

# 소프트웨어 유지보수 아웃소싱 벤더의 비용에 관한 연구

유지나\*, 김병수\*, 최민석\*, 오정석\*

## Estimation of Vendor Cost for Software Maintenance and Repair Outsourcing

Jeena Yoo, Byoung Soo Kim, Minn Seok Choi, Jungsuk Oh

This paper considers probable determinants of the application software outsourcing costs and focuses on the service level agreement (SLA) influence. Recently, outsourcing clients and vendors have contracted SLA and tried to reflect them on outsourcing prices. While application software maintenance has been one of the most frequently outsourced functions, its cost did not reflect the service level requirements by the client. This paper empirically shows that the service urgency requirement on SLA increases the maintenance programmers' effort and thus increases the vendor's cost. This result can be used by vendors for program allocation purposes.

**Keywords :** IS Outsourcing, SLA, Vendor Costs

## I. 서 론

아웃소싱은 최근 10년간 경영학자들과 경제학자들의 입에 자주 오르내리는 주제이다. 가트너(Gartner Research Group)의 시장조사에 따르면 세계 주요 기업의 경영인 대부분이 비용절감이나 핵심역량 확보를 위해 정보시스템(Information Systems: IS) 아웃소싱을 고려하고 있는 것으로 나타났다. 이에 따라 정보시스템 아웃소싱은 비즈니스 수행함에 있어 주요 동향으로 자리잡고 있다[Maurer, 2000].

정보시스템 아웃소싱의 개념은 정보시스템 활용이 증가함에 따라 아웃소싱의 목적 및 대상, 그리고 아웃소싱 기간이 지속적으로 변화하고 있다. 과거에는 고객사의 IT부서에서 비용 절감이나 핵심 역량으로의 집중을 목적으로 하여 애플리케이션 패키지를 도입하거나 계약을 통해 프로그래밍을 주문 제작하거나 특정 정보처리 서비스를 아웃소싱 하였던데 반해, 오늘날에는 시스템 통합 및 운영, 애플리케이션 및 소프트웨어의 운영 등과 같은 IT 부서의 핵심 업무들을 아웃소싱 하고 있다[Grover *et al.*, 1994]. 즉, 오늘날의 정보시스템 아웃소싱은 기업의 전략적 선택의 일환으로 활용되고 있다. 한편 아웃소싱 기간에서도 과거에는 아웃소싱 서비스 공급자들이 일회성 계약을 선호하였으나 오늘날에는 장기 계약을 선호하면서 서비스의 운영과 향후 서비스 책임에 더 많은 관심을 기울이고 있다[Aucoin, 1991].

정보 시스템 아웃소싱 가운데 소프트웨어 유지보수의 아웃소싱이 특히 각광 받고 있다. 이는 IT부서에서 차지하는 비중이 큰 소프트웨어 유지보수 업무가 아웃소싱 될 때 그 효과가 가장 클 것으로 기대하기 때문이다. 실제로 IT부서의 전체 예산에서 소프트웨어 유지보수가 차지하는 비율은 작게는 50%에서 많게는 80%에 이른다는 조사결과가 있다[Banker *et al.*, 1991]. 또한, 포레스터 리서치(Forrester Research, Inc)는 소프트웨어 유지보수 비용이 정보시스템 운영 비용의 70%

를 삼켜버린 “블랙홀(black hole)”이라고 묘사한 바 있다. 이런 이유에서 많은 기업들이 기업의 효율성 제고를 위해 소프트웨어 유지보수 업무를 아웃소싱 하고 있다.

소프트웨어 유지보수 아웃소싱을 포함한 모든 정보시스템 아웃소싱에서 아웃소싱 서비스 제공사업자(이하 벤더)와 고객, 모두가 수용할 수 있는 서비스 가격을 산정하는 방법에 대해 지속적으로 논의되고 있다. 일반적으로 고객과 벤더는 서비스 범위의 해석과 성과 측정의 기준, 비용 산출 방법, 측정된 성과 및 산출된 비용에 대한 수용 여부에서 견해 차이를 드러내고 있다[Lacity *et al.*, 1996; Bryson and Ngwenyama, 2000]. 한국산업연구원[2002]에서 실시한 기업의 CIO들을 대상으로한 설문조사에서는 ‘적절한 서비스 범주 및 성과 측정의 기준 부재’를 정보시스템 아웃소싱에서의 가장 큰 걸림돌로 꼽았다.<sup>1)</sup> 또한, Swanson[1997] 연구에서도 성과 측정 기준이 명확하게 정립되지 있지 않을 경우 측정된 성과와 부과된 가격에 대해 고객과 벤더간의 합의에 이르기 어렵다는 사실을 밝힌바 있다.

아웃소싱 서비스의 고객과 벤더 사이에서 발생할 수 있는 측정 기준의 모호성을 해결하기 위해 서비스 수준 협약(Service Level Agreement: SLA)이 등장하였다. SLA는 아웃소싱 서비스의 범위 및 수준을 정의하여 고객과 벤더간에 합의한 문서이다[Schneither and Bowen, 1993]. 가트너에서는 SLA를 정보시스템 아웃소싱 서비스의 이해당사자인 고객과 벤더가 주고 받는 서비스에 대해 명확하고 세부적으로 기술한 계약서로 정의하고 있다. 최근 SLA는 아웃소싱에서의 핵심 성공요인으로 더욱 주목 받고 있다. 고객과 벤더가 SLA를 의사소통의 수단으로 적극적으로 활용하면 고객은 필요한 서비스의 수준을 명시적으로 벤더에게 요구할 수 있고 벤더는 책임 범위

1) Gartner에서도 같은 결과의 설문 조사가 이루어졌다[2002].

를 명시적으로 고객에게 전달할 수 있으며 합의된 기준으로 성과를 측정할 수 있기 때문에 성과와 연계된 보상 시스템 운용을 쉽게 할 수 있다 [Marriott, R. 2003].

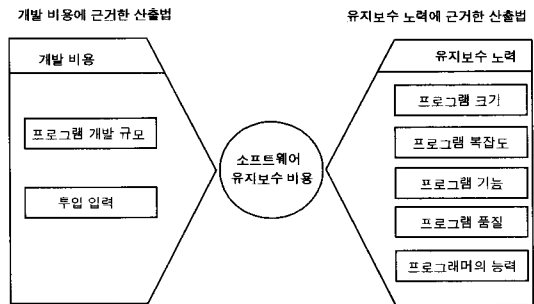
그러나 SLA는 아웃소싱의 의사소통 도구로서 많이 활용되어 고객과 벤더가 성과 측정에 대해 합의할 때는 도움을 주고 있지만, 아웃소싱 서비스의 가격 및 비용 산정할 때에는 아직까지 SLA가 고려되지 않고 있다. 많은 아웃소싱 연구자들은 SLA에서 합의한 서비스 범위 및 수준과 서비스 가격을 잘 연계시켜야만 SLA의 효과를 최대화할 수 있을 것이라고 제안한다[남기찬 외, 1999]. 일반적으로 서비스 가격은 곧 서비스 비용을 근거로 책정되기 때문에 SLA와 연계한 아웃소싱 서비스의 가격 책정의 문제는 곧 SLA와 연계한 비용 산출로 귀착된다.

