

---

# 회귀분석을 이용한 UCP 기반 소프트웨어 개발 노력 추정 모델

## Software Cost Estimation Model Based on Use Case Points by using Regression Model

---

박주석\*, 양해솔\*\*  
근로복지공단 정보시스템실\*, 호서대학교\*\*

Ju-Seok Park(iage2k@dreamwiz.com)\*, Hea-Sool Yang(hsyang@office.hoseo.ac.kr)\*\*

---

### 요약

최근 객체지향 개발 방법론을 적용하는 소프트웨어 개발 프로젝트에서 개발 노력 추정 기법으로 사용사례점수(Use Case Point, UCP)에 대한 연구가 계속되고 있다. 기존의 연구는 기술적 요인과 환경적 요인을 적용한 AUCP(Adjusted Use Case Point)에 상수를 곱하여 개발 노력을 추정하는 선형모델을 제안하고 있다. 그러나 소프트웨어 규모가 증가하면 개발기간은 기하급수적으로 증가함으로써 비선형 회귀모델이 적합하다는 사실과 UCP 계산과정에서 TCF(Technical Complexity Factor)와 EF(Environmental Factor)를 적용함에 따른 FP(Function Point) 오차가 발생함으로써 AUCP로 규모를 추정하는 것은 비현실적이다. 이 논문은 사용사례점수 기반의 기존 연구의 문제점을 제시하고, 기존 연구의 문제점인 TCF와 EF를 고려하지 않고 직접 UUCP로 부터 개발 노력을 추정할 수 있는 모델(선형, 로그형, 다항식, 거듭제곱, 지수형)을 도출하고 평가한다. 그 결과, 기존의 선형 모델보다 비선형모델인 지수형 모델이 우수한 결과를 보였다. 따라서 개발될 소프트웨어 시스템의 UUCP를 계산한 후 제안된 모델을 이용하여 개발 노력을 추정함으로써 개발에 소요되는 직접비용 산정이 가능하다.

■ 중심어 : | 사용사례점수 | 기술복잡도요인 | 환경요인 | 기능점수 | 조정된사용사례점수 |

### Abstract

Recently, there has been continued research on UCP from the development effort estimation method to a software development project applying object oriented development methodology. Current research proposes a linear model estimating the development effort by multiplying a constant to AUCP which applies technical and environmental factors. However, the fact that a non-linear regression model is more appropriate as the software size increases, the development period increases exponentially. In addition, in the UCP calculation process the occurrence of FP errors due to the application of TCF and EF, it is unrealistic to estimate the size with AUCP. This paper presents the issue of current research based on UCP without considering problems of the research, for example, TCF and EF and expresses the models (linear, logarithmic, polynomial, power and exponential type) estimating the development effort directly from UUCP. Consequently, the exponential model within non-linear models exhibit more accurate results than the current linear model. Therefore, after calculating the UUCP of the developing software system, using the proposed model to estimate the development effort, it is possible to estimate the direct cost required in development.

■ keyword : | UCP | TCF | EF | FP | AUCP |

## I. 서론

소프트웨어 개발비용 추정 분야는 20년 전부터 관심의 대상이 되어 왔으며, 비용에 영향을 미치는 인자(Cost Factor)에 기반을 두고 개발비용을 설명하기 위한 많은 방법들이 연구되고 실제 적용되고 있다. 그러나 이 분야에 대한 집중적인 연구 활동에도 불구하고 현재까지도 일반화된 결론을 유도하지 못하고 있는 실정이다[1][2]. 소프트웨어 비용(Cost) 추정은 소프트웨어 시스템을 구축하는데 소요되는 시간인 노력(Effort)의 양을 예측하는 과정이다[3].

소프트웨어 프로젝트에 소요되는 노력, 비용과 개발 일정을 적절히 추정하기 위한 핵심적인 인자 중 하나로 소프트웨어 규모(Size)에 대한 정량화가 일반적으로 인식되고 있다[4]. 소프트웨어 규모는 길이(Length), 기능성(Functionality)과 복잡도(Complexity) 인자들에 의해 정의될 수 있으며[4], 라인 수(Lines Of Code, LOC)와 기능점수(Function Point, FP)가 일반적으로 사용되고 있다[3-6]. 기능점수 방법[6]은 데이터 관리 위주인 관리정보시스템(Management Information System, MIS) 소프트웨어에 적합하도록 설계되었다. 이에 따라 기능점수 기법을 제어 위주인 실시간 시스템(Real-time System)과 내장형 시스템(Embedded System)으로 적용 범위가 확장되었고 이를 완전 기능점수(Full Function Point, FFP)라 부르고 있다.

기능점수나 완전 기능점수는 구조적 개발 방법에 적합한 기능성에 기반을 두고 있어 트랜잭션 처리나 파일 형태에 적합하며, 자료흐름도(Data Flow Diagram), 계층적 프로세스 모델 또는 데이터베이스 구조와 같은 전통적인 구조적 설계기법으로부터 유도된다[4].

대부분의 소프트웨어 개발의 실패 원인이 개발 초기에 사용자의 요구사항 도출 어려움에 기인하여 이를 해결하려는 연구가 계속되었다. 1992년 Jacobson[7]에 의해 요구사항 도출방법으로 유스케이스(Use Case)가 제안되었고, 1998년 UML(Unified Modeling Language)에 유스케이스도(Use Case Diagram)가 포함된 이후 소프트웨어 산업계에서는 유스케이스-구동(Use Case-driven), UML 기반의 단일화된 프로세스(Unified

Process)로 객체지향 소프트웨어를 개발하고 있다[4].

