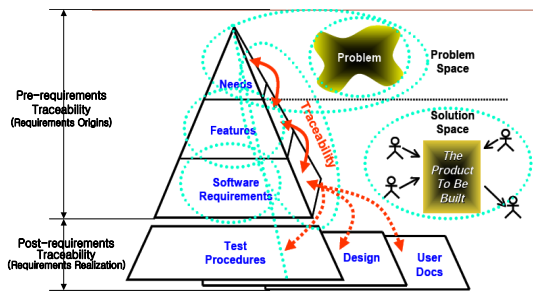


그림 1. 소프트웨어 요구사항 체계
Fig. 1. Software requirement system

지금까지 사후-요구사항 추적성에 대해서는 많은 연구가 이루어졌고, 소프트웨어 형상관리 (Configuration Management) 분야로 발전하였다. 그러나 아직까지도 소프트웨어 프로젝트의 성공률이 30% 이하가 되는 주된 요인은 사전-요구사항 추적성에 대한 체계적인 과정이 설정되지 못한 결과라 할 수 있다. 따라서 사전-요구사항 추적성에 대한 체계적인 과정을 설정하기 위해서는 “소프트웨어 요구사항의 기원은 어디인가”, “어떤 과정을 거쳐 소프트웨어 요구사항으로 명세화되는가”를 알아볼 필요가 있다. 특히 고객의 욕구를 어디로부터 어떻게 도출하는지가 초점의 대상이다. 왜냐하면, 고객의 욕구를 이해하는 것은 항상 난제이며, 아직까지는 고객 욕구 수집이 프로젝트 성공의 결정적 요인이기 때문이다.^[8]

그림 1에서 문제 (Problem)는 “인식하고 있는 생각과 바라는 생각과의 차이점”이며, 문제영역 (Problem Domain 또는 Problem Space)은 “시스템 개발을 성공하기 위해 기술되어야만 하는 실제적인 이해당사자들의 욕구에 대한 발상지”이다.^[7] 이해당사자 욕구에 대해 Suen^[7]과 Leffingwell^[12]는 “문제 영역과 관련된 비즈니스, 개인 또는 운영상의 문제점에 대한 생각”으로,

McEwen^[13]은 “문제영역의 일부분으로 성공적인 프로젝트를 위한 이해당사자들의 요청을 기술한 것”으로 정의하고 있다. 특징 (Features)은 “하나 또는 그 이상의 이해당사자 욕구를 충족시키기 위해 시스템이 제공하는 서비스”이다.^[7]

이해당사자 욕구로부터 시작하여 해결영역 (Solution Domain)의 특징을 거쳐 소프트웨어 요구사항이 도출된다.^[12-15] 소프트웨어 요구사항은 기능 요구사항 (Functional Requirements), 비기능 요구사항 (Non-functional Requirements, 또는 Quality Requirements)과 설계 제약사항 (Design Constraints)으로 구분된다.

이해당사자의 욕구는 현재의 업무를 수행하면서 발생하는 문제영역으로부터 도출된다. 따라서 문제영역에 있는 실 사용자와 이해당사자의 실질적인 욕구를 이해하여야 하며, 이를 위해서는 문제를 정의하기 전에 문제영역에 대한 이해가 필요하다. 문제를 수집하는 최적의 방법은 문제 보고서 (Problem Statement)이며^[16], 문제 보고서를 작성하기 위해 가장 중요한 사항은 사용자의 목표 (Users' Goal)이다.^[17]

III. SRS 도출 방법 문제점

SRS를 도출하는 과정은 그림 2와 같이 다양한 방법들이 제시되어 있다.

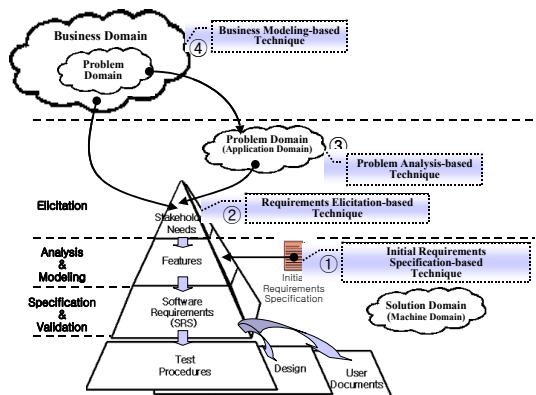


그림 2. SRS 도출 과정
Fig. 2. SRS derivation process

본 논문에서는 이들 과정을 각각 초기 요구사항 명세서-기반 (Initial Requirements Specification-based), 요

구사항 도출-기반 (Requirement Elicitation-based), 문제분석-기반 (Problem Analysis-based)과 비즈니스 모델링-기반 (Business Modeling-based) 방법으로 분류한다.

시스템 분석을 위해 공인 받은 최적의 방법은 없으며, 분석가의 경험, 사용자의 선호도, 프로젝트 규모, 프로젝트 복잡성 등 프로젝트 환경에 따라 적절히 적용된다. 따라서 훌륭한 시스템분석가는 특정한 프로젝트 배경에 따라 이들 방법들 중 적절한 방법을 선택하여 사용해야만 한다. 본 장에서는 이들 방법들을 고찰해 본다.

1. 초기 요구사항 명세서-기반 SRS 생성

과거에는 특정 고객이 서명한 초기 요구사항 명세서에 기반하여 요구사항에 대한 분석-정의-명세 과정이 수행되었다.^[18] 이 방법을 적용할 경우 개발된 프로젝트가 고객의 욕구를 불만족하는 결과를 초래한다. 왜냐하면 고객이 제시한 초기 요구사항 명세서는 다음과 같은 문제점으로 인해 불완전하고 부정확해 질 수 밖에 없다.^[19]

- (1) 사용자는 그들의 요구사항을 알지 못한다.
- (2) 사용자는 요구사항을 알고 있지만 명확히 표현하지 못한다.
- (3) 사용자의 생각은 변한다.
- (4) 다른 사용자는 다른 요구사항을 가지고 있으며, 동일한 문제에 대해서도 다른 관점에서 본다.
- (5) 다른 사용자는 요구사항을 통합하는 것에 동의하지 않는다.
- (6) 사용자는 요구사항의 개발 우선순위에 동의하지 않는다.
- (7) 개개의 사용자는 모든 사용자의 대표자가 되지 못한다.
- (8) 사용자와 이해당사자들은 자신이 수행하는 일을 보다 능률적으로 수행하는데 무엇이 필요한가를 설명하는 데는 전문가이다. 그러나 조직의 전체 문제를 어떻게 해결하는지에 대해서는 알지 못한다.

