

소프트웨어 개발 세부단계 노력 추정 모델 (A Model for Software Effort Estimation in the Development Subcycles)

박 석 규* 박 영 목** 박 재 흥***
(Seok-Kyu Park) (Young-Mok Park) (Jae-Heung Park)

요 약

성공적인 프로젝트 계획은 활용 가능한 일정과 더불어 프로젝트를 완수하는데 요구되는 노력을 얼마나 정확히 추정하느냐에 달려있다. 새로운 또는 보다 나은 모델 개발에 많은 연구가 이루어졌지만 현존하는 소프트웨어 노력 추정 모델들은 개발 전순기에 대해 투입되는 총 개발노력과 단위시간당 소요되는 인력인 노력 함수만을 제공한다. 또한, Putnam은 세부단계별로 일정한 개발노력 투입 비율을 제시하였다. 그러나 소프트웨어의 규모, 복잡도와 운영환경의 영향으로 인해 프로젝트 별로 투입되는 총 개발노력의 규모에 차이가 발생하며, 그 결과, 개발 세부단계별로 투입되는 노력의 규모도 프로젝트마다 차이가 발생한다. 본 논문은 총 개발노력 변동에 따른 소프트웨어의 명세화, 구축과 시험단계에 투입될 개발노력을 추정하는 선형과 다항식 모델을 제시하였다. 이 모델들은 128개의 다른 소프트웨어 프로젝트들로부터 유도되었다. 제안된 모델은 프로젝트의 일정과 노력 할당 관리에 실질적인 지침을 제공할 것이다.

ABSTRACT

Successful project planning relies on a good estimation of the effort required to complete a project, together with the schedule options that may be available. Despite the extensive research done developing new and better models, existing software effort estimation models are present only the total effort and effort (or manpower : people per unit time) function for the software life-cycle. Also, Putnam presents constant effort rate in each subcycles. However, the size of total efforts are variable according to the software projects under the influence of its size, complexity and operational environment. As a result, the allocated effort in subcycle also differ from project to project. This paper suggests the linear and polynomial effort estimation models in specifying, building and 'testing phase followed by the project total effort. These models are derived from 128 different projects. This result can be considered as a practical guideline in management of project schedule and effort allocation.

키워드(Keywords) : 소프트웨어 개발노력 (Software Development Effort), 인력 (Manpower),
개발 단계 (Development Phase), 선형회귀분석 (Linear Regression Analysis),
다항식 회귀분석 (Polynomial Regression Analysis),

* 정회원 : 강원전문대학 교양전산 전임강사
** 정회원 : 경상대학교 컴퓨터과학과 박사과정
*** 정회원 : 경상대학교 컴퓨터과학과 교수

논문접수 : 2001. 6. 15.
심사완료 : 2001. 6. 23.

1. 서론

대형 소프트웨어 프로젝트의 1%만이 계획된 기간과 예상 비용한도 내에서 고객을 만족시키며 완료되었는데 반해, 대부분의 프로젝트들은 1년 이상의 일정이 지연되고 초기 예상 비용의 2배 정도가 초과되었다 [1]. 이와 같은 이유로 인해, 프로젝트 관리 측면에서 소프트웨어 개발 및 유지보수 비용을 줄이고자 체계적인 연구가 수행되고 있으며, 소프트웨어 비용산정 및 프로젝트의 일정관리 모델을 개발하는 계기가 되었다. 계획단계에서 보다 정확한 비용산정 및 일정관리는 프로젝트를 관리할 때 발생하는 다양한 의사결정, 소요 예산 및 개발인원 할당과 계약체결 여부에 신뢰할 만한 정보를 제공한다.

소프트웨어 개발노력과 비용 산정 (Development Effort and Cost Estimation)과 관련한 연구로는 Putnam의 SLIM (Software Lifecycle Management) 모델 [2], LOC (Line Of Code)를 이용한 Boehm의 COCOMO (CONstructive COSt MOdel) 모델 [3, 4]이 있다. COCOMO 모델은 코딩이 종료된 시점에서 정확한 LOC (Line Of Codes)를 기반으로 개발노력과 비용을 추정할 수 있다. 이에 비해, 소프트웨어 개발 초기인 요구 명세화 (Requirement Specification) 단계에서 개발노력과 비용을 추정하기 위한 방법으로 기능점수 (Function Point, FP)를 이용한 기능점수 분석 (Function Point Analysis, FPA) 기법이 있다 [5-7].

