성장곡선을 이용한 소프트웨어 비용 추정 모델

박 석 규[†] · 이 상 운[††] · 박 재 흥[†††]

요 약

정확한 소프트웨어 비용 추정은 개발자와 고객 모두에게 중요하다. 대부분의 비용 추정 모델들은 규모 추정으로부터 얻은 라인 수와 기능점수와 같은 규모 측도에 기반을 두고 있다. 규모 추정의 정확도는 비용 추정 정확도에 직접적으로 영향을 미친다. 이에 따라 대부분의 회귀기반 비용추정 모델들은 규모에 기반한 멱함수 형태를 적용하고 있다. 생물의 성장, 기술의 발전과 인간의 학습 능력 등 많은 성장 현상들은 S자 곡선을 따른다. 본 논문은 성장곡선을 이용하여 개발노력을 추정하는 모델을 제시하였다. 제시된 모델은 소프트웨어 규모가 증가함에 따라 소요되는 개발 비용이 성장곡선을 따른다고 가정한다. 일반적인 소프트웨어 규모 추정 기법인 기능점수, 완전기능점수와 유스케이스 점수에 기반하여 성장곡선 모델의 적합성을 검증하였다. 제안된 성장곡선 모델들은 멱함수 모델과 비교시 상호 견줄만한 성능을 보여 소프트웨어 비용 추정 분야에 적용 가능함을 보였다.

A Software Cost Estimation Using Growth Curve Model

Seok-Gyu Park[†] · Sang-Un Lee[††] · Jae-Heung Park[†††]

ABSTRACT

Accurate software cost estimation is essential to both developers and customers. Most of the cost estimating models based on the size measure methods, such as LOC and FP, are obtained through size estimation. The accuracy of size estimation directly influences the accuracy of cost estimation. As a result, the overall structure of regression-based cost models applies the power function based on software size. Many growth phenomenon in nature such as the growth in living organism, performance of technology, and learning capability of human show an S-shaped curve. This paper proposes a model which estimates the developing effort by using the growth curve. The presented model assumes that the relation cost and size follows the growth curve. The appropriateness of the growth curve model based on Function Point, Full-Function Point and Use-Case Point, which are the general methods in estimating the software size have been confirmed. The proposed growth curve model shows similar performance with power function model. In conclusion, the growth curve model can be applied in the estimation of the software cost.

키워드 : 규모(Size), 노력(Effort), 성장곡선(Growth Curve), 시그모이드 함수(Sigmoidal Function), 기능 요구사항(Functional Requirements)

1. 서 론

소프트웨어 개발계획 작성 단계에서 프로젝트 개발에 소요되는 인력, 비용과 개발기간 등의 신뢰할 만한 수준의 정보 제공은 사업의 성공 여부를 결정하는데 필수적이다. 소프트웨어 개발비용 추정 분야는 비용에 영향을 미치는 잠재적인 다양한 비용 인자(Cost Factor)들을 이용해 소프트웨어 개발비용을 설명하기 위한 많은 방법들이 연구되고 실제 적용되고 있다.

전형적인 비용 추정 모델들은 과거 소프트웨어 프로젝트들로부터 수집된 데이터의 회귀분석을 통해 유도된다. 여러 다양한 프로젝트들에 대한 개발 노력은 주요 비용 인자에 대해 그려지고 최적으로 적합한 직선이 데이터들 사이에서 계산된다. 만약 주요 비용인자가 개발 노력의 완전한 예측이 되면 그래프 상의 모든 점들은 최적으로 적합된 직선상에 위치한다. 그러나 실제로는 일반적으로 상당한 오차를 초래한다. 그러므로 예측 값과 실제 개발 노력 간 편차의 원인이 되는 요인을 식별하는 것이 필요하다. 이들 요인들은 비용 인자들로 모델에 추가된다[1]. 개발 비용에 영향을 미치는 주요 인자들로는 규모, 복잡도와 개발팀의 능력이며, 이들 중에서 비용 추정 정확도에 직접적인 영향을 미치는 것이 개발될 소프트웨어의 규모이다. 따라서, 대부분의 경험적 연구들은 개발 노력을 곱셈 항 a와 지수 항 b를 가진 규모 S의 멱 함수인 $E = aS^b$로 표현하여 해답을 찾고자 하였다.

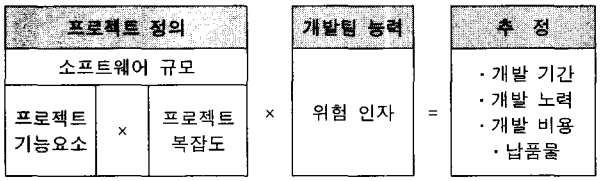
† 정 회 원 : 도립 상원전문대학 컴퓨터응용과 조교수
†† 정 회 원 : 국립 원주대학 여성교양과 전임강사
††† 정 회 원 : 성상대학교 컴퓨터과학과 교수
논문접수 : 2004년 2월 18일, 심사완료 : 2004년 3월 25일

비록 대부분의 연구자와 개발자들은 규모가 노력의 주요 결정인자라는 데는 동의하고 있지만 규모와 노력간의 정확한 관계는 불명확하다[1]. 따라서, 소프트웨어 측정분야는 30년 이상 수많은 연구가 있어 왔으나 소프트웨어 개발노력과 비용에 영향을 미치는 다양한 속성들과 이들간의 불명확한 관계로 아직까지 구체적인 소프트웨어 비용 추정 모델이 거의 없는 실정이다[2]. 본 논문은 소프트웨어 규모가 증가함에 따라 개발노력이 어떤 함수 형태를 취하는지를 적절히 표현할 수 있는지를 살펴보고, 이에 적합한 함수를 연구하였다. 적합한 함수를 찾기 위해 대부분의 분야에서 성공적으로 적용하고 있는 성장곡선을 도입하였다. 성장곡선은 인구 증가, 생물의 성장과정, 건축과정의 투입인력 분포 또는 기술 습득 학습 효과 등을 적절히 표현하고 있다. 이와 같은 적용 사례들로부터, 소프트웨어의 규모와 개발노력의 관계도 성장곡선 형태를 따를 수 있다는 가정하에 본 논문은 성장곡선이 소프트웨어 개발노력을 추정하는 분야에도 적용될 수 있는지를 증명하고자 한다.