Karner[8]는 Albrecht[6]의 기능점수 계산방법을 수정(Modify)하여 유스케이스 점수(Use Case Point, UCP)를 계산하는 방법을 제시하였다. 이후에도 유스케이스 점수에 대한 연구는 계속되고 있으나 유스케이스 점수에 기반하여 개발노력을 추정할 수 있는 모델 연구는 미미한 수준이다. 본 논문은 소프트웨어 개발에 소요되는 노력을 추정함에 있어, 객체지향 개발방법론에 적합한 유스케이스 점수로 측정된 소프트웨어 규모에 따른 개발노력을 추정할 수 있는 통계적 모델을 제시한다.

2장에서는 소프트웨어 개발노력 추정과 관련된 연구를 살펴보고, 3장에서는 기존의 개발 노력 추정 모델들의 문제점을 제시한다. 4장에서는 유스케이스 점수 기반의 소프트웨어 개발노력추정 모델을 제시하고, 5장에서는 실무 데이터 적용을 통한 모델의 적합성을 평가해 본다.

## II. 개발 노력 추정 관련 연구

사용사례 점수는 기능점수를 수정하여 유도된 관례로 사용사례 점수를 이해하려면 기능점수를 먼저 이해하는 것이 필요하다. 본 장에서는 기능점수와 사용사례 점수를 계산하는 방법을 간략히 살펴본다.

### 2.1 기능점수

사용자에게 제공되는 응용프로그램의 규모를 계산 가능한 기능성으로 측정하기 위해, 응용 프로그램을 데이터기능(Data Function)과 처리기능(Transaction Function)으로 구분한다. 데이터 기능은 내부와 외부 데이터 요구사항에 일치하도록 사용자에게 제공되는 것으로 화일(Internal Logical Files, ILF)과 인터페이스(External Interface File, EIF)가 있다. 처리 기능은 처리된 데이터를 사용자에게 제공하는 것으로 입력(External Inputs, EI), 출력(External Outputs, EO)과 조회(External Inquiries, EQ)의 3가지 기능요소(Function Element)로 세분화된다.

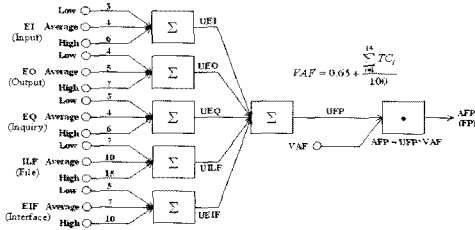


그림 1. 기능점수 계산과정

기능점수 계산 과정은 [그림 1]에 간략히 표현하였으며, [그림 1]의 계산과정에 따라 식 (1)과 같이 계산된다 [9].

$$AFP = VAF \cdot UFP \quad (1)$$

$$VAF = (TDI \cdot 0.01) + 0.65$$

### 2.2 완전기능점수

완전 기능점수는 [그림 2]와 같이 소프트웨어 시스템을 4 종류의 데이터 이동 서브프로세스로 구분한다 [10][ 11]. 소프트웨어 관점에서 볼 때, 입력(ENTRIES)와 출력(EXITs)은 사용자와 주고받는 데이터 속성이며, 읽기(READS)와 쓰기(WRITEs)는 저장장치와 주고받는 데이터 속성이다.

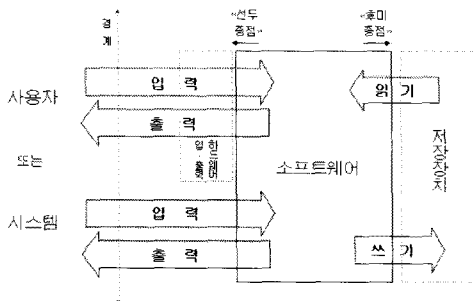


그림 2. FFP 구성요소

완전기능점수에 기반한 소프트웨어 규모는 Cfsu (Cosmic Functional Size Unit)로 표기하며, 임의의  $i$  번째 계층에 대한 Cfsu는 [그림 3]의 계산 과정에 따라 식 (2)과 같이 계산된다[10][11].

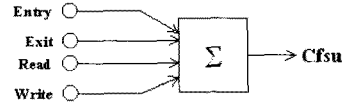


그림 3. FFP 계산과정

$$Size_{Cfsu}(layer_i) = \sum size(entries_i) + \sum size(exits_i) + \sum size(reads_i) + \sum size(writes_i) \quad (2)$$

### 2.3 유스케이스 점수

유스케이스 점수(Use Case Point)의 개략적인 계산 과정은 [그림 4]에 제시되어 있다. 먼저, 각 액터(Actor)의 복잡도(Simple, Average, Complex) 가중치가 합해져 UAW(Unadjusted Actor Weights)가 계산되고, 각 유스케이스의 복잡도 가중치가 합해져 UUCW(Unadjusted Use Case Weights)가 구해진다. 이어서  $UAW + UUCW = UUCP$ (Unadjusted Use Case Point)가 계산된다. 여기에 기술복잡도요인(Technical Complexity Factor, TCF)과 환경요인(Environmental Factor, EF)이 곱해져 AUCP (Adjusted Use Case Point)가 최종적으로 구해지며 식(3)으로 표현된다. AUCP를 일반적으로 UCP라 부른다.

$$AUCP = UUCP \times TCF \times EF \quad (3)$$

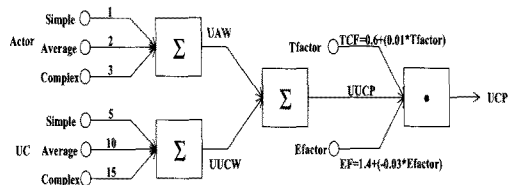


그림 4. UCP 계산과정

### III. 기존 UCP기반 개발노력추정 모델 문제점

기능점수 방법을 적용하는 실무자들은 UFP가 AFP와 같이 정확성을 가지고 있기 때문에 VAF 값을 무시하고 있다[4]. 또한, Kitchenham과 Käänsälä [12]도 VAF 값이 개발노력의 추정 능력을 향상시키지 못함을

보였다. 이에 따라 Ribu[4]도 식 (3)의 AUCP 계산에서 TCF를 제외시켜도 개발노력 (E)를 추정하는데는 영향이 없음을 제시하였다.

