

소프트웨어 완성도 감정을 위한 요구사항 완전성 평가

김유경*†

A Study on the Completeness Measurement of Requirements Specifications for Software Completion Appraisal

Yukyong Kim*†

요 약

소프트웨어 요구사항 (Software requirements)은 소프트웨어 개발 프로젝트의 작업, 비용, 기간을 정량화한 후 합의한 것으로, 향후 분석, 설계, 구현, 유지보수 전 과정에서 의사결정의 중요한 판단 기준이 된다. 소프트웨어 요구사항은 발주자와 개발자 간의 계약 역할을 하면서 양 당사자가 개발하고자 하는 제품의 사양과 기대 수준을 이해하여 도출하므로 소프트웨어 개발에 필수적인 항목이다. 소프트웨어 완성도 분쟁의 대부분은 계약 관계에 있는 개발자와 발주자 사이에서 최종 납품된 제품에 대한 기능적 이견에서 주로 비롯된다. 이에 따라 현재의 소프트웨어 완성도 감정은 개발계약서에 포함된 고객의 요구사항이 구현되어 제대로 작동되는지 확인하는 방식이다. 분쟁의 원인이 되는 책임소재를 규명하는 것이 필요한 만큼, 사업참여자로서 발주자의 역할 수행이 충분히 이루어진 상태에서 완전한 요구사항이 제공되었는지를 살펴보는 것은 매우 중요하다. 본 논문에서는 요구사항 명세의 완전성 평가 방안을 살펴보고, 취업포털사이트 완성도 감정사례에서 드러난 요구사항 명세의 문제점을 통해, 소프트웨어 완성도 감정에서 활용 가능성을 중심으로 그 의미를 논의하고자 한다.

Abstract

Software requirements are agreed upon after quantifying the work, cost, and duration of a software development project and serve as a contract. It is an essential item for software development because it is derived by understanding the specifications and expectations of the product that both parties, the developer and the customer, want to develop. Most software completeness disputes arise mainly from functional differences between the developer and the customer about the final delivered product. Accordingly, the current software completeness appraisal verifies that the customer's requirements included in the contract are implemented and operated properly. As it is necessary to identify the source of responsibility that causes the dispute, it is very important to examine whether the role of the client as a project participant has been sufficiently fulfilled and complete requirements have been provided. In this paper, we look at the completeness evaluation method of requirements specification and discuss the possibility of using it in software completeness evaluation through the problems of requirements specification revealed in the case of job portal site completeness appraisal.

한글키워드 : 소프트웨어 요구사항, 완전성 평가, 완성도 감정, 요구명세, 요구공학

keywords : SW requirements, completeness metrics, completeness appraisal, requirement specification, requirement engineering

* 숙명여자대학교 기초공학부

접수일자: 2023.03.06. 심사완료: 2023.03.15.

† 교신저자: 김유경(email: ykim.be@sm.ac.kr)

게재확정: 2023.03.20.

1. 서론

소프트웨어 감정은 법적 분쟁이 발생했을 때 저작물의 실질적 유사성이나 소프트웨어 완성도, 소프트웨어 개발 비용 등 감정신청인의 감정 요청사항에 대해 소프트웨어를 전문적으로 분석하고 판단하여 그 결과를 제시하는 것이다. 특히 소프트웨어 완성도 감정의 경우에는 감정 목적물인 대상 시스템 및 관련 산출물의 법적인 유효성을 검토하고, 감정 목적물에 대한 개발공정의 진척도 및 실제 작동의 검증을 통해 완성 및 하자의 정도를 평가한다.

소프트웨어 완성도 감정의 핵심적인 사항은 발주자의 요구사항에 대한 소프트웨어의 구현 여부 및 실제 작동 여부에 대한 평가이다. 이를 위해서 먼저 제안서, 개발계약서, 요구사항 명세서, 데이터 및 기능 설계서, 프로젝트 운영 상황에 대한 기록 등을 분석하고 검토하여 전체적인 소프트웨어 요구사항을 도출한다.

