소프트웨어 개발인력 배분 모델

A Model for Software Development Manpower Allocation

박 석 규*
Seok-Gyu Park

요 약

노력 (인력)과 일정이 추정된 경우, 인력을 일정 시간대별로 효율적으로 배분하기는 매우 어렵다. 지금까지 Putnam의 Rayleigh 분포모델이 널리 적용되고 있다. 이 분포는 프로젝트 전반부에서 인력이 집중적으로 투입되는 형태를 표현하고 있다. 그러나 실제 수행된 프로젝트의 대부분이 중반부나 후반부시점에서 인력이 집중적으로 투입되는 현상을 나타내 Rayleigh 분포를 적용할 수 없다. 따라서 본 논문은 인력을 효율적으로 배분하기 위한 모델을 제안한다. 먼저, 소프트웨어 개발 분야에서 제시하고 있는 전형적인 형태와 실제 수행 프로젝트들의 인력 프로파일 형태를 고찰하였다. 다음으로, 효율적인 인력 배분을 위해 인력 프로파일에 대한 윤곽을 그려 모델을 유도하는 방법을 제안하였다. 제안된 모델은 기존의 Rayleigh와 Gamma 모델보다 좋은 성능을 나타내었다. 제안된 모델을 적용할 경우 소프트웨어 개발계획단계에서 개발기간에 대한 투입될 인력을 적절히 배분하여 프로젝트를 성공적으로 관리할 수 있을 것이다.

ABSTRACT

If the effort (manpower) and schedule are estimated, there is very difficult to allocate the manpower according to the schedule time units efficiently. Generally, the Putnam's Rayleigh Model has been applied popularly. This distribution model is expressing topology that manpower is consumed concentrically in first-half point. But, actual manpower of projects are consumed concentrically at middle or second half point. Therefore, this model cannot be applied in software project planning area. This paper suggests a model to distribute manpower efficiently. First of all, we investigate a typical type presenting in software development field and manpower profile type of actuality development projects. Next, we suggested a method to present the model by a drawing a contour about manpower profile for the efficient manpower distribution. The proposed model shows better performance than Rayleigh and Gamma model. By applying proposed model, we will properly distribute manpower to schedule in software development planning phase, and finally we may manage project successfully.

☞ Keyword : 인력 배분 (Manpower Allocation), 인력 프로파일 (Staffing Profile), 윤곽 (Contour), 인력배분 모델 (Manpower Distribution Model)

1. 서 론

소프트웨어 관리자에게는 할당된 인력 (비용)을 낭비 없이 적시에 효율적으로 활용하여 주어진 기간 내에 개발을 완료하는 능력이 필수적으로 요구된다. 왜냐하면 개발 과정에서 효율적인 인력 투입은 주어진 일정과 예산으로 개발을 성공할 수 있는 조건 중 하나가 될 수 있기 때문이다.

고객은 일반적으로 개발기간과 비용을 확정하여 개발 제안서 (RFP : Request For Proposal)를 제시한다. RFP에 기반하여 개발 관리자는 주어진 예산에 따른 총 소요 인력을 고객이 요구하는 개발기간 내에서 적절히 분배하여 프로젝트를 성공적으로 완료할 수 있도록 소프트웨어 개발계획을 작성한다. 이와 같이 작성된 개발인력 분포를 인력 프로파일 (Staffing Profile)이라하며, 개발 관리를 위한 Baseline이 된다.

프로젝트 관리를 위해서는 다음 사항을 고려해

* 종신회원: 강원도립대학 컴퓨터응용과 교수
skpark@gangwon.ac.kr
[2006/07/28 투고 - 2006/08/31 심사 - 2007/01/08 완료]

야만 한다. 첫째, 고객이 제시한 개발기간과 개발 비용이 현실적으로 타당한가? 이 질문에 대한 해답은 먼저 소프트웨어 규모를 추정하고 이에 기반하여 소요되는 총 인력의 양과 개발기간을 추정해야 한다. 이를 추정하는 기법은 COCOMO II[1]를 활용할 수 있다. 다음으로, 인력 낭비가 없도록 개발에 필요한 총 소요인력의 양을 개발 각 시간단위별로 어떻게 적절히 배분 (또는 할당)할 수 있는가? 이 질문에 대한 해답은 아직 제시되지 않고 있다고 할 수 있다. 왜냐하면 Standish CHOAS 보고서[2]에 따르면 365개 회사의 8,380개 소프트웨어 프로젝트를 대상으로 설문 조사한 결과 단지 16.2% 만이 고객이 요구하는 일정과 비용 내에서 개발을 성공하였다. 이러한 낮은 성공률의 주된 원인은 50%가 요구사항과 관련되어 있지만 관리자의 경험과 신뢰 있는 추정이 19%를 차지하고 있다.[3]

