

반복 프로세스의 투입인력과 프로젝트 완성도 (A Manpower Loading and Project Completeness for Iterative Process)

박석규 (Seok-Gyu Park)¹⁾ 이상운 (Sang-Un Lee)²⁾

요 약

일반적으로 프로젝트를 수행할 경우 투입되는 인시수, 소요 비용과 프로젝트의 완성도는 학습효과가 반영되어 S자 형태의 곡선을 따른다. 도목건축 분야나 소프트웨어 개발의 폭포수 프로세스는 모두 하나의 주기 동안 한 단계가 완료된 후 다음 단계로 이동하는 순차적인 형태를 취하는 특징을 갖고 있다. 이에 비해 소프트웨어 개발의 반복 프로세스는 하나의 주기동안 폭포수 프로세스를 중첩적으로 여러 번 반복 수행하는 특징을 갖고 있어 일반적으로 적용되는 S자 곡선을 적용할 수 있는지 여부를 판단할 필요가 있다. 이에 대해 Marasco는 반복 수에 비례하는 S자 곡선을 쌓아 놓은 곡선 형태를 따름을 제시하였다. 본 논문은 반복 프로세스를 적용하여 소프트웨어를 개발할 경우 소요되는 누적 인시수, 소요 비용과 프로젝트 완성도에 대한 프로파일의 단지 하나의 S자 곡선을 따름을 증명하고, 이를 추정할 수 있는 모델도 제시하였다. 제시된 이론을 바탕으로 하여 반복 프로세스를 따르는 소프트웨어 개발계획 작성시 의사결정 정보로 활용할 수 있을 것이다.

Abstract

The typical profile, or curve, when plotted cumulative man-hours, cost, and project completeness over time for a whole project results in the shape of the letter "S" because of the learning effect. The numbers of process in Waterfall model or civil/architectural engineering fields have characteristic that pass through various phases for all one cycle. The cycle adopts sequential form that moves by next phase after phase is completed. While, iterative process of software development proceeds by developing in iterations that encompass the activities of phases of waterfall process. How to apply general S-shaped man-hours curve to iterative process? About this, Marasco presented that follow S-shaped curves staked one on top of another that is proportional in number of iteration. In this paper, we propose only one S-shaped curve profile about cumulative man-hour, cost and project completeness in case develop software applying iterative process. Also we present the model that can be estimate of this information. May do with presented theory and utilize to decision-making information at software development plan creation that follow iterative process.

키워드 (Key words) : 폭포수 프로세스 (Waterfall process), 반복 프로세스 (Iterative Process), S자 곡선 (S-shaped Curve), 인시수 (Man-hour), 프로젝트 완성도 (Project Completeness)

논문접수 : 2004. 8. 29.

심사완료 : 2004. 10. 3.

1) 정회원 : 강원도립대학 컴퓨터응용과

2) 정회원 : 국립원주전문대학 여성교양과 전임강사

1. 서 론

프로젝트 관리자가 소프트웨어 개발계획 작성시 직면하는 어려움으로는 “개발될 소프트웨어의 총 규모 (Size)는 얼마인가”, “소프트웨어 개발에 투입되는 총 투입 인력은 얼마인가”, “소프트웨어를 개발하는데 소요되는 기간은 얼마인가”를 추정하는 것이다. 이와 관련하여 소프트웨어 규모 추정에 관해서는 COCOMO[1], COCOMO-II[2], 기능점수 (Function Point) [3,4], COSMIC-FFP[5], 쓰임새점수 (Use Case Point) 기법[6] 등이 적용되고 있다. 추정된 소프트웨어 규모에 기반하여 소프트웨어 개발에 소요되는 총 인시수 (man-hour)와 개발기간 (duration)을 추정하는 모델들이 다양하게 연구되고 있다. 이러한 정보를 토대로 “적용될 개발 프로세스의 단계별로 투입 인력과 기간을 어떻게 할당할 것인가”, “프로젝트 개발과정을 어떻게 관리할 것인가”에 따라 소프트웨어 개발의 성공여부가 결정된다.

소프트웨어 개발 프로세스로는 폭포수 프로세스 (waterfall process)가 일반적으로 널리 적용되고 있다[7]. 그러나 소프트웨어가 점점 더 복잡해지고 대형화되는 추세에 따라 프로젝트의 위험요소를 조기에 제거하기 위한 방법으로 프로토타이핑, 베타 버전 후 정식 버전의 2단계 납품 방식을 취하는 순수한 폭포수 프로세스의 변형형태 또는 반복 프로세스 (iterative process)로 전환되고 있다[7]. 소프트웨어 개발 과정에서 투입되는 인력 규모는 프로세스 단계별로 고유한 특성을 고려하지 않고 단위시간당 일정한 인원을 투입 (즉, 일양분포를 따름) 하면 프로젝트를 능률적으로 수행하지 못한다.[8] 이를 뒷받침하는 근거로 단위시간에 따른 인시수 프로파일 (man-hour profile)과 누적 투입 인시수 (cumulative man-hour)에 관한 연구로서 토목건축공학에 대해서는 Wideman[9], 일반적인 프로젝트 관리분야에 대해서는 Gendall[10]이 있다. 소프트

웨어의 폭포수 프로세스에 대해서는 Putnam[11]의 Rayleigh 분포와 Phillai et al.[12]의 Gamma 분포, 반복 프로세스에 대해서는 Kruchten[8],과 Marasco[13] 등이 있다.

