

소프트웨어 개발 라이프사이클 인력 프로파일

이 상 문

요 약

성공적인 프로젝트 계획은 프로젝트를 개발하는데 요구되는 인력을 얼마나 정확히 추정하느냐에 달려있다. 이 정보들은 Putnam의 Rayleigh 모델이나 Phillai et al.의 Gamma 모델들로부터 유도된다. 그러나 이들 모델들은 투입 인력이 지수적으로 계속적으로 증가하여 개발이 종료되는 시점에서 투입되는 인력이 절정에 도달하는 소프트웨어 생명주기 전체에 대한 인력분포를 대상으로 하고 있다. 그러나 실제 프로젝트에서는 대부분의 인력이 개발에 투입되고 단지 적은 규모의 인력만이 유지보수에 할당된다. 또한, 대표적인 개발 프로세스인 폭포수 모델이나 단일화된 프로세스도 개발단계만을 대상으로 하고 있다. 이러한 개발환경에는 개발완료시점에서 절정에 도달하는 기존 인력분포 모델의 투입인력 분포 곡선을 적용할 수 없다. 본 논문은 이러한 문제점을 해결하기 위해 개발단계에 적합한 인력분포 모델을 제시한다. 먼저, Putnam이 제시한 인력분포로부터 폭포수 모델의 개발단계 투입 인력 분포를 유도하고, 다음으로, 단일화된 프로세스에 대한 개발단계 투입인력 분포를 유도하였다. 두 프로세스 투입인력을 비교한 결과 개발 각 시점에 대한 투입인력의 양, 수행 업무 단계는 차이가 발생하지만 총 투입 인력분포는 유사한 형태를 나타내었다. 이 결과로부터 두 개발 프로세스 모두에 적용할 수 있는 단일화된 개발인력 분포 모델을 유도하였다.

A Software Manpower Profile for Software Development Life Cycle

Sang-Un Lee[†]

ABSTRACT

Successful project planning relies on a good estimation of the manpower required to complete a project. The good estimation can be derived from Putnam's Rayleigh Model or Phillai et al.'s Gamma Model. These models only can be applied for the projects which the need of manpower is increased exponentially and the highest of manpower is required at the end of development phase. However, in a practical project, most manpower is required during development phase and a small amount of manpower is assigned during maintenance phase. In addition, the Waterfall Model and Unified Process only can be adopted for development phase. So the current development environments cannot be adopted into the existing manpower distribution models which the highest manpower is required at the end of development phase. This paper suggests an appropriate model for development phase to solve this problem. First, the appropriate manpower distribution for development phase of the Waterfall model was derived from Putnam's manpower distribution and then manpower distribution of development phase was derived for Unified Process. After comparing the required manpower of two processes, total manpower distribution is similar each other even though the required manpower and task is different for each point of development phase. From this result, a unified model is derived and it can be applied for both development processes.

키워드 : 소프트웨어 생명주기(Software Life Cycle), 폭포수 모델(Waterfall Model), 단일화된 프로세스(Unified Process), 생명주기 인력(Life Cycle Manpower), Rayleigh 모델(Rayleigh Model), Gamma 모델(Gamma Model)

1. 서 론

소프트웨어 개발계획 작성 단계에서 프로젝트 개발에 소요되는 인력, 비용과 개발기간 등의 신뢰할 만한 수준의 정보 제공은 사업의 성공 여부를 결정하는데 필수적이다[1, 2]. 이들 정보들은 소프트웨어의 유지보수 단계(Maintenance Phase)가 아닌 개발 단계(Development Phase)와 관련되어 있다. 소프트웨어 생명주기(Life Cycle)는 개발 시작 시점부터 폐기까지를 의미하며, 개발 주기(Development Cycle)와 유지보수 주기(Maintenance Cycle)로 구성된다.

KRM[3]은 프로젝트의 생명주기를 일반적으로 프로젝트에 대한 구상과 종결 사이에 있는 모든 단계들로 개발단계만으로 한정하여 정의하고 있다. 운영단계와 프로젝트 납품물에 대한 제안 단계를 포함하는 경우 이를 확장된 생명주기 또는 소프트웨어 생명주기로 구분하였다.

소프트웨어 생명주기 동안 투입되는 인력이 어떤 분포를 따르는지에 대한 대표적인 연구로는 Putnam[4]의 SLIM(Software Life cycle Management)으로 불리는 Rayleigh 분포 모델과 Phillai et al.[5]의 Gamma 분포 모델이 있다. 이들 모델들의 특징은 소프트웨어의 개발주기와 운영주기를 합친 소프트웨어 생명주기 전반에 걸친 인력분포 프로파일을 거론하고 있다.

[†] 정 회 원 : 국립 원주대학 여성교양과 교수
논문접수 : 2004년 3월 12일, 심사완료 : 2004년 6월 28일

대표적인 소프트웨어 개발 프로세스(Development Process)로는 순차적(Sequential) 방법과 반복적(Iterative) 방법이 있다[6]. 순차적 개발 방법에는 폭포수 모델이, 반복적 개발 방법에는 UP(Unified Process)가 가장 널리 적용되고 있다[7]. 이들 개발 프로세스는 소프트웨어 생명주기 중에서 유지보수 주기를 제외시키고 개발주기만으로 범위를 한정시키고 있다[8].

소프트웨어 개발 단계별로 소요되는 인력의 규모(인력 프로파일)를 판단하기 위해서는 소프트웨어 개발주기에 한정된 인력분포 추정 모델을 적용하여야 한다. 그러나 지금까지 개발인력 분포만을 추정할 수 있는 모델이 없는 실정으로 생명주기 전체에 투입되는 인력분포 모델인 Putnam[4]의 Rayleigh 분포와 Phillai et al.[5]의 Gamma 분포 모델을 적용해야 한다.

본 논문은 생명주기 전체에 대한 인력 분포 모델이 개발주기에만 한정된 상황에서도 적합한지 여부를 검토하고 문제점을 도출한다. 또한 개발주기에 한정하여 다양한 개발 프로세스에도 적합한 인력 추정 모델을 제시한다.

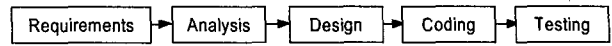
2장에서는 소프트웨어 개발 프로세스, 생명주기 인력분포 모델들을 고찰해보고 기존 제안된 모델들을 실제 개발 프로세스에 적용시 문제점을 살펴본다. 3장에서는 적용될 개발 프로세스별로 적합한 개발인력 분포를 유도하여 본다. 4장에서는 3장에서 유도된 개발인력 분포를 표현할 수 있는 모델을 제시하고 실제 개발인력 데이터에 적용하여 모델의 적합성을 평가한다.