결국 개발자는 전체 문제영역을 이해하기 위해서는 모든 이해당사자들의 욕구를 이해하는 “요구사항 도출 과정”을 거쳐야 한다.^[13]

2. 요구사항 도출-기반 SRS 생성

요구사항 도출-기반 방법은 이해당사자들로부터 직접 욕구 들을 도출하는 활동부터 수행하는 방법이다. 요구사항 도출 방법에 대해 Young^[20]는 “규정, 명세서, 표준

서, 현존하는 시스템과 프로세스, 고객과 사용자의 욕구와 기대”로부터, Whitney^[21]는 “문제영역 전문가, 사용자, 현존하는 수기식 프로세스와 시스템, 유사한 소프트웨어 프로그램 검토”로부터 도출할 수 있다고 제시하고 있다. 이해당사자들로부터 욕구를 도출하기 위한 기법으로는 표 2와 같이 다양하게 제안되어 있다.

표 2. 이해당사자 욕구 도출 기법
Table 2. Stakeholder needs derivation method

기법	참고문헌	[11]	[18]	[21]	[23]	[24]	[25]	[26]	[27]	[28]	[29]	[30]
인터뷰		○	○	○	○	○	○	○	○	○	○	○
질문서		○	○	○	○	○	○	○	○	○	○	○
조사			○		○							
RAD/JAD		○	○	○	○	○	○	○				
프로토타이핑		○	○	○	○	○	○	○				
모델기반 기법			○									
Social Analysis		○			○							
계약서 종류 요구사항 목록								○				
QFD					○							
협력 요구사항 수집					○	○						
초점 그룹 (인터뷰, 워크숍)					○	○		○	○		○	○
유스케이스 워크숍		○	○								○	○
Introspection						○						
Wall Chart							○					
브레인스토밍					○							
역할 연기						○					○	
스토리 보딩						○					○	
관찰(Contextual)		○	○	○	○	○						
직무/프로세스 분석		○	○		○						○	
현존 문서 분석			○	○	○	○			○			○

그러나 보편적으로 가장 좋은 하나의 방법은 없다는 것이 일반적인 견해이다. 정확한 접근법은 개발될 시스템의 본질, 이해당사자들의 경험과 배경 (기초지식)에 의존한다.^[22]

요구사항 도출 단계부터 시작하여 SRS를 생성하는 방법은 표 3과 같이 Sommerville와 Sawyer^[6], Matthews^[11], Nuseibeh와 Easterbrook^[18], Offen^[23], Wikipedia^[26], RIT^[31], Rehman^[32], Pohl^[33], Darke^[34]와 Zhu^[35] 등이 제안하고 있다. 이들을 종합하면 SRS 생성 과정은 도출, 분석과 모델링, 명세와 검증 단계로 표현할 수 있다.

표 3. 요구사항 도출-기반 SRS 생성 과정
Table 3. Requirement derivation-base SRS creation process

참고문헌	수행 과정
[23, 31, 32]	도출-분석과 모델링-명세-검증
[11, 36]	도출-분석-정의-명세
[6]	도출-분석-검증
[33]	도출-결충-명세-검증
[18]	도출-모델링과 분석-대화-동의
[34]	결정/획득/수집-모델링/구조화/명세
[35]	결정 (도출, 문서화, 조직, 추적)-구조화

요구사항 도출 단계에서 수행하는 활동에 대해서는 표 4와 같이 다양하게 제시하고 있다.

표 4. 요구사항 도출-기반 방법의 단계별 수행활동
Table 4. Fulfill activity according to level of requirement derivation-base method

단계	참고 문헌	수행활동
도출	[22, 36]	이해당사자 욕구 발견, 욕구들 간 잠재적 충돌사항 확인/협의, 시스템 경계 (범위와 영역) 설정
	[23]	해결할 문제에 대한 이해당사자 욕구 발견, 개발될 시스템 경계 확인, 이해당사자 확인, 시스템의 목표와 임무 확인, 요구사항 (유스케이스) 확인
	[26]	이해당사자 욕구 도출
	[37]	사업 수준의 요구사항 (사업 목표, 시스템 범위와 한계, 시스템 배경도, 비전 문서) 확인, 모든 이해당사자를 확인, 이해당사자들의 대표 확인, 욕구 도출
	[13]	모든 이해당사자 확인, 이해당사자 대표 확인, 이해당사자 욕구 도출, 이해당사자 욕구 들 간 충돌사항 해결
	[38]	요구사항 도출 (유스케이스 모델링)
분석과 모델링	[22, 36]	제품과 상호작용에 대한 보다 심도 깊은 이해를 하며, 고 수준의 아키텍처 설계를 정의하기 위해 시스템에 전체적인 영향을 미치는 요구사항 확인 (확인), 아키텍처 구성품에 요구사항을 할당, 아키텍처 구현과정에서 발생하는 추가적인 충돌사항 확인
	[24]	문제에 대한 보다 나은 이해, 시스템 영역 발견, 욕구간의 중첩, 연관성 확인, 시스템 요구사항을 소프트웨어 요구사항으로 상세화, 잠재적 문제점(충돌, 불완전, 모순, 설계충족, 불필요, 모호성, 실행 불가능, 시험 불가능 등) 확인 및 해결, 분류, 개념적 모델링 (Context, 시스템 지원 활동, 프로세스, 프로세스 간 데이터 흐름, 시스템이 처리하는 데이터의 논리적 구조, 제어 흐름, 상태 모델, 사용자 상호작용 (use case), 객체 모델 등), 아키텍처 구성품으로 할당, 협의 (요구사항 충돌 문제 해결)
	[26]	이해당사자 욕구의 불명확, 불완전, 애매모호, 충돌 해결
	[37]	분석은 요구사항 모델링과 제품 요구사항 확인 모델링 방법 : 객체지향 (Use Case Diagram, Class Diagram, Activity Diagram, Sequence Diagram), 구조적 분석 (DFD, ERD, State Transition Diagram & Table), 기타 (Process Flow Diagram, Decision Tree, Event/Response Table)
명세	[22, 36]	요구사항을 문서화하는 과정
	[24]	시스템과 외부환경 (Actor)간 인터페이스를 완전히 기술
	[36]	폭포수 모델은 자연언어, UML은 Use Cases, Extreme Programming은 User Stories
검증	[22, 36]	제품이 고객의 요구사항과 일치하는지 검증하는 활동
	[24]	요구사항 문서 검증 (일관성, 완전성, 정확성)

이 방법의 문제점은 다음과 같다.

