



COCOMO 모델 중심의 소프트웨어 프로젝트 평가에 관한 연구

- A Study on the COCOMO Oriented Software
Project Evaluation -

광운대학교산업정보대학원

전자계산학전공

최 호 영

COCOMO 모델 중심의 소프트웨어 프로젝트 평가에 관한 연구

- A Study on the COCOMO Oriented Software
Project Evaluation -

지도 : 장 덕 철 교수

이 논문을 이학석사학위 논문으로 제출함.

1990년 6 월 일


광운대학교산업정보대학원


전자계산학전공


최 호 영



최호영의 이학석사 학위논문을 인준함.

심사위원장 장택진 

심사위원 김경태 

심사위원 조국현 

광운대학교산업정보대학원

1990년 6월 일

감사의 글

오늘의 결실을 맺도록 사랑과 희생으로 보살피 주신 부모님께 감사와 함께 이 논문을 바칩니다.

본 학위 논문을 위하여 아낌없는 지도와 격려로 학문의 길을 이끌어 주신 장덕철 지도교수님께 무한한 감사를 드립니다.

논문 심사에 지도와 충고를 하여 주신 김경태 교수님과 조국현 교수님께 감사를 드립니다. 여러면에서 도움을 주시고 격려를 해주신 유황빈 학과장님께도 진심으로 감사를 드리며, 평소 가르침과 지도를 주신 심재홍 교수님, 이동호 교수님, 최영근 교수님께 감사를 드립니다.

또한 본 논문 작성에 많은 도움을 준 직장 선후배 여러분의 고마움을 잊지 못할 것이며 특히, 논문 정리에 도움을 준 이준동, 이용생 씨에게 감사를 드립니다.

그리고 오늘이 있기까지 어려울 때마다 많은 힘이 되어 주었고 항상 마음을 써주신 쌍문동의 장인 장모님, 모든 식구들, 친구들에게 감사를 드립니다.

끝으로 이 작은 결실의 기쁨을 사랑하는 은지,은호, 은지 어머니와 함께 나누며 그동안의 어려움에 답하고자 합니다.

1990 년 6 월 15 일

최 호 영 올림

요 약

본 논문에서는 COCOMO 모델 중심의 소프트웨어 프로젝트 평가를 위한 환경 특성값 산출과 적용 방법을 제안한다.

환경 특성값은 비 영리 조직에서 관리적 기능을 수행하는 자료 처리 개발 프로젝트 중 임의의 8개를 선정하여 COCOMO 모델에 투입된 인력과 개발기간을 구하고, 실제적인 인력과 개발기간을 비교하여 산출한다.

그리고 환경 특성값을 적용한 소프트웨어 평가 방법의 타당성 검증을 위하여 COCOMO, 실제, 환경 특성값을 적용한 프로젝트의 인력과 개발기간을 종합 분석하고 그 결과표를 제시한다.



ABSTRACT

This paper suggest the COCOMO oriented software project evaluation methodology, which produce an environment of characteristic metric.

The environment of characteristic metric, is computed from man-month and development time from the eight projects among business information system development projects which perform management functions only in the nonproject organization.

Here with it provide the man-month and project time analysis table for the purpose of verification and acceptance to the suggested methodology from the real project applying COCOMO model and its environment characteristic metric.





차 례

국문 요약

영문 요약

I. 서론	1
II. 소프트웨어 비용산정	3
2.1 개요	3
2.2 소프트웨어 비용 결정 요소	3
2.2.1 프로그래머 자질과 가용 시간	3
2.2.2 복잡도와 크기	5
2.2.3 신뢰도와 기술 수준	5
2.3 소프트웨어 개발비용 예측 절차	6
2.4 소프트웨어 비용 산출 기법	8
2.4.1. 전문가 감정	8
2.4.2. 델파이(Delphi)식 비용산정	9
2.4.3. 업무 분류 구조	10
2.4.4. 연산 방식의 비용 모델	12
2.5 추정 모델	13
III. COCOMO 모델	16
3.1 COCOMO 모델 개요	16



3.2 COCOMO 모델의 용어와 가정	16
3.3 COCOMO 모델의 분류	17
3.3.1 Basic COCOMO	17
3.3.2 Intermediate COCOMO	19
3.3.3 Detailed COCOMO	20
3.4 COCOMO 특성값	20
IV. 소프트웨어 평가상의 COCOMO 모델 적용	22
4.1 적용 대상 프로젝트	22
4.2 환경 특성값	23
4.2.1 인 력	24
4.2.2 개발기간	25
4.2.3 환경 특성값 산출	26
V. 적용 결과	27
5.1 COCOMO와 실제 프로젝트의 비교	27
5.2 환경 특성값 적용 모델과의 비교	28
5.3 종합 분석	29
VI. 결 론	30
● 참 고 문 헌	31



그림 차례

그림 1. 소프트웨어 개발비용 분석 흐름도	7
그림 2. WBS 도표	11

표 차례

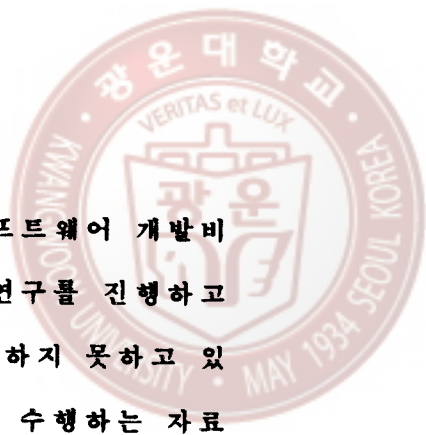
표 1. 프로그래머의 생산성 비교	4
표 2. 신뢰도에 대한 노력 승수	6
표 3. Basic COCOMO의 인력과 개발기간 방정식	18
표 4. Intermediate COCOMO의 인력과 개발기간 방정식	19
표 5. COCOMO 특성표	21
표 6. 수집 자료	22
표 7. 인력 비교표	24
표 8. 개발기간 비교표	25
표 9. COCOMO와 실제 프로젝트 비교표	27
표 10. 환경 특성값 적용 비교표	28
표 11. 종합 분석 결과표	29

I . 서 론

소프트웨어 프로젝트의 개발비용 산정 모델로서 1965년 SDC (system development corporation) 회사에서 Nelson에 의해 개발된 SDC 모델이 발표되었는데 이 모델은 소프트웨어 프로젝트 개발에 미치는 특성들에 대하여 조사한 것으로서 최초의 모델이라는데 의미가 있다.

또한 SDC 모델 이후 많은 비용산정 모델들이 발표되었는데 이 모델들을 살펴보면 TWR(Wolverton,1974), IBM-FSD(Walston,1977), Doty(Herd,1977), Boeing(Black,1977), SLIM(Putnam,1978), RCA PRICES(Freiman,1979), GRC(Carriere,Thibodeau,1979), Bailey-Basili Meta(Bailey,Basili,1981)와 COCOMO(Boehm,1981) 등이 있다. 각 모델들에서 소프트웨어 프로젝트 개발비용 산정시 비용 유발 요인이 가장 중요한 요인으로 나타났다. 그러나 각 모델들이 고려한 비용 요인들이 완전한 것이 아니기 때문에 실제 프로젝트에 적용할 경우 부정확한 모델이 되기 쉬웠다.[11,22]

J.R Distraso[3]에 의하면 소프트웨어 비용 측정은 "아무리 잘되어도 부정확한 기술이다(software cost estimating is, at best, an imprecise art)." 라고 표현하였는데 소프트웨어 개발비용이 GNP에 차지하는 비중이 증대하고 소프트웨어가 사회에 미치는 영향이 빠른 속도로 증가하고 있어 이에 대한 지속적인 연구와 개선된 비용측정 모델 개발의 필요성이 크게 대두되었다.



