

소프트웨어 개발비 대가기준 개선에 관한 연구

권 기 태[†] · 변 분 희^{**}

요 약

소프트웨어 개발 초기 단계에서 소프트웨어 개발비용을 정확하게 예측하는 것은 프로젝트의 성패를 결정짓는 중요한 요소이다. 우리나라에서는 소프트웨어 개발비용 산정을 위해 소프트웨어 사업대가기준을 마련하였으며, 이것으로 소프트웨어 사업의 예산수립이나 발주 시 적정비용 산정에 활용하고 있다. 그러나 소프트웨어 기술 발전과 환경 변화에 대응하여 그에 적합한 사업대가기준이 되기 위해서는 지속적인 개선 작업이 필요시 된다. 따라서 본 논문은 소프트웨어 기술 발전과 여러 가지 환경 변화 요인을 반영하여 사업대가기준의 정확성을 향상시키고자 한다. 이를 위해 14가지 일반시스템특성을 반영하는 값 조정 인자의 도입과 실무자들에 의해 필요성이 제기되고 있는 납기 보정계수의 도입을 통한 개선 방안을 제시하였다. 개선 방안의 정확도를 평가하기 위해 실제 데이터를 이용하여 MMRE, PRED 값을 측정해 보았다.

결론적으로, 규모 보정과 함께 14가지 일반 시스템 특성을 반영한 값 조정인자에 추가 조정인자를 적용한 보정계수를 적용함으로써 비용 산정의 정확도가 뚜렷이 향상됨을 확인할 수 있었다. 또한 제안된 두 가지 납기 보정계수의 정확도를 평가하였다.

키워드 : 소프트웨어 비용산정, 소프트웨어 사업대가기준, 기능점수, 값 조정 인자, MMRE, PRED

A Study for Enhancing the Criterion of the Software Cost Estimation

Ki Tae Kwon[†] · Boon Hee Byun^{**}

ABSTRACT

It is very important that they accurately predict the software development cost in the early stage of a software development. Because cost estimations are required when bidding for a contract or determining whether a project is feasible in terms of a cost-benefit analysis. The criterions of the software cost estimation was set up to calculate software development cost more exactly, which is applied to made up a budget of the software business or to calculate a suitable cost to start the business in our country.

However, as the software technology and environment are changing very rapidly, it need to enhance the criterion of the cost estimation continuously. Therefore, we tried to apply technology of software and a variety of factors of environment changes in present. Most of all, we proposed an introduction and readjustment of the adjustment factor applying 14 general system characteristics to improve the accuracy of the cost estimation and the schedule adjustment factor that is required by practitioners. For evaluating the accuracy in terms of the real data, we have used MMRE & PRED.

In result, we proved that the accuracy was clearly improved by applying the scale factor and readjusted VAF with 14 general system characteristics. Moreover, we evaluated the accuracy of the schedule adjustment factor.

Key Words : Software Cost Estimation, Criterions of the Software Cost Estimation, Function Point, Value Adjustment Factor, MMRE, PRED

1. 서 론

소프트웨어 개발 초기 단계에서 소프트웨어 개발비용을 정확하게 예측하는 것은 프로젝트의 성패를 결정짓는 중요한 요소이다. 비용을 과대 추정하여 고객이 프로젝트를 취소할 수도 있고, 실제 소요될 비용보다 비용을 적게 예측함으로써 개발 업체는 실제로 이윤을 남기지 못하고 많은 시

간을 낭비하게 될 수도 있다. 소프트웨어 생명주기 초기에 개발비용을 정확히 예측하면 프로젝트 관리자들은 어떤 자원이 필요한지 적합한 자원들을 언제 배정해야 할지를 알 수 있다[1].

정확한 소프트웨어 개발비용 예측을 위해 소프트웨어공학 분야에서는 개발비용과 구성원의 특성과의 관련성, 프로젝트 요구 사항 및 개발 기간, 노력, 비용에 영향을 미치는 다양한 요소들을 발견하는 기술들을 개발하기 위해 노력해 왔다. 소프트웨어 비용산정 모델에 관한 주요 연구는 1965년 169개 소프트웨어 프로젝트의 104가지의 속성에 관한 SDC의 광범위한 연구로 시작되었다. 이 모델을 기초로 1960년대

* 본 연구는 정보통신부의 정보통신연구개발사업(한국전산원 위탁과제: NCA VI-PER-04027)의 연구결과임.

† 종신회원: 강릉대학교 컴퓨터공학과 교수

** 준 회 원: 강릉대학교 컴퓨터공학과 박사수료

논문접수: 2006년 7월 20일, 심사완료: 2006년 9월 12일

후반과 1970년대 초반 부분적으로 유용했던 일부 모델들이 유도되었다. 1970년대 후반에 SLIM, Checkpoint, PRICE-S, SEER, COCOMO 등과 같은 알고리즘 모델이 개발되었다.

이들 소프트웨어 비용산정 모델의 개발자 대부분이 동일한 시기에 비용산정 모델들을 개발하기 시작했지만, 그들은 모두 유사한 딜레마에 빠졌다. 즉, 소프트웨어의 크기가 커지고 복잡도가 높아짐에 따라 소프트웨어 개발비용을 정확하게 예측하기가 점점 더 어렵다는 것이다. 알고리즘 모델 자체의 문제점과 더불어 매우 빠르게 변화하는 개발 환경의 영향으로 정확도 높은 알고리즘 모델을 개발하기란 매우 어렵다.