이와 같은 연구들은 모두 소프트웨어 프로젝트 전체에 소요되는 개발노력과 비용을 추정하는 것이며, 개발이 진행되면서 단위시간 당 소요되는 노력의 함수를 제공하고 있다. 또한, 추정된 전체 개발노력을 소프트웨어 개발 각 단계별로 어느 정도의 비율로 할당해야 하는지에 대한 연구는 Putnam [2]이 각 개발단계별로 일정한 값을 제시하였다. 그러나 개발되는 프로젝트의 규모, 복잡도와 운영환경의 차이로 인해 총 소요되는 개발노력의 규모뿐만 아니라 각 개발단계별로 투입되는 노력의 규모에도 심한 변동이 발생한다. 따라서, 본 논문은 기존 모델을 이용해 추정된 총 개발노력 규모에 따른 소프트웨어 개발 각 단계별로 최적의 노력을 할당할 수 있는 모델을 제시하고자 한다.

2장에서는 기존의 개발노력과 비용을 추정하는 모델들을 살펴보고, 3장에서는 추정된 총 개발노력 규

모에 따른 소프트웨어 개발 각 단계별로 최적으로 할당하는 모델을 유도해 본다. 4장에서는 개발 각 단계별로 소요되는 노력을 추정하는 모델을 실험을 통해 제시한다.

2. 기존 연구 및 연구배경

COCOMO 모델 [3,4]은 소프트웨어의 소스코드인 LOC를 기반으로 하기 때문에 개발 초기단계에서는 사용이 불가능하고, 코딩단계가 끝난 시점에야 정확한 계산이 될 수 있다. LOC를 이용한 소프트웨어 개발 노력 및 비용 추정 모델은 식 (1)의 형태를 취한다.

$$E = a \cdot KLOC^b \text{ or } E = a + b \cdot KLOC^c \quad (1)$$

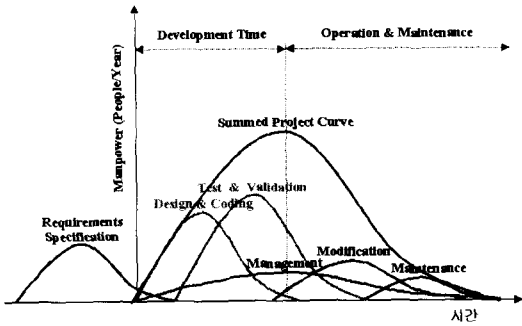
여기서, 개발노력 E 는 예측된 생명주기 노력으로 Man-Months 또는 Man-Years로 측정되고, a, b, c 는 상수이며, KLOC (Thousands of Line Of Code) 최종 코딩된 소프트웨어의 라인 수이다.

기능점수분석 기법은 사용자에게 양도될 시스템의 기능으로 소프트웨어 시스템의 규모와 복잡도를 정량화하는 방법이다 [7, 8]. 이 기법은 소프트웨어 프로젝트를 개발하기 위해 사용되는 언어 또는 도구와 독립적이며, 개발 생명주기의 초기단계인 요구분석 단계에서 측정 가능한 장점을 갖고 있어 LOC를 사용할 때의 주요 문제점을 극복할 수 있는 접근법이다. [5] 기능점수를 이용한 개발노력 추정은 개발노력 E 가 FP에 대해 식 (2)의 형태를 취한다.

