

<http://dx.doi.org/10.7236/JIIBC.2013.13.4.99>

JIIBC 2013-4-14

## 소프트웨어 프로젝트의 비용-일정 타협

### Cost-Schedule Tradeoff in Software Project

이상운\*, 최명복\*\*

Sang-Un Lee, Myeong-Bok Choi

**요약** 일반적으로 소프트웨어를 개발하는데 소요되는 일정 단축은 추정된 명목상 일정을 기준으로 75% 범위 내에서 결정된다. 그러나 지금까지 명목상 개발 일정이 명확히 제시되지 않아 개발일정 단축 가능성을 판단하는데 어려움이 있다. 따라서 본 논문은 다양하게 제안된 명목상 개발일정 기준을 고찰하여 최적의 범위를 결정하였다. ISBSG Release 8의 기능점수 기반 실제 수행된 1,595개 프로젝트를 대상으로 이 기준 범위에 속하는 614개의 프로젝트를 추출하였다. 이 데이터들을 대상으로 개발노력 추정 모델을 유도하였다. 또한, 추정된 개발노력 모델에 기반하여 개발기간을 추정하는 모델을 유도하였다. 제안된 모델을 적용시 실제 프로젝트를 개발시 요구되는 개발노력과 개발기간을 보다 현실적으로 추정할 수 있을 것이다.

**Abstract** Generally, software development schedule compression is decided within 75% of nominal schedule. However, there are some difficulties when we judge the possibility of development schedule compression because a nominal schedule has not been definitely. Therefore, this paper investigates various nominal schedule estimation models and decides the optimum range. Basing on the performance of ISBSG Release 8, 614 projects are extracted from the actual 1,595 projects. We presented development effort estimation model from those data. Also, we derived the development schedule model from the development effort. When you apply the proposed model, you will be able to estimate development effort and schedule required for the development more actually.

**Key Words** : Impossible Region, Practical Region, Optimal Region, Impractical Region, Cost-Schedule Trade-off

## 1. 서론

소프트웨어 개발의 성공 여부는 고객이 요구하는 품질을 만족하는 제품을 주어진 일정과 예산범위 내에서 납품할 수 있는지로 결정된다<sup>[1]</sup>. 대부분의 개발조직들은 계약된 일정에 맞추어 프로젝트를 성공적으로 종료시킬 수 있는지, 계약된 일정 자체가 타당하지 조차도 알지 못한 상태에서 프로젝트를 진행한다<sup>[2]</sup>. 이러한 결과로 인해,

2000년 기준으로 28%의 프로젝트만이 주어진 일정과 비용범위 내에서 개발이 완료되었으며, 23%가 개발이 취소되었고, 49%가 개발비용 초과(계획의 45% 초과)와 개발기간 지연(계획의 63% 지연)이 발생하였다<sup>[3]</sup>. 따라서 개발 성공률을 높이기 위해서는 적절한 일정 수립과 노력 투입이 요구되며 이에 필요한 비용(노력)-일정 상관관계를 표현할 수 있는 모델이 필요하다.

\*정회원, 강릉원주대학교 멀티미디어공학과

\*\*중신회원, 강릉원주대학교, 멀티미디어공학과

접수일자 : 2013년 1월 5일, 수정완료 : 2013년 6월 25일

게재확정일자 : 2013년 8월 16일

Received: 5 January, 2013 / Revised: 25 June, 2013 /

Accepted: 16 August, 2013

\*\*Corresponding Author: cmb5859@gmail.com

Dept. of Multimedia Engineering, Gangnung-Wonju National University, Korea

일반적으로, 개발될 소프트웨어의 규모(Size,  $S$ )에 기반하여 개발에 소요되는 노력(Effort,  $E$ )은  $E = aS^b$ 으로 추정되며, 개발 노력 (또는 비용)에 근거하여 개발 기간 (또는 일정) (Duration or Schedule,  $D$ )은  $D = aE^b$  식으로 추정된다<sup>[4]</sup>. 여기서  $a$ 와  $b$ 는 상수이다. 이와 같이 추정된 개발기간  $D$ 를 명목상 일정(Nominal Schedule,  $T_n$ )이라 한다. 그러나 소프트웨어 고객은 보다 빠른 기간(Faster)에, 보다 저렴한 비용(Cheaper)으로 보다 좋은 품질수준(Better Quality)의 산출물을 요구한다. 이러한 고객의 요구를 수용하여 명목상 개발기간 데이터에 근거하여 비용과 일정 간에 타협(Trade-off)이 이루어진다.