- (1) 이해당사자가 제기한 욕구들 사이의 충돌사항 확인/해결은 도출 단계와 분석 단계에서 혼용되어 수행되고 있다.
- (2) 모델링 (요구사항 확인) 활동은 도출단계, 모델링 단계와 명세 단계에서 혼용되어 수행되고 있다.
- (3) 분석 단계는 우리가 일반적으로 알고 있는 분석 개념이 아닌 이해당사자의 욕구를 검증 (충돌, 불완전, 모순, 모호성, 실행 불가능, 설계 충족, 불필요, 시험 불가능 등)하는 과정으로 검증단계로 잘못 이해할 수 있다.

(4) 도출단계에서는 이해당사자 욕구 도출까지로 보는 견해와 이해당사자 욕구로부터 시스템의 서비스인 특정 도출 단계를 거치지 않고 바로 소프트웨어의 기능적 요구사항까지 도출하는 것을 범위로 정하기도 하여 도출단계에서 수행하는 활동의 범위를 결정하는데 혼란을 겪고 있다.

본 논문의 목적은 이들 4 단계에서 수행되는 활동들을 명확히 구분하고자 함이 아니며, 어떻게 이해당사자들의 욕구를 도출할 수 있는가에 초점을 맞추고자 한다. 왜냐하면 이해당사자의 욕구를 어디로부터, 어떻게 도출할 수 있는지가 관심의 대상이기 때문이다. 따라서 본 논문에서는 도출단계에서 수행하는 활동을 “이해당사자들 확인, 이해당사자들 욕구 도출과 개발될 시스템 경계 설정”으로 한정시킨다.

요구사항 생성에 대한 대부분의 일반적인 시작점은 해결하고자 하는 문제를 확인하기 전에 먼저 해법을 생각하는 것이다.^[39] 이와 같은 생각은 잘못된 문제를 해결하거나 불필요하게 제약에 대한 해결만을 도출한다. 지금까지의 요구사항 공학과정에서는 “누가 시스템을 사용하며, 시스템이 무엇을 수행해야하는가”에 대한 단계는 거의 전적으로 누락되었다. 객체지향 전문용어로 문제정의 (Problem Definition)의 첫 번째 시도는 형식적인 문제 보고서 (Formal Problem Statement)로 이는 “해결하고자 하는 문제를 기술한 것”이다.^[16,40] 구조적 (절차적) 개발방법론에서는 요구사항 도출 (수집)이라 불리는 활동으로 인해 문제 보고서는 빈번히 누락되었다. 그러므로 요구사항 도출 대신 문제 정의 용어를 사용하는 것이 타당해 보인다.^[17]

3. 문제분석-기반 SRS 생성

문제는 “인식하는 것과 원하는 것과의 차이점 또는 수행될 의문이나 주제”로 정의된다.^[11] 문제영역은 개발될 시스템과 관련하여 문제가 존재하는 실세계 부분으로 어플리케이션 영역 (Application Domain) 또는 운영 환경 (Operational Environment)이라고도 한다.

새로운 시스템의 기능과 속성들을 정의하기 위해서는 문제영역에 대한 포괄적인 이해가 요구된다.^[37] 문제영역과 문제에 대한 근본 원인 분석의 중요성은 Davis^[41]와 Leffingwell^[42]에서 제기하고 있다. 문제분석은 “현실세계에서 발생되고 있는 문제점과 이해당사자들의 욕구를

이해하고 이들 욕구를 해결할 해법을 제안하는 과정”이다.^[47,16] Magsarjav^[36]은 “문제영역에 대한 정보는 이해당사자들, 조직의 문서와 현존하는 시스템으로부터 도출할 수 있다”고 제시하고 있다.

요구사항 생성을 문제 분석부터 시작하는 방법으로는 표 5와 같이 Suen^[7], Finkelstein^[24], Baghai^[29], Cofod^[43]와 Krasner^[44]가 있다.

표 5. 문제분석-기반 SRS 생성 과정

Table 5. Problem analysis-base SRS creation process

참고문헌	수행 과정
Suen ^[7]	문제 분석 → 이해당사자 욕구 이해 → 시스템 정의 (소프트웨어의 특정 확인, 요구사항 형태별 분류, SRS를 작성) → 시스템 범위 관리 → 시스템 정의 정제
Finkelstein ^[24]	문제 영역 분석 → 이해당사자 확인 → 사업 목표 도출 → 이해당사자 욕구 도출 → 요구사항 도출
Baghai ^[29]	문제 분석 (비즈니스 모델링을 통한 문제 확인) → 이해당사자 욕구 이해 (수집) → 시스템 정의 (특정 확인, 요구사항 분류) → 시스템 범위 관리 (요구사항 우선순위, 개발 우선순위 결정) → 시스템 정의 정제 (유스케이스의 Event 흐름도 완성)
Cofod ^[43]	문제 발견과 분석 (문제 확인, 원인과 효과 분석) → 이해당사자 욕구 수집-요구사항 초안 작성 → 분석 (검증) → 형식화 (SRS)
Krasner ^[44]	이해당사자 욕구 확인과 문제분석 → 요구사항 결정 → 요구사항 명세

