

논문 2019-2-5 <http://dx.doi.org/10.29056/jsav.2019.12.05>

SW감정에서 완성도와 기성고의 의미 및 산출 방법

윤영선*†

Meaning and Computation of Completeness and Payment in SW Appraisal

Young-Sun Yun*†

요 약

본 논문에서는 SW감정에서 많이 사용되는 완성도와 기성고의 용어의 정의를 재조명하고, 최근 SW분쟁에서 요구하는 완성도와 기성고의 의미와 산출 방법을 제시한다. 일반적으로 SW감정에서의 완성도는 최종 구축/구현된 SW의 기능상의 완성된 정도를 의미하고 있으며, 기성고는 개발비를 정산하기 위하여 현재까지 완료된 산출물 또는 기능을 기반으로 비용을 산출하는 것을 의미한다. 따라서 완성도는 최종적으로 개발 완료된 제품/산출물의 기능이나 인터페이스를 기준으로 판단하며, 기성고는 SW개발 단계별로 산출물 및 투입된 비용으로 산출된다. 최근에는 SW감정이 복잡해지고 완성도와 기성고에 대한 요구사항이 구체적으로 변화되고 있으므로, 본 논문에서는 각 용어의 의미와 목적을 다시 살펴보고 그 산출 방법을 제시한다.

Abstract

In this study, we reviewed the definition of completeness and payment in SW appraisal along to being complicated requirements of the appraisal, and we also presented their meaning and computation method. The completeness in SW appraisals means the degree of functional completion of the final product, and the payment refers cost spent to develop the product. Therefore, the completeness is evaluated based on the functions or interfaces of the final product, while the payment is calculated on the outputs or expenses of development steps. Recently, SW appraisal is complicated and the requirements for completeness and payment are changing. In this paper, we review the meanings and objectives of completeness and payment and present the evaluation methods for those.

한글키워드 : SW감정, 완성도, 기성고, SW 기성고 산출

keywords : SW Appraisal, Completeness, Payment, Calculation of SW payment

1. 서론

세계 정보기술(IT) 산업의 중심은 하드웨어

(HW)에서 소프트웨어(SW)로 빠르게 이동하고 있으며, 한국 또한 2006년 9%에서 2016년 15%로 SW 분야 비중은 증가하고 있다[1]. 이러한 SW시장은 2022년까지 연평균 4%씩 성장할 것으로 전망되며, 2017년의 경우 4조2850억원 규모로 집계되었다. SW시장의 성장 동력은 금융권

* 한남대학교 정보통신공학과 교수

† 교신저자: 윤영선(email: ysyun@hnu.kr)

접수일자: 2019.11.08. 심사완료: 2019.12.09.

게재확정: 2019.12.20.

대형 프로젝트와 제조부문의 스마트팩토리 증설 등이 국내 SW수요 증가에 긍정적인 영향을 주는 것으로 파악되었다[2]. SW시장의 증가와 더불어 SW산업의 경쟁력을 강화하기 위해서는 안전한 SW생태계 구축과 이를 뒷받침하는 SW지식재산권의 보호가 무엇보다도 중요하다. 그렇지만 현재까지는 관련 법제도 및 국가적 정책이 SW산업에 효과적 영향을 주지 못하여 SW지식재산권을 중심으로 다양한 개선방안과 그 절차 등이 제시되고 있다[3].

최근 SW감정 요청사항 중에 완성도와 기성고 산출을 요구하는 경우가 많아지고 있어, 본 논문에서는 완성도와 기성고의 용어와 그 의미를 고찰하고, 널리 사용되는 SW개발 방법론에 따른 산출방법 등을 제안한다. 본 논문에서 제안하는 방식은 기존의 산출방식과 일부 상이하거나 다를 수 있기 때문에 토론과 논의과정이 필요할 수 있으며, 이는 SW감정 과정에서 면밀히 검토할 사항으로 의미가 있을 수 있다고 생각한다.