일정을 단축시키면 비용이 증가되며, 반면에 일정을 연장시키면 비용은 어느 수준까지는 감소되다가 증가하는 경향을 나타내고 있다. 여기서 한 가지 유의할 점은 개발 일정을 어느 한계 이하로는 단축시킬 수 없으며, 어느 한계 이상으로는 연장시킬 수 없다<sup>[4-8]</sup>. 일반적으로 일정단축 한계(Limit of Schedule Compression,  $T_c$ )는 75%  $T_n$ 으로, 일정 연장 한계(Limit of Schedule Stretch,  $T_s$ )는 130%  $T_n$ 으로 제시되어 있다<sup>[9]</sup>.

$T_c$ 에 관한 연구로는 Carbo<sup>[5]</sup>, Putnam과 Myers<sup>[7]</sup>, Fakharzadeh<sup>[10]</sup>, Boehm<sup>[11]</sup>과 Ward<sup>[12]</sup>가 있다. 또한  $T_n$  관련 연구로는 COCOMO<sup>[6]</sup>, Fakharzadeh<sup>[10]</sup>, Pressman, Jones, Basilli와 Putnam<sup>[13]</sup>, Richard<sup>[14]</sup>, Holtsinger<sup>[15]</sup>, COCOMO II<sup>[16]</sup>, Jones<sup>[17]</sup>, Marin<sup>[18]</sup>, McConnell<sup>[19]</sup>, Walston-Felix<sup>[20]</sup>과 Booker<sup>[21]</sup>이 있다. 실제로 이들 제안된 모델들을 적용하기 위해서는 다음과 같은 문제점이 제기된다.

- (1) 제안된 모델들은 대부분이 소량의 데이터들로부터 유도되어 다양한 개발환경과 조건에 적합한 일반화된 하나의 모델이 없는 실정이다.
- (2) 또한, 대부분이 과거의 SLOC (Source Line of Codes), 폭포수 프로세스로 개발된 프로젝트에 기반하여 모델들이 유도되었다. 그 동안 순차적 개발의 폭포수 프로세스에서 반복적이고 점진적 개발 형태인 UP (Unified Process.), Agile 등 개발 프로세스의 변화, 개발 조직의 생산성 향성에도 불구하고 이들 인자를 적절히 반영하지 못하고 있다.
- (3) 다양한 명목상 개발기간 추정 모델이 제시되어 있음에도 불구하고, 개발될 소프트웨어에 적합한 어떤 하나의 모델을 선택할 수 있는 기준이 제시되어 있지 않아 특정 모델을 채택하더라도 타당성 있게 채택하였는지를 판단할 수 없다.

이러한 문제점을 해결하기 위해 본 논문은 최근에 개발된 기능점수 기반의 대용량 프로젝트를 대상으로 개발 노력으로부터 명목상 개발기간을 추정하는 모델을 제시하고, 비용-일정 타협 모델을 제시한다. 모델을 제시하기 위해 기능점수에 기반하여 개발된 ISBSG (International Software Benchmarking Standards Group) Benchmark Release 8<sup>[22]</sup>을 이력 데이터로 선정하였다. 이 이력 데이터는 20개국에서 1989년부터 2002년까지 개발된 2,027개의 대용량 소프트웨어 프로젝트들을 갖고 있으며, 75%가 최근 8년 이내에 개발된 것으로 개발조직, 사업 분야, 개발형태, 응용 형태, 언어, 개발방법론, 개발기법 등에서 매우 다양성을 갖고 있어 데이터 분석에 적용하기에는 매우 좋은 표본이라 할 수 있다. 그러나 한 가지 문제점은 “모든 소프트웨어 프로젝트들의 실제 개발기간이 최적으로 설정된 명목상 일정인지가 불분명하며, 개발노력과 개발기간 사이에 일정한 관계를 유도하기가 불가능하다. 따라서 본 논문은 실험 데이터의 표본이 가능한 동일한 부류가 될 수 있도록 층화 샘플링 방법을 적용하여 다음과 같이 새로운 명목상 개발일정 모델을 제시한다.

- (1) 제안된 개발일정 추정 모델들이 개발 노력에 따라 어떤 분포를 갖고 있는지 분석하여 최대 값과 최소 값을 가지는 모델을 선정한다.
- (2) 이력 데이터들 중에서 최대 값과 최소 값을 가지는 개발일정 추정 모델 범위 내에 속한 데이터들만 분류한다.
- (3) 추출된 표본 데이터들에 기반하여 개발노력과 개발기간 관계로부터 명목상 개발일정 모델을 제시한다.
- (3) 제안된 모델에  $T_c$  (75%  $T_n$ )와  $T_s$  (130%  $T_n$ )를 적용하여 비용-일정을 타협하는 모델을 제시한다.

2장에서는 개발기간 추정과 관련된 연구와 문제점을, 3장에서는 기능점수에 기반한 명목상 개발일정 추정 모델과 비용-일정 타협 모델을 제시한다.

