

# 소프트웨어 개발노력 최적 분배에 관한 연구

이 상 운<sup>†</sup> · 김 용 수<sup>††</sup> · 한 판 암<sup>†††</sup>

## 요 약

성공적인 프로젝트 계획은 활용 가능한 일정과 더불어 프로젝트를 완수하는데 요구되는 노력을 얼마나 정확히 추정하느냐에 달려있다. 현존하는 소프트웨어 노력 추정 모델들은 생명주기 전반에 걸쳐 투입되는 총 개발노력과 단위시간당 소요되는 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 발생한다. 그 결과, 개발 세부단계별로 투입되는 노력의 규모도 프로젝트마다 차이가 발생한다. 본 논문은 소프트웨어 규모와 개발형태에 따라 계획, 명세, 구축, 시험과 이행단계에 투입될 노력을 추정하는 모델들을 제시하였다. 이 모델들은 155종의 다른 소프트웨어 프로젝트들로부터 유도되었다. 제안된 모델들은 프로젝트의 일정과 노력 분배 관리에 실질적인 지침을 제공할 것이다.

## A Study of Optimal Effort Allocation over Software Development Phase

Sang-Un Lee<sup>†</sup> · Young-Soo Kim<sup>††</sup> · Pan-Am Han<sup>†††</sup>

### 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. Existing software effort estimation models are present only the total effort and instantaneous effort function for the software life-cycle. Also, Putnam presents constant effort rate in each phase. However, the size of total effort varies according to the software projects under the influence of its size, complexity and operational environment. As a result, the allocated effort in each phase also differs from one project to another. This paper suggests models for effort allocation in planning, specifying, building, testing and implementing phases followed by the project size and development types. These models are derived from 155 different projects. Therefore, these models can be considered as a practical guideline in management of project schedule and effort allocation.

**키워드 :** 소프트웨어 개발노력(Software Development Effort), 개발 단계(Development Phase), 노력 분배(Effort Allocation), 중앙값(Median)

### 1. 서 론

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 분배와 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] 등이 있다. LOC는 코딩이 종료된 시점에서 정확한 측정이 가능하다. 이에 비해, 소프트웨어 개발 초기인 요구사항 명세화(Requirement Specification) 단계에서 개발노력과 비용을 추정하기 위한 방법으로 기능점수(Function Point, FP)를 분석하는 기법이 있다[5-8]. 이와 같은 연구들은 모두 소프트웨어 프로젝트의 생명주기 전반에 걸쳐 투입되는 총 개발노력과 비용을 추정하는 것이다. 또한, 추정된 총 개발노력을 소프트웨어 생명주기의 각 단계별로 어느 정도의 비율로 분배해야 하는지에 대한 연구는 Putnam[2]과 Yamada et al.[9]이 있다. Putnam[2]은 소프트웨어의 규모, 특성 등을 고려하지 않고 개발의 각 단계별로 일정한 값을 제시하였다. 또한, Yamada et al.[9]은 개발단계에 투

† 정 회 원 : 국방품질관리소 항공전자장비 및 소프트웨어 품질보증 담당  
 †† 정 회 원 : 거창전문대학 컴퓨터정보시스템과 교수  
 ††† 총신회원 : 경남대학교 컴퓨터공학과 교수  
 논문접수 : 2002년 4월 3일, 심사완료 : 2002년 7월 2일

입되는 총 인력의 약 40~50퍼센트가 소프트웨어 시험단계에 투입됨을 밝혔다. 그러나 개발되는 프로젝트의 규모, 복잡도, 운영환경 등의 차이로 인해 총 소요되는 개발노력의 규모뿐만 아니라 각 개발단계별로 투입되는 노력의 규모에도 심한 변동이 발생할 수 있어 일정한 값을 적용하는 데는 문제점이 있다. 또한, 연구 결과에 따라 개발의 각 단계별로 투입되는 노력의 양에도 심한 변동이 발생한다. 이러한 문제점을 해결하고, 개발의 각 단계별로 최적의 노력을 분배하기 위해, 본 논문은 최근에 실제 개발된 프로젝트들의 사례를 근거로 소프트웨어 개발 각 단계별로 최적의 노력을 분배하여 개발기간을 설정할 수 있는 모델을 제시하고자 한다.

제2장에서는 기존의 개발노력 분배비율에 관한 연구들을 살펴보고, 제3장에서는 실제 수행된 프로젝트들의 사례를 살펴본다. 제4장에서는 개발에 총 소요되는 노력에 대해 개발단계 수행 유형에 따라 최적으로 노력을 분배하여 개발기간을 설정할 수 있는 모델을 제시한다.

## 2. 기존 연구 및 연구배경

소프트웨어 생명주기(Life Cycle) 전반에 걸쳐 투입되는 노력에 관한 연구로 Putnam[2], Boehm[3], Nordan[10, 11]과 Pillai et al[12] 등이 있다. Norden[9]은 IBM에서 개발된 다양한 하드웨어 개발과정에 투입된 노력은 근사적으로 Rayleigh 분포를 따른다는 사실을 관찰하였다. 이후 Putnam[2]에 의해 이 관찰 결과가 소프트웨어 프로젝트에 적용되었다. 150개의 소프트웨어 프로젝트가 Norden[11]과 Putnam[2]에 의해 연구되었으며, 프로젝트 수준에서 소프트웨어 생명주기 전반에 걸쳐 소요되는 총 노력뿐만 아니라 생명주기의 각 단계(즉, 요구명세 분석, 설계, 코딩, 시험 및 유지보수 등)에 대한 노력도(그림 1)의 Rayleigh 분포를 따름을 관찰하였다. 임의의 시간  $t$  시점까지 프로젝트에 투입된 누적 노력  $y$ 를 식 (1)로, 임의의 시간  $t$  시점에서 프로젝트에 투입되는 노력  $\dot{y}$ 는 식 (1)을 미분한 식 (2)로 표현된다.

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

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

여기서  $E$ 는 Rayleigh 곡선 아래에 있는 영역으로 년 인원으로 표시되며, 소프트웨어 생명주기 전반에 걸쳐 투입된 총 인력(또는 노력)이다.  $a = 1/2t_d^2$ 이며,  $t_d$ 는  $\dot{y}$ 이 최대가 되는 시점으로 경험적으로 볼 때, 시스템이 운영되는 시점에 근접한다. 따라서,  $t_{y_{max}} = t_d$ 를 시스템의 개발기간이라 하며, 소프트웨어 생명주기 전반에 걸쳐 총 소요되는 개발노력의 약 40%에 도달하는 시점이다. 식 (2)에서  $a = 1/2t_d^2$ 를 치환하면, 식 (3)이 된다.

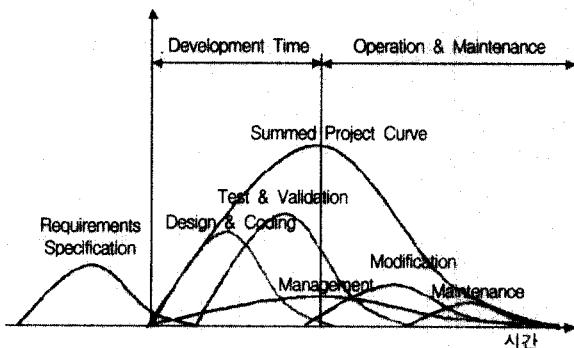
$$\dot{y} = \frac{E}{t_d^2} te^{-\frac{t^2}{2t_d^2}} \quad (3)$$

Boehm[3]은 소프트웨어 개발 초기에 느린 개발노력 형성 부분과 마지막 단계에서의 점차 길게 감소하는 Rayleigh 분포 특성이 대부분의 소프트웨어 프로젝트의 개발노력 곡선에 일치하지 않음을 지적하였다. 일반적으로 소프트웨어 프로젝트는 개발 초기에 하드웨어 프로젝트보다 빠른 형성 비율(Buildup Rate)을 가지며, 이는 Rayleigh 분포와 편향된 결과를 나타낸다. 이 연구 결과를 토대로, Pillai et al.[12]는 Putnam[2]의 개발노력 분포가 Rayleigh 분포를 따르는데 따른 문제점을 해결하기 위해 식 (4)의 Gamma 모델을 제시하였다.

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

이들 제안된 모델들을 이용하면 소프트웨어 프로젝트의 일정한 시점에서의 투입된 노력과 총 개발노력을 산정할 수 있어 사업초기에 의사결정 자료로 활용이 가능하다.