이들 제안된 이론의 공통된 특징은 프로젝트가 진행됨에 따라 투입되는 개발인력이나 프로젝트 달성도의 누적 분포는 학습 효과 (learning effect)를 반영하여 불가사의하게도 “S자 곡선” (s-shaped curve) 형태를 취한다는 것이다.

지금까지 제시된 토목건축 분야와 소프트웨어의 폭포수 프로세스는 계획-요구사항분석-설계(도면작성)-건축 (코딩)-감리(시험)과정을 단지 1회만을 수행 (1 iteration)하는 형태이다. 그러나 반복 프로세스는 기존의 방법들과 다르게 다수의 반복과정을 거쳐 최종적으로 제품이 납품되는 형태를 취하는 특징을 갖고 있다. 이와 관련하여, Marasco[13]는 다수의 반복으로 나누어 개발할 경우 프로젝트 완성도는 반복수에 비례하는 S자 곡선을 쌓아 올린 복잡한 형태를 따른다고 제안하였다. 이에 반해 Gendall[10]은 일반적으로 시간에 따른 누적 개발노력, 개발비용과 프로젝트 완성도는 모두 1개의 S자 곡선을 따름을 제시하였다. Marasco[13]가 제시한 논리를 Gendall[10]의 S자 곡선 이론에 적용하면 개발시간이 경과할수록 누적 투입 인시수, 비용과 프로젝트 달성도 모두 프로젝트의 반복 수에 비례하는 수 만큼의 S자 곡선을 쌓아 올린 분포를 따름을 의미한다. 그러므로 Marasco[13]의 반복 프로세스와 Gendall[10]의 일반적인 프로젝트 관리분야에 대한 이론간에 상반된 결과를 나타내고 있다. 따라서, 반복 프로세스를 적용하는 소프트웨어 개발계획 작성시 폭포수 프로세스에 적용되는 한 개의 S자 곡선을 적용할 수 있는지 아니면 S자 곡선을 반복 수 만큼 쌓아올린 형태를 적용할 것인지에 대한 연구가 필요하다.

본 논문은 반복 개발 프로세스를 적용시 투입되는 누적 인력, 프로젝트 완성도의 프로파일이 Marasco[13]가 제안한 반복 수에 비례하

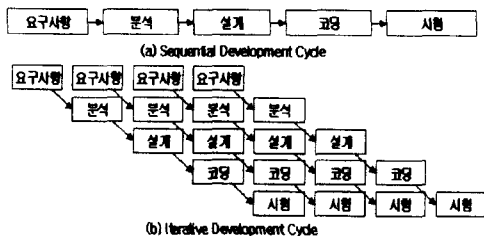
는 S자 곡선을 따르지 않음을 증명하고, 프로젝트 관리를 어떻게 효율적으로 할 수 있는가를 제시한다.

2장에서는 프로젝트 계획과정과 소프트웨어 개발 프로세스의 양대 산맥인 폭포수 프로세스와 반복적 프로세스에 대해 간략히 비교한다. 또한 개발에 투입되는 인력 프로파일과 관련한 연구결과와 문제점을 제시한다. 3장에서는 반복적 프로세스에 대해 투입되는 인력 프로파일에 대한 모델을 제시하고, 모델의 적합성을 검증한다.

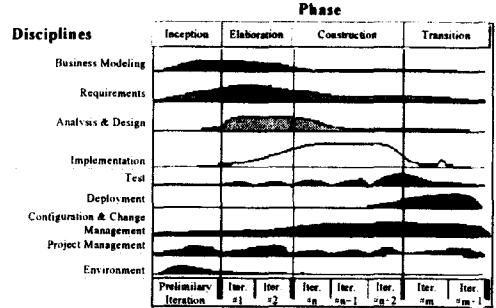
2. 개발 프로세스와 인력 프로파일 관련연구

2.1 소프트웨어 개발 프로세스

소프트웨어 생명주기 (life-cycle)는 개발 주기와 유지보수 주기로 구분된다. 개발과정과 연관된 활동과 산출물을 특정 단계들로 구분시킨 것이 소프트웨어 개발 프로세스이다. 개발 프로세스에는 제품이 한번에 납품되는 빅뱅 (big bang)과 점진적으로 납품되는 반복 프로세스가 있다. 빅뱅 방법은 고전적 생명주기 프로세스로 하나의 단계 (phase)를 100% 완료한 후 다음 단계를 진행하는 선형 (linear), 순차적인 개발 형태를 취하는 폭포수 프로세스가 대표적이다. 이에 비해, 반복적 방법은 나선형 (spiral), 점진형 (incremental), 진화형 (evolutionary) 등 많은 변형된 형태들이 있다 [14]. 순차적인 폭포수 모델과 반복적 개발방법의 예를 [그림 1]에 제시하였다[7]. (그림 2)는 4개의 순차적 단계와 9개의 discipline으로 구성된 UP (unified process)의 생명주기를 나타내고 있다[15].