$$E = a + b \cdot FP \text{ or } \sqrt{E} = a + b \cdot FP \quad (2)$$

Norden [9]은 IBM에서 개발된 다양한 하드웨어 개발과정에 투입된 인력 (Manpower)의 근사 모델로 Rayleigh 분포를 따른다는 사실을 관찰하였다. 이후 이 관찰 결과가 Putnam [2]에 의해 소프트웨어 프로젝트에 적용되었다. 150개의 소프트웨어 프로젝트가 Norden [10]과 Putnam [2]에 의해 연구되었으며, 프로젝트 단계의 개발에 참여하는 전체 노력 (단위시간 당 소요 인원) 뿐만 아니라 개발 사이클의 각 단계 (즉, 요구명세, 분석, 설계, 코딩 등)에 대한 노력도

[그림 1]의 Rayleigh 함수로 표현됨을 관찰하였다. 임의의 시간 t 까지 프로젝트에 투입된 누적 노력 y 를 식 (3)으로, 임의의 시간 t 에서 프로젝트에 투입되는 노력 \dot{y} 는 식 (3)을 미분한 식 (4)로 표현된다.



[그림 1] 소프트웨어 개발노력 분포

[Fig. 1] The distribution of software development effort

$$y = E(1 - e^{-at^2}) \text{ MY} \quad (3)$$

$$\dot{y} = 2Eate^{-at^2} \text{ MY/YR} \quad (4)$$

여기서 E 는 Rayleigh 곡선 아래에 있는 영역으로 년 인원으로 표시되며 생명주기 전체에 소요되는 노력이다. $a = \frac{1}{2t_d^2}$ 이며, t_d 는 y 이 최대가 되는 시점으로 경험적으로 볼 때, 시스템이 운영되는 시점에 근접한다. 따라서, $t_{y_{\max}} = t_d$ 를 시스템의 개발기간이라 하며, 총 소요되는 개발노력의 약 40%에 도달하는 시점이다. 식 (4)에서 $a = \frac{1}{2t_d^2}$ 를 치환하면, 식 (5)가 된다.

$$\dot{y} = \frac{E}{t_d^3} te^{-\frac{t^2}{2t_d^2}} \quad (5)$$

Boehm [3]은 소프트웨어 개발 초기에 느린 개발 노력 형성 부분과 마지막 단계에서의 점차 길게 감소하는 Rayleigh 분포 특성이 대부분의 소프트웨어 프로젝트의 개발노력 곡선에 일치하지 않음을 지적하였다. 일반적으로 소프트웨어 프로젝트는 개발 초

기에 하드웨어 프로젝트보다 빠른 형성 비율 (Build-up Rate)을 가지며, 이는 Rayleigh 분포와 편향된 결과를 나타낸다. 이 연구 결과를 토대로, Pillai et al. [11]은 Putnam의 개발노력 분포가 Rayleigh 분포를 따르는데 따른 문제점을 해결하기 위해 식 (6)의 Gamma 모델을 제시하였다.

$$\dot{y} = \frac{8K}{I(3)t_d^3} (t^2 e^{-\frac{2}{t_d^3}t}) = \frac{4K}{t_d^3} t^2 e^{-\frac{2}{t_d^3}t} \quad (6)$$

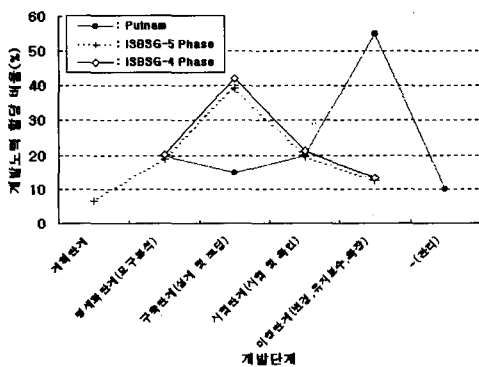
이들 제안된 모델들을 이용하면 소프트웨어 프로젝트 전체에 대한 개발노력과 비용을 산정 할 수 있어 사업초기에 의사결정 자료로 활용이 가능하다. 그러나, 소프트웨어를 개발하기 시작할 때, 총 개발노력을 개발의 각 단계별로 얼마의 비율로 할당해야 하는가의 문제를 해결해야만 한다. [그림 1]에서 보는 바와 같이 개발 시간 단위당 투입되는 노력 분포 곡선 데이터만 가진 경우, 중첩되어 각 개발단계가 수행되므로 인해 각 단계별로 투입되는 노력의 규모를 구할 수 없다. Putnam [2]은 60-70년대에 개발된 150개 프로젝트에 대한 경험치를 적용하여 요구명세 단계에는 약 20% 까지 소요되며, 개발이 시작된 이후 프로젝트 개발 및 유지보수단계에서 소요된 총 개발노력을 100으로 보았을 때, 설계 및 코딩단계는 15%, 시험 및 검증단계는 20%, 변경, 유지보수와 확장 단계는 55%와 관리부분에 10%가 소요됨을 밝혔다. 이를 적용하는데는 다음과 같은 문제가 발생한다.