문제분석 단계에서 수행하는 활동은 “문제 정의, 문제의 근본 원인 이해, 이해당사자들 확인, 이해당사자들 욕구 도출, 시스템 경계 확인, 시스템에 영향을 미치는 제약 사항 확인”이다.^[47,15,22,29,45-47] 요구사항 도출단계에서는 “이해당사자 욕구를 시스템 요구사항으로 변환, 해법 시스템 경계 정의” 활동을 수행한다.^[47]

4. 비즈니스 모델링-기반 SRS 생성

UML을 사용하고 있는 많은 개발 프로세스들은 유스케이스 모델링 (소프트웨어의 기능적 요구사항) 도출부터 시작한다. 그러나 이 방법을 적용하면 시스템이 운영될 비즈니스를 최적으로 지원하는 모든 또는 정확한 유스케이스 들을 도출할 수 없다. 따라서 시스템이 운영될 현존 환경에 대한 모델과 이해가 필요하며, 문제 영역에 대한 이해 없이는 적절하고 유용한 유스케이스를 생성하기가 불가능하다.^[48,49]

고객은 문제에 대해 단지 피상적으로 이해한다. 고객은 증상과 모순되는 문제를 체계화 시킨 것에 대해 확신을 가지고 있다. 고객은 근본적인 문제를 이해하기 전에 해법 영역을 생각한다. 이와 같은 이유로 인해 요구사항 분석가는 영역지식(Domain Knowledge)이 부족하다. 개

발자는 사업 목표와 관련된 식견과 고객과 개발자간의 대화를 위한 비즈니스 모델 없이 소프트웨어 시스템이 지원해야 하는 프로세스를 정확히 이해하여 개발하는 것은 불가능하다.^[49] 시스템이 어디에 배치되는지, 누가 사용하는지, 현존하는 시스템과 통합을 어떻게 하는지와 어떤 특정 비즈니스 직무 (Task)가 전산화되는지에 대한 이해는 성공적인 시스템 개발의 열쇠가 된다.^[50] 따라서 고객과 분석가 간의 원활한 대화를 위해 분석가는 현존하는 시스템 (수기식 또는 전산화)과 개발될 시스템의 운영 환경에 대한 이해가 필요하다.^[47]

현재의 IT 개발 흐름은 프로젝트가 사업에 명확한 가치를 제공하지 않는 한 조직은 IT 프로젝트에 더 이상 투자하지 않는다. 따라서 프로젝트를 시작하기 전에 고객과 사업 이해당사자들은 비즈니스 분석을 수행해야 한다.^[22] 특정 문제분석 기법으로 IS/IT 응용분야는 비즈니스 모델링을, 내장형 시스템영역의 소프트웨어 집약형 시스템에는 시스템 모델링을 수행한다.^[15]

Rational^[50], Ng^[51]와 Roggio^[52]은 “비즈니스 모델링의 목적은 모든 비즈니스 프로세스 (관련된 데이터, 활동들, 임무와 비즈니스 규칙들)를 기술하고, 시스템이 구축되는 것과 관련된 조직의 활동을 이해하기 위해”라고 정의하고 있다. Ng^[51]는 “비즈니스 모델링은 프로젝트 팀에게 조직의 구조와 기능 이해, 현존하는 문제점 이해와 개선될 잠재적 영역 확인, 목표하는 조직에 대한 공통된 이해와 시스템 요구사항 도출을 제공한다”고 제시하고 있다.

Roggio^[52]는 비즈니스 모델링은 문제 보고서, 조직의 전반적인 비전, 조직의 전문가 지식, 문제영역과 이해당사자의 욕구를 기술한 모델 또는 문서로부터, Cahill^[19]는 이해당사자들, 조직, 현존하는 시스템, 업계, 고객, 프로젝트, 운영환경, 사용자환경 (사업 분야, 문제영역, 표준서, 업계 규정)으로부터 비즈니스 모델링을 도출할 수 있다고 제시하고 있다.

비즈니스 모델링부터 시작하여 요구사항 명세서를 도출하는 방법으로는 표 6과 같이 Cofod^[43], Rational^[50], Ritter^[53], Lamsweerde^[54]와 강문석^[55]이 있다.

Ray^[4]와 Offen^[23]는 비즈니스 모델링에서 수행하는 활동으로 “상세한 사업 환경 이해, 문제 해결에 대한 다양한 제약사항 이해, 주요 이해당사자들 확인, 영역 모델 생성, 개발 프로세스 이해/정의”를 제안하였다.

표 6. 비즈니스 모델링-기반 SRS 생성 과정

Table 6. Business modeling-base SRS creation process

참고문헌	수행 과정
Cofod ^[43]	현존 시스템 이해 (비즈니스 모델링) → 개선 기회 확인 (이해당사자 욕구 수집) → 새로운 시스템 특정 도출
Rational ^[50]	비즈니스 모델링 단계 : 시스템이 개발될 비즈니스의 구조와 기능 수집 → 문제 이해 → 비즈니스 프로세스의 잠재적 개선사항 확인 요구사항 단계 : 요구사항 설정과 시스템 경계 설정 분석 단계 : 기능적과 비기능적 요구사항 분석
Ritter ^[53]	요구사항 분석 (현존 시스템 관찰로부터 시스템 요구사항 도출) → 요구사항 정의 → 시스템 요구사항 명세 → 소프트웨어 요구사항 명세
Lamsweerde ^[54]	영역 분석 (Domain analysis) → 이해당사자 욕구 도출 → 절충과 동의 → 명세 분석 → 명세 분석 → 문서화
강문석 ^[55]	구조적 시스템 분석절차 : 현 운영 시스템 (전산화 또는 수기식)의 물리적 시스템 분석과 논리적 모델 분석 → 이해당사자 요구사항 도출 → 새로운 시스템의 논리적 모델 (시스템 특징)을 도출하고 물리적 모델 구현