본 연구에서는 SLA에서 합의한 서비스 수준을 반영한 소프트웨어 유지보수 아웃소싱의 비용 산정 모형을 제안하고자 한다. 특히, 대표적인 소프트웨어 유지보수 대상 중 하나인 애플리케이션 소프트웨어의 유지보수 아웃소싱의 비용을 산정에 대해 살펴볼 것이다. 먼저 기존 문헌을 통해 애플리케이션 소프트웨어 유지보수를 비롯한 정보시스템 아웃소싱에서의 기존 비용 산정 모형을 살펴보고 SLA의 항목을 포함하는 새로운 비용 산정 모형을 제시하고자 한다. 이어서 제시한 모형에서의 SLA변수들과 유지보수 비용과의 관계를 국내 한 SI업체의 현장 데이터를 이용하여 실증적으로 유의함을 보여 주고 있다. 이 논문 결과를 통해 향후 시장의 확대가 이루어질 것으로 보이는 애플리케이션 소프트웨어 개발/유지보수 아웃소싱 계약에서 고객과 벤더간에 SLA의 활용도를 높이고, 벤더 입장에서 서비스 수준에 기초하여 비용을 산정할 수 있을 것으로 기대된다.

## II. 문헌 연구

소프트웨어의 비용 산출 모형은 대부분 소프

트웨어 엔지니어링 분야에서 주로 연구되어 왔다. 그런데 유지보수 비용의 산정이 연구 주제로 관심 받게 된 것은 최근의 일이다. 주로 유지보수 비용보다는 개발 비용의 산정에 관한 연구가 많이 진행되었다. 이런 연유로 유지보수 비용에 관한 초기의 연구는 대개 개발 비용의 일정 비율을 유지보수 비용으로 산정하는 방식으로 하고 있다. 최근에는 유지보수와 관련된 원가 동인에 기초한 직접 추정 방식이 주를 이루고 있다. 본 연구에서는 <그림 1>에서 보는 바와 같이 소프트웨어 엔지니어링의 개발노력을 추정하는 모형과 유지보수 비용을 직접 추정하는 모형에 관한 문헌을 모두 고찰하고 있다.



<그림 1> 소프트웨어 유지보수 비용 산출 연구

### 2.1 소프트웨어 개발비용에 근거한 산출법

개발 비용에 근거한 유지보수 비용의 산정을 이해하기 위해서는 소프트웨어 개발 비용 산정 방법에 대한 이해가 선행되어야 한다. 소프트웨어 개발 비용은 프로그램의 개발노력(effort)을 추정하여 단위 노력당 비용을 곱하여 계산되거나 개발에 참여한 투입인력과 인력단가를 곱하여 산정된다. 즉, 프로그램 개발노력을 추정하거나 투입인력의 정보를 확보하면 비용을 추정할 수 있다는 논리이다. 이때 프로그램의 개발노력을 추정하는 대표적인 모형에는 코드라인수 방식과 기능점수 방식이 있다. 구체적으로 Boehm [1991]의 COCOMO(Constructive Cost) 모형은

코드라인수 방식으로 프로그램의 크기를 측정하여 개발노력을 산출하고, IBM의 기능 점수(Function Point: FP) 모형은 기능 점수를 이용하여 프로그램의 개발노력을 추정한다. 일반적으로 개발노력의 단위로는 공수(MM, man-month)를 사용한다.

COCOMO의 최신 모형인 COCOMO II-PA(Post-Architecture)는 원가를 발생시키는 동인으로 제품(product), 플랫폼(platform), 직원(personnel), 프로젝트(project)를 고려하고 있으며, 소프트웨어 프로그램의 크기와 개발 자원의 투입량을 기준으로 공수를 산출한다. 프로그램의 크기는 프로그램의 LOC(Line of Code)로 측정한다. 소프트웨어 개발비 증감의 조정 인자(scale-factor)는 경험도, 개발의 유연성, 아키텍처/위험 해결성, 팀의 결속력, 프로세스 성숙도에 따라 다른 값을 갖는다.

IBM의 기능점수 모형은 개발하고자 하는 소프트웨어 프로그램에서 구현될 기능의 규모와 난이도를 근거로 프로그램의 개발노력을 추정한다. 프로그램에서 구현하는 기능을 트랜잭션(transaction)과 데이터(data) 측면에서 모두 나열한 후 각 기능에 대해 복잡도를 고려한 조정인자를 곱하여 모든 기능의 점수를 합하면 소프트웨어 프로그램의 기능점수가 산출된다.

한편 국내 정보통신부에서는 소프트웨어 사업 대가 기준을 2000년부터 고시하고 있는데, 2004년 개정된 고시(정보통신부 고시 제2004-8호)에서는 프로그램의 개발 규모(앞의 프로그램 개발노력과 같은 개념)에 의한 방식과 투입 인력을 기준으로 하는 방식 모두를 수용하고 있다. 특히, 개발 규모는 코드라인과 기능 점수 방식으로 계산할 수 있는 각각의 기준을 포함하고 있다. 투입입력을 기준으로 개발 비용을 산정할 때 '소프트웨어사업의 노임 단가'를 참고할 것을 권장하고 있다. 한편, 유지보수 비용은 식 (1)과 식 (2)에서 보는 바와 같이 소프트웨어 개발비의 10~15%내에서 산정하도록 하고 있는데, 유지보수의 난이도를 계산하는 상세한 기준은 <표 1>에서 확인할 수 있다.

$$\text{유지보수난이도 (\%)} = 10 + 5 \times \text{TMP}/100, \quad (\text{단, TMP는 업무 복잡도 점수}) \quad (1)$$

$$\text{유지보수대가} = \text{유지보수난이도 (\%)} \times \text{SW개발비산정가} \quad (2)$$

<표 1> 용역 유지보수 대가 산정기준

유지보수 대상 시스템의 특성	단순		보통		복잡	
	기준 (년간)	점수	기준(년간)	점수	기준(년간)	점수
유지보수 횟수	4회 이하	0	12회 이하	20	12회 초과	35
자료처리 건수	10만 미만	0	10-50만	10	50만 초과	25
타시스템 연계	없음	0	1-2 시스템	5	3개 이상	10
실무지식 필요	별도지식 불필요	0	기초지식 이해 필요	5	전문실무 능력 필요	10
분산처리 여부	실시하지 않음	0	통합하의 분산처리	10	순수분산 처리	20

자료: 정보통신부[2004].

위와 같이 소프트웨어 개발 비용의 일정 비율을 유지보수 비용으로 산정한 것은 문제점을 가지고 있다. 첫째, 소프트웨어 개발에서의 원가 동인과 유지보수에서의 원가 동인이 동일하지 않을 가능성이 높기 때문에 개발 비용에 영향을 주는 변수를 바탕으로 산출한 위와 같은 방식에는 문제가 있다. 특히, 개발 비용이 비합리적으로 산정되었을 때, 유지보수 비용도 왜곡되어 산출될 수 있다. 둘째, 내부에서 개발한 소프트웨어의 유지보수를 외부 업체에 아웃소싱하는 경우에는 근본적으로 유지보수 비용을 산출할 수 있는 근거가 없다. 그러므로 개발 원가를 바탕으로 유지보수 비용을 산출하기보다는 유지보수 업무의 특성을 바탕으로 유지보수 비용을 산출하는 것이 더 바람직하다.