2. 소프트웨어 인력분포

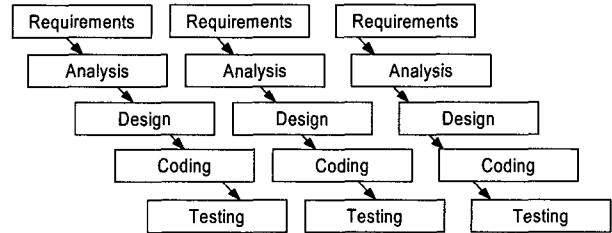
2.1 대표적인 소프트웨어 개발 프로세스

소프트웨어 생명주기는 개발 주기와 유지보수 주기로 구성되어 있으며, 소프트웨어 개발이 완료되고 배치된 후 소프트웨어 시스템은 마지막 퇴역 때까지 오랜 기간 지속되는 유지보수, 관리와 진화 단계로 접어들다[9].

개발과정과 연관된 활동과 산출물을 가지는 특정 단계로 일반화시킨 것이 소프트웨어 개발 프로세스이다. 고전적 생명주기 프로세스로 선형(Linear), 순차적인 폭포수(Waterfall) 모델이 있다. 이에 비해, 반복적 방법은 나선형(Spiral), 점진형(Incremental), 진화형(Evolutionary) 등 많은 변형된 형태들이 있다[6]. (그림 1)에 순차적인 폭포수 모델과 반복적 개발방법의 예를 제시하였다[10]. RUP는 Rational사에서 제안한 UP로, 나선형 모델을 상용화한 대표적인 개발 프로세스이다[11]. 최근에는 기존의 선형, 순차적 개발형태인 폭포수 프로세스를 적용할 경우 사용자의 요구사항을 만족시키는 소프트웨어 개발 어려움으로 점진적, 반복적인 프로세스로 전환되고 있는 실정이며, 객체지향 소프트웨어 개발에 적합하도록 설계된 UP를 채택하고 있다[7].



(a) Sequential Development Life cycle

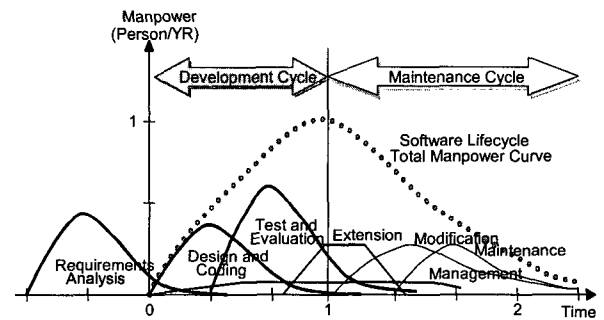


(b) Iterative Development Life cycle

(그림 1) 대표적인 소프트웨어 개발 방법

2.2 소프트웨어 생명주기 인력분포 모델과 문제점

소프트웨어 생명주기 전반에 걸쳐 투입되는 인력에 관한 연구로 Putnam[4], Phillai et al[5], Boehm[12], Norman[13,14]과 Warburton[15] 등이 있다. Putnam[4]는 전형적인 소프트웨어 생명주기 전반(계획, 요구명세 분석, 설계, 코딩, 시험과 유지보수 중 계획과 요구사항 분석단계 제외)에 걸쳐 소요되는 총 인력뿐만 아니라 생명주기의 각 단계(즉, 설계, 코딩, 시험 및 유지보수)에 대한 인력도 (그림 2)와 같이 Rayleigh 분포를 따름을 관찰하였다.



(그림 2) 소프트웨어 인력 분포

임의의 시간 t 시점까지 프로젝트에 투입된 누적 인력 y 를 식 (1)로, 임의의 시간 t 시점에서 프로젝트에 투입되는 인력 \dot{y} 는 식 (1)을 미분한 식 (2)로 표현된다.

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

$$\dot{y} = 2Eate^{-at} \text{ MY/YR} \tag{2}$$

여기서 E 는 Rayleigh 곡선 아래에 있는 영역으로 년 인원으로 표시되며, 소프트웨어 생명주기 전반에 걸쳐 투입된 총 인력이다. $a = 1/2t_d^2$ 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 인력의 약 40%에 도달하는 시점이다. 식 (2)에서 $a = 1/2t_d^2$ 를 치환하면, 식 (3)이 된다.

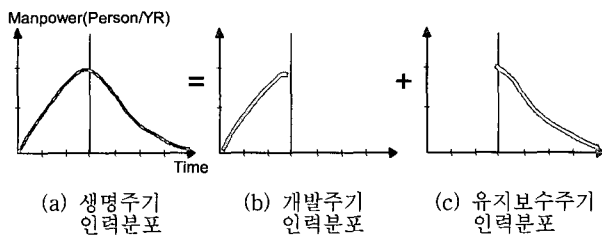
$$\dot{y} = \frac{E}{t_d^2} t e^{-\frac{t^2}{2t_d^2}} \quad (3)$$

Boehm[12]은 소프트웨어 개발 초기에 느린 개발인력 형성 부분과 마지막 단계에서의 점차 길게 감소하는 Rayleigh 분포 특성이 대부분의 소프트웨어 프로젝트 개발인력 분포에 일치하지 않음을 지적하였다. Pillai et al.[5]은 Putnam[4]의 Rayleigh 인력분포 모델의 문제점을 해결하기 위해 식 (4)의 Gamma 모델을 제시하였다.

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

기존에 제안된 소프트웨어 생명주기 인력분포 모델들은 소프트웨어의 생명주기 전체를 거론하고 있다는 점과 순차적 개발 프로세스인 폭포수 모델에만 한정되어 있다는 한계점을 갖고 있다.

Warburton[15]은 실제 프로젝트에 투입된 인력분포를 관찰한 결과 대부분이 개발에 투입되고 약간의 인력이 유지보수 단계에 투입됨을 제시하였다. 따라서 소프트웨어 개발이 종료된 시점이 납품 시점이 된다. 이와 같은 이유로 인해 일반적으로 소프트웨어 개발여부를 결정하는 계획단계에서 소프트웨어 개발에 투입되는 인력과 비용이 얼마나 되는지를 추정하는 경우, 개발주기만을 고려하지 운영주기까지 포함된 소프트웨어 생명주기 전체를 고려하지 않는다. 또한, 폭포수 모델과 UP도 전체 소프트웨어 생명주기를 다루지 않고 단지 개발 주기만을 고려대상으로 하고 있다. 만약에 (그림 3)(a) 생명주기 전체에 대한 인력분포 모델을 개발 프로세스에 적용할 경우, 유지보수 주기 인력분포인 (그림 3)(c)를 제외시키고 (그림 3)(b)의 인력이 계속 상승하는 지수분포만을 적용해야 하므로 개발과정에서 실제 투입되는 인력분포를 표현할 수 없는 문제점에 직면한다.



(그림 3) Rayleigh 분포의 생명주기 인력분포 구성

그러므로 개발 주기에 한정하여 얼마의 인력과 비용이 투입되며, 개발기간이 얼마나 소요되는가를 추정하는 것이 사업 관리 측면에서 중요한 고려대상이 된다. 그러나 지금까지 개발 주기에 한정된 인력분포 모델이 제시되지 않고 있다. 따라서, 3장에서는 소프트웨어 개발주기에 한정된 인력 프로파일 이 어떤 분포를 갖고 있는지 유도하고 적합성을 검증해 본다.