인력 프로파일은 시간단위 설정 방법에 따라 단위 시간 (일, 주간, 월)으로 설정하는 시간기반 (Time-based) 방법과 소프트웨어 생명주기의 단계 (Phase)로 설정하는 단계기반 (Phase-based)방법이 있다. 시간기반 인력 프로파일과 관련된 대표적인 연구로는 Putnam의 Rayleigh 분포 모델[4]이 있다. 이 모델의 단점을 보완하고자 와이블[5], Sech2[6]와 Gamma[7] 분포 모델이 제안되었다. 이 모델들은 하드웨어 개발 방법에 적합한 모델을 소프트웨어에 적용한 사례로 현실적으로 소프트웨어 개발 과정에서 투입되는 인력 프로파일을 적절히 표현하지 못하는 단점을 갖고 있다. 단계기반 인력 프로파일과 관련된 연구로는 Clark[8]의 COCOMO II 활용 방법, UP[9], Sophos[10]와 Catherine[11]의 폭포수 모델 기반 방법이 있다. 이 방법을 적용시 각 단계에 할당된 총 인력에 대해 다시 단위시간별로 어떤 분포로 인력을 할당할 것인가가 문제로 대두된다. 따라서 시간기반 모델의 문제점에 다시 직면한다. 그러므로 본 논문에서는 시간기반 인력 프로파일에 초점을 맞춘다.

본 논문에서는 프로젝트의 일정과 인력을 관리하여 프로젝트 성공률을 높이는데 활용하도록 소프트웨어 개발단계의 인력 프로파일을 적절히 표현하는 모델을 제시하고자 한다. 2장에서는 전형적인 인력 프로파일 형태에 관한 연구 결과를 고찰해 보고, 개발팀의 인원 배분과 인력 프로파일을 구하는 모델들의 문제점을 살펴본다. 3장에서는 전형적인 인력 프로파일로부터 현실적으로 발생할 수 있는 인력 프로파일을 제시하고, 이에 기반하여 인력 프로파일의 윤곽에 기반한 모델을 제시한다. 4장에서는 제안된 모델의 성능을 평가한다.

2. 관련연구

2.1 전형적인 인력 배분 사례

소프트웨어 개발 분야에서 전형적인 인력 프로파일의 윤곽 (Contour)에 관한 연구는 (그림 1)에 제시하였다. (그림 2)은 실제 수행된 프로젝트들의 시간기반 인력 프로파일을 나타내고 있다.

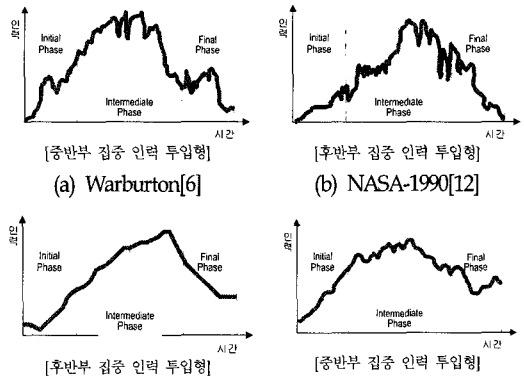
소프트웨어 개발팀의 인원을 어떻게 결정할 수 있는가? 먼저, 추정된 노력 (Person-Months)과 일정 (Months)에 기반하여 개발팀의 평균 소요 인원 = 노력/일정으로 얻는다. 이 인원은 프로젝트 생명주기 전체에 걸쳐 매월 평균적으로 투입될 인원으로 일양분포를 따른다. 실제 개발을 수행하면서 일양분포를 갖도록 개발팀의 인원을 구성하지는 않는다.

실제 투입되는 개발인력의 분포를 얻기 위해 편의상 개발기간을 3등분하여 전반부, 중반부와 후반부로 분류하여 보자. 각 등분에 인력이 집중 투입되는 현상을 "전반부 집중 인력 투입형", "중반부 집중 인력 투입 형"과 "후반부 집중 인력 투입형"으로 분류하자.