이러한 필요성에 따라 영리 조직인 기업들은 소프트웨어 개발비용 측정과 평가에 대한 중요성을 인식하여 꾸준한 연구를 진행하고 있으나 비 영리 조직에서는 이의 중요성을 거의 인식하지 못하고 있는 실정이다. 따라서 비 영리 조직의 관리적 기능을 수행하는 자료처리 개발 프로젝트를 대상으로 COCOMO 모델에 적용하여 인력과 개발기간을 구하고, 실제 투입된 인력과 개발기간을 비교하여 환경 특성값을 산출한다.

환경 특성값 산출의 대상이 되는 COCOMO 모델은 1981년 Barry W. Boehm에 의해 개발된 것으로서 생명주기의 진행에 따라 Basic COCOMO, Intermediate COCOMO와 Detailed COCOMO 모델로 나누어지고 프로젝트의 유형에 따라 또 다른 관계함수 Organic, Semidetached와 Embedded mode가 주어지며 기간 분석도 생명주기의 과정마다 주어지는 가장 널리 통용되고 가장 좋은 모델로 평가되고 있다.[1,11,16]

본 논문에서는 COCOMO 모델과 환경 특성값을 대입한 COCOMO 모델을 상호 비교 평가하여 문제점 및 개선점을 제안하고 소프트웨어 평가방법의 타당성 검증을 위하여 COCOMO, 실제, 환경 특성값 적용 프로젝트의 인력과 개발기간을 종합 분석한 결과표를 제시한다.

본 논문의 구성은 제 II장에서 소프트웨어 비용 산정을 기술하고 제 III장에서는 COCOMO 모델에 대하여 설명하며 제 IV장에서는 소프트웨어 평가상의 COCOMO 모델 적용을 제시하였으며 제 V장에서는 적용결과를 종합 분석하여 제 VI장에서는 결론을 맺는다.



Ⅱ . 소프트웨어 비용산정

2.1 개 요

소프트웨어가 대형화되고 복잡 다양해짐에 따라 개발 비용을 산출한다는 것은 매우 어려운 일이다. 소프트웨어 개발 계획과 분석 단계에서 사용자 요구의 제반 특성 문제 즉, 적용성(applicability), 전달성(communicability), 완전성(completeness), 이해성(comprehensibility), 정확성(correctness), 일관성(consistency), 작성용이성(constructibility), 설계와의 독립성(design freedom), 확장성(extensibility), 실현가능성(feasibility), 형식성(formality), 최소성(minimality), 필요성(necessity), 추적가능성(traceability), 일의성(Unambiguity) 등의 많은 요소가 산재해 있기 때문이다. 따라서 이러한 특성 문제들을 소프트웨어 개발시 비용산정에 적용 시키는 방법이 필요하게 되었다.[15]

2.2 소프트웨어 비용 결정 요소

2.2.1 프로그래머 자질과 가용 시간

소프트웨어의 품질과 프로그래머의 생산성은 소프트웨어 제품을 개발하고 유지 보수하는데 사용되는 공정을 개선함으로써 향상시킬 수 있으며, 프로그래머의 개인적인 능력은 Sackman이 실험한 결과 일반적인 수준의 적합성에 대해 프로그래머의 생산성을 일괄처리와



Ⅱ . 소프트웨어 비용산정

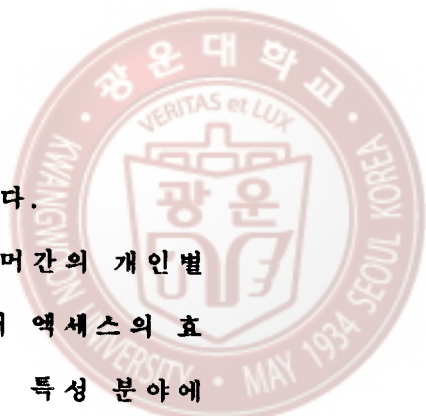
2.1 개 요

소프트웨어가 대형화되고 복잡 다양해짐에 따라 개발 비용을 산출한다는 것은 매우 어려운 일이다. 소프트웨어 개발 계획과 분석 단계에서 사용자 요구의 제반 특성 문제 즉, 적용성(applicability), 전달성(communicability), 완전성(completeness), 이해성(comprehensibility), 정확성(correctness), 일관성(consistency), 작성용이성(constructibility), 설계와의 독립성(design freedom), 확장성(extensibility), 실현가능성(feasibility), 형식성(formality), 최소성(minimality), 필요성(necessity), 추적가능성(traceability), 일의성(Unambiguity) 등의 많은 요소가 산재해 있기 때문이다. 따라서 이러한 특성 문제들을 소프트웨어 개발시 비용산정에 적용 시키는 방법이 필요하게 되었다.[15]

2.2 소프트웨어 비용 결정 요소

2.2.1 프로그래머 자질과 가용 시간

소프트웨어의 품질과 프로그래머의 생산성은 소프트웨어 제품을 개발하고 유지 보수하는데 사용되는 공정을 개선함으로써 향상시킬 수 있으며, 프로그래머의 개인적인 능력은 Sackman이 실험한 결과 일반적인 수준의 적합성에 대해 프로그래머의 생산성을 일괄처리와



대화식 프로그래밍의 효과를 산정하는데 목표를 두었다.

이 실험 결과는 표 1 에 나타난 것처럼, 프로그래머간의 개인별 생산성 차이는 일괄처리 방식과 대화식에 의한 컴퓨터 액세스의 효과보다도 훨씬 컸으며, 개인의 일반적인 적합성과 특성 분야에 대한 친숙성 등에 의하여 민감한 반응이 나타나고 있으며 이러한 개인적인 능력은 소프트웨어 품질과 생산성 향상에 중요한 요소가 됨을 알 수 있다.

표 1. 프로그래머의 생산성 비교
Table 1. Comparison of programmer's productivity

능 력 측 정 요 소	비 율	
	프로그램 1	프로그램 2
· 디 버깅 시 간	28 : 1	26 : 1
· C P U 시 간	8 : 1	11 : 1
· 코 딩 시 간	16 : 1	25 : 1
· 프로그램 규모	6 : 1	5 : 1
· 실행 시 간	5 : 1	13 : 1

또한 소프트웨어 개발 기간이 최적의 기간보다 단축되거나 늘어난다면 보다 많은 노력이 필요하게 되는데, 이러한 특징은 Putnam 곡선에 잘 나타나며 소프트웨어 개발 노력은 개발 비용의 4배 곱에 반비례 한다.[15,18]



2.2.2 복잡도와 크기

프로젝트의 복잡도는 응용 프로그램, 유틸리티 프로그램과 시스템 프로그램으로 나눈다.

응용 프로그램은 포트란, 파스칼 그리고 코볼과 같은 고급 언어로 작성된 과학 계산·자료 처리용 루틴등을 말하며, 유틸리티 프로그램은 컴파일러, 어셈블러, 링커지 에디터와 로더 등으로서 파스칼 혹은 Ada와 같은 고급 언어로 작성되거나 어셈블리 언어로도 작성될 수 있다.