소프트웨어 요구사항 (Software Requirement Specification)은 소프트웨어 개발 프로젝트의 과업, 비용, 기간을 정량화한 후 합의하여 향후 분석, 설계, 구현, 유지보수 전 과정에서 의사결정의 중요한 판단 기준이 된다. 소프트웨어 요구사항은 발주자와 개발자 간의 계약 역할을 하면서 양 당사자가 개발하고자 하는 제품의 사양과 기대 수준을 이해하여 도출하므로 소프트웨어 개발에 필수적인 항목이라고 할 수 있다 [1]. 그림 1은 요구공학에서 정의하고 있는 요구사항 개발단계와 요구사항 관리단계를 나타낸다 [2].

그렇지만, 그 중요성에 비해 국내에서는 제대로 활용되지 못하고 있다. 처음부터 애매한 요구사항으로 RFP(Request For Proposal)를 공고하고, 계약 체결 과정까지 명세 수준으로 구체화하지 못한 상태에서 프로젝트를 일단 시작하는 주먹구구식 개발 관행이 일반적이었기 때문이다[3].

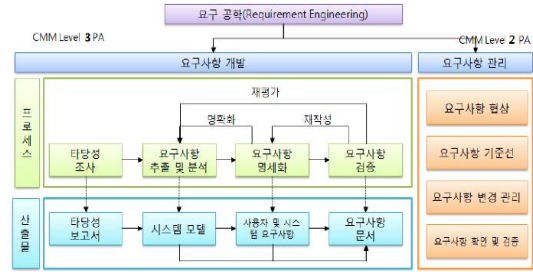


그림 1. 요구공학 절차
Fig. 1. Requirement engineering process

소프트웨어 개발 프로젝트는 발주자와 개발자 상호 간의 협업이 중요한데, 특히 개발될 소프트웨어에 대한 요구분석은 발주자인 고객의 협조가 없이는 제대로 이루어지기 힘들고, 불명확한 요구분석 결과에 따른 소프트웨어 설계 및 구현은 처음부터 문제점을 안은 채 시행되어 결국 고객이 만족하지 못하는 개발 상태로 이어진다 [4].

소프트웨어 완성도 분쟁의 대부분은 계약 관계에 있는 개발자와 발주자 사이에서 최종 납품된 제품에 대한 기능적 이견에서 주로 비롯된다. 이에 따라 현재의 소프트웨어 완성도 감정은 개발계약서에 포함된 고객의 요구사항이 구현되어 제대로 작동되는지 확인하는 방식으로 이루어진다. 분쟁 발생 시 분쟁의 원인이 되는 책임소재를 규명하는 것이 필요한 만큼, 사업참여자로서 발주자의 역할 수행이 충분히 이루어진 상태에서 완전한 요구사항이 제공되었는지를 살펴보는 것은 매우 중요하다. 이는 하자발생의 원인이나 사업참여자의 역할 수행 여부를 다투고 있는 최근 판례를 통해서도 알 수 있다 [5][6][7].

본 논문에서는 요구사항 명세의 완전성 평가 방안을 살펴보고, 취업포털사이트 완성도 감정 사례에서 드러난 요구사항 명세의 문제점을 통해, 소프트웨어 완성도 감정에서 활용 가능성을 중심으로 그 의미를 논의하고자 한다. 본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로서

취업포털사이트 완성도 감정 사례에서 나타난 요구사항 명세의 문제점을 통해, 소프트웨어 요구사항 명세의 중요성과 역할에 대해 살펴보고, 3장에서는 요구사항의 완전성 평가 방법을 제시한다. 4장에서는 완성도 감정에서 요구사항 완전성 평가 방법을 활용하는 방안에 대해 논의하고 5장에서 결론을 맺는다.