[그림 1] 대표적인 소프트웨어 개발 방법
[Fig. 1] Representative software development method



[그림 2] 반복 프로세스 (전형적인 1-2-3-2 반복을 수행하는 경우)

[Fig. 2] Iterative process (when achieve typical 1-2-3-2 iteration)

전통적으로 그 동안의 프로젝트들은 discipline들을 순차적으로 한번에 하나씩 진행해 왔으며, 이를 폭포수 프로세스라 한다. 이 방법은 제품을 완성하고 시험하는 시점에서 분석, 설계와 구현단계에서 발견되지 않은 문제들이 수면위로 떠오르게 되어 결함을 수정하기까지 프로젝트 진행이 불가하다. 프로젝트들을 원활히 진행하기 위해 더욱 유연하며 위험이 적은 방법으로는 여러 번에 걸쳐 다양한 개발 discipline들을 수행하는 것이다. 이로써 요구사항을 더욱 확실하게 이해하고 완벽한 아키텍처를 정립하며, 개발팀의 기술을 향상시켜 결국 더욱 완벽하게 구현된 소프트웨어를 생산하게 된다. 이를 반복프로세스라 한다.

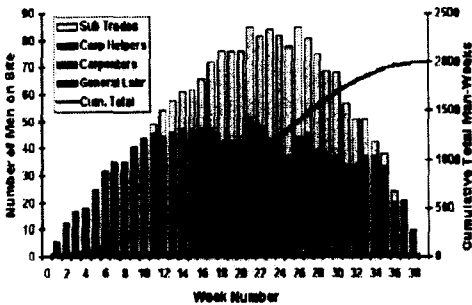
순차적 (폭포수) 프로세스의 개발 주기는 요구사항 (시스템과 소프트웨어), 요구사항 분석, 설계, 코딩, 시험으로 단계가 구분되는데 반해 반복 프로세스의 주기는 4개의 순차적 단계인 도입 (inception, I), 정련 (elaboration, E), 구축 (construction, C)와 전이 (transition, T)로 분할된다. 따라서, 폭포수 프로세스의 단계에 대응되는 것이 반복 프로세스의 discipline들이라

할 수 있다.

다이어그램의 하단부에는 반복 (iteration)을 볼 수 있으며, discipline들의 묶음을 반복이라 한다. 소프트웨어 개발은 반복의 연속으로 이루어지며, 반복을 통해 소프트웨어는 점진적으로 개발된다. 각 단계는 여러 개의 반복으로 나뉘어질 수 있다. 전형적인 순차적 프로세스는 단지 한번의 주기만을 수행한 후 제품을 납품하기 때문에 빅뱅 형태로 한번의 반복만을 수행한다고 할 수 있다. 이에 반해 반복 프로세스는 각 단계에서 하나 또는 그 이상의 반복을 포함하고 있으며, 반복 수는 단계의 중요성, 단계의 목적 달성에 따라 크게 변할 수 있다.

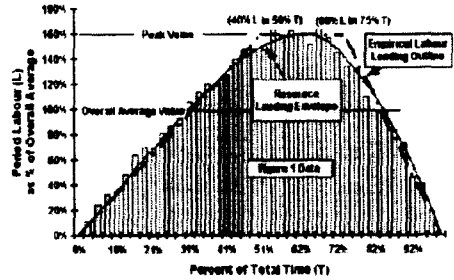
2.2 타 분야의 인력 프로파일

단위시간에 따른 인시수 프로파일과 누적 투입 인시수에 관한 연구로서 토목건축공학에 대해서는 Wideman[9]가 (그림 3)과 (그림 4)의 S자 곡선 분포를 제시하였다. Wideman[9]는 건축공학에서의 경험에 의하면 전형적인 누적 투입 인력과 공사의 진척도 프로파일은 S자 곡선을 나타내며, 차이점은 학습효과에 기인한다고 제안하였다.



[그림 3] 토목 건축의 경우 인력 프로파일과 누적 인력 분포

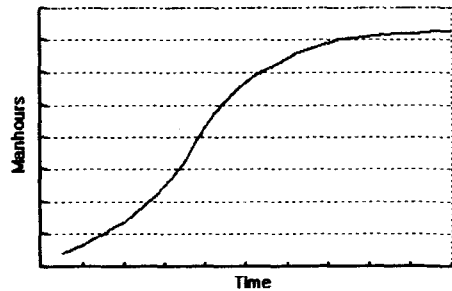
[Fig. 3] Manpower profile analysis and accumulation manpower distribution for civil engineering



[그림 4] 토목건축의 경우 인력 프로파일
[Fig. 4] Manpower distribution for civil engineering

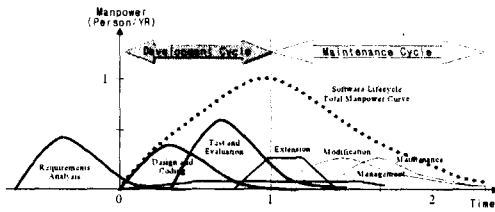
그림에서 알 수 있듯이 토목 건축분야의 경우 건물을 신축하는데 소요되는 인력의 프로파일은 공사기간의 후반부에서 많은 인력이 투입되는 경향을 보이고 있다.

일반적으로 프로젝트 관리를 위해 Gendall[10]은 누적 투입 인시수 vs. 시간, 투입 비용 vs. 시간과 프로젝트 달성도 vs. 시간 곡선이 모두 (그림 5)의 S자 형태를 따른다고 제안하였다. 일반적인 프로젝트 관리시에도 토목 건축공학과 동일하게 후반부에 접어들면서 인력이 많이 필요한 형태를 취한다.



[그림 5] 누적 인시수 프로파일
[Fig. 5] Cumulative manpower profile

2.3 소프트웨어분야 개발주기 인력 프로파일
소프트웨어 생명주기 인력분포는 Putnam[11]이 (그림 6)을 제시하였다.