시스템 프로그램은 데이터 통신 패키지, 실시간 공정 제어 시스템 그리고 운영 체제 루틴 등이 있으며, 흔히 어셈블리 언어로 작성되거나 PL/I 혹은 Ada와 같은 고급 언어로 작성 되기도 한다.

프로젝트 복잡도의 함수로서 1일당 프로그래머가 작성할 수 있는 라인 수는 시스템 프로그램의 경우 1라인 미만, 유틸리티 프로그램은 5에서 10라인, 응용 프로그램은 25에서 100라인 정도로 작성된다.[15,18]

2.2.3 신뢰도와 기술 수준

Boehm은 신뢰도의 범주를 다섯 가지로 분류하고 각 범주에 대한 노력 승수는 표 2 와 같으며, 승수는 0.75에서 1.40까지의 신뢰도가 있으므로 노력비는 $1.87(1.4/0.75)$ 이다.[15]

표 2. 신뢰도에 대한 노력 승수
Table 2. Effort multipliers for reliability

범 주	실 패 효 과	노 력 승 수
. 매우 낮 음	. 약간 불편	0.75
. 낮 음	. 쉽게 회복	0.88
. 보통	. 손실의 복구가 어려움	1.00
. 높 음	. 높은 재정 손실 위기	1.15
. 매우 높 음	. 대단한 위험	1.40

또한 Boehm은 기술 수준에 관하여 프로그래밍 실재에 대한 노력 승수는 0.82에서 1.24까지, 소프트웨어 도구에 대한 노력 승수는 0.83에서 1.24까지의 영역으로 제안하였다. 이러한 예로서 본다면 어떠한 프로그래밍 기법을 사용했느냐에 따라 0.67배의 프로그래밍 노력을 감소시킬 수 있으며, 개발 도구의 사용에 따라 0.45배로 개발 노력을 줄일 수 있다.[1,15]

2.3 소프트웨어 개발비용 예측 절차

일반적인 소프트웨어 개발비용 예측 모델들은 그림 1 과 같은 절차를 따른다. 이를 단계별로 기술하면 아래와 같다.[19,24]

- (1) 소프트웨어 프로젝트 특성 분석, 크기 예측, 비용요소를 고려한다.
- (2) 고려할 비용요소가 없을 경우에는 단계 (4)를 실행하고 고려할 비용요소가 있는 경우에는 단계 (3)을 실행한다.
- (3) 비용요소를 고려한 소프트웨어 규모를 인력 노력으로 변환한다.
- (4) 개발노력과 개발기간을 일정 계획에 따라 단계적으로 분배한다.
- (5) 계획단계, 유지보수단계의 비용을 고려한다.

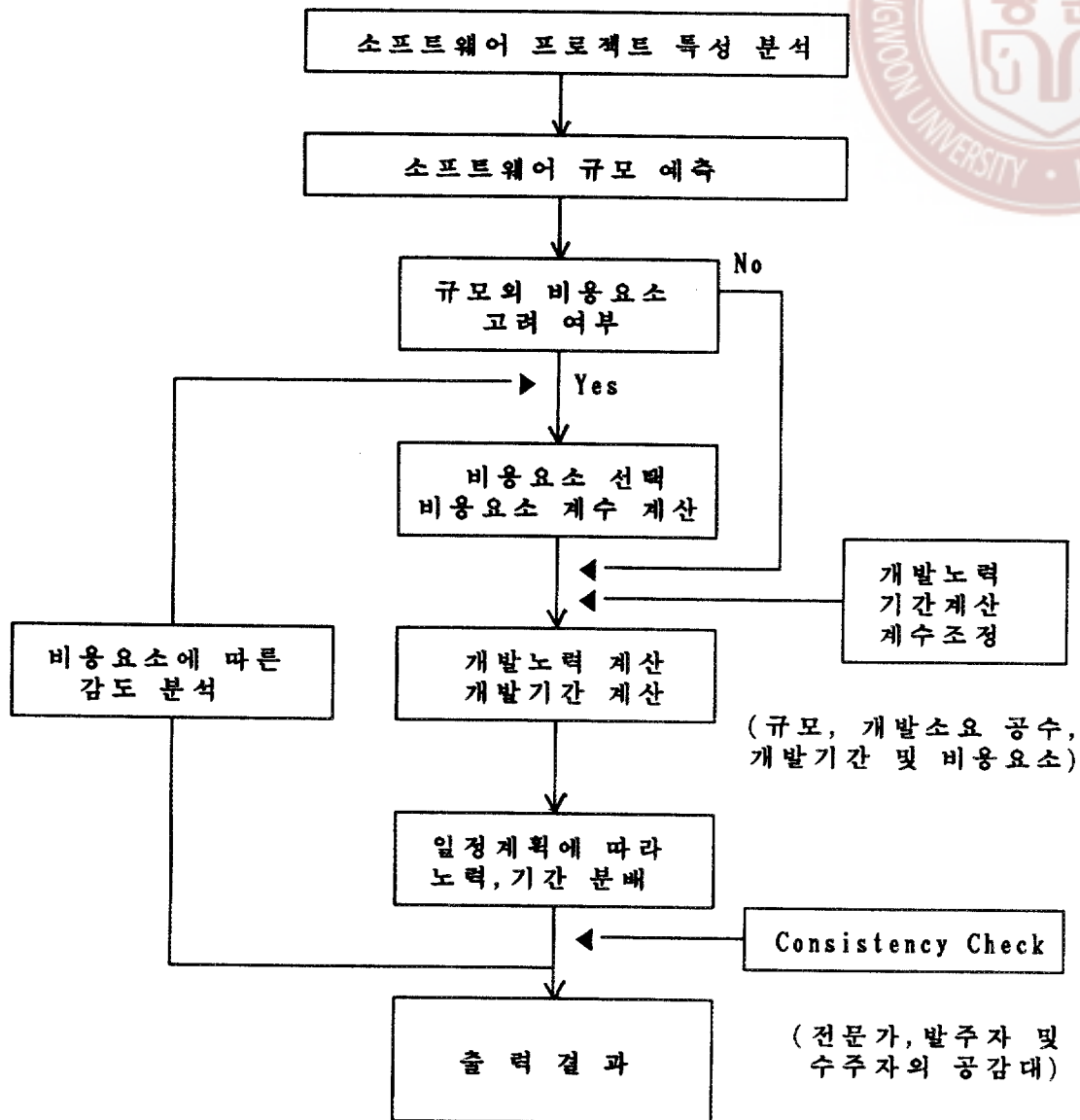


그림 1. 소프트웨어 개발비용 분석 흐름도
 Fig 1. Software development cost analysis flowchart



2.4 소프트웨어 비용 산출 기법

2.4.1 전문가 감정

비용 산정은 전문가의 감정에 의해 하는 것으로 이는 근본적으로 하향식 산정 기법의 일종이다. 전문가의 판정은 그 조직내에 있는 한 명 이상의 핵심 요원으로부터 경험, 배경과 업무처리 감각에 의존한다.