- (1) 이들 프로젝트들은 70년대까지 개발된 프로젝트의 사례들이며, 3세대와 4세대 언어를 사용하고, 보다 복잡하고 고 신뢰성을 요구하고 있는 90년대 이후 소프트웨어 프로젝트에 적용하는데는 오차가 발생한다.
- (2) 소프트웨어의 프로젝트 규모, 복잡도, 운영환경 등의 차이로 인해 프로젝트에 소요되는 총 개발노력에 차이가 발생하며, 개발 각 단계별 투입되는 개발노력의 비율에도 큰 편차가 발생한다. 따라서, 프로젝트의 특성에 관계없이 일정하게 결정된 값을 적용할 수 없다.

위와 같은 문제점은 ISBSG Benchmark Release 6 [12]의 데이터를 분석하여 보면 분명하게 밝혀진다.

ISBSG Benchmark Release 6 [12]는 '90년대에 개발된 789종의 소프트웨어 프로젝트에 대한 기능점수와 개발 노력, 이들에 영향을 미치는 속성들에 대한 정보를 갖고 있다. 이들 프로젝트들은 계획단계 (Planning Phase), 명세화 단계 (Specifying Phase), 구축단계 (Building Phase), 시험단계 (Testing Phase)와 이행단계 (Implementing Phase)로 구분하여 각 단계별로 소요된 개발노력 정보를 갖고 있다. 계획단계는 실행가능성 고찰, 비용 효과 고찰과 사업 초기 책임 등을 고려하는 단계이며, 명세화 단계에서는 시스템 분석, 기능적 설계, 외부 설계, 요구사항 정의와 패키지 선택 등의 업무를 수행하며, 요구정의 단계에 해당된다. 구축단계는 물리적 설계, 내부 설계, 코딩, 프로그래밍, 인터페이스 등의 업무를 수행하는 단계로 설계 및 코딩단계와 관련된다. 시험단계는 통합시험과 사용자 수락시험을 수행하는 단계로 시험 및 확인단계와 관련되며, 이행단계는 이행, 설치와 사용자 훈련을 수행하는 것으로 변경, 유지보수 및 확장 단계와 관련시킬 수 있다.

Putnam [2]이 제시한 개발 각 단계별 개발노력 투입 비율과 ISBSG Benchmark Release 6 [12]에서 단계별 투입된 개발노력 데이터를 모두 갖고 있는 76개 프로젝트에 대해 단계별로 투입된 개발노력 비율을 평균한 결과는 [그림 2]와 같다.



