

논문 2021-2-2 <http://dx.doi.org/10.29056/jsav.2021.12.02>

# 소프트웨어 완성도 감정과 기성고 감정 분리 필요성에 대한 고찰

김도완\*†

## A Study on the Need for Separation of Software Completeness Appraisal and Software Ready-made Appraisal

DoWan Kim\*†

### 요 약

본 연구에서는 감정사례 및 판례를 분석하여, 기존 소프트웨어 완성도 감정으로 분류되어 수행된 완성도 감정, 기성고 감정, 하자 감정 및 비용 감정의 문제점을 적시하고, 그 해결 방안을 제시한다. 판례와 법률적 관점에서 완성도와 기성고율은 큰 차이를 가지고 있다. 완성도는 개발프로세스가 종료된 소프트웨어를 대상으로 전제하는 반면, 기성고율 감정은 미완성된 소프트웨어의 개발진척도를 평가하기 때문이다. 종종 소프트웨어 기성고와 관련된 판례에서는 소프트웨어 공학 개발 절차에 따른 단계별 가중치를 인정하여 전체 기성고 또는 완성도를 산정하는 것을 볼 수 있는데, 감정에서는 대부분 기능의 구현-작동여부만을 완성도 비율 산정의 척도로 삼고 있는 문제도 존재한다. 또한 기존 소프트웨어 완성도 감정사례에서 다루지 않았던 문제 중 하나는 소프트웨어 하자에 대한 책임 소재 분석 및 감정이 언급되지 않고 있는데 반하여, 판례에서는 분쟁이 발생한 원인을 찾아 책임 소재를 다루고 있다. 본 논문에서는 위 제기된 문제를 체계적으로 분류하여 소프트웨어 완성도 감정과 소프트웨어 기성고 감정을 분리할 것을 제안하고 감정 방안을 제시한다.

### Abstract

In this study, problems of software completeness appraisal are pointed out and their solutions are presented by analyzing appraisal cases and judicial precedents. Completeness appraisal, ready-made appraisal, defect appraisal, and cost appraisal have been classified as and have been evaluated with extant software completeness appraisals. From a legal point of view, and in judicial precedents, however, there is a big difference between the definition of completeness and the completion rate. This is because the degree of completeness is evaluated under the premise that the software's development is complete, whereas the ready-made appraisal inspects the development progress of unfinished software. Often, in cases involving software completion rate, the total completion level is calculated by weighting each step of the software development process. However, completeness evaluations use the software's realization-operation as its sole criterion. In addition, another issue not addressed in existing software completeness appraisal cases is that there is no mention of who is responsible for software defects, whereas in case law, the responsible party is determined by finding who caused the dispute. In this paper, we systematically classify these problems, and present a novel evaluation method that separates software completeness evaluations from software completion evaluations.

**한글키워드** : 소프트웨어 완성도 감정, SW기성고율 감정, SW하자, SW저작권 판례, SW공학 절차  
**keywords** : SW completeness evaluation, SW ready-made rate appraisal, SW defect appraisal, SW copyright precedent, SW Engineering process

\* 배재대학교 인터넷소프트웨어공학과

접수일자: 2021.11.30. 심사완료: 2021.12.11.

† 교신저자: 김도완(email: dwkim@pcu.ac.kr)

게재확정: 2021.12.20.