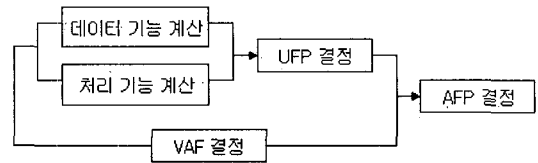
국내의 비용산정 모델로는 1989년 제정된 소프트웨어 사업 대가기준을 들 수 있다. 소프트웨어 사업 대가기준은 1989년 이래로 소프트웨어 기술발전과 환경변화에 따라 계속적으로 개정되어 왔으며, 현재는 2004년 개정 고시된 소프트웨어 사업 대가기준을 활용하고 있다[2]. 현행 소프트웨어 대가기준에서는 4가지 보정치를 반영하여 소프트웨어 개발원가를 계산하고 있다(그림 1). 그러나 이 4가지 보정 계수로는 소프트웨어 개발 시 요구되는 다양한 특성들을 반영하지 못 하며, 또한 실무자들에 의해 납기 보정계수의 필요성이 대두되고 있다[3]. 따라서, 본 논문에서는 비용산정의 정확성을 높이고 현실성 있는 소프트웨어 사업대가기준을 위해 기본 보정 계수의 조정 및 새로운 보정 계수 도입을 제안한다.

2장에서는 소프트웨어 사업대가기준 및 보정계수를 설명하며, 3장에서는 기본 보정계수의 조정 및 신규 보정계수의 도입 방안을 제안하고, 4장에서는 개선 모델의 성능을 평가한다. 끝으로 5장에서는 결론 및 향후 연구과제로 맺는다.

2. 관련 연구 및 연구배경

2.1 소프트웨어 사업 대가 기준

소프트웨어 산업진흥법 제22조 및 동 시행령 제15조에 의 해 정보통신부 고시로 규정하고 있는 소프트웨어 사업대가 기준은 국가 또는 공공기관에서 소프트웨어 개발, 데이터베



(그림 2) FP 계산과정

이스 구축, 정보전략계획 수립등의 정보화사업을 추진함에 있어 정보통신기술의 발전 및 사회적 여건변화에 유연하게 대처하고, 소프트웨어 산업과의 선순환적 구조를 가질 수 있도록 소프트웨어사업에 대한 예산수입이나 발주 시 적정 비용 등을 산정하기 위한 기준 제공을 목적으로 하는 한국형 소프트웨어 비용 산정 모델이다. 소프트웨어 사업 대가 기준을 이용한 비용 산정 과정은 (그림 1)과 같다[2].

소프트웨어 규모 산정 시 LOC(Line Of Code)는 작성된 프로그램의 코드 라인 수로써 측정하는 방식이며, 기능점수(FP, Function Point)는 시스템의 기능성으로 측정하는 방식으로 기능점수를 계산하는 과정은 (그림 2)와 같다[4].

현행 소프트웨어 사업 대가기준에서는 기능점수 측정 시 값 조정 인자(VAF, Value Adjustment Factor)를 반영하지 않은 미조정 기능점수(UFP, Unadjusted Function Point)를 이용하고 있다. 그러나, 비용 산정시 기능점수뿐 아니라 소프트웨어 개발환경 및 기술적 특성을 반영하기 위해서는 보정치 산정 시 14가지 일반 시스템 특성(GSC, General System Characteristics)을 반영하는 값 조정 인자를 적용하는 것이 적합할 것이다.

2.2 소프트웨어 사업 대가 기준의 보정치

소프트웨어 비용 산정에서 각종 보정치를 반영하는 것은 비용 산정의 정확도를 높이기 위한 중요한 요소이다.

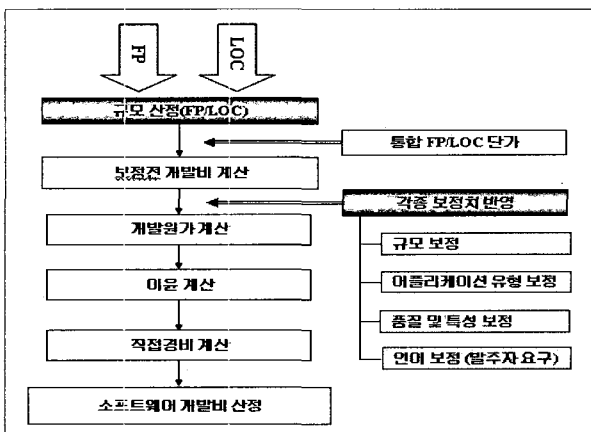
현행 소프트웨어 사업 대가 기준에서 사용하고 있는 보정치는 언어 보정, 어플리케이션 유형 보정, 규모 보정, 품질 및 특성 보정 등 4가지 보정계수가 있다. 보정계수는 아래의 <표 1>, <표 2>, <표 3>, <표 4>와 같다.