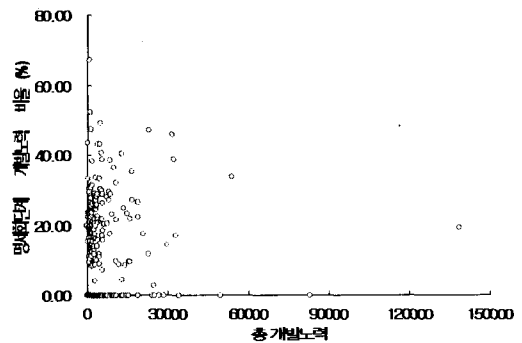
[그림 2] 단계별 개발노력 할당 비율

[Fig. 2] The partition ratio of development effort for lifecycle subphase

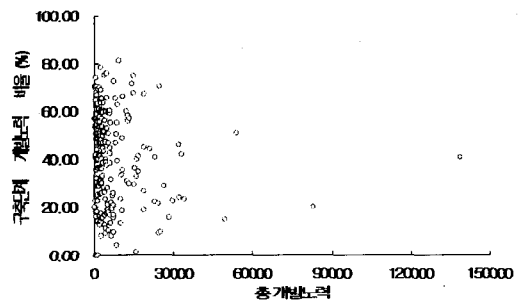
[그림 2]에서 Putman [2]의 결과와 ISBSG의 5단계 (ISBSG-5 Phase) 또는 계획단계를 제외한 4단계

(ISBSG-4 Phase)의 경우 명세화 단계와 시험단계는 유사한 개발노력이 투입되나 구축단계 (설계와 코딩)와 이행단계 (변경, 유지보수 및 확장)에서는 커다란 차이가 발생한다.

또한, 단계별 투입된 개발노력 자료를 보유한 210종의 프로젝트에 대해 총 개발노력에 따른 단계별 개발노력의 비율을 그래프로 나타낸 결과는 [그림 3]과 같다. 그림에서 0%를 나타내는 것은 해당 프로젝트의 개발 형태 (재개발 또는 개정)에 따라 해당 단계를 수행하지 않아 발생된 것이다.



(a) 총 개발노력에 따른 명세화 단계 개발노력 비율



(b) 총 개발노력에 따른 구축단계 개발노력 비율

[그림 3] 단계별 개발노력 비율

[Fig. 3] The ratio of development effort in lifecycle subphase

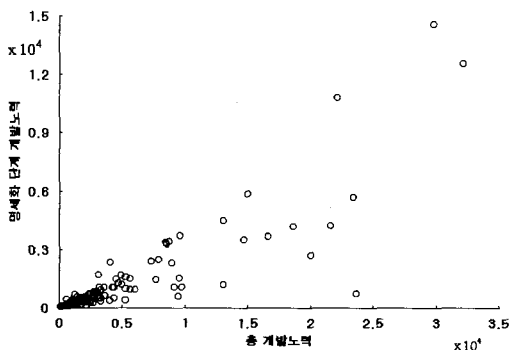
[그림 3]에서 프로젝트에 소요된 총 개발노력의 양에 따른 해당 개발단계에 투입되는 개발노력의 비율이 명세화 단계에서는 10 ~ 40%, 구축단계에서는 10 ~ 80%로 심한 변동을 가짐을 알 수 있다. 시험단계와 운영단계도 유사한 경향을 보였다. 그러나

Putnam [2]을 비롯한 이전 연구에서는 프로젝트 개발시 소요되는 총 개발노력의 규모에 따라 단계별 소요되는 개발노력의 규모를 추정하는 모델을 제시하지 못하였고, 프로젝트의 총 개발노력 규모에 상관없이 단지 일정한 비율을 할당하는 정보만을 제시하고 있다.

실제로 소프트웨어를 개발하기 전에 산정된 총 개발노력에 대해 개발의 각 단계별로 개발노력을 적절히 할당할 수 있는 모델을 사용한다면, 보다 효율적인 개발과 요구되는 제품의 품질수준 달성이 가능하다. 따라서, 3장에서는 이 부분에 대한 연구를 수행하고자 한다.

3. 개발 단계별 소요 노력 추정 모델

[그림 3]의 관계를 이용할 경우 프로젝트에 투입되는 총 개발노력의 양에 따른 단계별 투입 개발노력 비율의 변동이 심하고 특별한 관계를 도출할 수 없으므로 각 단계별로 투입되는 개발노력을 추정하는 모델 제시가 불가능하다. 따라서, 총 개발노력의 양에 따른 어떤 변수의 값이 특정 관계를 가지는 것을 찾아 이에 대한 식을 유도함으로써, 모델을 제시할 수 있다. ISBSG Benchmark Release 6 [12]에서 210종의 프로젝트는 신규 개발, 재개발 또는 개정되는 관계로 프로젝트 별로 수행된 단계가 모두 다르게 나타난다.