전문가의 감정에 의한 최대 장점은 신뢰성에 있다. 전문가는 프로젝트가 이전 것과 유사하다고 확신할 수 있으나, 새로운 프로젝트에는 상당히 차이 있게 만드는 요소를 간과할 수도 있다. 또는 산정을 실시하는 전문가는 현재 프로젝트와 유사한 프로젝트에 경험을 가지고 있지 않을 수도 있다. 이러한 문제점의 방지를 위하여 전문가 그룹이 일치되는 산정값을 준비하기도 하는데, 이는 개인적인 간과나 특별한 프로젝트와의 친밀감 결여를 최소화하여 개인적인 편견과 낙관적인 추산으로 계약을 중화시키는 경향이 있다. 그룹에 의한 산정이 지니는 주요 약점은 요원간의 그룹 역할은 그룹내의 개인들에 끼치는 영향이며, 그룹 요원들은 정책적인 안배, 그룹내의 권위 혹은 독단적인 그룹 요원의 억압 때문에 정확하지 못할 수도 있다. 델파이(Delphi) 기법은 이러한 약점을 보강하기 위해 사용된다.[18,23]



2.4.2 델파이(Delphi)식 비용산정

델파이 기법은 그룹 회의의 부작용을 야기시키지 않으면서도 전문가의 의견 일치를 얻기 위하여 1948년 랜드 회사 (Rand Co.)에서 개발된 것이다.

델파이 기법은 아래와 같은 방법으로 소프트웨어 비용산정을 한다.[18,23]

- (1) 조정자는 각 산정 요원에게 시스템의 정의 문서와 비용을 산정한 내역을 기록할 서식을 제공한다.
- (2) 산정 요원들은 정의 문서를 분석하여 익명으로 산정을 실시한다.
- (3) 조정자는 산정 요원들의 반응을 요약, 배포하며 산정 요원들에 의해 지적된 이론적인 근거도 포함시킨다.
- (4) 산정 요원들은 이전에 산정한 결과를 이용하여 다시 익명으로 다른 산정을 하며, 이전의 산정 내용과 아주 다르게 산정한 사람은 그 산정 내용을 정당화시키기 위하여 익명으로 질문을 받는다.
- (5) 이 과정을 필요한 만큼 반복하며, 이 과정에서 어떤 논의도 허용 되지 않는다.

산정을 여러 차례 실시하더라도 산정 내용이 일치하지 않을 가능성이 있다. 이 경우에 조정자는 차이가 나는 사유를 판단하기 위해



각 산정 요원과 관련 문제를 토의 한다.

조정자는 견해 차이를 해결하기 위해 추가 정보를 수집하며 이를 산정 요원에게 제시한다.[15,18,23]

2.4.3 업무 분류 구조

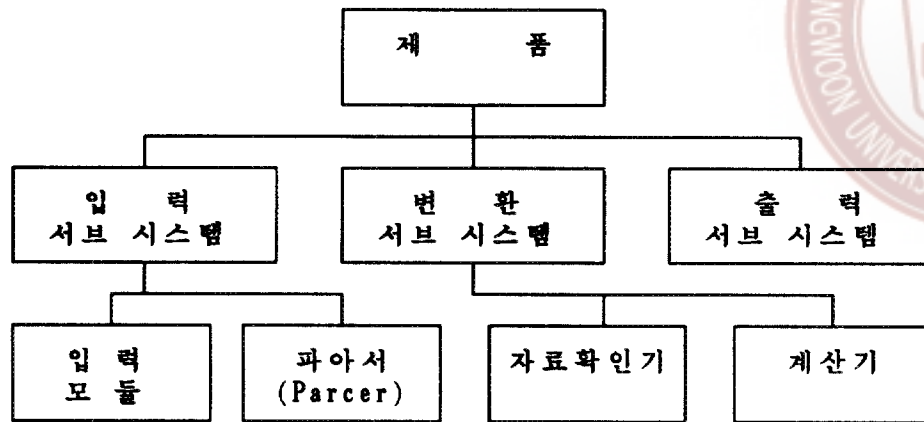
전문가의 감정과 그룹의 의견 일치에 의한 방법은 하향식 산정 기법인 반면에, 업무 분류 구조 방법(WBS, work breakdown structure)은 상향식 산정 도구이며, 업무 분류 구조는 시스템의 각 부분을 나타내는 계층형 도표이다. WBS 도표는 제품의 구조이거나 프로세스 계층 구조를 나타낼 수 있다.

제품 계층 구조는 제품의 성분을 밝히고, 성분이 상호 연결되는 방법을 나타내며, 프로세스 계층 구조는 작업 활동을 밝히고 이들 활동간의 상호 관계를 나타낸다. 전형적인 WBS 도표는 그림 2 의 (a) 와 (b)로 나타낼 수 있다.[18]

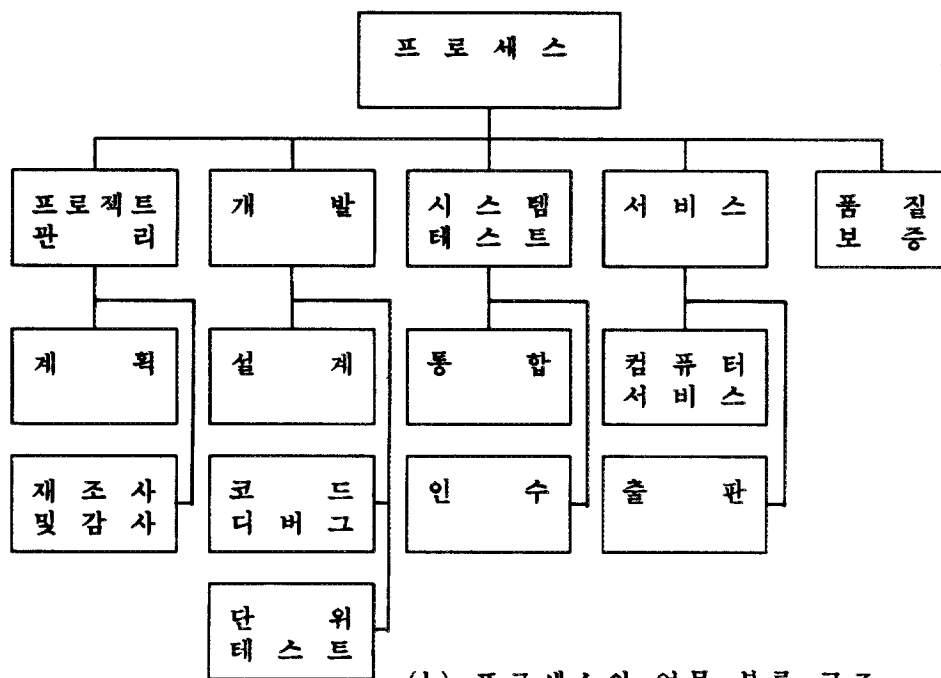
WBS 기법을 사용하는 경우에 비용은 도표내에 있는 각 성분의 비용을 할당하고 이들을 합산함으로써 추정된다.

WBS 기법의 주요 이점은 여러 프로세스와 제품의 성분을 밝힐 수 있고, 정확하게 비용을 산정할 수 있다.

전문가 감정, 그룹의 의견 일치 그리고 WBS는 현재 폭 넓게 사용 되는 비용산정 기법이다.