소프트웨어 생명주기 전반에 걸쳐 투입된 총 노력  $E$ 가 결정된 경우, 실제로 소프트웨어를 개발하는 각 세부단계인 요구사항 분석, 설계, 코딩, 시험과 유지보수의 각 단계별로 투입될 노력과 기간을 확정해야만 한다. 이에 대한 연구로, Putnam[2]과 Yamada et al.[9]등이 있다. Putnam[2]은 1960~1970년대에 개발된 150개 프로젝트에 대한 경험치를 적용하여 요구사항 명세화 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 확장단계는 55%와 관리부분에 10%가 소요됨을 밝혔다. 즉, 전체 투입노력에 대해 설계 및 코딩단계는 12%, 시험 및 검증단계는 16%, 변경, 유지보수와 확장단계는 44%, 관리는 8%가 소요된다. 또한, Yamada et al.[9]은 소프트웨어 개발 자원의 전체 양에서 약 40~50퍼센트가 시험단계에 소요됨



(그림 1) 소프트웨어 개발노력 분포

을 밝혔다. 이들 연구 결과를 종합한 결과는 <표 1>에 제시되어 있다.

이들 연구 결과를 토대로 개발단계별로 노력을 분배하는 데는 다음과 같은 문제가 발생한다.

<표 1> 개발단계별 투입 노력 비율

구분	요구사항 명세	설계, 코딩	시험, 검증	변경, 유지보수, 확장	관리	
Putnam [2]	요구사항명세 단계 제외시	20%	15%	20%	55%	10%
	요구사항명세 단계 포함시	20%	12%	16%	44%	8%
Yamada et al.[9]	-	-	40~50%	-	-	

- 이들 프로젝트들은 1970년대까지 개발된 프로젝트의 사례들로 보다 복잡해지고 고 신뢰성을 요구하고 있는 1990년대 이후 개발되고 있는 소프트웨어 프로젝트에 적용하는 데는 오차가 발생한다.
- 소프트웨어의 프로젝트 규모, 복잡도, 운영환경 등의 차이로 인해 프로젝트에 소요되는 총 개발노력에 차이가 발생하며, 개발 각 단계별 투입되는 개발노력의 비율에 관계없이 일정하게 결정된 값을 적용하는 데는 제한을 받는다.

이러한 문제점들을 해결하기 위해, 최근에 개발된 프로젝트들의 개발 사례를 살펴보고, 프로젝트 개발형태(신규 개발 프로젝트와 유지보수 프로젝트)와 소프트웨어 규모에 따라 개발노력을 단계별로 분배할 수 있는 모델을 제시한다.

### 3. 소프트웨어 규모별 개발노력 분배 사례

소프트웨어 생명주기의 각 단계별로 투입되는 노력을 분배하기 위해서는 먼저, 소프트웨어 프로젝트 개발에 투입될 총 노력을 결정하는 것이 필요하다. 신뢰성 있는 이들 정보들을 바탕으로 생명주기의 각 단계별로 투입될 노력을 설정해야만 최적의 결과를 얻을 수 있다.

ISBSG Benchmark Release 6[13] 데이터베이스 프로젝트들은 계획단계(Planning Phase), 명세단계(Specifying Phase), 구축단계(Building Phase), 시험단계(Testing Phase)와 이행단계(Implementing Phase)로 구분하여 각 단계별로 소요된 개발노력 정보를 갖고 있다. 계획단계는 실행가능성 고찰(Feasibility Study), 비용 효과 분석(Cost Benefit Analysis)과 프로젝트 착수(Project Initiation) 등을 고려하는 단계이며, 명세 단계에서는 시스템 분석(System Analysis), 기능적 설계(Functional Design), 외부 설계(External Design), 요구사항 정의(Requirements Definition)와 패키지 선택(Package Selection) 등의 업무를 수행한다. 구축단계는 물리적 설계(Physical Design), 내부 설계(Internal Design), 코딩(Coding) 또는 프로그래밍(Programming), 인터페이스(Interface)

등의 업무를 수행하는 단계이다. 시험단계는 통합시험(Integration Testing)과 사용자 수락시험(User Acceptance Testing)을 수행하는 단계이며, 이행단계는 이행(Implementation), 설치(Installation)와 사용자 훈련(User Training)을 수행하는 단계이다. 이들 단계를 Putnam[2]이 제시한 단계와 비교하면 <표 2>와 같다.

<표 2> 개발단계 비교

구분	Putnam[2]	ISBSG Benchmark Release 6[13]
개발 단계	요구명세	계획 명세
	설계 코딩	구축
	시험 및 검증	시험
	-	이행
	변경, 유지보수, 확장	-

계획단계를 P, 명세단계를 S, 구축단계를 B, 시험단계를 T, 이행단계를 I라 하자. 따라서 계획단계부터 이행단계까지 모두 수행한 프로젝트를 P-S-B-T-I 수행 프로젝트라 칭하자. 프로젝트별로 P-S-B-T-I 단계를 모두 수행하지 않고 특정 단계만을 수행한 프로젝트들이 다수 있다. 따라서, P-S-B-T-I 단계를 모두 수행한 프로젝트를  $P_{PSBTI}$ , S-B-T-I 단계만을 수행한 프로젝트를  $P_{SBTI}$ , P-S-B-T 단계만을 수행한 프로젝트를  $P_{PSBT}$ , S-B-T 단계만을 수행한 프로젝트를  $P_{SBT}$ , B-T-I 단계만을 수행한 프로젝트를  $P_{BTI}$ 라 하자. ISBSG Benchmark Release 6[13]의 789개 프로젝트들 중에서  $P_{PSBTI}$ 는 74건,  $P_{SBTI}$ 는 13건,  $P_{PSBT}$ 는 24건,  $P_{SBT}$ 는 15건,  $P_{BTI}$ 는 29건이다. 개발단계를 수행한 프로젝트들의 단계별 노력분배에 대한 평균값을 <표 3>에 제시하였다. 표에서 S-B-T-I의 표본 수는  $P_{PSBTI}$ 의 74건과  $P_{SBTI}$  13건을 합한 값이 된다. P-S-B-T, S-B-T와 B-T-I도 이와 같은 방법으로 표본수가 결정되었다.

<표 3> 개발단계별 노력분배 비율

개발단계 적용 프로젝트	노력 분배 비율 (%)					표본 수	
	계획(P)	명세(S)	구축(B)	시험(T)	이행(I)		
ISBSG [13]	P-S-B-T-I	6.72	19.56	40.42	20.38	12.91	74
	S-B-T-I	-	22.63	44.26	20.98	12.13	87
	P-S-B-T	8.45	22.00	48.22	21.32	-	98
	S-B-T	-	24.94	51.38	23.68	-	126
	B-T-I	-	-	38.66	22.77	38.57	116
Putnam[2]	20.00		12.00	16.00	44.00	150	
Yamada et al.[9]			40~50				

Putnam[2]이 제시한 단계별 노력분배 비율과 비교하여 보자. Putnam[2]은 요구명세 단계까지 약 20%가 소요된다고 제시하였다. 이에 대해 ISBSG Benchmark Release 6 [13] 데이터베이스 프로젝트들은 P-S-B-T-I 수행 프로젝트들은 P-S-B-T 수행 프로젝트에 포함되며, S-B-T-I 수

행 프로젝트는 S-B-T 수행 프로젝트에 포함되므로 P-S-B-T와 S-B-T 수행 프로젝트를 대상으로 요구명세단계에 소요되는 개발노력을 비교하여 보자. P-S-B-T 프로젝트는 30.45%(8.45% + 22.00%), S-B-T 프로젝트는 24.94%의 노력이 분배되어 Putnam[2]의 결과보다 많은 노력이 필요함을 알 수 있다. Putnam[2]은 설계 및 코딩단계는 12%, 시험 및 검증단계는 16%의 노력이 투입됨을 제시하였다. 구축과 시험단계를 비교하기 위해, ISBSG Benchmark Release 6[13] 데이터베이스 프로젝트들 중 P-S-B-T-I, S-I와 P-S-B-T 수행 프로젝트들은 S-B-T 수행 프로젝트 데이터에 포함되어 있으며, P-S-B-T-I, S-B-T-I 수행 프로젝트들은 B-T-I 수행 프로젝트들에 포함되어 있다. 따라서, S-B-T와 B-T-I 프로젝트 데이터를 기준으로 비교한다. 구축단계에 대해 S-B-T 프로젝트는 51.38%, B-T-I 프로젝트는 38.66%로 Putnam[2]이 제시한 개발노력 비율보다 월등히 많은 노력이 구축단계에 필요함을 알 수 있다. 또한, 시험단계도 S-B-T 프로젝트, B-T-I 프로젝트 결과를 근간으로 할 때, Putnam[2]의 결과보다 많은 노력이 필요함을 알 수 있다.

개발될 프로젝트의 규모, 개발노력과 개발기간이 결정되면, 적용 대상인 개발단계 결정 결과에 따라 해당 개발단계에 대한 위의 기준을 참조하여 개발 각 단계별로 소프트웨어 규모별 최적의 개발노력 분배 비율을 적용할 수 있다.

