

논문 2022-2-3 <http://dx.doi.org/10.29056/jsav.2022.12.03>

# 소프트웨어 공학의 관점에서 소프트웨어 기성고 감정 필요성 및 방법론에 대한 고찰

김도완\*†

## A Study on the Need and Methodology for Appraisal of Ready-made Software from the Software Engineering Perspective

DoWan Kim\*†

### 요 약

개발프로세스가 종료된 소프트웨어를 대상으로 해당 소프트웨어의 품질을 다투는 소프트웨어 완성도 감정과, 소프트웨어 개발프로세스가 진행 중에 통합테스트 또는 인수시험에 이르지 못한 미완성 소프트웨어를 대상으로 하는 감정은 구분되어야 한다. 대법원 판례에 따르면, 예정된 최후의 공정까지 일을 종료한다는 점으로는 부족하고 프로그램의 주요기능 부분이 약정된 대로 개발되어 사회통념상 일반적으로 요구되는 성능을 갖추고 있을 때 완성이라고 하였다. 우리는 미완성 소프트웨어를 대상으로 하는 감정을 소프트웨어 완성도 감정과 분리하여, 소프트웨어 기성고 감정이라 한다. 소프트웨어 공학 개발 프로세스에서 도급자와 수급자의 협업 및 각각 역할이 요구되고 있으며, 소프트웨어 기성고 감정에서는 프로젝트 참여자(Skateholders)가 각자 주어진 역할을 어느 정도 어떻게 수행했는지가 평가되며, 개발 프로세스 단계에 따른 가중치를 소프트웨어 유형에 따라 부여하여 전체 기성고를 산정하도록 제시된다.

### Abstract

It is necessary to distinguish between the evaluation of software maturity, which relates to the quality of the software for which the development process has been completed, and the evaluation of unfinished software that has not reached integration testing or acceptance testing. According to the Supreme Court precedent, it is not enough to complete the work until the scheduled final process, and it is said that it is completed when the main functional parts of the program are developed as agreed and have the performance generally required in terms of social norms. We separate the evaluation of incomplete software from the evaluation of software completeness and call it the evaluation of ready-made software. The software engineering development process requires collaboration and specific roles among project stakeholders. A ready-made evaluation in software should evaluate how well the roles assigned to the project stakeholders have been performed. In addition, it is proposed to calculate the total established ratio by assigning weights according to the stages of the development process for each software type.

**한글키워드** : 소프트웨어 기성고 감정, SW완성도감정, SW하자, 소프트웨어 유형, SW공학 절차

**keywords** : SW ready-made rate appraisal, SW completeness evaluation, SW defect appraisal, SW types, SW Engineering process

\* 배재대학교 인터넷소프트웨어학과

† 교신저자: 김도완(email: dwkim@pcu.ac.kr)

접수일자: 2022.11.28. 심사완료: 2022.12.08.

게재확정: 2022.12.20.

※ 본 논문은 소프트웨어감정평가학회 2022년 추계학술  
대회 발표 논문을 수정 보완하여 투고한 논문입니다.

## 1. 서론

소프트웨어 프로젝트에서 실제 코딩이 이루어지는 기능구현이 소프트웨어 개발 프로세스의 중심이라는 관점은 현재 한국저작권위원회 소프트웨어 완성도 감정의 본체라 할 수 있다. 한국저작권위원회에 의뢰되는 소프트웨어 완성도 감정은 전체 감정의 30%를 차지할 정도이나, 한국저작권위원회에서 수행되는 소프트웨어 완성도 감정은 문제점을 안고 있다. 대표적으로, 첫째 문제점은 개발프로세스가 종료된 소프트웨어를 대상으로, 해당 소프트웨어의 품질을 다루는 완성도 감정이 아니라는 점이다. 두 번째 문제점은 개발 프로세스가 진행 중에 도급/수급 관계에서 개발 중단되거나, 통합테스트/인수테스트에 이르지 못한 미완성 소프트웨어를 대상으로도 같은 감정프로세스를 적용하여 완성도를 산출하는 것이다.