현행 대가기준의 규모 보정계수는 코드 라인 수의 경우 <표 1>를 기준으로 보간법에 의하여 계산하고, 10,000코드 라인 미만의 경우는 0.65를 적용하며 1,000,000코드라인 이상의 경우는 10,000코드라인당 0.0005씩 추가한다.

기능점수의 경우 아래의 식(1)에 의해 계산하며, 300기능

<표 1> 코드라인당 보정계수

| 규모(코드라인수) | 보정계수 |
|-----------|------|
| 10,000 | 0.65 |
| 30,000 | 0.85 |
| 70,000 | 0.97 |
| 150,000 | 1.05 |
| 300,000 | 1.11 |
| 500,000 | 1.17 |
| 700,000 | 1.21 |
| 1,000,000 | 1.24 |



(그림 1) 소프트웨어 비용산정 과정

점수 이하인 경우는 0.65를 적용한다[2].

$$\text{규모 보정계수} = 0.108 \times \log_e FP + 0.2229 \quad (1)$$

어플리케이션 유형 보정계수는 아래의 <표 2>에서와 같이 가장 낮은 보정계수 값인 '1.0'을 적용하는 업무처리용 어플리케이션에서부터 가장 높은 보정 계수 값인 '2.2'를 적용하는 지휘통제용 어플리케이션까지 총 8가지로 분류하여 적용한다.

언어 보정계수는 발주자가 특정 언어를 요구할 경우에 사업대가기준에 명시된 소프트웨어 개발 단계 중 구현 단계와 시험 단계에만 적용되는 보정 항목으로 <표 3>과 같다.

분산처리, 성능, 신뢰성, 다중 사이트 수 등 시스템 특성을 추가적으로 보정하기 위해 4가지 보정항목을 반영하는 보정계수 값은 <표 4>를 기초로 영향도를 결정한다.

품질 및 특성 보정계수는 아래의 식(2)에 적용하여 계산한다[2].

$$\text{품질 및 특성 보정계수} = 0.025 \times \text{총 영향도} + 1 \quad (2)$$

그러나, 이 네 가지 보정계수만으로 다양한 비용 요소들의 보정을 충분히 했다고 볼 수 없다. 실제로 대표적인 상용 비용 산정 모델과 현행 대가기준의 보정계수를 비교하면 현행 대가기준에서는 인적 속성에 해당하는 프로그래머 능력(Personnel attributes), 인력 연속성(Personnel continuity),

<표 2> 어플리케이션 유형 보정계수

| 어플리케이션 유형 | 범 위 | 보정계수 |
|-----------|---|------|
| 업무처리용 | 인사, 회계, 급여, 영업 등 경영관리 및 업무처리용 소프트웨어 등 | 1.0 |
| 과학기술용 | 과학계산, 시뮬레이션, 스프레드시트, 통계, OR, CAE 등 | 1.2 |
| 멀티미디어용 | 그래픽, 영상, 음성 등 멀티미디어 응용분야, 지리정보시스템, 교육, 오락용 등 | 1.3 |
| 지능정보용 | 자연어처리, 인공지능, 전문가시스템 | 1.7 |
| 시스템용 | 운영체제, 언어처리프로그램, DBMS, 인간과 기계 인터페이스, 윈도시스템, CASE, 유틸리티 등 | 1.7 |
| 통신제어용 | 통신프로토콜, 에플리케이션, 교환기소프트웨어, GPS 등 | 1.9 |
| 공정제어용 | 생산관리, CAM, CIM, 기기제어, 로봇제어, 실시간, 내장형 소프트웨어 등 | 2.0 |
| 지휘통제용 | 군, 경찰 등 군장비 인력의 지휘통제를 요하는 소프트웨어 | 2.2 |

<표 3> 언어 보정계수

| 언어구분 | 보정계수 |
|--|------|
| Assembly, 기계어, 자연어 | 1.9 |
| COBOL, FORTRAN, PL/1, PASCAL, Ada | 1.2 |
| C, CHILL, C++, JAVA, C#, PROLOG, RPG, UNIX Shell Scripts | 1.0 |
| ABAP4, Delphi, HTML, Power Builder, Program Generator, Query default, Small Talk, SQL, Visual Basic, Statistical default, XML default, Script default(JSP, ASP, PHP 등) | 0.8 |
| EXCEL, Spreadsheet default, Screen painter default | 0.6 |