3. 소프트웨어 개발인력 분포 유도

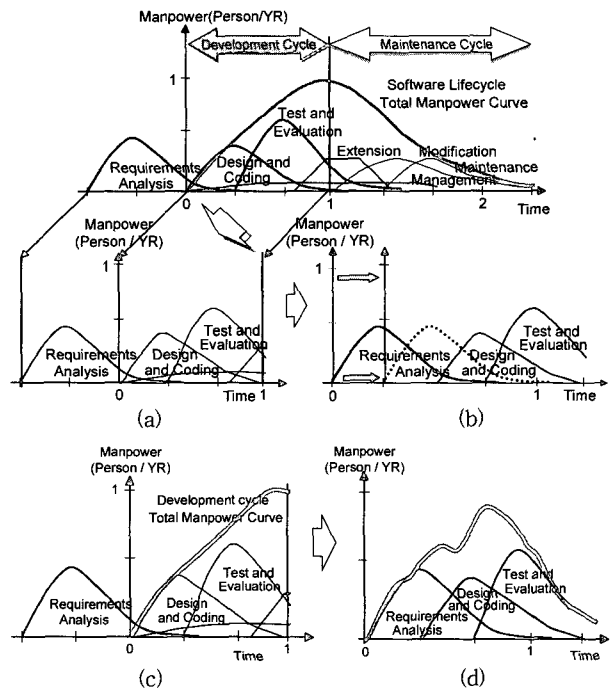
3.1 개발인력 분포 유도

본 장에서는 소프트웨어의 생명주기에서 운영주기를 제외한 개발주기만을 대상으로 개발인력을 어떻게 투입하고 있는지를 유도하고자 한다.

3.1.1 순차적 개발 프로세스

Putnam[4]의 Rayleigh 인력 분포는 소프트웨어 개발주기와 유지보수 주기를 합친 전체 생명주기를 거론하고 있다. 여기서 개발주기만을 고려해 보자. 먼저 (그림 2)의 소프트웨어 생명주기 인력분포로부터 폭포수 프로세스에 대한 개발주기 인력분포만을 유도하였다. 인력분포 유도 과정은 (그림 4)에 제시되어 있으며, 다음 순서로 진행되었다.

- 단계 1 : 일반적으로 폭포수 프로세스의 소프트웨어 개발주기는 (그림 1)과 같이 일반적으로 설계단계가 아닌 요구사항 분석단계부터 시작한다. 따라서, Requirements Analysis, Design and Coding, Test and Evaluation 단계까지를 고려하여 소프트웨어 개발 시작 시점을 Design & Coding에서 Requirements Analysis 시작시점으로 이동시킨다. (그림 4)(a)
- 단계 2 : 그림에서 다음 단계는 이전 단계의 절정에 도달한 시점(약 40% 인력 투입)에서 중첩되어 시작됨으로, Requirements Analysis 분포를 우측으로 약 40% 시점(1칸) 이동시킨다. (그림 4)(b)
- 단계 3 : 단계 2의 결과로부터 각 시점에서 투입되는 인력을 누적시킨 분포를 그리면 (그림 4)(d)가 얻어진다.

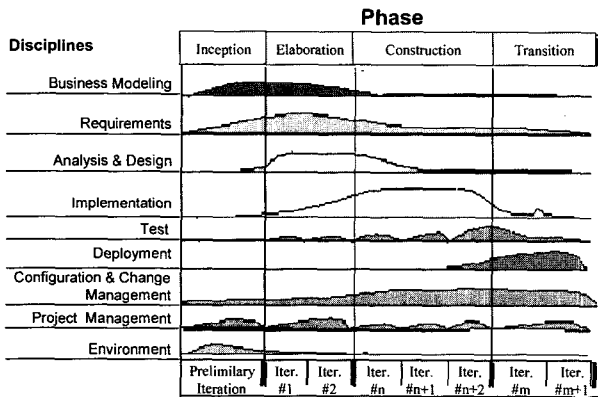


(그림 4) 폭포수 모델의 개발 인력 분포 유도

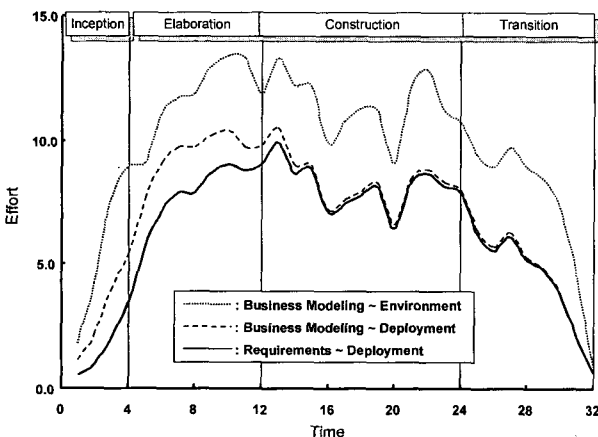
따라서, (그림 4)(d)에 대한 개발주기 인력분포를 얻을 수 있다. 모델의 단순화를 기하기 위해 이러한 변환 과정에서 Extension과 Management에 소요되는 인력분포는 제외시켰다. (그림 4)로부터 개발주기에 투입된 인력을 누적한 총 투입 인력 분포가 Rayleigh 분포의 전반부 40%인 인력 상승 부분 곡선과는 완전히 다른 형태의 분포를 나타내었다. 따라서, 폭포수 프로세스를 적용하여 소프트웨어를 개발할 경우 투입되는 인력의 분포를 추정하기 위해서는 (그림 4)(c)의 Putnam[3]의 Rayleigh 인력분포 중 인력이 상승하는 부분 (개발 주기)만을 적용할 수 없으며, (그림 4)(d) 인력분포에 적합한 모델 제시가 필요하다.

3.1.2 반복적 개발 프로세스

(그림 5)는 4개의 순차적 단계 (Phase)와 9개의 Discipline으로 구성된 UP의 생명주기를 나타내고 있다[16] 다이어그램의 하단부에는 반복 (Iteration)을 볼 수 있으며, 이는 프로젝트를 작은 규모로 반복적으로 개발하여 초기에 고객에게 제공하고, 고객의 의견을 반영하여 수정 보완시키는 장점이 있다. 이와 같이 함으로써 고객과 개발자간 의사소통 향상으로 프로젝트의 위험요소를 감소시키는 효과를 얻을 수 있다.