소프트웨어 완성도라 함은 프로그램의 질적인 완성도, 즉 “구현된 기능의 품질이 사용자의 요구를 얼마나 만족시키고 있는가?”이다. 예를 들어 ISO/IEC 9126은 소프트웨어 품질에 대하여 Functionality(기능성), Reliability(신뢰성), Usability(사용성), Efficiency(효율성), Maintainability(유지보수성) 그리고 Portability(이식성) 6개의 주특성과 주 특성에 따른 세부적인 23개 부특성을 규정하고 있다[2]. 이와 더불어 ISO/IEC 9241.10은 SW 사용성 품질제고를 위한 표준으로 작업적합성(Suitability for the task), 자명성(Self descriptiveness), 조종성(Controllability), 기대부합성(Conformity with user expectations), 에러견고성(Error tolerance), 개별화성(Suitability for individualization) 그리고 학습성(Suitability for learning) 7개 속성을 규정하고 있다[3]. 즉 소프트웨어 완성도는 통합 시험 및 인수시험이 성공적으로 이루어진 소프트웨어를 대상으로 그 품질의 완성도를 다루어야

한다.

완성된 소프트웨어의 품질을 다루는 완성도 감정과 달리, 소프트웨어 기성고 감정은 통합시험 또는 인수시험을 통과하지 못한 소프트웨어의 기성고 및 하자율을 산정하기 위하여, 미완성 건축물의 기성고 평가와 유사하게, 미완성된 소프트웨어를 대상으로 하여야 한다.

본 연구는 소프트웨어 기성고 감정의 당위성 및 감정 방법을 소프트웨어 공학에 기초하여 제안하고자 한다.

## 2. 관련 연구

### 2.1. 소프트웨어 개발 프로세스와 참여자의 책무

한국저작권위원회에 의뢰되는 소프트웨어 완성도 감정의 대부분은 소프트웨어 도급/수급관계에서 도급자와 수급자 사이에 발생하는 법적 분쟁의 결과물이다[4][5]. 즉 소프트웨어 개발 프로세스 상 통상적으로 수행되는 통합시험 및 인수 시험을 마치지 않은 채 분쟁 대상물이 된 소프트웨어에 대해서도 완성도를 다루고 있는 것이다 [5][6][7].

J. McManus & T. Wood-Harper 연구에 따르면 23.8%의 SW 프로젝트가 실패하거나 중도에 포기되어지며, 수행 완료된 SW 프로젝트 중 30% 이상 프로젝트는 개발자와 발주자 사이에 품질, 개발기간초과, 비용초과 등으로 인한 심각한 문제를 일으킬 수 있는 상태로 종결되었다 [1][14]. 실패한 소프트웨어 프로젝트의 원인에 대한 Standish Group 연구결과에 따르면 “불완전한 요구분석” 13.1%, “사용자(도급자)의 협력 결여” 12.45%, “도급자 요구사항 변경” 8.8%, “도급자의 비현실적 기대” 9.9%, “계획의 결여” 8.1% 등 비율로 그 원인이 분석되었다[8][9].

또한 연구 결과에 따르면 EU 국가들에서 수행된 소프트웨어 개발 프로젝트 214개 중 23.8%에 해당하는 프로젝트는 미완성인 상태로 중단된 것으로 분석되었다[8]. 단언적으로 소프트웨어 개발 도급/수급 관계에서 프로젝트 성공은 도급자의 충분한 협력이 없이는 불가능한 것으로 알려져 있다.

소프트웨어 공학에서 소프트웨어 개발프로세스는 일반적으로 ①계획(Feasibility Study)의 결과물로 도급자가 제공하는 RFP(Request for Proposal)에 따라, 수급자가 프로젝트를 수주하기 위하여 제시하는 제안서(Proposal)를 바탕으로 도급/수급 계약으로 진행되며, ②계약에 따라 도급자의 요구를 파악하기 위한 ③요구분석(Requirement Analysis)이 이루어진다. 요구분석 결과로서 요구분석서에 기초하여 소프트웨어/시스템 설계 및 디자인이 수행된다. ④수급자는 설계안을 토대로 유형의 소프트웨어를 개발한다. 다음 프로세스는 ⑤테스팅 및 통합이다. 테스팅 및 통합단계에서는 단위테스팅과 통합테스팅, 시스템테스팅 그리고 최종적으로 인수테스팅이 수행된다. 인수테스팅을 통과한 소프트웨어 프로젝트는 ⑥실제 환경에서 운영하며 버그 및 성능개선과 같은 유지보수 단계를 거쳐 완성된다[10][11][15].