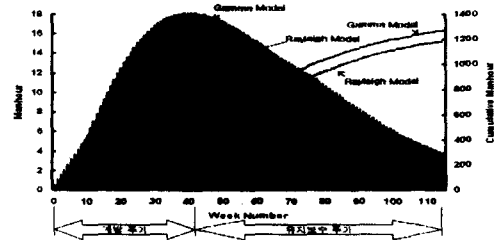


[그림 6] 폭포수 프로세스의 생명주기 인력 분포
 [Fig. 6] Life cycle manpower distribution of waterfall process

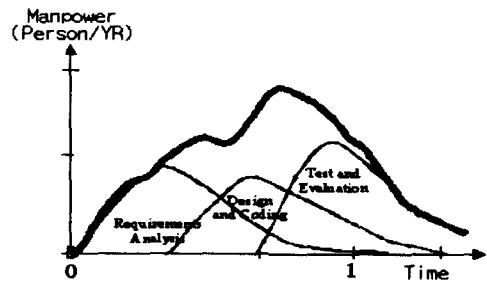
폭포수 프로세스에 대한 생명주기 전체에 소요되는 인력분포로는 (그림 7)과 같이 Putnam[11]의 Rayleigh 분포와 Phillai et al.[12]의 Gamma 분포가 제안되었다. 이들 모델들은 소프트웨어의 개발주기만을 대상으로 하지 않고 개발과 유지보수 주기를 모두 고려한 소프트웨어 생명주기 전반에 걸쳐 투입되는 인력분포를 대상으로 하였다. 소프트웨어 개발 여부를 결정하는 계획단계에서 소프트웨어 개발에 투입되는 인력과 비용이 얼마나 되는지를 추정하는 경우, 일반적으로 개발주기만을 고려하며, 운영단계의 유지보수 주기까지 포함하는 소프트웨어 생명주기 전체를 고려하지 않는다. 소프트웨어 개발 여부를 결정하는 계획단계에서 소프트웨어 개발에 투입되는 인력과 비용이 얼마나 되는지를 추정하는 경우 개발주기에만 한정시켜야 할 필요성이 제기된다. (그림 6)에서 요구사항 분석, 설계와 코딩과 시험과 평가 단계만을 고려하면 폭포수 프로세스에 대한 (그림 9)의 개발주기 인력분포를 유도할 수 있다. 폭포수 프로세스의 인력프로파일은 일반적인 프로젝트나 토목건축공학분야와 유사하게 개발의 후반부에서 인력이 많이 소요되는 특징을 갖고 있다.

반복 프로세스에 대해 Kruchten[8]은 (그림 9)의 개발주기 인력분포를 제안하였으며, 이를 누적시킨 형태는 S자 곡선을 따른다. 인력 프로파일은 보통 도입 단계에서 구축 단계까지 상승하고 잠시동안 안정기를 거친 후 전이 단

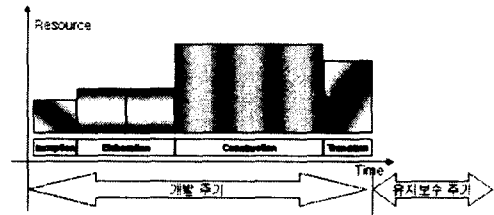
계의 끝으로 하락하는 경향을 나타낸다. 이 프로파일 또한 개발의 후반부로 갈수록 인력이 많이 투입되는 현상을 보인다. 그러나 실제의 인력 프로파일은 프로젝트에 따라 변할 수 있다.



[그림 7] 폭포수 프로세스의 생명주기 투입 인력 (Rayleigh와 Gamma 분포)
 [Fig. 7] Life cycle manpower of waterfall process (Rayleigh & Gamma distribution)

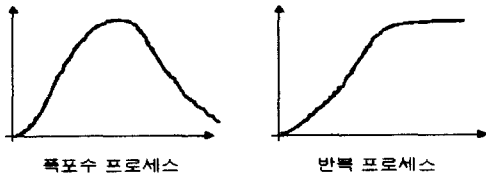


[그림 8] 폭포수 프로세스의 개발주기 인력 분포
 [Fig. 8] development life cycle manpower distribution for waterfall process



[그림 9] 반복 프로세스 (1-2-3-1 반복)의 개발주기 인력분포
 [Fig. 9] Development manpower distribution for iterative process (1-2-3-1 iteration)

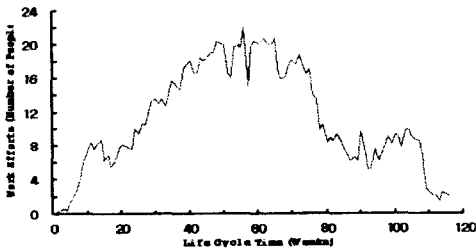
Marasco[16]은 개발주기에 한정하여 (그림 10)의 이론을 제시하였다. 폭포수 프로세스는 초기 단계에서 인력이 상승하는데 비해 반복 프로세스는 인력 투입이 지연된다. 즉, 반복 프로세스는 적은 규모의 엘리트 팀이 도입과 정련 단계에서 빠르게, 최상으로 대부분의 작업을 수행하고, 구축과 전이 단계에서 상당한 규모의 인력을 추가하는 형태를 취한다. 이 이론의 (그림 8)과 (그림 9)의 인력 프로파일과 동일한 형태이다.