## 1. 서론

소프트웨어 도급/수급 관계에서 도급자와 수급자 사이에 발생하는 분쟁의 대부분은 완성도 감정으로 이어진다[1][3]. 현행 완성도 감정은 계약 관계 프로세스 종료 후 발생하는 하자에 대한 분쟁 뿐만이 아니라, 프로세스 진행 중에 발생하여 중단된 분쟁 또한 대부분 완성도 감정 범주에 포함되고 있다. 이는 완성도 감정과 기성고 감정에 대한 법적 차이를 명확하게 구분하지 않고 있는 점에 연유한다. 또한 소프트웨어 도급/수급 계약으로 진행되는 프로젝트는 상호간의 협업이 매우 중요하며[2], 따라서 분쟁 발생 시 분쟁의 책임 소재를 규명하는 것 또한 감정의 중요한 부분으로 여겨지나, 현행 감정에서 이에 대하여 대부분 다루어지고 있지 않다. 분쟁의 책임소재를 규명하는 것이 중요한 이유는 특히 기성고 감정에서 개발 프로세스 단계 별 사업 참여자의 역할이 다르며, 전체 기성고에 대한 단계별 가중치가 있기 때문이다. 판례에서도 하자발생의 원인 및 개발 프로젝트에서 사업 참여자의 역할 수행 여부를 다루고 있는 점을 다수 찾아 볼 수 있다[4][5][6].

본 논문은 기존 완성도 감정에서 혼재되어 사용되는 하자, 버그, 완성도, 기성고 등 개념을 명확하게 정의하고, 법률적 정의에 맞는 소프트웨어 완성도 감정 체계를 제시한다.

## 2. 관련 연구

### 2.1. 소프트웨어 완성도 관련 분쟁 유형

한국저작권위원회에 의뢰된 소프트웨어 완성도 감정 관련 분쟁 유형은 저작권 침해 판단을 목적으로 하는 것 보다는, 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 있다[1]. Dr. John McManus & Dr. Trevor Wood-Harper의 연구 결과에 따르면 EU 국가들에서 수행된 소프트웨어 개발 프로젝트 214개 중 23.8%에 해당하는 프로젝트는 미완성인 상태로 중단되었으며, 완료된 76.2% 163개 프로젝트 중, 69개 프로젝트는 계약된 일정을 넘겨 납품되거나 요구된 완성도를 충족시키지 못한 채 추가적 작업 비용이 요구된 것으로 조사되고 있다[7]. 더 나아가 개발되어 납품된 소프트웨어의 많은 수가 이용자의 요구를 만족시키지 못하여 사용되지 않고 사장되어버린다고 한다[8]. 이는 소프트웨어 개발 도급/수급 관계에서 다양한 유형의 완성도 관련 분쟁이 발생할 수 있는 것을 의미한다.

### 2.2. 소프트웨어 완성도 관련 법률적 문제

소프트웨어 개발 도급/수급 계약은 다양하고 복잡한 법률적 문제를 내포하고 있다[3]. 소프트웨어는 무형의 지적 창작으로 시작하여 유형의 제작품 형태로 그 실체를 내 보인다. 소프트웨어 개발 도급/수급 계약은 민법상 계약상의 법리가 작용하는 반면, 계약 대상물인 소프트웨어는 지적창작물로서 저작권법에 따라 지적재산권의 법리를 따른다.[3]

또 다른 측면에서 소프트웨어 도급/수급 개발 프로젝트는 도급자와 수급자의 협력이 필수적이라 여겨진다. 도급자는 업무 프로세스 및 비즈니스 로직에 대한 전문가이며 개발될 소프트웨어의 잠재적 이용자 이지만, 수급자는 도급자의 요구를 충족시키는 소프트웨어를 개발할 의무를 가지고 있는 기술 전문가 일 뿐이기 때문이다[2][3]. 따라서 도급자의 협력 없이 수급자가 도급자의 요구를 만족시키는 소프트웨어 개발 한다는 것은 매우 어려운 일이다[2][8]. 한국저작권위원회에

의뢰되는 소프트웨어 완성도 감정 분쟁의 대부분은, 소프트웨어 프로젝트가 가지는 이러한 특성을 이해하지 못한 상태에서 진행되는, 각각의 역할 분담이 모호한 부실한 용역계약에 기인하고 있다고 할 수 있다.