2장에서는 소프트웨어 규모 추정 방법과 비용추정 모델의 형태를 살펴본다. 3장에서는 성장곡선을 살펴보고, 4장에서는 기능점수로 계산된 소프트웨어 규모에 기반한 비용추정 분야에 적용할 수 있는지를 검증한 이후 완전기능점수와 유스케이스 점수로 계산되는 소프트웨어 규모측정 분야에 대해서도 성장곡선으로 비용을 추정할 수 있는지를 평가하여 제안된 모델을 일반적으로 적용할 수 있음을 보인다.

2. 관련 연구

추정을 하는데 있어서 가장 핵심이 되는 사항은 (그림 1)과 같이 가능한 빨리 개발될 소프트웨어의 문제 영역(Software Problem Domain)에 대한 정의(Definition)와 특정 환경에 요구되는 소프트웨어를 개발할 수 있는 개발팀의 능력(Capability)을 이해하여야 한다[3]. 이에 근거하여 프로젝트 개발에 요구되는 노력, 비용 또는 기간 등을 보다 정확히 예측할 수 있다. 문제영역 정의는 프로젝트의 규모(Size)와 복잡도(Complexity)로 평가되는 소프트웨어의 범위(Scope)로 정의될 수 있다.



(그림 1) 추정 원칙

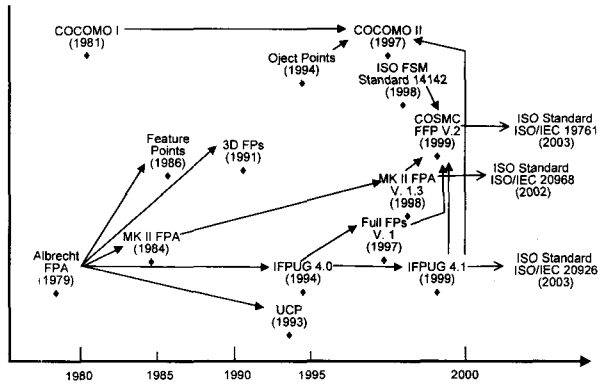
개발 능력은 적시에 경제적 효과를 가진 소프트웨어를 납품할 수 있는 개발 조직의 능력에 영향을 미치는 다양한 위험인자(Risk Factor)에 의존한다. 이들 위험인자들로는 사용되는 소프트웨어 프로세스, 개발조직의 숙련도 수준, 자동화, 개발 조건과 비즈니스 환경의 영향 등이 있다. 따라서, 실질적인 추정 모델은 규모, 복잡도와 위험 인자의 3가지 요소를 고려해야 한다. 이들 3가지 요소 중 비용 추정 정확도에 직접적인 영향을 미치는 것이 규모이다. <표 1>에서 알 수 있듯이 지금까지 소프트웨어의 규모를 추정하는 많은 방법들이 제안되었다.

〈표 1〉 규모 추정 기법

측정 기법		프로젝트 규모		프로젝트 복잡도	개발팀 능력 (위험인자)
		기 능 요 소	복잡도		
COCOMO I		라인 수	×		Cost Factor
구조적	기능점수 (FP)	· 입 력 · 출 력 · 조 회 · 파 일 · 인터페이스	○	일반적인 시스템 특성 (14가지)	-
	3차원 기능점수 (3D FP)	· 데이터영역(정보/비지니스 시스템) : 기능점수 기법 적용 · 기능 영역(과학/공학 시스템) : 처리 스탭 수 + 의미문장 수 · 제어 영역(실시간시스템) : 상태모델에서의 전이 수	○	-	-
	완전 기능점수 (FFP)	· 입구(입력) · 출구(출력, 조회) · 읽기(파일) · 쓰기(파일)	×	-	-
	특징점수 (FePs)	· 입 력 · 출 력 · 조 회 · 파 일 · 인터페이스 · **알고리즘**	○	-	-
	객체점수 (Object Point)	· 화 면 · 보고서 · 3GL 컴포넌트	○	-	개발조직의 능력
	COCOMO II	· 기능점수 · 특징점수	○	-	Cost Factor
객체지향	유스케이스 점수 (UCP)	· 액 터 · 트랜잭션(or클래스)	○	기술 복잡도	환경 복잡도
	MK II FP	· 입 력 · 출 력 · 조 회 · 파 일 · 인터페이스	○	일반적인 시스템 특성 (20가지)	-

소프트웨어 프로젝트의 규모는 길이, 기능성 또는 복잡도의 속성들로 표현될 수 있다[1,4-6]. 길이에 기반한 소프트웨어 규모 추정 방법은 라인 수(Line Of Code, LOC)로 대표적으로 COCOMO 모델[7]이 있다. 대부분의 규모추정 방법들은 사용자의 기능 요구사항(Functional User Requirements, FUR)에 기반을 두고 기능성과 복잡도를 모두 고려하고 있다. 기능에 기반한 규모 추정방법으로는 기능점수

(Function Point, FP), 완전 기능점수(Full Function Point, FFP), COCOMO II, 유스케이스 점수(Use Case Point, UCP), 특징점수(Feature Point, FePs)와 객체점수(Object Point, OP) 등이 있다[8-13]. 이들 방법 중 실제로 라인 수와 기능점수 방법이 가장 많이 적용되고 있다[14]. (그림 2)는 소프트웨어 규모추정 기법들의 변천 과정을 기술하고 있다.