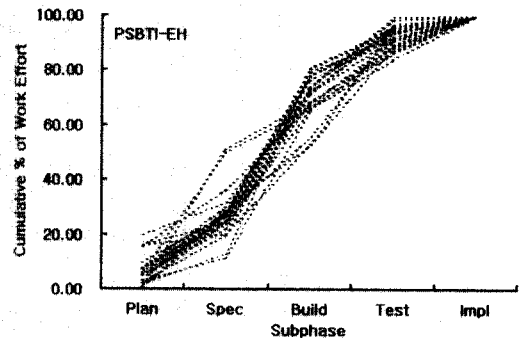
#### 4. 개발단계별 개발기간(노력 분배) 모델

제 3장에서 살펴본 바와 같이 개발 각 단계별로 투입되는 노력은 개발 세부단계를 어떤 것으로 적용할 것인가와 소프트웨어의 규모에 따라 변동이 발생한다. 본 장에서는 개발 각 단계별로 투입되는 노력을 추정할 수 있는 모델을 제시한다. Gokhale et al.[14, 15]은 통계적 소프트웨어 신뢰성 추정 모델을 제안함에 있어, 가정에 근거한 모델을 제시하는 방법 대신 실제 측정된 데이터로부터 경험적으로 모델을 결정하는 방법을 제시하였다. 본 논문도 개발단계에 투입되는 노력의 분포를 표현할 수 있는 모델을 제안함에 있어, 실제 수행된 프로젝트 데이터들로부터 얻은 분포의 특성을 표현할 수 있는 모델을 제시한다.

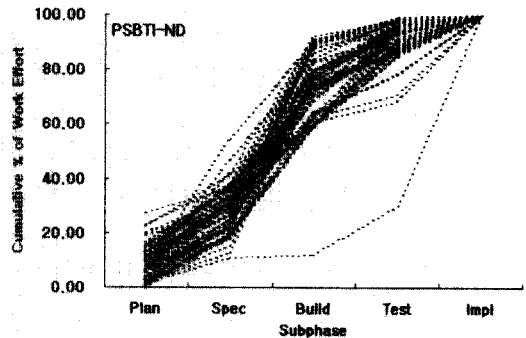
##### 4.1 P-S-B-T-I 단계 수행 프로젝트

P-S-B-T-I 개발단계를 모두 수행한 프로젝트들에 대한 단계별 누적 투입 노력이 어떤 분포를 따르는지 살펴보자. 유지보수 프로젝트는 (그림 2)에, 신규개발 프로젝트는 (그림 3)에 제시되어 있다. (그림 2)와 (그림 3)에서 유지보수 프로젝트나 신규개발 프로젝트들 대부분이 S자형 분포를 따르며, 일부는 지수형으로 증가하는 분포를 따름을 알 수 있다. 따라서, 이들 데이터들로부터 직접 모델을 제시할 수 있다. 그러나 프로젝트별로 각 단계별로 투입된 노력의 비

율은 심한 편차를 보임을 알 수 있다. 예로, 신규개발 프로젝트의 계획단계를 비교시 0.14%~23.24%의 편차를 갖고 있으며, 명세, 구축과 시험단계도 모두 유사한 편차를 보임을 알 수 있다. 따라서, 이들 전체 데이터를 대상으로 개발 노력의 비율을 추정하는 모델을 제안하는 대신 프로젝트의 규모별로 세분화하여 단계별 투입되는 노력의 비율을 추정하는 모델을 제안하고자 한다.



(그림 2) P-S-B-T-I 단계 수행 유지보수 프로젝트의 단계별 누적 노력 비율



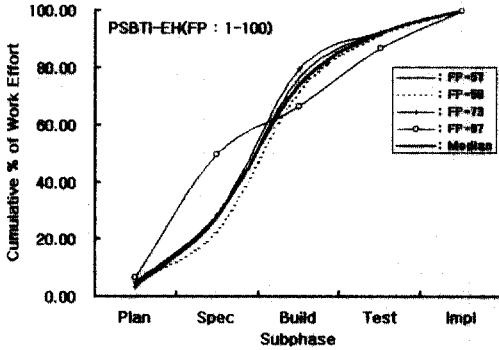
(그림 3) P-S-B-T-I 단계 수행 신규개발 프로젝트의 단계별 누적 노력 비율

(그림 2)의 유지보수 프로젝트들에 대해 소프트웨어의 규모인 기능점수  $FP$  가  $1 \leq FP \leq 100$ 인 프로젝트들에 대한 분포는 (그림 4)에,  $101 \leq FP \leq 200$ 인 프로젝트들에 대한 분포는 (그림 5)에 제시되어 있다. (그림 4)와 (그림 5)와 같이 기능점수 규모별로 세분화하면 세분화된 규모 범위에 대해서는 단계별로 투입되는 노력의 비율 편차가 상당히 감소함을 알 수 있다. 따라서, 이들 그래프에서 중앙값을 취해 소프트웨어 각 규모별 개발노력의 비율을 추정하는 모델을 제안한다. P-S-B-T-I 단계 수행 프로젝트에 대한 소프트웨어 규모인 기능점수  $FP$  범위별로 단계별 개발노력 분배 비율은 <표 4>에 제시되어 있다. 표에서, 제시된 단계별 개발노력 분배 비율은 해당 소프트웨어 규모인  $FP$  범위에 있는 표본들에 대해 중앙값을 취하였다. 이후 모든 결과는 이 방법을 적용하였다. P-S-B-T-I를 수행한 유지보수 프로젝트의 규모별 투입 노력 비율에 대한 누적 분포는

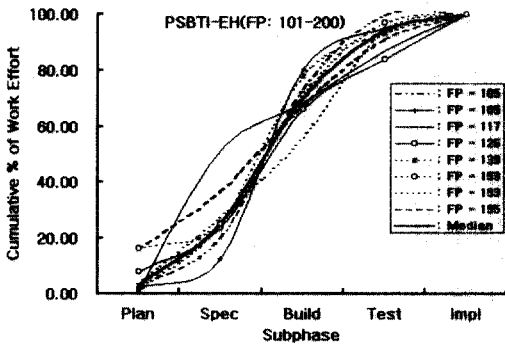
(그림 6)에, 신규개발 프로젝트의 규모별 투입 노력 비율에 대한 누적 분포는 (그림 7)에 제시되어 있다. Gokhale et al.[17, 18]은 통계적 소프트웨어 신뢰성 추정 모델을 제안함에 있어, 가정에 근거한 모델을 제시하는 방법 대신 실제 측정된 데이터로부터 경험적으로 모델을 결정하는 방법을 제시

하였다. 본 논문도 개발 단계에 투입되는 노력의 분포를 표현할 수 있는 모델을 제안함에 있어, 실제 수행된 프로젝트 데이터들로부터 얻은 분포의 특성을 표현할 수 있는 모델을 제시한다.

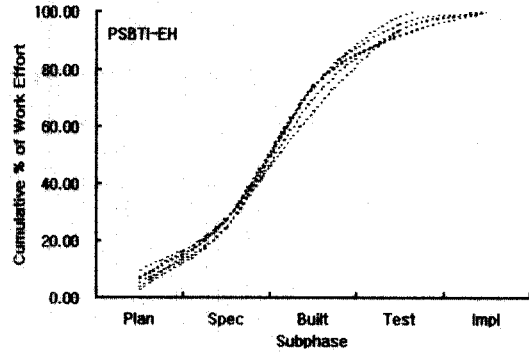
그림에서 보는 바와 같이, 개발 각 단계를 거치면서 투입된 누적 노력의 비율은 S자형(S-shaped) 또는 지수형 분포를 따른다.



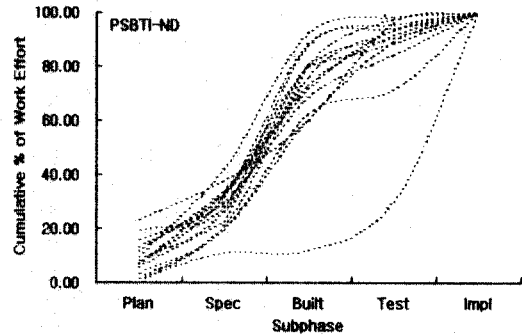
(그림 4) 유지보수 프로젝트의 누적 노력 비율(  $1 \leq FP \leq 100$  )



(그림 5) 유지보수 프로젝트의 누적 노력 비율 (  $101 \leq FP \leq 200$  )



(그림 6) P-S-B-T-I 단계 수행 유지보수 프로젝트의 소프트웨어 규모별 누적 투입 노력 비율(증앙값)



(그림 7) P-S-B-T-I 단계 수행 신규개발 프로젝트의 소프트웨어 규모별 누적 투입 노력 비율(증앙값)