(그림 5) Unified Process

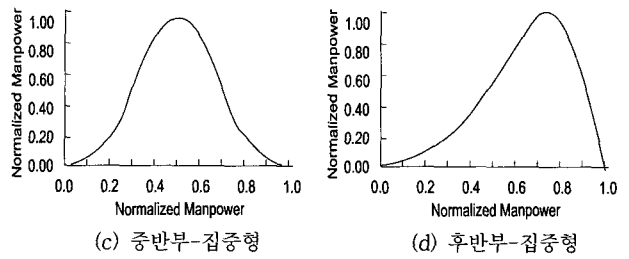
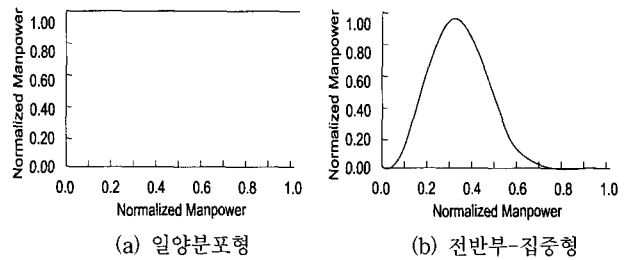


(그림 6) UP의 총 투입 인력 분포

(그림 5)의 UP에서 개발 행위와 직접 관련된 상위 6개 Discipline에 투입되는 인력 분포를 살펴보자. Business Modeling, Requirements와 Analysis & Design은 Rayleigh 분포를 따르지만 Implementation, Test와 Deployment는 Rayleigh 분포를 따르지 않는 특성을 갖고 있다. (그림 5)의 각 반복에서 수행되는 Discipline 들의 투입 인력들을 누적한 결과가 반복 프로세스에 기반하여 소프트웨어 개발에 투입된 인력 프로파일에 해당된다. 이는 (그림 6)에 제시하였다.

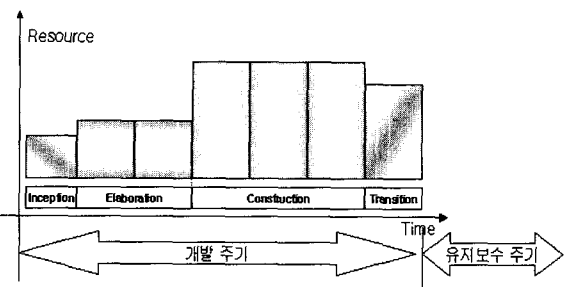
3.2 유도된 모델의 적합성 검증

일반적으로 소프트웨어 개발에 투입될 수 있는 가능한 모든 인력 프로파일에 대한 형태는 (그림 7)과 같이 판단할 수 있을 것이다.



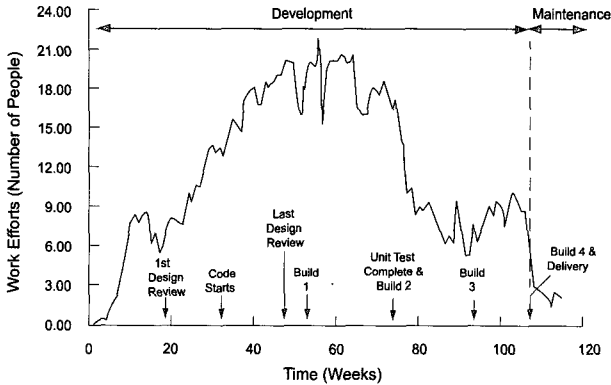
(그림 7) 가능한 소프트웨어 개발 인력 프로파일 형태

반복 프로세스에 대해 Kruchten[17]은 (그림 8)의 개발주기 인력분포를 제안하였다. 이 프로파일은 (그림 7)(d)에 해당하는 후반부-집중형에 해당한다.



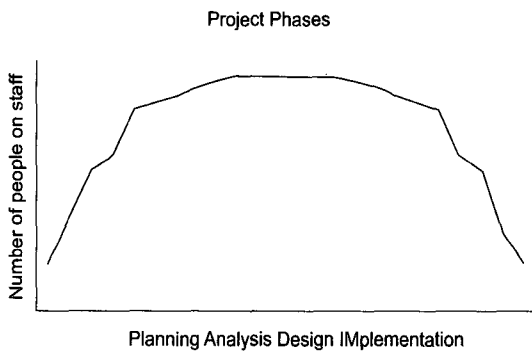
(그림 8) 반복 프로세스(1-2-3-1 반복)의 개발주기 인력분포

Warburton[15]가 제시한 폭포수 프로세스를 적용한 실제 프로젝트의 개발인력 프로파일은 (그림 9)에 제시되어 있으며 폭포수 모델을 적용하여 구조적 방법으로 개발된 경우이다. 이 프로파일은 중반부-집중형 부류에 속한다고 할 수 있다.

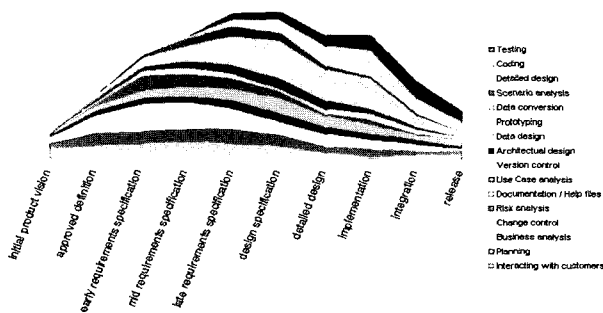


(그림 9) 소프트웨어 생명주기 투입인력 사례

Mejias[18]은 (그림 10)의 전형적인 폭포수 프로세스에 기반한 프로젝트의 여러 단계에 대한 인력 프로파일을 제안하였다. 또한, DSW 그룹[19]은 폭포수와 반복 프로세스에 대해 (그림 11)과 같은 인력 프로파일을 제안하였다. 이 두 프로파일 모두 중반부-집중형에 속한다.



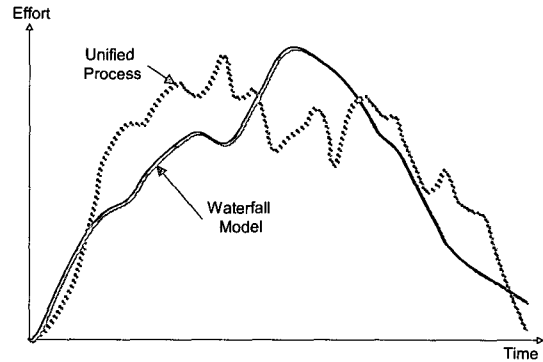
(그림 10) 전형적인 폭포수 프로세스의 인력 프로파일



(그림 11) 폭포수 프로세스와 반복 프로세스의 인력 프로파일

대부분의 프로젝트들이 (그림 7)(a)와 같이 일정한 인력을 투입한다면 프로젝트가 효율적으로 수행되지 않는다[9]. 투입 인력이 절정에 도달하는 시점은 프로젝트의 특성이나 적용되는 프로세스에 따라 프로젝트의 전반부, 중반부 또는 후반부에 인력을 집중 투입하는 다양한 곡선 형태를 보일 수 있다. 따라서, (그림 7)(b)~(그림 7)(d)와 같이 프로젝트 인력을 투입할 수 있다.

지금까지 거론된 대부분의 인력 프로파일들은 중반부-집중형에 속한다고 할 수 있다. (그림 4)의 폭포수모델 기반 개발인력 분포와 (그림 6)의 UP 기반 개발인력 분포에 대해 시간과 인력을 동일한 비율(Scale)로 설정하고 그리면 (그림 12)와 같다.