2. SW 감정 제도 개요

우리나라는 SW관련 분쟁이 발생하였을 때 저작권법 및 소송법에 근거를 두고 공적 SW감정 제도를 이용하여 해결하고자 한다. SW 감정은 SW지적재산권 관련 분쟁이 발생하여 이를 담당하는 법원이나 수사기관 등이 의뢰하는 경우 관련전문가의 경험과 지식을 활용하여 분쟁대상 SW에 대한 동일·유사성, 완성도, 개발하자 등을 판단하여 그 결과를 제시하는 증거조사 방법을 말한다[4]. 이때 SW에 대한 기술적 분석 및 판단을 전제로 하는 사건들의 경우 SW감정결과는 사건 해결의 결정적 판단근거로 기능하고 있으며, 최근 2015년부터 2017년까지 일반저작물 12건과 컴퓨터프로그램저작물 15건의 판례를 살펴

볼 때 88.9%에 해당하는 24건이 감정결과가 판례에 반영되었다고 확인되었으며, 3건의 경우는 반영여부를 확인할 수 없었고, 감정결과에 반하는 판례는 찾아볼 수 없었다. 이로부터 SW감정 결과는 SW관련 분쟁의 중요한 판단 근거가 될 수 있다[5].

SW감정은 비교대상 SW, 두 프로그램간의 상호 비교 및 분석을 통하여 양 SW가 어느 정도의 유사성을 가지는가를 판단하는 유사도 감정, 위탁 개발된 SW에서 발생하는 기능상의 문제점, 단위작업 프로세스 결과에 대한 신뢰상의 문제점, 운영상의 문제점 등을 관련 자료의 분석과 실제 시스템의 구동을 통하여 완성 및 하자의 정도를 판단하는 완성도(하자) 감정, SW개발에 있어서 그 공정상의 개발비용 등을 소프트웨어 공학적인 측면에서 판단하여 산정하는 개발비용 산정 감정으로 구성된다.

SW 감정 중 유사도감정은 저작권법 제119조의 규정에 근거를 두고 있으며, 이는 “저작권의 침해 등에 관한 감정”으로 간주하고 있기 때문이다. 그러나 완성도 감정이나 개발비 산정 등의 감정은 저작권을 대상으로 하지 않지만 한국저작권위원회 정관과 감정업무 규정에 따라 저작권의 침해 등에 관한 감정 업무를 효율적으로 수행하기 위하여 (제1조), 동일·유사성, 완성도(하자), 가치 및 기타 감정 (제18조)로 명시하고 있어 SW감정의 수행근거를 명시하고 있다[6].

3. 완성도 개요 및 하자 재정의

완성도의 공식적인 정의는 존재하지 않지만, 사전적 의미로 완성도를 “어떤 일이나 예술 작품 따위가 질적으로 완성된 정도”, “목표로 이른 정도”로 정의하고 있다. 이를 기반으로 프로그램의 완성도는 “프로그램이 질적으로 완성된 정도” 후

은 “프로그램이 사용자의 요구를 만족한 정도”로 정의할 수 있다. 그러나 위 두 정의 모두 주관적 개념이기 때문에 최종 제품이 인도될 때에 분쟁이 발생하기도 한다. 따라서 각 프로그램 단계별로 산출물에 의한 정량적 판단 및 요구사항에 대한 정성적 평가가 수반되어야 한다. 요구사항이 모호하게 기술되어 있는 경우, 최종 납품된 제품에 대한 발주사와 개발사 사이에는 기능적 요구사항에 대한 이견이 존재할 수 있으며, 유지보수 기간동안 양사의 협의를 거쳐 수정할 수 있다. 이와 반대로 비기능적 요구사항은 시스템의 특성과 제약조건을 정의하고, 품질 또는 개발 프로세스에 대한 요구사항을 포함할 수 있다. 특히 비기능적 요구사항은 기능적 요구사항보다 더욱 중요할 수 있고, 요구사항 정의 및 명세서에 기준 척도가 제시되지 않는 경우가 많아 분쟁 가능성이 높기 때문에, SW 개발 프로세스 및 감정 단계에서 완성도에 대한 정확한 판단 기준이 필요하다[7].