(a) 제품의 업무 분류 구조



(b) 프로세스의 업무 분류 구조

그림 2. WBS 도표
Fig 2. WBS Chart



2.4.4 연산 방식의 비용 모델

연산 방식의 비용산정은 시스템을 구성하고 있는 모듈과 서브 시스템의 비용 합계를 계산하여 소프트웨어 시스템의 비용을 산정하며, 연산 방식의 모델은 상향식 산정 방법이다.

COCOMO (constructive cost model)는 1981년 Boehm에 의해 고안된 연산 방식의 비용 모델로서 COCOMO를 이용한 비용산정 절차는 아래와 같다.[18]

- (1) 제품내의 모든 서브 시스템과 모듈로 분류한다.
- (2) 각 모듈의 크기를 산정하며 각 서브 시스템과 전체 시스템의 크기를 계산한다.
- (3) 각 모듈에 대한 모듈 수준의 노력 승수를 정의한다. 모듈 수준의 승수는 다음과 같다.
 - 소프트웨어 복잡도
 - 프로그래머의 능력
 - 사용 언어의 경험 정도
 - 가상 기계의 경험
- (4) 계산 방정식과 노력 승수를 이용하여 각 모듈에 대한 모듈 단위의 노력(module effort)과 개발기간을 산정한다.
- (5) 각 서브 시스템에 대해서 나머지 11개의 노력 승수를 정의한다.
- (6) (4), (5) 단계로부터 각 서브 시스템에 대한 노력과 개발기간



을 산정한다.

- (7) 단계 (6)으로부터 전체 시스템의 노력과 개발 기간을 계산한다.
- (8) trade-off 효과를 설정하기 위하여 산정값에 대한 민감도 분석을 실시한다.
- (9) 계획 수립, 분석과 같은 다른 개발 비용을 추가한다.
- (10) 하향식 델파이 산정에 의한 값과 비교하여 산정값에 차액이 있는가를 밝히고 이를 조정한다.

COCOMO의 최대 장점은 이 모델이 조직내의 비용요소를 제시하는 데 사용될 수 있으며, 자료를 수집 분석하여 새로운 요소가 밝혀질 수 있고 노력 승수는 COCOMO를 일정한 환경에 적용할 경우에 조정될 수 있다.

2.5 추정모델

기 개발된 추정 모델은 1965년 SDC 모델이 개발된 이래 10여개가 있으며 이들을 살펴보면 아래와 같다.[11,16,22]

(1) SDC (Nelson, 1965)

최초의 모델로서의 의미가 있으며 소프트웨어 개발시 미치는 특성들에 대하여 조사한 것에 의의가 있다.

(2) TWR (Wolverton, 1974)

한 DSI (developed source instruction) 당 가격을 0 - 100 사

이의 난이도, 프로젝트의 타입, 응용에 따라 새로운 구성들의 관계를 함수로 나타내었으며, 소프트웨어를 단위별로 나누어서 평가하는데 유용하다.

(3) SLIM (Putnam, 1978)

Rayleith 분산을 이용하여 Putnam의 분석에 기초를 둔 모델로 현재 많이 사용되며 입력과 출력의 관계를 다양하게 분석한 모델이다.

(4) Doty (Herd, 1977)

SDC 모델의 데이터틀 이용한 확장 모델로서 DSI를 10만 이상과 이하로 나누어 두 개의 함수로 주어진다.

(5) RCA PRICES (Freiman, 1979)

가격 변이치를 복잡도와 요소에 관계된 함수로써 표현할 수 있으며 단위별로도 분석이 가능하다.

(6) IBM - FSD (Walston, Felix, 1977)

29개의 특성을 이용하여 weighted 합으로 생산성을 계산할 수 있다.

(7) Boeing (Black, 1977)

5 - 6 개의 특성을 이용하여 크기를 변수로 주어 man-month를 측정하는 모델이다.

(8) GRC (Carriere, Thibodeau, 1979)

매우 다양한 상관관계를 측정할 수 있는 모델이다.

(9) Bailey - Basili Meta (Bailey, Basili, 1981)

통계적인 방법을 이용하여 상당한 추정치를 계산할 수 있는 모델로서 $\text{Effort} = a(\text{size})^b + c$ 의 형식으로 주어진다.

(10) COCOMO (Boehm, 1981)

생명주기의 진행에 따라 3개의 모델이 있고 프로젝트의 유형에 따라 또 다른 3개의 관계 함수가 주어지며, 기간 분석도 생명주기의 과정마다 주어지는 가장 좋은 모델로 평가된다.



Ⅲ. COCOMO 모델

3.1 COCOMO 모델 개요

소프트웨어 순기 비용 추정 의 한 방법으로서 한 개 내지 여러 개의 중요한 비용변수(cost driver)를 입력 변수로 한 함수식에 의하여 소프트웨어 순기 비용을 추정하는 방법을 algorithmic 모델이라 하며 Boehm에 의하여 개발된 COCOMO는 이러한 algorithmic 모델의 전형적인 예이다.

COCOMO 모델은 소프트웨어 프로젝트의 크기를 입력 변수로 하여 소프트웨어 개발에 필요한 비용, 인력과 일정 등을 구할 수 있으며, 이를 위해 필요한 소프트웨어 순기 비용 추정식을 얻는데 63개의 소프트웨어 프로젝트 데이터를 통계적으로 이용하였다.

3.2 COCOMO 모델의 용어와 가정

COCOMO 모델에서 사용되는 용어와 가정은 아래와 같다.

(1) DSI (delivered source instruction)

소프트웨어 개발 과정에서 작성되는 source program의 line수를 의미하며, comment는 line 수에서 제외된다.

(2) KDSI (kilo - DSI)

delivered source instruction in thousands



Ⅲ. COCOMO 모델

3.1 COCOMO 모델 개요

소프트웨어 순기 비용 추정 의 한 방법으로서 한 개 내지 여러 개의 중요한 비용변수(cost driver)를 입력 변수로 한 함수식에 의하여 소프트웨어 순기 비용을 추정하는 방법을 algorithmic 모델이라 하며 Boehm에 의하여 개발된 COCOMO는 이러한 algorithmic 모델의 전형적인 예이다.

COCOMO 모델은 소프트웨어 프로젝트의 크기를 입력 변수로 하여 소프트웨어 개발에 필요한 비용, 인력과 일정 등을 구할 수 있으며, 이를 위해 필요한 소프트웨어 순기 비용 추정식을 얻는데 63개의 소프트웨어 프로젝트 데이터를 통계적으로 이용하였다.

3.2 COCOMO 모델의 용어와 가정

COCOMO 모델에서 사용되는 용어와 가정은 아래와 같다.

(1) DSI (delivered source instruction)

소프트웨어 개발 과정에서 작성되는 source program의 line수를 의미하며, comment는 line 수에서 제외된다.

(2) KDSI (kilo - DSI)

delivered source instruction in thousands



(3) 투입공수 (MM = man-month)

1 MM = 152 man-hours (working hours) = 19 man-days

(4) 소프트웨어 개발기간 (TDEV = developed time, month)

- 소프트웨어 개발공정은 계획과 요구분석, 소프트웨어 설계, program 설계, coding과 개별 test, 소프트웨어 종합 test와 system 설치공정으로 분류한다.
- COCOMO에서 적용되는 개발공정은 소프트웨어 설계공정에서부터 소프트웨어 종합 test(소프트웨어 사용자 검수) 완료 공정까지를 의미한다.[24]

3.3 COCOMO 모델의 분류

COCOMO는 Boehm에 의해 개발된 project 비용추정모델로서 비용추정 단계와 적용변수의 구체성에 따라 3가지 모델로 분류되며 Basic COCOMO로 시작해서 Intermediate COCOMO와 Detailed COCOMO로 발전시켰다.