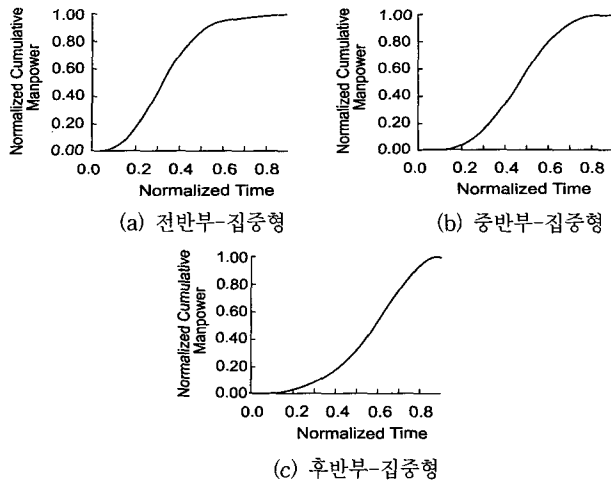
(그림 12) 폭포수와 UP 투입인력분포 형태

그림에서 UP는 폭포수 모델과 동일하도록 Requirements 부터 시작하여 Deployment 까지를 개발 생명주기로 범위를 설정한 인력분포를 고려하였다. 순차적과 반복 개발 프로세스 상호간에 개발단계의 각 시점에서 수행되는 업무와 투입되는 인력의 양에는 차이가 발생한다. 그럼에도 불구하고 두 가지 다른 개발 프로세스의 총 투입 인력의 분포는 (그림 12)와 같이 유사한 형태를 취한다는 결론을 얻을 수 있다. 이 프로파일도 중반부-집중형임을 알 수 있다. 즉, 본문에서 유도된 인력 프로파일은 Warbuton[14], Mejias[15]와 DSW 그룹[19]이 제안한 중반부-집중형 인력 프로파일에 속하므로 인력 프로파일 유도의 적합성이 검증되었다고 할 수 있다. 단지, 각 프로세스의 구별되는 특징은 UP는 개발 전반부인 Requirement Analysis 단계에 많은 인력이 투입되는 반면, 폭포수모델은 Design & Coding, Testing & Evaluation등 개발의 후반부로 접어들수록 많은 인력이 투입된다는 점이다. 따라서, 모든 소프트웨어 개발주기에 투입되는 인력분포는 적용되는 프로세스와 무관하게 하나의 통계적 인력분포 모델로 제안이 될 수 있다는 결론을 얻을 수 있으며, 4장에서는 이 모델을 제안한다.

4. 소프트웨어 개발 라이프사이클 인력분포 모델

4.1 모델 제안

본 논문에서 유도된 (그림 12)의 순차적과 반복 프로세스에 대한 중반부-집중형 개발인력 프로파일이나 Putnam의 Rayleigh 분포[4] 모델과 같은 전반부-집중형, Kruchten[17]의 후반부-집중형 개발 인력 프로파일 형태들에 대해 시간에 대한 누적된 투입 인력을 그리면 (그림 13)과 같이 유사하게 모두 S자 곡선 형태를 따른다. 단지 굴곡이 발생하는 시점만이 다른 차이점을 갖고 있다.



(그림 13) 개발인력 프로파일의 누적 분포 형태

S자 곡선을 적절히 표현할 수 있는 통계적 모델로는 본 논문에서는 식 (5)의 시그모이드 함수(Sigmoid Function) [20]를 적용한다. 그러나 다른 모델 적용도 가능하다. 시그모이드 함수는 그리스 문자인 Sigma 글자와 같은 형태를 갖고 있어 불리워진 함수로 신경망에서 뉴론의 입력과 출력간의 비선형성 정도를 표현하는데 사용되고 있다[21].

$$y = \frac{1}{1 + e^{-at}}, \quad \dot{y} = \frac{e^{-at}}{(1 + e^{-at})^2} \quad (5)$$

식 (5)의 형태를 취하는 식 (6)의 모델을 제안하여 시간에 따른 투입인력을 추정한다. 여기서 a 와 b 는 상수이다. 본 제안 모델을 “SDLM(Software Development Life Cycle Model)” 이라 칭한다.

$$y = \frac{Ke^{-a}}{(1 + be^{-a})^2}, \quad a = \frac{t-a}{b}, \quad a > 0, b > 0 \quad (6)$$

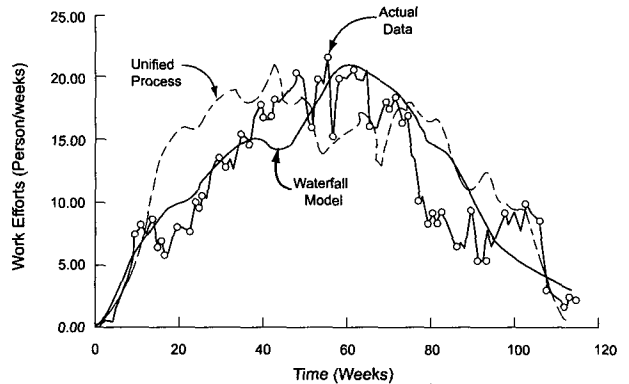
4.2 모델 평가

본 논문에서는 순차적 개발형태인 폭포수 모델과 반복적 개발형태인 UP에 대한 개발인력 프로파일은 중반부-집중형 형태를 취함을 유도하였다. 따라서, 제안된 SDLM 모델을 중반부-집중형 개발인력 프로파일에 적용하여 적합성 여부를 검증한다. 추가로, 제안된 SDLM 모델이 전반부-집중형과 후반부-집중형 개발인력 프로파일에도 유연성을 갖고 적합한지를 검증한다.

4.2.1 중반부-집중형 개발인력 프로파일 적용

(그림 12)의 중반부-집중형 개발인력 프로파일에 적용하여 제안된 SDLM 모델의 적합성을 평가하여 보자. 실험에는 Phillai et al.[5]이 Gamma 모델 제안시 적용한 (그림 8)의 Warburton[15]의 데이터를 이용한다. 먼저 (그림 9) 데이터가 적절히 선택되었는지 검증하기 위해 (그림 12)의 인력 분포를 이 데이터에 중첩시켜 본 결과를 (그림 14)에 제시하

었다. 그림에서 알 수 있듯이 제시된 실제 투입인력 분포 데이터는 구조적 개발의 폭포수 모델과 객체지향 개발의 UP 투입인력 분포가 모두 적합함을 알 수 있다.