## II. 개발기간 추정 관련 연구와 문제점

### 1. 기능점수 기반 개발기간 모델 유도 문제점

ISBSG Benchmark Release 8<sup>[22]</sup>의 2,027개 소프트웨어 프로젝트들 중에서 개발노력과 개발기간이 기술된 데이터들은 1,595건으로 성능개량(Enhancement, EH) 875건과 신규개발(New Development, ND) 720건이다. 1,595건에 대한 개발노력( $E$ , Person-Months)과 개발

일정 ( $D$ , Months) 관계는 그림 1과 같다.

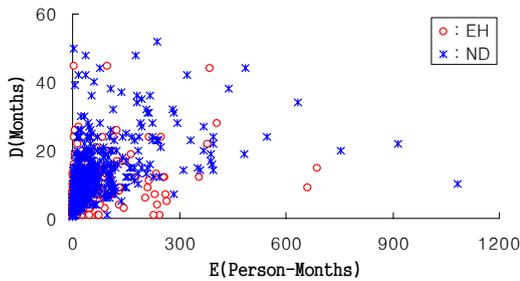


그림 1. 개발노력과 개발기간의 관계  
Fig. 1. Relation of Development Effort and Period

개발노력은 시간 단위 (Person-Hours, PH)로 기술되어 있는 자료를 기존의 개발일정 추정 모델들이 적용하고 있는 월 단위 (Person-Months, PM)으로 변환시키기 위해 COCOMO II<sup>[16]</sup>의 1 PM = 152 PH 기준을 적용하였다. 그림에서 보는 바와 같이, 개발노력과 개발기간 간에는 상관관계가 거의 없어 관계식을 유도하여도 의미가 없는 문제점을 갖고 있어 새로운 방법을 연구할 필요가 있다.

## 2. 비용-일정 타협 관련 연구 및 문제점

소프트웨어 개발 분야에서는 개발인력을 무한히 추가하더라도 어느 시점 이하에서는 시스템을 성공적으로 완료할 수 없는 최소한의 개발기간이 존재하며, 이를 불가능 영역(Impossible Region)이라 부른다<sup>[4-7,23]</sup>.

프로젝트를 성공적으로 완료하기 위해서는 불가능영역을 회피할 수 있어야만 한다. 그러나 고객은 개발 불가능 일정에 상관없이 보다 빠르게 제품을 획득하려는 경향이 있다. 반면에 개발자는 단기간에 개발하는데 동의하지 않을 것이다. 왜냐하면 단기간에 개발할 경우 소요되는 노력이 기하급수적으로 증가하며, 이로 인해 간접비용 증가와 더불어 결함도 추가로 증가하기 때문이다. 따라서 개발노력과 개발일정 모두를 고려하여 적절한(최적의) 값을 설정하는 것이 중요하며 이를 비용-일정 타협 (Cost-Schedule Tradeoff)이라 한다. 따라서 하나의 소프트웨어를 개발하기 위한 일정과 노력 (비용) 간에는 그림 2의 관계를 갖고 있으며, 일정 시점은 다음과 같이 정의된다<sup>[19,24]</sup>.

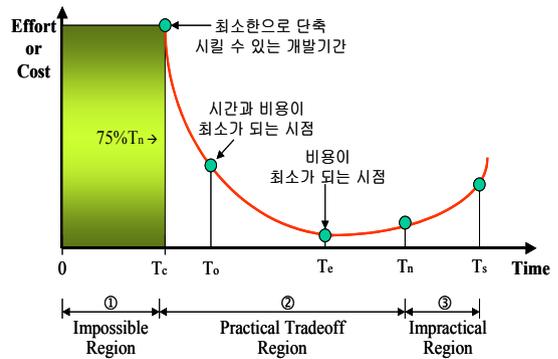


그림 2. 전형적인 프로젝트의 노력과 개발기간 관계  
Fig. 2. Relation of Effort and Period of General Projects

- $T_c$  : 최소 개발완료 시점(더 이상 단축할 수 없는 개발기간)
- $T_o$  : 최적의 개발완료 시점(개발노력과 개발기간 고려)
- $T_e$  : 훌륭한 수준의 개발자, 방법론, 개발도구, 위험관리 등을 갖춘 조직인 경우 개발 완료 시점
- $T_n$  : 평균적인 수준의 개발조직으로 대부분의 조직이 달성할 수 있는 개발완료 시점
- $T_s$  : 최대 개발완료시점(더 이상 연장시킬 수 없는 개발기간)