## 2.2 소프트웨어 유지보수 노력에 근거한 산출법

소프트웨어 유지보수는 매우 어렵고, 효율적으

로 관리하기 어려운 분야이다[Lientz and Swanson, 1980]. 소프트웨어 유지보수는 소프트웨어 개발 못지 않게 상당한 시간과 자원이 소요된다. 설문 조사에 따르면 프로그래머들은 프로그램의 개발뿐 아니라 유지보수에도 전문 인력을 배치해야 한다는 공감대가 형성되어 있는 것으로 나타났다.

일반적으로 유지보수 비용은 유지보수 노력으로부터 산출되는데, 소프트웨어 유지보수의 노력의 산정은 다른 유지보수 업무의 노력의 산정에 비해 어렵다. 한국 전산원[2000]에서는 유지보수 업무를 4 가지(애플리케이션 운영, 시스템 운영, 네트워크 운영, 헬프데스크 운영)로 구분한 후 정보시스템 아웃소싱의 노력을 산정할 수 있는 모형을 추정하였는데, 각 업무의 기본 업무량을 식 (3)과 식 (4), 식 (5)와 같이 비교적 쉽게 추정한 후, 난이도의 곱하여 점수로 환산해서 비용을 산정할 수 있다고 한데 반해, 애플리케이션은 고려해야 할 변수가 상대적으로 많아서 단순한 모형을 추정하기 어렵다고 밝히고 있다.

$$\text{시스템운영노력} = f(\text{MIPS}) \quad (3)$$

$$\text{네트워크운영노력} = f(\text{line수}) \quad (4)$$

$$\text{헬프데스크운영노력} = f(\text{PC수}) \quad (5)$$

한편, 소프트웨어 유지보수 비용은 프로그램의 특성과 프로그래머의 특성에 따라 차이가 있다[Lientz and Swanson, 1980].

프로그램 특성 변수 가운데 프로그램의 크기가 유지보수 노력에 도 가장 큰 영향을 주는 것으로 알려져 있다[Banker, 1991]. Banker[1993]는 기능점수 모형과 COCOMO 모형으로 개발 비용을 산정하는 것과 마찬가지로 프로그램의 크기를 바탕으로 유지보수 비용도 산정할 수 있을 것이라고 주장하였다. Lientz and Swanson[1980]와 Kemerer and Slaughter[1997]도 프로그램 크기와 유지보수 노력 사이에 양의 상관관계가 있음을 밝혀냈으며, Banker et al.[1991]는 소프트웨어 개발 단계의 프로그램의 크기가 유지보수 노력

에 주는 영향을 실증적으로 분석하였다.

또 다른 프로그램의 특성에는 소프트웨어의 복잡도가 있다. 복잡도란 소프트웨어의 데이터 구조와 프로세스를 이해하고 변경하는데 있어서의 어려운 정도이다[Boehm, 1991]. 프로그램이 복잡할수록 프로그래머가 프로그램을 이해하는데 시간이 오래 걸려서, 오류 코드를 발견하는데 더 많은 시간이 소요되기 때문에 더 많은 유지보수 노력이 필요하다. 소프트웨어의 복잡도의 측정 도구로 브랜치의 수(number of branches)[Boehm, 1991]와 모듈(single logical task)의 크기[Banker, 1993] 등이 사용되고 있다.

$$\text{유지보수노력} = f(\text{프로그램 크기}, \text{프로그램 복잡도}, \text{다른원가 동인들}) \quad (6)$$

식 (6)을 바탕으로 회귀분석을 실시하였더니 프로그램의 크기가 클수록, 복잡도가 높을수록 유지보수 노력이 증가하였다[Boehm, 1991]. 식 (6)에서의 다른 원가 동인들에는 프로그램 기능, 품질, 유지보수 소프트웨어의 생명 주기, 프로그래밍 언어 등이 있다.

유지보수 프로그램의 담당 기능에 따라 유지보수 노력의 정도가 달라진다는 연구들이 최근에 등장하고 있다. Kemerer and Slaughter[1997]는 애플리케이션 소프트웨어가 전략적으로 중요한 기능을 담당할 수록 더 많은 유지보수 노력이 필요하다는 것을 밝혀냈다.

소프트웨어 유지보수 서비스의 비용은 고객이 요구하는 품질과 벤더가 제공하는 품질에 따라 다르게 책정된다. Palvia[1995]와 Kemerer and Slaughter[1997]는 소프트웨어 유지보수 노력과 고객이 느끼는 품질간의 관계를 실증적으로 연구하였으나, 유의성이 높지 않은 것으로 나타났다.

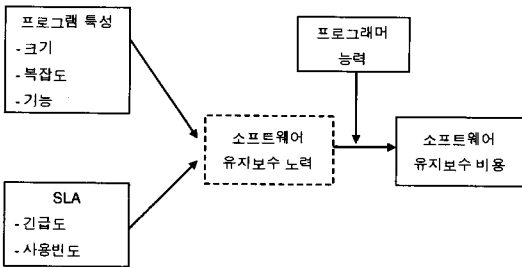
Banker et al.[1991]과 Demarco[1982]는 프로그래머들의 능력에 따라 소요되는 유지보수 노력이 상이하다는 것을 실증 연구를 통해 증명하였다. 두 연구는 경험이 풍부한 프로그래머들일수록 문제 발생의 원인을 신속하게 파악하여 오류

코드를 수정하기 때문에 소프트웨어 유지보수의 생산성이 높다고 주장하였다.

### Ⅲ. 연구 모형 및 자료 수집

#### 3.1 연구 모형

앞서 고찰한 소프트웨어 유지보수 산정 모형에서 유지보수 노력에 유지보수에 참여한 인력의 단가를 곱하면 유지보수 비용을 직접적으로 계산할 수 있다. 따라서 만약 유지보수 노력을 추정하는 대신 직접 측정하고 여기에 인력 단가를 곱하여 유지보수 비용을 산출한다면 유지보수 노력을 추정할 때 이용한 독립변수들을 유지보수 비용을 추정할 때 이용해도 될 것이다. 따라서 본 연구에서는 소프트웨어 유지보수 비용 산출과 연관 있는 것으로 밝혀진 프로그램의 특성과 함께 SLA를 바탕으로 한 비용 산정 모형을 구성하였고, <그림 2>에 나타내었다. 한편, 아래 모형에서 고려하고 있지 않은 프로그램 품질 등의 기타 프로그램 특성 변수들은 분석을 위해 통제하고 있다.



<그림 2> 연구 모형

#### 3.2 변수 정의

##### 3.2.1 종속 변수: 유지보수 비용

종속변수인 유지보수 비용은 프로그래머가 일

정 기간 동안 일한 시간 비용을 의미하는데, 프로그래머가 일한 공수에 각 프로그래머의 인력 단가를 곱하여 산출한다. 이때, 프로그래머의 공수는 직접 측정하고, 각 프로그래머의 인력 단가를 직접 곱하는 대신에 중급기술자를 기준으로 기술 등급별로 공수에 조정값을 곱하여 종속변수를 산출할 수도 있다. <표 2>에서 보는 바와 같이, 한국 소프트웨어 산업협회에서 발표한 소프트웨어 사업의 노임 단가가 특급, 고급, 중급, 초급간에 차이가 각각 1.2배다라는 것에 근거하여 초급, 고급, 특급 기술자의 조정값으로 각각 0.8과 1.2, 1.5를 곱한다.