2. 감정사례로 살펴본 소프트웨어 요구사항의 중요성

본 논문에서 소프트웨어 요구사항 중요성의 사례로 다루게 되는 감정 대상 목적물은 구인 구직 서비스를 제공하는 취업포털 사이트와 개발문서이다. 피고인 개발자가 원고인 발주자에게 개발하여 납품한 개발 결과물의 완성도 및 수행 가능 여부 뿐만 아니라, 개발 문서의 제공 여부 및 수준에 대한 감정요청사항을 포함하고 있다. 보통 감정 당시 개발 결과물에 대한 완성도의 판단 기준은 목적물의 주요 부분이 약정된 대로 구현되어 사회 통념상 일반적으로 요구되는 성능을 갖추고 있어야 한다는 것이다. 그러나 본 감정 사례는 개발 범위에 대한 원고와 피고의 이견이 분쟁의 주요 원인이며, 이는 개발결과물이 갖추고 있어야 하는 약정된 내용 자체에 대한 불완전성에서 기인한 것이다. 소프트웨어 개발 과정에서 요구사항 명세는 개발자와 발주자 사이의 개발범위에 대한 약정이며, 개발이 완료된 후 개발 결과물의 완성도를 평가할 수 있는 척도가 되므로 요구사항 명세의 불완전성은 완성도에 대한 분쟁을 야기하게 된다.

감정 대상인 취업포털사이트는 개발자가 개발 용역 범위를 해당 사이트의 화면 기획과 스토리 보드를 설계서로 정의하여 이를 제시하고 개발을

진행하였다. 계약서 상의 개발 완료 시점이 지난 이후 원고인 발주자가 개발 미흡을 주장하는 가운데, 피고측에서 완료의 기준으로 삼았던 개발 대상 소프트웨어 기능목록과 설계서가 증거로 제출되었다. 본 감정 사례에서 쟁점은 설계서에 나타난 항목들이 실제 시스템에 구현된 기능 항목으로 모두 포함되어 있는지, 그리고 제작되어 제대로 작동되고 있는지의 여부였다.

감정은 취업포털사이트에 직접 접속하여, 감정 목적물의 총 19개 메뉴기능을 감정대상으로 정의하고, 감정 대상 메뉴별 항목인 총 794개를 일일이 실행하여 작동 여부를 확인하였다. 감정 결과 총 794개 기능항목 가운데 완성된 항목은 706개로 확인되어 완성도는 약 89%로 판단하였다.

그러나 구현된 기능항목 794개 가운데, 370개 항목은 설계서에 나타난 항목들과 일치하지 않았으며, 이 부분이 발주자와 개발자 사이에 이견이 발생한 부분이다. 이것은 구현된 기능항목 794개를 기준으로 46.6%에 해당한다.

또한 설계서에 나타난 기능항목은 총 652개였으며, 이중 228개 항목이 구현된 기능항목과 일치하지 않았다. 설계서를 기준으로 약 35%에 해당한다. 이를 근거로 발주자는 설계서에 적시된 기능의 약 35%가 구현되지 않았다고 보고 있었다. 그러나 개발자측에서는 통상적으로 요구되는 기능으로 구현되어, 발주자가 원하는 결과를 거의 모두 제공하고 있으므로 개발이 완료된 것으로 보고 있었다.

이는 개발 소프트웨어에 대한 요구사항 명세인 설계서에 해당 기능에 대한 메뉴명과 같은 항목명만 제시된 상태로 상세수준으로 설명되지 않았기 때문으로 분석된다. 본 감정 사례를 통해, 발주자와 개발자 사이의 개발대상 소프트웨어의 완성 수준에 대한 이견을 좁히기 위해서는 완전한 요구사항 명세가 매우 중요함을 알 수 있다.

3. 요구사항의 완전성 평가 방법

3.1 요구사항의 완전성

INCOSE는 요구사항을 제대로 작성하기 위한 가이드라인을 제시하고 있으며, 완전한 요구사항의 특성은 분명함(clear), 원자적(atomic), 비모호성(unambiguous), 증명가능성(verifiable), 필요성(necessary), 디자인 독립(design independent), 실현가능성(feasible), 완전성(complete), 정확성(correct)이다 [8].