명목상 개발 일정 ( $T_n$ )을 최대로 단축시킬 수 있는 시점 ( $T_c$ )에 대해 대부분의 연구자들은 약 75%에 한계가 있다고 동의하고 있다( $0.75\%T_n \leq T_c \leq 0.80\%T_n$ )<sup>[6,19,25,27]</sup>.  $T_c$  시점에서는  $T_n$  시점에서 소요되는 노력의 43%가 추가로 소요된다<sup>[25]</sup>. Putnam과 Myers<sup>[7]</sup>는 일반적으로  $T_n$ 은  $T_c$ 의 약 130% 근방에 존재한다고 제시하였다. 이 결과는  $T_n$ 을 1로 가정하면  $T_c = 77\% T_n$ 이 된다. 또한,  $T_s$ 는  $130 T_n$ 으로 일반적으로 제시되어 있다. 실제 프로젝트들을 대상으로 한 경험적 결과로 얻은 각 시점 ( $T_c, T_s$ ) 연구 결과는 다양하게 제시되어 있으며, 표 1에 나타내었다.

표 1.  $T_c$ 와  $T_s$ 의 정의

Table 1. Definition of  $T_c$  and  $T_s$

참고문헌	$T_c$	$T_s$	분석 대상
Sifri <sup>[22]</sup>	$75\% T_n$	$2 T_n$	750개 프로젝트
Carbno <sup>[5]</sup> , Ward <sup>[12]</sup>	-	1명 개발기간	Rules of Thumb
COCOMO II 2000 <sup>[16]</sup>	$75\% T_n$	$130\% T_n$	161개 프로젝트
Putnam과 Myers <sup>[28,29]</sup>	$77\% T_n$	-	4000개 프로젝트
Construx <sup>[9]</sup>	$75\% T_n$	-	Rules of Thumb
Simons <sup>[30]</sup>	PRICE	$70\% T_n$	$130\% T_n$
	DSN	$70\% T_n$	$130\% T_n$
	COCOMO	$75\% T_n$	$130\% T_n$
	Jenson	$85\% T_n$	$120\% T_n$

이와 같이 다양하게 정의된  $T_c$ 와  $T_s$  값들로부터 실제로 어느 기준을 적용할 것인가에 혼란을 초래하고 있으나 일반적 결과인  $T_c = 75\% T_n$ ,  $T_s = 130\% T_n$ 을 적용하는 것이 타당해 보인다.

### 3. 명목상 개발일정 관련 연구 및 문제점

일반적으로 개발일정 단축과 연장 기준을 적용하기 위해서 우리는 먼저 개발될 소프트웨어의 정확한 명목상 개발일정 ( $T_n$ )이 얼마인지를 알아야만 한다. 기존에 제시된  $T_c$ 와  $T_n$  관련 모델들을 요약하여 그림 3에 제시하였다.

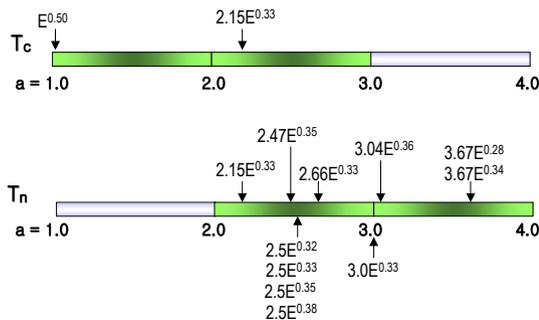


그림 3.  $a$  값 기준 개발일정 추정 모델 분포  
Fig. 3. Development Schedule Estimation Model Distribution with value  $a$

$T_c$ 에 대한 세부 연구결과를 살펴보면, Putnam과 Myers<sup>[7]</sup>와 Boehm<sup>[11]</sup>는  $2.15E^{0.33}$ 을, Carbno<sup>[5]</sup>와

Ward<sup>[12]</sup>는  $\sqrt{E} (= E^{0.50})$ 을 제시하였다. 경험에 의하면 일반적으로  $T_c = \sqrt{E}$  (Square Root Law)을 적용한다<sup>[10]</sup>.  $T_n$ 에 대한 연구결과로 COCOMO<sup>[6]</sup>는 Organic (Basic) Mode는  $2.5E^{0.38}$ , Semi-detached Mode는  $2.5E^{0.35}$ , Embedded (System, Utility) Mode는  $2.5E^{0.32}$ 를 제시하고 있다. Pressman, Jones, Basilli와 Putnam<sup>[13]</sup>은  $2.66E^{0.33}$ , Richard<sup>[14]</sup>는  $3.0E^{0.33}$  ( $2.0 \leq a \leq 4.0$ )을, Holsinger<sup>[15]</sup>와 Booker<sup>[21]</sup>는  $3.0E^{0.33}$  ( $2.5 \leq a \leq 4.0$ )을, COCOMO II<sup>[16]</sup>는  $3.67E^f$  ( $0.28 \leq f \leq 0.34$ )를, Jones<sup>[17]</sup>는  $2.5E^{0.33}$ 을, Marin<sup>[18]</sup>, McConnell<sup>[19]</sup>과 Ardis<sup>[31]</sup>는  $3.0E^{0.33}$ 을 제시하고 있다. 또한, Walston-Felix<sup>[20]</sup>는  $2.47E^{0.35}$ 를 제시하였다. 경험에 의하면  $T_n = a \cdot \sqrt[3]{E} (= aE^{0.33})$  (Cube Root Law)를 적용하고 있다<sup>[10]</sup>.

