

## AHP를 이용한 소프트웨어 개발비 보정계수 산정

김우제<sup>1†</sup> · 박찬규<sup>2</sup> · 신수정<sup>3</sup>

<sup>1</sup>서울산업대학교 산업정보시스템공학과 / <sup>2</sup>동국대학교 경영학과 / <sup>3</sup>한국전산원 전자정부지원팀

### Estimating Cost Adjustment Factors of Software Development Projects using Analytic Hierarchy Process

Woo-Je Kim<sup>1</sup> · Chan-Kyoo Park<sup>2</sup> · Soo Jeong Shin<sup>3</sup>

<sup>1</sup>Department of Industrial and Information Systems Engineering, Seoul National University of Technology, Seoul, 139-743

<sup>2</sup>Department of Management, Dongguk University, Seoul, 100-715

<sup>3</sup>e-Government Project Team, National Computerization Agency, Seoul, 100-775

The purpose of this paper is to reorganize cost adjustment factors of software development projects in estimating software development cost, and derive adjustment coefficients of application types and language types by analytical hierarchy process. We constructed a decision-making hierarchy of various criteria which determine the complexity of application types and language types, and conducted a survey on the pairwise comparison among alternatives. Finally, the cost adjustment coefficients of application types and language types were derived by analytic hierarchy process. This paper is the first study in which the analytic hierarchy process was applied to the field of estimating cost adjustment factors of software development projects.

**Keywords:** software development cost, adjustment coefficient, application types, language types, analytic hierarchy process

#### 1. 서론

정보시스템이 모든 업무의 필수적인 수단으로 정착함에 따라 정보시스템을 구축·유지보수·운영하는 데 소요되는 비용이 지속적으로 증가하고 있다. 2001년 기준으로 우리나라의 정보화지출 총액은 약 48 조원으로 GDP의 8.9%를 차지하고 있다(NCA, 2002). 특히, 정보화지출 중 소프트웨어 부문이 차지하는 비중이 매년 증가하여 2001년에는 전체 정보화지출 중 24%를 차지하여 하드웨어 부문보다 그 비중이 높게 나타나고 있으며, 국내 소프트웨어 산업 시장규모도 연간 100억 달러 이상으로 추정될 정도로 급속히 성장하고 있다(NCA, 2002). 이처럼 국가예산에서 정보화 관련 비용이 차지하는 비중이 높아짐에 따라, 정보화예산의 효과적 관리를 위해 정보화사업의 소

요비용을 정확히 예측하는 것이 중요한 문제로 대두되고 있다.

정보시스템 구축사업의 경우 전체 사업비용은 크게 소프트웨어 개발비와 하드웨어/소프트웨어 구입비로 구분할 수 있다. 하드웨어/소프트웨어 구입비는 시장가격 조사를 통해 쉽게 예측할 수 있으므로 본 연구에서는 논외로 한다. 소프트웨어 개발비는 요구사항의 불확실성과 모호성, 소프트웨어 생산성 측정의 어려움, 과거 데이터의 부족, 신기술에 의한 위험 및 개발 환경 의존성 등으로 인해 예측이 쉽지 않다(Vidger, 1994). 미국의 경우 정보화사업의 50% 이상이 당초 예산보다 1.8배 이상 소요되는 것으로 나타났으며, 16% 정도의 사업만이 당초 납기와 비용 내에 완료된 것으로 조사되었다(The Standish Group, 1995). 또한, Gartner 보고서에 제시된 정보화 사업비 과소평가의 법칙(law of IS cost underestimating)에 의하면 사업의 범위나

†연락처 : 김우제 교수, 139-743 서울시 노원구 공릉2동 서울산업대학교 산업정보시스템공학과, Fax : 02-974-2849, E-mail : wjkim@snut.ac.kr

목표 등이 불확실한 경우에는 정보화사업에 투입되는 실제비용은 예측치의 2배 이상으로 증가할 수 있다(Gomolski, 2002).

소프트웨어 개발비 예측기법에는 크게 모델기반(Model-based) 방법, 전문가예측방법, 학습(Learning-oriented)방법, 동적기반(Dynamics-based) 방법, 회귀분석방법, 베이지안(Bayesian) 기법 등이 있다. 이중 가장 널리 사용되는 방법은 모델기반 방법으로서 1965년 SDC(System Development Corporation) 모델이 개발된 이후 COCOMO II(Boehm, 2000), ISBSG 모형(ISBSG, 2001), ESTIMACS(Jones, 1998), PRICE-S(Price Systems), SEER-SEM(Golorath Inc.), KnowledgePLAN(Software Productivity Research) 등 최근까지 수십여 개의 공개 또는 상용 소프트웨어 비용산정 모형들이 개발되었다. 모델기반 방법을 통해 소프트웨어 개발비를 예측하기 위해서는 먼저 개발하고자 하는 소프트웨어의 규모를 측정하고, 소프트웨어 규모와 사업의 특성 등을 고려하여 소프트웨어 개발비를 추정한다.

소프트웨어 규모를 측정하는 방법으로 가장 널리 사용되는 방법은 프로그램 라인수(Line of code) 방식과 기능점수방식(Function point model)이 있다(Pressman, 1997; Putnam, 1992). 우리나라에서는 소프트웨어 개발 및 유지보수사업의 규모산정에 주로 라인수 방식을 사용해 왔는데, 라인수 방식은 개발자 중심으로 사업 초기에는 라인수 예측이 어렵고 개발환경에 영향을 받는 등 여러 가지 단점을 갖고 있다(Furey, 1997; Kemerer, 1992; Matson, 1994). 반면, Albrecht가 제안한 기능점수 방식은 소프트웨어가 제공하는 기능의 수를 사용자 관점에서 측정하여 소프트웨어 규모를 산정하는 방법(Albrecht, 1979)으로, 사용자가 이해하기 쉽고 개발환경에 독립적이라는 장점을 갖고 있다(Furey, 1997; Matson, 1994). 기능점수를 이용한 비용산정모형으로는 COCOMO II(Boehm, 2000), SPQR/20(Jones, 1998), ISBSG (ISBSG, 2001) 등이 있다.