개발팀 투입 규모 (y축, 독립변수)는 개발기간 (x축, 종속변수)의 처음 1/3 영역 (전반부)에서는 투입 인원이 상승하며, 계획대로 개발이 진행될

관련근거	순차적 개발방법	반복 · 점진적 개발 방법
Putnam[4]	[전반부 집중 인력 투입형]	-
UP[9]	-	[후반부 집중 인력 투입형]
NASA SEL-1990 [12]	[중반부 집중 인력 투입형]	-
Marasco [13]	[중반부 집중 인력 투입형]	[후반부 집중 인력 투입형]
Mejias [14]	[중반부 집중 인력 투입형]	-
DSW Group [15]	[중반부 집중 인력 투입형]	[중반부 집중 인력 투입형]
NASA SEL-1993 [16]	[중반부 집중 인력 투입형]	-
Schwalbe [17] Stuckenbruck [18] Kapur [19]	[후반부 집중 인력 투입형]	-

(그림 1) 전형적인 시간기반 인력 프로파일 윤곽



[중반부 집중 인력 투입형]
(a) Warburton[6]

[후반부 집중 인력 투입형]
(b) NASA-1990[12]

[후반부 집중 인력 투입형]
(c) NASA-1995[20]

[중반부 집중 인력 투입형]
(d) Mockus et al.[21]

(그림 2) 실제 수행 프로젝트들의 시간기반 인력 프로파일

경우 마지막 1/3 영역 (후반부)에서는 하강하는 인력 분포를 나타내는 "중반부 집중인력 투입형"이 이상적인 형태이다. 왜냐하면 개발 초기 단계(개념 정의와 요구사항 분석 단계)에서는 필수 요원만으로 개념을 정의하며, 중반 단계 (설계와 코딩)에서는 많은 인원이 요구되며, 마지막의 시험단계에서는 점차 인원이 감소한다.

일양분포로 개발인력을 투입할 경우, 전반부와 후반부에서는 인력 낭비를, 중반부에서는 인력 부족 현상을 겪기 때문에 효율적인 인력 관리를 할 수 없게 된다. "전반부 집중 인력 투입형"의 경우, 전반부에서는 과다한 인력 낭비를, 중반부와 후반부에서는 과다한 인력 부족 현상을 나타낸다. "후반부 집중 인력 투입형"의 경우는 토목건축분야에서 일반적으로 적용하는 사례이며, 소프트웨어 개발 분야의 계획단계에서는 가급적 적용하지 않는 것이 좋다. 왜냐하면 "중반부 집중 인력 투입형"으로 개발을 시작하더라도 실제로 일정지연으로 인해 대부분은 "후반부 집중 인력 투입형"으로 변경되기 때문이다.

결론적으로 소프트웨어 개발계획 단계에서는 "중반부 집중 인력 투입형"을 적용하는 것이 타당해 보인다. 따라서 본 논문에서는 이 유형에 대한 모델을 어떻게 도출할 것인가를 제안한다.

2.2 인력 배분 프로파일 모델

개발 각 단계에 할당된 인력의 양을 어떻게 시간 단위로 배분할 수 있는가? 또한, 인력 프로파

일을 계획하고자 할 때, 어떤 모델들을 적용할 수 있는가? 지금까지 "중반부와 후반부 집중 인력 투입형"에 대해 제안된 모델은 없는 실정이다. 단지 "전반부 집중 인력 투입형"에 대해, Putnam의 Rayleigh 분포 모델[4]과 Phillai et al.의 Gamma 분포 모델[7]이 있다. Putnam의 Rayleigh 모델[4]은 하드웨어에 기반을 둔 인력프로파일로 설계단계부터 운영 후 폐기될 때까지를 고려하고 있어 초기 단계인 설계와 코딩, 시험과 평가가 수행되는 과정에서 인력이 계속 증가하다가 개발주기가 종료되는 시점에서 개발노력이 최대가 되고 운영단계에서 인력이 점차 감소하는 Rayleigh 분포를 따르고 있다.

Putnam의 Rayleigh 인력분포 모델[4]은 임의의 시간 t까지 프로젝트에 투입된 누적 인력 Y로, 임의의 시간 t 시점에서 프로젝트에 투입되는 인력 y는 식 (1)로 표현된다. 여기서 MY는 년인원(Man-Year)이다.

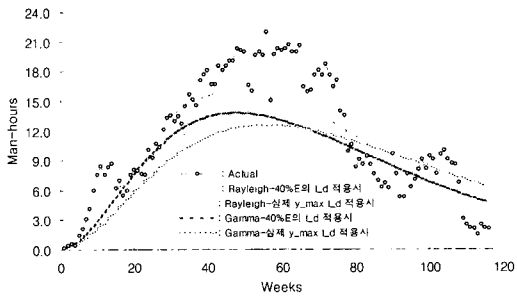
$$Y = E(1 - e^{-at^2}) \text{ MY}, \quad y = 2Eate^{-at^2} \text{ MY/Y} \qquad (1)$$

여기서 E는 소프트웨어 생명주기 전체에 걸쳐 투입된 총 인력, $a = \frac{1}{2t_d^2}$, t_d는 y가 최대가 되는 개발 완료시점 (총 인력의 약 40% 소요)이다.