<표 2> 2004년 소프트웨어 사업의 노임단가

구분	소프트웨어 노임단가(원)
기술사	212,056
특급기술자	190,624
고급기술자	146,154
중급기술자	118,744
초급기술자	92,622
고급기능사	75,790
중급기능사	67,577
초급기능사	56,900

자료: 한국소프트웨어산업협회[2004].

##### 3.2.2 SLA 특성 변수

본 연구에서 제안한 모형에 포함된 독립변수는 크게 SLA 특성 변수와 프로그램 특성 변수로 구분한다. SLA특성 변수는 사용빈도 변수와 긴급도 변수로 구분한다. 사용빈도와 긴급도 변수가 실증 분석을 위해 선정된 것은 다음과 같은 이유 때문이다. 첫째, 긴급도와 사용빈도가 높을수록 프로그램 유지보수가 어렵고 실제로 유지보수 비용이 증가한다는 프로그래머들의 현장 경험을 반영한 것이다. 둘째, 대부분의 SLA 계약에 두 항목이 포함되어 있기 때문에 측정된 데이터를 얻기 쉬운 것으로 판단하였다. 실제로 소프

트웨어 유지보수 프로그램 별로 성과를 측정할 때, 계약 수준에서 합의한 사용빈도와 긴급도를 바탕으로 인센티브(incentive)와 패널티(penalty)를 관리하고 있었다.

- (1) 사용빈도: SLA에 계약된 사용빈도 A, B, C는 해당 애플리케이션이 얼마나 자주 사용되는 것인지를 나타낸다. 고객과 관련된 프로그램들은 업무시간 내에 계속 원활히 운영되어야 하며 온라인 프로그램들은 잠시라도 장애가 생기면 곤란하다. 즉 사용빈도가 높을수록 전체 시스템에서 기본적이고도 주요한 역할을 하는 애플리케이션이라고 볼 수 있으므로 사용빈도가 높을수록 유지보수 노력이 높을 것으로 예상하여 양의 관계가 있을 것이다.
- (2) 긴급도: SLA에서 요구하고 있는 긴급도 A, B, C는 해당 애플리케이션에 장애 발생시 긴급도가 높은 것부터 우선적으로 해결되어야 함을 말한다. 따라서 긴급도가 높을수록 더 많은 노력이 투입될 것이다.

사용빈도와 긴급도는 각각 프로그래머가 담당하는 프로그램의 비율로 나타내었다. 예를 들면 긴급도 A 프로그램의 비율은 식 (7)과 같이 정의한다.

$$URG\_AL = \frac{SLA의 긴급도가 A라고 표시된 프로그램의 LOC}{한 사람이 유지보수하고 있는 프로그램의 총 LOC} \quad (7)$$

### 3.2.3 프로그램 특성 변수

본 연구에서 이용한 프로그램 특성 변수로는 프로그램 크기, 복잡도, 프로그램 기능이 있다. 프로그램 크기는 전체 담당 프로그램의 코드라인수와 담당하는 프로그램 수로 구분하고, 복잡도는 프로그램의 내부와 외부 커넥션 수로 구분

하며, 프로그램 기능은 배치(batch) 프로그램 담당과 서버/클라이언트 프로그램 담당으로 구분하고 있다.

- (1) 프로그램 크기: 소프트웨어 운영의 투입 노력에 가장 큰 영향을 주는 것으로 알려진 것이 크기 변수이다. 많은 연구자들이 크기가 클수록 유지보수 노력이 많이 투입된다는 결과를 얻었다. 본 연구에서는 KLOC를 기본 단위로 프로그램의 크기를 측정하였다.
- (2) 복잡도: 많은 연구에서 프로그램의 복잡도가 높을수록 유지보수 노력이 많이 필요하다는 결과를 얻었다. 본 연구에서는 각 프로그램들이 내부에서의 커넥션 수와 그리고 외부 프로그램과 연관되어 있는 커넥션의 수를 측정도구로 삼았다.
- (3) 프로그램 기능: 프로그램 기능에 따라 유지보수 비용에 차이가 있다. 문헌연구에서 많은 소프트웨어 공학 연구자들은 애플리케이션 유지보수 업무는 애플리케이션의 특성에 영향을 받는다고 밝혀냈다. 프로그래머와의 인터뷰에서도 빌링과 고객관리에 사용되는 애플리케이션에 소요되는 유지보수 노력의 형태가 다르다는 사실을 확인하였다. 빌링 업무는 한달 단위로 정산되는 배치 프로그램인 것으로 나타났고, 고객관리업무는 실시간으로 운영되는 업무임을 알 수 있었다.

앞에서 정의한 종속 변수와 독립 변수의 조작적 정의와 측정 방식 및 도구는 <표 3>과 같다.

### 3.2.4 통제 변수

한편 본 연구에서는 프로그램의 특성 중 품질과 애플리케이션의 유지보수 주기, 그리고 프로그램 언어에 대해서는 통제 변수로 선정하고 동일한 조건에서 데이터를 수집하였다.

<표 3> 변수의 조작적 정의 측정

구분	변수		조작적 정의	측정 및 단위	비고	
종속 변수	유지보수비용		소프트웨어 유지보수에 소요한 비용	공수(Mh) *인력 단가조정계수	SCALED_HRS	
독립 변수	SLA 특성	사용 빈도	높은 사용빈도의 프로그램 담당	SLA에서 합의한 높은 등급의 사용빈도 프로그램 담당 비율	사용빈도 A등급의 프로그램 담당 비율	FREQ_AL
			평균 사용빈도의 프로그램 담당	SLA에서 합의한 평균 등급의 사용빈도 프로그램 담당 비율	사용빈도 B등급의 프로그램 담당 비율	FREQ_BL
			낮은 사용빈도의 프로그램 담당	SLA에서 합의한 낮은 등급의 사용빈도 프로그램 담당 비율	사용빈도 C등급의 프로그램 담당 비율	FREQ_CL
		긴급도	높은 긴급도의 프로그램 담당	SLA에서 합의한 높은 긴급도 프로그램 담당 비율	긴급도 A등급의 프로그램 담당 비율	URG_AL
			평균 긴급도의 프로그램 담당	SLA에서 합의한 평균 긴급도 프로그램 담당 비율	긴급도 B등급의 프로그램 담당비율	URG_BL
			낮은 긴급도의 프로그램 담당	SLA에서 합의한 낮은 긴급도 프로그램 담당 비율	긴급도 C등급의 프로그램 담당비율	URG_CL
	프로그램 특성	크기	프로그램의 크기	담당하는 모든 프로그램의 코드라인 수의 합	KLOC	KLOC_TOT
			프로그램의 수	담당하는 프로그램의 수	개	PROC_NO
		복잡도	내부 연계	프로그램 내부 참조 정도	내부 커넥션 수	INNER_CO
외부 연계			외부 프로그램과의 연계 정도	외부 커넥션 수	OUTTER_CO	
기능		배치 프로그램 담당	배치 프로그램의 비율	전체 담당 프로그램 중 배치 프로그램의 비율	BATCH_L	
	서버/클라이언트 프로그램담당	서버/클라이언트 프로그램의 비율	전체 담당 프로그램 중 서버/클라이언트 프로그램의 비율	CLIENT_L		