SW 완성도 감정의 경우, SW에서 발생하는 기능적 문제점, 신뢰성, 운용상의 문제점 등을 대상 SW, 제안서, 개발계약서, 요구사항 정의서 및 명세서, 시스템 설계도, 개발 작업 일지 등의 자료 분석과 실제 프로그램의 동작으로 완성 및 하자의 정도를 판단하고 있다. 일반적으로 하자란 “최종 제품 인도 후 정상적으로 동작하지 않아, 사용하기 어렵거나 사용자가 불편을 느끼는 기능적 또는 비기능적 불능상태”를 말하며, 하자의 유형을 미구현, 부분구현, 미동작, 오동작 등으로 구별한다. 이외에도 미완성, 부분동작 등의 용어로도 사용되나 크게 구현 및 작동으로 구분할 수 있다. 미구현, 부분구현(미완성)은 설계나 구현단계에서 누락되어 개발 자체가 이뤄지지 않은 것을 말하며, 미동작, 오동작은 구현은 되었으나 그 기능의 완성도 또는 신뢰도가 낮은 것을 의미한

다. 구현과 관련된 하자는 설계의 유무에 따라 하자의 정도가 달라져야하며, 동작과 관련된 하자는 설계는 진행되었으나 구현단계에서 하자가 발생한 것으로 판단하여야 한다. 이를 근거로 표 1과 같이 하자의 기준을 제정의 한다.

표 1. 완성도 감정에서 하자 정의
Table 1. Definition of deficiency in appraisal of SW completeness

분류	세부내용	설계	구현	확인
구현	미구현	확인	미흡	- 설계문서 - 소스코드 - 동작확인
	부분구현			
동작 (기능)	미동작	완료	미흡	- 소스코드 - 동작확인
	부분동작			
	오동작			

표 1의 하자 정의에서 구현과 동작의 판단은 요구사항정의서의 항목과 일치하나, 설계 및 구현 작업이 완료되었는지에 따라 다르게 정의할 수 있다. 미구현 및 부분구현의 경우, 설계와 구현이 완료되지 않은 상태를 의미하며, 미동작과 부분동작, 오동작은 설계 및 구현이 완료되었으나 요구사항정의서의 항목을 충족시키지 못한 경우이다. 일반적으로 완성도 감정시 감정결과를 미구현, 부분구현(완성), 완료(완성)으로 표현하는 경우에는 미구현과 부분구현(완성)을 하자로 판단하여 하자보수 비용을 산정하는 경우, 정확한 비용이 산정되지 않을 수 있다. 이에 본 논문에서 설계 및 구현단계를 고려한 하자를 재정의하고, 하자보수 개발비용 산정 등에서 정확한 비용을 산정할 수 있도록 한다. 따라서 설계관련 자료 및 소스코드가 확보되는 경우에는 구현단계의 하자를 세분화하고, 그렇지 않은 경우에는 동작 분류에 의한 감정을 진행하여야 할 것이다.

4. 완성도와 기성고 의미

일반적으로 완성도의 산출방식은 감정요청사항에 따라 결정되며, 다음과 같이 구분 가능한 기능별로 산출한다.

$$\text{완성도} = \frac{\text{정상작동 총 기능 수}}{\text{감정대상 총 기능 수}} \times 100$$

최근에 개발된 SW는 GUI(Graphical User Interface)에 기반하여 구현되기 때문에, 사용자는 쉽게 메뉴나 버튼의 실행에 의한 기능으로 구분할 수 있다. 그러나 일부 기능의 경우 단일 기능이 복합적으로 결합되어 혼재하는 경우가 많아 SW 완성도 분쟁시 분쟁 당사자간의 조율에 의하여 기능구분을 진행하여야 한다. 그렇지 않은 경우, 감정인의 전문적 지식에 의하여 각 기능의 중요도나 복잡도를 반영한 가중치를 이용하거나, 모듈의 비중에 따라 모듈별 가중치를 부여하기도 한다.

또한 완성도 산출시 하자의 부분구현이나 부분동작, 오동작 등의 반영여부 또는 사전에 조율되어야 한다. 완성도를 산출시 “최종 결과물이 요구사항 항목을 충족시키는가?”와 같이 문구에 충실하다면 하자로 판단된 항목은 완성도 산출에 반영되지 않기 때문이다. 그러나 최근에는 완성도 감정이 요청되는 SW분쟁의 경우 전체 시스템이 완성되지 않은 경우가 많아 기성고를 고려하여 하자 항목 중 부분구현/부분동작/오동작을 완성도에 반영하기도 한다.