[그림 10] 폭포수와 반복 프로세스의 개발주기 인력 프로파일

[Fig. 10] Development manpower distribution for waterfall & iterative process

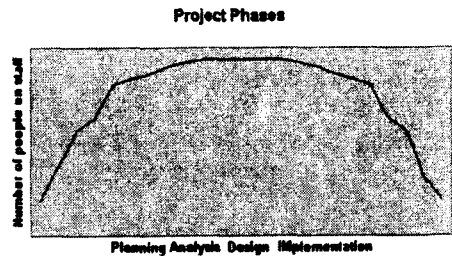
Warburton[17]가 제시한 폭포수 프로세스를 적용한 (그림 11)의 실제 프로젝트의 사례, Mejias[18]이 추천한 (그림 12)의 전형적인 폭포수 프로세스에 기반한 프로젝트의 여러 단계에 대한 인력 프로파일과 더불어 DSW 그룹[19]이 제안한 폭포수와 반복 프로세스의 (그림 13)과 같은 인력 프로파일도 (그림 8)과 유사한 곡선 형태를 나타내고 있다.



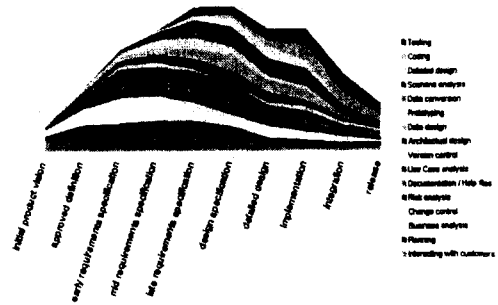
[그림 11] 폭포수 프로세스 적용 실제 프로젝트의 인력분포

[Fig. 11] Manpower distribution of actual project in waterfall process

이와 같이 소프트웨어의 인력 프로파일은 적용되는 프로세스의 차이와 프로젝트의 특성에 따라 (그림 14)와 같이 다양한 곡선 형태를 보일 수 있다.



[그림 12] 전형적인 폭포수 프로세스의 인력 프로파일
[Fig. 12] Manpower distribution for typical waterfall process

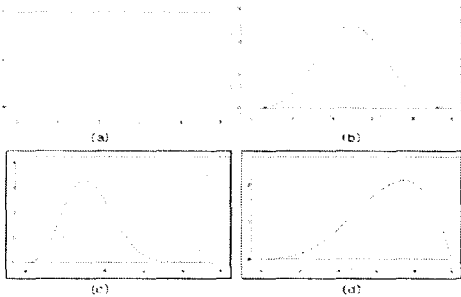


[그림 13] 폭포수 프로세스와 반복 프로세스의 인력 프로파일

[Fig. 13] Manpower distribution for waterfall & iterative process

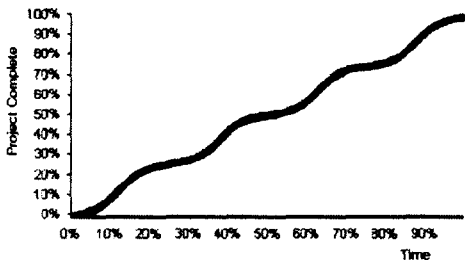
그러나 대부분의 프로젝트들이 (a)와 같이 일정한 인력을 투입한다면 프로젝트가 효율적으로 수행되지 않는다[8]. 투입 인력이 절정에 도달하는 시점에 차이가 발생한다. 따라서, (b)-(d)와 같이 프로젝트 인력을 투입할 수 있으며, 이들 그래프의 누적 투입 인력을 그리면 (그림 5)와 유사하게 모두 하나의 S자 곡선 형태를 따름을 알 수 있다. 결론적으로 대부분의 제안된 이론들의 공통된 특징은 프로젝트가 진행됨에 따라 투입되는 개발인력이나 프로젝트 달성도의 누적 분포는 학습 효과를 반영하여

불가사의하게도 단지 하나의 S자 곡선 형태를 취한다.



[그림 14] 소프트웨어 개발 인력 프로파일
[Fig. 14] Development manpower profile for software

이에 반해, Marasco[13]는 다수의 반복으로 나누어 개발할 경우 프로젝트 완성도는 반복수에 비례하는 S자 곡선을 쌓아 올린 복잡한 형태를 따른다고 제안하였다. 예로, 도입, 정련, 구축과 전이 단계별로 각각 1개의 반복만을 수행하는 경우 각 반복별로 개발자의 학습효과를 반영하여 S자 형태의 프로젝트 완성도가 달성되며, 프로젝트의 누적 완성도는 4개의 S자 곡선을 쌓아 올린 (그림 15)의 형태를 제시하였다.



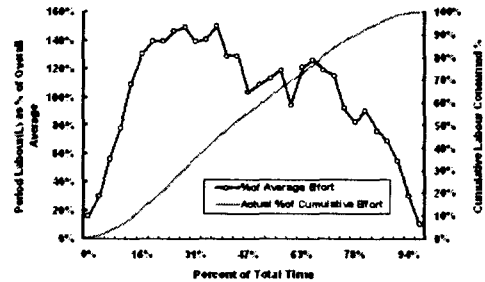
[그림 15] 4개의 반복을 수행하는 경우 개발인력 누적 분포 형태
[Fig. 15] Cumulative development manpower distribution in case of 4 iteration

3. 반복 프로세스의 인력 프로파일 모델

Marasco[13]는 개발시간이 경과할수록 누적 투입 인시수, 비용과 프로젝트 달성도 모두 프로젝트의 반복 수에 비례하는 수 만큼의 S자 곡선을 쌓아 올린 분포를 따름을 제시하였다. 그러나 현실적으로 반복프로세스도 이러한 복잡한 형태의 분포를 따르지 않으며, 단지 하나의 S자 형태의 분포만을 따르지 않을 수 있다. 본 장에서는 반복 개발 프로세스를 적용시 투입되는 누적 인력, 프로젝트 달성도의 프로파일인 Marasco[13]가 제안한 반복 수에 비례하는 수 만큼이 아니라 단지 하나의 S자 곡선을 따름을 증명하고, 프로젝트 계획에 활용하는 방안을 모색하여 본다.