소프트웨어 개발비 예측의 정확도를 높이기 위해서는 규모 외에 소프트웨어, 프로젝트, 개발환경 및 투입인력 등의 특성들을 반영하는 과정이 필요한데, 소프트웨어 개발비용에 영향을 미치는 특성들을 흔히 보정요소라 부르며 각 보정요소의 가중치를 보정계수라 부른다. 보정요소와 보정계수는 사용하는 비용예측모형에 따라 달라진다. 예를 들어, COCOMO II 모형(Boehm, 2000)은 소프트웨어 신뢰성, DB 크기, 복잡도(Product complexity), 문서화(Documentation) 수준 등을 비롯한 18개의 소프트웨어, 플랫폼(Platform), 인력(Personnel) 및 프로젝트의 특성을 반영하여 비용을 예측한다. 또한, 간단한 ISBSG 모형(ISBSG, 2001)은 하드웨어 플랫폼, 개발언어 등을 고려하여 소프트웨어 개발비용을 예측하고 있다.

다양한 비용예측모형들이 개발·활용되고 있는 해외와는 달리 국내에서는 국가에서 제정·고시한 소프트웨어 사업대가기준을 통일적으로 사용해 왔다. 소프트웨어 사업대가기준은 1989년 제정된 이래 공공기관에 발주하는 소프트웨어 사업의 원가와 대가를 산정하는 기준으로 활용되어 왔으며, 소프트웨어 사업에 관한 전문지식 없이도 소프트웨어 사업의 예산

을 산정할 수 있는 통일적 기준을 제시함으로써 소프트웨어 사업의 발주와 관리를 용이하게 하는 데 기여하였다. 그러나 소프트웨어 사업대가기준은 라인수 방식의 규모산정, 각종 보정계수의 비현실성 등 변화하는 소프트웨어 사업의 현실을 제대로 반영하지 못하고, 일부 사업에서는 기준에 의해 산정된 결과와 실제 소요비용과의 오차도 큰 것으로 조사되고 있다. 따라서 기능점수방식의 도입, 각종 보정계수의 현실화 등을 통해 소프트웨어 사업대가기준의 정확성을 향상시킬 필요가 있다.

본 연구는 소프트웨어 사업대가기준의 개선작업의 일환으로 소프트웨어 사업대가기준의 보정요소를 현실에 맞게 재분류하고 AHP(Analytic hierarchy process) 방법을 도입하여 보다 정확하고 객관적인 보정계수를 제시하는 데 그 목적이 있다. 이를 위해 현행 소프트웨어 사업대가기준에서 사용하고 있는 5가지 보정요소 중 AHP 방법 적용이 가능한 두 개의 보정요소(애플리케이션 유형, 개발언어)를 선정하였다. 선정된 보정요소를 국내 소프트웨어 사업 현실에 맞게 재분류하거나 또는 수정하고, AHP 분석이 가능한 데이터를 수집할 수 있도록 설문을 작성하였다. 소프트웨어 사업 경험이 풍부한 발주자, 관리자, 프로그래머 등 145명의 설문지응답을 기초로 보정계수 분석결과를 제시하였다. 본 연구결과는 소프트웨어 사업대가기준의 개정을 위한 기초자료로 활용될 수 있으며, 또한 공공부문 정보화사업을 기획·관리하거나 직접 수행하는 사람이 정보화사업의 비용을 예측하고 그 결과를 해석하는 데 도움을 줄 수 있을 것이다.

본 논문의 구성을 살펴보면, 2장에서 국내외의 소프트웨어 비용산정모형과 소프트웨어 사업대가기준을 소개하고, 3장에서는 AHP 방법에 대해 간략히 정리한다. 4장에서는 AHP 분석이 가능한 보정요소를 선정하고, 5장에서는 AHP 분석결과를 토대로 보정계수를 제시한다. 마지막으로, 6장에서는 본 연구의 결론을 제시한다.

## 2. 소프트웨어 비용예측모형 및 소프트웨어 사업대가기준

1965년 미국 SDC(Systems Development Corporation)사에서 169개의 소프트웨어 프로젝트를 분석하여 104개의 비용보정요소 중 통계적으로 의미가 있는 14개의 보정요소를 갖는 소프트웨어 비용예측모형을 개발한 후 현재까지 많은 비용예측모형이 개발·개선되어 왔다. 그 중에서 가장 널리 사용되고 있는 공개모형인 COCOMO/COCOMO II 모형(Boehm, 2000), IFPUG 모형(IFPUG, 2000), ISBSG 모형(ISBSG, 2001)과 상용모형인 PRICE-S(Price Systems), KnowledgePLAN(Software Productivity Research)에서 비용예측에 사용되는 보정요소를 살펴본다.

