# 소프트웨어 개발 분쟁해결을 위한 평가방안 연구

김우성*, 황진옥, 민성기
고려대학교 컴퓨터정보통신대학원
e-mail:ds5gdq, withmind, sgmin@korea.ac.kr

# A Study on Valuation Method for Dispute Resolution of Software Development

Woo-Sung Kim*, Jin-Ok Hwang, Sung-Gi Min
Graduate School of Computer and Information Technology,
Korea University

요 약

본 논문은 소프트웨어의 완성도와 관련하여 발주사와 개발사간의 분쟁이 있을 시에 필요한 분쟁조정을 위한 평가방법을 연구하였다. 먼저 소프트웨어 개발 과정에서 발생할 수 있는 분쟁의 유형을 분류하였고, 객관적인 평가를 위한 감정 절차를 분류하였다. 본 연구에서는 분쟁조정의 객관성 확보를 위한 가중치 설정, 각 기능들에 대한 중요도를 설정하여 어느 정도의 완성도를 보였는지를 정량적으로 평가하였다. 또한, 프로그램의 복제 문제를 판단하기 위하여 필요한 감정 항목 설정 및 도용 여부를 판단하기 위한 기본 자료로 활용될 수 있을 것으로 기대한다.

## 1. 서론

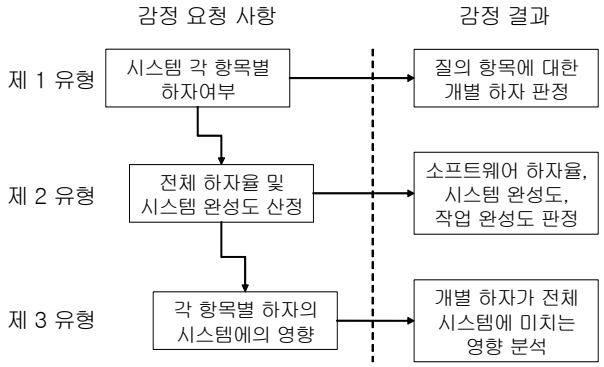
본 연구는 다음과 같은 상황을 모델링하여 이를 가정하였다. 즉, “A가 B를 상대로 소송을 제기한 것으로, B는 A가 발주한 “C 시스템 개발(가명)”에 대한 업무제휴 계약을 체결하여, 시스템 개발을 추진하였으나 완료하지 못하여 소송이 발생하였다.” 이에 관련기관이 요청한 감정사항은 업무제휴계약서 및 개발작업 명세서 대비 완성도를 판단하는 것이다.

따라서 본 연구에서는 개발작업 명세서에 기술된 항목들을 대・중・소분류로 정리하고 최종적으로 소분류에 해당하는 최하위 기능들을 감정대상으로 설정하였다.

본 연구는 현재까지 개발된 프로그램이 제출된 “업무제휴 계약서” 및 “개발작업 명세서”에 근거하여 어느 정도의 완성도를 보였는지를 정량적으로 평가하는 것이 핵심이라 판단된다. 프로그램 개발 완성도란 업무제휴 계약서 및 개발작업 명세서상의 기능들을 개발사에서 개발・완료하여 발주사가 활용 가능한 상태로 제공한 프로그램이 어느 정도인지를 정량화한 것이다.

## 2. 소프트웨어 개발 분쟁 유형 분석

소프트웨어 개발 분쟁에 따른 감정 요청에 대한 유형별 분류를 그림 1과 같이 구분할 수 있다.




(그림 1) 소프트웨어 개발 분쟁의 유형별 분류

유형 1은 개별 항목에 대하여 하자 여부를 측정하

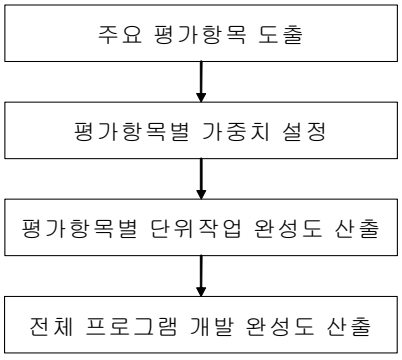
는 것이다. 유형 2는 각 항목별 하자 여부, 전체 하자율, 시스템 완성도를 판정하기 위함이다. 유형 3은 계산된 하자율을 통하여 전체 시스템에 미치는 영향을 감정인의 전문 지식으로 판단하는 것이다. 전문 지식을 갖춘 감정인이 필요한 부분으로 유형에서 평가된 항목을 통하여 전체시스템에 미치는 영향을 측정하기 위함이다.

본 논문에서는 유형 1에 대한 사례 분석을 통하여 표준화된 감정 기법의 기초를 제공하고자 한다.

## 3. 소프트웨어 개발 분쟁 감정

### 3. 1 감정절차

본 감정의 객관적 평가를 위해 평가과정을 그림 2와 같이 4단계로 분류하여 진행하였다.




(그림 2) 감정 절차

#### (1) 평가항목 도출

본 연구와 관련하여 유일하게 시스템이 가져야 하는 기능이 구체적으로 명시된 것이 “개발작업 명세서”이다. 개발작업 명세서에 기술된 단위업무들을 기술하면 표 1과 같다.