<표 4> 품질 및 특성 보정항목

| 품질 및 특성 항목 | | 영향도 판단기준 | 영향도 |
|------------|--------------------------------|--|-----|
| 분산처리 | 어플리케이션이 구성요소간에 데이터를 전송하는 정도 | 분산처리에 대한 요구사항이 명시되지 않음 | 0 |
| | | 클라이언트/서버 및 웹 기반 어플리케이션과 같이 분산 처리와 자료 전송이 온라인으로 수행됨 | 1 |
| | | 어플리케이션상 처리기능이 복수개의 서버 또는 프로세서 상에서 동적으로 상호 수행됨 | 2 |
| 성능 | 응답시간 또는 처리율에 대한 사용자 요구수준 | 성능에 대한 특별한 요구사항이나 활동이 명시되지 않으며, 기본적인 성능이 제공됨 | 0 |
| | | 응답시간 또는 처리율이 피크타임 또는 모든 업무시간에 중요함. 연동 시스템의 처리 마감시간에 대한 제한이 있음. | 1 |
| | | 성능 요구사항을 만족하기 위해 설계 단계에서부터 성능 분석이 요구되거나, 설계, 개발, 구현 단계에서 성능 분석 도구가 사용됨. | 2 |
| 신뢰성 | 장에서 미치는 영향의 정도 | 신뢰성에 대한 요구사항이 명시되지 않으며, 기본적인 신뢰성이 제공됨 | 0 |
| | | 고장 시 쉽게 복구가능한 수준의 약간 불편한 손실이 발생함. | 1 |
| | | 고장 시 복구가 어려우며, 재정적 손실이 만이 발생하거나, 인명피해 위험이 있음. | 2 |
| 다중사이트 | 상이한 H/W와 S/W 환경을 지원하도록 개발되는 정도 | 설계 단계에서 하나의 설치 사이트에 대한 요구사항만 고려됨. 어플리케이션이 동일한 하드웨어 또는 소프트웨어 환경 하에서만 운영되도록 설계됨. | 0 |
| | | 설계 단계에서 하나 이상의 설치 사이트에 대한 요구사항이 고려됨. 어플리케이션이 유사한 하드웨어 또는 소프트웨어 환경하에서만 운영되도록 설계됨. | 1 |
| | | 설계 단계에서 하나 이상의 설치 사이트에 대한 요구사항이 고려됨. 어플리케이션이 상이한 하드웨어 및 소프트웨어 환경 하에서 동작하도록 설계됨 | 2 |

프로그래머 경험(Personnel experience)과 프로젝트 속성에 해당하는 개발 도구 및 기술(Tools and Techniques), 스케줄 제한성(Schedule constraints), 프로세스 성숙도(Process maturity), 개발팀의 응집도(Team cohesion), 보안 이슈(Security issues)등을 거의 반영하지 못하고 있다[3]. 따라서 현행 대가기준의 지속적인 보정계수 개선 및 조정 과정이 필요하다.

2.3 IFPUG의 일반 시스템 특성

현행 대가기준은 소프트웨어 기능 점수 측정 시 값 조정 인자를 반영하는 것이 측정의 정확도에 거의 영향을 미치지 못한다고 하여 미조정기능점수를 이용하였다[2]. 그러나 IFPUG(International Function Point User's Group)의 CPM 4.2에서는 현재의 개발환경에 쉽게 적용할 수 있는 일반 시스템 특성의 정의 및 영향도 선정기준을 발표하였으며, 또한 최근 연구[5]에 의하면 일반 시스템 특성을 반영하는 것이 산정의 정확도를 향상시켰다고 한다. 따라서 기능점수 산정 시 산정의 정확도를 높이기 위해 14가지 일반 시스템 특성을 반영하는 값 조정 인자를 적용하는 것이 바람직할 것이다.

값 조정 인자에 반영되는 14가지 일반 시스템 특성은 아래와 같다[6].

- ① 데이터 통신(Data Communications)
- ② 분산 데이터 처리(Distributed Data Processing)
- ③ 성능(Performance)
- ④ 컴퓨터 자원 제한성(Heavily Used Configuration)
- ⑤ 트랜잭션 비율(Transaction Rate)
- ⑥ 온라인 데이터 입력(On-Line Data Entry)
- ⑦ 최종 사용자 효율성(End-User Efficiency)
- ⑧ 온라인 갱신(On-Line Update)
- ⑨ 복잡한 처리(Complex Processing)
- ⑩ 재사용성(Reusability)
- ⑪ 설치 용이성(Installation Ease)
- ⑫ 운영 용이성(Operational Ease)
- ⑬ 다중 사이트(Multiple Site)
- ⑭ 변경 촉진(Facilitate Change)

<표 5> 14가지 일반 시스템 특성의 영향도

| 영향도 | 영향도 판단기준 |
|-----|--|
| 0 | 존재하지 않거나 영향이 없음(Not Present or No Influence) |
| 1 | 우연한 영향(Incidental Influence) |
| 2 | 보통의 영향(Moderate Influence) |
| 3 | 평균적인 영향(Average Influence) |
| 4 | 중대한 영향(Significant Influence) |
| 5 | 지속적으로 강력한 영향(Strong Influence Throughour) |

각 일반 시스템 특성의 영향도(DI, Degree of Influence)는 <표 5>와 같이 0에서 5까지의 등급으로 평가한다.

아래의 식(3)과 같이 14가지 일반 시스템 특성 모두에 대한 영향도를 합하여 전체영향도(TDI, Total Degree of Influence)를 계산한다.

$$TDI = \sum_{i=1}^{14} DI_i \tag{3}$$

식(3)을 통해 계산한 TDI를 아래의 식(4)에 대입하여 값 조정 인자를 계산한다[6].

$$VAF = (TDI \times 0.01) + 0.65 \tag{4}$$

3. 소프트웨어 사업대가 기준 개선

3.1 보정계수 조정

현행 소프트웨어 대가기준의 보정계수 중 어플리케이션 자체의 특성을 반영하는 보정계수는 어플리케이션 유형 보정계수와 품질 및 특성 보정계수이다.