### 3.3 연구 대상

본 연구에서 제시한 연구 모형을 검증하기 위해 소프트웨어 유지보수 아웃소싱 서비스를 제공하면서 SLA로 계약을 체결하고 있는 국내 SI 업체인 M사를 선정하였다. M사는 통신 서비스 사업자를 대상으로 빌링과 고객서비스 프로그램을 유지보수하고 있으며 3년 전부터 SLA를 서비스 계약에 활용하고 있다. 성과 관리 전담 부서에서 업무 그룹 단위로 사용빈도, 긴급도, 장애범위를 기준으로 3단계(A, B, C)로 측정하여 계약수준과 비교하여 계약금액의 4% 내에서 인센티브와 페널티를 부여하고 있다. 특히, SLA에서는 장

에 발생시간부터 종료시간까지를 분(min)으로 측정하여 감점하는데 긴급도가 높을수록 감점되는 점수 폭이 크다.

H팀은 총 94명으로 구성되어 있는데 기술등급별로 각각 특급 5%, 고급 10%, 중급 34%, 초급 45%, 기타 6%로 분포되어 있었다. 이때 특급 프로그래머들은 전원 관리자이고, 프로그램을 담당하고 유지보수 업무를 담당하는 프로그래머들은 고급 이하의 프로그래머들이기 때문에 고급 이하의 프로그래머를 분석 대상으로 한다.

H팀의 업무는 고객관리, 빌링, 지원관리 등으로 구성되어 있는데 각 업무마다 복수의 단위 시스템을 유지보수하는 임무를 담당하고 있었다.



총 41개의 단위 시스템 중 프로그래머들은 하나 이상의 단위 시스템을 유지보수하고 있다. 각 단위 시스템은 수 십 내지 수 백 개의 모듈로 구성되어 있는데 각 모듈별로 사용빈도와 긴급도가 다르다.

<표 4> 연구 대상(M사 H팀)의 업무

업무 그룹	프로그램 라인수	인원
공통 관리	242,478	13
고객관리	2,706,548	34
빌링	1,346,271	26
지원관리	710,564	8
통계	68,820	7
종합인사	1,134,448	6
합 계	6,209,129	94

H팀에서 관리하는 단위 시스템의 품질은 비슷하다. SLA에 따라 모든 애플리케이션의 성과 점수가 96점을 만족하여야 하며 이를 기준으로 삼아서 그보다 좋은 성과를 냈을 때는 인센티브를 받고 그보다 못한 성과를 냈을 때는 패널티를 받는다. 그러나 기본적으로 H팀은 베이스라인(baseline)만큼의 서비스를 제공하는 것을 목표로 유지보수 서비스를 제공하고 있고 실제로도 그렇게 하고 있는 것으로 나타났다. 따라서 일정한 품질의 서비스를 제공하고 있는 것으로 가정하였다.

H팀의 애플리케이션들은 모두 1999년 4월에 코딩이 완료되어 유지보수로 접어들었으므로 애플리케이션 수명 주기는 모두 같은 단계에 있다고 볼 수 있다.

한편, H팀에서 담당하는 프로그램은 대부분 C로 작성된 것이지만 델파이(Delphi)로 작성된 것도 일부 있다. H팀이 유지보수하는 프로그램들은 통계팀을 제외하고는 모두 자체 개발한 프로그램들인데 Delphi와 C언어로 작성되어 있었다. 그러나 본 연구에서 고려된 주요 프로그램들은 대부분 C로 작성된 것들이었고, 실무진들과의 회

의 결과 Delphi로 작성된 프로그램의 경우에는 LOC로 크기를 측정하는 것이 비합리적이라고 판단하였다. 따라서 Delphi로 작성된 프로그램들의 LOC는 포함시키지 않고 동일한 프로그램 언어로 작성된 소프트웨어들만 분석에 이용하였다.

### 3.4 데이터 수집

H팀은 SLA를 기초로 매월 성과를 측정하기 때문에 SLA 관련 변수에 관한 데이터는 쉽게 확보할 수 있었지만 개별 프로그래머가 특정 프로그램에서 유지보수하는 작업시간에 대한 데이터는 축적되어 있지 않았다. 그래서 분석을 위해 2004년 8월 한달 동안 개별 프로그래머의 작업일지를 작성하게 하여 필요한 데이터를 얻을 수 있었다.

M사 H에서 유지보수 아웃소싱 업무를 전담하고 있는 총 73명의 담당자들을 대상으로 데이터를 수집하였는데, SLA 변수와 무관한 3명은 분석대상에 포함시키지 않았다. 그리고 1,000라인 이하의 프로그램을 담당하거나 10개 미만의 프로그램을 담당하는 6명은 이상치(outlier)라고 판단하여 분석 대상에서 제외되었다. 그러므로 총 64명의 프로그래머를 대상으로 분석하였다.

### 3.5 분석 방법

연구모형에서 제시한 소프트웨어 유지보수 비용 산정 모형이 실증적으로 유의하기 위해서는 SLA 변수와 다른 변수의 조합으로 유지보수 비용을 설명할 수 있어야 한다. 그래서 먼저 유지보수 비용에 영향을 미치는 독립변수를 선별하기 위해 상관분석(correlation analysis)과 요인분석(factor analysis)을 실시하여 3개의 선형 회귀 모형(linear regression model)을 설정하였다. 모형 1은 요인분석을 통해 선정된 각 요인에서 종속변수와 상관관계가 가장 높은 변수들만으로 구성되었다. 모형 2는 각 요인의 요인 값(factor score)

을 독립 변수로 하여 구성되었으며, 모형 3는 가중 평균한 SLA 변수와 종속변수와 상관관계가 높은 다른 변수들로 구성되었다.

## IV. 연구 결과

### 4.1 상관분석

종속변수와 독립변수, 그리고 독립변수들간의 상관관계를 정리하면 <표 5>와 같다. 연구모형에서 예상한 바와 같이, 프로그램 크기와 관련된 변수들은 종속변수와 상관관계를 가지고 있으며, SLA의 긴급도와 사용빈도, 그리고 프로그램 기

능 가운데 배치프로그램들도 종속변수와 높은 상관관계를 가지고 있다.

이때 긴급도 A보다 긴급도 B나 C가 종속변수인 유지보수 비용과 상관관계가 높은 것으로 나타났다. 인터뷰 결과 긴급도가 높은 애플리케이션일 경우 기본적인 기능을 수행하는 프로그램이면서도 한번 장애가 오면 전사적으로 심각한 피해를 가져올 수 있기 변경 요청이 있어도 쉽게 수정이나 변경을 할 수 없는 애플리케이션들로 밝혀졌다. 오히려 긴급도가 낮은 프로그램들이 더 유지보수의 요청이 많고 이들을 긴급하게 수행해야 하는 경우는 많기 때문에 유지보수 비용과 양의 상관 관계를 갖는 것을 확인할 수 있다.