3.3.1 Basic COCOMO

소프트웨어 규모, 개발유형 · man-month 당 평균 비용만이 인력과 개발기간을 추정하는데 사용되고 개발 초기 단계에서 빠르고 쉽게 개략적인 소프트웨어 비용을 추정할 수 있다.

Organic mode는 소프트웨어 규모가 작고 program 난이도가 낮으며 잘 훈련된 소수의 인원으로 구성된 경우에 사용되며 Semidetached mode는 소프트웨어 중요도와 구성원의 질이 보통인 경우에 사용하고 Embedded mode는 난이도가 높고 소프트웨어가 제대로 작동되지 않으면 인명과 재산에 막대한 손해를 초래하여 개발시 신중을 기해야 하는 경우에 사용한다.

Basic COCOMO 방정식은 아래 표 3 과 같다.[11,15,22,23]

표 3. Basic COCOMO의 인력과 개발기간 방정식
Table 3. Basic COCOMO effort and schedule equations

구 분	인 력	개 발 기 간
Organic	$MM = 3.0(KDSI)^{1.05}$	$TDEV = 2.5(MM)^{0.38}$
Semidetached	$MM = 3.0(KDSI)^{1.12}$	$TDEV = 2.5(MM)^{0.35}$
Embedded	$MM = 2.8(KDSI)^{1.20}$	$TDEV = 2.5(MM)^{0.32}$



3.3.2 Intermediate COCOMO

Intermediate COCOMO는 Basic COCOMO에 15개의 비용 특성값을 첨가하였다. 그러므로 프로젝트의 성격이 좀 더 명확해지는 개발 시점에서 여러 가지 비용변수를 이용하여 실제에 가까운 비용추정을 할 수 있다.

Intermediate 방정식은 아래 표 4 와 같다.[24]

표 4. Intermediate COCOMO 인력과 개발기간 방정식
Table 4. Intermediate COCOMO nominal effort estimating equations

구 분	인 력	개 발 기 간
Organic	$MM = 3.0(KDSI) \sum_{i=1}^{15} fi$	$TDEV = 2.5(MM)^{0.38}$
Semidetached	$MM = 3.0(KDSI) \sum_{i=1}^{15} fi$	$TDEV = 2.5(MM)^{0.35}$
Embedded	$MM = 2.8(KDSI) \sum_{i=1}^{15} fi$	$TDEV = 2.5(MM)^{0.32}$

KDSI : 1000 line

MM = man-month

TDEV : 개발기간(developed time, month)

fi = 특성값



3.3.3 Detailed COCOMO

Detailed COCOMO 모델은 Intermediate COCOMO 모델과 대체로 유사하나 다음과 같은 두 가지 차이점을 가지고 있다.

첫째, Detailed COCOMO 모델에서는 소프트웨어 프로젝트를 system, subsystem, module 등 하부 체계로 분류하여 각 수준별로 미치는 영향을 달리 고려해 준다. 즉 각 module 수준에서 고려해야 할 비용변수가 있다. 이와 같이 각 수준별로 고려해야 할 비용변수를 분류하여 적용하고 있다.

둘째, Detailed COCOMO에서는 각 개발 단계별로 비용변수의 영향을 달리 고려한다. 즉 비용변수가 design 단계에서 끼치는 영향과 test 단계에서 끼치는 영향이 서로 다를 수 있으므로 이를 단계별로 고려하여 비용추정(cost estimation)을 하고 있다.

3.3.4 COCOMO 특성값

비용변수는 비용특성에 따라 4가지로 구분되며 5개의 신뢰도에 대한 노력 승수와 15개의 특성으로 구분된다.

COCOMO 특성값을 소프트웨어 특성, 컴퓨터의 특성, 개발자 특성, 프로젝트 특성으로 구분하며 제시된 내용은 표 5 와 같다.[1,11]

표 5. COCOMO 특성표
Table 5. Software development effort multipliers

구분	특성	의 미	매우 낮음	낮음	중간	높음	매우 높음	특히 높음
소프트웨어 특성	RELY	필요한 신뢰도	.75	.88	1.00	1.15	1.40	
	DATA	DB 의 크기		.94	1.00	1.08	1.16	
	CPLX	소프트웨어 복잡도	.90	.85	1.00	1.15	1.30	1.65
컴퓨터 특성	TIME	기계사용 제한정도			1.00	1.11	1.30	1.66
	STOR	기억장치 사용 제한 정도			1.00	1.06	1.21	1.56
	VIRT	가상기계의 휘발성		.87	1.00	1.15	1.30	
	TURN	턴어라운드의 정도		.87	1.00	1.07	1.15	
개발자 특성	ACAP	분석가의 능력	1.46	1.19	1.00	.86	.71	
	AEXP	경험 정도	1.29	1.13	1.00	.91	.82	
	PCAP	프로그래머 능력	1.42	1.17	1.00	.86	.70	
	VEXP	가상기계의 경험	1.21	1.10	1.00	.90		
	LEXP	사용 언어의 경험 정도	1.14	1.07	1.00	.95		
프로젝트 특성	MODP	선진 기술 습득 정도	1.24	1.10	1.00	.91	.82	
	TOOL	소프트웨어 도구	1.24	1.10	1.00	.91	.83	
	SECD	개발 스케줄	1.23	1.10	1.00	1.04	1.10	

Ⅳ. 소프트웨어 평가상의 COCOMO 모델 적용

4.1 적용 대상 프로젝트

적용 대상 프로젝트는 아래의 표 6 과 같이 8개의 기 개발된 소프트웨어 프로젝트를 선정하였으며, 그 특성은 표 5 와 비교해보면 쉽게 알 수 있다.

표 6. 수집자료
Table 6. Correction data

프로젝트 No. 특성	1	2	3	4	5	6	7	8
개발년도 개발소프트웨어	'88 COB	'86 PL/I	'82 COB	'87 PL/I	'85 PI/I	'88 DB	'84 PI/I	'88 DB
RELY	1.00	1.00	0.88	1.15	0.88	1.00	0.88	0.88
DATA	0.94	1.00	1.08	0.94	0.94	0.94	1.16	1.00
CPLX	1.00	1.15	1.30	1.00	0.85	1.15	1.30	1.00
TIME	1.00	1.15	1.00	1.15	1.15	1.15	1.15	1.00
STOR	1.21	1.06	1.00	1.06	1.06	1.06	1.00	1.00
VIRT	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
TURN	1.00	1.07	1.00	1.07	1.07	1.07	1.00	1.00
ACAP	0.86	0.86	1.00	0.86	0.86	0.86	0.86	1.00
AEXP	0.91	0.82	0.91	0.91	0.82	0.91	1.13	1.00
PCAP	0.86	1.00	0.86	1.00	0.86	0.86	1.00	1.17
VEXP	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
LEXP	1.00	0.95	0.95	1.07	1.00	1.00	1.00	1.07
MODP	0.91	0.91	1.00	0.91	0.91	1.00	1.10	1.00
TOOL	0.83	0.91	1.00	1.00	0.91	1.00	1.10	1.00
SECD	1.10	1.00	1.10	1.04	1.04	1.00	1.00	1.00
특성값	0.64	0.83	1.01	1.12	0.48	0.95	1.56	1.37
개발유형	Org	Sem	Sem	Sem	Org	Sem	Sem	Sem
KDSI	10.8	14.2	34	15.2	9.8	5.5	22	6.7
실제투입 인력	27	67	192	84	30	16	135	30
실제개발 기간	10	12	24	14	9	9	25	12

표 6 에서 개발 연도는 기 개발된 소프트웨어 프로젝트의 개발 완료 연도이며 개발 소프트웨어로는 COBOL, PL/I, Data Base 로 구성 되어있다.