[그림 4] 총 개발노력과 명세화 단계 개발노력 관계
[Fig. 4] The relation between total development effort and specification phase

이들 프로젝트들 중에 128종의 프로젝트가 명세화, 구축과 시험 단계에 대한 개발노력 정보를 갖고 있어 이들 단계를 대상으로 모델을 제시한다. 128종의 프로젝트에 대해 프로젝트에 투입된 총 개발노력의 규모와 명세단계에 투입된 개발노력과의 관계를 그래프로 나타내면 [그림 4]와 같다.

구축과 시험단계도 [그림 4]와 유사한 경향을 나타내었다. 그림에서 명세단계에 투입되는 개발노력은 총 개발노력이 증가함에 따라 선형이나 비선형 형태로 증가하는 관계를 가짐을 알 수 있다.

따라서, 소프트웨어 개발 각 단계별로 투입되는 개발노력을 추정하기 위해서 식 (7)의 1차선형 모델 또는 식 (8)의 2차 다항식 모델을 적용해 적합 (Curve Fitting)시킬 수 있다.

$$E_s = a + bE_{total} \tag{7}$$

$$E_s = a + bE_{total} + cE_{total}^2 \tag{8}$$

여기서 E_s 는 명세화 단계에 투입된 개발노력이며, E_{total} 은 프로젝트의 3개 단계 (명세화, 구축과 시험단계) 모두에 투입된 총 개발노력이다. 이후부터 구축단계와 시험단계에 투입되는 개발노력을 각각 E_b 와 E_t 로 기술한다.

4. 실험 및 결과 분석

제안된 식 (7)과 식 (8)의 모델을 이용해 모수를 추정하면 총 개발노력의 양에 따라 각 단계별로 투입될 개발노력을 산정할 수 있다. ISBSG Release 6 [13]의 789개 프로젝트 중에 명세, 구축과 시험단계에 대한 개발노력 정보를 보유한 128종의 프로젝트에 대해 이들 모수를 추정하기 위해, 회귀분석을 수행하였다. 식 (7)에 대해서는 선형회귀분석 (Linear Regression Analysis)을, 식 (8)에 대해서는 다항식 회귀분석 (Polynomial Regression Analysis)을 수행하였다. 명세화 단계에 투입될 개발노력 추정 모델은 선형과 다항식 모델로 각각 식 (9)와 식 (10)으로 제시되어 있으며, 개발노력 추정 결과는 [그림 5]에 나타내었다.

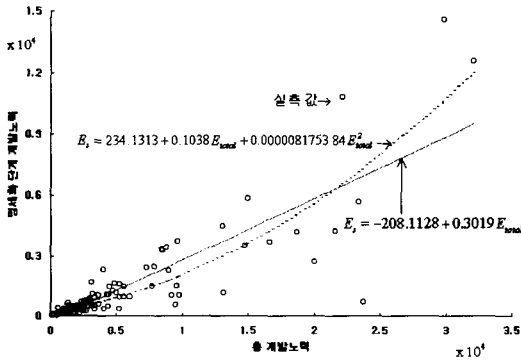
$$E_s = -208.1128 + 0.3019E_{total} \quad (9)$$

$$E_s = 234.1313 + 0.1038E_{total} + 0.000008175384E_{total}^2 \quad (10)$$

구축단계에 투입될 개발노력 추정 모델은 선형과 다항식 모델로 각각 식 (11)과 식 (12)로 제시되어 있으며, 개발노력 추정 결과는 [그림 6]에 나타내었다.