UCP를 이용하여 LOE(Level Of Effort)를 얻기 위해 식 (4)의 형태에 대한 연구가 다양하게 진행되었다.

$$E = AUCP \times \text{쓰임새 점수 당 개발 소요시간} \quad (4)$$

사용사례점수 당 개발 소요시간(시간/UCP)는 개발 조직의 생산성에 의존한다[13][14]. Banerjee[13]는 실제 적용 경험에 의하면  $15 \leq \text{시간}/UCP \leq 30$ 의 범위를 갖고 있다고 제안하였으며, Probasco[14]는 20시간부터 시작하여 개발조직의 생산성에 따라 조절해야 함을 제시하였다. Kamer[8]는 20 시간/UCP를 적용하였으며, Schneider와 Winter는  $EF \leq 2$ 인 경우 20 시간/UCP를,  $3 \leq EF \leq 4$ 인 경우 28 시간/UCP,  $EF \geq 5$ 인 경우 위험요소가 허용이 불가능할 정도로 크므로 EF를 조절하는 것이 필요함을 제시하였다[13]. Ribu[4]는 36 시간/UCP를 적용하는 특별한 경우도 경험하였으며, 학생들이 수행한 프로젝트에는 10 시간/UCP를 적용하였다. Sun 사는 30 시간/UCP, 어떤 업체에서는 중간값을 취함을 경험하였다[13].

사용사례 점수에 기반을 둔 개발노력 추정 기존 연구들은 다음과 같은 문제점들을 내포하고 있다.

- (1) 기존의 연구들은 AUCP에 상수를 곱하여 개발노력을 계산하였다. 즉, 개발노력은 AUCP에 선형적인 관계를 갖는 1차 함수( $E = a \cdot AUCP$ )로 프로젝트의 규모에 상관없이 UCP 당 일정한 시간이 투입됨을 의미하나 이는 현실적으로 적합하지 않다. 왜냐하면 프로젝트의 규모가 커지면 프로그램의 복잡도가 기하급수적으로 증가하는 경향을 띠고 있어 개발에 투입되는 시간도 프로젝트 규모에 따라 기하급수적으로 증가하는 비선형적인 관계를 나타낸다. 따라서, 선형회귀모델 보다는 비선형 회귀모델이 보다 적합할 수 있다.
- (2) UCP 계산 과정에서 VAF와 동일한 개념인 TCF는 최종적으로 계산되는 UCP에  $\pm 40\%$ 의 편차를 나타내며, EF는  $-30 \sim +40\%$ 의 편차를 유발시킬

수 있다. 그러나 기능점수 계산 과정에서 VAF는 최종적으로 계산되는 FP에 대해  $\pm 35\%$ 의 오차를 유발시킬 수 있음에도 불구하고 효용성이 없는 것으로 연구되어 AFP 보다는 UFP를 적용하고 있다. 따라서, TCF나 EF를 적용한 AUCP 보다는 UUCP만으로 개발노력을 추정할 수 있을 것이다.

AUCP나 UUCP를 이용하여 개발노력이나 비용을 추정하는 통계적인 모델은 제시되지 않고 있다.

#### IV. UCP 기반 개발노력 추정 모델

추정된 개발노력 (시간)에 대해 개발업체의 단위 시간당 평균 소요 비용을 곱하면 개발비용이 산출된다. 이와 같은 이유로 인해 일반적으로 개발노력 추정 모델을 개발비용 추정 모델이라 부른다[3]. 이후부터는 개발노력을 추정하는 것을 개발비용 추정이라 칭한다. 소프트웨어 비용추정 모델들은 일반적으로 주요 비용인자인 규모와 부수적인 조절인자로 구성되어 있다. 전형적인 비용 추정 모델은 과거 개발된 소프트웨어 프로젝트로부터 수집된 데이터에 대한 회귀분석으로부터 유도되며, 모델의 구조는 식 (5)로 표현된다[15].

$$E = a + b \cdot S^c \quad (5)$$

여기서,  $E$ 는 개발노력 (Person-months 또는 Person-Hours),  $a, b, c$ 는 경험적으로 유도된 상수이며  $S$ 는 주요 비용인자인 규모로서 LOC 또는 FP가 된다.

본 제안 모델은 3장에서 제시한 기존 연구의 문제점 (2)에 따라 TCF와 EF를 고려하지 않고 직접 UUCP로부터 개발노력을 추정할 수 있는 통계적 모델을 제시한다. 모델 제시를 위해 Libu[4]가 유럽의 주요 소프트웨어 회사에서 수행한 4개 프로젝트와 Oslo 대학에서 수행한 10개 프로젝트에서, Nageswaren[16]가 북미 대형 소프트웨어 회사의 웹사이트 구축 프로젝트에서 수행한 프로젝트 데이터를 활용한다. 활용한 데이터는 사용자

요구사항을 비즈니스 수준에서 Use Case를 식별하고, 식별된 Use Case에서 상이한 Actor를 식별하였으며, 식별된 데이터 [표 1]과 같다

표 1. 사용사례점수 관련 데이터

프로젝트	Actor				Use Case				
	Simple	Medium	Complex	UAW	Simple	Medium	Complex	UUCW	
S.Nageswaren [19]				55				64	
Ribu [4]	Project B-Subsystem 1	1	3	2	13	10	7	2	150
	Project B-Subsystem 2	0	4	1	11	12	9	0	150
	Project B-Subsystem 3	0	5	0	10	2	4	6	145
	Project B-Subsystem 4	1	3	2	13	0	8	2	110
	1-Q	0	1	2	8	0	6	0	60
	2-S	0	0	3	9	6	1	0	40
	3-S	0	0	3	9	4	2	0	40
	4-Q	1	0	2	7	2	3	1	55
	5-Q	1	0	2	7	0	4	0	40
	6-S	1	0	2	7	2	7	0	80
	7-Q	1	1	2	9	1	2	0	25
	8-S	0	1	3	11	2	3	1	55
	9-S	1	0	2	7	1	2	1	40
	10-Q	1	1	2	9	4	4	0	60
Project A	0	0	2	6	18	41	4	560	