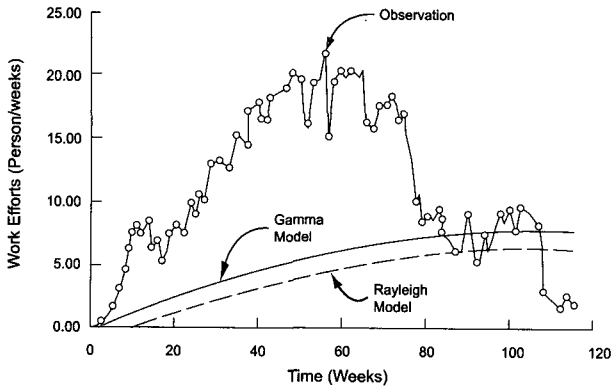
(그림 14) 개발인력 분포 추세 비교

시스템 개발이 t_n 시점까지 진행되었다고 가정하면, 각 개발시간에서의 투입인력 $\hat{y}_i, y_i = \sum_{i=1}^n \hat{y}_i, i = 1, 2, \dots, n$ 데이터를 얻을 수 있다. 현재 시점에서 투입된 인력 추정과 $t_n + d, (d \geq 1)$ 시점에서 투입되어야 하는 인력을 예측하기 위해서는 먼저 모델에 있는 모수를 추정해야만 한다. 모수를 추정하는 방법들 중에서 최우추정법(Maximum Likelihood Estimation Method, MLE)을 많이 사용하고 있으며 이의 대안으로 최소자승법(Least Squares Method)도 많이 적용하고 있다. 최소자승 추정치는 식 (7)을 최소화시킴으로써 얻어진다[22]. 여기서, \bar{y}_i 는 y_i 의 평균값이다.

$$\sum_{i=1}^n [y_i - \bar{y}_i]^2 \quad (7)$$

제안된 모델의 적합성을 평가하기 위해 Putnam[4]의 Rayleigh 분포인 식 (3), Phillai et al.[5]의 Gamma 분포인 식 (4)와 더불어 제안된 SDLM 모델인 식 (6)에 대해 모수를 추정하고 실제 투입 인력 데이터와 비교해 본다. Putnam[4]의 Rayleigh 분포 모델과 Phillai et al.[5]의 Gamma 분포 모델은 인력이 절정에 도달하는 시점 (식 (3)과 식 (4)에서 t_d)이 개발 종료 시점인 경우(원 모델이 제안한 시점)와 개발과정에서 설계나 코딩이 진행되는 시점(개발 종료시점에서 실제시점으로 이동)로 변경한 경우에 대해 적용하였다.

먼저, 인력이 절정에 도달하는 시점을 개발 종료시점인 경우 Putnam[4]의 Rayleigh 분포 모델과 Phillai et al.[5]의 Gamma 분포 모델의 인력분포 추정 결과는 (그림 15)에 제시되어 있다. 그림에서 보듯이 두 모델 모두 실제 투입된 인력분포를 표현하는 데 실패하였음을 알 수 있다.



(그림 15) 개발종료시점에서 인력이 절정에 도달한 경우 추정 결과

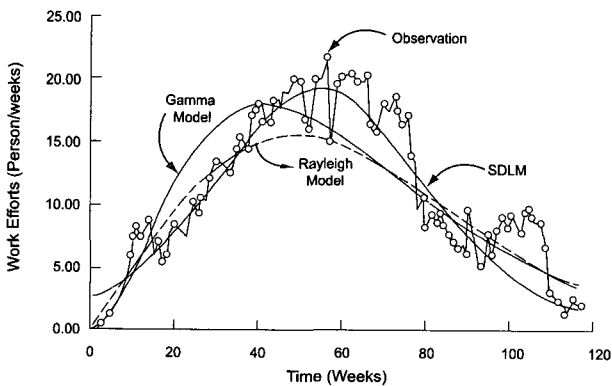
다음으로, 투입인력이 절정에 도달하는 시점을 원 모델에서 가정한 개발종료시점이 아닌 개발 중 발생한다고 가정하고 모수를 추정한 후, 실제 투입 인력분포를 얼마나 잘 표현하는지를 (그림 16)에 제시하였다. 그림에서 Rayleigh 모델

$$\text{의 추정 값은 } y = 2 * 1,381.2 * \frac{1}{(2 * 106)^2} * t * e^{-\frac{1}{(2 * 106)^2} t^2},$$

$$\text{Gamma 모델의 추정 값은 } y = 4 * \frac{1,252.1}{106^3} * t^2 * e^{-\frac{2}{106t}},$$

$$\text{SDLM 모델은 } y = 1,300.8 * \frac{e^{-\left(\frac{t-7.014}{16.711}\right)}}{\left(1 + 16.711 * e^{-\left(\frac{t-7.014}{16.711}\right)}\right)^2}$$

를 얻었다. 이러한 가정에서는 Putnam[4]의 Rayleigh 분포 모델과 Phillai et al.[5]의 Gamma 분포 모델들이 실제 투입된 인력분포를 어느 정도는 표현할 수 있음을 알 수 있다. 투입 인력이 절정에 도달하는 시점이 개발 중에 발생한다는 가정하에 Putnam[4]의 Rayleigh 분포 모델과 Phillai et al.[5]의 Gamma 분포 모델을 적용하였음에도 불구하고, 제안된 SDLM 모델이 실제 인력분포를 전반적으로 가장 정확히 표현하고 있다. 그러나, 일부 프로젝트 시작시점과 종료시점에서는 많은 편차를 가짐을 알 수 있다.

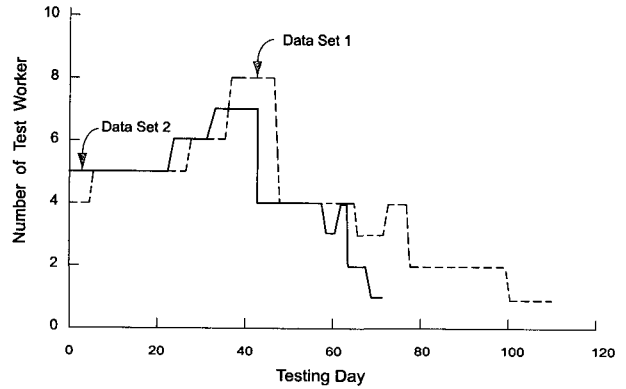


(그림 16) 개발 중에 투입인력이 절정에 도달한 경우 추정 결과

프로젝트 종료시점은 실제 데이터가 구축된 프로그램의 오류를 수정하기 위한 과다한 인력 투입에 따라 비정상적으로 발생한 현상으로 고려대상에서 제외시킬 수 있다. 따라서, 제안된 모델은 프로젝트 시작시점에서만 편차를 많이 보이는 단점을 가지고 있다.

4.2.2 전반부-집중형 개발인력 프로파일 적용

실험에는 (그림 17)의 2개 데이터를 적용하였다. Data Set 1은 Tohma와 Tokunaga[23]으로부터 얻어졌으며 실시간 제어 시스템에 대해 총 331명의 인력이 투입되어 시험을 수행한 결과 72일 동안에 481개의 결함을 발견하였다. Data Set 2는 Tohma, Jacoby, Murate와 Yamamoto[24]의 Table 1 데이터로 111일 동안 운영시험을 한 결과 481개의 결함이 발견되었으며, 이에 투입된 노력은 482명이다. 이들 2개 데이터는 모두 전반부-집중형 인력 프로파일 형태를 취하고 있다.