Warburton[6]은 Putnam의 Rayleigh 모델[4]이 인력이 최고점에 도달한 이후에야 모수 t_d와 E 값을 추정할 수 있음을 제기하였다. 이는 개발이 종료되고 유지보수 단계에 도달하였을 때 비로소 투입인력을 추정할 수 있으며, 개발단계에서는 불가능함을 의미한다. Phillai et al[7]은 개발노력 분포가 Rayleigh 분포를 따르는데 따른 문제점을 해결하기 위해 $y = e^{\nu}t^2e^{-\mu t} = \frac{1}{\Gamma(\alpha)\beta^{\alpha}} t^{\alpha-1}e^{-\frac{t}{\beta}}$, $\alpha = 3, \beta = \frac{t_d}{2}$로 치환하여 식 (2)의 Gamma 모델을 제시하였다.

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

개발단계에서 모수 추정이 불가한 Putnam의 Rayleigh 모델[4]의 단점을 보완한 Gamma 모델[7] 조차도 t_d를 모수로 사용하고 있다. (그림 3)은 Warburton[6]의 폭포수 모델을 적용한 프로젝트 수행 데이터에 대해 식 (1)의 Rayleigh 분포와 식 (2)의 Gamma 분포 모델을 적용한 결과이다. 여기서 t_d는 40%의 인력이 투입된 시점과 실제 인력이 최대가 되는 y_{max} 시점을 기준으로 설정하였다. 두 가지 t_d를 모두 적용한 결과 실제의 인력 프로파일을 적절히 표현하지 못하고 있다.



(그림 3) 시간기반 인력프로파일 모델 적용 결과

일반적으로 협의의 소프트웨어 생명주기인 개발단계에 대한 인력 분포만을 고려해 볼 때 Rayleigh 분포를 따르지 않는다. 또한, Putnam의 Rayleigh 분포 모델[4]은 소프트웨어 개념정의와 요구사항 분석단계를 고려하지 않고 있어 개발주기에 대한 정확한 인력프로파일을 제시하지 못하고 있다.[22] 따라서 Rayleigh나 Gamma 분포를 적용할 경우 실제 투입 인력과 계획과의 불일치가 발생하여 프로젝트 관리에 어려움이 발생한다.

소프트웨어 개발여부를 결정하는 계획단계에서 소프트웨어 개발에 투입되는 인력과 비용이 얼마나 되는지를 추정하는 경우, 일반적으로 개발주기만을 고려하며, 운영단계의 유지보수 주기까지 포함하는 확장된 소프트웨어 생명주기 전체를 고려하지 않는다. 소프트웨어 개발여부를 결정하는 계획단계에서 소프트웨어 개발에 투입되는 인력과 비용이 얼마나 되는지를 추정하는 경우 개발주기

에만 한정시켜야 할 필요성이 제기된다. 개발주기에 한정되어 추정된 비용을 개발 각 시점별로 어떻게 적절히 할당하여 프로젝트를 성공적으로 수행할 수 있느냐가 개발 인력 프로파일이며, 본 논문에서 연구하고자하는 프로젝트 계획분야이다.

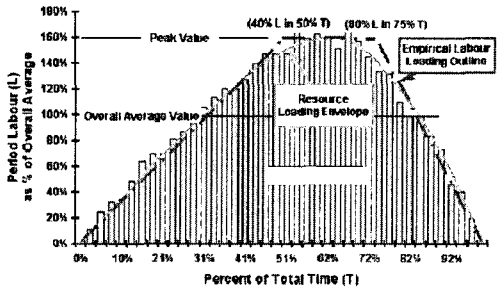
폭포수 프로세스에 대한 확장된 생명주기 전체에 소요되는 인력분포인 Putnam의 Rayleigh 분포 모델[4]과 Phillai et al.의 Gamma 분포 모델[7]들은 소프트웨어의 개발주기만을 대상으로 하지 않고 개발과 유지보수 주기를 모두 고려한 확장된 소프트웨어 생명주기 전반에 걸쳐 투입되는 인력분포를 대상으로 하였다. 대표적인 소프트웨어 개발 프로세스로 순차적 개발 방법인 폭포수 프로세스와 반복적 개발방법인 UP는 일반적으로 개발주기만으로 범위를 한정시키고 있다.