### 2.3 소프트웨어 완성도 감정과 판례 분석

한국저작권위원회 저작권관련 감정사건 판례집[4][5][6]에 수록된 완성(하자)도 감정 29개 판례 사례를 분석하여 보면, 첫째 소프트웨어 개발이 미완성 상태에서 분쟁이 발생하였을 경우 대부분 도급자가 승소하였으며(도급자 용역대금 지급 시), 완성은 하여 납품되었으나 기능상의 하자가 발생하여 분쟁이 일어난 경우 하자의 경중에 따라 도급자와 수급자의 승소 비율은 비슷한 것으로 나타났다. 이때 특이한 점은 계약서 상에 나타난 원고의 역할(예; 자료제공 및 협조 미흡)이 결여되어 하자가 발생한 것으로 판단되는 경우 법원은 수급자의 손을 들어주고 있는 점이다.

## 3. 소프트웨어 완성도, 하자, 기성고에 대한 정의 정리

현재 소프트웨어 완성도, 또는 하자 및 기성고에 대한 정의는 혼동되어 쓰이고 있는 실정이다. 각각의 정의에 대하여 규정집 또는 법률적 정의를 살펴보면 다음과 같다.

### 3.1. 소프트웨어 완성도

한국저작권위원회 컴퓨터프로그램 완성도 감정 매뉴얼은 완성도를 다음과 같이 정의하고 있다[12]:

“프로그램의 완성도는 프로그램이 질적으로 완성된 정도, 혹은 프로그램이 사용자의 요구를 만족한 정도”

대법원 판례에 따르면 SW 도급관계에서 제작물의 공급 업무가 완성되었음을 판단하는 기준은 다음과 같다[13];

“제작물 공급계약에서 일이 완성되었다고 하려면 당초 예정된 최후의 공정까지 일단 종료하였다는 점만으로는 부족하고 목적물의 주요구조 부분이 약정된 대로 시공되어 사회통념상 일반적으로 요구되는 성능을 갖추고 있어야 함으로 ...”

위 정의 및 판례에 따르면 SW 완성도라고 함은 당초 예정된 최후의 공정까지 일단 종료되었음을 전제로, 프로그램이 도급자(사용자)의 요구를 ‘질적으로 만족시키는 정도’라 정의할 수 있다. ‘질적으로 만족시키는 정도’란 다시금 도급계약관련 문서에 포함될 수 있는 ‘표현될 수 있는 기능적 요구사항’을 만족시키는 정도와, 도급계약관련 문서에 포함되기 어려운 ‘표현되기 어려운 사용상의 요구사항’을 만족시키는 정도로 나누어 볼 수 있다. ‘표현되기 어려운 사용상의 요구사항’은 “사회통념상 일반적으로 요구되는 성능”을 의미한다.

### 3.2 소프트웨어 하자

‘하자’의 사전적 의미는 “옥의 얼룩진 흔적이라는 뜻으로, ‘흠’을 이르는 말”이며, 법률적으로는 “당사자가 예기한 상태나 성질이 결여되어 있는 일”을 지칭한다. 민법주해XV에는 하자를 “일이 보증된 성질을 갖지 않는 경우 및 그 경제적 가치 혹은 통상 또는 계약상 예정된 일정한 효용을 멸실 또는 감소시키는 결점이 있는 경우”로 정의하고 있다[9]. 즉 하자의 개념은 기본적으로 제품이 완성된 상태에서 예상하였던 상태나 성질이 결여된 ‘흠결’을 의미한다. 따라서 ‘하자’와 미완성은 명백히 다른 차원의 문제이다. 소프트웨어 개발공급계약에서 법적으로 도급자는 하자를 이유로 수급자의 보수청구를 거절할 수 없으며, 하