소프트웨어 개발 프로젝트는 도급자/수급자 상호간의 협력이 매우 중요하며[12], 각각의 단계에 따라 충분한 도급자 역할 수행과 수급자의 역할 수행이 요구된다. 예를 들어 도급자의 적극적 협조 없이는 개발될 소프트웨어에 대한 요구분석이 이루어질 수 없으며, 불명확한 요구분석 결과에 따른 소프트웨어/시스템 설계 및 디자인은 처음부터 문제점을 안은 채 시행되어 결국 도급자가 만족하지 못하는 개발 상태로 이어진다. 따라서 도급/수급 관계에서 분쟁 발생 시 분쟁의 원인이 되는 책임 소재를 규명하는 것이 필수적

이라 여겨진다[12][15][16]. 분쟁의 책임소재를 규명하는 것이 중요한 이유는 특히 기성고 감정에서 개발 프로세스 단계 별 사업 참여자의 역할이 다르며, 전체 기성고에 대한 개발 프로세스 별 가중치가 있기 때문이다. 학자에 따라 다소 편차는 있지만, ②요구분석 단계 15%~30%, ③설계단계 10%~20%, ④구현단계 30%내외, ⑤테스팅 및 인수시험 20~30% 소프트웨어 프로젝트 각 단계 별 비용 분포가 제시되고 있다[10][11]. 판례에서도 하자발생의 원인 및 개발 프로젝트에서 사업 참여자의 역할 수행 여부를 다루고 있는 점을 다수 찾아 볼 수 있다[5][6][7][16].

다른 하나의 관점은 소프트웨어 유형에 따라 다양한 개발 형태가 존재한다는 점이다. 수급자가 기 확보하고 있는 소프트웨어 소스코드를 도급자의 요구에 맞추어 커스터마이징 하는 프로젝트의 경우 ②요구분석 단계와 ⑤테스팅 및 인수 시험에 소모되는 비용은 ④구현단계 비용보다 클 가능성이 높다. 반면에 모든 소프트웨어 기능에 따른 소스코드가 신규 개발되어야 한다면, 매우 구현하기 어려운 기능 및 알고리즘을 요구하는 프로젝트의 경우 ④구현단계 비용은 다른 개발 단계보다 높아질 수밖에 없으며, 이러한 점들이 기성고 감정에서 종합적으로 고려되어야 한다.

## 2.2. 소프트웨어 기성고와 건축/건설 기성고

‘기성고’ 또는 ‘기성고율’ 개념은 건설-건축 도급/수급계약관계에서 건설비용의 지불, 또는 법적 분쟁이 발생하였을 때 쓰이는 개념으로, 시공된 공정 비율 감정에 따른 댓가 산정 목적으로 이용된다[13]. 기성고는 기본적으로 미완성을 전제로 하고 있으며, 기성고가 100%일 때는 완성된 상태로 수급자는 하자보수의 의무를 가지는 반면, 도급자는 보수지급의 의무를 가진다[12]. 건축-건설 공정에서는 각 단계별 전문적이고 실

체적인 결과물이 있어 기성고율 산정 및 분쟁의 책임소재 규명이 비교적 명확한 반면, 소프트웨어 개발 공정에서는 도급자와 수급자의 협업에 따라 대부분 개발공정이 진행되며 결과물이 구체화되는 특징을 가진다[11].

한국저작권위원회에 의뢰된 소프트웨어 완성도 감정 관련 분쟁 유형은 도급자와 수급자 사이에 계약사항의 완전 이행 여부를 다투는 행태를 보이고 있다. 이에 따라 판결의 청구 유형 또한 채무부존재확인, 용역대금청구, 손해배상, 대금반환, 부당이익금 반환 등과 같은 소송 판결이 주를 이루고 있으며, 판결의 기초로 삼기위하여 감정 전문가에게 해당 시스템에 대한 “작업진척도”(기성고) 감정을 요청하고 있다[5][6][7][16].

### 3. 소프트웨어 기성고 감정 방법

본 연구는 올바른 기성고 감정을 위해 2단계 감정 방법을 제안한다. 첫째 소프트웨어 개발 프로세스의 참여자(Stakeholder)의 책무 분담과 그 책무의 수행 정도 및 여부를 평가 감정하여 프로젝트 미완성의 원인을 밝힌다. 둘째 소프트웨어 개발 전체 프로세스에서 각 단계 별 가중치를 부여하여 기성고율을 산정한다. 최종적으로 1단계 참여자의 책임 수행 여부 비율과, 2단계 산정된 기성고율을 종합하여 전체 소프트웨어 기성고 산정 방법을 제안한다.