3. 윤곽설정 방식 이용 중반부 집중인력 투입형 프로파일 모델 유도

"중반부 집중 인력 투입형"에 대해 일반적으로 적용 가능한 통계적 모델을 찾기는 어려운 실정이다. 왜냐하면, (그림 2)와 같이 실제로 수행된 프로젝트 데이터 수집 개수가 작고, 다양한 개발 환경을 반영한 데이터들이 아니기 때문에 이들 데이터에만 적합한 통계적 모델이 일반적으로 적용할 수 있는 모델이라고 제시할 수 없기 때문이다.

따라서 본 논문에서는 개략적인 인력 프로파일 윤곽을 그리는 방법을 적용하고자 한다. 개략적 인력 프로파일 윤곽을 그리는 방법으로 소프트웨어공학 분야에서는 NASA SEL[20]과 Allen[23]의 방법이 있으며, 토목건축공학 분야에서는 Wideman [24]이 (그림 4)의 분포를 제시하였다.

Allen[23]은 비대칭 돔 형태인 이 인력 곡선을 표현하는 식이 복잡하여 계획 작성 목적으로는 특히 유용하지 않음을 제기하였다. 따라서 직관에 의한 경험법칙(Rules of Thumb)으로 (그림 4)에 대해 다음 가정에 의해 단순한 선형 형태를 취하는 Baseline 곡선을 제시하였다. 여기서 L을 인력, T를 시간으로 가정한다.

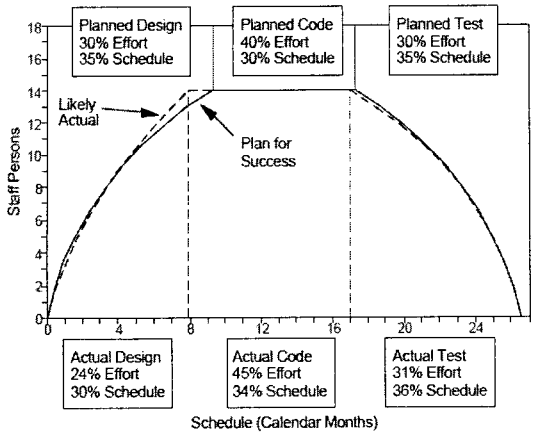


(그림 4) 토목건축 분야의 투입인력 프로파일 윤곽

(1) 최대 인력은 평균 인력의 160% 소요: 평균 투입 인력의 양을 100%로 했을 때, 최대 투입인력의 양은 평균 투입인력의 1.6배가 소요된다.
(2) 최대 인력은 총 인력의 40%를 소비한 시점에서며, 프로젝트가 50% 경과한 후 처음으로 발생 (40% L in 50% T): 최대로 투입되는 인력은 프로젝트에 투입되는 총 인력이 40% 소요된 시점에서 발생하며, 이때 프로젝트 수행 기간은 50% 진행된 시점이다.
(3) 최대 인력 지속 기간은 총 요구 인력의 40%를 차지 (80% L - 40% L) 하며, 프로젝트 일정의 25% 동안 발생 (75% T - 50% T): 최대로 투입되는 인력은 어느 한 시점에서 만 발생하지 않고 총 투입 인력의 40%를 프로젝트 개발 일정의 25% 기간 동안 지속적으로 투입된다.

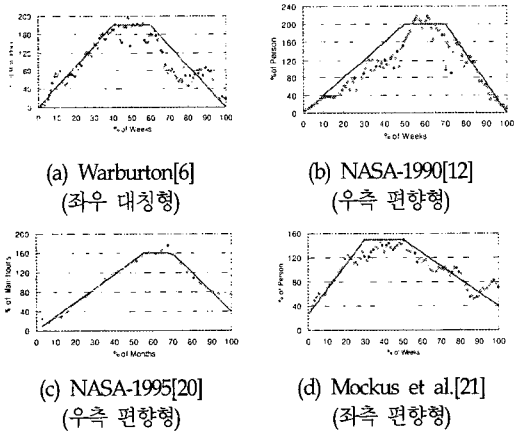
(그림 5)는 NASA SEL[16]이 제시한 인력 프로파일 윤곽으로, 초기에 추정된 소프트웨어 규모에 기반한 개발일정과 인력 비율에 규모 성장 비율을 반영하여 최종적으로 결정된 Baseline 모델이다. 그러나 이 방법은 소프트웨어 개념 정의와 요구사항 분석 단계를 고려하지 않아 본 논문에서는 다루지 않는다.