자에 대하여 하자보수청구권을 행사할 수 있다 [3]. 반면 미완성 상태일 경우 도급자는 수급자의 보수청구 요구를 거절할 수 있으며, 계약이행요구권, 계약해제권, 손해배상 청구권 등의 행사가 가능하여진다[3]. 이를 위하여 기성고율 산정이 이루어 질 수 있다. 하자과 버그는 구별되어야 한다. 하자과 버그는 기본적으로 수정-보완되어야 한다는 점에서는 같지만, 하자는 사용자 액션이 예기치 않은 시스템 여러 상황을 불러일으킬 때를 의미하며 일반적으로 개발자의 실수로 불완전한 코드 개발에 기인한다[14]. “버그는 소프트웨어가 예상하지 못한 잘못된 결과를 내거나, 오류가 발생하거나, 착오나 오작동이 발생하는 등의 문제를 뜻한다. 버그는 프로그램의 소스 코드나 설계 과정에서의 오류 때문에 발생한다”[15]. 따라서 하자과 버그는 운영 상태에서 계약된 유지보수 기간 동안 무상으로 보완되는 것이 일반적이라 할 수 있다. 일반적으로 하자는 해당 소프트웨어 서비스 운영 중 내외부적 요인으로 예를 들어 예상하지 못하는 입력 값, 연관 시스템과의 인터페이스 불일치, 운영환경과의 불일치 등으로 발생할 수 있는 반면, 버그는 시스템 기능에 문제를 야기할 수 있는 문제점을 소스코드에 내포하고 있는 경우라고 할 수 있다 [14][15][16].

### 3.3 소프트웨어 기성고

‘기성고’ 또는 ‘기성고율’ 개념은 주로 건설-건축 도급/수급계약관계에서 법적 분쟁이 발생하였을 때 쓰이는 개념으로, 계약 대비 시공된 공정 비율 감정에 따른 댓가 산정 목적으로 이용된다 [10]. 기성고는 기본적으로 미완성을 전제로 하고 있으며, 기성고가 100%일 때는 완성된 상태로 수급자는 하자보수의 의무를 가지는 반면, 도급자는 보수지급의 의무를 가진다[3]. 소프트웨어 프로그램 또한 건설-건축 대상물과 같이 도급계

약의 대상으로 볼 수 있으나, 공정의 진행 방법에 있어서 차이를 가진다. 공학적인 측면에서 보면 요구분석->설계->개발->검수(통합테스트)->이전 및 하자보수의 단계를 거치는 공통점이 있으나, 건축-건설 공정에서는 각 단계별 전문적이고 실제적인 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. 그러므로 소프트웨어 기성고율 분쟁에서 소프트웨어 개발 단계별 가중치와 개발 단계별 도급자/수급자의 수행 역할을 분석하여, 이를 함께 고려한 기성고율을 산정하는 것이 올바르다 할 수 있다.

### 3.4 소프트웨어 완성도/기성고율 분쟁에서 하자 또는 미완성 책임규명

소프트웨어 공학에 따르면, 소프트웨어 개발공정 단계에 따라 도급자와 수급자의 수행작업 분담 및 협업의 중요성이 강조되고 있다[2]. 이는 소프트웨어 제작물이 지적창작물로서 도급 계약 당시 시각적으로 표현될 수 없는 많은 도급자의 요구사항(Requirements)이 존재할 수 있으며, 도급자의 협력 없이 수급자는 도급자가 원하는 개발계약 대상물을 완성하기 어렵다는 것을 의미한다. 도급계약의 기초자료로는 도급자가 제공하는 제안요청서(RFP: Requirement for Proposal)와 수급자가 제시하는 제안서(Proposal) 있으나, 이는 일반적으로 추상적 제안 수준을 넘지 못하며, 도급계약 후 요구분석(Requirements Analysis) 단계에서 수급자의 주도하에 도급자의 협력으로 점차 구체성을 더하게 된다. 요구분석 단계 결과물을 토대로 수급자는 디자인을 수행하며, 도급자는 요구분석 결과물과 디자인 단계 결과물을 검토하여 개발 승인 여부를 판단한다. 개발 단계에서 또한 수급자는 프로그램 모듈 별 산

출물을 도급자와 함께 테스트하여 도급자의 요구 사항이 올바르게 반영되었는지 피드백을 받을 필요가 있다. 완성 후 통합테스트는 도급자가 지정한 전문가 또는 도급자가 주도적으로 진행하며 수급자는 테스트를 돕는 역할을 수행한다. 위와 같이 소프트웨어 개발 공정은 도급자와 수급자의 협력이 필수적으로, 어느 한쪽의 협력이 없으면 개발 공정의 마지막에 이르러서야 문제가 발견되는 것이 일반적이며, 이는 법적 분쟁으로 이어진다. 따라서 분쟁의 책임소재를 밝히는 것은 완성도/기성고율 감정의 중요한 부분이라 할 수 있다.