분명함이란 요구사항은 명확하고 읽기 쉽고 이해하기 쉬워야 한다는 특성이다. 모든 요구사항은 명확한 성공 기준을 설명해야 하고, 간단한 어휘를 사용하고 약어를 피해서 작성한다. 원자적의 의미는 각 요구사항은 개별 테스트 케이스로 취급되어야 한다. 복잡한 요구사항을 개별적으로 테스트할 수 있을 때까지 더 작은 부분으로 나누어야 한다. 비모호한 특성으로 인해 불분명하거나 불완전하거나 모순되는 요구사항은 오류 및 재작업으로 이어질 수 있다. 이해관계자 사이의 격차를 줄이기 위해서는 요구사항이 완료되기 전에 모든 이해관계자가 요구사항을 검토해야 한다. 증명가능성은 개발 팀의 모든 사람은 문서에 액세스하여 필요할 때마다 참조할 수 있어야 한다는 특성이다. 필요성은 각 요구사항은 사용자가 실제로 필요로 하거나 외부 인터페이스의 존재로 인해 표준 또는 통합 요구사항을 충족하는데 필요한 것을 문서화해야 한다는 것을 의미한다. 또한 각 요구사항에는 승인된 소스가 있어야 한다. 디자인 독립은 각 요구사항은 구현 방법이 아니라 무엇이 필요한지를 정의해야 함을 나타낸다. 요구사항은 내부 세부 사항이 아니라 외부에서 관찰될 시스템의 특성을 정의하는 것이다. 실현가능성은 각 요구사항은 기술적으로 구현 가능해야 하며 프로젝트에 영향을 미치는 예산, 마감일 및 기타 제한 사항을 고려하여 구현되어야 한

다는 의미이다. 완전성은 요구사항 문서에는 개발 팀과 테스터가 제품을 완성하고 버그 없이 사용자의 요구사항을 충족하는지 확인할 수 있는 충분한 정보가 포함되어야 한다는 것이다. 정확성은 문서에 명시된 요구사항은 혼동을 피하기 위해 정확하게 기술되어야 함을 나타낸다. 올바른 요구사항을 작성하려면 올바른 정보를 얻고 수집된 정보를 올바르게 문서화 해야 한다.

특히 완전성은 소프트웨어 완성도 분쟁의 주요 원인으로, 소프트웨어 프로그램이 의도한 요구사항과 기대치를 충족하고 시간과 예산 내에서 제공되도록 하는 데 매우 중요하다. 완전하고 정확한 소프트웨어 요구사항은 소프트웨어 개발자, 프로젝트 관리자 및 발주자와 같은 이해관계자 간의 커뮤니케이션 및 협업을 촉진하는 데도 매우 중요한 역할을 한다. 그림 2는 소프트웨어 요구사항의 완전성이 여러 요인과 관련되어 있음을 보여주고 있다 [1].

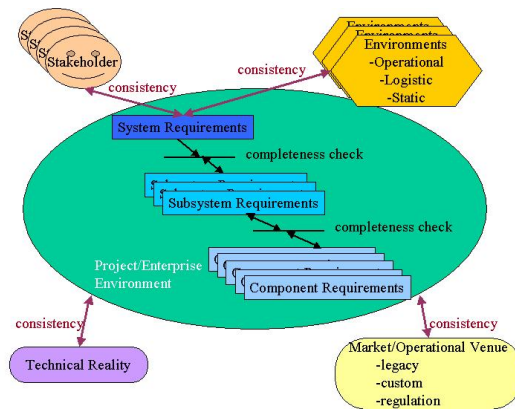


그림 2. 요구사항의 완전성 컨텍스트
Fig. 2. Requirement completeness context

3.2 요구사항의 완전성 평가 방법

소프트웨어 요구사항에 대한 완전성 지표는 요구사항 명세(specification)의 정확성, 완전성 및 포괄성을 평가하는 데 사용되는 척도를 말한

다. 요구사항의 완전성은 모든 요구사항에 대한 설명이 나타나 있음을 표현하는 것이지만, 모든 요구사항에 대한 설명이 있는 것만으로는 완전성을 나타내기에 충분하지 않다. 요구사항의 설명에는 해당 요구사항의 기능적 및 비기능적 요소, 입력 속성 및 결과와 관련된 세부 정보가 포함되어야 한다. 소프트웨어 요구사항에 대한 몇 가지 일반적인 완전성 지표는 다음 표1과 같다 [9].