현행 대가기준에서 이 두 가지 보정계수를 통하여 나타낼 수 있는 보정계수 가능 범위와 IFPUG의 일반 시스템 특성으로 식별한 후 식(4)을 통해 얻은 값 조정 인자로 나타낼 수 있는 보정계수 가능 범위는 <표 6>과 같다.

현행 대가기준에서는 일반 시스템 특성의 판단 기준이 모호하여 미조정 기능점수를 채택하였다. 그러나 David Consulting Group[7]과 IFPUG의 CPM 4.2에서 현재의 개발 환경에 쉽게 적용할 수 있는 어플리케이션 분류 기준을 제시하였다. 그 기준에 근거하여 각 어플리케이션의 대략적인 값 조정 인자의 가능 범위를 잡으면, 단순한 배치 어플리케이션은 총 영향도가 15 이하, 프론트-엔드 배치 어플리케이션은 15에서 30사이, 대화식 어플리케이션은 30에서 45사이, 온라인 어플리케이션은 30에서 65사이, 웹기반 어플리케이션은 30에서 70사이, 실시간, 원격통신 또는 프로세스 제어 시스템은 30에서 70사이로 예상할 수 있다. <표 7>은 위의 6 가지 어플리케이션 유형 별 총 영향도를 식(4)에 적용하

<표 6> 현행 기준의 보정계수와 VAF 범위

| 어플리케이션 유형 | 보정계수 가능범위 (어플리케이션 유형 × 품질 및 특성보정계수) | VAF 범위 |
|-----------|--|-------------|
| 업무처리용 | 1.00 ~ 1.20 | 0.65 ~ 1.30 |
| 과학기술용 | 1.20 ~ 1.44 | 0.95 ~ 1.30 |
| 멀티미디어용 | 1.30 ~ 1.56 | 0.95 ~ 1.30 |
| 지능정보용 | 1.70 ~ 2.04 | 0.95 ~ 1.30 |
| 시스템용 | 1.70 ~ 2.04 | 0.95 ~ 1.30 |
| 통신제어용 | 1.90 ~ 2.28 | 0.95 ~ 1.30 |
| 공정제어용 | 2.00 ~ 2.40 | 0.95 ~ 1.35 |
| 지휘통제용 | 2.20 ~ 2.64 | 0.95 ~ 1.35 |

〈표 7〉 GSC의 VAF와 보정계수의 범위

| VAF 기준 분류 | VAF 범위 | 현행 대가기준 보정계수 (유형, 품질 및 특성) |
|------------------------|-------------|-------------------------------|
| 배치 어플리케이션 | 0.65 ~ 0.80 | 1.00 ~ 1.44 |
| Front-end 배치 어플리케이션 | 0.80 ~ 0.95 | 1.00 ~ 1.44 |
| 대화식 어플리케이션 | 0.95 ~ 1.10 | 1.00 ~ 2.64 |
| 온라인 어플리케이션 | 0.95 ~ 1.30 | 1.00 ~ 2.64 |
| 웹기반 어플리케이션 | 0.95 ~ 1.30 | 1.00 ~ 2.64 |
| 실시간, 원격통신, 프로세스 제어 시스템 | 0.95 ~ 1.35 | 1.00 ~ 2.64 |

여 계산한 VAF범위와 현행 기준을 이용하여 식별하였을 때 가능한 보정계수의 범위를 비교한 것이다.

〈표 6〉에서 보여지는 바와 같이 현행 대가기준의 어플리케이션 유형만을 기준으로 품질 및 특성 보정계수를 고려하면, 지휘통제용이라고 하더라도 실제로는 매우 단순한 프로그램임에도 보정계수의 범위는 매우 높은 값들로 결정되고, 반대로 업무처리용이나 과학기술용이라고 하더라도 어플리케이션의 특성이 매우 복잡한 경우임에도 보정계수가 더 이상 높아질 수 없는 문제점이 있다. 그러나 IFPUG의 일반 시스템 특성의 관점에서 보정계수를 정하면 실제적으로 어플리케이션이 가지는 다양한 속성에 따라 적절한 보정계수를 부여할 수 있다.

〈표 7〉을 통해 IFPUG의 일반 시스템 특성을 이용한 값 조정 인자의 범위와 현행 어플리케이션 유형 보정계수와 품질 및 특성 보정계수의 범위를 비교한 결과 값 조정 인자의 범위는 최대 1.35로 +35%의 보정 효과가 있으나, 현행 기준의 상기 보정계수의 범위는 최대값이 2.64로 +164%의 보정 효과가 있다.

따라서 일반 시스템 특성을 개선안에 적용하기 위해서는 값 조정 인자를 그대로 보정계수에 적용하는 것이 아니라, 값 조정 인자 범위와 현행 기준의 보정계수 범위가 큰 차이를 보이는 어플리케이션 유형의 경우에만 값 조정 인자를 조정할 필요가 있다. 이들 유형에는 지능정보용 및 시스템용, 통신제어용 및 공정제어용, 지휘통제용등이 해당된다.