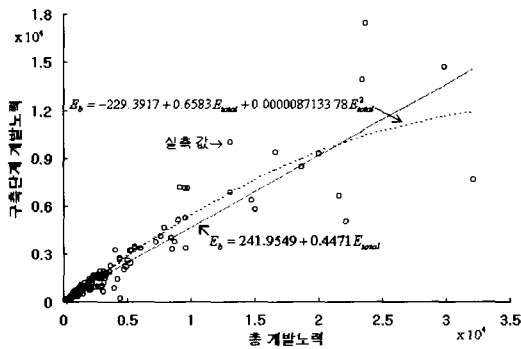
$$E_b = 241.9549 + 0.4471E_{total} \quad (11)$$

$$E_b = -229.3917 + 0.6583E_{total} - 0.000008713378E_{total}^2 \quad (12)$$



[그림 5] 명세화 단계 개발노력 추정

[Fig. 5] Estimate of development effort in specification phase



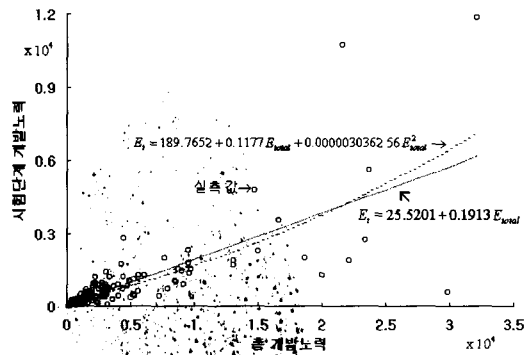
[그림 6] 구축단계 개발노력 추정

[Fig. 6] Estimate of development effort in build phase

또한, 시험단계에 투입될 개발노력 추정 모델은 선형과 다항식 모델로 각각 식 (13)과 식 (14)로 제시되어 있으며, 개발노력 추정 결과는 [그림 7]에 나타내었다.

$$E_t = 25.5201 + 0.1913E_{total} \quad (13)$$

$$E_t = 189.7652 + 0.1177E_{total} + 0.000003036256E_{total}^2 \quad (14)$$



[그림 7] 시험단계 개발노력 추정

[Fig. 7] Estimate of development effort in testing phase

식 (10), (12)와 (14)에서 2차항은 무시했을 경우도 거의 유사한 결과를 나타낸다. 따라서, 이들 식은 각각 식 (9), (11)과 (13)과 같이 1차항 형태를 취해 보다 단순한 모델이 될 수 있다.

각 단계별 제안된 모델로 추정된 결과 모델의 평균 제곱오차 (MSE, Mean Squared Error)는 <표 1>과 같다. 표에서 보는바와 같이 3개 단계 모두 선형 모델 보다는 다항식 모델의 오차가 작음을 알 수 있다. 따라서, 본 논문은 선형모델 보다는 다항식 모델을 사용하는 것을 권장한다.

<표 1> 모델의 추정오차

<Table 1> Estimation error of model

단계	MSE	
	선형 모델	다항식 모델
명세화 단계	1,189,691.75	1,008,050.97
구축 단계	1,581,002.84	1,374,789.13
시험 단계	1,109,103.71	1,084,068.51

5. 결론 및 향후 연구과제

기존 소프트웨어 개발노력과 비용 추정 모델들은 소프트웨어의 코딩 단계 또는 개발 초기에 소프트웨어 개발에 투입되는 총 개발노력과 비용, 시간당 투입되는 노력의 함수만을 추정하였으며, 개발 각 단계별로 투입되는 개발노력의 비율을 개발노력의 양에 상관없이 일정한 값으로 제시하였다. 그러나, 프로젝트의 특징, 규모, 복잡도와 운영환경 등의 요인에 의해 프로젝트별로 다르게 투입되는 총 개발노력의 규모에 따라 단계별 투입될 개발노력의 비율이 달라진다. 또한 개발 각 단계는 분리되어 순차적으로 진행되지 않고, 중첩되어 진행되므로, 단위시간당 투입되는 노력 데이터만을 갖고 있는 경우, 각 단계별로 노력을 적절히 할당할 수 없다. 따라서, 기존 제안된 모델을 이용해 프로젝트 개발에 소요되는 총 개발노력을 추정하더라도 실제로 소프트웨어를 개발하는 과정에서 각 단계별로 노력을 얼마로 할당하느냐에 따라 개발의 성공 여부가 결정될 수 있는 문제점이 있었다.

