

프로젝트 인도를 그룹 분할 방법을 이용한 소프트웨어 개발노력 추정

이 상 운[†] · 노 명 옥^{††} · 이 부 권^{†††}

요 약

소프트웨어 개발시 중요하게 제기되는 문제점으로 소프트웨어 생명주기의 초기단계에서 개발에 투입될 노력과 비용을 추정하는 능력이다. 측정된 소프트웨어 규모인 기능점수 (FP, Function Point)로부터 프로젝트 개발에 소요되는 노력과 비용을 추정하는 회귀모델들이 제안되었다. 그러나 이들 제안된 모델들은 사용한 표본의 크기가 작고, 과거에 개발된 프로젝트를 대상으로 하고 있다. 1990년대에 개발된 789개 소프트웨어 개발 프로젝트들에 투입된 개발노력에 이들 모델을 적용한 결과 결정계수 R^2 가 0.53 이하로 데이터의 변동을 설명하였다. 본 논문은 프로젝트 인도를 (PDR)을 이용하여 동질성을 갖는 그룹으로 분류하여 개발노력을 추정하는 모델을 제안하였다. 제안된 모델은 랜덤한 잔차 분포를 갖고 있고, 대부분의 PDR 범위에서 0.99 이상의 결정계수로 데이터의 변동을 설명하였다.

Software Development Effort Estimation Using Partition of Project Delivery Rate Group

Sang-Un Lee[†] · Myeong-Ok No^{††} · Bu-Kwon Lee^{†††}

ABSTRACT

The main issue in software development is the ability of software project effort and cost estimation in the early phase of software life cycle. The regression models for project effort and cost estimation are presented by function point that is a software size. The data sets used to conduct previous studies are often small and not too recent. Applying these models to 789 project data developed from 1990; the models only explain fewer than 0.53 R^2 (Coefficient of determination) of the data variation. Homogeneous group in accordance with project delivery rate (PDR) divides the data sets. Then this paper presents general effort estimation models using project delivery rate. The presented model has a random distribution of residuals and explains more than 0.99 R^2 of data variation in most of PDR ranges.

키워드 : 개발노력 추정 (Development Effort), 소프트웨어 규모 (Software Size), 기능점수 (Function Point), 프로젝트 인도를 (Project Delivery Rate), 회귀분석 (Regression Analysis)

1. 서 론

소프트웨어 개발시 중요하게 제기되는 사항은 소프트웨어 생명주기의 초기단계에서 개발에 투입될 노력과 비용을 추정하는 능력이다. 소프트웨어 측정분야는 30년 이상 수많은 연구가 있어왔으나 소프트웨어 개발노력과 비용에 영향을 미치는 다양한 속성들과 이들간의 불명확한 관계로 아직까지 구체적인 모델이 없는 실정이다. 실제로 소프트웨어 개발노력과 비용을 추정하려면 소프트웨어의 규모를 측정해야 한다. 소프트웨어 규모를 측정하기 위한 척도로 가장 많이 사용되

고 있는 방법들 중에 LOC (Line Of Codes)와 FPA (Function Point Analysis)가 있다. 길이로서 소프트웨어의 규모를 측정하기 위해 최초로 개발된 소프트웨어 척도 중 하나가 LOC (Line Of Code)이다[1, 2]. LOC는 언어에 따라 다르게 측정되며, 요구분석 또는 설계단계에서 정확한 LOC 추정이 어려운 문제점을 갖고 있다. 이에 비해, FPA는 사용자에게 양도될 시스템의 기능에 기초하여 소프트웨어 시스템의 규모와 복잡도를 정량화하는 방법으로 소프트웨어 프로젝트를 개발하기 위해 사용되는 언어 또는 도구와 독립적이며, 개발 생명주기의 초기단계인 요구분석 단계에서 측정 가능한 장점이 있어 LOC를 사용할 때의 문제점을 극복할 수 있는 접근법이다[2].

소프트웨어 규모인 기능점수 (FP, Function Point)에 따른 소프트웨어 개발노력 (Effort)을 추정하기 위한 모델들[2-7]은 회귀분석방법을 적용하였다. 그러나 이들 제안된 모델들이 사

* 본 연구는 2001년도 경상대학교 연구년제 연구교수 연구지원비에 의하여 수행되었음.