#### 3.1. 소프트웨어 개발 프로세스 별 참여자 역할 수행 평가 감정

도급/수급관계에서 소프트웨어 프로젝트 성공/실패의 핵심 열쇠는 프로젝트 참여자의 협력과 소통이다. 소프트웨어 프로젝트가 미완성 상태에서 법적 다툼으로 진행되는 원인의 절반은 ②요구분석 결과의 부실(20%)에서 출발하며, 이는 ③

설계 및 디자인의 오류(30%)를 일으킨다. ④구현 단계에서 수급자의 역량부족으로 발생하는 문제가 나머지 절반을 차지한다고 볼 수 있다[11]. 그리고 ②요구분석 단계에서의 오류는 프로젝트 프로세스 전체에 걸쳐 참여자 간의 갈등과 비용을 야기시킨다[14].

표 3-1 SW공학에서 개발프로세스와 참여자의 역할 비율 예시

Table 3-1 Example of the role ratio of the development process and stakeholders in SW engineering

프로세스	세부 프로세스	도급자 역할 (비율)	수급자 역할 (비율)	결과물
요구분석		60	40	요구분석서
설계	SW설계	20	80	SW설계서
	시스템설계	50	50	시스템 사양서
	디자인	20	80	UI 디자인
구현	코딩	0	100	소스코드
	단위 테스트	0	100	
테스팅	통합시험	50	50	통합시험서
	인수시험	80	20	인수시험서
유지보수	해당 사항 없음			

위 표 3-1은 소프트웨어 개발 프로세스와 참여자의 역할 비율 및 결과물을 보여주고 있다.

감정인은 각 개발 프로세스 단계별 참여자의 역할 비율을 도급/수급 계약관계 및 유관 문서 분석을 통하여 조정할 수 있다. 요구분석 단계 결과물인 요구분석서는 인수시험의 자료로 활용될 수 있으며, 설계단계 결과물들은 단위시험 및 통합시험 검증의 기초 자료이다. 따라서 소스코드 코딩 구현을 위하여 요구분석서와 설계단계 결과물들은 도급자의 승인을 필요로 한다. 소프트웨어 개발 프로젝트가 도급자의 적극적인 협조와 상호소통 없이 성공하기 어렵다는 것을 감안할 때, 도급자의 역할 수행은 수급자 못지않게 중요하다는 것을 알 수 있다. 개발될 소프트웨어의 기능모델/동작모델/정보모델 또는 모든 유즈케이스가 전체적으로 포함 분석되어 도급자의 승인을 취하였다면 요구분석 단계는 성공적으로 이루어졌다 할 수 있다. 설계단계는 요구분석서에 기초하여 전반적으로 수급자의 주도 아래 이루어진다. 다만 소프트웨어가 실행되는 환경인 시스템 설계는 계약관계에 따라 전적으로 도급자가 책임을 지든지? 또는 수급자가 소프트웨어와 함께 납품할 수도 있기 때문에 계약관계를 살펴야 한다. 도급자는 SW설계서와 UI디자인을 검토하여 승인 또는 변경을 요구할 책무가 있다. 구현 단계는 전반적으로 수급자의 책무이다. 테스트에서 통합테스팅은 통상적으로 테스트 시나리오 또는 요구분석서를 토대로 하여 수급자 주도로 이루어진다. 도급자는 통합시험 결과를 분석하여 발견된 문제점을 해결하도록 수급자에게 요구하여야 한다. 인수테스트는 도급자 주도로 이루어져야 한다. 인수테스트가 성공적으로 이루어져 도급자에게 소프트웨어 납품이 이루어지고 운영에 들어가면 수급자는 계약 상 정해진 유지보수의 의무가 있을 수 있으며 전체 개발 프로세스는 성공한 것으로 본다.

소프트웨어 개발 프로세스 별 참여자 역할 수

행 감정은 도급/수급 계약서 및 개발 단계별 결과물 문서를 토대로 감정인이 판단할 수 있으며, 문서 상으로 판단하기 어려운 경우 이메일, SNS, 증언 등 정황 증거를 바탕으로 소프트웨어 공학에서 언급되는 참여자의 역할 비율 예시<표3-1>를 참고하여 감정할 수 있다. 감정 결과는 각각의 개발 단계에서 프로젝트 참여자의 역할 수행 여부를 정성적/정량적으로 제시되어야 한다. 예를 들어 요구분석 결과물이 없거나 불완전하다면 발주자와 수급자 누구에게 어떠한 책임이 있는지 정성적으로 밝히고, 정량적 비율로 이를 평가한다.