프로젝트	UUCP	TC Factor	E Factor	TCF	EF	AUCP	Staff-Hour per UCP	Effort	
S. Nageswaren[19]	119	87	13	1.47	1.01	176.68	20	3,120	
Ribu [4]	Project B-Subsystem 1	163	43	17.5	1.03	0.88	146.90	28	3,723
	Project B-Subsystem 2	161	43	17.5	1.03	0.88	145.10	28	3,665
	Project B-Subsystem 3	155	43	17.5	1.03	0.88	139.69	28	3,835
	Project B-Subsystem 4	123	43	17.5	1.03	0.88	110.85	28	2,710
	1-Q	68	25	19	0.85	0.83	47.97	10	294
	2-S	49	19	19	0.79	0.83	32.13	10	298
	3-S	49	19	19	0.79	0.83	32.13	10	232
	4-Q	62	25	19	0.85	0.83	43.74	10	371
	5-Q	47	25	19	0.85	0.83	33.16	10	420
	6-S	87	19	19	0.79	0.83	57.05	10	580
	7-Q	34	25	19	0.85	0.83	23.99	10	243
	8-S	66	19	19	0.79	0.83	43.28	10	595
	9-S	47	19	19	0.79	0.83	30.82	10	578
	10-Q	69	25	19	0.85	0.83	48.68	10	490
Project A	566	44	19	1.04	0.92	541.55	10	10,043	

[표 1]에서 Project A는 다른 데이터들에 비해 UUCP, UCP와 Effort 값이 월등히 큰 이상점을 갖고 있어 이 데이터를 포함시켜 모델을 유도하면 편향된 결과를 얻을 수 있다. 따라서 본 논문에서는 Project A 데이터를 포함시킨 모델은 참고용으로 제시한다.

적합한 통계적 회귀모델을 선정하기 위해 결정계수 (Coefficient of determination,  $R^2$ )[17]를 적용하며, 모델의 성능을 평가하기 위해 Briand et al.[2]이 적용한 MMRE (Mean Magnitude of Relative Error)를 이용한다. 종속변수의 값 (E)은 독립변수(UUCP)에 의해 결정되는 부분과 미지의 오차의 합으로 나타나며, 총 변동을 설명하는데 있어서 회귀직선에 의해 설명되는 변동 비율을 결정계수라 하며, 값이 클수록 쓸모 있는 회귀직선이 되지만 좋은 모델로 선정하기 위한 기준은 없는 실정이다. 모델의 정확도 (Accuracy)를 평가하는 MMRE는 다음과 같이 측정된다.  $RE = \frac{\text{추정치} - \text{실측치}}{\text{실측치}}$ , MRE (Magnitude of RE) =

$$ABS(RE), MMRE (Mean MRE) = 100/n * \sum_{i=1}^n MRE.$$

MMRE가 작은 값이면 평균적으로 좋은 모델임을 알 수 있다. Conte et al.[18]는  $MMRE \leq 0.25$  (25%)이면 개발비용을 예측하는 모델로 채택 가능한 것으로 고려하였다.

#### 4.1 UUCP기반 개발노력 추정 모델(Project A 포함)

Project A를 포함한 경우에 대해 UUCP에 따른 개발 비용 추정모델은 [그림 6]에, 모델의 성능 분석 결과는 [표 2]에 제시되어 있다.

Project A를 포함한 경우, 선형, 로그와 지수 함수 형태는 100 이상의 UUCP에 대한 개발노력(시간)을 표현하지 못하고 있으며, 거듭제곱 함수 형태가 가장 적합한 모델로 선정될 수 있다.

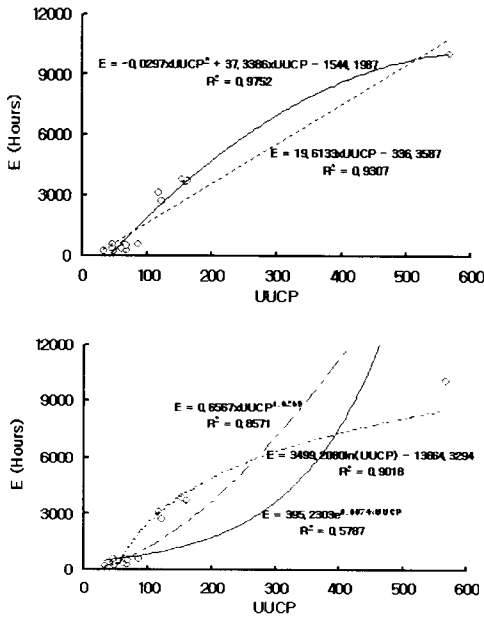


그림 5. 회귀분석모델(프로젝트 A 포함)

표 2. 개발비용추정모델성능(프로젝트 A 포함)

모델 형태	모 델	결정 계수	MM RE
선형	$E = 19.6133 \cdot UUCP - 336.3587$	0.9307	73.68
로그	$E = 3499.2080 \cdot \ln(UUCP) - 13664.3294$	0.9018	127.90
다항식	$E = -0.0297 \cdot UUCP^2 + 37.3386 \cdot UUCP - 1544.1987$	0.9752	61.03
거듭제곱	$E = 0.6567 \cdot UUCP^{0.6299}$	0.8571	43.42
지수	$E = 393.2303 \cdot e^{0.0074 \cdot UUCP}$	0.5787	70.84