3.1 반복 프로세스의 인력 프로파일 제안

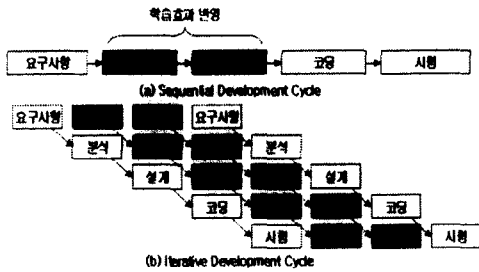
전형적인 UP 형태 (1-2-3-2 반복 수행)인 (그림 2)의 단위시간당 수행되는 모든 discipline들에 대해 누적된 인력분포와 시간에 따른 총 누적 투입인력 분포를 그리면 (그림 16)이 생성된다.



[그림 16] 반복 프로세스의 투입인력 분포
[Fig. 16] Manpower distribution for iterative process

반복 프로세스의 누적 인력 프로파일은 (그림 16)과 같이 S자 곡선을 따르지만 왜 학습효과가 거의 반영되지 않는 것처럼 보이는가? (그림 1)에서 학습효과가 반영되는 단계를 휴리스틱하게 분석과 설계시점으로 판단하여 그

리면 (그림 17)과 같다.



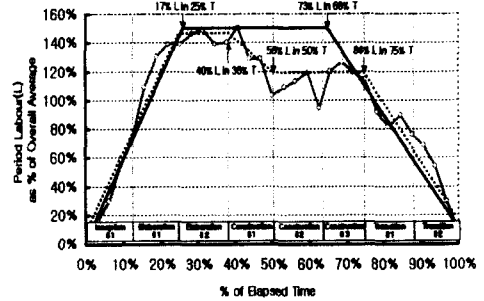
[그림 17] 학습효과 반영 시점
[Fig. 17] Learning effect reflection visual point

Kruchten[20]은 반복 프로세스의 장점은 다양한 분야의 개발자들이 프로젝트 수행 초기부터 경험을 쌓고 능력과 전공이 보다 충만해 지기 때문에 학습효과가 보다 빠르게 나타난다고 제시하였다. 예로, 시험요원은 시험을 프로젝트 초기에 수행하고, 설계요원도 프로젝트 초기부터 설계를 수행하기 때문이다. 이에 비해 순차적 프로세스는 동일한 요원이 그들의 작업을 수행할 때까지 기다려야 한다는 것이다. 따라서, 폭포수 프로세스는 한 단계를 완전히 수행하고 다음 단계로 전환되므로 분석, 설계단계로 접어들수록 프로젝트에 대한 이해력이 향상되어 학습효과가 한번에 반영되는 현상을 보인다. 그러나 반복 프로세스는 요구사항, 분석, 설계, 코딩과 시험이 반복적으로 동시에 수행되므로 인해 개발과정이 경과함에 따라 여러 시점에 걸쳐 학습효과가 분산되어 해당 시점에서 수행되는 다른 여러 discipline들과 혼합되어 전체적으로 보았을 때는 학습효과가 거의 없는 것으로 보일 수 있기 때문일 것이다.

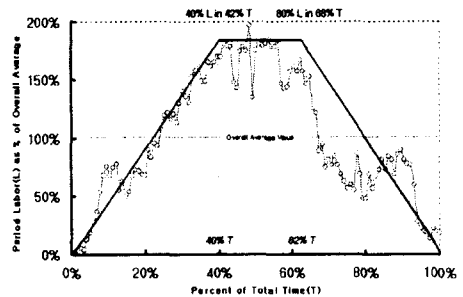
3.2 소프트웨어 개발주기 투입인력 판단

소프트웨어 분야에 대해서도 Wideman[9]이 제시한 (그림 4)와 같이 인력이 어느 시점에서 절정에 도달하여 얼마간 지속되는지를 분석하여 볼 필요가 있다. 반복 프로세스에 대해서는 (그림 18)에, 폭포수 프로세스를 적용한 실제

개발 프로젝트에 대해서는 (그림 19)에 제시하였다. (그림 19)는 Warburton[17]가 제시한 데이터로 폭포수 모델을 적용하여 구조적 방법으로 개발된 경우이다. 이 데이터는 소나와 화력 제어계통의 훈련/시뮬레이터 프로젝트로 요구사항 명세화, 설계, 코딩, 시험, 시스템 통합과 납품을 포함한 개발 생명주기 전 과정에 투입된 인력분포이다. 이 시스템은 116주의 기간 동안 1300.8명 (man-year)이 투입되었으며, 7일 이상의 MTTF (mean time to failure) 품질 수준이 달성되었을 때 개발이 종료되고 고객에게 납품되었다.