이와 같이  $T_c$ 와  $T_n$ 에 대해 다양한 모델이 제시됨에도 불구하고 개발될 소프트웨어에 대해 명확히 특정한 하나의 모델을 적용할 수 있는 기준이 설정되어 있지 않다.

## III. 기능점수 기반 비용-일정 타협

### 1. 데이터 표본 추출

그림 1과 같이 다양한 환경과 조건에서 개발된 다량의 이질적인 데이터들로부터 유도할 수 없는 개발기간 추정 모델의 문제점을 해결하기 위해, 본 논문에서는 층화 샘플링 (Stratified Sampling) 방법을 적용한다. 층화 샘플링이란 모집단에서 균질 (Homogeneous)의 표본을 추출하기 위해 상호 배타적 하위 표본들인 층들 (Strata)로 분할하는 방법이다<sup>[32]</sup>.

층화 샘플링 방법을 적용하기 위해 먼저 균질의 데이터를 추출하는 기준을 설정해야 한다. 본 논문에서는 그림 3에서 제시된 기존의 개발일정 추정 모델들이 갖는 분포를 분석하여 최소 값과 최대 값의 범위를 균질 데이터 추출 기준으로 선정하였다.

ISBSG Benchmark Release 8<sup>[22]</sup> 데이터들의 개발노력은  $0.03PM \leq E \leq 1,100PM$  범위를 갖고 있다. 그림 3에 제시된 개발일정 추정 모델들을 이 값에 대입하여 본 결과는 그림 4와 같다.

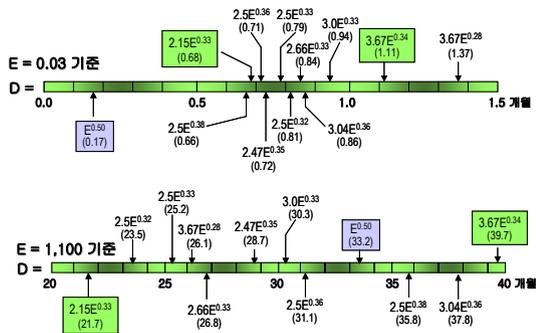


그림 4. 개발일정 모델 값 분포  
Fig. 4. Development Schedule Model Value Distribution

그림 4로부터  $T_c = E^{0.5}$ 는 최소 값을 갖지 못해  $T_c$ 로서 채택이 불가하며,  $T_n$ 의 일부분으로 포함시켜야만 한다. 다음으로  $2.15E^{0.33}$ 를 살펴보자. 그림 3에서  $2.15E^{0.33}$ 은  $T_c$ 와  $T_n$  모두에 거론되어 있다. 그러나 다른 모델들의 75%  $T_n$  값들이  $2.15E^{0.33}$ 보다 적기 때문에  $2.15E^{0.33}$ 는  $T_c$ 로 고려할 수 없으며,  $T_n$ 의 최소 값으로 고려해야만 한다.  $T_n$ 의 최대 값으로는  $3.67E^{0.34}$ 가 타당해 보인다. 따라서 표본으로 추출하기 위한 데이터 범위 기준은 그림 5와 같이 설정하였다.

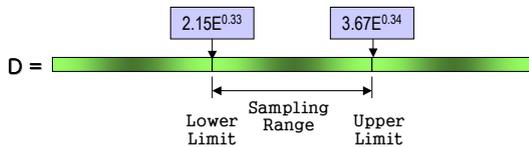
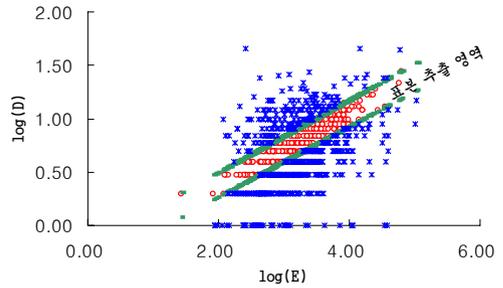


그림 5. 표본 데이터 추출 범위  
Fig. 5. Extraction Range of Sample Data

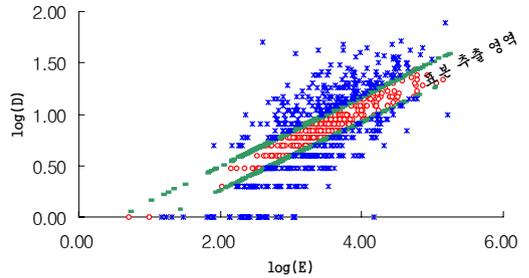
그림 5의 기준에 따라 ISBSG Benchmark Release 8<sup>[21]</sup>로부터 추출된 데이터는 표 2 및 그림 6과 같다.