윤곽설정 방법을 적용하여 "중반부 집중 인력 투입형"의 모델을 제시하기 위해 본 논문에서는 Allen[23]의 방법을 적용한다. (그림 2)의 데이터에 대해 윤곽을 어떻게 정확히 그릴 것인가에 대해서는 이견이 있을 수 있으며, 법칙이 존재하지 않는다.



(그림 5) 소프트웨어 개발 인력 프로파일 윤곽

따라서 본 논문에서는 (그림 2)의 실제 데이터들을 적절히 표현할 수 있도록 개략적으로 윤곽을 그렸으며, 최종적인 결과는 (그림 6)에 제시되어 있다.



(a) Warburton[6] (좌우 대칭형)

(b) NASA-1990[12] (우측 편향형)

(c) NASA-1995[20] (우측 편향형)

(d) Mockus et al.[21] (좌측 편향형)

(그림 6) 실제 투입 인력의 프로파일의 윤곽

(그림 6)의 인력 프로파일 윤곽은 다음과 같이 유도되었다. 먼저, 개략적인 윤곽을 그린 결과는 (표 1)과 같다. (표 1)의 4개 데이터에서 L과 T가 모두 일정한 값을 갖지 못하고 있어 일반적인 법칙을 유도할 수 없다. 이를 해결하기 위해 (표 1)의 데이터에서 최대 인력 발생 시점 (T)은 그대로 유지하면서 최대 인력 발생시점까지 투입된 인력의 양 (L)에 대한 표준을 유도하기 위해 조절한 결과는 (표 2)와 같다. 이 결과 3개 데이터에서 최대 인력 지속은 40%L에서 70%L 또는 75%L로 유사한 결과를 얻을 수 있었다.

(표 1) 개략적 윤곽의 인력분포

구분	최대인력 발생시점	최대인력 감소시점	최대인력 지속기간	평균인력 대비 최대인력
Warburton [6]	40%T -37%L	60%T -70%L	20%T	180%
NASA-1990 [12]	50%T -37%L	70%T -75%L	20%T	220%
NASA-1995 [20]	50%T -40%L	70%T -72%L	20%T	160%
Mockus et al. [21]	30%T -27%L	50%T -55%L	20%T	150%

(표 2) 개략적 윤곽 조절 결과

구분	최대인력 발생시점	최대인력 감소시점	최대인력 지속기간	평균인력 대비 최대인력
Warburton [6]	40%T -40%L	60%T -70%L	20%T	180%
NASA-1990 [12]	50%T -40%L	70%T -75%L	20%T	200%
NASA-1995 [20]	50%T -40%L	70%T -75%L	20%T	160%
Mockus et al. [21]	30%T -30%L	50%T -55%L	20%T	150%

(표 2)의 기준에 따라 각 데이터의 인력 (L)과 시간 (T)를 계산한 결과는 (표 3)과 같다. 고객이 요구하는 개발기간과 비용 데이터에 기반하여 (표 3)의 자료를 활용하면 적절한 인력을 할당할 수 있을 것이다.

(표 3) 시간대별 인력 투입 비율

구분	개략적 윤곽	
	시간 비율(T)	인력 비율(L)
Warburton[6] (좌우 대칭형)	1% ~ 40% 40% ~ 60% 60% ~ 100%	$4.5T$ 180% $-4.5T+450$
NASA-1990[12] (우측 편향형)	1% ~ 50% 50% ~ 70% 70% ~ 100%	$4T$ 200% $-6.67T+666.7$
NASA-1995[20] (우측 편향형)	1% ~ 50% 50% ~ 70% 70% ~ 100%	$2.9T$ 160% $-4T+440$
Mockus et al.[21] (좌측 편향형)	1% ~ 30% 30% ~ 50% 50% ~ 100%	$4.14T+26$ 150% $-2.2T+260$

제안된 윤곽기반 소프트웨어 개발인력 배분 모델에 의하면 인력을 배분하는 법칙은 다음과 같다.

(1) 최대 인력은 평균 인력의 150%, 160%, 180%과 200% 소요: 평균 투입 인력의 양을 100%로 했을 때, 최대 투입인력의 양은 평균 투입인력의 1.5배에서 2배까지 소요된다.
(2) 최대 인력은 총 인력의 30%, 40%와 50%를 소비한 시점에서며, 프로젝트가 50% 경과한 후 처음으로 발생 한다.
(3) 최대 인력 지속 기간은 프로젝트 일정의 20% 동안 발생 한다.