IV. 체계적인 요구사항 공학 과정

1. 이해당사자 욕구 도출 이전에 문제분석과 비즈니스 모델링 수행의 필요성

이해당사자 제기 욕구-기반 방법을 적용하여 소프트웨어를 개발하는 것이 고객 만족의 가장 중요한 요인이 된다. 그러나 이해당사자들은 욕구를 정확히 표현하기 어렵고, 동일한 문제에 대해서도 다른 관점을 가지고 있어 하나의 문제에 대해 이해당사자 욕구들이 다르게 도출된다. 이에 따라 표 2에 제시된 이해당사자 욕구 도출 방법들 중에서 가장 적합한 방법들을 발췌하여 보다 심도 있는 이해당사자 욕구 도출을 시도하고 있다. 이 시점에서 이해당사자의 욕구는 어디에서 나오는지 고려해 볼 필요가 있다. 근본적으로 이해당사자들의 욕구는 문제영역에서 도출되며, 문제영역은 조직이 현재 수행하고 있는 비즈니스 영역의 일부이다. 즉, 이해당사자 욕구를 도출하기에 앞서 요구사항 분석가는 “해결할 문제, 조직이 현재 수행하는 비즈니스 프로세스, 시스템이 사용되는 방법, 시스템의 문제영역”에 대한 포괄적인 이해가 요구된다.^[37,38]

문제 확인과 이해당사자 욕구 도출의 우선순위를 살펴보면 Ray^[4], Suen^[7], Ormandjieva^[15], Finkelstein^[24], Baghai^[29], Cofod^[43]와 Sidky et al.^[47]은 “문제 확인 후 이해당사자 욕구 도출”을 제안하였다. 또한, Bittner^[16]은 “문제를 식별하고 이해당사자 욕구 도출”을 제안하였다. 왜냐하면, 이해당사자의 실제적인 욕구를 만족시키려면 그들이 해결하고자 하는 문제가 무엇인지 이해하여야만

하기 때문이다. 반면에 Krasner^[44] 만이 “이해당사자 요구 확인과 문제분석”을 제안하고 있어 이해당사자 요구 확인과 문제분석의 수행 순서를 명확히 구분하기가 어렵다. 이 시점에서 문제분석이 요구사항 도출 단계보다 우선하여 수행되어야 한다는 결론을 얻을 수 있다. 결론적으로, 요구사항 도출-기반 방법의 문제점을 해결하기 위해서는 이 방법에 문제분석 단계를 추가한 문제분석-기반 방법을 적용하는 것이 보다 효율적일 수 있다.

Roggio^[52]는 “비즈니스 모델링과 이해당사자 요구로부터 요구사항 분석”을, Amsden^[56]와 Rational^[50]은 “비즈니스 목표 → 비즈니스 프로세스 모델링 → 소프트웨어 요구사항 분석을, Muchandi^[57]는 ”비즈니스 모델 생성 → 전산화 영역 확인“을 제안하고 있다. 따라서 비즈니스 모델링은 이해당사자 요구 도출 단계보다 우선하여 수행되어야 한다.

2. 문제분석과 비즈니스 모델링의 수행 우선순위

비즈니스 모델링을 문제영역에 대해서만 수행하는지 아니면 조직의 현재 수행되는 비즈니스 전체에 대해 수행하는지를 파악하여야만 비즈니스 모델링과 문제분석의 수행 우선순위를 결정할 수 있다.

문제 영역을 도출하고 문제영역에 대해 비즈니스 모델링을 수행한다는 견해로 Ormandjieva^[15], Eustace^[58], Heumann^[59]과 Booker^[60]가 있다. Ormandjieva^[15]는 “특정 문제분석 기법으로 IS/IT 응용분야는 비즈니스 모델링을, 내장형 시스템영역의 소프트웨어 집약 시스템에는 시스템 모델링을 수행한다”. Eustace^[58]는 “일단 문제가 간결하게 정의되고 문제와 관련된 비즈니스 프로세스와 사건을 발견하면 문제에 대해 보다 선명한 초점을 가질 수 있다”. Heumann^[59]은 “훌륭한 비즈니스 모델은 전산화 대상인 문제영역에 대한 비즈니스 프로세스 (Business Process)로 표현된다”. Booker^[60]는 “이해당사자들이 수행하기를 원하는 활동들이 비즈니스 모델링의 핵심이며, 기본적인 비즈니스 모델링은 시스템을 사용해 수행되는 활동들이 문제영역의 형태를 결정하는 과정이다”고 제안하고 있다.

반면에 조직이 현재 수행하는 업무에 대해 비즈니스 모델링을 수행하고 이로부터 문제영역을 도출한다는 견해로 Cahill^[19], Baghai^[29], Rational^[50], Ng^[51], Roggio^[52]와 Muchandi^[57]가 있다. Cahill^[19]는 “문제영역은 공통된 요

구사항의 원천이며, 사업 영역과 중첩되어 보이지만 사업영역의 특정한 부분이다”. Baghai^[29]는 “문제분석단계에서 비즈니스 모델링으로 문제를 확인한다”. Rational^[50], Ng^[51]와 Roggio^[52]는 “비즈니스 모델링은 시스템이 배치될 조직의 구조와 기능을 이해하고 조직에 있는 현존 문제점 이해와 잠재적 개선 영역 확인을 위해 수행 된다”. Rational^[50]은 “비즈니스 모델링은 전적으로 조직에 대해 수행되며, 문제영역에 대해 수행되는 것은 아니다”. Muchandi^[57]는 “개발하고자 하는 전산화 범위는 조직에서 현재 수행중인 비즈니스 프로세스 변경을 유발시키기 때문에 시스템 개발 범위 내에 포함되지 않는 비즈니스 프로세스가 개발될 시스템의 기능적 요구사항과 결합되지 못하면 시스템의 경계가 불명확해진다. 따라서 전산화 범위에 포함되지 않는 비즈니스 프로세스들도 비즈니스 모델로 문서화된다”고 제안하고 있다. 즉, 비즈니스 모델링을 조직에서 현재 수행되는 비즈니스 프로세스 전체를 대상으로 하는 방법과 현행 비즈니스 프로세스들 중 문제영역에 한해서만 수행하는 방법으로 혼란을 야기 시킨다.

또한, “문제분석 → 비즈니스 모델링 → 이해당사자 요구 도출” 순서로 수행하는 방법으로는 Ray^[4], Suen^[7], Ormandjieva^[15]과 Eustace^[58]가 제안하고 있으며, “비즈니스 모델링 → 문제분석 → 이해당사자 요구 도출” 순서로 수행하는 방법으로는 Wikipedia^[26], Zdravkovic^[61]가 제안하고 있다. 또한 소프트웨어 개발 성공 기준 및 신뢰도에 대한 연구가 각각 Lee^[62], Kim^[63]에 의해 이루어졌다.