<표 1> 항목별 최하위 기능 현황

| 대분류 | 중 분 류 | 최하위 기능의 개수 |
|---|---|---|
| A | A1 | 38 |
| | A2 | 34 |
| | A3 | 8 |
| B | B1 | 3 |
| | B2 | 5 |
| | B3 | 5 |
| C | C1 | 4 |
| 합 계 | | 97 |

표 1의 내용을 근거로 본 감정에서는 중분류에 해당하는 7개의 업무내용을 주요 평가항목으로 지정하여 수행하였다.

#### (2) 평가항목별 가중치 설정

먼저 가중치 1은 평가항목별 개발에 소요되는 투입시간을 근거로 하여 가중치를 산정하였다. 이를 위한 공식은 식(1)과 같으며, 단위는 %이다.

$$W_{1i}(\%) = \frac{T_i}{\sum_{i=1}^{n} T_i} \quad \cdots\cdots\cdots\cdots (1)$$

여기에서 $W_i$는 $i$번째 평가항목에 대한 가중치를 의미하며, $T_i$는 $i$번째 평가항목의 개발기간을 의미한다. 또한 $n$은 평가항목의 개수를 의미한다.

다음으로 고려해야 하는 가중치 2는 투입 개발자의 규모 및 고급정도를 작업의 중요도로 반영키 위한 것이다. 이와 같은 두 번째 가중치를 공식으로 표현하면 다음 식 (2)와 같다.

$$W_{2i}(\%) = \frac{MM_i}{\sum_{i=1}^{n} MM_i} \quad \cdots\cdots\cdots\cdots (2)$$

여기서 $W_{2i}$는 가중치 2를 구하기 위해 $i$번째 평가항목에 대한 가중치를 나타내며, $MM_i$는 $i$번째 평가항목의 Man/Month를 나타낸다. 또한 $n$은 평가항목의 개수를 나타낸다.

마지막으로 최종적으로 각 항목에 적용할 가중치 3은 다음 식(3)과 같다.

$$W_i(\%) = \frac{W_{1i} \times W_{2i}}{\sum_{i=1}^{n} (W_{1i} \times W_{2i})} \quad \cdots\cdots\cdots\cdots (3)$$

여기서 $W_i$는 $i$번째 평가항목에 대한 가중치 3을 의미하며, $W_{1i}$는 $i$번째 평가항목의 가중치 1이며 $W_{2i}$는 $i$번째 평가항목의 가중치 2이다. 또한 $n$은 평가항목의 개수를 의미한다. 따라서 식 (3)의 계산에 의해 최종적으로 각 항목에 적용될 최종 가중치(가중치 3)의 결과는 표 2와 같다.

#### (3) 각 항목별 단위작업 완성도 산출

각 평가항목별로 개발이 완성된 것으로 판단되는 내용들을 정리하면 표 3과 같다.

<표 2> 공식 3에 의한 결과

| 대분류 | 중 분 류 | 가중치 1(%) | 가중치 2(%) | 최종가중치 (%) |
|---|---|---|---|---|
| A | A1 | 20 | 26.4 | 35.0 |
| | A2 | 30 | 14.4 | 28.6 |
| | A3 | 15 | 9.1 | 9.1 |
| B | B1 | 5 | 11.7 | 3.9 |
| | B2 | 5 | 15.6 | 5.2 |
| | B3 | 15 | 9.1 | 9.1 |
| C | C1 | 10 | 13.7 | 9.1 |
| 합 계 | | | | 100.0 |

<표 3> 평가항목별 단위작업 완성도

| 대분류 | 중 분 류 | 최하위 기능개수 | 완성기능 개수 | 작업 완성도 |
|---|---|---|---|---|
| A | A1 | 38 | 9 | 23.68 |
| | A2 | 34 | 8 | 23.53 |
| | A3 | 8 | 1 | 12.50 |
| B | B1 | 3 | 3 | 100 |
| | B2 | 5 | 1 | 20.0 |
| | B3 | 5 | 0 | 0 |
| C | C1 | 4 | 0 | 0 |
| 합 계 or 평 균 | | 97 | 22 | 22.68 (평균) |

### 3.2 부문별 평가내역

평가항목별 작업완성도를 계산하기 위해 "개발작업 명세서" 상에 기술된 개발해야 되는 기능들과 현재까지 개발사에서 개발된 내용들을 기 제출된 자료들을 중심으로 각 항목별로 비교・분석하였다.

#### (1) A1업무 완성도

전체적으로 23.68%의 완성률을 보이고 있으며 또한 하나의 단위 기능으로 기술된 사항이 부분적으로 개발되었으며 부분적으로 개발된 사항들은 발주사에서는 경제적으로 활용할 수 없기 때문에 완료되지 못한 것으로 판정하였다.

#### (2) A2업무 완성도

전체적으로 23.53%의 완성률을 보이고 있으며 기타 사항들은 부분적으로 개발이 이루어짐.