4. 모델 분석 및 평가

제안된 모델의 성능은 Putnam의 Rayleigh 모델[4]과 Phillai et al.의 Gamma 모델[7]과 비교 분석하였다. 모델 비교를 위해 얻은 기준 데이터는 (표 4)와 같다.

(표 4) 모델 비교를 위한 기준 데이터 분석

구분	40% 인력 소요	최대인력 발생		총 인력		평균 인력	시간 단위 수
		시점	인력	실제	개략 윤곽		
Warburton [6]	42.24%T	48.28%T	50.35%	1,300.8	1,404.8	11.21	116
NASA-1990 [12]	52.63%T	56.14%T	48.56%	1,318.4	1,581.6	11.56	114
NASA-1995 [20]	51.61%T	67.74%T	67.04%	194.2	191.4	6.26	31
Mockus et al. [21]	41.38%T	50.57%T	54.49%	2,357.5	2,452.1	27.10	87

모델의 성능을 평가하는 기준으로 모델의 정확도를 판단하기 위해 MMRE (Mean Magnitude Relative Error)를 분석하였다. MMRE는 다음과 같이 측정된다. 상대오차(RE, Relative Error) = (추정치 - 실측치)/실측치, MRE(Magnitude of RE) = $|RE|$, MMRE(Mean MRE) = $100/n * \sum_{i=1}^{n} MRE$. MMRE가 작은 값이면 평균적으로 좋은 모델임을 알 수 있다.

제안된 "중반부 집중 인력 투입형" 모델의 적합성을 평가하기 위해, "전반부 집중 인력 투입형" 모델인 Rayleigh 분포[4]와 Gamma 분포 모델[7]과 비교 분석하여 본다.

Rayleigh 분포[4]와 Gamma 분포 모델[7]은 t_d를 40%E가 발생한 시점으로 결정하고 있다. 그러나 본 논문에서는 실제 y_{max}가 발생한 시점인 t_d도 추가로 적용하여 Rayleigh와 Gamma 모델의 적합성을 검증하였다. 모델별 MMRE를 비교한 결과는 (표 5)와 같다.

(표 5) 모델 성능 분석

구분	40%E t_d	y_{max} t_d	MMRE				
			Rayleigh 분포 모델 [4]		Gamma 분포 모델 [7]		제안 모델 (개략적 윤곽)
			40%L t_d 적용	y_{max} t_d 적용	40%L t_d 적용	y_{max} t_d 적용	
Warburton [6]	49	56	35.1	40.3	33.5	44.9	**30.1**
NASA-1990 [12]	60	64	42.0	44.7	49.1	54.1	**28.4**
NASA-1995 [20]	16	21	23.7	28.7	23.0	43.1	**9.8**
Mockus et al. [21]	36	44	15.8	22.3	19.6	30.0	**12.5**

제안된 개략적 윤곽 모델이 40%L 시점의 t_d를 적용한 Rayleigh와 Gamma 분포 모델 보다 성능이 좋음을 알 수 있다. 더불어 실제 y_{max}가 발생한 시점으로 t_d를 적용한 경우의 Rayleigh와 Gamma 분포 모델 보다 성능이 우수함을 알 수

있다. 즉, 최대 인력이 발생하는 시점을 기존 모델이 제안한 시점이나 실제 발생한 시점을 모두 적용하더라도 제안된 모델의 성능보다 좋지 못한 결과를 나타내고 있다. 결론적으로 소프트웨어 개발인력을 효율적으로 배분하기 위해서는 제안된 모델을 적용하는 것이 보다 타당할 것이다.

5. 결론 및 향후 연구과제

본 논문에서는 소프트웨어 개발 총 노력과 일정이 추정되었을 경우 인력 낭비와 부족 현상을 방지하여 적절한 인력 분배를 할 수 있는 방법을 제안하였다. 먼저, 소프트웨어 개발 분야에서 전형적으로 어떤 인력 프로파일 형태를 나타내는지와 실제 수행 프로젝트들의 인력 프로파일 형태를 고찰하였다. 이들로부터 개발단계에서 발생할 수 있는 인력 프로파일인 중반부와 후반부에 인력이 집중 투입되는 프로파일을 유도하였다. 기존의 인력 프로파일 유도 모델들은 개발단계에만 한정되지 않는 문제점을 갖고 있어 개발단계의 실제 수행 인력 프로파일을 적절히 표현하지 못하는 단점을 갖고 있다. 따라서 인력 프로파일을 설명할 수 있는 모델을 유도하기 위해 인력 프로파일의 윤곽을 그려 모델을 유도하는 방법을 제안하였다.