#### 4.2 UUCP기반 개발노력추정모델(Project A 제거)

Project A를 제거하고 나머지 데이터들만을 이용하여 UUCP에 따른 개발비용 추정모델은 [그림 6]에, 모델의 성능 분석 결과는 [표 3]에 제시되어 있다. Project A를 제거 시 선형과 로그 함수 형태는 40 이하의 UUCP를 가진 프로젝트 규모에 대해서는 음수 값을 가지는 개발시간을 추정하여 현실성이 없다. 지수와 다항식 함수 형태의 모델이 전반적으로 모든 데이터를 가장 잘 표현하는 모델이라 할 수 있으며, 지수함수 형태의 모델이 MMRE가 가장 적고 Conte et al.[18]의 개발노

력 예측 모델 선정조건을 거의 만족시키고 있다.

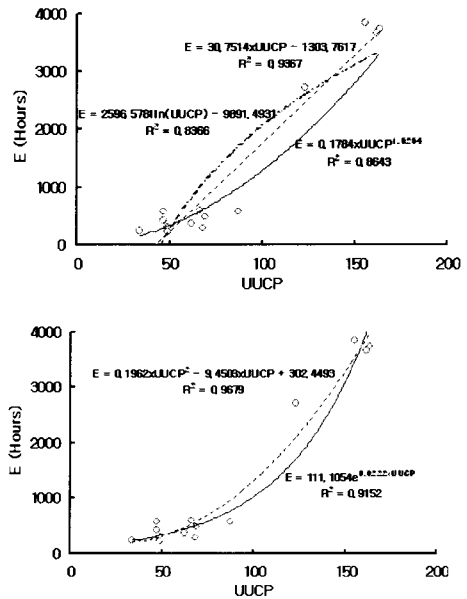


그림 6. 회귀분석 모델 (프로젝트 A 제거)

표 3. 개발비용추정모델성능(프로젝트 A 제거)

모델 형태	모 델	결정 계수	MM RE
선형	$E = 30.7514 \cdot UUCP - 1303.7617$	0.9367	59.45
로그	$E = 2596.5781 \cdot \ln(UUCP) - 9891.4931$	0.8366	94.96
다항식	$E = 0.1962 \cdot UUCP^2 - 9.4503 \cdot UUCP + 302.4493$	0.9679	28.50
거듭제곱	$E = 0.1784 \cdot UUCP^{1.9294}$	0.8643	34.90
지수	$E = 111.1054 \cdot e^{0.0222 \cdot UUCP}$	0.9152	25.85

따라서, [표 3]의 모델성능분석결과를 토대로 MMRE가 25%이고 회귀식의 설명력을 판단하는 결정계수가 91%가 되는 지수모델인  $E = 111.1054 \cdot e^{0.0222 \cdot UUCP}$ 를 UUCP를 이용한 개발노력을 추정하는 모델로 제안한다.

기존 연구들이 수행한 조절된 사용사례 점수에 상수를 곱한 형태의 모델들과 제안된 모델의 성능을 비교하여 보자. 기존 연구들은  $10 \leq \text{시간}/AUCP \leq 30$ 으로 다양한 결과를 보였다. 그러나 일반적으로

15 ≤ 시간/UCP ≤ 30의 범위를 갖고 있으므로 AUCP 당 개발시간을 15, 20과 30을 적용하여 [표 1]의 데이터들에 대한 MMRE를 계산하면 각각 48.93, 72.87과 142.95를 얻을 수 있다. 이는 제안된 모델의 MMRE 성능인 25.85에 비해 월등히 좋지 않은 결과를 나타내며, 기존 연구들을 이용할 경우 AUCP당 개발시간을 얼마로 결정할 것인가에 따라 개발비용에 큰 편차를 나타낼 수 있다.

개발될 소프트웨어 시스템의 UUCP를 계산한 후, 제안된 모델을 이용하여 개발노력을 추정하고 개발업체의 시간당 평균 소요 비용을 곱하면 개발에 소요되는 직접비용을 구할 수 있다.

### V. UUCP기반의 개발노력추정 실무사례 적용 및 평가

제4장에서 제안한 비선형 모델은 적용된 데이터의 갯수가 매우 적고 소프트웨어 규모가 매우 작은 부분에 한정되어 일반화된 모델 제시로는 불충분하다.

따라서, 모델 검증을 위한 데이터수집이 필요하여 조사해 본 결과, UML을 기반으로 하여 객체 지향 방법론을 적용한 국내의 소프트웨어 개발은 극히 제한적으로 이루어지고 있다. 다행스럽게도 국내의 IT 5개 업체에서 개발된 특정기관에 대한 경영정보체계(MIS)를 개발한 33개 프로젝트의 개발 사례를 획득할 수 있었다. 획득된 소프트웨어 프로젝트에 대하여 [표 4]와 같이 Kerner[8]가 제시한 방법에 기반하여 UCP를 계산하기 위한 액터와 사용사례를 식별하였다.

또한 [표 4]의 데이터는 실제 프로젝트 개발 당시 관련 문서와 개발 환경을 참조하여 TCF와 EF를 산출하였으며, 실제 투입된 노력을 제시하였다. 따라서, 이러한 데이터를 사용함으로써 일반화되고 모델로 적용될 수 있을 것이다.