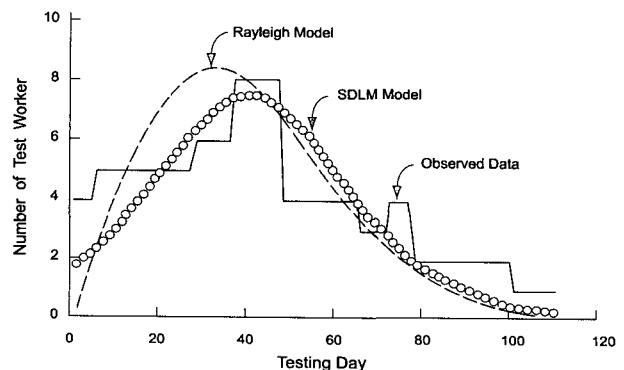


(그림 17) 시험단계 투입 인력

Data Set 1에 대해 Rayleigh 분포와 USDLM 모델의 추정 결과와 실측치 (Observed Value)는 (그림 18)에 제시하였다. 본 데이터에 대한 Rayleigh 모델의 추정 값은 $y = 337 * 0.001 * t * e^{-\frac{0.001}{2} * t^2}$, SDLM 모델의 추정 값은 $y =$

$$337 * \frac{e^{-\left(\frac{t-0.001}{12.529}\right)}}{\left(1 + 12.529 * e^{-\left(\frac{t-0.001}{12.529}\right)}\right)^2}$$

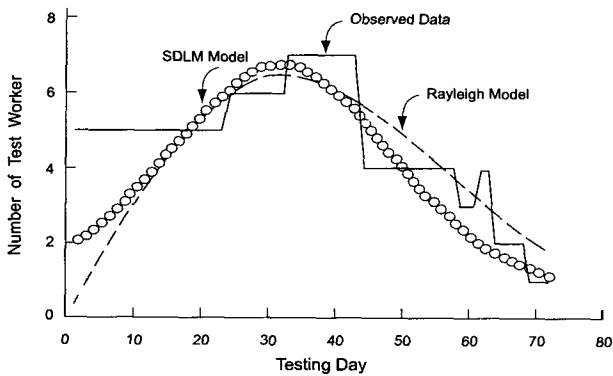
를 얻었다.



(그림 18) 시험단계 투입 인력의 실측치와 추정치(Data Set 1)

직관적으로 볼 때, Data Set 1의 시험노력(시험인력)에 대해 Rayleigh 모델은 개발노력의 분포는 잘 표현하고 있으나 시험 시작시점에서의 개발노력의 양을 표현하지 못하며, 일정한 시점이 지난 후에야 개발노력의 양을 표현할 수 있음을 알 수 있다. 이에 반해, SDLM 모델은 시험노력의 분포 형태도 잘 표현할 수 있으면서 시험 시작 시점의 큰 노력의 양도 어느 정도는 표현하는 모델임을 알 수 있다.

Data Set 2에 대한 모델의 추정 결과와 실측치는 (그림 19)에 제시하였다. Rayleigh 모델의 추정 값은 $y = 442 * 0.001 * t * e^{-\frac{0.001}{2} * t^2}$, SDLM 모델의 추정 값은 $y = 442 * \frac{e^{-\frac{(t-0.001)}{14.927}}}{(1 + 14.927 * e^{-\frac{(t-0.001)}{14.927}})^2}$ 를 얻었다.

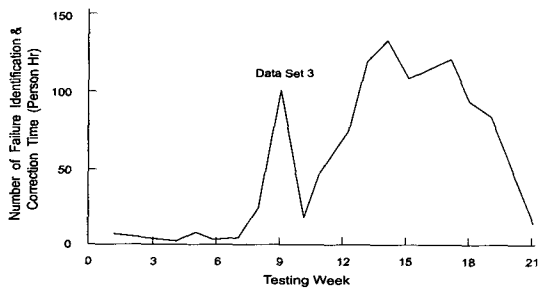


(그림 19) 시험단계 투입 인력의 실측치와 추정치(Data Set 2)

그림에서 Data Set 2의 시험노력(시험인력)에 대해 Rayleigh 모델은 시험노력의 분포 형태는 잘 표현하고 있으나, 시험 시작 시점의 투입 노력의 양을 일정 시점이 지난 후에야 적절히 표현할 수 있다. 이에 비해, SDLM 모델은 시험노력의 분포 형태와 시험 시작 시점의 노력의 양도 어느 정도는 표현하는 모델임을 알 수 있다.

4.2.3 후반부-집중형 개발인력 프로파일 적용

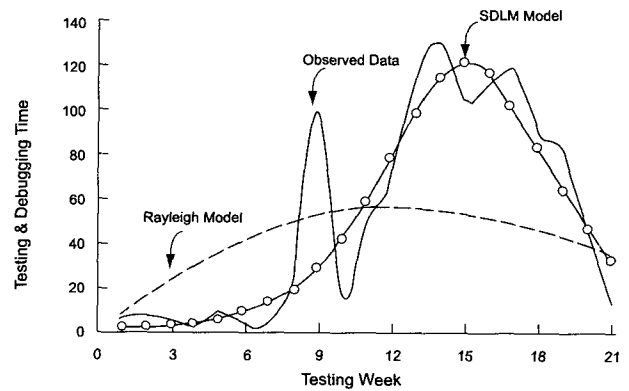
실험에는 (그림 20)의 Data Set 3을 적용하였으며, 이 데이터는 Musa, Iannino와 Okumoto[25]로부터 획득하였다. 실험 시간 명령을 수행하는 21,000 라인의 System T1 프로그램으로 21주 동안 고장 식별과 제거에 1141 시간이 투입되었다.



(그림 20) 시험과 디버깅 투입 인력

DS5에 대한 모델의 추정 결과와 실측치는 (그림 21)에 제시되어 있다. Rayleigh 모델의 추정 값은 $y = 1,141.6 * 0.007 * t * e^{-\frac{0.007}{2} * t^2}$, SDLM 모델의 추정 값은 $y = 1,141.6 * \frac{e^{-\frac{(t-13.244)}{2.273}}}{(1 + 2.273 * e^{-\frac{(t-13.244)}{2.273}})^2}$ 를 얻었다. 그림에

서 시험노력 (시험과 디버깅 시간)에 대해 Rayleigh 모델은 시험노력의 분포 형태를 추정하는데 실패하였다. 단지 SDLM 모델만이 시험노력의 분포 형태와 시험 시작 시점의 노력의 양도 모두 잘 표현하는 모델임을 알 수 있다.



(그림 21) 시험과 디버깅 투입 인력의 실측치와 추정치

직관적 모델평가와 더불어 모델의 성능을 보다 이론적 관점에서 평가해 보자. 모델의 성능을 비교하기 위해 MMRE (Mean Magnitude of Relative Error)를 비교한다. n 개의 데이터에 대한 $MSE = 1/n \sum_{i=1}^n (\text{실측치} - \text{추정치})^2$ 이다. 상대

오차 (Relative Error, RE)는 $RE = \frac{(\text{실측치} - \text{추정치})}{\text{실측치}} \times 100(\%)$ 로 구해지며, MRE (Magnitude of the RE) = |RE| 이며, n 개의 데이터에 대한 MMRE (Mean MRE) = $\frac{1}{n} \sum_i MRE_i, i = 1, 2, \dots, n$ 로 계산된다. MMRE가 작은 값이면

좋은 모델임을 알 수 있다. 실험에 적용된 각 데이터에 대해 모델의 성능을 평가한 결과는 <표 1>에 제시되어 있다. MMRE 측면에서 모델의 성능을 분석한 결과도 모두 SDLM 모델이 Putnam[4]의 Rayleigh 모델보다 좋은 결과를 나타내고 있다.