<표 4> P-S-B-T-I 수행 프로젝트의 단계별 누적 노력 비율

개발 구분	FP규모	개발 단계별 누적 노력 비율 (%)				
		P	S	B	T	I
유지보수 프로젝트	1~100	4.15	27.98	74.05	91.95	100.00
	101~200	2.92	24.86	70.07	93.92	100.00
	201~300	7.27	27.87	65.29	93.97	100.00
	301~400	6.71	27.44	74.63	92.41	100.00
	401~500	5.48	25.23	74.00	95.97	100.00
	1001~1200	9.86	27.45	73.90	98.65	100.00
신규개발 프로젝트	1~100	12.95	32.98	79.96	90.68	100.00
	101~200	8.13	26.14	80.57	97.65	100.00
	201~300	3.33	20.69	76.14	93.42	100.00
	301~400	11.70	33.82	72.94	91.73	100.00
	401~500	6.81	31.45	88.29	94.06	100.00
	501~600	6.86	32.57	71.57	93.60	100.00
	601~700	19.13	27.87	87.87	95.30	100.00
	801~900	9.54	41.70	92.31	98.21	100.00
	901~1000	23.24	37.57	60.00	96.22	100.00
	1001~1200	1.27	10.83	12.10	29.94	100.00
	1201~1400	15.72	29.44	68.58	84.38	100.00
	1401~1600	5.77	22.50	80.39	88.32	100.00
1801~2000	1.56	19.22	62.63	71.19	100.00	
4001~5000	0.14	19.41	60.31	89.22	100.00	

이와 같은 현상으로부터 지수형, S자형 또는 지수와 S자 혼합형의 모델을 제안할 수 있다. 그러나, 좋은 모델이란 모델의 성능이 좋아야 하며, 단순하면서도 널리 사용 가능해야 한다. 이와 같은 모델 선정 조건에 따르면, 지수와 S자형의 모델로는 단순 조건은 만족하지만 특정 개발단계만을 수행한 프로젝트에 대해서는 모델의 성능을 만족시킬 수 없는 경우가 다수 발생할 수 있다. 또한, 지수와 S자의 혼합형 모델을 사용할 경우, 모델의 성능은 만족시킬 수 있을지라도 단순 조건을 만족시키지 못한다. 따라서 단순 조건과 모델의 성능 모두를 만족시킬 수 있으면서, 다양한 개발단계를 수행하는 프로젝트 유형에 모두 적용할 수 있는 모델을 제안하고자 한다. 제안된 모델은 개발단계별로 투입되는 노력의 비율(Effort Rate,  $E_{rate}$ )은 S자형과 지수형을 모두 표현 가능한 식 (5)를 따르는 모델로 표현된다.

$$E_{rate}(\%) = a \frac{(1 - e^{-bt})}{(1 + ce^{bt})}, t = 1, 2, 3, 4, 5 \quad (5)$$

여기서,  $a$ 는 소프트웨어 개발단계에 투입된 총 노력의 비율로 100%이며,  $b$ 와  $c$ 는 상수이다.  $t$ 는 개발의 각 단계별로  $P$ 는 1,  $S$ 는 2,  $B$ 는 3,  $T$ 는 4로,  $I$ 는 5로 치환하면 된다.  $b$ 와  $c$  모수는 비선형 최소자승 절차 (Nonlinear Least Squares Procedure)를 사용해 추정하였다. 모수 추정 결과와 모델의 성능은 <표 5>에 요약되어 있다.

<표 5> P-S-B-T-I 단계 수행 프로젝트 모수 추정과 모델 성능

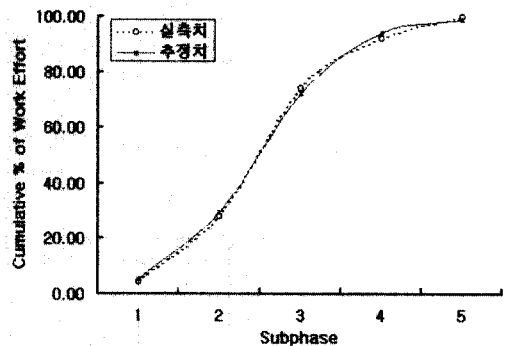
개발 구분	FP규모	모수추정값		모델 성능		
		$\hat{b}$	$\hat{c}$	$R^2$	MMRE	MSE
유지 보수 프로젝트	1~100	1.808	86.000	0.9980	8.85%	2.5833
	101~200	1.811	100.000	0.9985	14.94%	1.7410
	201~300	1.692	76.500	0.9977	7.22%	3.1140
	301~400	1.897	107.076	0.9977	7.11%	2.5904
	401~500	2.005	147.500	0.9982	6.48%	1.0498
신규 개발 프로젝트	1001~1200	1.997	134.000	0.9950	12.44%	7.9293
	1~100	1.696	49.000	0.9869	11.49%	13.0280
	101~200	2.208	201.000	0.9956	13.17%	5.7698
	201~300	2.305	344.500	0.9973	7.54%	3.4193
	301~400	1.511	34.538	0.9973	6.21%	2.8004
	401~500	2.597	371.000	0.9949	12.59%	7.9364
	501~600	1.695	60.156	0.9995	1.15%	0.5514
	601~700	2.295	194.595	0.9608	20.96%	55.0752
	801~900	2.495	190.511	0.9969	9.87%	4.7472
	901~1000	1.105	11.260	0.9337	14.68%	53.0024
	1001~1200	1.509	521.322	0.8756	41.45%	147.7274
	1201~1400	1.295	23.598	0.9789	12.57%	16.6573
	1401~1600	2.305	299.875	0.9844	14.89%	20.5162
	1801~2000	1.305	41.005	0.9517	67.92%	49.7499
	4001~5000	1.895	192.890	0.9979	386.61%	2.7811

표에서 모델의 평가기준으로  $R^2$ , MMRE를 제시하고 있다. 다른 여러 가지 모델들을 비교하는데 있어, 어떤 의미 있는 척도로서 모델의 추정 정확도를 평가하는 것이 필요하다. 회귀분석의 경우 회귀직선에 의해 종속변수가 설명되는 정도를 결정계수(Coefficient of determination,  $R^2$ )라 한다. 종속변수의 값은 독립변수에 의해 결정되는 부분과 미지의 오차의 합으로 나타나며, 총 변동 중에서 회귀직선에 의해 설명되는 비율이  $R^2$ 이다. 따라서,  $R^2(0 \leq R^2 \leq 1)$ 이 0에 가까우면 추정된 회귀직선은 쓸모가 없으며, 값이 클수록 쓸모 있는 회귀직선이 된다. MMRE를 살펴보자. 상대오차(Relative Error, RE)는  $\frac{(\text{실측치} - \text{추정치})}{\text{실측치}} \times 100\%$ 로 구해지며, MRE (Magnitude of the RE)는  $|RE|$ 이며,  $n$ 개의 데이터에 한 MMRE (Mean MRE)는  $\frac{1}{n} \sum_i MRE_i, i = 1, 2, \dots, n$ 로 계산된다. MMRE가 작은 값이면 모델은 평균적으로 좋은 모델임을 알 수 있다. Conte et al.[16]는  $MMRE \leq 0.25$  (25%) 이면 개발노력을 예측하는 모델로 채택 가능한 것으로 고려하였다. 예로, P-S-B-T-I 단계를 수행한 유지보수

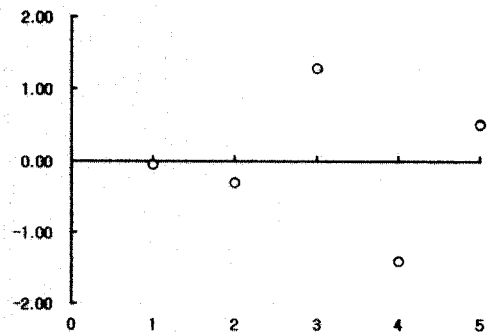
프로젝트에 대해,  $1 \leq FP \leq 100$  인 프로젝트에 대한 개발 각 단계별로 투입된 누적 노력의 비율은 식 (6)의 모델로 추정된다. 식 (6)의 추정치와 실측치를 비교한 결과는 (그림 8)에 제시되어 있다.

$$\hat{E}_{rate}(\%) = 100 \cdot \frac{(1 - e^{-1.808 \cdot t})}{(1 + 86.000 e^{-1.808 \cdot t})}, t = 1, 2, 3, 4, 5 \quad (6)$$

회귀분석에 있어서 결정계수 이외에 유의성 검정과 잔차 분석을 통해 모델이 좋은지 여부를 결정한다. 잔차(Residual)는 실제 값과 추정된 값과의 차이로, 잔차분석은 단순 회귀모델에서 설정한 등분산성, 독립성, 정규성과 직선관계의 가정이 옳은가를 토할 때 가장 많이 사용되는 방법이다. 따라서, 잔차가 어떤 일정한 형태를 취하지 않고 랜덤하게 분산되어 있는 경우가 좋은 모델이 된다. (그림 9)의 잔차분석 결과 잔차들이 일정한 경향을 보이지 않고 랜덤한 결과를 보여 좋은 모델로 판명되었다. 모델의 적합성을 평가하기 위해, 분산분석결과는 <표 6>에 제시되어 있다. 제안된 모



(그림 8) P-S-B-T-I 단계 수행 유지보수 프로젝트의 단계별 노력 비율 추정 결과 ( $1 \leq FP \leq 100$ )



(그림 9) 잔차분석

<표 6> 분산분석

요인	제공합	자유도	평균제곱	F	$F(1, 3; 0.05)$
회귀	6951.6538	1	6951.6538	2014.0497	10.13
잔차	10.3547	3	3.4516		
계	6962.0085	4			

델의 유의성 검정결과  $F \geq F(1, 3; 0.05) = 2014.0497 > 10.13$  으로 회귀 분석이 유의함을 알 수 있다.