표 4. 유스케이스점수 관련 국내 데이터

프로젝트	Actor (UAW)	Use Case (UUCW)	UUCP	AUCP	Staff-Hour per UCP	Effort
P1	12	75	87	74.57	-	9,918
P2	12	45	57	48.86	-	6,314
P3	3	75	78	66.86	-	5,411
P4	3	60	63	54.00	-	4,642
P5	9	85	94	91.79	-	11,023
P6	9	60	69	67.83	-	5,576
P7	9	90	99	111.17	-	9,347
P8	9	55	64	71.87	-	3,177
P9	18	150	168	154.94	-	14,711
P10	18	100	118	108.83	-	9,555
P11	3	115	118	101.14	-	7,057
P12	3	155	158	135.43	-	13,976
P13	30	175	205	189.06	-	37,246
P14	30	120	150	138.34	-	10,705
P15	24	165	189	162.36	-	15,678
P16	23	180	204	175.25	-	22,123
P17	6	205	211	153.99	-	34,620
P18	6	165	171	124.80	-	17,370
P19	12	260	272	205.77	-	37,654
P20	12	160	172	130.12	-	11,081
P21	9	230	239	205.31	-	19,011
P22	9	255	264	226.79	-	20,855
P23	6	260	266	201.23	-	39,326
P24	6	290	296	223.92	-	42,359
P25	21	310	331	284.35	-	46,920
P26	21	295	316	271.46	-	43,618
P27	12	315	327	280.91	-	59,766
P28	12	260	272	233.66	-	39,785
P29	12	160	172	147.76	-	13,934
P30	9	170	179	135.41	-	12,369
P31	12	210	222	156.62	-	22,542
P32	12	330	342	241.28	-	55,302
P33	12	285	297	209.53	-	36,920

Ribu[4]가 제시한 사용사례 점수당 개발 소요시간(시간/UCP)을 15시간/UCP, 20시간/UCP, 30시간/UCP를 적용시 소요되는 노력 및 본 절에서 제시한 통계적 모델에 따른 소요되는 노력과 실제 개발에 소요되는 노력을 비교 / 분석하였다.

UUCP에 따른 실무 사례를 적용한 개발비용 추정 모델은 [그림 7]에, 모델의 평가 분석한 결과는 [표 5]에 제시되어 있다. 지수 형태의 모델이 전반적으로 모든 데이터를 가장 잘 표현하는 모델이라 할 수 있다. 또한 지수 함수 형태의 모델이 MMRE가 가장 적고 Conte et al.[18]의 개발노력 예측 모델 선정 조건에 적합한 가장 현실적인 모델이다.

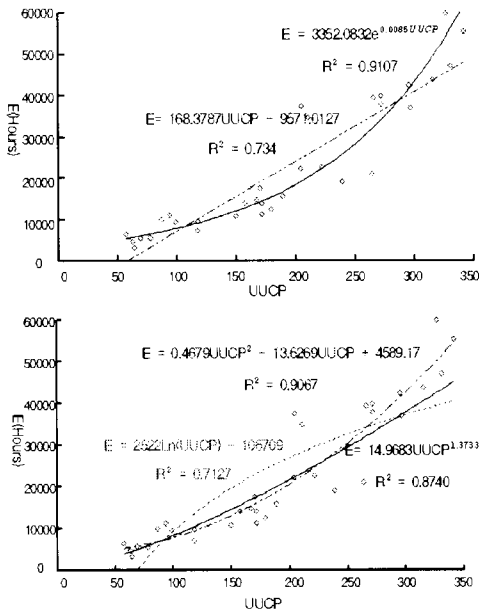


그림 7. 실무사례를 적용한 개발비용추정모델

표 5. 실무사례를 적용한 개발비용추정모델 평가

모델 형태	모 델	결정계수	MMRE
선 형	$E = 168.3787 \cdot UUCP - 9571$	0.8431	30.65
로 그	$E = 25222 \cdot \ln(UUCP) - 106709$	0.7127	44.50
다항식	$E = 0.4679 \cdot UUCP^2 - 13.6269 \cdot UUCP + 4589.17$	0.9067	25.30
거듭제곱	$E = 14.9683 \cdot UUCP^{1.3733}$	0.8740	26.16
지 수	$E = 3352.0832 \cdot e^{0.0085 \cdot UUCP}$	0.9107	25.20

따라서 [표 5]의 실무사례를 적용한 모델성능분석결과를 토대로 MMRE가 25%이고 회귀식의 설명력을 판단하는 결정계수가 91%가 되는 지수함수 모델인  $E = 3352.0832 \cdot e^{0.0085 \cdot UUCP}$ 를 제안한다.

또한 기존 연구들이 수행한 조절된 사용사례 점수(AUCP)에 상수를 곱한 형태의 모델들과 제안된 모델의 성능 및 실제 프로젝트를 수행한 개발노력(실측치)에 대한 성능을 비교한다. 기존 연구들의 AUCP는  $10 \leq \text{시간}/\text{UCP} \leq 30$ 으로 다양한 결과를 보였다.

그러나 일반적으로  $15 \leq \text{시간}/\text{UCP} \leq 30$ 의 범위를 갖고 있으므로 AUCP당 개발시간을 15, 20과 30을 적용

하여 UCP당 개발 노력과 제안된 지수 함수 모델을 적용시 개발 노력을 산출한 결과를 [그림 8]로 비교하였다.

[표 4]의 데이터를 기반한 AUCP 당 개발 시간을 15, 20, 30을 적용할 시 MMRE를 계산하면 각각 87.33, 81.90과 72.84를 얻을 수 있다. [표 6]은 추정형태 및 제안된 모델의 평가 결과를 제시한 것이다.

따라서 기존 연구들은 제안된 모델에 비해 상대오차 측면에서 월등히 좋지 않은 결과를 나타내며, 기존 연구들을 이용할 경우 AUCP당 개발시간을 얼마로 결정할 것인가에 따라 개발비용에 큰 편차를 나타낼 수 있다.