기성고란 약정된 총공사비 중에서 공사한 부분만큼의 공사비[8]를 말하며, 기성율은 총 공사비용에서 기성고가 차지하는 비율로 정의하고 있으나, 실무적으로는 기성고, 기성율, 기성고비율 등이 혼재되고 있다. SW개발 및 공급 계약은 일종의 도급계약으로서 원칙적으로 프로그램 개발을 완성하여야 보수를 청구할 수 있으며, 민법 제668조에 의하여 SW프로그램의 하자로 인하여

도급인이 계약의 목적을 달성할 수 없는 경우에는 계약을 해제할 수 있다고 규정하고 있다. 또한 민법 제665조에서는 보수지급의무는 일의 완성에 대한 대가이며, 통상 도급인은 그 목적물에 하자가 존재하더라도 일이 완성된 경우에는 대금지급 의무를 질 수 있다고 하고 있다[9]. 이에 따라 SW 완성도 감정은 완성도뿐만 아니라 일의 진척 정도에 따른 보수지급시의 범위에서 판단할 수 있도록 기성고와 기성율에 대한 결과 또한 제시함이 바람직하다. 이런 이유로 최근의 감정요청사항에서는 완성도와 기성고를 동시에 요구하는 경향이 강하다.

5. 기성고 산출

발주사가 지급하여야 할 SW개발대금은 약정된 계약금액을 기준으로 기성율을 곱하는 방식으로 산정하며, 기성율은 요구사항 정의서에서 개발이 완성된 항목과 아직 완성되지 않은 항목을 확정된 후, 완성된 부분의 개발비와 미완성된 부분의 소요 공사비(하자개발비)에서 완성된 부분의 개발에 소요된 비용이 차지하는 비율로 산출한다¹⁾. 위 내용을 기준으로 한다면 기성율은 완성된 개발비와 미완성된 개발비(하자개발비)를 산출하고, 완성된 항목의 개발에 소요된 금액의 비율을 먼저 계산한 후, 기성율을 반영한 개발대금을 지급하여야 한다. 최근의 SW개발은 오픈소스 및 여러 개발도구를 사용하기 때문에, 정확한 개발비용의 산정이 어렵고 계약금액을 총 개발비용으로 판단하기가 어렵다. 따라서 기성율과 약정 계약금액에 의한 지급대금을 산출하는 것이 합리적일 수 있다고 판단한다.

기성고 산출시 소프트웨어 개발 방법론에 따

1) 대법원 1996.1.23., 선고94다31631, 31648판례를 SW개발 측면에서 재해석하여 작성하였음

른 개발단계를 반영하는데, 일반적으로 폭포수 모델(waterfall model)을 많이 사용한다. 폭포수 모델은 오랜 기간 사용되어 왔으며, 적용사례가 많고 단계별로 정형화된 접근 방식을 사용하기 때문에 기술적인 위험요소가 적다는 장점이 있다. 하지만, 다음 단계에 대한 이해나 경험 없이 각 단계를 마무리하고 다음 단계를 진행하기 어렵다는 한계를 가지고 있다. 이의 대안으로 전체적인 계획안에서 문서 등의 산출자료를 통해 진행과정을 파악하지 않고 일정한 주기를 가지고 반복적으로 프로토타입을 만들어 요구사항을 추가하고 수정하는 방식의 애자일 방식이 제안되었으며, 대규모 소프트웨어 개발 방법론으로 제시되고 있다. 하지만, 투입공수 측정 등과 같은 기존의 평가방식을 사용하지 않기 때문에 구체적인 개발단계의 품질 평가 및 관리가 미흡하다는 약점이 지적된다. 본 논문에서 논의하는 기성고 산출은 소프트웨어 개발을 각 단계별로 구분하여 소요된 비용을 산출하기 때문에 폭포수 모델에 한정하여 산출 방법을 검토한다. 폭포수 모델은 소프트웨어 개발 단계를 순차적으로 모델링한 방법으로 요구사항 분석, 설계, 구현, 시험, 통합, 유지 등으로 구분하고 있다. 이를 바탕으로 한국소프트웨어산업협회에서는 국가기관 등에서 소프트웨어 사업을 추진할 때 SW개발비 등에 대한 예산수립, 사업발주, 계약시 적정대가를 산정하는 경우 기준이 될 수 있도록 “SW사업 대가산정 가이드”를 공표하고 있다. 이 가이드는 행정자치부의 “행정기관 및 공공기관 정보시스템 구축·운영 지침”과 기획재정부의 “예산안 편성 및 기금운영 계획안 작성 세부지침” 등에서 인용하고 있다 [10]. SW사업 대가산정 가이드에서는 SW개발 단계별 비중을 표 2와 같이 정의하고 있으며, 이를 반영하여 기성율을 산출할 수 있다.