† 정 회 원 : 국방품질관리소 항공전자장비 및 소프트웨어 품질보증 담당

†† 정 회 원 : 경상대학교 대학원 컴퓨터학과

††† 정 회 원 : 경상대학교 컴퓨터학과 교수

논문접수 : 2001년 9월 21일, 심사완료 : 2001년 12월 12일

용한 표본은 크기가 작고, 대부분이 특정 업체에서 개발된 프로젝트를 대상으로 하고 있어 모든 소프트웨어 프로젝트들에 일반적으로 적용할 수 있는 모델이 없는 실정이다. 또한 1990년대 20여개 국에서 다양한 개발환경하에서 개발된 789개 소프트웨어 프로젝트에 관한 데이터베이스를 갖고 있는 ISBSG (International Software Benchmarking Standards Group) Benchmark Release 6[8]를 이용하여 기존 제안된 모델들을 적용해본 결과 결정계수 R^2 가 0.53 이하로 좋은 모델이 되지 못하였다. 즉, 실제 소프트웨어 개발에 투입된 개발노력의 53% 이하 만을 설명할 수 있는 능력을 갖는 모델이었다.

다양한 환경과 조건에서 개발된 전체 데이터를 대상으로 개발노력을 추정하는 모델의 성능 한계를 극복하기 위해, 보다 동질성을 갖는 그룹으로 데이터들을 세분화시켜 개발노력을 추정하는 모델을 제시한다. 제시된 모델은 동질성을 갖는 그룹으로 세분화하는 기준으로 프로젝트 인도율(PDR , Project Delivery Rate)을 적용하였다. 제안된 모델은 ISBSG가 보유하고 있는 789개의 실제 수행된 프로젝트 데이터베이스를 대상으로 모델이 유도되었다. 따라서, 표본의 크기가 상당히 크고, 다양한 개발환경 하에서 개발된 시스템들에 적합한 모델로서, 모든 소프트웨어 프로젝트들에 일반적으로 적용이 가능한 개발노력 추정 모델이 될 수 있다.

2장에서는 개발노력 추정과 관련된 연구 및 문제점을, 3장에서는 기능점수를 이용하여 개발노력을 추정한 모델들의 성능을 평가하였다. 4장에서는 소프트웨어 개발노력 추정을 위해 일반적으로 적용 가능한 회귀 모델을 제시한다. 또한, 제안된 모델과 기존의 회귀 모델의 결과를 비교하여 제안된 모델이 가장 좋은 추정 결과를 얻음을 보인다.

2. 관련 연구 및 연구배경

2.1 LOC 이용 개발노력 추정 모델

LOC를 이용한 소프트웨어 개발노력 추정 모델은 식 (1) 또는 식 (2)의 형태를 취한다.

$$E = a + b \times KLOC^c \quad (1)$$

$$E = a \cdot KLOC^b \quad (2)$$

식 (1)과 식 (2)에서 E 는 예측된 개발노력으로 Man-Months로 측정되고, a , b , c 는 상수이며, $KLOC$ 는 최종 코딩된 소프트웨어의 추정된 KLOC(Thousands of Line Of Code)의 수이다. 예로 Bailey-Basili 모델은 $E=5.5+0.73 \times (KLOC)^{1.16}$, Walston-Felix 모델은 $E=5.2 \times (KLOC)^{0.91}$, Boehm의 Complex 모델은 $E=2.8 \times (KLOC)^{1.20}$ 이다[2]. 그러나 소프트웨어 규모 측정 단위로 LOC를 사용할 경우 발생하는 문제점은 다음과 같다.

- LOC에 대한 일반적으로 받아들일 수 있는 정확한 정의가

부족하다. Jones[9]는 라인을 계산하는 방법에 대해 11개의 변형된 방법을 확인했으며, 라인 계산 방법에 따라 약 500%의 불확실성을 가지고 있음을 지적했다.

- LOC는 언어에 종속되어 있다[10]. 다른 언어를 사용하여 개발된 프로젝트들에 대해 상호간에 직접 비교하기가 불가능하다. 예를 들면, 고차원 언어의 라인당 투입되는 시간은 저차원 언어보다 많다. 또한 동일한 기능을 제공하기 위해 고차원 언어에 대해 보다 적은 LOC가 요구됨은 자명한 사실이다.
- 요구분석 또는 설계단계에서 정확한 LOC 추정이 어려우며, 코딩이 종료된 후 측정이 가능하다[11]. 그러나, 소프트웨어 개발노력은 프로젝트 착수 초기에 보다 정확히 추정하여 사업관리 측면이나 자원 재분배 측면에서 활용할 수 있어야만 한다.
- 특정한 하나의 관점인 길이만을 고려하고 있으며, 소프트웨어의 기능성 또는 복잡도를 고려하지 못해 소프트웨어 특성을 정확히 반영하지 못한다.