COCOMO(CONstructive COSt MOdel)는 1981년 Boehm에 의해 개발된 모델로 소프트웨어 비용예측에 가장 널리 통용되어 왔으며 다른 비용예측모형 개발에도 많은 영향을 주었다. COCO-

MO는 모형의 상세수준(Level)의 따라 소프트웨어 규모만을 이용하여 비용을 예측하는 기본 COCOMO와 여기에 15개의 비용요소를 첨가한 중간 COCOMO, 소프트웨어 프로젝트 생명주기 각 단계별로 비용요소를 구분한 상세 COCOMO 등으로 나누어지며, 프로젝트 개발유형에 따라 모델을 다시 3가지로 세분하고 있다. COCOMO를 통해 전체 소프트웨어 개발비용을 예측할 수 있을 뿐만 아니라 수명주기 각 단계별로 투입되는 비용도 산출할 수 있다. COCOMO의 비용예측공식은 다음과 같다. PM은 투입노력(Person-month)을 나타내고, S는 소프트웨어 규모이며, a, b는 수준(Level)과 개발유형에 따라 정해지는 상수이다.

$$PM = a \times S^b \times \prod_{i=1}^{15} f_i$$

또한,  $f_i$ 는 노력승수(Effort multiplier)라고 하는데, 프로젝트의 특성과 여러 가지 보정요소를 고려하여 결정되는 값이다. COCOMO에서 사용하는 보정요소에는 15가지가 있으며 각 보정요소의 복잡도를 매우 낮음(Very low), 낮음(Low), 보통(Nominal), 높음(High), 매우 높음(Very high), 극히 높음(Extra high) 등 6단계로 구분하여 가중치를 주고 있다. 예를 들어, 요구되는 소프트웨어 신뢰성이 매우 낮은 경우 0.75에서부터 매우 높은 경우 1.40까지의 값을 갖고 있다. COCOMO에서 고려하는 보정요

소와 보정계수를 정리하면 <표 1>과 같다.

COCOMO II는 변화된 소프트웨어 개발환경과 기법에 맞도록 COCOMO를 개선한 모형으로 소프트웨어 규모로 라인수뿐만 아니라 기능점수방식 및 객체점수방식을 사용할 수도 있도록 수정되었다. 또한, 현실에 맞도록 보정요소들을 수정하고, 비용예측시기에 따라 초기설계(Early design)모형과 아키텍처설계 후(Post-architecture)모형으로 구분하였다. COCOMO II의 개발비용(노력) 예측식은 다음과 같다.

$$PM = A \times S^E \times \prod_{i=1}^n EM_i,$$

$$\text{단, } E = B + 0.01 \times \sum_{j=1}^5 SF_j$$

노력승수  $EM_i$ 는 최대 16개가 있으며, A와 B는 상수로 각각의 데이터 특성에 따라 달라질 수 있으나, Boehm이 수집한 데이터의 분석에 의하면  $A=2.94, B=0.91$ 로 설정되어 진다. 스케일 인자(Scale factor)  $SF_j$ 는 5개가 있으며, 유사 프로젝트 수행여부, 개발유연성, 아키텍처 위험도, 팀 응집도, 프로세스 성숙도 등이 있다. COCOMO II의 노력승수는 COCOMO 모형에서 반응시간, 최신 프로그래밍 기술사용이 보정요소에서 제거되고 대신 문서화, 인력연속성, 사이트(Site) 수가 추가되었다.

다른 모형들도 COCOMO와 유사하게 소프트웨어 규모와 특

표 1. COCOMO 모형의 보정요소

보정요소	보정계수						보정요소	보정계수					
	매우 낮음	낮음	보통	높음	매우 높음	극히 높음		매우 낮음	낮음	보통	높음	매우 높음	극히 높음
제품속성							인력속성						
소프트웨어 신뢰성	0.75	0.88	1.00	1.15	1.40		분석능력	1.46	1.19	1.00	0.86	0.71	
DB 크기		0.94	1.00	1.08	1.16		애플리케이션 경험	1.29	1.13	1.00	0.91	0.82	
제품복잡도	0.70	0.85	1.00	1.15	1.30	1.65	프로그래머 능력	1.42	1.17	1.00	0.86	0.70	
컴퓨터 속성							가상기계 경험	1.21	1.10	1.00	0.90		
실행시간 제약			1.00	1.11	1.30	1.66	개발언어 경험	1.14	1.07	1.00	0.95		
주기억장치 제약			1.00	1.06	1.21	1.56	프로젝트 속성						
가상기계 가변성		0.87	1.00	1.15	1.30		최신 프로그래밍 기술사용	1.24	1.10	1.00	0.91	0.82	
반응시간		0.87	1.00	1.07	1.15		소프트웨어 도구사용	1.24	1.10	1.00	0.91	0.83	
							개발일정	1.23	1.08	1.00	1.04	1.10	

표 2. 여러 가지 모형의 보정요소

모형	개발자	주요 보정요소
IFPUG	국제기능점수 사용자그룹	14개의 시스템 일반특성(데이터 통신, 분산처리, 성능, 하드웨어 제약, 처리율, 온라인 데이터 입력, 최종사용자 용이성, 온라인 갱신, 처리복잡도, 재사용성, 설치용이성, 운영용이성, 복수 사이트, 변경용이성)
ISBSG	국제S/W 벤치마킹 표준그룹	플랫폼(메인프레임, 중간, PC), 개발언어(3GL, 4GL, ApG)
PRICE-S	PRICE Systems	응용 도메인(상용, 군용) 통합수준, 개발언어, 생산성요소, 복잡도
Knowledge PLAN	Caper Jones	플랫폼, 응용 도메인, 복잡도, 개발언어, 인력구성, 사용기술, 소프트웨어 도구, 개발환경, 보안/성능 요구수준, 문서화수준, 테스트 복잡도

성들을 고려하고 소프트웨어 비용을 예측하고 있다. 공개된 모형인 IFPUG 모형, ISBSG 모형과 상용도구로 구현된 PRICE-S, KnowledgePLAN에서 고려하는 보정요소를 정리하면 <표 2>와 같다. <표 2>에서 IFPUG 모형은 기능점수를 측정하는 모형으로 14개의 시스템 일반특성을 평가하여 소프트웨어의 기능점수 크기를 보정한다.