표 6 에서 도출된 특성값은 프로젝트별로 15개의 특성값을 곱한 값이며 개발유형은 Organic mode와 Semidetached mode를 선정하고 KDSI, 실제투입 인력과 실제개발 기간은 기 개발된 소프트웨어를 대상으로 적용 하였다.

4.2 환경 특성값

표 6 의 각 프로젝트는 비 영리 조직에서 관리적 기능을 수행하는 자료처리 개발 프로젝트 중심으로 선정하였고 2 KDSI 이하의 소형 프로젝트는 특수성이 너무 강하여 제외시켰으며 3 KDSI 이상 100 KDSI 이하의 프로젝트로 구성되어 있다.

COCOMO 모델 선정은 Basic COCOMO에 15개의 특성값을 포함시켜 비교적 실제에 가까운 추정이 가능한 Intermediate COCOMO 모델을 적용하고(표 4 참조), 개발유형 선정은 소프트웨어 규모가 작고 program 난이도가 낮으며 잘 훈련된 소수의 인원으로 구성된 경우에 사용되는 Organic mode, 소프트웨어 중요도와 구성원이 보통인 경우에 사용되는 Semidetached mode를 적용하였다.

4.2.1 인 력

표 6 에서 각 프로젝트의 KDSI와 특성값을 Intermediate COCOMO의 방정식에 대입하여 비교한 결과 소프트웨어 프로젝트의 크기가 소형인 6과 8 프로젝트는 Intermediate COCOMO보다 인력 분야에 서 더 적은 인원으 로 개발된 것으로 나타났다. 그 이유는 프로젝트 규모가 작고 개발환경이 비교적 좋은 상태에서 개발되었기 때문이다. 그러나 소프트웨어 프로젝트의 크기가 큰 3 프로젝트는 인력 분야에서 실제 프로젝트의 투입 인력이 훨씬 더 큰 것으로 나타났다. 그 이유는 당시 하드웨어 사용 환경이 어려웠기 때문이다.

그 결과는 아래의 표 7 과 같다.

표 7. 인력 비교표
Table 7. Manpower comparative table

구분 프로 젝트 No.	인 력 (명)		
	COCOMO	실제	(실제) - (COCOCO)
1	25	27	2
2	52	67	15
3	131	160	29
4	75	84	9
5	17	30	13
6	20	16	- 4
7	128	135	7
8	37	30	- 7
합 계	485	549	64

4.2.2 개발기간

표 7 에서 각 프로젝트 인력을 Intermediate COCOMO의 방정식에 대입하여 비교한 결과 소프트웨어 프로젝트의 크기가 큰 3과 7 프로젝트는 Intermediate COCOMO 보다 개발기간 분야에서 큰 차이를 보여 실제 개발기간이 더 소요된 것으로 나타났다. 그 이유는 규모가 크고, 소프트웨어 복잡도가 매우 높은 프로젝트이기 때문이다. 그러나 그 외의 프로젝트들은 차이는 크지 않았으나 전반적으로 실제 프로젝트의 개발기간이 더 소요된 것으로 나타났다.

그 결과는 아래 표 8 과 같다.

표 8. 개발기간 비교표
Table 8. Developed time comparative table

구분 프로젝트No.	개발기간 (월)		
	COCOMO	실제	(실제) - (COCOMO)
1	8	10	2
2	10	12	2
3	14	20	6
4	11	14	3
5	7	9	2
6	7	7	0
7	14	19	5
8	9	9	0
합 계	80	100	20



4.2.3 환경 특성값 산출

표 7 과 표 8 에서 프로젝트 1과 8은 비교적 Intermediate COCOMO와 거의 일치하나 그 외의 소프트웨어 프로젝트들은 실제 투입 인력과 개발기간 분야에서 더 소요된 것으로 나타났다. 따라서 실제 프로젝트 수행에 절실히 요구되는 정확한 인력과 개발기간의 소요를 측정하기 위하여 Intermediate COCOMO와 실제 프로젝트와의 인력과 개발기간에 차이값의 합을 프로젝트의 수로 나눈 평균값을 환경 특성값이라고 정의하고 수식으로 표현하면 아래와 같다.

$$\begin{aligned} (1) \text{ 인력 환경 특성값} &= (\text{표 7 의 실제 인력의 합계} - \\ &\quad \text{COCOMO 인력의 합계}) \quad / \quad \text{프로젝트 수} \\ &= (549 - 485) \quad / \quad 8 \\ &= 8 \end{aligned}$$

$$\begin{aligned} (2) \text{ 개발기간 환경 특성값} &= (\text{표 8 의 실제 개발기간의 합계} - \\ &\quad \text{COCOMO 개발기간의 합계}) \quad / \quad \text{프로젝트 수} \\ &= (100 - 80) \quad / \quad 8 \\ &= 2.5 \end{aligned}$$



V. 적 용 결 과

5.1 COCOMO와 실제 프로젝트의 비교

표 7 과 표 8 에서 나타난 Intermediate COCOMO와 실제 프로젝트의 인력과 개발기간을 종합하면 아래의 표 9 와 같다.

아래의 표 9 에서와 같이 실제 프로젝트의 인력과 개발 기간이 비교적 더 소요됨을 알 수 있으며, 이것은 비 영리 조직의 특수성에 따라 인력과 개발 기간이 더 소요된 것이다.

표 9. COCOMO 와 실제 프로젝트 비교표
Table 9. Project comparison table between COCOMO and practical

구분 프로젝트 No.	인 력(명)		개발기간(월)	
	COCOMO	실 제	COCOMO	실 제
1	25	27	8	10
2	52	67	10	12
3	131	160	14	20
4	75	84	11	14
5	17	30	7	9
6	20	16	7	7
7	128	135	14	19
8	37	30	9	9



5.2 환경 특성값 적용 모델과의 비교

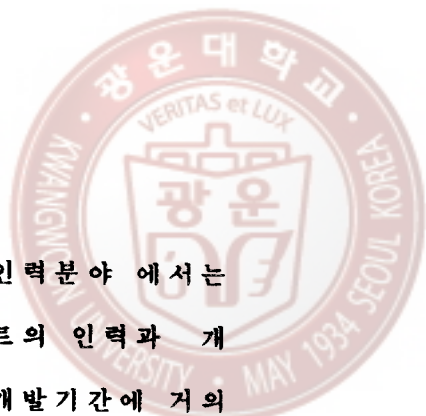
Intermediate COCOMO의 방정식에 환경 특성값을 적용한 실제 프로젝트의 인력과 개발기간을 종합하면 아래의 표 10 과 같다.