본 논문에서는 "중반부 집중 인력 투입형"에 대한 통계적 모델을 제시하지는 못하였다. 제안된 모델의 일반화와 더불어 보다 이론적인 모델로서의 발전을 위해서는 좌우대칭형, 좌측 편향형과 우측 편향형의 인력 프로파일을 모두 표현 가능한 통계적 모델 개발이 필요하다. 따라서 추후 이 분야에 대한 연구를 수행할 것이다.

참고문헌

[1] Boehm, B. W. et al, "Software Cost Estimation with COCOMO II," Prantice-Hall, 2000.

[2] Standish group, "The CHAOS Report," http://www.standishgroup.com/sample/PDFpages/chaos1994.pdf, 1994.

[3] Gurd, A., "Dynamic Requirements Management for Iterative/ Incremental Development," Telelogic, 2005.

[4] Putnam, L. H., "A General Empirical Solution to the Macro Software Sizing and Estimating Problem," IEEE Trans. on Software Eng., Vol. SE-4, No. 4, pp. 345-361, 1978.

[5] Knafl, G. J. and Gonzales, C., "An Evaluation of Software Cost Models," http://facweb.cs.depaul.edu/research/techreports/TR98-005.doc, DePaul University, 1998.

[6] Warburton, R. D. H., "Managing and Predicting the Costs of Real-Time Software," IEEE Trans. on Software Eng., Vol. SE-9, No. 5, pp. 562-569, 1983.

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

[8] Clark, B., "COCOMO II Model," CSE Research Associate 15th COCOMO/SCM Forum, 1998.

[9] Kruchten, P., "Planning an Iterative Project," The Rational Edge e-zine for the Rational Community, http://www.therationaledge.com/content/oct_02/f_iterativePlanning_pk.jsp, 2002.

[10] Sophos, "Observations from a Tech Architect: Enterprise Implementation Issues & Solutions," ITtoolbox, Blogs, 2005.

[11] Catherine, M. B., "ISM 5316 - Project Management," Computer Information Systems, College of Business, Florida Gulf Coast University, 2002.

[12] NASA, "Manager's Handbook for Software Development, Revision 1, NASA Software Engineering Lab. Series, SEL-84-101, 1990.

[13] Marasco, J., "Iterative Development," The Rational Edge e-zine for the Rational Community, http://www.therationaledge.com/content/oct_02/k_iterativeDevelopment_jm.jsp, 2002.

[14] Mejias, E., "Project Management Overview," http://www.enterprise-works.com/ProjMgmt.html, Enterprise-Works.com, 2001.

[15] DSW Group, "The Software Estimation Story," http://www.dswgroup.com/html/reference/SoftwareEst/html, 2003.

[16] NASA, "Cost and Schedule Estimation Study Report," NASA Software Engineering Lab. Series, SEL-93-002, 1993.

[17] Schwalbe, K., "Information Technology Project Management," 2nd Edition, Cambridge, MA, 2002.

[18] Stuckenbruck, L. C., "The Implementation of Project Management," Project Management Institute, PA, Wiley, 1981.

[19] Kapur, G. K., "The Seven Deadly Sins of Project Management," Center of Project Management, CA, 1995.

[20] NASA, "Software Management Guidebook, Revision 1," NASA Software Engineering Lab. Series, SEL-94-102, 1995.

[21] Mockus, A., Weiss, D. M. and Zhang, P., "Understanding and Predicting Effort in Software Projects," ICSE, 2003.

[22] Briand, L. C. and Wieczorek, I., "Resource Estimation in Software Engineering," 2nd Edition of the Encyclopedia of Software Engineering, Wiley, 2001.

[23] Allen, W., "Developing the Project Plans," Engineering Institute of Canada Annual Congress Wokshop, Toronto, pp. 3-9, 1979.

[24] Wideman, R. M., "Appling Resource Loading, Production & Learning Curves to Construction: A Pragmatic Approach," http://www.maxwideman.com/papers/resource/, 2001.

◐ 저 자 소 개 ◑

박 석 규(Seok-Gyu Park)
1992년 경남대학교 대학원 컴퓨터공학과 졸업(공학석사)
2005년 경상대학교 대학원 컴퓨터과학과 졸업(공학박사)
2001년~ 강원도립대학 컴퓨터응용과 교수
관심분야 : 소프트웨어 신뢰성, 시스템설계, 멀티미디어
E-mail : skpark@gangwon.ac.kr