그 결과는 도급자/수급자에게 주어진 역할을 각각 전체 100%로 하였을 때, 몇%를 수행하였는지 이다.

### 3.2. 소프트웨어 개발 프로세스 별 가중치 산정

소프트웨어 유형에 따라 소프트웨어 개발 프로세스 별 가중치 산정은 달라져야 한다. 예를 들어 S\*P사의 ERP(Enterprise Resource Planning)시스템과 같은 소프트웨어 유형은 대부분의 시스템 기능이 구현되어 존재하며, 고객(도급자)의 요구에 맞게 수급자에 의해 커스터마이징되는 형태로 공급된다. 이러한 유형의 개발 프로젝트에서는 요구분석 단계가 전체 프로젝트의 절반을 차지하는 핵심 개발 행위에 해당하며, 반면 구현 단계는 비교적 단순한 작업이 될 수 있다. 이와 달리 어떤 소프트웨어는 전체적으로 소스코드가 신규로 개발되어야 하거나, 구현하기 매우 까다로운 알고리즘을 요구할 수 있다. 이러한 소프트웨어 유형은 전체 개발 프로세스에서 구현 단계에 들어가는 비용이 여타 프로세스에 비하여 가장 크게 발생함으로 높은 가중치를 가져야 한다. 다음 <표3-2>는 소프트웨어 개발 프로세스 별 가중치 범주를 제시하고 있다.

표 3-2 SW 유형별 개발 단계 가중치 범주 예시  
Table 3-2 Example of SW process stage weight category by SW type

프로세스	가중치 범주(%)	결과물
요구분석	15 ~ 30	요구분석서
설계/디자인	10 ~ 20	설계/디자인 시안
구현	20 ~ 50	소스코드
테스팅/인수 시험	10 ~ 30	시험 결과서
유지보수	해당사항 없음	

감정인은 전문적 식견에 기초하여 각 단계별 가중치를 <표3-2> 범주 내에서 총100%로 결정 한 후, 결과물을 기초로 하여 작업의 완전성을 판단하여 기성고 비율 산정을 할 수 있다. 예를 들어 요구분석 단계 30%, 설계단계 20%, 구현단계 30%, 테스트 20% 가중치를 부여하였다면, 각각 단계별 작업의 완전성을 평가하여 전체 기성고율을 감정한다.

### 3.3. 소프트웨어 종합 기성고 산정 방법

3.1에서 살펴본 바와 같이 소프트웨어 프로젝트의 실패 원인의 시작은 프로젝트 참여자의 역할 결여에서 출발한다. 소프트웨어 공학에서는 프로젝트 참여자의 협업과 소통없이 성공적인 소프트웨어 프로젝트를 기대하기 어렵다고 정의하고 있다. 따라서 SW 기성고 감정에서는 개발 프로세스에서 요구되는 프로젝트 참여자의 역할이 수행되었는지? 수행되었다면 어느 정도 비율로 평가되는지 산정하고, 미완성에 이르게 된 원인에 대한 명확한 감정을 정성적으로 기술하여야 한다.

두 번째로 기성고율 산정을 하여야 한다. 기성고율은 <표3-2>를 참조하여 소프트웨어 개발 단계별 결과물에 근거하는 평가를 수행하고, 소프트

트웨어 개발 유형에 따라 프로세스 단계에 대한 가중치를 부여하여 산정한다.

따라서 소프트웨어 종합기성고 감정보고서는 SW 미완성에 이르게 된 원인과 그 책무에 대한 비율을 제시하며, 최종적으로 SW 기성고율을 산정함으로써 감정의 목적성 (법적 분쟁 판결의 전문 자료제공)을 갖추도록 한다.

## 4. 결 언

법적의미에서 SW완성도는 완성된 소프트웨어의 품질 정도를 의미한다. 따라서 완성도 감정과, 통합시험/인수시험에 이르지 못한 미완성 SW 감정은 구별되어야 한다.

통합시험/인수시험을 통과하여 유지보수 단계에서 발견되는 SW 기능상의 하자 또는 버그와 사용상의 기능 개선은 모든 소프트웨어에서 나타나는 숙명이다. 그 정도가 본래 목표하였던 소프트웨어 성능을 현저히 만족시키지 못하여 법적 분쟁 대상이 된 소프트웨어 대하여 소프트웨어 완성도 감정은 자명하다.