한편, 소프트웨어 비용예측에 관한 국내연구는 데이터 수집 및 제도적 제약으로 인해 활발하지 못한 실정이다. 김현수(Kim, 1997)는 클라이언트/서버 환경, 대형 소프트웨어 프로젝트, 사무처리용 및 알고리즘 중심적인 소프트웨어 개발 등에 적합한 수정된 기능점수모형을 제시하였고, 이양규(Lee, 1997)는 34개의 공공부문 프로젝트를 조사하여 기능점수와 투입인력 간의 회귀식을 제시하였다. 또한, 박찬규 등(Park, 2002)은 사업초기단계에 소요인력예측을 위한 간이기능점수 모형을 제시하고 간이기능점수와 투입인력 간의 회귀식을 제시하였다. 그러나 이러한 연구에도 불구하고 공공기관에서 발주되는 대부분의 프로젝트는 소프트웨어 사업대가기준에 의해 예산과 대가가 결정되어 왔다.

소프트웨어 사업대가기준은 공공기관이 국내에서 발주하는 소프트웨어 개발, 유지보수, 데이터 구축 등 소프트웨어 사업의 비용을 산정하기 위한 기준으로 1989년 제정된 이래로 공공부문뿐만 아니라 민간부문 소프트웨어 개발사업에도 적용되어 왔다. 소프트웨어 사업대가기준에 명시된 소프트웨어 개발비 계산절차를 정리하면 <그림 1>과 같다.

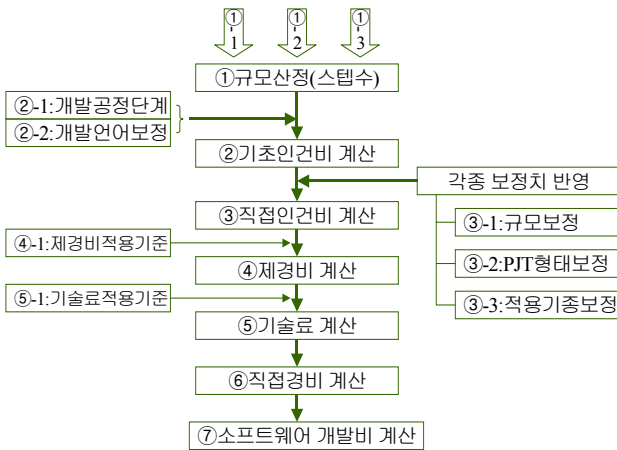


그림 1. 소프트웨어 개발비 사업대가기준 적용 절차 (KSIA, 2001).

소프트웨어 사업대가기준을 적용하기 위해선 먼저 소프트웨어의 규모를 산정해야 한다. 현재는 스텝수(라인수) 방식으로 소프트웨어 규모를 산정하지만, 이를 기능점수방식으로 개정하는 작업이 진행 중이다. 소프트웨어 규모가 결정되면 개발언어와 수행공정에 따라 기초인건비가 계산되고, 여기에 규모, 프로젝트형태, 적용기준에 따른 보정을 통해 직접인건비가 계산된다. 직접인건비에 일정비율을 곱하여 제경비(110%

또는 120%) 및 기술료(20%)를 계산하며, 직접경비는 프로젝트에 직접 사용되는 경비들을 말한다. 직접인건비, 제경비, 기술료, 직접경비를 모두 합하여 소프트웨어 개발비가 계산된다. 소프트웨어 사업대가기준에서는 직접인건비가 전체 소프트웨어 개발비에 결정적인 영향을 미치는데, 직접인건비를 구하는 식을 정리해 보면 다음과 같다.

$$\text{직접인건비} = a \times S \times \prod_{i=1}^5 f_i$$

S는 소프트웨어 규모, a는 단가를 나타내며, f<sub>i</sub>는 보정계수를 뜻하는데, 규모, 애플리케이션 유형, 개발언어, 하드웨어 등 4가지 보정요소가 있다. 규모보정계수는 f<sub>1</sub>=0.108 × log(라인수)+0.2229로 계산되며, 애플리케이션 유형 보정계수는 1.0(사무처리용)에서 4.0(지휘통제용)까지의 범위를 갖는다. 개발언어 보정계수는 0.65(4세대언어)에서 2.0(어셈블리어)까지의 값을 가지며, 하드웨어 보정계수는 0.8(PC)에서 1.1(클라이언트/서버)까지의 범위를 갖는다. 그밖에 납기가 단축될 경우 제경비율을 3%~10%까지 증가시킬 수 있고, 개발방법론과 품질보증기준 적용여부에 따라 기술료가 20%까지 증가될 수 있다. 각 보정요소의 상세한 분류와 보정계수는 4장에서 논의한다.

### 3. AHP 방법

AHP 방법은 1970년대 초반 T. Saaty에 의하여 개발된 의사결정 방법으로 의사결정의 계층구조를 구성하고 있는 요소 간의 쌍대비교(Pairwise comparison)에 의한 판단을 통하여 평가자의 지식, 경험 및 직관을 포착하고자 하는 하나의 의사결정방법론이다. AHP는 이론의 단순성 및 명확성, 적용의 간편성 및 범용성이라는 특징으로 여러 의사결정분야에서 널리 응용되어 왔으며, 이론구조 자체에 관해서도 활발한 연구가 진행되고 있다. 일반적으로 의사결정문제는 서로 상반된 기준과 불완전한 정보 및 제한된 자원하에서 최적의 대안을 선택하는 문제인데, AHP는 이러한 다수 기준하에서 평가되는 다수 대안들의 우선순위를 선정하는 문제를 해결할 수 있다.

AHP에서의 의사결정 분석단계는 다음과 같이 4단계로 구성된다(Satty, 1990).

#### 단계 1: 의사결정계층(Decision hierarchy) 구축

의사결정문제를 상호 관련된 의사결정사항들의 계층으로 분류하여 의사결정계층을 설정한다. 계층은 목적(Goal), 기준(Criteria), 대안(Alternatives)으로 구성된다. 계층의 최상위층에는 가장 포괄적인 의사결정의 목적이 놓여지며, 그 다음의 계층들은 의사결정목적에 영향을 미치는 다양한 기준(Criteria)들로 구성된다. 이들 기준들은 낮은 계층에 있는 것일수록 구체적인 것이 된다. 계층의 최하층은 선택의 대상이 되는 여러 의사결정 대안들로 구성된다.