[그림 18] 반복 프로세스의 단위시간당 투입인력 프로파일 분석
[Fig. 18] Manpower distribution analysis per unit time of iterative process



[그림 19] 폭포수 프로세스의 단위시간당 투입인력 분포
[Fig. 19] Manpower distribution analysis per unit time of waterfall process

반복 프로세스는 최대 투입 인력이 평균 투입인력의 약 150%에 도달하여 토목건축분야의 160%와 유사한 경향을 나타내고 있다. 그러나 투입인력의 프로파일은 다른 형태를 취함을 알 수 있으며, 2가지 형태로 해석할 수 있다. 하나는 (그림 18)에서의 실선 부분으로 도입과 정련 #1까지는 계속적으로 투입되는 인력이 증가하다가 정련 #2가 시작되는 17% L in 25% T에서 시작하여 구축 #2가 끝나는 73% L in 66% T까지 절정에 도달하고 이후 전이 #2가 종료되는 시점까지 계속적으로 인력이 감소하는 형태를 취하는 경우이다. 다른 하나는 점선 부분으로 정련 #1 종료시점까지 계속적으로 인력이 증가하다가 정련 #2가 수행될 때 최대를 유지하고 구축 #1에서 감소하여 구축 #2와 #3에서 일정한 인력을 유지하고 전이 #1과 #2에서 점진적으로 감소하는 형태를 취한다고 할 수 있다. 이에 비해 폭포수 프로세스는 개발시간이 42% 진행된 시점에서 인력이 절정에 도달하여 68%까지 절정을 유지한 후 계속적으로 감소하는 형태를 취한다. 또한, 특이한 것은 최대 인력이 평균 인력의 180%까지 증가한다는 점이다.

결론적으로 소프트웨어 개발 프로세스를 폭포수나 반복 프로세스를 선택함에 상관없이 누적 인력 프로파일은 다른 분야의 프로젝트 수행 형태와 마찬가지로 하나의 S자 곡선 형태를 취한다. 단지 인력을 최대로 투입하는 시점이 프로젝트 특성을 반영하여 차이점을 갖고 있다.

3.3 반복 프로세스의 인력 프로파일 추정 모델

반복 프로세스를 따를 경우 (그림 2) 또는 (그림 17)과 같이 여러 discipline들이 동시에 중첩되어 수행됨으로 인해 Marasco[13]가 제시한 반복별 학습효과가 독립적이지 않고 통합된 형태로 하나의 S자 곡선을 따른다고 할 수 있다. 따라서, 반복 프로세스나 폭포수 프로세스 모두에 대한 누적 개발 노력, 개발비용과

프로젝트 완성도 분포는 식 (1)-(3)의 S자형 분포를 따르는 모델로 정의될 수 있다. 본 식들은 생물체의 성장 패턴을 모형화하는데 많이 사용되고 있는 성장곡선 (S자형 곡선)으로부터 도입되었다.

$$P_E = a \cdot \frac{(1 - e^{-bt})}{(1 + ce^{-bt})} \quad (1)$$

$$P_C = a \cdot \frac{(1 - e^{-bt})}{(1 + ce^{-bt})} \quad (2)$$

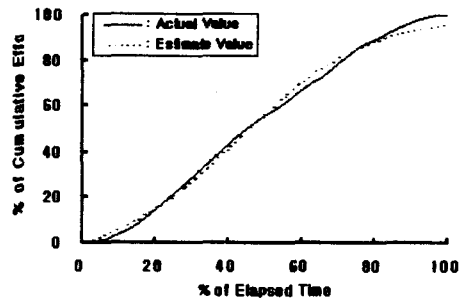
$$P_P = a \cdot \frac{(1 - e^{-bt})}{(1 + ce^{-bt})} \quad (3)$$

여기서 P_E 는 개발에 투입되는 노력, P_C 는 개발비용, P_P 는 프로젝트 완성도를 나타낸다. 또한, a 는 100%, b, c 는 상수이다.

단위시간당 개발노력, 개발비용과 프로젝트 완성도는 식 (1)-(3)을 미분하여 얻을 수 있다. 예로, 단위시간당 소요되는 노력은 식 (1)을 미분하여 식 (4)로 표현된다.

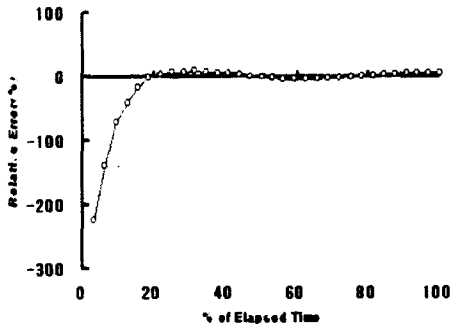
$$p_e = \frac{(abe^{-bt})}{(1 + ce^{-bt})} + a \cdot \frac{(1 - e^{-bt})}{(1 + ce^{-bt})^2} \cdot cbe^{-bt} \quad (4)$$

상용 통계 패키지를 이용하여 (그림 16) 데이터에 대한 식 (1)의 모수를 추정한 결과 $b=0.056$, $c=10.868$ 을 얻었다. 추정된 모수를 대입하여 실제 값과 추정된 값을 그린 결과는 (그림 20)에, 제안된 모델의 상대오차는 (그림 21)에, 분산분석 결과는 <표 1>에 제시하였다.