결론적으로 이해당사자들이 현재 겪고 있는 문제들에 대한 영역 (나중에 개발될 시스템으로 대체되는 영역)을 도출하기 위해서는 조직이 현재 수행하고 있는 비즈니스 프로세스들 (이해당사자 들에게 주목할 만한 비즈니스를 제공하는 프로세스)을 모두 포함하고 있는 비즈니스 영역에 대해 어떤 활동들이 수행되고 있는지를 우선적으로 분석하여야만 한다. 이를 통해 문제영역의 경계를 설정할 수 있으며, 이 문제영역을 대체할 시스템 개발이 이루어질 수 있다. 따라서 본 논문에서는 비즈니스 모델링이 문제분석 단계보다 우선하여 수행되는 것이 타당하다고 판단된다.

3. 체계적인 소프트웨어 요구사항 공학 과정

지금까지의 결과를 토대로 체계적인 소프트웨어 요구사항공학과정을 설정하기 위해 다음과 같이 수행한다.

사전-요구사항 추적성을 위해 그림 3과 같이 비즈니스 영역, 비즈니스 모델링 영역, 문제영역, 시스템 영역과 해법 영역으로 분류한다.

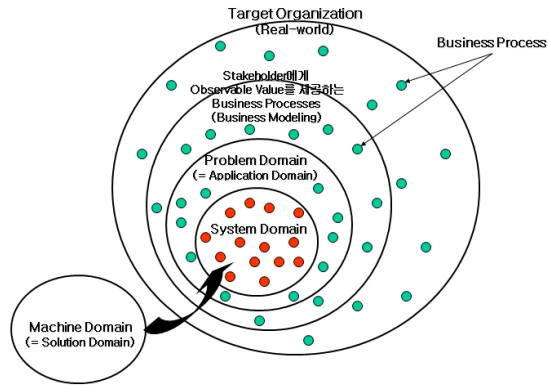


그림 3. 영역 정의
Fig. 3. Domain Definition

다음으로 고객이 설정한 전산화 목표에 기반하여 소프트웨어 요구사항 공학과정을 “요구사항 생성과 요구사항 관리” 단계로 구분한다. 요구사항 생성 단계는 다시 “비즈니스 프로세스 파악 → 비즈니스 모델링 → 문제분석 → 이해당사자 욕구 도출 → 시스템 특징 도출 → 요구사항 도출 → 요구사항 명세와 검증”의 세부단계로 분류한다.

체계적인 SRS 도출과정은 다음과 같이 수행한다.

I. 전산화 목표 파악

1. 전산화 목표 설정 : 비즈니스 재공학, 업무 향상 또는 전산화를 통한 업무 단순화, 비용 절감, 고객만족 향상

II. 요구사항 명세서 (SRS) 생성

2. 조직에서 수행되고 있는 모든 비즈니스 프로세스 파악 (AS-IS System) - [비즈니스 영역]
3. 비즈니스 모델링 - [비즈니스 모델링 영역]
 - 3.1 이해당사자 (액터) 확인
 - 3.2 이해당사자들에게 현저한 가치를 제공하는 비즈니스 프로세스들 확인
 - 3.3 비즈니스 유스케이스 모델링과 비즈니스 영역 모델링

4. 문제분석 - [비즈니스 모델링 영역]

- 4.1 문제 도출
- 4.2 비즈니스 모델링에서 문제영역 확인
- 4.3 문제분석 (문제의 근본 원인 및 해결 방안)
5. 이해당사자 욕구 도출 - [문제영역]
 - 5.1 문제영역에 기반하여 이해당사자 욕구 도출
 - 5.2 이해당사자 욕구 분석과 검증
6. 시스템 특징 도출 (To-Be System) - [시스템 영역]
 - 6.1 이해당사자 욕구에 기반하여 시스템 특징 도출
 - 6.2 현재 수행되는 비즈니스 프로세스에 대한 전산화 범위 (시스템 영역) 결정
7. 요구사항 도출 - [해법영역]

기능요구사항, 품질 (비기능 요구사항), 설계 제약사항
8. 요구사항 명세 (SRS) 및 검증

III. 요구사항 관리

9. 요구사항 관리 (사후 요구사항 추적 관리)

V. 결론 및 향후 연구과제

본 논문은 지금까지 제안된 요구사항 공학과정을 체계적으로 분석하고 최적의 과정을 제안하였다. 세부적으로는 요구사항 추적관리와 기원을 살펴보고, 요구사항 명세서를 도출하는 과정을 고객이 제시한 요구사항 명세서로부터, 요구사항 도출로부터, 문제분석으로부터, 비즈니스 모델링부터 수행하는 방법들을 고찰하였다. 또한, 이해 당사자 욕구 도출, 문제 분석, 비즈니스 모델링의 수행 우선순위에 대한 다양한 방법을 고찰하였다. 이들 결과로부터 “전산화 목표 파악 → 비즈니스 프로세스 파악 → 비즈니스 모델링 → 문제분석 → 이해당사자 욕구 도출 → 시스템 특징 도출 → 요구사항 도출 → 요구사항 명세와 검증” 과정을 수행하는 것이 가장 체계적으로 요구사항 명세서를 도출하는 과정임을 제안하였다. 이와 같이 체계적으로 요구사항을 도출하여 개발단계에서 관리하면 개발단계에서의 요구사항 변경을 최소화 시키고 프로젝트 성공률을 향상시킬 수 있으며, 고객의 만족도도 향상시킬 수 있을 것이다.

제안된 이론은 단지 지금까지의 연구 결과들을 종합 검토하여 최선의 방법을 제안한 것이며, 실제 프로젝트

에 적용하여 유용성을 검증하지는 못하였다. 따라서 추후 실제 검증과정을 거쳐 모델의 적합성을 판단할 계획이다.