따라서 IFPUG의 일반 시스템 특성을 이용한 값 조정 인자를 보정계수로 적용하되 보정계수 범위에 큰 차이를 보이는 상기 어플리케이션 유형에는 추가 조정인자를 적용하는 것이 적합할 것이다. 값 조정 인자에 추가 조정 인자를 적용하여 조정된 보정계수는 아래의 식(5)와 같이 계산한다.

$$\bullet \text{ 보정계수} = \text{VAF} + \text{추가 조정인자} = (\text{TDI} \times 0.01) + 0.65 + \text{추가 조정인자} \quad (5)$$

단, 추가 조정 인자는 지능정보용및시스템용: 0.75
통신제어용및공정제어용:1
지휘통제용: 1.3
그 외의 유형은 0

〈표 8〉 보정계수 변환표

| 어플리케이션 유형 | 현행 기준 | 개선(案) |
|---------------|-------------|-------------|
| 지능정보용 및 시스템용 | 1.70 ~ 2.04 | 1.40 ~ 2.10 |
| 통신제어용 및 공정제어용 | 1.90 ~ 2.40 | 1.65 ~ 2.35 |
| 지휘통제용 | 2.20 ~ 2.64 | 1.95 ~ 2.65 |
| 기타 | 1.00 ~ 1.56 | 0.65 ~ 1.35 |

각 유형별 추가 조정인자를 적용하여 아래와 같이 계산한다.

$$\textcircled{1} \text{ 지능정보용 및 시스템용: } (\text{TDI} \times 0.01) + 0.65 + 0.75 = (\text{TDI} \times 0.01) + 1.4$$

$$\textcircled{2} \text{ 통신제어용 및 공정제어용: } (\text{TDI} \times 0.01) + 0.65 + 1 = (\text{TDI} \times 0.01) + 1.65$$

$$\textcircled{3} \text{ 지휘통제용: } (\text{TDI} \times 0.01) + 0.65 + 1.3 = (\text{TDI} \times 0.01) + 1.95$$

$$\textcircled{4} \text{ 현행 대가기준의 그 외의 유형: } (\text{TDI} \times 0.01) + 0.65$$

위 식(5)에 따른 보정계수의 허용범위를 어플리케이션 유형에 따라 현행 기준과 비교하면 〈표 8〉과 같다.

대체적으로 현행 기준의 보정계수 범위와 일치하면서 하한의 범위가 확대되었다. 개선 방안은 현행 기준에 문제 시되었던 어플리케이션 유형에 의해서만 일방적으로 과대추정되거나 과소추정 되는 것을 막을 수 있고, 어플리케이션이 가지는 다양한 특성을 반영할 수 있다.

3.2 납기보정계수

소프트웨어 개발 시 개발비용뿐만 아니라 주어진 기간 내에 프로젝트를 완료하는 것은 매우 중요하다. 이러한 이유로 일반적인 소프트웨어 비용산정 모델에는 적정 납기 산정 과정과 납기 보정 계수를 제공하고 있다. 그러나 현행 소프트웨어 사업 대가기준에서는 적정 납기 산정 및 납기 보정 계수를 제공하지 않으므로 인해 실무자들은 이에 대한 필요성을 제기하고 있다[3].

3.2.1 적정납기

(1) 공수로부터 추정

COCOMO나 SLIM 모델과 같이 공수로부터 개발기간을 추정하는 방안이다. 그러나 현행 사업대가 기준은 공수를 추정하는 모델이 아니라 소프트웨어 규모를 이용하여 개발비를 직접 추정하는 모델이므로 이 방안을 적용할 수는 없다.

(2) 생산성 수준을 이용

평균 투입 인력이 제시되는 경우, 규모(FP) → 공수 (E) → 평균 투입 인력(E/D) → 개발기간(D)의 순서대로 구하는 방안이다. 그러나 이 방안을 활용하기 위해서는 평균 투입

인력이 제시되어야 하므로 사업대가 기준에 적용할 수 없는 방안이다. 또한 이 방안은 상이한 생산성 수준의 조직 간에 적용하기에도 곤란하므로, 특정 수주자를 전제로 하지 않으며 다양한 수주자들을 대상으로 하는 사업대가 기준에 적합하지 않은 방안이다. 뿐만 아니라 이 방안은 상기 추정 단계별로 오차가 누적되어 편향된 결과를 유발할 가능성이 높으므로 실제 적용하기에 무리가 있다[8].

(3) 규모를 통한 추정

ISBSG[9]의 개발기간 추정 모델과 같이 규모를 통해 직접 적정 납기를 추정하는 방식으로 가장 단순하고, 누적 오차가 적어 실무 적용이 용이한 현실성 있는 방식이다. 따라서 본 논문에서는 이 방안을 채택하기로 한다.