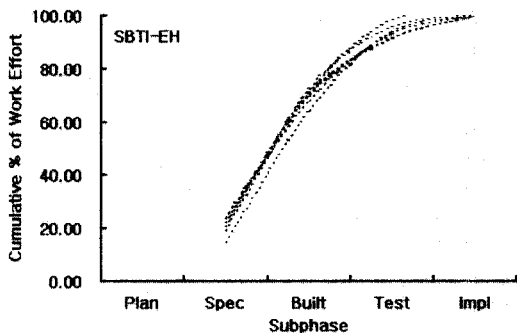
4.2 S-B-T-I 단계 수행 프로젝트

S-B-T-I 단계 수행 프로젝트들은  $P_{PSBTI}$  의 74건과 S-B-T-I 단계에 투입된 개발노력 데이터와  $P_{SBTI}$  13건을 합한 87건으로 <표 7>과 같이 단계별 누적 노력 분배 비율이 계산되었다. S-B-T-I 단계를 수행한 유지보수 프로젝트의 규모별 투입 노력 비율에 대한 누적 분포는 (그림 10)에, 신규개발 프로젝트의 규모별 투입 노력 비율에 대한 누적 분포는 (그림 11)에 제시되어 있다.

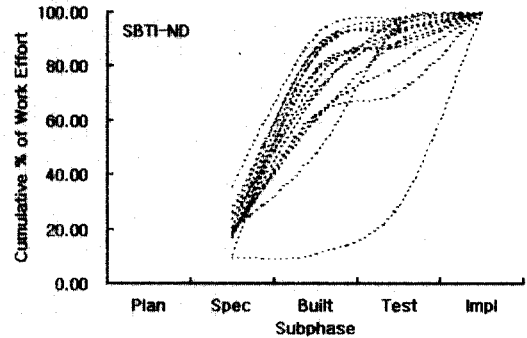
그림에서 보는 바와 같이, 개발 각 단계를 거치면서 투입된 누적 노력의 비율은 S자형 분포를 따른다. 따라서 개발단계별로 투입되는 노력의 비율은 식 (5)를 따르는 모델로 표현되며, 단지 차이점은  $t$ 에 대해  $P$ 를 제외하고,  $S$ 는 2,  $B$ 는 3,  $T$ 는 4로,  $I$ 는 5로 치환하면 된다. 모수 추정

<표 7> S-B-T-I 단계 수행 프로젝트의 단계별 누적 노력 비율

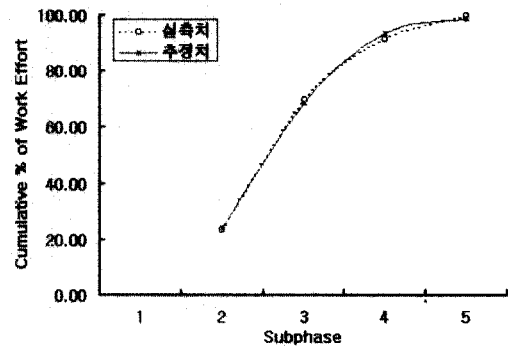
개발구분	FP규모	개발 단계별 누적 노력 비율 (%)				
		P	S	B	T	I
유지보수 프로젝트	1~100		23.66	69.91	91.59	100.00
	101~200		24.18	68.05	93.42	100.00
	201~300		14.84	64.26	93.93	100.00
	301~400		22.22	70.76	91.84	100.00
	401~500		20.90	72.49	95.74	100.00
	1001~1200		19.51	71.04	98.50	100.00
신규개발 프로젝트	1~100		21.00	78.06	88.90	100.00
	101~200		17.90	69.46	97.45	100.00
	201~300		16.96	74.85	93.38	100.00
	301~400		23.72	70.91	92.72	100.00
	401~500		23.98	86.90	93.42	100.00
	501~600		20.51	84.22	96.49	100.00
	601~700		10.81	85.00	94.19	100.00
	701~800		26.33	86.84	97.77	100.00
	801~900		35.55	91.50	98.02	100.00
	901~1000		18.66	47.89	95.07	100.00
	1001~1200		9.68	10.97	29.03	100.00
	1201~1400		17.60	61.32	81.06	100.00
	1401~1600		17.76	79.19	87.61	100.00
	1801~2000		17.94	62.04	70.73	100.00
4001~5000		19.30	60.26	89.20	100.00	



(그림 10) S-B-T-I 단계 수행 유지보수 프로젝트의 소프트웨어 규모별 누적 투입 노력비율 추정 결과



(그림 11) S-B-T-I 단계 수행 신규개발 프로젝트의 소프트웨어 규모별 누적 투입 노력 비율



(그림 12) S-B-T-I 단계 수행 유지보수 프로젝트의 단계별 투입 누적 노력 비율 추정 결과 ( $1 \leq FP \leq 100$ )

<표 8> S-B-T-I 단계 수행 프로젝트의 모수 추정과 모델 성능

개발구분	FP규모	모수추정값		모델 성능		
		$\hat{b}$	$\hat{c}$	$R^2$	MMRE	MSE
유지보수 프로젝트	1~100	1.912	138.764	0.9970	1.86%	1.7782
	101~200	1.893	135.017	0.9996	0.61%	0.2818
	201~300	2.208	429.495	0.9997	2.36%	0.5495
	301~400	2.015	183.922	0.9962	2.25%	2.4542
	401~500	2.282	357.522	0.9998	0.32%	0.1088
	1001~1200	2.305	403.881	0.9990	1.07%	1.4565
신규개발 프로젝트	1~100	2.392	416.667	0.9761	4.45%	18.6941
	101~200	2.307	443.871	0.9997	1.14%	0.7654
	201~300	2.505	666.022	0.9946	3.47%	4.4966
	301~400	1.995	166.975	0.9981	1.23%	1.1317
	401~500	2.805	809.806	0.9928	3.88%	9.0400
	501~600	2.636	640.237	0.9996	4.78%	5.4164
	601~700	2.505	578.778	0.9625	25.84%	46.2054
	701~800	2.705	587.958	0.9997	1.39%	1.4151
	801~900	2.905	598.611	0.9990	0.59%	0.5787
	901~1000	1.993	350.811	0.9740	12.04%	23.3046
	1001~1200	1.531	590.669	0.8430	41.67%	184.9621
	1201~1400	1.591	87.601	0.9729	8.65%	17.4347
	1401~1600	2.505	587.174	0.9648	7.39%	28.7137
	1801~2000	1.359	49.143	0.8952	14.38%	60.7294
4001~5000	1.803	147.510	0.9987	1.00%	1.0246	

결과와 모델의 성능은 <표 8>에 요약되어 있다. 유지보수 프로젝트의  $1 \leq FP \leq 100$  범위에 있는 프로젝트의 투입노력 비율에 대한 실측치와 추정치를 비교한 결과는 (그림 12)에 제시되어 있다.

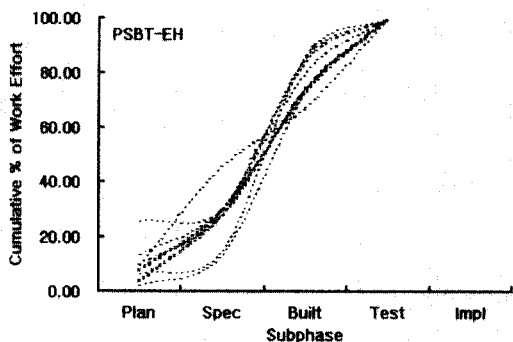
4.3 P-S-B-T 단계 수행 프로젝트

P-S-B-T 개발단계 수행 프로젝트로  $P_{PSBT}$ 의 74건에 대한 P-S-B-T 단계에 투입된 개발노력 데이터와  $P_{PSBT}$  24건을 합한 98건으로 <표 9>와 같이 단계별 누적 노력 분배 비율이 계산되었다. P-S-B-T를 수행한 유지보수 프로젝트의 규모별 투입 노력 비율에 대한 누적 분포는 (그림 13)에, 신규개발 프로젝트의 규모별 투입 노력 비율에 대한 누적 분포는 (그림 14)에 제시되어 있다.