<표 5> 상관분석 결과

	SCALED_HRS	KLOC_TOT	PROC_NO	INNER_CO	OUTTER_CO	URG_AL	URG_BL	URG_CL	FREQ_AL	FREQ_BL	FREQ_CL	BATCH_L	CLIENT_L
SCALED_HRS	1.00												
KLOC_TOT	-0.23*	1.00											
PROC_NO	0.21*	0.55***	1.00										
INNER_CO	0.07	0.20	0.11	1.00									
OUTTER_CO	-0.05	-0.03	-0.09	0.26***	1.00								
URG_AL	0.03	-0.02	-0.18	-0.06	0.31**	1.00							
URG_BL	0.52***	-0.10	-0.24*	0.15	-0.17	-0.26**	1.00						
URG_CL	-0.54***	-0.09	0.29***	-0.14	0.10	0.02	-0.97***	1.00					
FREQ_AL	0.02	-0.03	-0.04	0.16	0.40***	0.37***	-0.10	0.01	1.00				
FREQ_BL	0.25**	0.09	-0.16	-0.07	-0.01	0.14	0.38**	-0.43***	-0.04	1.00			
FREQ_CL	-0.23*	-0.07	0.16	-0.02	-0.20	-0.31**	-0.29**	0.37***	-0.48***	-0.86***	1.00		
BATCH_L	0.42***	-0.14	-0.36***	-0.27**	-0.15	-0.26**	0.76***	-0.73***	-0.18	0.47***	-0.32***	1.00	
CLIENT_L	-0.02	0.22*	-0.05	0.37***	0.53***	0.15	-0.24*	0.21*	0.20	-0.19	0.07	-0.36***	1.00

주) \*는  $p < 0.10$ , \*\*는  $p < 0.05$ , 그리고 \*\*\*는  $p < 0.01$ 이다.

본 연구를 위해 소프트웨어 유지보수 비용(SCALED\_HRS)과 유의 수준 10%에서 유의한 7개의 변수를 최종 독립 변수로 추출하였다. 선택된 독립변수는 프로그램의 KLOC와 담당하고 있는 프로그램의 수, 긴급도 B의 비중과 C의 비중, 사용빈도 B의 비중과 C의 비중, 그리고 배치프로그램의 비중이다.

### 4.2 요인분석

요인분석에서는 상관분석으로 선택된 최종 독립 변수들간 상관관계를 바탕으로 공통요인을 추출하였다. 요인분석을 하게 되면 한 변수는 동일한 요인에 속한 독립 변수들과는 상관관계가 높고 다른 요인의 변수들과는 상관관계가 매우 낮아서 다중공선성을 고려하지 않아도 되는 이점이 있다. 요인분석을 위한 통계패키지 프로그램으로 SAS를 이용하였으며, 추출기법으로 주성분분석(principal components analysis)을 사용하였고, 회전(rotation)방법은 베리맥스(verimax)로 하였다.

<표 6> 요인분석 결과

독립변수	요인 1 (MOD_URG)	요인 2 (LOW_FREQ)	요인 3 (SIZE)
URG_BL	0.97*	-0.13	0.02
BATCH_L	0.82*	-0.24	-0.24
URG_CL	-0.95*	0.21	0.01
FREQ_CL	-0.15	0.95*	0.02
FREQ_BL	0.27	-0.93*	0.00
KLOC_TOT	0.10	-0.09	0.91*
PROC_NO	-0.26	0.12	0.84*

<표 6>에서는 보는 바와 같이 요인분석을 통해 모두 3개의 공통요인이 추출되었고, 각 요인의 특징은 다음과 같다. 요인 1은 긴급도 B의 특성이 높고 긴급도 C의 특성이 낮으며 배치 형태

의 프로그램의 특성을 나타낸다. 배치 프로그램의 대표적인 예로는 월말의 빌링(billing) 작업을 위한 프로그램이 있는데 빌링 업무는 주기적인 일정 시간에 신속하게 마무리해야 하기 때문에 중간 등급의 긴급도를 필요로 하는 작업으로 볼 수 있다. 이러한 특성으로 인해 요인 1은 짧은 시간에 긴급도 B의 업무를 담당 해야 하는 배치 프로그램을 설명하는 요인이다.

요인 2는 프로그램의 사용 빈도와 관련된 요인으로서 사용빈도 C와는 양의 상관관계를 가지고 있으나 사용빈도 B와는 음의 상관관계를 가지고 있다. 즉, 요인 2는 사용빈도가 낮은 프로그램의 특성을 나타내는 요인이다.

요인 3은 프로그램 크기를 나타내는 프로그램 KLOC와 프로그램의 수와 모두 양의 상관관계를 가지고 있다. 따라서 요인 3은 프로그램 크기의 요인이라고 할 수 있다.

### 4.3 선형 회귀 모형

이 절에서는 유지보수 대상 프로그램의 특성과 SLA 변수들이 유지보수 비용에 영향을 미치는지 확인하기 위해 여러 회귀 모형식을 구성하였다.

#### 4.3.1 모형 1

모형 1은 요인분석의 결과 도출된 3개의 요인, 즉 요인 1(MOD\_URG)과 요인 2(LOW\_FREQ), 요인 3(SIZE)에서 대표 변수를 각각 1개씩 추출하여 구성하였다. 그래서 아래 식 (8)과 같이 회귀 모형식을 구성하였고, 이 모형식을 바탕으로 타당성과 각 상관계수의 유의성을 분석하여, <표 7>과 같은 결과를 얻었다.

$$SCALED\_HRS = \beta_0 + \beta_1 (KLOC\_TOT) + \beta_2 (URG\_BL) + \beta_3 (FREQ\_CL) \quad (8)$$

<표 7> 모형 1에 의한 회귀분석 결과

종속 변수: SCALED_HRS 방법: 최소 제곱법 관측수: 64				
변수	상관계수	표준 오차	t-통계량	확률
Intercept	194.85565	11.29357	17.25	<.0001
KLOC_TOT	0.00007408	0.00004918	1.51	0.1373
URG_BL	34.88668	8.16312	4.27	<.0001
FREQ_CL	-9.50591	12.93376	-.73	0.4652
R-squared	0.3049	종속변수 평균		214.40625
Adjusted R-squared	0.2702	F-통계량		8.77
Durbin-Watson 통계량	2.577	확률(F-통계량)		<.0001

<표 8> 모형 2에 의한 분석 결과

종속 변수: SCALED_HRS 방법: 최소 제곱법 관측수: 64				
변수	상관계수	표준 오차	t-통계량	확률
Intercept	214.40625	2.56237	83.68	<.0001
MOD_URG	12.79392	2.58262	4.95	<.0001
LOW_FREQ	-3.50725	2.58262	-1.36	0.1795
SIZE	1.28462	2.58262	0.50	0.6207
R-squared	0.3074	종속변수 평균		214.40625
Adjusted R-squared	0.2728	F-통계량		8.88
Durbin-Watson 통계량	2.559	확률(F-통계량)		<.0001

분석 결과로부터 알 수 있는 것은 다음과 같다. SLA의 긴급도 B비중은 99% 신뢰구간에서 유의하지만, 프로그램 크기와 사용빈도는 유의하지 않다. 즉, SLA의 긴급도가 B인 프로그램이 증가할수록 유지보수 비용이 증가하는 것을 알 수 있다. 따라서 아웃소싱 벤더가 소프트웨어 유지보수 서비스 비용을 산정할 때 긴급도 B의 비중을 기준으로 산정하는 것도 한 방법이 될 수 있다.