(그림 2) 소프트웨어 규모추정 기법 변천과정

Leung과 Fan[15]은 "소프트웨어 비용(Cost) 추정은 소프트웨어 시스템을 개발하는데 요구되는 노력(Effort)을 예측하는 과정이다"라고 정의하고 있다. 대부분의 비용 추정 모델은 노력 추정을 산출하려고 시도한다. 왜냐하면 개발 노력은 개발 기간과 비용으로 변환될 수 있기 때문이다. 따라서, 이후부터는 노력 추정을 비용 추정이라 칭한다.

지금까지 많은 정량적인 소프트웨어 비용추정 모델들이 개발되었으며, 알고리즘 모델과 비알고리즘 모델로 구분된다. 비알고리즘 모델들은 유추 원가계산(Analogy Costing), 전문가 감정(Expert Judgement), 파킨슨, 가격 경쟁 방법(Price-to-Win), 상향식(Bottom-Up)과 하향식(Top-Down) 등의 방법이 있다[7]. 알고리즘 모델은 다양한 변수들의 함수로 비용을 추정할 수 있는 수학적 모델에 기초하고 있다. 현존하는 알고리즘 모델은 소프트웨어의 노력을 식 (1)의 형태로 표현한다. 여기서 E는 노력이며, $x_1, x_2, \cdots, x_n$은 비용 인자이다. 즉, 종속변수인 개발 노력은 독립변수인 비용 인자들에 어떠한 함수를 취한 형태를 따른다.

$$E = f(x_1, x_2, \cdots, x_n) \quad (1)$$

알고리즘 모델들은 비용인자와 함수 f의 형태를 어떤 것으로 선택하는가에 따라 구분될 수 있다. 비용인자는 전형적으로 주요 원가 요인(Cost Factor)인 규모와 다양한 부수적인 조절 인자로 원가 유발요인(Cost Driver)을 가진다. 원가 유발요인은 개발노력에 영향을 미치는 프로젝트, 프로세스, 제품 또는 자원의 특성들로 대표적으로 COCOMO II[10]에서 고려되고 있다.

함수 f의 형태는 소프트웨어의 규모가 증가함에 따라 소요되는 노력의 양이 어떠한 관계를 갖는지를 표현한다. 함수 f의 형태는 일반적으로 다음 5가지 범주로 설명될 수 있다. 여기서 S는 소프트웨어 규모이며, a, b, c, d는 상수이다.

- 선형함수 모델 : $E = aS + b$
- 로그함수 모델 : $E = a\ln(S) + \mathrm{b}$
- 다항식 모델 : $E = a + bS + cS^2 + \cdots$
- 멱 함수 모델 : $E = aS^b$

  $\log(E) = blog(S) + \log(a)$
- 지수함수 모델 : $E = ae^{bS}$

  $\ln(E) = bln(S) + \ln(a)$

Johnsen[16]이 제시한 <표 2>에서 보는 바와 같이 대부분의 경험적 연구들은 개발 노력을 곱셈 항 a와 지수 항 b를 가진 규모의 멱 함수인 $E = aS^b$로 표현한다.

〈표 2〉 소프트웨어 비용추정 모델 사례

규모	모델명	모델형태	사 례
LOC	Bailey-Basili	멱 함수	$E = 5.5 + 0.73KLOC^{1.16}$
	Walston-Felix	멱 함수	$E = 5.2KLOC^{0.91}$
	Boehm's COCOMO Basic	멱 함수	$E = 3.2KLOC^{1.05}$
	Doty	멱 함수	$E = 5.288KLOC^{1.047}$
	Nelson	멱 함수	$E = 4.0KLOC^{0.98}$
	Freburger-Basili	멱 함수	$E = 1.48KLOC^{1.02}$
	Herd	멱 함수	$E = 5.3KLOC^{1.06}$
FP	Albrecht-Gaffney	선 형	$E = -12.39 + 0.0545FP$
	Kemerer	다항식	$E = 60.62 + 7.72810^{-8}FP^3$ $E = -69.13 + 0.72FP - 0.0008FP^2$
	Matson. et al.	지 수	$\ln(E) = 2.51 + 1.00\ln(FP)$

3. 성장 곡선

멱 함수 모델들은 노력이 규모에 비례한다고 가정하여 지수항 b를 조절 인자로 표현하고 있다. 따라서 보다 큰 프로젝트는 작은 프로젝트보다 많은 노력이 요구된다. 직관적으로 볼 때 이것은 이치에 맞는다. 그러나 보다 큰 프로젝트는 증가하는 복잡도를 다루는데 보다 많은 노력이 요구되는 것 같아 보이나 실제적으로 이를 뒷받침할만한 지원하는 증거가 거의 없다. 그러면 소프트웨어 규모가 증가함에 따라 소요되는 개발 노력은 어떤 함수 관계를 취하는

가? 본 장에서는 이러한 경향을 적절히 표현할 수 있는 함수를 선택해 적용 가능성을 평가하고자 한다.

성장 곡선(Growth Curve)을 일명 S자(S-shaped) 곡선, 시그모이드(Sigmoidal) 곡선 또는 학습 곡선(Learning Curve)이라 부르며 생명체의 성장과 기술 성취도와의 닮은 점을 표현하고 있다[17]. 현존하는 많은 성장 현상은 초기에는 점진적으로 증가하다가 중간 구간에서 보다 빠르게 증가하고 한계점으로 도달하면서 성장 속도가 늦쳐지며 일정한 시간 구간 후에는 최대 값에서 수평이 되는 S자 패턴을 보이고 있다.

성장곡선을 표현하기 위해 다양한 수학적 함수가 제안되었다. 대칭 로지스틱 성장 곡선, 일반화된 로지스틱 곡선, Pearl 곡선과 Gompertz 곡선 등 대표적인 성장곡선을 살펴본다.