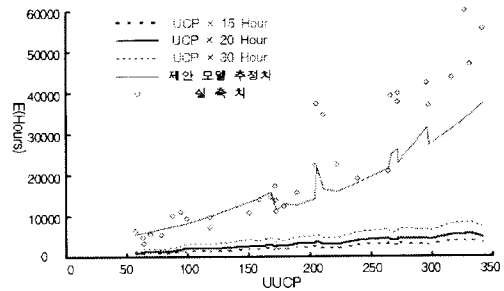


그림 8. 기존연구와 제안모델 및 실측치 비교

또한 제안된 모델이 소프트웨어 개발 노력 추정에 적용될 때 실무 개발 사례의 실측치와 비슷한 결과를 보여 주고 있다.

그럼으로 개발될 소프트웨어 시스템의 UUCP를 계산한 후, 제안된 모델인 지수 함수의 형태를 이용하여 개발노력을 추정하고 개발업체의 시간당 평균 소요 비용을 곱하면 개발에 소요되는 직접비용을 구할 수 있다.

표 6. 추정형태 및 제안된 모델 평가

추정 형태	MMRE
UCP × 15 hour	87.33
UCP × 20 hour	81.90
UCP × 30 hour	72.84
제안 모델(지수)	25.20



따라서 앞 절에서의 Ribu[4]와 Nageswaren[16]의 데이터를 이용한 모델과 실무 사례를 적용한 모델의 모두가 지수형 모델로 좋은 결과를 나타내고 있어 지수형 모델을 제안한다.

다음은 제안된 회귀 모델에 대한 적합성을 평가해 본다.

[표 7]은 지수모델에 대한 회귀분석을 수행한 결과이다.

표 7. 지수모델의 회귀분석 결과

추정 모델	$E = 3352.0832 \cdot e^{0.0035 \cdot UCP}$
다중 상관계수	0.96589
결정계수(R <sup>2</sup> )	0.913726
조정된 결정계수	0.910943
표준오차	4831.646

결정계수(Determinent of Coefficient)는 91%로서 회귀식의 설명력이 91%정도 된다고 말할 수 있다.

[표 7]과 더불어 지수 모델의 분산분석표는 [표 8]에 제시하였다.

표 8. 지수모델의 분산분석표

	자유도	제곱합	제곱평균	F 비	유의한 F
회귀	1	7.63E+ 09	7.63E+ 09	328.3199	4.76E -18
잔차	31	7.21E+ 09	23248266		
계	32	8.35E+ 09			

[표 8]은 지수모델의 분산분석표로서 UUCP를 독립 변수로 하고 Effort를 종속변수로 하였을때 회귀식의 유의성 검증 한 것이다. 유의수준을 0.05를 기준으로 하여 제안된 모델의 유의성 검증결과  $F \geq F(\alpha) = 328.3199 > 4.76E-18$ 로 회귀 분석이 유의함을 알 수 있다. 따라서 회귀식은 UUCP를 이용하여 Effort를 설명할 수 있는 설명력은 충분하다고 판정한다.

다음으로 제안된 모델의 적합성을 검증하기 위해 잔차분석을 하여보자. 잔차(Residual)는 실제 값과 추정된 값과의 차이로, 잔차분석은 단순회귀모델에서 설정한 등분산성, 독립성, 정규성과 직선관계의 가정이 옳은가를 검토할 때 가장 많이 사용되는 방법이다. 따라서

잔차가 어떤 일정한 형태를 취하지 않고 랜덤하게 분산되어 있는 경우가 좋은 모델이 된다. [그림 9]의 표준 잔차 분포는 증가하거나 감소하는 경향을 나타내지 않고 어느 정도는 분산되어 있어 좋은 모델로 판단된다.

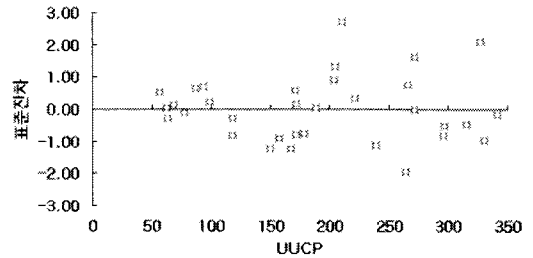


그림 9. X1의 잔차도

## VI. 결론 및 향후 연구과제

소프트웨어 개발에 적용되는 방법론과 언어는 보다 좋은 품질, 보다 빠른 개발과 보다 저렴한 개발비용을 투입하기 위한 방향으로 급격한 변화를 거듭하고 있다. 현재는 구조적 개발 방법론보다는 객체지향 방법론이 소프트웨어 산업계를 주도하고 있다. 소프트웨어 비용 추정 분야도 이러한 변화에 적응하기 위해서는 기존의 구조적 개발방법론에 적합한 라인 수와 기능점수 기반의 추정모델을 적용하기보다는 객체지향 개발방법론에 적합한 사용사례 점수 기반의 비용추정 모델로의 전환이 필요하다. 그러나 이 분야에 대한 연구가 거의 수행되지 않고 있다. 이러한 추세에 부응하고자 본 논문은 사용사례 점수에 기반한 소프트웨어 비용추정 모델을 제시하였다.

본 논문에서는 사용사례점수 계산 방법을 살펴보고, 사용사례점수를 기반으로 하여 개발비용을 추정하는 기존의 선형 모델의 문제점을 제기하였다. 제안된 모델은 최종적으로 계산된 조절된 사용사례점수(UCP)에 기반하지 않고, 조절이 안 된 사용사례 점수(UUCP)에 기반하여 최적의 개발비용을 추정하는 모델을 제시하고 다양한 실측 데이터를 획득하여 이에 적합한 모델을 고찰하였다. 모델의 성능을 평가한 결과 지수형태의 비

선형 회귀모델( $E = a \cdot e^{b \cdot UUCP}$ )이 가장 적합한 모델로 선정되었다.