본 논문은 소프트웨어 개발과정에서 추정된 총 개발노력의 규모에 따라 개발 각 단계별로 어느 정도의 비율로 할당하는 것이 최적인가를 추정할 수 있는 모델을 제시하였다. 128종의 프로젝트를 이용하여 명세화, 구축과 시험 단계별로 소요되는 개발노력을 추정할 수 있는 선형 모델과 다항식 모델들을 제안하였다. 본 제안된 모델을 이용하여 소프트웨어 개발 각 단계별 투입되는 개발노력을 결정할 경우, 보다 향상된 관리능력 및 인력 할당 효과를 얻을 것이다.

본 논문에서는 프로젝트의 총 개발노력의 변동에 따른 개발단계별로 투입되는 개발노력을 추정하는 모델을 제시하였다. 그러나, 프로젝트의 적용분야, 개발 형태, 사용 언어와 운영 형태 등에 따라라도 단계별로 투입되는 개발노력의 비율이 영향을 받을 수 있다. 따라서, 추후 이 분야에 대한 연구를 수행할 것이다.

※ 참고문헌

- [1] K. H. Möller and D. J. Paulish, "Software Metrics-A Practitioner's Guide to Improved Product Development", Chapman & Hall Co., New York, 1993.
- [2] L. H. Putnam, "A General Empirical Solution to the Macro Software Sizing and Estimating Problem," IEEE Trans. on Software Eng., Vol. SE-4, No. 4, 1978.
- [3] B. W. Boehm, "Software Engineering Economics," Prentice Hall, 1981.
- [4] B. W. Boehm, "Software Engineering Economics," IEEE Trans. on Software Eng., Vol. 10, No. 1, pp. 7-19, 1984.
- [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.
- [6] A. J. Albrecht and J. E. Gaffney, "Software Function, Source Line of Code and Development Effort Prediction : A Software Science Validation," IEEE Trans. on Software Eng., Vol. SE-9, No. 6, pp. 639-648, 1983.
- [7] A. J. Albrecht, "Measuring Application Development Productivity," in Programming Productivity : Issues for the Eighties, C. Jones, ed. Washington, DC : IEEE Computer Society Press, 1981.
- [8] C. F. Kemerer, "An Empirical Validation of Software Cost Estimation Models," Communication ACM, Vol. 30, No. 5, pp. 416-429, 1987.
- [9] P. V. Norden, "Curve Fitting for a Model of Applied Research and Development Scheduling," IBM J. Research and Development, Vol. 3, No. 2, pp. 232-248, 1958.

- [10] P. V. Norden, "Project Life Cycle Modeling : Background and Application of the Life Cycle Curves," U. S. Army Computer System Command, 1977.
- [11] K. Pillai and V. S. Sukumaran Nair, "A Model for Software Development Effort and Cost Estimation," IEEE Trans. on Software Eng., Vol. 23, No. 8, pp. 485-497, 1997.
- [12] ISBSG, "Worldwide Software Development - The Benchmark Release 6," Victoria, Australia International Software Benchmarking Standards Group, 2000. Can be ordered at <http://www.isbsg.org.au>

박 석 규



1988. 4. ~ 2001. 2.
 진주산업대학교 전산실장
 1990. 3 ~ 1992. 8.
 경남대학교 컴퓨터공학(석사)
 2001. 3. ~ 현재
 강원전문대학 교양전산
 전임강사

박 영 목



1983. ~1989. 경상대학교
 전산통계학과 학사
 1997. ~ 1999. 8. 경상대학교
 산업정보공학과 석사
 2000. ~ 현재 경상대학교
 컴퓨터과학과 박사과정
 관심분야 : 소프트웨어 공학
 (소프트웨어 시험 및
 소프트웨어 신뢰성), 신경망

박 재 흥



1973. ~ 1978. 충북대학교
 수학과 학사
 1978. ~ 1980. 중앙대학교
 전산학과 석사
 1985년 ~ 1988년 중앙대학교
 전산학과 박사
 1983년 ~ 현재 경상대학교
 컴퓨터과학과 교수
 관심분야 : 소프트웨어 신뢰성,
 시험도구 자동화,
 시스템 분석 및 설계, 신경망