**단계 2: 쌍대비교 자료수집**

이 단계에서는 상위계층에 있는 요소를 기준으로 하위계층에 있는 각 요소의 가중치를 측정하고, 상위계층의 요소하에서 각 하위요소가 다른 하위요소에 비하여 상대적으로 우수한 정도를 측정하여 상위계층의 요소하에서의 하위요소들 간의 쌍대비교행렬(Pairwise comparison matrix)을 작성한다.

**단계 3: 가중치 산정**

쌍대비교행렬로부터 고유치방법(Eigenvalue method)을 이용하여 계층의 각 레벨마다 정규화한 하나의 가중치를 산출한다. 일반적으로 가중치산정을 위한 방식으로 고유치방법이 널리 활용되고 있으나, 이외에도 산술평균, 기하평균, 최소자승법, 조화평균, 평균치변환 등의 여러 방법이 있다. 또한 일관성 지수(Consistency index)와 일관성 비율(Consistency ratio)을 구하여 판단자료의 일관성 여부를 측정한다.

**단계 4: 대안별 우선순위 벡터 산정**

마지막으로 계층의 최상위에 위치한 의사결정의 목적을 달성할 수 있도록 해주는 최하위단계에 있는 대안들의 상대적인 우선순위를 나타내 주는 전체 계층에 대한 하나의 종합 우선순위 벡터(Priority vector)를 산출한다.

AHP에서는 이론적 배경으로 역수성(Reciprocal), 동질성(Homogeneity), 종속성(Dependency), 기대성(Expectations)의 4가지 공리를 갖는다. 역수성은 A요소가 B요소보다 x배 중요시된다면, B는 A보다 1/x배 중요하다는 의미이다. 동질성은 제한된 범위 내에서는 정해진 척도(Bounded scale)가 적용된다는 것이며, 종속성은 한 계층의 요소들은 인접한 상위계층의 요소에

대해 종속적임을 의미한다. 기대성은 의사결정의 목적에 관한 사항을 계층이 완전하게 포함하고 있다고 가정한다.

AHP의 응용분야는 대단히 많다. 에너지, 자원, 교통, 입지 등을 비롯한 경제문제에서부터 재무, 금융, 회계, 인사조직, 마케팅, 호텔, 관광 등의 경영문제, 정부, 국방, 등의 정치문제, 교육, 안전, 재해, 복지, 도시, 환경, 건설, 보건, 의료, 농업, 체육 등의 사회문제, R&D, 신제품개발, 생산, 제조, 품질, 컴퓨터, 정보 등의 기술문제에 이르기까지 광범위하게 활용되고 있다(Cho, 2003).

정보시스템 분야에서 AHP 방법이 적용된 사례는 소프트웨어 외주업체 선정방안과 ERP 시스템 도입 시 선정과정에서 AHP를 적용한 사례가 있었으며, 소프트웨어 품질평가와 웹사이트 평가에 AHP 방법을 적용한 사례가 있다(Cho, 2003).

**4. 분석대상 보정요소 선정**

소프트웨어 사업대가기준에서 개발비용 산정 시 고려하는 보정요소에는 애플리케이션 유형, 개발언어, 규모, 하드웨어 플랫폼, 납기, 품질수준 등이 있다. 여기서, 규모보정계수는 보통 라인수 또는 기능점수로 측정된 소프트웨어 규모의 함수로 표현되므로 AHP 분석에 적합하지 않다. 마찬가지로, 납기 보정계수도 소프트웨어 규모와 투입인력의 함수로 표현되므로 본 분석대상에서 제외한다. 나머지 4개의 보정계수 중에서 하드웨어 플랫폼과 품질수준은 시스템 특성 및 품질 보정계수로 조정·통합할 목적으로 수정작업이 진행 중이므로 본 연구에서는 논의하지 않기로 한다. 따라서 본 연구에서는 애플리케이션 유형과 개발언어 보정계수를 AHP 방법을 사용하여 결정하고자 한다. 특히, 애플리케이션 유형은 보정계수가 1.0에서

표 3. 현행 애플리케이션 유형 및 개발언어 보정

애플리케이션 유형	애플리케이션 유형 보정		개발언어 보정	
	적 용 범 위	보정계수	언 어	보정계수
사무처리형	인사, 회계, 급여, 영업, 의사결정지원 등 경영관리용 및 특정분야의 업무용 소프트웨어 등	1.0	4세대 언어 (4GL, Javascript 등)	0.65
과학기술형	과학계산, 시뮬레이션, 스프레드시트, 통계, OR, CAE 등	1.5	RPG	0.7
멀티미디어형	그래픽, 영상처리, 음성처리, 멀티미디어 응용분야, 교육·오락용 등	2.0	C언어, C++, Java, Perl 언어	1.3
지능정보형	자연어처리, 인공지능, 지식정보, 전문가시스템,	2.0	기타 고급언어 (COBOL, FORTRAN, PL/1, PASCAL, BASIC 등)	1.0
시스템형	운영체계, 언어처리 프로그램, DBMS, 인간·기계 인터페이스, 윈도시스템, CASE, 유틸리티 등	3.0		
통신형	통신프로토콜, 에뮬레이션, 교환기 소프트웨어 등	3.0	ASSEMBLY 언어	2.0
공정제어형 및 교환기형	생산지원, CAM, CIM, 기기제어, 로봇 분야 등	3.2		
지휘/통제형	군, 경찰 등 군장비·인력의 지휘통제를 요하는 소프트웨어	4.0		

4.0의 범위를 가지므로 애플리케이션 유형 보정계수에 의해 사업비가 최대 4배까지 차이가 날 수 있는 매우 중요한 보정계수이다. 현행 소프트웨어 사업대가기준에 제시된 애플리케이션 유형과 개발언어 보정계수는 다음 <표 3>과 같다.