References

- [1] E. Dijkstra, "The Humble Programmer," Communications of the ACM, 1972.
- [2] B. Lewis, "The 70-Percent Failure," Infoworld. <http://archive.infoworld.com/articles/op/xml/01/10/29/011029/opservival.html>, 2003.
- [3] Standish group, "The CHAOS Report," <http://www.standishgroup.com/sample/PDFpages/chaos1994.pdf>, 1994.
- [4] G. L. Ray, "IS 2510 Information Systems: The Requirements Problem," Department of Information Science, School of Information Sciences, University of Pittsburgh, <http://www.sis.pitt.edu/~gray/INFSCI2510/PowerPoints/01Lecture.ppt>, 2005.
- [5] IEEE, "IEEE STD 610.12-1990 IEEE Standard Glossary of Software Engineering Terminology," http://standards.ieee.org/reading/ieee/std_public/description/se/610.12-1990_desc.html, IEEE-SA, 2004.
- [6] I. Sommerville and P. Sawyer, "Requirements Engineering: A Good Practice Guide," John Wiley and Sons Ltd., 2000.
- [7] J. Suen, "ELEC3605: Engineering Software Requirements," http://www.users.on.net/~jsuen/elec3605_chp_summ_ts.doc, Sydney University, 2005.
- [8] R. Hossenlopp, "Diagnostic Modeling in Business Analysis: Solving the Right Problem is Key to Project Success," ESI International Inc., 2004.
- [9] K. Wiegers, "Software Requirements 2nd Edition," Microsoft Press, 2003.
- [10] J. Brackett, "Software Requirements Engineering Overview," Requirements Engineering course material, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA. 1992.
- [11] M. G. Matthews, "Software Requirements and Specification," <http://mason.gmu.edu/~mmatthe1/Fall01/CourseOverview.pdf>, The George Mason University, 2001.
- [12] D. Leffingwell and D. Widrig, "Managing Software Requirements," Addison-Wesley, 2000.
- [13] S. McEwen, "Requirements: An Introduction," <http://www-128.ibm.com/developerworks/rational/library/4166.html>, The Rational e-zine, 2004.
- [14] O. Ormandjieva, "SOEN 342 Software Requirements and Specifications," Department of Computer Science and Software Engineering, Concordia University, 2006.
- [15] O. Gotel and A. Finkelstein, "An Analysis of the Requirements Traceability Problem," Proceedings of the IEEE International Conference on Requirements Engineering, pp: 94-102, Colorado Springs, Colorado, 1994.
- [16] K. Bittner and I. Spence, "Establishing the Vision for Use Case Modeling," <http://www.awprofessional.com/articles/article.asp>, Addison Wesley Professional, 2002.
- [17] A. Holub, "The OO Design Process: Beginning to Design Software," The Rational E-zine, 2000.
- [18] B. Nuseibeh and S. Easterbrook, "Requirements Engineering: A Roadmap," <http://www.doc.ic.ac.uk/~ban/pubs/sotar.re.pdf>, ICSE, 2000.
- [19] A. Cahill, "CS5702 Software Engineering Requirements: Requirements Elicitation Sources of Requirements," Department of Computer Science and Information Systems, University of Limerick, 2004.
- [20] R. R. Young, "Implementing Effective Requirements Practices," Systems and Process Engineering, Northrop Grumman Information Technology, <http://www.ralphyoung.net/publications/ImplementingEffReqPractices.ppt>,
- [21] E. Whitney, "An Introduction to Gathering Requirements, Creating Use Cases and the UML," http://www.eps-software.com/download/whitepaper_the_development_process.pdf,

- Component Developer (CoDe) Magazine, 2001.
- [22] J. Huang, "Software Requirements," School of Computer Science, Telecommunications, and Information Systems, Depaul University, http://facweb.cs.depaul.edu/jhuang/se681/CSDP_RE.pdf, 2004.
- [23] R. Offen, "ITEC 821: Requirements Engineering," Division of Information and Communication Sciences, Macquarie, http://www.comp.mq.edu.au/units/itec821/ITEC821_s104_week1.pdf, 2004.
- [24] A. Finkelstein, "Requirements Engineering Research: Coordination and Infrastructure," Requirements Engineering, Vol. 1, pp: 63-69, 1996.
- [25] S. Feldman, "SE031 - Software Engineering: Application Requirements Determination," http://www.cs.bgu.ac.il/~se031/Slides/Lecture/select2_02.pdf, 2002.
- [26] Wikipedia, "The Free Encyclopedia," http://en.wikipedia.org/wiki/main_page. Wikimedia Foundation, 2007.
- [27] N. Iyengar, K. Kapoor, J. S. Greenstein, and K. Gramopadhye, "Data Gathering Methodologies to Identify Impact Variables in Aviation Maintenance," Human Computer Systems Lab., Department of Industrial Eng., Clemson University, <http://www.hf.faa.gov/docs/508/docs/maint/ERC2004WebSAT2.pdf>, 2004.
- [28] S. Giacoppo, "Development Methods: User Needs Assessment & Task Analyses," Department of Psychology, Catholic University, 2001.
- [29] R. Baghai, "Requirements Management: Building Quality from the Beginning," Rational Software Corporation, <http://www.seasim.org/archive/sim92001e.pdf>, 2002.
- [30] S. W. Ambler, "Agile Requirements Modeling," <http://www.agilemodeling.com/essays/agileRequirements.htm>, Amblysoft Inc., 2006.
- [31] RIT, "Why is Developing Software Hard?," Rochester Institute of Technology, <http://www.se.rit.edu/~se361/Slides/TJR-Requirements.pdf>
- [32] A. U. Rehman, "Requirement Engineering," <http://www.feditec.com/downloads/RequirementEng.ppt>, Feditec Enterprise, 2005.
- [33] K. Pohl, "Process Centered Requirements Engineering," John Wiley & Sons, Ltd., 1996.
- [34] P. Darke, "ISM9110 - Systems Analysis and Design: Information Gathering for Information Systems Development," School of Information Management and Systems, Monash University, 2005.
- [35] Z. Zhu, "Requirements Determination and Requirements Structuring," http://www.umsl.edu/~sauter/analysis/6840_f03_papers/zhu/, Information Systems, College of Business Administration, University of Missouri, 2003.
- [36] U. Magsarjav, "Requirements Documents Evolution and Synchronization with Activities in the Refined Requirements Generation Model," Master of Science in Computer Science and Applications, Blacksburg, Virginia University, 2004.
- [37] S. P. Maliyakkal, "Requirements Elicitation," Methoden und Werkzeuge zur Softwareproduktion, 1998.
- [38] B. Bruegge and A. H. Dutoit, "Object-oriented Software Engineering: Conquering Complex & Changing Systems," Prentice Hall, 2000.
- [39] D. C. Gause and G. M. Weinberg, "Are Your Lights On? How to Know What the Problem Really Is," Dorset House Publishing, New York, 1989.
- [40] L. Probasco, "The Ten Essentials of RUP: The Essence of an Effective Development Process," <http://www.devarticles.com/c/a/Development-Cycles/The-Ten-REssentials-of-RUP/>, IBM Corporation, 2001.
- [41] A. Davis, "Software Requirements: Objects, Functions, and States," Prentice-Hall, 1993.
- [42] D. Leffingwell. "Features, Use Cases, Requirements, Oh My!," <http://www.download.boulder.ibm.com/ibmdl/pub/software/dw/rationaledge/dec00/FeaturesUseCasesRequirementsOhMyDoc00.pdf>, The Rational Software, 2000.
- [43] A. Cofod, "System Analysis," Computer Science