위와 달리 프로젝트 수행 중 중단되거나, 통합 시험/인수시험에 이르지 못한 소프트웨어를 대상으로 법적 판단을 구하는 감정의 경우, 미완성에 이르게 된 원인을 판단하고, 단순히 기능의 구현 비율을 산정하는 것이 아니라, 전체 개발 프로세스에서 각 단계별로 구현 비율을 산정하고 프로세스 가중치를 적용하여 SW기성고율을 산정하는 것이 올바르다 판단된다.

향후 연구에서는 소프트웨어 유형을 보다 체계적으로 분류하여, 프로세스 별 가중치를 세밀하게 제시할 수 있도록 하는 연구가 필요하며, 프로젝트 참여자(도급자/수급자)의 역할에 대한 각 개발 프로세스 단계에서 비율 또한 다양한 관점에서 연구가 필요하다.

참 고 문 헌

- [1] J. Mcmanus, T. Wood-Harper. (2007). Understanding the Sources of Information Systems Project Failure. Journal of the management services institute, Vol.51(2). URL:https://www.researchgate.net/profile/John-Mcmanus-11/publication/329539985\_A\_study\_in\_project\_failure/links/5c1138aea6fdc494fef184b/A-study-in-project-failure.pdf
- [2] ISO/IEC9126. URL: https://en.wikipedia.org/wiki/ISO/IEC\_9126
- [3] G.Daeuble et.al, Usability Evaluation. (2014) in:Dienstleistungsproduktivitaet, Springer. DOI: 10.1007/978-3-658-05301-7\_9
- [4] S.Y. Kim. (2013). A study of appraisal system on copyright : the effect and the improvement. Korea copyright commission
- [5] Copyright-related appraisal case law book (1). (2010). Korea copyright commission
- [6] Copyright-related appraisal case law book (2). (2013). Korea copyright commission
- [7] Copyright-related appraisal case law book (3). (2016). Korea copyright commission
- [8] The chaos report. (1995). standish group international inc. URL: http://ftp.it.murdoch.edu.au/units/ICT327/2005%20Semester%201/02.%20Lectures/Topic%2001%20Introduction%20to%20Project%20Management/chaos1994%20report%20on%20project%20failure.pdf
- [9] The chaos report. (2020). standish group international inc. URL: https://www.standishgroup.com/news/45
- [10] I. Sommerville. (2011). Software Engineering, 9th Edition, Edison Wesley. URL: https://engineering.futureuniversity.com/BOOKS%20FOR%20IT/Software-Engineering-9th-Edition-by-Ian-Sommerville.pdf
- [11] E.M. Choi. (2022). All about software engineering, Saengnung publ.
- [12] TaeKyung Kim. (2012). The regal issues of the personalized computer software's "development and supply to order" contract as a softwares bug, in: Journal of korea bar association, Vol. 425. URL: https://www.koreanbar.or.kr/pages/board/view.asp?teamcode=&page=17&seq=6757&types=11&category=&searchtype=&searchstr=
- [13] G.C. Jeong. [Construction work] Relationship between completion rate, ready-made amount, and ready-made rate (3). URL: https://www.koscaj.com/news/articleView.html?idxno=205647
- [14] J. MacManus & T. Wood-Harper, A study in project failure. URL: https://www.bcs.org/articles-opinion-and-research/a-study-in-project-failure/
- [15] R. Gabriel. The Rise of "Worse is Better". URL: https://web.stanford.edu/class/archive/cs/cs295/cs295.1086/
- [16] Supreme Court 2006. 1 Judgment of 0.13 2004Da21862

저 자 소 개



김도완((DoWan Kim)

1990.2 독일 레겐스부르크대학교 정보공학 학사  
 1992.8 독일 레겐스부르크대학교 정보공학 석사  
 1996.6 독일 레겐스부르크대학교 정보공학 박사  
 1996.8~1997.2 한국전자통신연구원 선임연구원  
 1997.3~현재 배재대학교 소프트웨어학과교수  
 2012.2~2018.8 독일 힐데스하임 대학교 객원교수  
 2006.8~현재 한국저작권위원회 SW감정전문위원  
 <주관심분야> SW저작권, SW인간공학, 시맨틱웹, 지식기반시스템, 빅데이터 처리