소프트웨어 사업대가기준이 제정된 이후 소프트웨어 사업 환경과 제반 여건이 변화되면서 현행 애플리케이션 유형 및 개발언어 보정계수는 몇 가지 문제점을 안고 있다. 첫 번째, 충분한 데이터 축적이거나 전문가 의견수렴 없이 보정계수가 결정되었다. 그러나 국내 소프트웨어 사업에 관한 데이터 축적이 매우 부족한 실정이므로 소프트웨어 사업에 관한 실제 데이터를 기반으로 해서 보정계수를 결정하는 것은 현실적으로 불가능하다. 또한, 다양한 유형의 사업들에 관한 데이터를 골고루 수집하는 것도 쉽지 않다. 그럼에도 불구하고, 소프트웨어 사업에 대한 풍부한 경험을 갖고 있는 개발자·발주자·관리자들의 의견을 체계적으로 수렴하여 보정계수의 정확성을 제고할 필요가 있다. 두 번째, 보정요소의 세부분류가 변화된 현실을 제대로 반영하지 못하고 있다. 예를 들어, 현재의 애플리케이션 유형에서는 토지/건축물뿐만 아니라 해양/생태계 관리 등에 널리 사용되고 있는 지리정보시스템, 위치기반 서비스에 활발하게 사용되는 GPS(Global positioning systems), 그리고 실시간 또는 내장(Embedded) 소프트웨어 등을 고려하지 못하고 있다. 마찬가지로 개발언어의 분류도 명확하지 않아 Visual C++, Visual Basic 등을 일반 C++, BASIC과 혼동하거나, 다양한 스크립트(Script) 언어와 엑셀 등의 제5세대 언어 등도 포함하지 못하고 있다. 따라서 보정요소의 세부분류를 보다 체계화하고 변화된 현실에 맞게 수정할 필요가 있다.

위의 문제점을 해결할 수 있도록 애플리케이션 유형과 개발

언어 보정을 수정한 결과가 <표 4>에 제시되어 있다. 애플리케이션 유형은 ISO/IEC 12182의 분류를 참고하되 급격한 변동으로 인한 혼란을 줄일 수 있도록 애플리케이션 유형의 추가나 재분류보다는 보정계수의 정확성을 제고하는 데 중점을 두었다. 개발언어는 최근의 추세를 반영할 수 있도록 Caper Jones (Jones, 1996; Jones, 1998)가 제안한 언어세대별 분류에 따르도록 수정되었다.

## 5. 분석결과

<표 4>에 제시된 애플리케이션 유형 및 개발언어분류에 따라 애플리케이션 유형별 보정계수와 개발언어별 보정계수를 구하기 위해 각각 설문조사를 실시하고 그 결과를 AHP로 분석하였다.

### 5.1 애플리케이션 유형 보정계수

애플리케이션 유형별 보정계수를 AHP 기법에 의하여 산정하기 위하여 의사결정계층을 다음 <그림 2>와 같이 구성하였다.

목표 레벨로 애플리케이션 유형별 보정계수를 산정하는 것을 삼았으며, 각 대안들은 업무처리용, 과학기술용, 멀티미디어용, 지능정보용, 시스템용, 통신제어용, 공장제어용, 지휘통제용으로 구분되며 이들을 제어복잡도, 데이터 관리복잡도, 처리복잡도의 판단기준으로 평가하는 의사결정 계층도를 구성하였다.

판단기준을 제어복잡도, 데이터 관리복잡도, 처리복잡도로

표 4. 수정된 애플리케이션 유형 및 개발언어 보정

애플리케이션 유형 보정		개발언어 보정	
애플리케이션 유형	적 용 범 위	언어수준	언 어
업무처리용	인사, 회계, 급여, 영업 등 경영관리 및 업무처리용 소프트웨어 등	1세대 언어	어셈블리, 기계어, 자연어
과학기술용	과학계산, 시뮬레이션, 스프레드시트, 통계, OR, CAE 등	2~3세대 언어	C, COBOL, PASACL, C++, PL/1, Ads, PROLOG, RPG, UNIX, FORTRAN, Shell Script
멀티미디어용	그래픽, 영상, 음성 등 멀티미디어 응용분야, 지리정보시스템, 교육·오락용 등		
지능정보용	자연어처리, 인공지능, 전문가시스템	4세대 언어	ABAP4, Delphi, HTML, Java, Power Builder, SQL, Visual C++/COBOL/ BASIC, Oracle Developer 2000, 객체지향언어,
시스템용	운영체제, 언어처리 프로그램, DBMS, 인간·기계 인터페이스, 윈도우 시스템, 유틸리티 등		
통신제어용	통신프로토콜, 에뮬레이션, 교환기 소프트웨어, GPS 등		
공장제어용	생산관리, CAM/CIM, 기기제어, 로봇 제어, 실시간·내장형 소프트웨어 등	5세대 언어	EXCEL, 스프레드시트, 스크린 페인터
지휘통제용	군, 경찰 등 군장비·인력의 지휘통제를 요하는 소프트웨어		



용, 지능정보용, 시스템용, 통신제어용, 공정제어용, 지휘통제용의 개발비에 대한 상대적 가중치는 각각 7.6%, 9.5%, 10.1%, 13.1%, 13.3%, 14.4%, 15.0%, 17.1%로 분석되었다. 최종 애플리케이션 유형별 보정계수는 업무처리용(1.0), 과학기술용(1.2), 멀티미디어용(1.3), 지능정보용(1.7), 시스템용(1.7), 통신제어용(1.9), 공정제어용(2.0), 지휘통제용(2.2)으로 산정되었다.