대칭 로지스틱 성장곡선은 생물체의 성장 패턴을 모형화하는데 많이 사용되고 있으며 식 (2)로 표현된다.

$$y = \frac{K}{(1+e^{a+bX})} \tag{2}$$

여기서, a와 b는 계수(인자 또는 모수)로 함수의 모양과 크기를 결정하며, b는 음수이다. K는 환경에 대한 적재량 또는 부양능력(Carrying Capacity)이라 한다. 이 함수는 중간지점인 $K/2$를 중심으로 상호 대칭이 되며, $K/2$ 지점부터 성장률이 감소하기 시작하지만 최대 값 점근선에 도달할 때까지 계속 성장한다.

일반화된 로지스틱 곡선은 Richard 곡선이라고도 불리우며, 성장을 모형화하는데 널리 사용되고 유연한 함수이다. 이 함수는 식 (3)으로 표현된다.

$$y = a\frac{K}{(1+te^{-b(x-M)})^{1/t}} \tag{3}$$

여기서, y는 무게, 키, 규모 등이 되며, x는 시간이다. a는 하한 점근선, K는 상한 점근선, M은 최대 성장시간을, b는 성장률을 결정한다. 또한 t는 최대 성장이 발생하는 점을 결정한다.

Pearl 곡선은 미국 통계학자인 Raymond Pearl[18]이 제안한 것으로 인구예측에 사용하였으며, 로지스틱 곡선으로 잘 알려져 있다. 표준화된 Pearl 곡선은 식 (4)의 형태를 취한다.

$$y = \frac{K}{1+ae^{-bt}} \tag{4}$$

여기서 K는 변수 y의 상한 값이며, t는 시간, 계수 a는 곡선의 위치를 결정하며, 계수 b는 모양을 결정한다. 이 함수는 시간이 $-\infty$에서 초기값 0을, 시간이 $+\infty$에서는 K 값을 가진다. 만약 초기 값이 0이 아니면 상수로서 위 공식에 더해진다. $t=\ln(a)/b$ 시점에서 $y=K/2$일 때 곡선이 굴곡된다. 따라서, 이 곡선은 $y=K/2$가 중간지점이 되므로 굴절되는 위쪽 반과 아래쪽 반이 대칭이 된다.

Gompertz 곡선은 영국의 공증인이자 수학자인 Benjamin Gompertz[19]에 의해 제안되었으며, 식 (5)로 표현된다.

$$y = Ke^{-ae^{bt}} \tag{5}$$

여기서 y는 성취도를 표현하는 변수이며, K은 상한 값, t는 시간, a와 b는 곡선이 데이터에 적합되어 얻어지는 계수이다. 이 곡선은 시간이 $-\infty$에서 초기값 0을, 시간이 $+\infty$에서는 K 값을 가진다는 점에서 Pearl 곡선과 닮았지만, 곡선이 굴절되는 지점은 $y=K/e$가 되는 $t=\ln(a)/b$ 시점으로 비대칭이 된다.

성장곡선을 이용하기 위해서는 다음 가정에 기반하여야 한다 : ① 성장 곡선의 상한 값을 알고 있어야 한다 ; ② 선택된 성장곡선이 수집된 과거 이력 데이터에 적합한 것이어야 한다 ; ③ 수집된 이력 데이터는 선택된 성장 곡선 함수의 계수를 적절히 결정하게 해주어야 한다. 이러한 가정에 기반하여 성장곡선을 이용하여 미래의 행위를 예측하는 과정은 다음과 같다 : ① 성장 곡선 모델이 주어진 데이터에 적절한지 결정한다 ; ② 적절한 성장곡선 모델을 선택한다(Gompertz 또는 Pearl 곡선) ; ③ MSE(Mean Squared Error)를 최소화하거나 변수변환으로 얻는 선형회귀 방정식을 이용하여 계수를 결정한다.

기술 예측에 가장 많이 적용하고 있는 성장곡선은 Pearl 곡선과 Gompertz 곡선이다. 이 두 곡선 중에서 적절한 것을 선택하는 기준은 다음 조건에 기반한다 : ① 얻고자 하는 데이터의 성장 곡선이 대칭인가? 만약 대칭이 될 경우에는 Pearl 곡선을, 비대칭일 경우에는 Gompertz 곡선을 선택한다 ; ② 성장과정에서 이미 오프셋 (Offset) 요인이 있는가? 만약 오프셋 요인이 있는 경우에는 Pearl 곡선을, 오프셋 요인이 없으면 Gompertz 곡선을 선택한다.

4. 성장곡선 소프트웨어 비용 추정

Wideman[20]는 건축공학에서의 경험에 의하면 전형적인 누적 투입 인력과 공사의 진척도는 S자 곡선(S-Shaped Curve)을 나타내며, 차이점은 학습효과에 기안한다고 제안하였다. 또한, Gendall[21]은 시간에 따른 누적 투입 인시수, 투입 비용과 프로젝트 진척도가 모두 S자 형태를 따른다고 제안하였다. 이와 같이 프로젝트 관리 분야에 S자 성장곡선이 널리 적용되고 있다. 또한, Shigeru[22]는 소프트웨어 신뢰성 측정 분야에 S자 곡선을 적용하였다. 소프트웨어 개발분야에서도 소프트웨어 규모가 증가함에 따라 소요되는 개발 노력이 선형적, 기하급수적 또는 멱함수 적으로 증가하지는 않는다. 이 분야에 대한 관계를 S자 성장곡선이

적절히 표현할 수 있는지를 살펴본다.