표 2와 같이 개발 단계별 비중을 적용하여 기성율 산출시 각 단계에 적용되는 단계별 완성도

에 대한 기준이 명확하지 않아 감정인마다 혼란스러운 경우가 존재한다. 또한 모든 SW 개발사가 SW사업 대가산정 가이드에 따라 개발을 진행하는 것이 아니기 때문에, 적절한 SW개발 단계별 비중을 부여하기도 어렵다. 따라서 본 장에서는 SW 완성도 감정시 단계별 비중 계산 방법과 단계별 완성도 계산시 필요한 산출물 및 항목을 제시하고, 기성고 산출 방법을 제안한다.

표 2. SW사업 대가산정 가이드 기준 개발 단계별 비중

Table 2. Weights of SW development cycle based on SW Development Cost Estimation Guide

단계	분석	설계	구현	시험
비중	0.19	0.24	0.32	0.25

일반적으로 SW개발 계약과정에서는 계약 후 개발일정과 산출물 목록을 확정하고, 산출물의 생산여부에 따라 단계별 완성여부를 판단한다. 따라서 본 논문에서는 계약서 또는 과업명세서 기준으로 개발일정에 따른 Week/Month를 반영하여 단계를 산출하고, SW사업 대가산정 가이드와 극단적으로 차이가 나지 않는 경우 계약서/과업명세서의 개발 비율로서 단계별 비중을 정의한다. 표 3은 SW 발주사와 개발사가 합의한 개발 일정에 의한 단계별 비중의 예를 제시한다.

표 3. 계약 당사자 합의에 의한 SW 개발 단계 및 비중 예시

Table 3. Example of weights of SW development cycle by both agreement

단계	분석	설계	구현	시험
계약 용어	현황 분석	시스템 설계	시스템 개발	TEST
일정 (주)	3	3	6	3
비중	0.20	0.20	0.40	0.20

SW개발 단계별 비중을 산출한 후, 단계별 완성도를 계산하기 위해서는 단계별 산출물 목록 및 확인이 필요하다. 분석, 설계 단계는 산출물에 의하여 단계의 완성도를 판단할 수 있으나, 구현 단계의 경우에는 산출물 이외에도 기능의 구현 여부, 소스코드 등이 같이 고려되어야 한다. 또한 마찬가지로 시험단계 또한 산출물 이외에도 기능

의 동작 확인이 필요하다. 일반적으로 구현 단계는 단위테스트가 포함되어 있으며, 시험단계는 통합테스트 등이 실시되기 때문에 구현과 시험 단계를 명확하게 구분하기는 쉽지 않다. 따라서 본 논문에서는 표 1에서 구분한 구현과 동작으로 분류한 하자발생 여부를 구현 및 시험단계에 각각 적용하는 것을 제안한다.

표 4. SW개발 단계별 활동, 작업, 산출물 정의
Table 4. Activities, tasks, and outputs of SW development steps