### 4.3.2 모형 2

모형 2는 앞의 요인분석에서 도출된 요인을 변

수로 간주하여 회귀분석을 실시한 모형이다. 각 요인은 아래의 식 (9)에서 보는 바와 같이, 중간 정도의 긴급도와 낮은 사용 빈도, 그리고 프로그램 크기가 변수로 사용되었다.

$$SCALED\_HRS = \beta_0 + \beta_1 (MOD\_URG) + \beta_2 (LOW\_FREQ) + \beta_3 (SIZE) \quad (9)$$

<표 8>에서 보는 바와 같이, 중간 정도의 긴급도 변수는 99% 신뢰구간에서 유의하지만 다른 요인들은 동일한 신뢰구간에서 유의하지 않다. 소프트웨어 아웃소싱의 유지보수 비용은 중간 정도의 긴급도를 가진 프로그램에 영향을 받는

것을 알 수 있고, 보다 구체적으로 긴급도 B의 비중과 배치 형태의 프로그램의 비중이 높을수록 긴급도 C의 비중이 낮을수록 유지보수 비용이 높아지는 것을 확인할 수 있다.

### 4.3.3 모형 3

회귀 모형 3은 상관분석 이전에 고려한 모든 변수들(프로그램의 크기와 프로그램 기능, SLA 변수)을 이용하여 구성하였다. SLA 변수들 등급 간의 특성을 반영하기 위해 SLA 변수들에(긴급도와 사용 빈도) 등급에 따른 조정 인자를 두고 각 변수를 가중 평균하여 새로운 독립변수  $FREQ\_L$ 과  $URG\_L$ 을 만들었다. 아래의 식 (10)과 식 (11)에서와 같이 등급 B를 1.0으로 기준으로 삼고 등급 A와 등급 C에 각각 1.5와 0.5의 가중치를 부여하였다.

$$FREQ\_L = 1.5 * FREQ\_AL + 1.0 * FREQ\_BL + 0.5 * FREQ\_CL \quad (10)$$

$$URG\_L = 1.5 * URG\_AL + 1.0 * URG\_BL + 0.5 * URG\_CL \quad (11)$$

위에서 설명한 변수를 바탕으로 식 (12)와 같은 회귀식 구성하여 회귀분석을 실시하였으며, 그 결과는 <표 9>와 같다.

$$SCALED\_HRS = \beta_0 + \beta_1 (KLOC\_TOT) + \beta_2 (PROC\_NO) + \beta_3 (URG\_L) + \beta_4 (FREQ\_L) + \beta_5 (BATCH\_L) \quad (12)$$

회귀 모형 3이 회귀 모형 1과 모형 2보다 높은 설명력(R-squared)을 보이기 때문에 소프트웨어 유지보수 비용을 보다 잘 설명하는 변수들로 구성되어 있다고 볼 수 있다. 구체적으로 살펴 보면, 프로그램 크기와 긴급도는 95% 신뢰구간에서 유의하다. 여기서, 프로그램의 크기가 일반적으로 10,000 줄을 넘기 때문에, 프로그램 크기의 계수가 매우 적은 것을 확인할 수 있다. 하지만 프로그램의 빈도와 프로그램의 기능은 유의하지 않음을 알 수 있다.

모형 3의 결과는 유지보수 대상 프로그램의 크기가 유지보수 비용에 양의 상관 관계를 갖는다는 선행연구 결과[Lientz and Swanson, 1980; Kemerer and Slaughter, 1997]와 일치한다. 더욱이 SLA의 긴급도 또한 유지보수 비용에 양의 영향을 미치는 중요한 요인으로 작용하고 있다. 이를 통해 프로그램 크기가 클수록, 긴급도가 높은 프로그램이 많을수록, 유지보수 비용이 증가하는 것을 알 수 있다. 이는 소프트웨어 유지보수 비용을 산정할 때, 프로그램 크기 뿐만 아니라 고객과 합의한 SLA 항목도 원가 동인으로 반영해야 한다는 것을 의미한다.

<표 9> 모형 3에 의한 회귀분석 결과

종속 변수: SCALED_HRS 방법: 최소 제곱법 관측수: 64				
변수	상관계수	표준 오차	t-통계량	확률
Intercept	168.08303	19.21065	8.75	<.0001
KLOC_TOT	0.00016201	0.00006089	2.66	0.0101
PROC_NO	-0.05003	0.02665	-1.88	0.0655
URG_L	45.87663	22.54189	2.04	0.0464
FREQ_L	-2.30588	19.48184	-0.12	0.9062
BATCH_L	11.22271	8.86615	1.27	0.2106
R-squared	0.3782	종속변수 평균		214.40625
Adjusted R-squared	0.3246	F-통계량		7.05
Durbin-Watson 통계량	2.670	확률(F-통계량)		<.0001

## V. 결 론

지금까지 정보시스템 아웃소싱 서비스의 비용은 프로그램 개발 비용 또는 유지보수 노력을 바탕으로 책정되어 왔으나 최근 아웃소싱 서비스에서 SLA가 활발하게 도입되면서 아웃소싱 비용 산정에서도 SLA에 포함된 내용을 반영해야 할 필요성이 대두되었다. 본 연구에서는 이러한 필요성을 바탕으로 연구모형을 제안하였으며 실증적으로 분석하고 있다.

본 연구에서는 소프트웨어 유지보수 비용은 프로그램의 크기뿐만 아니라 SLA 변수를 기반으로 책정되어야 한다는 사실을 실증적으로 보여주고 있다. 특히 SLA 변수 가운데 긴급도가 유지보수 비용을 가장 잘 반영하고 있으며, 긴급도의 등급마다 가중치를 두어 합산하면 유지보수 비용을 보다 잘 설명한다.

한편 본 논문은 소프트웨어 아웃소싱에 국한하여 실증적으로 검증하였으나 다른 정보시스템 아웃소싱으로도 확장하여 적용할 수 있을 것으로 기대한다. 인력비가 대부분의 비용을 차지하는 소프트웨어 유지보수 아웃소싱 뿐만 아니라 다른 정보시스템 아웃소싱 서비스에 합리적인 대가 산정을 위해서는 SLA에 있는 요구수준에 따른 투입 노동시간 증감을 기준으로 가격을 산

정하는 것이 합리적일 수 있다는 가능성을 보여준다고 할 수 있다. 본 연구는 첫째로 소프트웨어 유지보수 아웃소싱의 비용에 SLA가 영향을 줄 수 있다는 가설을 실제 데이터를 통해 실증분석했다는 점에서 의의가 있다고 할 수 있다. 특히 실제로 SLA에 포함되는 요인들을 소프트웨어 엔지니어링 연구에서 프로그래머들의 작업시간에 영향을 주는 요인들과 연관 지어 묶어냄으로써 SLA가 대가산정에 어떤 식으로 반영되어야 할지를 찾아낸 연구라고 볼 수 있다. 문헌 연구를 통해 애플리케이션 소프트웨어 유지보수에서 비용을 산정하는 모형을 기초하였으며, 실제 현장에 있는 업무추진팀과 인터뷰 및 논의를 바탕으로 모형을 구성하였기 때문에 현실에 적용하기에 적합한 모형이라는 점도 의의로 지적될 수 있다. 또한, 분석의 단위(unit of analysis)가 개별 프로그래머이므로 아웃소싱 벤더의 매니저가 프로그래머들에게 작업을 할당할 때도 사용할 수 있을 것이다.