[그림 20] 반복 프로세스의 누적 투입인력
실측값과 추정 값 비교

[Fig. 20] Cumulative manpower actual and estimate value comparison of iterative process



[그림 21] 반복 프로세스의 누적 투입인력 추정 모델의 상대오차

[Fig. 21] Relative error of cumulative Manpower estimating model for iterative process

<표 1> 제안된 모델의 분산분석표
<Table 1> ANOVA table for proposed model

요인	자유도	제곱합	평균제곱	F	F(1,30;0.05)
회귀	1	35031.7379	35031.7379	6099.3780	3.E-36
잔차	30	172.3048	5.7435		
계	31	35204.0427			

그림에서 개발 초기단계의 약 20% 시점까지는 제안된 모델이 실제 투입 노력과 편차가 크게 나타나나 이후에는 거의 일치하는 결과를 얻을 수 있음을 알 수 있다. 따라서, 제안된 모델을 이용하여 반복 프로세스의 개발노력, 개발비용과 프로젝트 완성도에 대해 추정하고, 개발계획 작성시 의사결정 정보로 활용할 수 있을 것이다. 제안된 모델의 유의성 검정 결과 $F \geq F(0.05) = 6099.3780 \geq 3.E-36$ 로 회귀 분석이 유의함을 알 수 있다.

4. 결론 및 추후연구

일반적으로 프로젝트를 수행할 경우 투입되는 인력, 소요 비용과 프로젝트의 완성도는 불가사의하게도 투입인력의 학습효과가 반영되어

모두 S자 형태의 곡선을 따른다. 그러나 토목 건축 분야나 소프트웨어 개발의 폭포수 프로세스는 모두 하나의 주기 동안 다양한 단계를 거치며 한 단계가 완료된 후 다음 단계로 이동하는 순차적인 형태를 취한다.

이에 비해 소프트웨어 개발의 반복 프로세스는 하나의 주기동안 폭포수 프로세스를 중첩적으로 여러 번 반복 수행하는 특징을 갖고 있어 일반적으로 적용되는 S자 곡선을 적용할 수 있는지를 증명할 필요가 제기되었다. 이에 대해 여러 연구가 진행되었고, 다양한 결과를 나타내고 있어 일반적으로 적용할 수 있는 이론적인 결과를 얻지 못하였다. 일 예로 Marasco[13]은 반복 수에 비례하는 S자 곡선을 쌓아 놓은 곡선 형태를 따름을 제시하였다. 따라서, 본 논문은 반복 프로세스를 적용하여 소프트웨어를 개발할 경우 소요되는 누적 인력, 소요 비용과 프로젝트 완성도에 대한 프로파일이 하나의 S자 곡선을 따름을 증명하고, 이를 추정할 수 있는 모델도 제시하였다. 제시된 이론을 바탕으로 하여 반복 프로세스를 따르는 소프트웨어 개발계획 작성시 의사결정 정보로 활용할 수 있을 것이다.

본 논문의 이론을 증명하기 위해 적용된 인력 프로파일은 데이터 획득의 어려움으로 반복 프로세스를 따르는 실제 개발 프로젝트를 대상으로 하지 못하였다. 따라서, 실제 개발 프로젝트를 대상으로 누적 투입 인력과 프로젝트 완성도가 하나의 S자 형태를 따르는지에 대한 추가적인 실무 적용사례에 대한 연구가 추후 이루어 질 것이다.

참고문헌

[1] B. W. Boehm, "Software Engineering Economics," IEEE Trans. on Software Eng., Vol.10, No.1, pp.7-19, 1984.
 [2] B. Boehm, B. Clark, E. Horowitz, R. Modachy, R. Shelby, and C. Westland, "The COCOMO 2.0 Software Cost Estimation Model," USC Center

for Software Engineering, 1995.

[3] 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.

[4] M. Bradley, "Function Point Counting Practices Manual, Release 4.1," International Function Point Users Group (IFPUG), 1999.

[5] C. Symons, "COSMIC-FFP Measurement Manual, Version 2.2

(The COSMIC Implementation Guide for ISO/IEC 19761: 2003)," Common Software Measurement International Consortium, 2003.

[6] K. Ribu, "Estimating Object-oriented Software Projects with Use Cases," University of Oslo Department of Informatics, Master of Science Thesis, 2001.

[7] P. Kruchten, "Going Over the Waterfall with the RUP," The Rational Edge e-zine for the Rational Community, http://www.therationaledge.com/content/sep_01/t_waterfall_pk.html 2001.12.

[8] 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, 2002.10.

[9] M. Wideman, "Applying Resource Loading, Production & Learning Curves to Construction: A Pragmatic Approach," <http://www.maxwideman.com/papers/resource/>, 2001.

[10] G. Gendall, "The Mysterious S Curve," PROJECTmagazine, Vol. 4, Iss. 3,

<http://www.projectmagazine.com/v4i3scuve1.html>, 2003.5.

[11] 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.

[12] 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.

[13] J. Marasco, "A Physicist Looks at Project Progress," The Rational Edge e-zine for the Rational Community, http://www.therationaledge.com/content/nov_01/k_projectProcess_jm.jsp, 2001.11.

[14] J. Lattanze, "Managing Software Development: The Software Development Lifecycles," Carnegie Mellon University School of Computer Science, Pittsburg, 2000.

[15] A Rational Software Corporation White Paper, "Rational Unified Process: Best Practices for Software Development Teams," Rational Software Corporation, 1998.

[16] J. Marasco, "Iterative Development," The Rational Edge e-zine for the Rational Community, http://www.therationaledge.com/content/oct_02/k_iterativeDevelopment_jm.jsp, 2002.10.

[17] 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.

[18] E. 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]P. Kruchten, "From Waterfall to Iterative Development - A hallenging Transition for Project Managers," The Rational Edgee-zine for the Rational Community,http://www.therationaledge.com/content/dec_00/m_iterative.hml, 2000.10.