표 2. 추출된 표본 현황  
Table 2. Extracted Sample Data

개발 구분	프로젝트 수	표본 추출		
		Lower Limit	추출된 표본 수	Upper Limit
성능개량	875개	363개	323개	189개
신규개발	720개	195개	291개	234개
계	1,595개	558개	614개	423개



(a) 성능개량 프로젝트



(b) 신규개발 프로젝트

그림 6. 추출된 표본 데이터 범위  
Fig. 6. Range of Extracted Sample Data

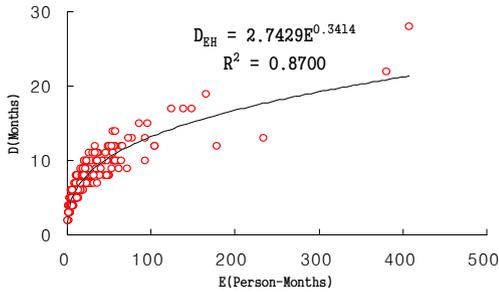
## 2. 명목상 개발일정 초과 비용-일정 타협

추출된 표본 데이터들에 기반하여 개발노력과 개발기간 관계로부터 회귀분석을 적용하여 명목상 개발일정 모델을 유도하면 그림 7과 같다.

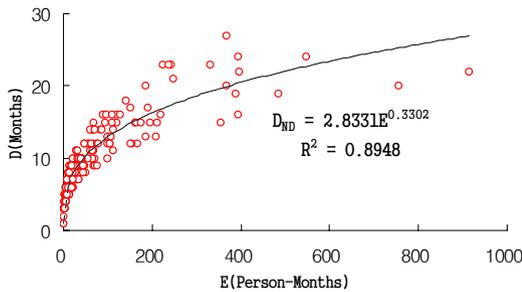
회귀분석 결과 기능점수에 기반한 성능개량 프로젝트의 명목상 개발일정  $D_{EH}$ 는  $2.7429E^{0.3414} \approx 2.74E^{0.34}$ 로, 신규개발 프로젝트의 명목상 개발일정  $D_{ND}$ 는  $2.8331E^{0.3302} \approx 2.83E^{0.33}$ 로 제시되었다. 제시된 모델들의 모수 값은 기존 명목상 개발일정 제시 모델들의 모수 값인  $1.0 \leq a \leq 3.67$ 과  $0.28 \leq b \leq 0.50$ 의 범위 내에 존재하며,  $a$  값은 대부분을 차지하고 있는 2.5~3.0에 포함되어 있으며,  $b$  값도 대부분을 차지하는 0.33에 근접한 결과를 나타내고 있어 기존 모델들과 유사한 결과를 나타내고 있다.

제안된 모델에  $T_c = 75\%T_n$ 과  $T_s = 130\%T_n$ 을 적용하면 그림 7의 결과를 얻는다. 그림 8은 비용-일정 타협이 마치 별 모양을 하고 있다. 이 모델을 CST-CORN 모델 (Cost-Schedule Trade-off Corn Model)이라 칭하자.

제안된 모델의  $T_c$ 와  $T_s$ 가 어떤 분포를 갖고 있는지 기존 모델들의 분포와 비교한 결과는 그림 9와 같다. 그림에서 제안 모델의  $T_c$ 와  $T_s$ 를 적용시 기존 모델들이 갖는 대부분의 값들을 표현할 수 있음을 알 수 있다.

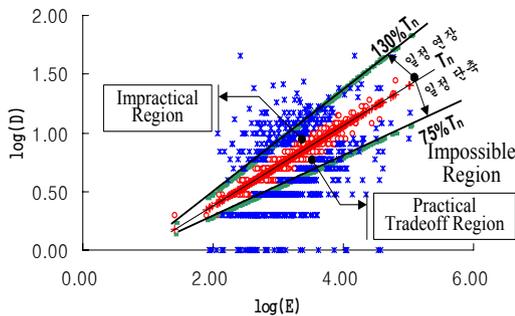


(a) 성능개량 프로젝트

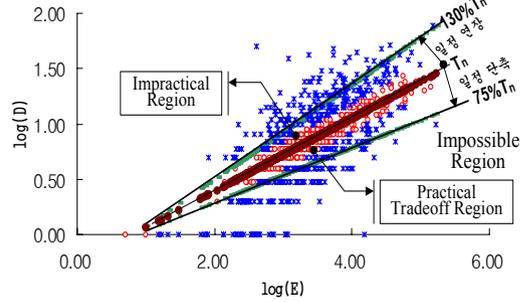


(b) 신규개발 프로젝트

그림 7. 명목상 개발일정 추정 모델  
Fig. 7. Nominal Development Schedule Estimation Model



(a) 성능개량 프로젝트



(b) 신규개발 프로젝트

그림 8. 개발노력과 개발일정 대안분석 뿔 모델  
Fig. 8. Alternative Analysis Corn Model of Development Effort and Schedule