인력 환경 특성값 산출은 각 프로젝트별로 Intermediate COCOMO 방정식에 인력 환경 특성값 8을 합산한 값이며 개발기간 환경 특성값 산출은 2.5를 합산한 값이다.

아래의 표 10 에서와 같이 실제 프로젝트의 인력과 개발기간이 Intermediate COCOMO 방정식에 적용한 것보다 환경 특성값을 적용한 소요 추정치가 비교적 정확한 것으로 나타났다.

표 10. 환경 특성값 적용 비교표
Table 10. Environmental characteristic value and practice comparative table

구분 프로젝트 No.	인 력 (명)		개발기간 (월)	
	환경특성값	실 제	환경특성값	실 제
1	33	27	11	10
2	60	67	12	12
3	139	160	16	20
4	83	84	14	14
5	25	30	10	9
6	28	16	10	7
7	136	135	16	19
8	45	30	11	9



5.3 종합 분석

아래의 표 11 에서와 같이 8개 프로젝트 중 인력분야에서는 5개 프로젝트, 개발기간 분야에서는 6개 프로젝트의 인력과 개발기간의 소요가 실제 프로젝트에 소요된 인력과 개발기간에 거의 근접하였으며 환경 특성값을 적용한 COCOMO 모델이 인력과 개발기간을 더 정확하게 측정할 수 있었다.

종합 분석 결과표는 아래의 표 11 과 같다.

표 11. 종합 분석 결과표
Table 11. Total analytic result

() : 실제 프로젝트와의 차이값

구분 프로젝트 No.	인 력 (명)			개 발 기 간 (월)		
	COCOMO	실재	환경 특성값	COCOMO	실재	환경 특성값
1	25 (2)	27	33 (-5)	8 (2)	10	11 (-1)
2	52 (15)	67	60 (7)	10 (2)	12	12 (0)
3	131 (29)	160	139 (21)	14 (6)	20	16 (4)
4	75 (9)	84	83 (1)	11 (3)	14	14 (0)
5	17 (13)	30	25 (5)	7 (2)	9	10 (-1)
6	20 (-4)	16	28 (-12)	7 (0)	7	10 (-3)
7	128 (7)	135	136 (-1)	14 (5)	19	16 (3)
8	37 (-7)	30	45 (-15)	9 (0)	9	11 (2)

V. 결 론

본 논문에서는 환경 특성값을 COCOMO 모델에 적용하여 기 개발되어 있는 소프트웨어 프로젝트의 인력과 개발기간을 비교 분석하여 새로운 프로젝트 수행시 예상되는 인력과 개발기간의 소요를 보다 정확하게 측정할 수 있도록 개선하였다.

환경 특성값을 적용한 COCOMO 모델의 환경 특성값은 기 개발된 소프트웨어 프로젝트의 실제 투입 인력과 개발기간을 Intermediate COCOMO 모델에 특성값을 적용하여 구하였다.

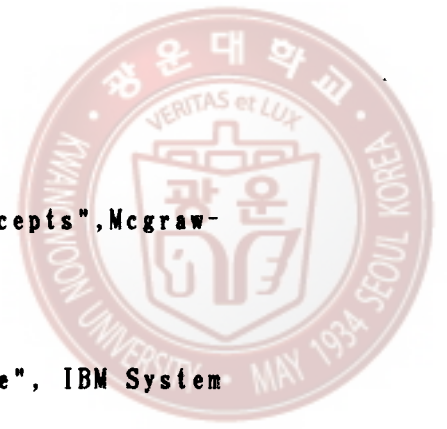
그리고 환경 특성값을 적용한 COCOMO 모델과 8개의 실제 프로젝트를 비교 분석한 결과, 기존의 Intermediate COCOMO보다 더 효과적인 인력과 개발기간의 소요를 얻을 수 있었다.

또한 환경 특성값을 적용하여 비교 분석한 실험 결과표 및 각종 결과표 작성에는 IBM PC AT, XT, 386 호환 기종에서 사용되는 스프레드시트(spread-sheet)를 tool로 사용하여 구현하였으며 그 유효성을 검증하였다.

앞으로의 연구 과제는 환경 특성값 산출시 프로젝트에 투입되는 인력과 개발기간에만 한정하지 않고 더 많은 COCOMO 특성값에 적용하여 환경 특성값을 산출하여야 하겠으며, 각 프로젝트의 크기에 따라 가중치를 적용한 환경 특성값의 산출에 대한 연구도 이루어져야 하겠다.

참 고 문 헌

1. Boehm, B.W., "Software Engineering Economics", Prentice-Hall, Inc., Englewood Cliffs, N.J., 1981.
2. Brooks, W.d., "The Mythical Man-Month", Addison-Wesley, Reading, M.A. 1975.
3. Distraso, J.R., "Software Management - A Survey of the Practice in 1980", Proc. of IEEE Vol. 68, No. 9, Sep. 1980, pp. 1103-1119.
4. Nelson, E.A., "Management Handbook for the Estimation of Computer Programming Cost", AD-A64870 System Development Corp., 1966.
5. Norden, P.V., "Useful Tools for Project Management" in Managements Production, M.K Starr, ed. Penguin Book, 1980.
6. Putnam, L.H., "The Economics of Software", in Software Cost Estimation and Life-Cycle Control, IEEE 1980.
7. Randall W. Jensen and Charles C. Tonies, "Software Engineering", Prentice-Hall, Inc., Englewood Cliffs, N.J., 1979.



8. Richard E. Faierley, "Software Engineering Concepts", McGraw-Hill, Inc., 1985.
9. Walston, C.E., and Felix, C.P., "Authors Response", IBM System Journal No., 1977.
10. Weinwurm, G.M. "On the Management Computer Programming", Auerbach, 1970.
11. 고진, 우치수, "소프트웨어 시스템의 원가 및 개발기간 평가에 관한 연구", 한국정보과학회 논문지, Vol.11, No.3, pp.182-188, 1984.
12. 금성 소프트웨어 주식회사, "정부 행정 전산망용 표준 : 하나 스프레드 시트", 금성 소프트웨어 주식회사, 1989.
13. 류성열, 윤창섭, 이철희, "소프트웨어 공학", 홍릉과학출판사, 1987.
14. 양해술, 노회영, "소프트웨어 Life Cycle의 비교분석", 정보과학회지, 제 4권 제 2호, 1986.
15. 양해술, "소프트웨어 공학의 현상과 동향", 하이테크 정보 출판부, 1989.



16. 우치수, "소프트웨어 경영 ", 정보과학회지 제 2권 제 2호, pp.43-49, 1984.
17. 이경한, "소프트웨어 공학 개념", 회중당, 1990.
18. 이동욱, 최병필, "소프트웨어 공학", 상조사, 1986.
19. 이재범, "소프트웨어 개발 비용산정 모델 소개", 경영과 컴퓨터, 1988.
20. 정왕호, 이기식, "소프트웨어 생산 기술", 정익사, 1982.
21. 조선형, "소프트웨어 공학", 상조사, 1986.
22. 한국과학기술연구원, "소프트웨어 비용산정 및 일정관리 시스템 개발에 관한 연구", 과학기술처, 1989.
23. 한국국방연구원, "소프트웨어 개발 사업관리", 1987.
24. 한국소프트웨어 산업협회, " 소프트웨어 개발비 산정 기준에 대한 연구", 1988.
25. 황종선, 백두권, "소프트웨어 공학", 교학사, 1990.