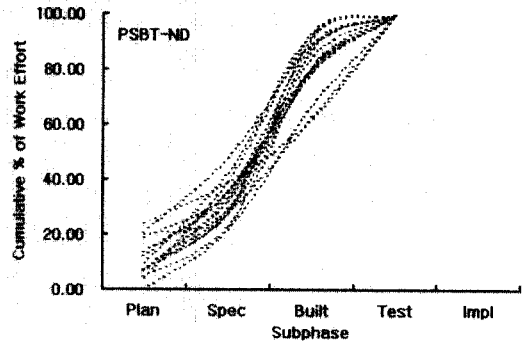
개발단계별로 투입되는 노력의 비율은 식 (5)를 따르는 모델로 표현되며, 단지 차이점은  $t$ 에 대해  $I$ 를 제외하고,  $P$ 는 1,  $S$ 는 2,  $B$ 는 3,  $T$ 는 4로 치환하면 된다. 모수 추정 결과와 모델의 성능은 <표 10>에 요약되어 있다. 유지보수 프로젝트의  $1 \leq FP \leq 100$  범위에 있는 프로젝트의 투

<표 9> P-S-B-T 단계 수행 프로젝트의 단계별 누적 노력 비율

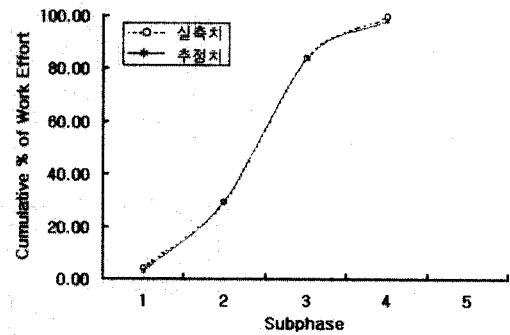
개발 구분	FP규모	개발 단계별 누적 노력 비율 (%)				
		P	S	B	T	I
유지보수 프로젝트	1~100	4.25	29.66	84.34	100.00	
	101~200	3.70	28.04	74.00	100.00	
	201~300	8.23	29.65	73.01	100.00	
	301~400	7.30	29.71	80.61	100.00	
	401~500	13.48	29.65	74.53	100.00	
	501~600	9.57	45.70	67.23	100.00	
	701~800	2.22	13.33	73.33	100.00	
	1001~1200	10.00	27.83	74.91	100.00	
	1201~1400	25.40	29.97	85.97	100.00	
	2001~3000	6.44	15.03	86.65	100.00	
신규개발 프로젝트	1~100	13.81	35.22	89.34	100.00	
	101~200	7.13	28.23	87.82	100.00	
	201~300	4.99	23.15	81.79	100.00	
	301~400	12.48	36.02	81.12	100.00	
	401~500	6.94	33.44	85.27	100.00	
	501~600	7.45	32.19	79.62	100.00	
	601~700	20.07	29.24	92.20	100.00	
	801~900	9.71	42.46	93.99	100.00	
	901~1000	24.16	39.04	62.36	100.00	
	1001~1200	6.97	27.88	63.15	100.00	
	1201~1400	19.08	35.10	81.27	100.00	
	1401~1600	7.26	27.74	80.51	100.00	
	1801~2000	4.08	39.44	93.08	100.00	
	2001~3000	20.86	48.00	89.59	100.00	
4001~5000	0.15	21.76	67.60	100.00		



(그림 13) P-S-B-T 단계 수행 유지보수 프로젝트의 소프트웨어 규모별 누적 투입 노력 비율



(그림 14) P-S-B-T 단계 수행 신규개발 프로젝트의 소프트웨어 규모별 누적 투입 노력 비율



(그림 15) P-S-B-T 단계 수행 유지보수 프로젝트의 단계별 투입 누적노력 비율 추정 결과 ( $1 \leq FP \leq 100$ )

<표 10> P-S-B-T 단계 수행 프로젝트의 모수 추정결과와 모델 성능

개발 구분	FP규모	모수추정값		모델 성능		
		$\hat{b}$	$\hat{c}$	$R^2$	MMRE	MSE
유지보수 프로젝트	1~100	2.512	350.000	0.9994	7.37%	0.9232
	101~200	2.098	173.718	0.9996	3.76%	4.4143
	201~300	1.905	102.070	0.9926	11.06%	8.6494
	301~400	2.205	181.466	0.9969	12.17%	4.3957
	401~500	1.805	73.940	0.9799	17.00%	21.4492
	501~600	1.395	20.973	0.9542	13.94%	35.6285
	701~800	2.635	1041.600	0.9983	16.59%	3.7028
	1001~1200	2.000	129.021	0.9902	15.78%	12.0102
	1201~1400	1.799	52.761	0.8861	27.35%	106.5232
	2001~3000	3.210	2780.000	0.9919	27.54%	11.3570
신규개발 프로젝트	1~100	2.301	155.908	0.9822	18.82%	23.2990
	101~200	2.795	646.506	0.9966	18.83%	6.3361
	201~300	2.695	712.072	0.9084	16.21%	2.9516
	301~400	1.887	67.633	0.9906	12.46%	10.4629
	401~500	2.395	229.339	0.9983	11.00%	2.6834
	501~600	2.095	132.351	0.9971	9.32%	3.7523
	601~700	2.605	352.169	0.9447	26.35%	80.7110
	801~900	2.597	225.310	0.9949	13.47%	7.5686
	901~1000	1.100	10.080	0.8759	17.27%	73.7714
	1001~1200	1.795	101.665	0.9785	14.06%	21.4853
	1201~1400	1.691	40.449	0.9605	18.09%	36.2573
	1401~1600	2.305	245.350	0.9965	14.41%	5.1609
	1801~2000	2.995	606.272	0.9999	6.67%	0.3242
	2001~3000	1.652	22.215	0.9758	11.00%	19.6581
4001~5000	2.203	322.340	0.9944	383.86%	8.3473	

입노력 비율에 대한 실측치와 추정치를 비교한 결과는 (그림 15)에 제시되어 있다.



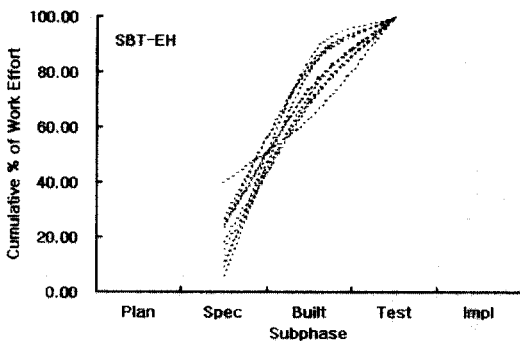
4.4 S-B-T 단계 수행 프로젝트

S-B-T 개발단계 수행 프로젝트로  $P_{FSBT}$ 의 74건에 대한 S-B-T 단계에 투입된 개발노력 데이터,  $P_{SBTI}$  13건에 대한 S-B-T 단계에 투입된 개발노력 데이터,  $P_{FSBT}$  24건에 대한 S-B-T 단계에 투입된 개발노력 데이터와  $P_{SBT}$  15건을 합한 122건으로 <표 11>과 같이 단계별 노력 분배 비율이 계산되었다.

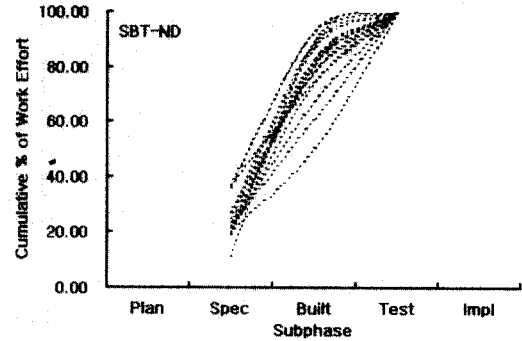
<표 11> S-B-T 단계 수행 프로젝트의 단계별 누적 노력 비율

개발 구분	FP규모	개발 단계별 누적 노력 비율 (%)				
		P	S	B	T	I
유지 보수 프로젝트	1~100		26.88	82.11	100.00	
	101~200		23.66	74.69	100.00	
	201~300		15.75	69.15	100.00	
	301~400		24.64	83.31	100.00	
	401~500		18.37	70.04	100.00	
	501~600		39.95	63.76	100.00	
	701~800		11.36	72.73	100.00	
	1001~1200		24.77	74.43	100.00	
	1201~1400		6.13	81.19	100.00	
신규 개발 프로젝트	2001~3000		9.18	85.73	100.00	
	1~100		23.61	82.58	100.00	
	101~200		20.10	83.32	100.00	
	201~300		19.87	81.32	100.00	
	301~400		27.36	78.63	100.00	
	401~500		24.89	87.34	100.00	
	501~600		21.62	81.41	100.00	
	601~700		11.48	90.24	100.00	
	701~800		26.93	88.83	100.00	
	801~900		36.27	93.35	100.00	
	901~1000		19.63	50.37	100.00	
	1001~1200		22.16	61.07	100.00	
	1201~1400		19.12	74.73	100.00	
1401~1600		22.09	78.89	100.00		
1801~2000		37.10	92.89	100.00		
2001~3000		36.70	74.94	100.00		
4001~5000		21.64	67.56	100.00		