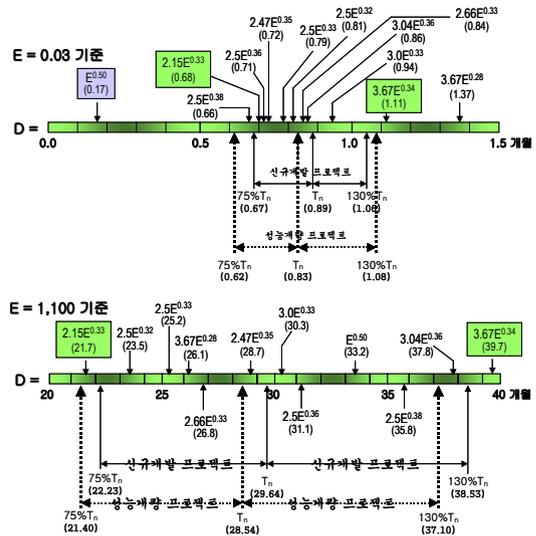


그림 9. 제안 모델의  $T_c$ 와  $T_s$  분포  
Fig. 9. Distribution of  $T_c$  and  $T_s$  of the Proposed Model

#### IV. 결론 및 향후 연구과제

소프트웨어를 개발시 일반적으로 고객은 보다 빠르게 보다 저렴하게 개발하도록 압력을 가하고 있다. 그러나 실제로 소프트웨어의 개발기간<sup>[33-35]</sup>을 더 이상 단축시킬 수 없는 개발 불가능 영역이 존재한다. 이 개념을 알고 있어야만 소프트웨어의 성공적인 개발이 가능하며, 입찰이나 계약시 보다 타당한 정보를 가질 수 있다.

기존에 제안된 개발기간 추정 모델들은 이러한 정보

를 고려하지 않고 현실적으로 타당하지 않은 단기간 또는 장기간의 개발기간 데이터들에 적합되어 현실적으로 적용시 괴리를 발생시키고 있다.

본 논문은 소프트웨어의 개발기간을 개발 가능영역, 최저 개발영역과 비현실적인 개발기간 영역에 대한 기준을 제시하였다. 기존 연구된 다양한 기준을 조합하여 5가지 가능한 기준을 유도하였다. 실제 수행된 프로젝트들을 대상으로 각각의 기준에 따라 실제 가능영역과 최적의 개발기간 영역의 데이터들을 대상으로 적합한 모델을 유도하였다.

본 제안된 모델은 개발노력에 따른 개발기간 추정 모델만을 제시하였으며, 소프트웨어 규모에 따라 개발기간이 어떠한 경향을 보이는지에 대한 기준은 제시하지 않았다. 따라서 추후 이 분야에 대한 연구를 수행할 예정이다.