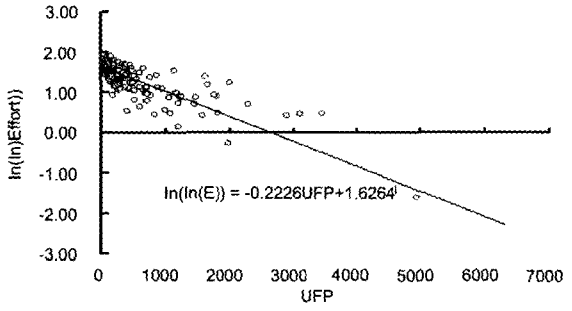
모델 평가 기준으로 결정계수($0 \le R^2 \le 1$), MMRE(Mean Magnitude of Relative Error)와 표준편차(Standard Deviation)를 적용하였다. 주어진 데이터들의 변동을 적절히 표현할 수 있는 모델을 찾기 위해 결정계수를 이용한다. 종속변수의 값(노력)은 독립변수의 값(소프트웨어 규모)에 의해 결정되는 부분과 미지의 오차의 합으로 나타나며, 총 변동(주어진 데이터의 실제 노력 값) 중에서 회귀모델에 의해 설명되는 변동 비율이 결정계수이다. 결정계수 값이 클수록 쓸모 있는 회귀직선인 모델이 얻어진다. 비록 결정계수에 의해 좋은 모델이 얻어졌더라도, 주어진 데이터들이 값이 큰 부분에 약간의 이상점들이 분포한 상태에서 회귀직선이 이 이상점들을 잘 표현하고 값이 작은 표본에 대해서는 표현을 하지 못할 경우에도 결정계수는 큰 값을 나타낼 수 있다. 이 경우 좋은 모델이라고 판단할 수 없다. 따라서, 모델 정확도를 평가하는 측도로 MMRE가 적용되며, 다음과 같이 측정된다. $\mathrm{RE} = \frac{\text{추정치} - \text{실측치}}{\text{실측치}}$, MRE(Magnitude of RE) $= |RE|$, MMRE(Mean MRE) $= 100/n * \sum_{i=1}^{n} MRE$. MMRE가 작은 값이면 모델은 평균적으로 좋은 모델임을 알 수 있다. 표준편차는 전체 평균으로부터 얼마나 떨어져 있는가를 나타내며 $\sqrt{\frac{n\sum x^2 - (\sum x)^2}{n(n-1)}}$ 식으로 계산된다. 제안된 성장곡선 모델은 대표적인 기존 비용 추정 모델인 먹 함수 모델과 비교한다. 모델 평가에 사용된 자료는 소수의 이상점들이 존재한다. 모델의 정확도는 이상점들을 잘 설명하는 모델이 좋은 평가를 받을 수 있다. 그러나 대부분의 관심사는 이상점들을 제외하고 대부분의 데이터가 분포하고 있는 부분이며, 이를 잘 설명하는 모델이 중요할 수 있다. 따라서, 모델을 평가함에 있어 이상점들을 포함하는 경우와 이상점들을 제거한 경우로 구분하여 모델의 적합성을 살펴본다.

4.1 기능점수 기반 비용 추정 성장곡선 모델

기능점수로 측정된 ISBSG Release 6[23]의 789개 프로젝트 데이터베이스에서 계획 - 명세화 - 구축 - 시험 - 구현 단계를 모두 수행한 221개 프로젝트를 발췌하여 분석하였다. 이들 데이터는 다양한 개발언어와 환경, 다양한 나라에서 개발된 프로젝트들로 개발 생산성에도 많은 차이를 보이고 있어 일반화된 개발비용 추정 모델을 얻기에는 적합한 표본이라 할 수 있다.

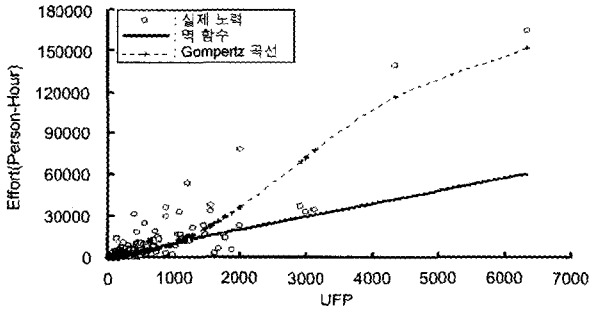
먼저, 이상점들을 포함하는 경우를 살펴보자. 성장곡선을 적용하기 위한 전제조건을 충족시키기 위해 개발 노력의 상한 값은 주어진 데이터로부터 17,000으로 결정하였다. 다음으로 적절한 성장곡선을 선택하기 위해 대칭 유무를 검토한 결과 비대칭이 되었다. 또한 오프셋 값을 갖고 있지 않기 때문에 Pearl 곡선 대신 Gompertz 곡선을 선택할 수 있다. Gompertz 곡선의 계수는 식 (6)에 의해 (그림 3)과 같이 변수변환을 거친 후 통계 패키지를 이용한 선형회귀분석으로 얻어진다.

$$\ln(\ln(K/y)) = \ln a - bt \tag{6}$$

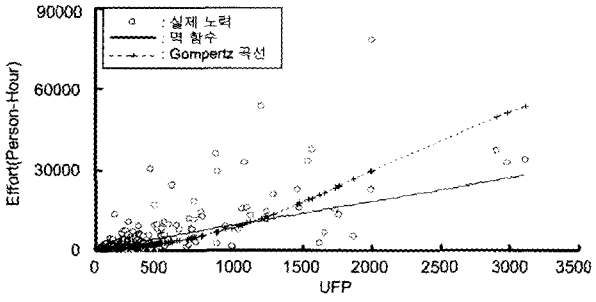


(그림 3) Gompertz 곡선의 계수 추정

(그림 3)으로부터 $a = \exp(\ln a) = \exp(1.6264) = 5.0855$를 구하여 각 계수를 식 (5)에 대입하면 개발비용을 추정할 수 있다. 이와 같이 추정된 값과 실제 값을 (그림 4)(a)에 나타내었다. 이상점들을 제거시킬 경우 대부분의 데이터들에도 제안된 모델이 적합한지를 판단하기 위해 동일한 방법으로 모수를 추정하고 실제 데이터와 추정 데이터를 (그림 4)(b)에 제시하였다.