S-B-T를 수행한 유지보수 프로젝트의 규모별 투입 노력 비율에 대한 누적 분포는 (그림 16)에, 신규개발 프로젝트의 규모별 투입 노력 비율에 대한 누적 분포는 (그림 17)에 제시되어 있다. 개발단계별로 투입되는 노력의 비율은 식 (5)



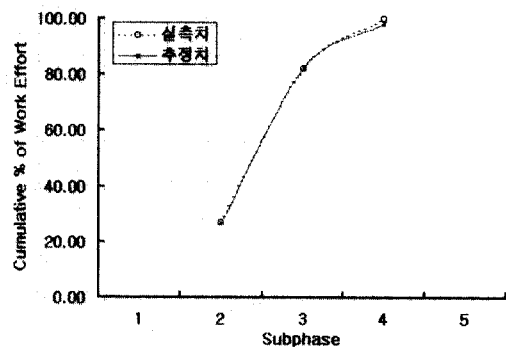
(그림 16) S-B-T 단계 수행 유지보수 프로젝트의 소프트웨어 규모별 누적 투입 노력 비율



(그림 17) S-B-T 단계 수행 신규개발 프로젝트의 소프트웨어 규모별 누적 투입 노력 비율

<표 12> S-B-T 단계 수행 프로젝트의 모수 추정과 모델 성능

개발 구분	FP규모	모수 추정값		모델 성능			
		$\hat{b}$	$\hat{c}$	$R^2$	MMRE	MSE	
유지 보수 프로젝트	1~100	2.551	446.458	0.9988	0.82%	0.9361	
	101~200	2.305	325.519	0.9959	1.84%	3.5202	
	201~300	2.506	798.782	0.9971	1.46%	4.0088	
	301~400	2.730	714.948	0.9995	0.47%	0.5496	
	401~500	2.425	586.724	0.9952	2.79%	4.5052	
	501~600	1.403	25.921	0.8548	9.63%	48.6910	
	701~800	2.515	798.527	0.9991	15.75%	12.7297	
	1001~1200	2.278	296.860	0.9940	2.63%	4.0969	
	1201~1400	2.705	1153.800	0.9831	58.13%	50.9521	
	2001~3000	2.805	1247.200	0.9827	35.03%	44.4763	
	신규 개발 프로젝트	1~100	2.745	782.649	0.9994	0.62%	0.5950
		101~200	2.749	867.795	0.9996	4.16%	2.8961
		201~300	2.889	1310.287	0.9995	0.77%	0.5427
301~400		2.300	261.159	0.9977	1.07%	2.2899	
401~500		2.805	767.273	0.9994	2.76%	2.0677	
501~600		2.705	782.949	0.9998	1.45%	0.9311	
601~700		3.411	4784.944	0.9932	15.35%	15.2349	
701~800		3.095	1325.808	0.9998	0.37%	0.1190	
801~900		3.207	1069.728	0.9999	0.10%	0.0275	
901~1000		2.093	414.870	0.9287	17.20%	49.3894	
1001~1200		2.015	227.810	0.9657	8.35%	22.2080	
1301~1400		2.585	760.296	0.9977	1.86%	2.1307	
1401~1600		2.611	656.727	0.9987	1.11%	1.2504	
1801~2000	3.101	834.541	0.9999	0.13%	0.0392		
2001~3000	1.889	78.836	0.9718	4.50%	10.2612		
4001~5000	2.182	303.252	0.9872	4.60%	9.2816		



(그림 18) S-B-T 단계 수행 유지보수 프로젝트의 단계별 투입 누적노력 비율 추정 결과 ( $1 \leq FP \leq 100$ )

의 모수를 조절하면 지수형 분포를 표현할 수 있다. 단지 차이점은  $t$ 에 대해  $P$ 와  $I$ 를 제외하고,  $S$ 는 2,  $B$ 는 3,  $T$ 는 4로 치환하면 된다. 모수 추정 결과와 모델의 성능은 <표 12>에 요약되어 있다. 유지보수 프로젝트의  $1 \leq FP \leq 100$  범위에 있는 프로젝트의 투입 노력 비율에 대한 실측치와 추정치를 비교한 결과는 (그림 18)에 제시되어 있다.

4.5 B-T-I 단계 수행 프로젝트

B-T-I 개발단계 수행 프로젝트들은  $P_{PSBTI}$ 의 74건에 대한 B-T-I 단계에 투입된 개발 노력 데이터,  $P_{SBTI}$  13건에 대한 B-T-I 단계에 투입된 개발 노력 데이터와  $P_{BTI}$  29건을 합한 116건으로 <표 13>과 같이 단계별 노력 분배 비율이 계산되었다.

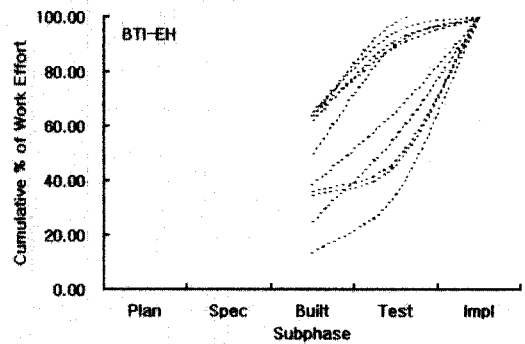
<표 13> B-T-I 단계 수행 프로젝트의 단계별 누적 노력 비율

개발 구분	FP규모	개발 단계별 누적 노력 비율 (%)				
		P	S	B	T	I
유지보수 프로젝트	1~100			63.87	89.21	100.00
	101~200			62.17	91.32	100.00
	201~300			50.00	89.84	100.00
	301~400			38.45	65.52	100.00
	401~500			65.23	94.61	100.00
	501~600			24.97	56.13	100.00
	701~800			13.62	35.09	100.00
	801~900			34.79	45.77	100.00
	1001~1200			64.02	98.14	100.00
	1601~1800			36.09	47.53	100.00
	1~100			66.58	84.17	100.00
	101~200			59.78	95.81	100.00
	201~300			59.85	86.45	100.00
신규개발 프로젝트	301~400			53.15	84.62	100.00
	401~500			49.93	82.06	100.00
	501~600			70.41	94.17	100.00
	601~700			57.76	65.37	100.00
	701~800			23.70	78.38	100.00
	801~900			86.81	96.93	100.00
	901~1000			29.02	66.85	100.00
	1001~1200			1.43	21.43	100.00
	1201~1400			61.16	86.28	100.00
	1401~1600			45.29	61.09	100.00
	1800~2000			53.74	64.33	100.00
	3001~4000			32.58	52.37	100.00
	4001~5000			50.76	86.62	100.00

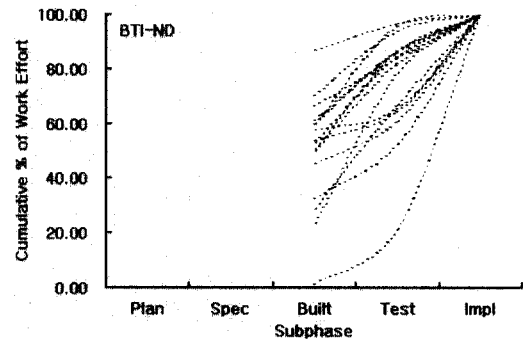
B-T-I를 수행한 유지보수 프로젝트의 규모별 투입 노력 비율에 대한 누적 분포는 (그림 19)에, 신규개발 프로젝트의 규모별 투입 노력 비율에 대한 누적 분포는 (그림 20)에 제시되어 있다. 개발단계별로 투입되는 노력의 비율은 S-B-T 단계 수행 프로젝트 모델과 같이 식 (5)의 모델로 지수형 분포를 표현 가능하다. 단지 차이점은  $t$ 에 대해  $P$ 와  $S$ 를 제외하고,  $B$ 는 3,  $T$ 는 4,  $I$ 는 5로 치환하면 된다. 모수 추정 결과와 모델의 성능은 <표 14>에 요약되어 있다.