표 6. 애플리케이션 유형 보정계수

	제어 복잡도	데이터 관리복잡도	처리 복잡도	상대적 가중치	보정계수
상대적 가중치	30.8%	36.1%	33.1%		
업무 처리용	5.1%	10.2%	7.2%	7.6%	1.0
과학 기술용	7.9%	10.3%	10.0%	9.5%	1.2
멀티 미디어용	9.5%	10.7%	10.1%	10.1%	1.3
지능 정보용	12.8%	13.5%	13.1%	13.1%	1.7
시스템용	13.5%	12.2%	13.6%	13.3%	1.7
통신 제어용	16.0%	13.1%	14.3%	14.4%	1.9
공정 제어용	16.5%	13.9%	14.9%	15.0%	2.0
지휘 통제용	18.6%	16.1%	16.8%	17.1%	2.2

5.2 개발언어 보정계수

개발언어 유형별 보정계수를 AHP 기법에 의하여 산정하기 위하여 의사결정계층을 다음 <그림 3>과 같이 구성하였다.

즉, 목표 레벨로 개발언어 보정계수를 산정하는 것을 삼았으며, 각 대안들은 1세대 언어, 2, 3세대 언어, 4세대 언어, 5세대 언어로 구분되며, 이들을 개발인력의 보편성과 개발 코딩시간,

디버깅시간의 판단기준으로 평가하는 의사결정 계층도를 구성하였다.

개발언어 측면에서 소프트웨어 개발비용에 영향을 미치는 요인이 개발언어에 대한 인력 측면과 개발언어의 생산성 측면이 주요요인이다. 따라서 판단기준을 개발언어의 인력 측면에서 개발인력의 보편성을 선정하였으며, 이는 각 개발언어별로 소프트웨어 인력시장에서 개발인력의 희소성의 정도를 의미한다. 개발언어에 대한 개발인력의 보편성이 낮을수록 인력을 구하기가 힘들게 되므로 소프트웨어 개발비용은 증가하는 관계를 판단기준으로 설정하였다. 다음으로 개발언어의 생산성 측면으로 개발언어에 대한 개발 코딩시간과 디버깅시간을 판단기준으로 선정하였다. 개발 코딩시간은 각 개발언어로 소프트웨어 개발 시 코딩에 소요되는 시간(또는 개발의 어려움 정도)을 의미한다. 개발 코딩시간이 길수록(또는 개발의 어려운 정도가 클수록) 소프트웨어 개발비용은 증가하는 것을 판단기준으로 설정하였다. 디버깅시간은 각 개발언어로 소프트웨어 개발 시 디버깅에 소요되는 시간이 길수록(또는 디버깅의 어려운 정도가 클수록) 소프트웨어 개발비용은 증가한다는 것을 판단기준으로 설정하였다.

AHP 분석을 위해서 판단기준에 대한 상대적 가중치를 평가한 설문으로 분석하였다. 여기서는 소프트웨어 개발비에 대한 보정계수를 산정하는 것을 의미하므로 단순한 판단기준에서의 상대적 가중치가 아닌 비용 측면에서의 상대적 크기를 설문 조사하였다. 애플리케이션 유형에 따른 보정계수 산정 시의 설문 조사와 동일하게, 일관성 지수를 사전에 검증하여 유의한 데이터를 유지할 수 있는 방법을 채택하였다.

회수된 145개의 설문 중 1개의 결측치(Missing data)를 제외한 데이터로 분석하였으며, 데이터의 이상치를 제거하기 위해 (평균 ± 3시그마)의 범위를 벗어나는 데이터를 이상치로 제거한 후 판단기준에 대한 상대적 가중치를 평균으로 산정하였다. 여기서 평균은 AHP 행렬의 특성인 대칭요소의 역수성을 보존할 수 있는 기하평균을 사용하였다. 이때 이상치로 판정된 데이터의 개수는 1개였으며, 143개의 유효 데이터로 분석하였다. 이들 데이터는 애플리케이션 유형 보정계수 산정 시와 동일하게 모두 일관성 지수가 0.1 이하로 유지되도록 조사하였다.

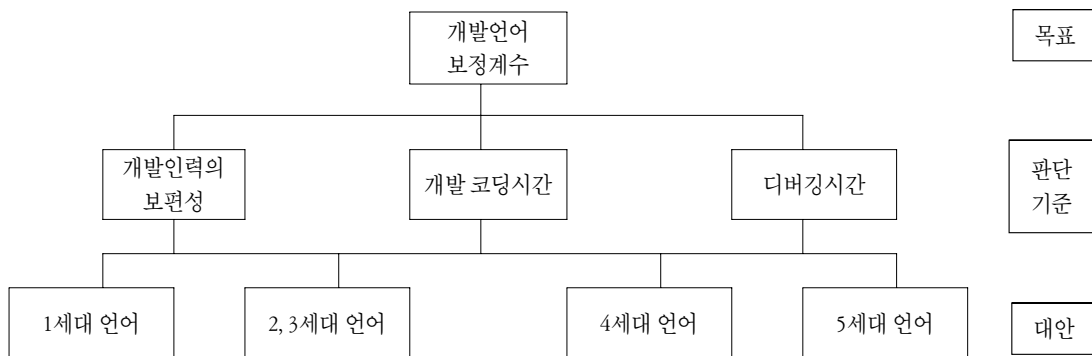


그림 3. 개발언어 보정계수 의사결정계층.



AHP 방법에 의해 산정된 각 판단기준에 대한 개발비의 상대적 가중치와 판단기준별 각 개발언어의 개발비에 대한 상대적 가중치를 산정한 후에, 각 개발언어별 개발비의 상대적 가중치를 산정하였다. 개발언어별 개발비의 상대적 가중치는 판단기준의 개발비에 대한 상대적 가중치와 각 판단기준별 개발언어의 개발비에 대한 상대적 가중치의 가중평균으로 산정되며, 보정계수는 2, 3세대 언어의 보정계수를 1.0으로 두었을 때 구해진 개발비에 대한 상대적 가중치의 상대적 크기로 산정되었다.