## References

- [1] A. J. Shenher, "Improving PM: Linking Success Criteria to Project Type," Project Management Institute, Creating Canadian Advantage through Project Management Symposium, Calgary, 1996.
- [2] M. Mah, "Controlling Software Development," SPIN Newsletter, 1997.
- [3] M. Holtsinger, "CIS 4251/CIS 5930 Software Development," 1999.
- [4] D. R. Jones, "Project Scheduling," Augsburg College, 1999.
- [5] C. Carbone, "Optimal Resource Allocation for Projects," Project Management Journal, 1999.
- [6] B. W. Boehm, "Software Engineering Economics," Prentice Hall, 1981.
- [7] L. H. Putnam and W. Myers, "Familiar Metric Management - Time-to-Market," [http://www.qsm.com/fmm\\_08.pdf](http://www.qsm.com/fmm_08.pdf)
- [8] D. L. Hollowell, "Software Project Management Meets Six Sigma," <http://software.isixsigma.com/library/content/c030813a.asp>, iSixSigma Magazine, 2003.
- [9] Construx, "10 Deadly Sins of Software Estimation," Construx Software Builders, Inc., 2002.
- [10] C. Fakhrazadeh, "CORADMO in 2001: A RAD Odyssey," 16th International Forum on COCOMO and Software Cost Modeling, USC-CSE, 2001.
- [11] B. Boehm, B. Clark, E. Horowitz, R. Madachy, R. Shelby, and C. Westland, "The COCOMO 2.0 Software Cost Estimation Model," USC Center for Software Engineering, 1995.
- [12] J. A. Ward, "Productivity Through Project Management: Controlling the Project Variables," Information Management, 1994.
- [13] R. Pressman, C. Jones, V. Basilli, and L. Putnam, "16 Critical Software Practices, Estimate Cost and Schedule Empirically," [http://www.iceinusa.com/16CSP/content/\\_2-cost/anmetrgt.html](http://www.iceinusa.com/16CSP/content/_2-cost/anmetrgt.html)
- [14] L. K. Richard, "Staff IT," <http://www.gantthead.com>, 2005.
- [15] M. Holtsinger, "CIS 4251/CIS 5930 Software Development," 1999.
- [16] C. Abts, A. W. Brown, S. Chulani, B. K. Clark, E. Horowitz, R. Madachy, D. Reifer, and B. Steece, "Software Cost Estimation with COCOMO II," Prentice-Hall, 2000.
- [17] C. Jones, "Applied Software Measurements," McGraw Hill, 1996.
- [18] Marin Consultancy, "Estimation, Marin Solutions Technical Paper, 2001.
- [19] S. McConnell, "Rapid Development: Taming Wild Software Schedules," Microsoft Press, 1996.
- [20] C. E. Walston and C. P. Felix, "A Method of Programming Measurement and Estimation," IBM Systems Journal, Vol. 16, No. 1, pp. 54-73, 1977.
- [21] G. Booker, "Info 638 Software Project Management: Estimation, WBS, and Scheduling," 2000.
- [22] ISBSG, "Worldwide Software Development - The Benchmark Release 8," Victoria, Australia International Software Benchmarking Standards Group, 2004.
- [23] G. Sifri, "Accurate Estimates Critical for Software Development Projects," ESI International, Inc., 2001

- [24] R. Mukcher, "Business Case for A Common Services Based Architecture," OMG Document #3-300, 2000.
- [25] Y. Yang, Z. Chen, R. Valerd, and B. Boehm, "Effect of Schedule Compression on Project Effort," 5th Joint International Conference & Educational Workshop, the 15th International Annual Conference for the Society of Cost Estimating and Analysis and 27th Annual Conference of the International Society of Parametric Analysis, Denver, Colorado, USA., 2005.
- [26] L. H. Putnam and W. Myers, "Measures of Excellence: Reliable Software on Time, Withen Budget," Yourdon Press, 1992.
- [27] C. Jones, "Assessment and Control of Software Risks," Yourdon Press, 1994.
- [28] L. H. Putnam and W. Myers, "Five Core Metrics: The Intelligence Behind Successful Software Management," Dorset House Publishing, 2003.
- [29] L. H. Putnam and W. Myers, "What We Have Learned," The Journal of Defense Software Engineering, 2000.
- [30] C. Simons, "Software Sizing and Estimating: MK II," John Wiley & Sons, 1991.
- [31] M. Ardis, "Estimation: CSSE 372, Software Project Management," Rose-Hulman Institute, 2004.
- [32] C. W. Cook and J. Stubbendieck, "Range Research: Basic Problems and Techniques," Society for Range Management, Denvor, Co., pp. 232-241, 1986.
- [33] J. B. Park, H. S. Yang, "Quality Evaluation Method of Open Source Software," Journal of the Korea Academia-Industrial, Cooperation Society, Vol. 13, No. 5 pp. 2353-2359, 2012.
- [34] D. S. Kim, H. C. Kim, "The Study of Software Reliability Model from the Perspective of Learning Effects for Burr Distribution," Journal of the Korea Academia-Industrial Cooperation Society, Vol. 12, No. 10 pp. 4543-4549, 2011.
- [35] S. U. Lee, M. B. Choi, "A Definition and Evaluation Criteria for Software Development Success," Journal of The Institute of Internet, Broadcasting and Communication (IIBC), pp.233~241, vol. 12, no.2, April 2012.

## 저자 소개

### 이 상 윤(정회원)



- 1998년 ~ 2001년 : 경상대학교 컴퓨터과학과 (박사)
- 2007년 3월 ~ 현재 : 강릉원주대학교 과학기술대학 멀티미디어공학과 부교수
- <주관심분야 : 소프트웨어공학 등>
- e-mail : sulee@gwnu.ac.kr

### 최 명 복(중신회원)



- 1997년 ~ 현재 : 강릉원주대학교 멀티미디어공학과 교수
- 2004년 1월 ~ 현재 : 한국인터넷방송통신학회 이사
- <주관심분야 : 알고리즘 등>
- e-mail : cmb5859@gmail.com