- Instructor, Business Division, San Juaguin Delta Community College, 2005.
- [44] H. Krasner, "Requirements Dynamics in Large Software Projects: A Perspective on New Directions in the Software Engineering Process," 11th World Computer Congress (IFIP89), Elsevier, New York, 1989.
- [45] G. Kotonya and I. Sommerville, "Requirements Engineering: Processes and Techniques," John Wiley and Sons, 2000.
- [46] G. Kotonya and I. Sommerville, "Requirements Elicitation and Analysis," http://www.compapp.dcu.ie/~nbrophy/ca222/week4_6/som3.pdf, 1998.
- [47] A. S. Sidky, R. R. Sud, S. Bhatia, and J. D. Arthur, "Problem Identification and Determination Within the Requirements Generation Process," Department of Computer Science, Virginia Tech, 2002.
- [48] H. E. Eriksson and M. Penker, "Business Modeling with UML," <http://www.therationaledge.com/rosearchitect/mag/archives/fall99/f15.html>, The Rational E-zine, 1999.
- [49] B. Baker, "Business Modeling with UML: The Light at the End of the Tunnel," http://www.therationaledge.com/content/dec_01/m_businessModeling_bb.html, The Rational E-zine, 2001.
- [50] Rational Software, "Business Modeling with the UML and Rational Suite AnalystStudio," A Rational Software White Paper, 2002.
- [51] P. W. Ng, "Business Process Modeling and Simulation with UML," http://www.therationaledge.com/content/apr_02/t_businessProcessModeling_pn.jsp, The Rational E-zine, 2002.
- [52] B. Roggio, "CSCI 630 Software Engineering: Business Modeling - Domain Analysis," [http://www.ecst.csuchico.edu/~broggio/8.DomainModeling\(B027\).ppt](http://www.ecst.csuchico.edu/~broggio/8.DomainModeling(B027).ppt), 2005.
- [53] E. Ritter, "Fundamentals of Computer Science: Software Requirements," <http://www.cs.bham.ac.uk/~exr/teaching/lectures/se/04/Requirements.pdf>, School of Computer Science, University of Birmingham, 2004.
- [54] A. V. Lamsweerde, R. Darimont, and E. Letier, "Managing Conflicts in Goal-driven Requirements Engineerin," IEEE Trans. on Software Eng., Vol. 24, No. 11, pp. 908-926, 1998.
- [55] 강문석, "시스템 분석 및 설계," 정익사, 2004.
- [56] J. Amsden, ""Business Services Modeling," http://www-128.ibm.com/developerworks/rational/library/05/1227_amsden/index.html, Rational E-zine, 2005.
- [57] V. Muchandi, "Establishing Traceability - From Requirements to Design," First Software Services, http://www.fcg.com/EK/Images/FCG/Technology/FCGSS_WP_Trace-ability_022306.pdf, 2006.
- [58] K. Eustace, "SME Business Modeling Advice for Client- Server Application Developers," School of Information Studies, Charles Sturt University, <http://www.csu.edu.au/faculty/sis/subjects/itc382/sme.doc>, 2006.
- [59] J. Heumann, "Introduction to Business Modeling Using the Unified Modeling Language (UML)," <http://www-128.ibm.com/developerworks/rational/library/360.html>, The Rational E-zine, 2003.
- [60] G. Booker, "INFO 627 Requirements Engineering and Management: Analyzing the Problem," <http://users.snip.net/~gbooker/INFO627/lect2.ppt>, Drexel University, 2003.
- [61] J. Zdravkovic, "Identifying Problems," http://www.hig.se/~jzc/isa/isa2_2.pdf, Computer Science Department, Gävle University, 2006.
- [62] S. U. Lee, M. B. Choi, "A Definition and Evaluation Criteria for Software Development Success," Journal of The Institute of Internet, Broadcasting and Communication (IIBC), pp.233~241, vol. 12, no.2, April 2012.
- [63] D. S. Kim, H. C. Kim, "The Study of Software Reliability Model from the Perspective of Learning Effects for Burr Distribution," Journal of the Korea Academia-Industrial Cooperation Society, Vol. 12, No. 10 pp. 4543-4549, 2011.

저자 소개

이 상 운(정회원)



- 1998년~2001년 : 경상대학교 컴퓨터 과학과 (박사)
- 2007년 3월~현재 : 강릉원주대학교 과학기술대학 멀티미디어공학과 부 교수
- <주관심분야 : 소프트웨어공학 등>
- e-mail : sulee@gwnu.ac.kr

최 명 복(중신회원)



- 1997년~현재 : 강릉원주대학교 멀티 미디어공학과 교수
- 2004년 1월~현재 : 한국인터넷방송 통신학회 이사
- <주관심분야 : 알고리즘 등>
- e-mail : cmb5859@gmail.com