#### 4. 소프트웨어 완성도감정과 소프트웨어 기성고율감정 분리 제언

3장에서 살펴본 바와 같이, 소프트웨어 완성도에 대한 법률적 정의는 당초 예정된 개발공정이 완료된 소프트웨어가 도급자(사용자)의 요구를 얼마나 만족시키고 있는지에 관계된다. 이 때 도급자의 요구는 기능상 품질에 대한 요구와 사용상 품질에 대한 요구로 나누어 볼 수 있다. 기능상 품질에 대한 요구는 정량적 평가가 가능하며, 현행 소프트웨어 완성도 감정평가는 기능상 품질에 대한 평가이다. 기능상 품질에 대한 완성도 감정의 대상이 될 수 있는 소프트웨어는 따라서 계약 시 '예정된 개발공정'이 완료된 소프트웨어에 한정되어야 한다. 즉 통합테스트를 거쳐 최소한 납품검사가 완료된 소프트웨어에 대하여 완성도를 이유로 도급자의 불만제기가 있을 경우 어떠한 하자 및 버그가 존재하는 지? 전체 소프트웨어 기능 중 하자를 가지는 부분이 어느 정도 인지? 여부에 대한 판단을 구하는 것이라고 할 수 있다. 사용상의 품질에 대한 감정은 하나의 큰 연구 주제로서 본 논문에서는 다루지 않는다.

3.2장에서 살펴본 기성고율은 제작품 완성을 위한 단계별 개발 진척도를 의미한다. 따라서 기성고율 감정을 완성도 감정의 범주에 포함시키는 것은 맞지 않는 것으로 보인다. 현재 소프트웨어 개발 프로젝트의 대부분은 “요구분석 단계->설계 및 디자인 단계->개발 단계(단위 기능 테스트 포함)->통합테스트->검수 및 인수단계->운영 및 유지보수단계”로 유기적으로 이어지는 소프트웨어 공학 프로세스를 따르고 있다고 할 수 있다. [7] 연구결과에서 본 바와 같이 소프트웨어 개발 프로젝트 중 23.8%는 다양한 이유로 개발 공정이 완료되지 못하고 중단되고 있으며, 이는 도급/수급 계약관계에서 법적 분쟁으로 이어질 개연성이 매우 크다. 또한 법적 분쟁이 발생할 경우 개발공정이 완료되지 못한 책임에 대한 법리적 다툼이 중요한 요소임에 틀림없다. 따라서 소프트웨어 기성고율 감정에서 소프트웨어 개발 프로세스 단계에 대한 비율 감정이 요구되며, 각 단계별 도급자/수급자의 역할 수행 여부 및 정도를 함께 고려하여 전체 프로세스에서 기성고율을 산정하는 것이 올바르다 할 수 있다. 예를 들어 전사적자원관리시스템 (ERP: Enterprise Resource Planning) 시스템에 대한 분쟁 사례 경우, 수급자는 도급자의 협력을 받아 도급자 업무 프로세스에 대한 정확한 요구분석을 필요로 하는데, 수급자 대부분은 보유한 ERP 시스템을 도급자의 요구에 맞추어 커스터마이징 하는 방식으로 개발이 이루어진다. 즉 수급자는 대부분의 시스템 기능 구현이 완료된 상태에서 개발단계를 수행하게 되는데, 기성고율 감정에서 단순히 개발된 시스템 기능만을 대상으로 기성고율 산정할 경우 오류가 발생할 수 있다. 즉 위와 같은 ERP 시스템 개발의 핵심은 사용자의 요구분석 및 비즈니스 로직 분석이라 할 수 있다. 소프트웨어 기성고율 감정에서 위 언급된 점을 고려하는 것이 올바른 기성고율 감정이라 할 수 있다.