각 판단기준에 대한 개발비의 상대적 가중치는 개발인력의 보편성 : 개발 코딩시간 : 디버깅시간 = 33.1% : 39.4% : 27.5%로 분석되었으며, 1세대 언어, 2, 3세대 언어, 4세대 언어, 5세대 언어의 개발비에 대한 상대적 가중치는 각각 42.3%, 27.4%, 17.8%, 12.5%로 분석되었으며, 최종 개발언어 유형별 보정계수는 1세대 언어(1.5), 2, 3세대 언어(1.0), 4세대 언어(0.7), 5세대 언어(0.5)로 산정되었다.

표 7. 개발언어 보정계수

	개발인력의 보편성	개발 코딩시간	디버깅 시간	상대적 가중치	보정계수
상대적 가중치	33.1%	39.4%	27.5%		
1세대 언어	47.0%	40.3%	39.5%	42.3%	1.5
2, 3세대 언어	24.1%	29.3%	28.6%	27.4%	1.0
4세대 언어	14.8%	19.1%	19.6%	17.8%	0.7
5세대 언어	14.1%	11.4%	12.3%	12.5%	0.5

## 6. 결론

본 연구에서는 소프트웨어 사업대가기준의 개선작업의 일환으로 소프트웨어 사업대가기준의 보정요소를 현실에 맞게 재분류하고, 애플리케이션 유형과 개발언어에 대하여 AHP(Analytic hierarchy process) 방법을 도입하여 보다 정확하고 객관적인 보정계수를 산정하였다.

본 연구결과는 소프트웨어 사업대가기준의 개정을 위한 기초자료로 활용될 수 있으며, 또한 공공부문 정보화사업을 기획·관리하거나 직접 수행하는 사람이 정보화사업의 비용을 예측하고 그 결과를 해석하는 데 도움을 줄 수 있을 것이다. 또한, AHP의 활용을 소프트웨어 사업대가기준의 보정계수 산정에 처음으로 적용한 시도는 의미가 있다고 볼 수 있다.

## 참고문헌

Albrecht, Allan J.(1979), Measuring application development productivity, *Proc. IBM Application Development Symposium*, GUIDE Int. and SHARE Inc., IBM Corp., Monterey, CA, Oct 14-17, 83-92.

Boehm, B-W. et al.(2000), *Software cost estimation with COCOMO II*, Prentice Hall PTR.

Cho, K-T., Jo, Y-G., and Kang, H-S.(2003) *The Analytic Hierarchy Process*, Donghyun Pub., Seoul, Korea.

Furey, S.(1997), Why we should use function points, *IEEE Software*, 4(3), 28.

Golorath Inc., <http://www.galorath.com>

Gomolski, B-S. and Dallas, N-F.(2002), *Seven steps to a better IS bottom line*, Gartner Research (R-16-1452)

IFPUG(2000), *Function Point Counting Practices Manual(Release 4.2)*, International Function Point Users Group.

International Software Benchmarking Standards Group(2001), *Practical project estimation: A tool kit for estimating software development effort and duration*, ISBSG.

Jones, T. Capers(1996), *Programming Languages Table*, Release 8.2, Software Productivity Research.

Jones, T. Capers(1998), *Estimating Software Costs*, McGraw-Hill.

Kim, Hyunsoo(1997), An Improvement of Software Sizing and Cost Estimation Model with Function Point Methods, *Korean Management Science Review*, 14(1), 131-149.

Kemerer, C-F. and Porter, B-S.(1992), Improving the reliability of function point measurement: An empirical study, *IEEE Transactions on Software Engineering*, 18(11), 1011-1024.

Korea Software Industry Association(2001), *A Study on the structure of software cost estimation guidelines*, Seoul Korea.

Lee, Y-K.(1997), Software development cost estimation using function point, *Business Administration Study*, 6, 241-261.

Matson, J-E., Barrett, B-E., and Mellichamp J-M.(1994), Software development cost estimation using function points, *IEEE Transactions on Software Engineering*, 20(4), 275-287.

National Computerization Agency(2002), *Informatization White Paper 2002*, Seoul, Korea.

Park, Chan-Kyoo, Gu, J-H., Kim, S-D., Shin, S-J., and Song, B-S.(2002), A Study on the Estimation of Software Development Cost of IT Projects in Public Sector, *Korean Management Science Review*, 19(2), 191-204.

Price Systems, <http://www.pricystems.com>

Pressman, R-S.(1997), *Software engineering: A practitioner's approach*, 4th ed., McGraw-Hill.

Putnam L. and Myers W.(1992), *Measures for excellence*, Yourdon Press.

Satty. T.L.(1990), *Multicriteria Decision Making: The analytic hierarchy process*, RWS Publications.

Software Productivity Research, <http://www.spr.com>

The Standish Group(1995), *Chaos*, Standish Group Report.

Vidger, M-R. and Kark A-W.(1994), *Software cost estimation and control*, Institute for Information Technology, National Research Council Canada, 1994

**김우제**

서울대학교 산업공학 학사  
 서울대학교 산업공학 석사  
 서울대학교 산업공학 박사  
 현재: 서울산업대학교 산업정보시스템공학과  
 교수  
 관심분야: 경영과학 및 응용, 소프트웨어 공학

**신수정**

이화여자대학교 컴퓨터공학과 학사  
 이화여자대학교 컴퓨터공학과 석사  
 현재: 한국전산원 전자정부지원팀 책임  
 연구원  
 관심분야: 데이터마이닝, 소프트웨어 공학

**박찬규**

서울대학교 산업공학 학사  
 서울대학교 산업공학 석사  
 서울대학교 산업공학 박사  
 현재: 동국대학교 경영학과 교수  
 관심분야: 경영과학, 정보시스템 및 수리계획  
 응용분야