(a) 이상점 포함시



(b) 이상점 제거시

(그림 4) 기능점수 기반 개발노력 추정

이상점을 포함하는 경우, 먹 함수 모델은 UFP가 2,000과

3,000 근방의 이상점 들에는 적합하나 4,000 이상인 영역에서는 개발노력을 거의 표현하지 못하는 단점을 갖고 있다. 이에 비해 Gompertz 곡선 모델은 UFP가 이상점으로 판단되는 2,000과 3,000 부근의 소수의 데이터를 제외하고는 전반적으로 모든 데이터에 적합한 모델로 판단된다. 이상점을 제외시킨 경우에도 Gompertz 성장곡선 모델이 전체 데이터를 보다 잘 표현하는 것을 알 수 있다.

모델의 성능을 알아보기 위해 멱 함수와 Gompertz 곡선 함수의 결정계수, MMRE와 표준편차를 이상점 포함시와 이상점 제거시로 구분하여 <표 3>에 제시하였다.

〈표 3〉 기능점수 기반 개발노력 추정 모델 성능비교

(a) 이상점 포함시

함수형태	모 델	결정계수	MMRE	표준편차
멱 함수	$E = 12.4577 UFP^{0.9706}$	0.7312	88%	8,788.23
성장곡선	$E = 170000 e^{-5.0855 e^{-0.0003 UFP}}$	0.7685	86%	8,282.37

(b) 이상점 제거시

함수형태	모 델	결정계수	MMRE	표준편차
멱 함수	$E = 14.6326 UFP^{0.9397}$	0.4528	80%	7069.203
성장곡선	$E = 61000 e^{-4.8538 e^{-0.0005 UFP}}$	0.5209	77%	6813.964

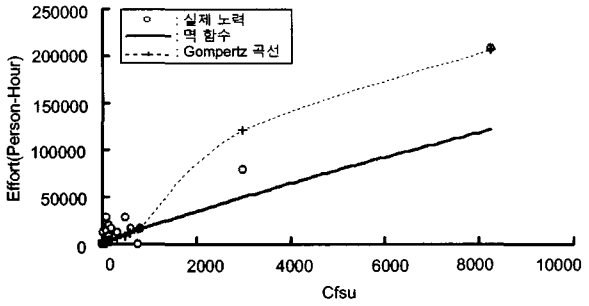
두 경우 모두 Gompertz 곡선 모델이 결정계수와 MMRE 측면에서 멱 함수 모델 보다 좋은 성능을 보이고 있다. 따라서, 성장곡선 모델도 멱함수 모델과 같이 소프트웨어 비용추정 분야에 적용할 수 있을 것이다.

지금까지 S자 성장곡선을 소프트웨어 비용추정 분야에 적용할 수 있음을 살펴보았다. S자 성장곡선을 소프트웨어 개발비용 추정 분야에 일반적으로 적용할 수 있는지를 알아보기 위해 완전 기능점수와 유스케이스 점수로 측정된 소프트웨어 규모 데이터에 적용하여 본다.

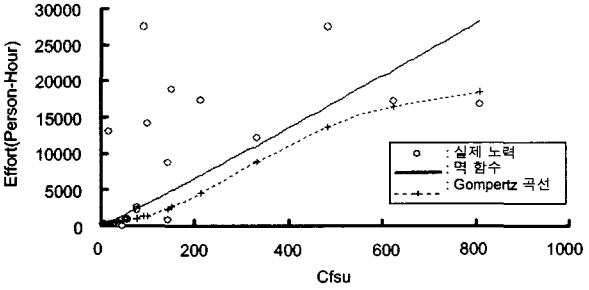
4.2 완전기능점수 기반 비용 추정 성장곡선 모델

완전기능점수는 최근들어 현장 검증 시험이 완료되고 적용되기 시작한 방법으로 실제 개발 데이터들이 극소수에 불과하다. Abran et al.[24]과 Moris[25]의 연구 결과로부터 25개 데이터를 획득하였다. Abran et al.[24]의 데이터는 12개의 실시간 소프트웨어 프로젝트들로 3개 국가의 3개 개발회사로부터 1999년부터 2000년 기간동안 개발이 완료된 프로젝트들이다. 12개의 프로젝트들 중 2개는 유지보수 프로젝트이며, 나머지 10개는 신규로 개발된 프로젝트들이다. Moris[25] 데이터는 4개 개발조직에서 개발된 13개 프로젝트로 개발기간은 5개월에서 75개월이 소요되었으며 C++, C, Ada, VB6 언어들로 개발되었다.

이들 25개 데이터에 대해 완전기능점수의 Cfsu에 따른 개발 노력 관계를 그래프로 그리고 멱 함수와 Gompertz 곡선 함수 모델을 비교한 결과는 (그림 5)에 나타내었다.



(a) 이상점 포함시



(b)이상점 제거시

(그림 5) 완전기능점수 기반 개발노력 추정

25개 데이터 중 대부분이 Cfsu가 1000 미만이며, 2개 데이터가 2,000과 8,000 Cfsu에 위치하여 이상점으로 취급될 수 있다. 이상점 포함시 이상점 2개 데이터에 대해 멱 함수 모델은 거의 표현을 하지 못하고 있는 반면 Gompertz 곡선 함수는 8,000 Cfsu 근방의 데이터는 잘 표현하는 특징을 갖고 있다. 이상점 제거시 데이터들이 크게 산포된 형태를 나타내고 있다. 이러한 경우에도 Gompertz 곡선 함수는 데이터들을 보다 잘 표현하는 능력을 갖고 있음을 알 수 있다. 이상점 포함시와 이상점 제거의 경우 각각에 대한 모델의 성능은 <표 4>에 제시하였다.