단계	활동	작업	산출물	하자확인
분석	요구분석	요구사항 수집	- 인터뷰계획서 - 인터뷰결과서	
		요구사항 정의	- 요구사항 정의서 - 요구사항 명세서	
		요구사항 추적	- 요구사항 추적표	
	업무분석	업무분석	- 현행업무절차 분석서 - 개선업무절차 정의서	
		데이터 분석	- 개선대상 업무 및 범위 정의서	
	분석단계 시험 계획	총괄테스트 계획	- 총괄테스트 계획서	
설계	구조설계	시스템 설계	- 시스템구조설계	미구현, 부분구현 확인
		SW 설계	- SW구조설계	
	응용설계	클래스 설계	- 클래스 설계서	
		기능 설계	- 기능설계서	
		화면 설계	- 화면설계서	
		사용자 인터페이스 설계	- 사용자인터페이스 설계서	
	DB설계	개념DB 모델 설계	- 개념 데이터 ERD	
		논리DB 설계	- 논리데이터요소정의서 - 물리데이터요소정의서 - 논리/물리 ERD	
		물리DB 설계	- DB 설계서	
		데이터흐름도 설계	- 데이터흐름도	
	설계단계 시험 계획	통합테스트 설계	- 통합테스트 시나리오	
사용자테스트 설계		- 사용자테스트 시나리오		
구현	개발	프로그램 개발	- 소스코드	미구현, 부분구현, 미동작, 부분동작 오동작 확인
	구현단계 시험	단위테스트	- 단위테스트 계획서(정의서) - 단위테스트 결과서	
시험	시험	통합테스트	- 통합테스트 결과서	오동작 확인
		사용자테스트	- 사용자테스트 결과서	

만약 발주사와 개발사 간에 과업명세서나 추진계획에서 산출물들에 대한 협의가 이뤄지지 않는 경우 일반적인 SW개발 방법론에서 제시하는 산출물들을 이용하여 단계별 완성도를 판단하여야 한다. 표 4는 특허청에서 2014년 발표한 소프트웨어 개발방법론에서 제시한 표준프로세스 중에서 SW사업 대가산정 가이드와 대응되는 개발 단계별 산출물을 참조하여 작성하였다. 특허청에서는 개발방법론 표준프로세스를 총 7단계, 23개 활동, 52개 작업과 74개 산출물로 제시하였으나 [11], 본 논문에서는 4단계 10개 활동, 22개 작업, 28 산출물로 축소하였으며, 최소한의 활동과 작업, 산출물로 정의하였다.

표 4의 설계단계와 구현단계, 구현단계와 시험단계는 SW개발단계에서 명확하게 구분하지 않고, 진행하기도 한다. 따라서 표 4의 산출물 및 확인절차는 발주사와 개발사의 협의가 없는 경우, SW 감정과정에서 단계별 완성도 산출시 참고할 수 있도록 제시한 것이다.

기성고 산출과정에서 분석 및 설계단계는 산출물의 결과에 따라 단계별 완성도를 책정하는 것이 일반적이나, 표 1과 같이 하자를 구현과 동작으로 세분화하는 경우 구현 단계의 완성도는 설계결과에 영향을 받게 된다. 구현단계의 완성도는 하자기능을 반영한 완성도를 적용하여 산출하는 것이 일반적이다. 문헌적 완성도는 전체 시스템의 완성도를 의미하지만, 기성고 산출에 사용되는 완성도는 하자라 할지라도 그 개발비용이 산정될 수 있기 때문이다.

미구현, 부분구현의 경우 설계가 진행되었는데도 불구하고 구현단계에서 오류가 발생하였을 수 있으며, 설계자체가 진행되지 않을 수 있다. 따라서 미구현, 부분구현으로 판단된 기능의 경우에는 소스코드와 설계문서를 판단하여 설계문서의 질적 평가까지도 반영되어야 한다. 반면, 미동작, 오동작, 부분동작의 경우 설계는 완료되었다고

판단하며 구현과정의 완성도를 저하시키는 요인이 된다. 또한, 동작기능의 오류인 경우에는 시험단계에서 수정 및 개선될 수 있기 때문에 시험단계의 산출문서가 존재하는 경우나 시험단계에 진입하지 못하는 SW개발 분쟁의 경우에는 별도의 고려가 필요하다. 이 경우, 부분구현과 부분동작을 포함하는 기능적 완성도는 구현단계의 완성도에 반영하고, 오동작 하자는 시험단계에 적용하여 기성고를 산출할 수도 있다. 이를 요약하면 표 5와 같이 정리할 수 있다.