<표 1> 실험 데이터에 대한 모델 성능 평가

인력 프로파일 형태	Data Set	MMRE	
		Rayleigh 모델	SDLM 모델
중반부-집중형	Warrenton 데이터	72.87	47.37
전반부-집중형	Data Set 1	41.91	32.85
	Data Set 2	26.14	20.11
후반부-집중형	Data Set 3	251.38	57.88

5. 결론 및 향후 연구과제

소프트웨어 프로젝트 관리 측면에서 사업계획 작성시 프로젝트 개발에 소요되는 인력, 비용과 개발기간 등의 신뢰할 만한 수준의 정보 도출은 사업의 성공 여부를 결정하는데 필수적이다. 프로젝트 개발에 투입되는 인력 규모를 판단하기 위해서는 Putnam[4]의 Rayleigh 분포 모델이나 Phillai et al.[5]의 Gamma 분포 모델을 적용해야 한다. 그러나 두 모델 모두 개발이 종료되는 시점에서 투입되는 인력이 절정에 도달한다는 가정하에 모델을 제시하였다. 이에 비해 실제 프로젝트에서는 대부분의 인력이 개발에 투입되고 단지 적은 규모의 인력만이 유지보수에 할당된다. 또한, 대표적인 개발 프로세스인 폭포수 모델이나 UP 모두 개발단계만을 대상으로 하고 있다. 이러한 개발환경에는 투입 인력이 지속적으로 증가하여 개발완료시점에서 절정에 도달하는 기존 인력분포 모델의 투입인력 분포 곡선을 적용할 수 없다. 본 논문은 이러한 문제점을 해결하기 위해 개발단계에 적합한 인력분포 모델을 제시하였다. 먼저, Putnam[4]이 제시한 인력분포로부터 폭포수 모델의 개발단계 투입 인력 분포를 유도하였다. 다음으로, UP에 대한 개발단계 투입인력 분포를 유도하였다. 두 프로세스 투입인력을 비교한 결과 개발 각 시점에서 투입되는 인력의 양과 수행되는 업무에는 차이가 발생하지만 총 투입 인력분포는 유사한 형태를 취한다는 결론을 도출하였다. 따라서, 이 두 프로세스에 적합한 통계적 개발인력 분포 모델을 제시하였다.

본 논문은 소프트웨어 개발시 적용되는 개발프로세스에 종속되지 않는 단일화된 인력분포 모델을 제시하였다는 점에서 연구 성과가 크다고 할 수 있다. 그러나 현실적으로 소프트웨어 개발에 투입되는 인력의 분포는 보다 다양한 패턴을 갖고 있을 것으로 예상됨에도 불구하고 다수의 실제 수행된 프로젝트를 대상으로 모델의 적합성을 보이지 못하여 일반화된 모델 제시는 되지 않을 수도 있다. 따라서, 추후 이 분야에 대한 연구를 수행할 것이다.

참 고 문 헌

- [1] J. Verner and G. Tate, "A Software Size Model," *IEEE Trans. on Software Eng.*, Vol.18, No.4, pp.265-278, 1992.
- [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] KRM, "Project Management Glossary," Key Results Management, Inc., 2004.
- [4] 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.
- [5] 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.
- [6] J. Lattanze, "Managing Software Development : The Software Development Lifecycles," Carnegie Mellon University School of Computer Science, Pittsburg, 2000.
- [7] P. Noe, "The Unified Software Development Process," San Antonio SPIN, 2000.
- [8] P. Kruchten, "Software Maintenance Cycles with the RUP," http://www.therationaledge.com/content/aug_01/t_softwareMaintenance_pk.html.
- [9] G. T. Heineman and W. T. Councill, "Component-Based Software Engineering - Putting the Pieces Together," Addison-Wesley, 2001.
- [10] P. Kruchten, "Going Over the Waterfall with the RUP," http://www.therationaledge.com/content/sep_01/t_waterfall_pk.html.
- [11] W. Royce, "The Rational Unified Process : A Commercially Available Spiral Model Implementation," <http://www.sei.cmu.edu/cbs/spiral2000/february2000/Royce/>.
- [12] B. W. Boehm, "Software Engineering Economics," Prentice Hall, 1981.
- [13] 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.
- [14] P. V. Norden, "Project Life Cycle Modeling : Background and Application of the Life Cycle Curves," U. S. Army Computer System Command, 1977.
- [15] R. D. H. Warburton, "Managing and Predicting the Costs of Real-Time Software," *IEEE Trans. on Software Eng.*, Vol. SE-9, No.5, pp.562-569, 1983.
- [16] A Rational Software Corporation White Paper, "Rational Unified Process : Best Practices for Software Development Teams," Rational Software Corporation, 1998.
- [17] P. Kruchten, "Planning an Iterative Project," *The Rational Edge e-zine for the Rational Community*, http://www.therationaledge.com/content/oct_02/f_iterativePlanning_pk.jsp, Oct., 2002.
- [18] Mejias, "Project Management Overview," <http://www.enterprise-works.com/ProjMgmt.html>, Enterprise-Works.com, 2001.
- [19] DSW Group, "The Software Estimation Story," <http://www.dswgroup.com/html/reference/SoftwareEst.html>, 2003.
- [20] M. T. Hargan, H. B. Demuth and M. Beale, "Neural Network Design," PWS Publishing Company, 1996.
- [21] <http://users.computerweekly.net/robmorton/projects/neural/sogmoid.html>.

- [22] Y. Thoma, K. Tokunaga, S. Nagase and Y. Murata, "Structural Approach to the Estimation of the Number of Residual Software Faults Based on the Hyper-Geometric Distribution," IEEE Trans. on Software Eng., Vol.15, No.3, pp.345-355, 1989.
- [23] Y. Thoma and K. Tokunaga, "A Model for Estimating the Number of Software Faults," Inst. Electron. Commun. Eng., (IECE) Japan. Tech. Rep., FTS 86-14, pp.41-46, 1986.
- [24] Y. Tohma, R. Jacoby, Y. Murata and M. Yamamoto, "Hyper-Geometric Distribution Model to Estimate the Number of Residual Software Faults," COMPSAC 89, Orland, Florida, pp.610-617, 1989.
- [25] J. D. Musa, A. Iannino and K. Okumoto, 'Software Reliability Measurement, Prediction, Application,' McGraw-Hill Book Company, 1987.



이 상 운

e-mail : sulee@sky.wonju.ac.kr

1983년~1987년 한국항공대학교 항공전자
공학사(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