표 1. 요구사항 완전성 지표
Table 1. Requirements completeness metrics

	항목
1	설명에 있는 요구사항의 비율
2	기능적 요소의 설명을 반영하는 요구사항의 비율
3	기능적 요인
4	입력 매개변수의 설명을 반영하는 요구사항의 비율
5	결과와 관련된 속성의 설명을 반영하는 요구사항의 비율
6	문서화 형식의 모든 척도를 준수하는 요구사항의 비율

또 다른 요구사항 완전성 확인 방법으로는 기능점수 분석(FPA, Function Point Analysis)방법이 있다. 소프트웨어 규모 산정 및 비용 산정에서 주로 사용되는 FPA는 구조화된 요구사항 검토를 위한 유용하고 구조화된 방법으로 인기를 얻고 있다 [10]. 기능점수는 LOC(Lines of Code)로 측정되는 물리적 구현과 달리 소프트웨어 프로젝트의 논리적 기능의 크기를 측정한다. FPA는 소프트웨어에서 제공할 기능적 사용자 요구사항을 검사하고, 논리적 사용자 기능에 가중된 기능점수를 할당하여 기능점수 크기를 계산한다. 즉, FPA는 사용자가 이해하고 표현한 대로 데이터 및 트랜잭션 측면에서 일련의 기능적 사용자 요구사항을 검사하게 되므로, 기능적 사용자 요

구사항이 완전함을 보장하기 위해 매트릭 기반 평가 방법이나 전문가의 개인적 경험 등의 평가 방법과 함께 사용할 수 있다.

[11]에서는 요구사항의 완전성을 평가하기 위해 완전성을 형식의 완전성, 의미의 완전성, 참조의 완전성으로 구분하여, 각 완전성에 대한 하위 요소들을 정의하고 이를 평가하기 위한 10개의 매트릭을 정의하였다. 그림 3은 [11]에서 정의한 완전성 매트릭과 측정 방법을 나타낸다.

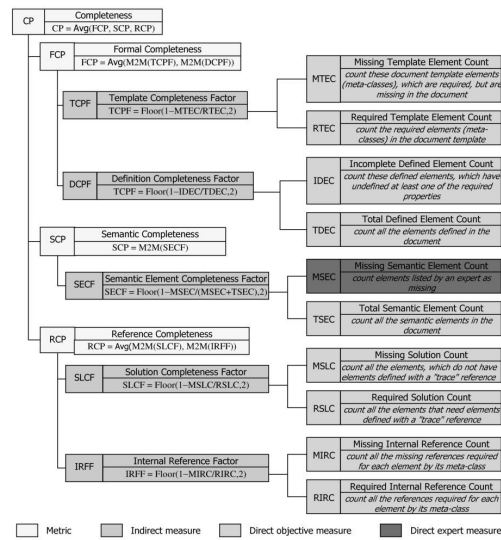


그림 3. 요구사항 완전성 평가 매트릭스 [11]
Fig. 3. Requirement completeness metrics and measures

이 외에도 요구사항을 해당 디자인, 구현 및 테스트 산출물에 매핑하여 모든 요구사항이 고려 되도록 하는 테이블인 요구사항 추적 가능성 매트릭스인 RTM(Requirements Traceable Matrix)이 사용되기도 한다. 이와 함께 각 요구사항의 중요도 및 긴급도를 기준으로 요구사항의 우선 순위를 측정하는 요구사항 우선 순위 지정 방식을 사용하기도 한다.

이러한 요구사항 완전성 평가 방법은 소프트웨

어 요구사항 명세가 정확하고 철저하며 포괄적이며 이해관계자의 목표, 요구 및 기대치를 정확하게 나타내도록 만드는 데 중요하다. 완전성 지표의 사용은 또한 소프트웨어 요구사항 명세의 품질을 개선하고 소프트웨어 개발 프로세스에서 결함 및 오류의 위험을 줄이는 데 도움이 될 수 있다.

4. 요구사항 완전성 평가를 활용한 완성도 감정

앞서 2장에서 기술한 감정 사례를 기반으로 완성도 감정에 요구사항의 완전성 평가를 적용할 수 있는 방법을 살펴보고자 한다.