〈표 4〉 완전기능점수 기반 개발노력 추정 모델 성능비교

(a) 이상점 포함시

함수형태	모 델	결정계수	MMRE	표준편차
멱 함수	$E = 45.0184 Cfsu^{0.8788}$	0.9535	552%	9,239.66
성장곡선	$E = 210000 e^{-4.5078 e^{-0.0007 Cfsu}}$	0.9252	784%	11,719.09

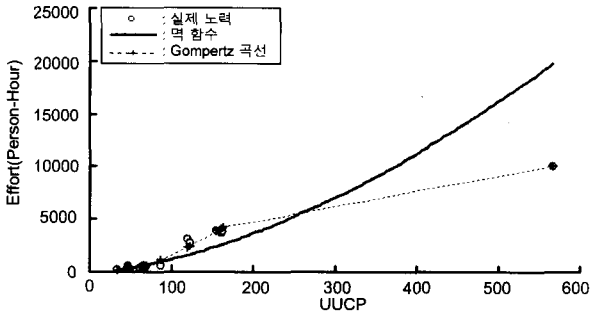
(b) 이상점 제거시

함수형태	모 델	결정계수	MMRE	표준편차
멱 함수	$E = 24.9262 Cfsu^{1.0507}$	0.2966	176%	6276.560
성장곡선	$E = 20000 e^{-4.2891 e^{-0.0007 Cfsu}}$	0.3079	95%	4595.878

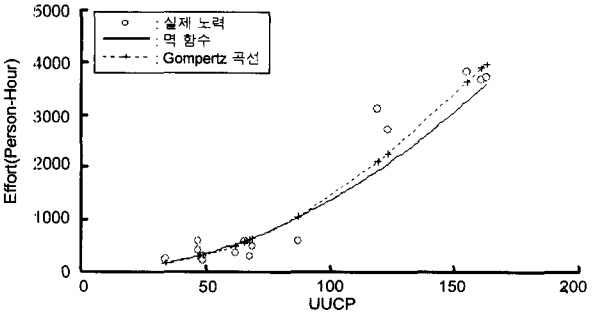
이상점을 포함하는 경우 데이터들이 균등하게 분포하지 못한 형태를 취한다. Gompertz 곡선 모델이 이상점들은 잘 표현하지만 대부분의 데이터들이 모여있는 작은 소프트웨어 규모 부분을 적절히 표현하지 못하여 결정계수와 MMRE 측면에서 모두 멱 함수 모델보다 성능이 저하된 결과를 보이고 있다. 그러나 이상점들을 제외시에는 Gompertz 곡선 모델이 보다 적합한 것으로 판명되었다.

4.3 유스케이스점수 기반 비용 추정 성장곡선 모델

유스케이스 점수도 실존 데이터가 많지 않은 실정이다. 모델 제시를 위해 Ribu[11]와 Nageswaren[26]의 연구결과로부터 15개의 데이터를 획득하였다. 획득된 데이터에 대해 멱 함수와 Gompertz 곡선을 적용하여 개발노력을 추정한 결과는 (그림 6)에, 모델의 성능은 <표 5>에 제시하였다.



(a) 이상점 포함시



(b) 이상점 제거시

(그림 6) 유스케이스점수 기반 개발노력 추정

〈표 5〉 유스케이스 점수 기반 개발노력 추정 모델 성능비교

(a) 이상점 포함시

함수형태	모 델	결정계수	MMRE	표준편차
멱 함수	$E = 0.6567 UUCP^{1.6269}$	0.8233	43%	1,090.624
성장곡선	$E = 10100e^{-6.5588e^{-0.0181UUCP}}$	0.9808	30%	359.814

(b) 이상점 제거시

함수형태	모 델	결정계수	MMRE	표준편차
멱 함수	$E = 0.1391 UUCP^{1.9952}$	0.9314	34%	390.416
성장곡선	$E = 11000e^{-6.1117e^{-0.0118UUCP}}$	0.9371	32%	373.849

이상점들을 포함시키는 경우 완전 기능점수 데이터들과 마찬가지로 데이터들이 균등하게 분포하지 못한 점도 있음에도 불구하고 Gompertz 곡선 모델이 결정계수와 MMRE 측면에서 모두 멱 함수 모델보다 성능이 향상된 결과를 보이고 있다. 이상점들을 제거시에는 두 모델 모두 주어진 데이터들을 잘 표현하고 있으며, Gompertz 곡선 모델이 보다 좋은 성능을 보이고 있다.

지금까지 살펴본 바와 같이, 다양한 소프트웨어 규모 측정 방법에 따른 소프트웨어 개발비용 추정에 있어 Gompertz 곡선 모델이 멱함수 모델보다 우수한 성능을 보였다. 따라서, 성장곡선 모델도 일반적인 멱함수 모델과 더불어 소프트웨어 비용추정 분야에 적용할 수 있음을 알 수 있다.

5. 결론 및 향후 연구과제

소프트웨어 측정분야는 30년 이상 수 많은 연구가 있어 왔으나 소프트웨어 개발노력과 비용에 영향을 미치는 다양한 속성들과 이들간의 불명확한 관계로 아직까지 구체적인 소프트웨어 비용 추정 모델이 거의 없는 실정이다.

본 논문은 소프트웨어 개발 노력이 규모에 따라 어떠한 관계를 가지는지를 살펴보고 이 분야에도 일반적인 성장곡선이 적용 가능한지를 실험하였다. 모델 검증은 일반적으로 적용되고 있는 소프트웨어 규모 추정 기법들인 기능점수, 완전기능점수와 유스케이스 점수에 기반한 소프트웨어 개발 노력을 대상으로 하였다. 제안된 성장곡선 모델은 소프트웨어 비용 추정 분야에 일반적으로 적용되고 있는 멱 함수 모델과 견줄만한 성능을 보이고 있어 제시된 모델의 적합함이 검증되었다.