#### (3) A3업무 완성도

전체적으로 12.5%의 완성률을 보이고 있으며 Internet 및 mobile 신청업무, 회원관리, 계좌이체 업무 등이 개발되었다는 자료를 발견할 수 없음

#### (4) B1업무 완성도

제출된 개발결과보고서의 화면설계 및 내용으로 볼 때 개발된 것으로 판단됨. 회사별 사용정산 청구내역 프로그램, 조합별 상호정산 내역 프로그램들이 개발된 것으로 판단되었다.

#### (5) B2업무 완성도

이용금액 및 원장차감 업무는 개발된 것으로 판단되나 환불수신 업무는 환불접수신청 기능만 개발되고 다른 환불내용들은 개발되지 않은 것으로 판단되며 기타 나머지 기능들은 개발되지 못한 것으로 판단하였다.

#### (6) B3업무 완성도

Cash 이용에 대한 정산에 대한 업무 전체가 개발되지 못한 것으로 판단됨.

#### (7) C1업무 완성도

Loyalty 업무에 대한 모든 기능들이 개발되지 못한 것으로 판단됨.

### 3.3 개발 완성도

이상의 검증내용을 토대로 각 평가항목별 적용가중치(표 2 참조) 및 작업 완성도를 계산(표 3 참조)하여 구한 최종적인 전체 프로그램 개발 완성도를 표 4에 기술하였다. 이에 대한 공식은 다음 식(4)와 같다.

$$C(\%) = \sum_{i=1}^{n} (Ci \times Wi) \quad \text{.....} \quad (4)$$

여기서, C(%)는 전체 시스템의 개발완성도를 의미하며, $C_i$는 $i$번째 평가항목의 단위작업완성도를, $W_i$는 $i$번째 평가항목의 최종가중치를 의미한다.

<표 4> 전체 프로그램 개발 완성도

| 대분류 | 중 분 류 | 작업완성도(%) | 최종가중치 (%) | **개발완성도(%)** |
|---|---|---|---|---|
| A | A1 | 23.68 | 35.0 | **8.29** |
| | A2 | 23.53 | 28.6 | **6.73** |
| | A3 | 12.50 | 9.1 | **1.14** |
| B | B1 | 100 | 3.9 | **3.90** |
| | B2 | 20.0 | 5.2 | **1.04** |
| | B3 | 0 | 9.1 | **0** |
| C | C1 | 0 | 9.1 | **0** |
| 합 계 | | | | **21.10** |

소프트웨어 개발 완성도는 발주자가 발주한 프로그램이 개발자에 의해 개발되어 발주자가 경제적으로 활용할 수 있는 프로그램이 몇 개인지가 핵심 사항이다. 이러한 관점으로 볼 때 본 감정의 최종 완성도는 21.1%로 산출되었다. 전체적으로 프로그램 완성도가 낮은 이유는 개발사의 전체적인 개발 내용을 검토해 보았을 때 대부분의 기능들이 완전한 상태가 아닌, 부분적으로 개발이 되어 있거나 현재 개발 상태에 있는 내용들이 있기 때문이라 판단된다.

## 4. 결론

본 논문에서는 소프트웨어 개발 과정에서 발생할 수 있는 분쟁의 유형을 분류하고, 그 중에서 시스템의 각 항목별 하자 여부에 대한 감정을 위하여 필요한 내용에 대하여 연구하였다. 감정을 위한 평가항목을 도출하고, 각 평가항목에 대한 가중치를 할당하여 단위 작업 및 전체 소프트웨어의 개발 완성도를 산출하였다.

평가항목의 설정 부분에서 분쟁 당사자 사이의 이견이나 논란의 소지가 있을 수 있어서, 소프트웨어 개발 계약 시에 작성된 개발작업 명세서의 내용을 근거로 평가항목을 도출하였다.

향후 평가항목에 대한 세분화 방안과 그 합리성에 대한 연구가 계속 수행되어야 할 것으로 생각된다. 예를 들어 평가 항목별 가중치를 설정할 때 개발 인력의 Man/Month 뿐만 아니라 각 인력들의 경력이나 보유 기술 수준에 대한 내용도 적절하게 반영할 수 있을 것으로 보인다.

또한 소프트웨어의 개발 완성도 문제와 더불어 고려할 부분은 프로그램의 복제 문제이다. 소프트웨어 개발이 증대하면서 제기되고 있는 프로그램의 복제 여부를 판단하기 위한 감정 항목 설정 및 도용 여부의 판단에 대한 연구가 진행되어야 할 것이다.

## 참고문헌

[1] J.Whitten. L.Bentley, *System Analysis & Design Methods*, IRWIN, 1994.

[2] T.Gildersleeve, *Successful Data Processing Systems Analysis*, 2nd ed, Englewood Cliffs, N.J. Prentice Hall, 1985.

[3] ITU-T, Recommendation Z.120: Message Sequence Chart (MSC). ITU, 2000.

[4] H. J. Bang, Extending SMV for Real-Time Verification of Statecharts, M.S. Thesis, KAIST, 2003.

[5] 소프트웨어공학연구회지, 한국정보처리학회, 2002. 3.

[6] W. Chan, R. J. Anderson and et al. "Model Checking Large Software Specifications," Software Engineering," Vol.24, No.7, 1998.