이 모델은 유스케이스 점수와 개발 노력 관계를 적절히 표현하는 능력과 더불어 잔차에 있어서도 일정한 형태를 취하지 않고 랜덤하게 분산되어 있어 좋은 모델이라 할 수 있다. 소프트웨어 개발 방법론은 전형적인 폭포수 모델에서 반복적이고 점진적인 단일화된 프로세스로 전환되었으며, 현재는 폭포수 모델과 단일화된 프로세스의 엄격한 절차 중심에서 탈피하여 기민한 프로세스(Agile Process)로 발전되고 있는 실정이다. 이에 따라 기술적 요구사항을 표현하는 형태도 유스케이스에서 스토리(Stories)나 형상(Features) 들을 적용하고 있다. 이러한 발전 추세를 반영한 모든 소프트웨어 규모의 정량화 방법에 적용할 수 있는 일반화된 소프트웨어 개발노력 추정 모델 연구가 필요한 시점이다.

#### 참 고 문 헌

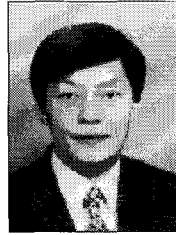
- [1] L. C. Briand, K. E. Elman, D. Surmann, I. Wiczork, and K. D. Maxwell, "An Assessment and Comparison of Common Software Cost Estimation Modeling Techniques," International Software Engineering Research Network, Technical Report, ISERN-98-27, 1998.
- [2] L. C. Briand and I. Wiczorek, "Resource Estimation in Software Engineering," International Software Engineering Research Network, Technical Report, ISERN 00-05, 2000.
- [3] K. Johnson, "Software Cost Estimation: Metrics and Models," Department of Computer Science University of Calgary, Alberta, Canada, <http://sern.ucalgary.ca/courses/seng/621/W98/johnsonk/cost.htm>, 1998.
- [4] K. Ribu, "Estimating Object-oriented Software Projects with Use Cases," University of Oslo Department of Informatics, Master of Science Thesis, 2001.
- [5] J. E. Matson, B. E. Barrett, and J. M. Mellichamp, "Software Development Cost Estimation Using Function Points," IEEE Trans. on Software Eng., Vol.20, No.4, pp.275-287, 1994.
- [6] A. J. Albrecht, "Measuring Applications Development Productivity," Proceedings of IBM Application Dev., Joint SHARE/GUIDE Symposium, Monterey, CA., pp.83-92, 1979.
- [7] I. Jacobson, M. Christerson, et al., "Object-oriented Software Engineering. A Use Case Driven Approach," Addison-Wesley, 1992.
- [8] G. Karner, "Metrics for Objectory," Diploma Thesis, University of Linköping, Sweden, No. LiTH-IDA-Ex-934421, 1993.
- [9] M. Bradley, "Function Point Counting Practices Manual, Release 4.1," International Function Point Users Group(IFPUG), 1999.
- [10] C. Symons, "COSMIC-FFP Measurement Manual, Version 2.1," Common Software Measurement International Consortium, 2001.
- [11] C. Symons, "COSMIC-FFP Measurement Manual, Version 2.2 (The COSMIC Implementation Guide for ISO/IEC 19761:2003)," Common Software Measurement International Consortium, 2003.
- [12] B. Kitchenham and K. Känsälä, "Inter-item Correlation Among Function Points," National Computing Centre Ltd, UK and VTT, Finland, 1997.
- [13] G. Banerjee, "Use Case Points - An Estimation Approach," <http://java.isawix.com/whitepapers/1035194512861.pdf>, 2001.
- [14] L. Probasco, "Dear Dr. Use Case: What About Function Points and Use Cases?,"

[http://www.therationaledge.com/content/aug\\_02/t\\_drUseCase\\_lp.jsp](http://www.therationaledge.com/content/aug_02/t_drUseCase_lp.jsp), Rational Software Canada, 2002.

- [15] C. Larman, "Applying UML and Patterns. An Introduction to Object-oriented Analysis and Design and the Unified Process," Prentice-Hall, 2002.
- [16] S. Nageswaren, "Test Effort Estimation Using Use Case Points," Quality Week 2001, San Francisco, California, USA, 2001.
- [17] A. Abran, C. Symons, and S. Oligny, "An Overview of COSMIC-FFP Field Trial Results," ESCOM 2001, London, England, 2001.
- [18] S. Conte, H. E. Dunsmore, and V. Y. Shen, "Software Engineering Metrics and Models," Benjamin/Cummings., 1986.

양 해 술(Hea-Sool Yang)

정회원



- 1975년 : 홍익대학교 전기공학과 졸업(학사)
  - 1878년 : 성균관대학교 정보처리학과(석사)
  - 1991년 : 日本 오사카대학 정보공학과 S/W공학전공(공학박사)
  - 1975년 ~ 1979년 : 육군중앙경리단 전산장교
  - 1980년 ~ 1995년 : 강원대학교 전자계산학과 교수
  - 1986년 ~ 1987년 : 日本 오사카대학 객원연구원
  - 1995년 ~ 2002년 : 한국S/W품질연구소 소장
  - 1999년 ~ 현재 : 호서대학교 벤처전문대학원 교수
- <관심분야> : 소프트웨어공학(특히, S/W 품질보증과 평가, 품질감리, 프로젝트관리, CBD기반기술, IT품질경영

저 자 소 개

박 주 석(Ju-Seok Park)

정회원



- 1984년 : 해군사관학교 전자공학과 졸업(공학사)
  - 1995년 : 국방대학원 전산계산학과(전산학석사)
  - 2004년 : 숭실대학교 컴퓨터학과(공학박사)
  - 1995년 ~ 2000년 : 해병대사 C4I과장
  - 2001년 ~ 2006년 : 국방대학교 직무연수부 교수
  - 2007년 ~ 현재 : 근로복지공단 정보시스템실 팀장
- <관심분야> : 비용산정, 표준 및 프로세스, 프로젝트관리, 경영정보시스템, 소프트웨어 품질보증