완성도 감정의 핵심적인 사항은 발주자의 요구사항에 대한 완성 정도에 대한 평가이다. 일반적으로 제안서나 개발계약서, 요구사항 명세서, 데이터 및 기능설계서 등을 검토한 후, 전체 요구사항을 목록화하여 감정 항목을 도출한다. 도출된 항목에 대해 요구되는 기능이 적절히 구현된 완성도와 그렇지 못한 미완성으로 구별하여 평가하게 된다. 즉, 감정대상 항목에 대한 완성된 기능항목의 비율로 완성도를 산정하게 된다.

표 1의 (1) 설명이 있는 요구사항의 비율, (4) 입력 매개변수의 설명을 반영하는 요구사항의 비율 (5) 결과와 관련된 속성의 설명을 반영하는 요구사항의 비율 지표를 활용하여, 제출된 설계서가 요구사항 명세서로서 어느 정도 완전성을 갖추고 있는지를 평가해보았다.

첫 번째 평가는 설명이 있는 요구사항의 비율로, 설계서의 총 652개 기능항목에 대해, 스토리보드와 화면설계서 외에 제출된 문서자료(이메일, 회의록)를 통해 각 항목에 대해 해당항목이 수행해야 하는 사항에 대한 설명이 제공되고 있는지를 판단하였다. 표 2와 같이 설명이 있는 요구사항의 비율은 19.6%이다.

표 2. 설명이 있는 요구사항의 비율
Table 2. Percentage of Descriptive Requirements

대상항목의 개수	설명이 있는 항목의 개수	비율
652	128	19.6(%)

두 번째 평가 항목은 입력 매개변수의 설명을 반영하는 요구사항의 비율로써, 설계서의 총 652개 기능항목에 대해, 스토리보드와 화면설계서 외에 제출된 문서자료(이메일, 회의록)를 통해 각 항목에 대해 해당항목이 실행되기 위해 필요한 입력 값에 대한 설명이 제공되고 있는지를 판단하였다. 총 652개 가운데 입력값이 필요한 항목은 587개였고, 이 가운데 입력 매개변수에 대한 설명이 있는 항목은 252개였다. 표 3과 같이 입력 매개변수에 대한 설명이 있는 요구사항의 비율은 42.9%이다.

표 3. 입력 매개변수 설명이 있는 요구사항의 비율
Table 3. Percentage of requirements with descriptions of input parameters

대상항목의 개수	설명이 있는 항목의 개수	비율
587	252	42.9(%)

마지막 평가항목은 결과와 관련된 속성의 설명을 반영하는 요구사항의 비율로써, 설계서의 총 652개 기능항목에 대해, 각 항목의 실행결과에 대한 설명이 제공되고 있는지를 판단하였다. 표 4와 같이 결과와 관련된 속성의 설명을 반영하는 항목은 599개로 85.7%이다.

표 4. 실행결과를 설명하는 요구사항의 비율
Table 4. Percentage of requirements that explain the outcome of execution

대상항목의 개수	설명이 있는 항목의 개수	비율
652	599	85.7(%)

이상의 평가 결과를 통해, 요구사항의 기능적인 부분이나 입력 값에 대한 설명보다는 실행결과에 대한 설명이 요구사항 명세로서 역할을 대신하고 있다고 볼 수 있다. 이는 [3]에서 언급한 것처럼, 계약 체결 과정까지 명세 수준으로 구체화하지 못한 상태에서 프로젝트를 일단 시작하는 주먹구구식 개발 관행 때문이라고 볼 수 있다. 완전하지 못한 요구사항을 가지고 개발을 한다면, 개발이 완료된 시점에서는 개발자와 발주자 사이의 완성 수준에 대한 판단이 크게 달라질 수밖에 없을 것이다. 따라서 구체화 하지 못한 요구사항 명세가 소프트웨어 개발 프로젝트에서 결과물의 완성도에 미치는 영향이 매우 크다고 할 수 있다.

요구사항 명세는 감정 대상 시스템의 개발 범위를 결정하고, 완성 수준을 판단하는 중요한 요소이다. 구체화 하지 못한 요구사항 명세는 소프트웨어 개발 프로젝트의 실패 원인의 시작이라고 할 수 있다. 따라서 요구사항의 완전성 평가는 소프트웨어 완성도 감정에서 미완성에 이르게 된 주요 원인에 대한 정량적 지표로 활용할 수 있을 것으로 기대한다.