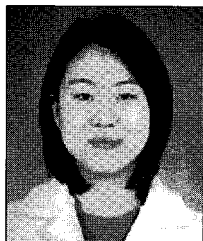
본 연구의 한계로 지적될 수 있는 것은 부족한 외적 타당성이다. 본 연구는 하나의 기업을 대상으로 하고 있기 때문에 다른 기업에도 동일하게 적용할 수 있다는 보장은 없다. 따라서 향후 연구는 여러 기업을 대상으로 데이터를 수집하여 앞서 제안한 모형을 보다 일반화할 수 있는지 검증하고자 한다.

## 〈참 고 문 헌〉

- [1] 남기찬, 이재남, *정보시스템 아웃소싱: 방법론과 사례*, 이진출판사, 1999.
- [2] 정보통신부, *용역 유지보수 대가 산정기준*, 2004.
- [3] 한국산업연구원, *서비스수준협정 적용에 있어서의 성공요인에 관한 연구*, 2002.
- [4] 한국전산원, *정보시스템 원가 산정과 관련된 요소 및 효과에 관한 연구*, 2000.
- [5] 한국전산원, *정보시스템 운영대가 기준개발*, 2002.
- [6] Aucoin, P., *Internalizing the Vendor's Resources: Outsourcing in the 1990s*, Chantico Publishing Co. Inc., 1991.
- [7] Banker, R., Datar, S., and Kemerer, C., "A Model to Evaluate Variables Impacting the Productivity of Software Maintenance Projects," *Management Science*, Vol. 37, No. 1, 1991, pp. 1-18.
- [8] Banker, R., Datar, S., Kemerer, C., and Zweig, D., "Software Complexity and

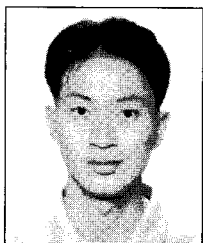
- Maintenance Costs," *Communications of the ACM*, Vol. 36, No. 11, 1993, pp. 81-94.
- [9] Banker, R., Davis, G., and Slaughter, S., "Software Development Practices, Software Complexity, and Maintenance Performance: A Field Study," *Management Science*, Vol. 44, No. 4, 1997, pp. 433-450.
- [10] Bohem, B., "Software Risk Management: Principles and Practices," *IEEE Software*, Vol. 8, No. 1, 1991, pp. 32-41.
- [11] Bryson, N. and Ngwenyama, O., "Structuring IS Outsourcing Contracts for Mutual Gain: An Approach to Analyzing Performance Incentive Schemes," *Journal of the Association for Information System*, Vol. 1, No. 9, 2000, pp. 9-50.
- [12] Demarco, T., *The estimating dilemma, Controlling Software Projects: Management Measurement and Estimation*, Yourdon Press, 1982.
- [13] Grover, V., Cheon, M., and Teng, J., "A Descriptive Study on the Outsourcing of Information Systems Functions," *Information & Management*, Vol. 27, No. 1, 1994, pp. 33-44.
- [14] Kemerer, C. and Slaughter, S., "Determinants of Software Maintenance Profiles: An Empirical Investigation," *Software Maintenance: research and practice*, Vol. 9, No. 4, 1997, pp. 235-251.
- [15] Lacity, M., Willcocks, L., and Feeny, D., "The Value of Selective IT Outsourcing," *Sloan Management Review*, Vol. 37, No. 3, 1996, pp. 13-25.
- [16] Lientz, B. and Swanson, E., *Software Maintenance Management*, Addison-Wesley, 1980.
- [17] Maurer, W., "Service-Level Agreements are Key to Outsourcing Success," *Gartner Group*, 2000.
- [18] Marriott, R., "Make Gain Sharing Part of IT Services Agreements," *Gartner Research*, 2003.
- [19] Palvia, P.C., "A Dialectic View of Information Systems Outsourcing: Pros and Cons," *Information & Management*, Vol. 29, No. 5, 1995, pp. 265-275
- [20] Swanson, E.B., "Maintaining IS Quality," *Information and Software Technology*, Vol. 39, No. 12, 1997, pp. 845-850.
- [21] Schneider, B. and Bowen, D.E., "The Service Organization: Human Resource Management is Crucial," *Organizational Dynamics*, Vol. 21, No. 4, 1993, pp. 39-52.

## ◆ 저자소개 ◆



유지나 (Yoo, Jee Na)

현재 IBM Business Consulting Services의 Strategy & Change Service Consultant로 활동 중이다. 포항공과대학교에서 신소재공학사를, 한국과학기술원 테크노경영대학원 경영공학석사를 취득하였다. 주요 관심분야는 IT 및 하이테크산업에서의 가격정책, Business value from technology, 마케팅 전략 등이다.



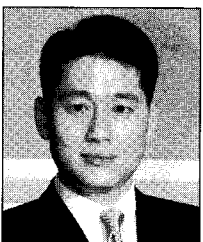
김병수 (Kim, Byoung Soo)

현재 한국과학기술원 테크노경영대학원 박사과정에 재학 중이다. 고려대학교 전자 및 전파 공학사를 취득하였고, 한국과학기술원 전자공학석사를 취득하였다. 주요 관심분야는 4세대 이동통신 수요예측과 IT와 미디어 산업에서 수익 분배와 가격 정책 등이다.



최민석 (Choi, Minn Seok)

현재 한국전자통신연구원 소속되어 있으며 한국과학기술원 테크노경영대학원 박사과정에 재학 중이다. 연세대학교 산업시스템공학과에서 공학사와 한국과학기술원 산업공학과에서 공학석사를 취득하였다. 주요 관심분야는 통신서비스 및 방송 시장의 정책 및 전략, 정보통신 사회의 트렌드 분석 및 미래 예측 등이다.



오정석 (Oh, Jungsuk)

MIT대 Management Science학과에서 학사를 마쳤으며, Stanford University 대학원에서 Operations Research로 석사학위를 취득하였고 동 대학원 Management Science & Engineering학과에서 박사학위를 취득하였다. 현재 한국과학기술원 테크노경영대학원 조교수로 재직 중이며 주요 연구 관심분야는 텔레콤/디지털/미디어 경제학, 디지털 마케팅, 분배 이론 등이 있다.

◆ 이 논문은 2006년 2월 16일 접수하여 1차 수정을 거쳐 2006년 6월 5일 게재확정되었습니다.