<표 14> B-T-I 단계 수행 프로젝트의 모수 추정과 모델 성능

개발 구분	FP규모	모수 추정값		모델 성능		
		$\hat{b}$	$\hat{c}$	$R^2$	MMRE	MSE
유지보수 프로젝트	1~100	1.640	76.470	0.9886	1.11%	1.7219
	101~200	1.903	182.242	0.9968	0.60%	0.6383
	201~300	2.205	744.973	0.9987	0.50%	0.4985
	301~400	1.499	158.411	0.8984	8.11%	36.6100
	401~500	2.285	504.971	0.9992	0.29%	0.1204
	501~600	1.907	840.269	0.9229	11.72%	44.3941
	701~800	1.721	1157.705	0.8790	17.35%	139.9228
	801~900	1.645	484.544	0.6340	26.11%	162.2738
	1001~1200	2.506	1013.210	0.9936	1.18%	2.0984
	1601~1800	1.662	481.241	0.6283	25.27%	158.2731
	1~100	1.222	18.701	0.9194	2.85%	8.6733
	101~200	2.315	682.924	0.9976	1.15%	1.4417
	201~300	1.578	75.529	0.9819	1.60%	3.2124
신규개발 프로젝트	301~400	1.699	144.277	0.9857	1.73%	3.4611
	401~500	1.663	148.067	0.9805	2.22%	5.2681
	501~600	1.962	150.090	0.9979	0.38%	0.2478
	601~700	0.995	15.296	0.5398	11.41%	80.4243
	701~800	1.911	743.977	0.9960	11.46%	25.9385
	801~900	1.611	17.937	0.9951	0.25%	0.1259
	901~1000	1.823	622.796	0.9969	5.58%	18.2294
	1001~1200	2.007	5659.831	0.9209	152.37%	203.7736
	1201~1400	1.609	79.934	0.9708	2.07%	3.9353
	1401~1600	1.385	97.269	0.7069	13.59%	79.4649
	1801~2000	1.998	75.540	0.5465	13.92%	94.2033
	3001~4000	1.505	246.107	0.7946	16.26%	91.0916
	4001~5000	1.912	300.730	0.9944	1.11%	1.6664

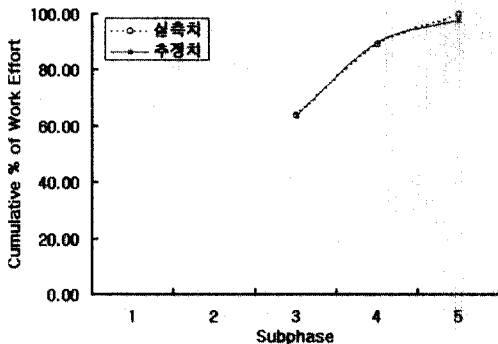


(그림 19) B-T-I 단계 수행 유지보수 프로젝트의 소프트웨어 규모별 누적 투입 노력



(그림 20) B-T-I 단계 수행 신규개발 프로젝트의 소프트웨어 규모별 누적 투입 노력

유지보수 프로젝트의  $1 \leq FP \leq 100$  범위에 있는 프로젝트의 투입노력 비율에 대한 실측치와 추정치를 비교한 결과는 (그림 21)에 제시되어 있다. 결론적으로, 결정계수  $R^2$ 가 작은 값이 MMRE가 큰 값으로 나타났으며, 프로젝트 개발 단계 수행 형태별로 몇 개의 FP 범위에서 Conte et al.[16]가 제안한 기준인  $MMRE \leq 25\%$  기준을 초과하는 성능을 보이고 있으나 대부분의 FP 범위별 모델들은 좋은 결과를 나타내었다. 따라서, 개발될 프로젝트의 규모인 기능점수 값을 계산한 후, 어떤 개발단계를 수행할 것 인가만 결정되면 개발 각 단계별로 투입될 노력의 비율을 제안된 모델을 적용하여 적용함으로써, 보다 이론적이고 실현 가능한 근거에 기반한 개발계획서 작성에 도움이 될 수 있을 것이다.



(그림 21) B-T-I 단계 수행 유지보수 프로젝트의 단계별 투입 누적노력 비율 추정 결과 ( $1 \leq FP \leq 100$ )

### 5. 결론 및 향후 연구과제

기존 소프트웨어 개발노력 추정 모델들은 소프트웨어의 코딩 단계 또는 개발 초기 단계에서 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 양에 따라 개발 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 분배할 수 없다. 따라서, 기존 제안된 모델을 이용해 프로젝트 개발에 소요되는 총 개발노력을 추정하더라도 실제로 소프트웨어를 개발하는 과정에서 각 단계별로 노력을 얼마나 분배하느냐에 따라 개발의 성공 여부가 결정될 수 있는 문제점이 있었다.

본 논문은 소프트웨어 개발과정에서 추정된 총 개발노력의 규모와 프로젝트의 개발형태인 신규개발과 유지보수 프로젝트에 따라 개발 각 단계별로 어느 정도의 비율로 분배

하는 것이 최적인가에 대해 실제 개발 실적 데이터들을 대상으로 모델을 제시하였다.

개발단계 수행 형태를 비교한 결과 계획(P), 요구명세(S), 구축(B), 시험(T)과 이행(I) 단계에 대해 프로젝트별로 수행된 단계들을 그룹화 하였다. 그룹화 결과 P-S-B-T-I, P-S-B-T, S-B-T-I, S-B-T와 B-T-I 단계를 수행한 프로젝트들로 구분이 되었으며, 155종의 프로젝트를 대상으로 이들 다른 프로젝트들 별로 개발단계별 분배된 노력을 제시하였다. 먼저, 1990년대에 개발된 155종의 프로젝트 데이터를 대상으로 프로젝트 개발 형태와 프로젝트 규모별로 개발 각 단계에서 투입된 누적 노력을 살펴보았다. 이어서, 개발 형태와 소프트웨어 규모별로 투입된 개발노력의 규모에 대한 중앙값을 취한 결과 S자 형태의 분포를 따름을 알 수 있었으며, 소프트웨어 규모별로 S자 형태의 모델을 제안하였다. 따라서, 제안된 모델들을 이용하여 소프트웨어 개발 각 단계별로 개발기간을 설정하는 기준으로 활용이 가능하다.

제안된 모델들은 대부분의 실측 데이터를 잘 추정하는 모델이었으며, 단지 몇 개의 데이터들에서만 성능이 저하됨을 알 수 있어 이 부분에 대한 적합한 모델 개발이 필요하다. 따라서, 추후 이 분야에 대한 연구를 수행할 것이다.

### 참 고 문 헌

- [1] K. H. Möller and D. J. Paulish, "Software Metrics-A Practitioners 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] S. Yamada, J. Hishitani, and S. Osaki, "Software-Reliability Growth with a Weibull Test-Effort : A Model & Application," IEEE Trans. on Reliability, Vol.42, No.1, pp.100-106, 1993.

[10] 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.

[11] P. V. Norden, "Project Life Cycle Modeling : Background and Application of the Life Cycle Curves," U. S. Army Computer System Command, 1977.

[12] K. Pillai and V. S. S. Nair, "A Model for Software Development Effort and Cost Estimation," IEEE Trans. on Software Eng., Vol.23, No.8, pp.485-497, 1997.

[13] 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>.

[14] S. S. Gokhale, T. Philip, P. N. Marinos, and K. S. Trivedi, "Non-Homogeneous Markov Software Reliability Model with Imperfect Repair," Technical Report TR-96/12, CACC Duke University, 1996.

[15] S. S. Gokhale, T. Philip, P. N. Marinos, and K. S. Trivedi, "Unification of Finite Failure Non-Homogeneous Poisson Process Models Through Test Coverage," Technical Report TR-96/36, CACC Duke University, 1996.

[16] S. Conte, H. E. Dunsmore and V. Y. Shen, "Software Engineering Metrics and Models," Benjamin/Cummings., 1986.

[17] S. S. Gokhale, T. Philip, P. N. Marinos, and K. S. Trivedi, "Non-Homogeneous Markov Software Reliability Model with Imperfect Repair," Technical Report TR-96/12, CACC Duke University, 1996.

[18] S. S. Gokhale, T. Philip, P. N. Marinos, and K. S. Trivedi, "Unification of Finite Failure Non-Homogeneous Poisson Process Models Through Test Coverage," Technical Report TR-96/36, CACC Duke University, 1996.



### 이상운

e-mail : sangun\_lee@hanmail.net

1983년~1987년 한국항공대학교 항공전자공학과(학사)

1995년~1997년 경상대학교 컴퓨터과학과(석사)

1998년~2001년 경상대학교 컴퓨터과학과(박사)

1992년~현재 국방품질관리소 항공전자장비 및 소프트웨어 품질보증 담당

관심분야 : 소프트웨어 공학(소프트웨어 시험 및 품질보증, 소프트웨어 신뢰성), 소프트웨어 메트릭스, 신경망, 뉴로퍼지



### 김용수

e-mail : yskim@kc.ac.kr

1985년~1993년 한국방송통신대학교 전자계산학과(이학사)

1993년~1995년 경남대학교 컴퓨터공학과(공학석사)

1998년~2001년 경남대학교 컴퓨터공학과(박사수료)

1997년~현재 거창전문대학 컴퓨터정보시스템과 조교수

관심분야 : 데이터베이스, 인공지능, 객체지향 시스템 설계, 소프트웨어 재공학



### 한판암

e-mail : pahan@zeus.kyungnam.ac.kr

1969년 동국대학교 졸업

1975년 동국대학교 경영대학원 졸업(경영학 석사)

1989년 명지대학교 대학원 졸업(공학 석사)

1992년 인천대학교 대학원 졸업(경영학 박사)

1980년~현재 경남대학교 공과대학 컴퓨터공학과 교수

1999년~현재 정보처리학회 자문위원

관심분야 : 소프트웨어 품질관리 및 신뢰성, 소프트웨어 개발 환경, 정보 공학