표 5. 기성고 산출 방법
Table 5. Computation method of payment

단계	완성도 산출	하자 반영	
분석	산출물		
설계	산출물, 기능동작	미구현, 부분구현	
구현	산출물, 기능동작		미동작, 부분동작,
시험	산출물, 기능동작		오동작

6. 결론

본 논문에서는 SW분쟁에서 발주사와 개발사 간에 완성된 제품의 완성도/하자 감정 또는 개발 기반의 중단에 따른 개발비 평가에서 많이 요청되는 완성도와 기성고의 개념 및 의미를 살펴보았다. 완성도의 경우 감정인마다 하자의 정의는 유사하나 정확히 일치하지 않고 그 단계 또한 서로 상이한 면이 있다. 이에 본 논문에서는 완성도를 기성고로 접근하는 경우나 하자보수 개발비 산정시 고려할 하자에 대한 정의를 재고하였다. 또한 최근의 완성도 감정의 경우 개발 중단된 경우의 분쟁도 포함하고 있어, 기성고 및 기성을 산출에 기반한 완성도 산출을 요구하고 있다. 기

성고 및 기성을 산출시 SW개발의 일반적인 방법론을 적용하여 각각의 산출물 및 기능의 완성도를 판단하여 적용하나, 감정인마다 각자 기준이 달라 감정 진행시 많은 어려움이 따른다. 따라서 본 논문에서는 기존의 SW개발 방법론에서 제시하는 표준 절차를 적용하되, 필수적인 산출물과 재정의한 하자에 따라 기성고 산출시 고려할 사항을 제안하였다.

제시된 방안은 모든 SW완성도 분쟁에 일괄적으로 적용할 수는 없지만, 분쟁형태에 따라서 감정인이 적절하게 반영한다면 보다 정확하고 객관적인 감정을 진행할 수 있을 것으로 기대한다.

이 논문은 2019년도 한남대학교 학술연구조성비 지원에 의하여 연구되었음.

참 고 문 헌

[1] 김재희, “SW로 뛰는 해외, HW 못 벗어난 한국”, 동아일보 2017.3.14. <http://www.donga.com/ISSUE/Vote2016/News?m=view&date=20170314&gid=83309067>

[2] 이희욱, “국내사용SW시장, 연평균4%씩 성장”, 블로터넷, 2018. 7. 31. <http://www.bloter.net/archives/316062>

[3] 홍승완, 박상호, 전민서, 김양훈, 장항배, “SW완성도 감정과 개발비산정 분석연구”, 2017 경영정보관련 춘계통합학술대회, pp.426-432, 2017. <https://www.earticle.net/Article/A303486>

[4] SW저작권 등 분쟁사건 감정사례집, 한국저작권위원회, 2009.

[5] 감정사건판례집(4), 한국저작권위원회, 2018.

[6] 김시열, 최재식, “SW감정 체계에 대한 거버넌스 관점에서의 고찰”, 한국소프트웨어감정평가학회 논문지 제13권 제1호, pp.1-14, 2017년 6월. <http://www.i3.or.kr/html/paper/2017-1/2017-1.pdf>

[7] 컴퓨터프로그램 감정매뉴얼(I), 한국저작권위원회, 2015년 11월.

[8] 네이버국어사전, <https://ko.dict.naver.com/#/search?range=all&query=기성고>

[9] 정희목, “소프트웨어 개발공급 계약에서 하자과 보수지급 및 해제와의 관계”, 가산종합법률사무소, 2017년10월 <https://tradesecret.tistory.com/382>

[10] 한국소프트웨어산업협회, “[사업대가](공공부문) SW사업대가산정가이드”, <https://www.sw.or.kr/site/sw/ex/board/List.do?cbIdx=276>

[11] 특허청, “소프트웨어(SW) 개발방법론”, 2014년 12월. <http://policy.nl.go.kr/cmmn/FileDown.do?atchFileId=128340&fileSn=10966>

저 자 소 개



윤영선(Young-Sun Yun)

1990.2 KAIST 전산학과 졸업
 1992.2 KAIST 전산학과 석사
 2001.2 KAIST 전산학과 박사
 2006.4-2007.2 한국전자통신연구원 초빙연구원
 2012.8-2013.7 University of Washington 방문학자
 2001.3-현재 : 한남대학교 교수

<주관심분야> 음성인식, 음성변환, 화자인식, 인공지능, 내장형시스템, 저작권침해, 유사도, 완성도 감정 등