3.2.2 납기보정 절차

(1) 적정납기 산정(案)

수집된 비용자료를의 프로젝트 규모와 개발기간 간의 관계를 이용하여 적정납기 산정을 아래의 식(6)과 같이 계산한다.

$$\text{적정납기} = 0.71 \times \text{규모(FP)}^{0.33} \quad (6)$$

(2) 납기 보정계수 적용

적정납기를 구하는 함수를 이용하여 프로젝트의 적정납기를 구한 다음, 이를 고객이 요구한 납기와 비교하여 아래의 식(7)과 같이 납기 단축율을 계산하고 다시 이 납기 단축율을 기초로 납기 보정계수를 적용한다. 즉, 프로젝트 규모를 기준으로 산정한 납기를 기준으로 고객이 요구하는 납품일자를 비교하여, 납품요구 기간이 적정납기보다 짧을 때 단축비율을 기준으로 <표 9>의 납기보정계수(1案)[10]와, <표 10>의 납기보정계수(2案)[11]을 납기 보정계수로 적용한다.

$$\text{단축율} = (\text{적정납기} - \text{요구납기}) / \text{적정납기} \quad (7)$$

<표 9> 납기 보정계수(1案)

| 납기 단축율 | 납기 보정계수 |
|--------|---------|
| 0%이하 | 1.00 |
| 0% | 1.00 |
| 5% | 1.00 |
| 10% | 1.03 |
| 15% | 1.05 |
| 20% | 1.07 |
| 25% | 1.08 |
| 30% | 1.10 |
| 30%이상 | 1.10 |

<표 10> 납기 보정계수(2案)

| 납기 단축율 | 납기 보정계수 |
|--------|---------|
| 0% 이하 | 1.00 |
| 15% | 1.14 |
| 25% | 1.43 |

4. 보정치 조정 후 모델의 평가

4.1 평가 척도

(1) 다중결정계수: R^2

$$R^2 = 1 - \frac{\sum(E_i - \hat{E}_i)^2}{\sum(E_i - \bar{E})^2} \quad (8)$$

다중결정계수는 실제 개발비용 E 와 개발비용 추정 \hat{E} 가 어느 정도 선형적 관계에 있는가를 나타내는 것으로 실제 개발비용 E 와 개발비의 추정치 \hat{E} 가 절대적 의미에서 어느 정도 상응하는지는 설명하지 못하는 문제점이 있다.

(2) 상대 오차 (mean) relative error: $(\overline{RE}), RE$

$$RE = \frac{E - \hat{E}}{E}, \overline{RE} = \frac{1}{n} \sum_{i=1}^n RE_i \quad (9)$$

상대 오차의 문제점은 상대 오차가 큰 양수는 상대 오차가 큰 음수에 의해 상쇄되므로 상대 오차가 작은 것이 좋은 모델임을 나타내지 못 한다. 따라서 비용산정 분야의 실제 연구에 유용한 척도가 되지 못한다.

(3) 상대 오차의 절대값 평균

(Mean) Magnitude of RE: $(MMRE = \overline{MRE})$

$$MRE = |RE|, MMRE = \overline{MRE} = \frac{1}{n} \sum_{i=1}^n MRE_i \quad (10)$$

상대오차의 절대값인 MRE 가 작을수록 모델의 정확도는 높아진다. 반대로 MRE 가 클수록 모델의 정확도는 낮아진다.

$MMRE$ 가 대부분의 비용산정 연구에서 비용산정 모델의 정확도를 평가하는 기준으로 사용되는 이유로는 상대 오차가 큰 양수는 상대 오차가 큰 음수에 의해 상쇄되지 못한다는 장점을 지니고 있기 때문이다.

(4) prediction at level 1: $PRED(l)$

$$PRED(l) = \frac{k}{n} \quad (11)$$

위 식(11)에서 n 은 전체 데이터의 개수이며 k 는 $MRE \leq l$ 에 포함되는 데이터의 개수이다. 예를 들어 $PRED(0.25) = 0.83$ 의 의미는 예측값이 오차 범위 25%안에 드는 데이터의 개수가 전체의 83%에 해당된다는 것이다. $PRED$ 는 일반적으로 예측 모델의 정확도를 평가하기에 적절한 척도이다[12].

4.2 조정된 값 조정 인자(VAF)를 적용한 보정계수

비용산정의 정확도를 측정하기 위하여 실제로 수집된 데이터를 대상[13]으로 '현행 기준의 규모보정', '규모 및 조정된 VAF를 적용한 보정', '규모 및 언어보정', '규모 및 언어보정과 조정된 VAF를 적용한 보정', '조정된 VAF만을 적용

〈표 11〉 현행 기준과 조정된 값 조정 인자(VAF)를 적용한 개선안의 비교

| 보정계수 \ 평가 척도 | MMRE | PRED(0.25) |
|----------------------------|------|------------|
| 현행 기준의 규모 보정 | 0.40 | 0.35 |
| 규모, 조정된 VAF를 적용한 보정 | 0.36 | 0.42 |
| 규모, 언어 보정 | 0.39 | 0.39 |
| 규모, 언어 보정과 조정된 VAF를 적용한 보정 | 0.37 | 0.35 |
| 조정된 VAF만을 적용한 보정 | 0.37 | 0.35 |

한 보정'등 5가지 경우를 들어 MMRE와 PRED를 이용해 비교하여 보았다. 비교 결과 <표 11>에서 알 수 있듯이 5 가지 경우 중 '규모 및 조정된 VAF를 적용한 보정'을 적용한 경우가 MMRE 측면에서는 0.36으로 가장 낮고, PRED 측면에서는 0.42로 가장 높게 나타났다. 즉, 규모 보정계수와 함께 조정된 값 조정 인자를 반영한 보정계수를 적용했을 때 모델의 정확도가 향상되었음을 확인할 수 있었다.