## 5. 결 언

기존 소프트웨어 완성도감정 범주에는 완성된 소프트웨어에 포함된 하자, 버그 등에 대한 도급자(사용자)의 기능상 불만족 정도에 대한 평가뿐만 아니라, 기능상 미완성된 소프트웨어 기성고율을 산정하는 것까지, 거의 동일한 감정 프로세스를 따르고 있다.

본 논문은 기존 소프트웨어 완성도 감정의 문제점을 적시하고, 소프트웨어 완성도감정과 소프트웨어 기성고율 감정을 명확하게 분리하여 감정 프로세스 및 감정 방법을 달리 하여야 하는 당위성을 주장하고 있다.

소프트웨어 완성도 감정 또는 기성고율 감정에서 종종 미완성된 부분으로 감정된 기능에 대한 손해액 또는 완성을 위해 소요되는 비용 산정이 감정 요청되는 것을 볼 수 있다. 완성도/기성고율 감정에서 비용 산정은 해당 감정의 부수적 감정분야로서 별도로 고려되어야 한다.

소프트웨어 완성도 감정평가에서 사용상 품질에 대한 요구는 기능상 품질 못지않게 중요한 부분이나, 정량적 평가가 상대적으로 어려운 사용성 감정평가를 필요로 하는 점에서 거의 이루어지고 있지 않다. 그러나 ‘사용자의 요구를 만족시키는 정도’를 완성도라고 정의하고 있는 판례와 법률적 규정을 준수하기 위하여 이에 대한 전문가 감정이 필요하다고 할 수 있다. Niels의 연구에 따르면, 소프트웨어 실제 사용자 테스트 없이, Software Ergonomics 전문가 감정만으로도 소프트웨어 질적 품질에 영향을 미치는 사용상의 결점 대부분을 찾아 낼 수 있다고 한다[11]. 사용자 관점에서 소프트웨어 질적 품질에 대한 감정 방법은 향후 연구 주제로 다루기로 한다.

## 참 고 문 헌

- [1] 김시열, 저작물 감정제도의 효과 및 개선제도 연구, 한국저작권위원회, 2013
- [2] I. Sommerville, Software Engineering, 9th Edition, 2011
- [3] 김태경, 컴퓨터프로그램 개발공급계약에 있어서 하자를 둘러싼 법률적 문제, 인권과정의, vol. 425, 2012
- [4] 한국저작권위원회, 저작권 관련 감정사건 판례집(1), 한국저작권위원회, 2010
- [5] 한국저작권위원회, 저작권 관련 감정사건 판례집(2), 한국저작권위원회, 2013
- [6] 한국저작권위원회, 저작권 관련 감정사건 판례집(3), 한국저작권위원회, 2016
- [7] <https://www.bcs.org/articles-opinion-and-research/a-study-in-project-failure/>
- [8] <https://web.stanford.edu/class/archive/cs/cs295/cs295.1086/>
- [9] 민법주해XV, 박영사, 2000
- [10] <https://www.koscaj.com/news/articleView.html?idxno=205647>
- [11] J. Nielsen, 10 Usability heuristics for user interface design, 1994
- [12] 감정사건 판례집, 한국저작권위원회, 2016
- [13] 대법원 2006. 10.13 선고 2004다21862판결
- [14] Andrew Troelsen, C# and the .NET Platform, a! apress, 2001
- [15] <https://ko.wikipedia.org/wiki/>
- [16] <https://www.lawinsider.com/dictionary/software-defect>

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 선임연  
구원
- 1997.3- 현재 배재대학교 정보통신공학과  
교수
- 2012.2~2018.8 독일 힐데스하임 대학교 객  
원교수
- 2006.8~ 현재 한국저작권위원회 SW감정전  
문위원
- <주관심분야> 지식기반시스템, SW인간공  
학, 시맨틱웹, SW저작권 보호 및 감정