5. 결론

소프트웨어 완성도 감정의 핵심은 발주자의 요구사항에 대한 소프트웨어의 구현 여부 및 실제 작동 여부에 대한 평가이다. 그러나 구체화하지 않은 불명확한 요구사항으로 인해, 처음부터 문제점을 안은 채 개발이 시작되어, 결국 고객이 만족하지 못하는 개발 상태로 이어지는 경우가 대부분이다.

완성도 감정에서는 문서와 기록을 분석하고 검토하여 도출한 요구사항을 기준으로 완성 수준을 판단하게 된다. 요구사항 명세는 감정 대상 시스템의 개발 범위를 결정하고, 완성 수준을 판

단하는 중요한 요소이다. 구체화 하지 못한 요구사항 명세는 소프트웨어 개발 프로젝트 실패의 직접적인 원인이 된다고 할 수 있다. 따라서 요구사항의 완전성 평가는 소프트웨어 완성도 감정에서 미완성에 이르게 된 주요 원인에 대한 정량적 지표가 될 수 있다. 이런 관점에서 본 논문에서는 요구사항 완전성 평가 방법을 살펴보고, 몇 가지 완전성 평가 메트릭을 이용하여 감정 사례에 적용해 봄으로써, 요구사항 명세의 완전성이 갖는 의미를 고찰해 보았다.

참고 문헌

- [1] Ronal Carson, et. al. (2004, June). Requirements Completeness. In Proceedings of the Symposium of International Council on Systems Engineering (pp.930-944). Toulouse, France. DOI: <https://doi.org/10.1002/j.2334-5837.2004.tb00546.x>
- [2] Ian Sommerville (2015). Software Engineering (10th ed.). Person Inc. ISBN: 0133943038
- [3] Hoseok Yoo. (2018). The Importance of Software Requirement Specification and its Institutionalization in Korea. SPRI Insight Report (2018-001). <https://www.spri.kr/posts/view/22234?code=insight>
- [4] DoWan Kim. (2022). A Study on the Need and Methodology for Appraisal of Ready-made Software from the Software Engineering Perspective. Journal of Software Assessment and Valuation, 18(2), 19-25. DOI: <https://doi.org/10.29056/jsav.2022.12.03>
- [5] Korea Copyright Commission. (2010). Copyright-related Appraisal Case Collection (1). <https://copyright.or.kr>
- [6] Korea Copyright Commission. (2013). Copyright-related Appraisal Case Collection (2). <https://copyright.or.kr>

- [7] Korea Copyright Commission. (2016). Copyright-related Appraisal Case Collection (3). <https://copyright.or.kr>
- [8] G. Fanmuy, J. Marvin, and R. Houde. (2017). Needs and System Requirements Definition (version 2.0). INCOSE Deployment Package. <https://incose.org>
- [9] M. R. Raja Ramesh, Ch. Satyananda Reddy. (2021). Metrics for software requirements specification quality quantification. Journal of Computers and Electrical Engineering, 96, 107445. DOI : <https://doi.org/10.1016/j.compeleceng.2021.107445>
- [10] C. Dekkers. (2001). Applying Function Point Analysis to Requirements Completeness. ti Metricas Ltd. <http://www.metricas.com.br/>
- [11] J. Juchta. (Oct. 2016). Completeness and Consistency of the System Requirement Specification. In Proceedings of the International Conference on Computer Science and Information Systems. DOI: <https://doi.org/10.15439/2016F468>

저 자 소 개



김유경(Yukyong Kim)

2001.8 숙명여자대학교 컴퓨터과학과 박사
2005.9 - 2006.8 UC Davis, Post-doc.
2006.9 - 2013.9 한양대학교 컴퓨터공학과 연구
교수
2018.2 - 현재 숙명여자대학교 기초공학부 교수
<주관심분야> 웹서비스 QoS 평가, SOA 기반
IoT 신뢰 평가, 소프트웨어 품질평가