그러나 주어진 데이터들이 개발노력과 개발기간이 선형적 관계를 갖지 못한다. 이는 개발 팀의 규모와 생산성에 따라 큰 영향을 받기 때문이다. 본 논문은 이러한 점을 고려하지 못하여 일반화된 모델을 유도하지는 못하였다. 따라서, 이러한 점들을 고려하여 S자 성장곡선이 적용 가능한지에 대해 추후 연구를 수행할 것이다.

참 고 문 헌

[1] N. E. Fenton and S. L. Pfleegr, "Software Metrics : A Rigoous and Practical Approach," 2nd Edition, PWS Publishing Company, 1997.

[2] 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.

[3] D. Garmus and D. Herron, "Estimating Software Earlier and More Accurately," Methods & Tools, 1997.

[4] V. B. Mišić, "Software Size and Cost Estimation," De-

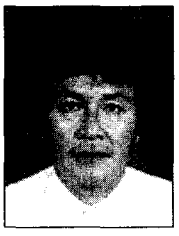
partment of Computer Science, University of Belgrade, 2003.
[5] C. Mcphee, "SENG 621 - Software Process Management," University of Calgary, 1999.
[6] S. Sultanoglu, "Software Measurement," Department of Computer Science & Eng., Hacettepe University, 1998.
[7] B. W. Boehm, 'Software Engineering Economics,' Prentice-Hall, 1981.
[8] M. Bradley, "Function Point Counting Practices Manual, Release 4.1," International Function Point Users Group (IFPUG), 1999.
[9] C. Symons, "COSMIC-FFP Measurement Manual, Version 2.2(The COSMIC Implementation Guide for ISO/IEC 19761 : 2003)," Common Software Measurement International Consortium, 2003.
[10] B. W. Boehm et al, "Software Cost Estimation with COCOMO II," Prantice-Hall, 2000.
[11] K. Ribu, "Estimating Object-oriented Software Projects with Use Cases," University of Oslo Department of Informatics, Master of Science Thesis, 2001.
[12] C. Jones, 'Applied Software Measurement, Assuring Productivity and Quality,' McGraw-Hill, 1997.
[13] S. Pressman, "Software Engineering : A Practitioner's Approach," 5th Edition, Quality, McGraw-Hill, 2001.
[14] R. E. Park, "Software Size Measurement : A Framework for Counting Source Statements," Technical Report CMU/SEI-92-TR-020, 1992.
[15] H. Leung and Z. Fan, "Software Cost Estimation," Department of Computing, Hon Kong Polytechnic University,
[16] K. Johnson, "Software Cost Estimation : Metrics and Models," Department of Computer Science, University of Calgary.
[17] C. Henry, "The Growth Curve," http://www.anzpug.org/jsp/index.jsp, PRIMAVERA Users Groups, Technology andOperations Management, California Polytechnic and State University.
[18] R. Pearl, "The Biology of Population Growth," New York : Knopf, 1978.
[19] B. Gompertz, "On The Nature of The Function Expressive of The Law of Human Mortality, and on a New Mode of Determining the Value of Life Contingencies," Phil. Trans. Roy. Soc. London, Vol.123, pp.513-585, 1832.
[20] M. Wideman, "Appling Resource Loading, Production & Learning Curves to Construction : A Pragmatic Approach," http://www.maxwideman.com/papers/resource/, 2001.
[21] G. Gendall, "The Mysterious S Curve," PROJECT magazine, Vol.4, Iss, 3, http://www.projectmagazine.com/v4i3scuvel.html, May, 2003.
[22] Y. Shigeru, "A Stochastic Software Reliability Growth Model with Gompertz Curve," IPSJ, Vol.33, No.7, Jurna, 2001.
[23] ISBSG, "Worldwide Software Development-The Benchmark Release 6," Victoria, Australia International Software Benchmarking Standards Group, 2000.
[24] A. Abran, C. Symons, and S. Oligny, "An Overview of COSMIC-FFP Field Trial Results," ESCOM 2001, London,England, 2001.
[25] P. Morris, "COSMIC-FFP Field Trials : 2000 Status Report," ACOSM(ASMA) Conference - Sidney Australia, 2000.
[26] S. Nageswaren, "Test Effort Estimation Using Use Case Points," Quality Week 2001, San Francisco, California, USA, 2001.

박 석 규

e-mail : sgpark@gw.ac.kr
1988년~2001년 진주산업대학교 전산실장
1992년 경남대학교 대학원 컴퓨터공학과 석사
2002년~현재 경상대학교 대학원 컴퓨터과학과 박사과정
2001년~현재 도립 강원전문대학 컴퓨터응용과 조교수
관심분야 : 소프트웨어 신뢰성, 시스템 분석 및 설계, 멀티미디어

이 상 운

e-mail : sulee@sky.wonju.ac.kr
1983년~1987년 한국항공대학교 항공전자공학과(학사)
1995년~1997년 경상대학교 컴퓨터과학과 (석사)
1998년~2001년 경상대학교 컴퓨터과학과 (박사)
1992년~2003년 국방품질관리소 항공전자장비 및 소프트웨어 품질보증 담당
2003년 도립 강원전문대학 컴퓨터응용과 전임강사
2004년~현재 국립 원주대학 여성교양과 전임강사
관심분야 : 소프트웨어 공학(소프트웨어 시험 및 품질보증, 소프트웨어 신뢰성), 소프트웨어 프로젝트 관리, 신경망, 뉴로-퍼지, Use-Case, RUP, CBD



박 재 흥

e-mail : pjh@nongae.gsnu.ac.kr
1978년 충북대학교 수학과(학사)
1980년 중앙대학교 대학원 전산학과(석사)
1988년 중앙대학교 대학원 전산학과(박사)
1983년~현재 경상대학교 컴퓨터과학과 교수, 컴퓨터정보통신연구소 연구원
관심분야 : 소프트웨어 신뢰성, 시험도구 자동화, 시스템 분석 및 설계, 신경망 등