4.3 납기 보정계수

<표 9>와 <표 10>의 두 가지 납기보정계수를 적용하여 모델의 정확도를 분석하였다. 분석결과는 <표 12>에서 알 수 있듯이 MMRE 측면과, PRED 측면 모두에서 1안이 더욱 정확한 것으로 나타났다.

〈표 12〉 납기 보정계수의 정확성 비교

| 구분 \ 평가 척도 | MMRE | PRED(0.25) |
|------------|------|------------|
| 1 案 | 0.39 | 0.37 |
| 2 案 | 0.49 | 0.29 |

5. 결론 및 향후 연구 과제

소프트웨어 사업대가기준은 소프트웨어 기술 발전과 여러 가지 환경 변화 요인을 반영하여 소프트웨어 개발비용 산정의 정확도를 향상시키기 위해 지속적으로 연구되어야 할 분야이다. 이에 따라, 본 논문에서는 현행 대가 기준에 14가지 일반시스템특성을 반영하는 보정계수의 개선 및 납기 보정계수의 도입 방안을 제시하였다.

현행 대가기준의 어플리케이션 유형 보정계수와 품질 및 특성 보정계수를 대치할 수 있는 일반 시스템 특성 기반의 보정계수를 적용하였으며, 또한 납기 보정계수를 추가하였다. 개선된 모델의 정확도를 평가하기 위해 MMRE와 PRED를 이용하여 분석하였다.

결론적으로 규모 보정계수와 함께 조정된 값 조정 인자를 적용한 보정계수에서 MMRE 측면에서는 가장 낮게, PRED 측면에서는 가장 높게 나타났다. 즉, 비용산정의 정확도가 현행 대가기준에 비해 향상되었음을 확인할 수 있었다. 또한 적정 납기 보정계수 적용 방안에서는 1案인 <표 9>의 보정계수 값을 적용하는 것이 적합한 것으로 평가되었다.

향후 연구 과제로는 일관성 있고 정확한 자료수집 양식을 이용하여 좀더 세부적이고 정확한 많은 데이터를 수집하여 분석할 것이다. 또한 비용 산정 시 소프트웨어 품질 척도를 추가하는 방안 및 품질 척도를 적용한 비용 산정 모델을 개발하고자 한다.

참 고 문 헌

- [1] Norman E. Fenton and Shari Lawrence Pfleeger, "Software Metrics," 2nd Ed., PWS Publishing Company, 1997.
- [2] 한국소프트웨어산업협회, "2004 소프트웨어 사업대가기준 해설", 2004. 3.
- [3] 채효근 외, "소프트웨어 개발비 대가기준 개선 연구", 한국전산원 최종보고서, 2004. 11.
- [4] 이상운, 강정호, 박중양, "기능점수를 이용한 소프트웨어 개발 노력 추정", 정보처리학회논문지 D, 제9-D권 제4호, pp.603-612, 2002. 8.
- [5] Maria I. Snaches-Segura et al., "Virtual Reality Systems Estimation vs. Traditional Systems Estimation," Journal of Systems and Software, 72, pp.187-194, 2004. 권 제4호, pp. 603-612, 2002. 8.
- [6] 권기태, 신수정, 박찬규, "기능점수와 소프트웨어 측정", 도서출판 그린, 2003. 8.
- [7] David Garmus and David Herron, "Function Point Analysis: Measurement Practices for Successful Software Projects," Addison-Wesley, 2001.
- [8] 이상운, "소프트웨어 개발기간 추정 모델", 정보과학회논문지: 소프트웨어 및 응용, 제31권 제1호, pp.20-28, 2004. 1.
- [9] Peter R. Hill, "Practical Project Estimation", ISBSG, 2001.
- [10] 김동익 외, "소프트웨어 사업대가기준 개선사업", 한국소프트웨어산업협회 최종보고서, 2001. 12.
- [11] Barry W. Boehm et al., "Software Cost Estimation with COCOMO II," Prentice-Hall, 2002.
- [12] S. D. Conte et al., "Software Engineering Metrics and Models," The Benjamin/Cummings Publishing Company, Inc. 1985.
- [13] 김동익 외, "정보화사업 비용자료 수집모형 개선연구 및 비용자료 수집", 한국전산원 최종보고서, 2004. 11.



권기태

e-mail : ktkwon@kangnung.ac.kr

1986년 서울대학교 계산통계학과 학사

1988년 서울대학교 계산통계학과 석사

1993년 서울대학교 계산통계학과 박사

1996년 Univ. of Southern California,

Post-Doc.

1990년9월~현재 강릉대학교 컴퓨터공학과 교수

관심분야: 소프트웨어 비용산정, 소프트웨어 매트릭스,
소프트웨어 아키텍처



변분희

e-mail : bbh@kangnung.ac.kr

1996년 강릉대학교 컴퓨터공학과 학사

2000년 강릉대학교 컴퓨터교육 석사

2005년 강릉대학교 컴퓨터공학과

박사수료

관심분야: 소프트웨어 비용산정, 소프트
웨어 아키텍처