2.2 기능점수 이용 소프트웨어 개발노력 추정모델

기능점수 FP 척도는 소프트웨어 프로젝트 규모 추정, 생산성과 품질 평가에서 일관성과 객관성을 제공하는 구체적으로 공인되고 표준화된 척도로 프로젝트 계획 중에 소요 비용과 일정을 추정하는데 유용하게 사용되며, LOC에 비해 소프트웨어의 개발 초기 단계인 요구분석 단계에서 측정이 가능한 장점을 갖고 있다. 기능점수 FP 규모에 따른 개발노력 E 추정에 관한 연구로서, Albrecht et al[5]은 IBM Data Processing Services에서 개발된 24개 응용 프로그램에 대해 식 (3)의 단순 회귀모델을 취함을 연구하였고, Matson et al.[2]은 이들 데이터에 대해 식 (4)의 비선형 형태를 취하는 것이 보다 정확한 추정을 할 수 있음을 보였다.

$$E = -13.39 + 0.0545 FP \quad (3)$$

$$\sqrt{E} = 1.000 + 0.00468 FP \quad (4)$$

Kemerer[7]는 ABC 회사에서 개발된 15개의 소프트웨어 프로젝트에 대해 식 (5)~식 (7)의 단순회귀직선모델과 곡선회귀 모델을 제시하였다.

$$E = -121.57 + 0.3411 FP \quad (5)$$

$$E = -69.13 + 0.723FP - 8.054 \times 10^{-4} FP^2 + 3.073 \times 10^{-7} FP^3 \quad (6)$$

$$E = 60.62 + 7.728 \times 10^{-8} FP^3 \quad (7)$$

그러나 이들 모델들은 표본의 개수가 작고 특정 업체에서 개발된 프로젝트에 대해 얻어진 모델로 다양한 프로젝트와 개발환경을 반영하지 못하고 있다. 또한, Matson et al.[2]은 대형업체로부터 획득된 104개의 프로젝트를 이용해 모델을 유

도하였다. 이들 업체들은 잘 훈련되고, 경험이 있는 시스템 개발 인력과 현존하는 소프트웨어 개발 방법론과 도구를 사용하고 있으며, 또한 수년간 기능점수기법을 사용하여 생산성과 품질을 측정할 경험을 갖고 있었다. 이들 프로젝트들은 다양한 응용 분야, 다양한 언어들로 구성되어 있다. Matson et al.[2]은 이와 같이 비교적 큰 표본을 사용해 식 (8)과 식 (9)의 모델을 제시하였다.

$$E = 585.7 + 15.12FP \quad (8)$$

$$\ln(E) = 2.51 + 1.00\ln(FP) \quad (9)$$

이들 참고 문헌에서의 연구결과를 종합한 결과와 모델의 성능은 <표 1>에 제시되어 있다.

<표 1> FP를 이용한 개발노력 추정 모델

표본수	개발노력 추정 모델		모델성능 (R^2)
24개	Albrecht et al.	$E = -13.39 + 0.0545 \cdot FP$	0.874
	Matson et al.	$\sqrt{E} = 1.000 + 0.00468 \cdot FP$	0.899
15개	Kemerer	$E = -121.57 + 0.3411 \cdot FP$	0.553
		$E = -69.13 + 0.723FP - 8.054 \times 10^{-4}FP^2 + 3.073 \times 10^{-7}FP^3$	0.884
		$E = 60.62 + 7.728 \times 10^{-8}FP^3$	0.847
104개	Matson et al.	$E = 585.7 + 15.12FP$	0.652
		$\ln(E) = 2.51 + 1.00\ln(FP)$	0.534

표에서 모델의 성능을 R^2 로 제시되어 있다. 다른 여러 가지 모델들을 비교하는데 있어, 어떤 의미있는 척도로서 모델의 추정 정확도를 평가하는 것이 필요하다. 회귀분석의 경우 회귀직선에 의해 종속변수가 설명되는 정도를 결정계수(Coefficient of determination, R^2)라 한다. 종속변수(E)의 값은 독립변수(FP)에 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.

표에서 알 수 있듯이, 기능점수 FP 규모에 따른 개발노력 E 추정을 위한 기존 모델들은 특정 프로젝트뿐만 아니라 표본의 크기에 따라서도 적합한 모델이 다르며, 모든 프로젝트에 공통적으로 적용할 수 있는 일반적인 모델(Universal Model)이 없음을 알 수 있다.

기능점수와는 다른 측면에서, 소프트웨어 구성요소인 입력, 출력, 조회와 파일의 개수를 이용하여 개발노력을 추정하는 연구로는 이상운[12], 이상운 et al.[13]이 있다.

3. 기능점수 이용 개발노력 추정 모델 평가

ISBSG Benchmark Release 6[8] 데이터베이스는 1990년대에 다양한 나라에서 개발된 789개 프로젝트들에 대한 다양

한 언어, 개발기법 및 적용분야 등을 갖고 있는 방대한 데이터베이스이다.

<표 1>의 모델들을 종합해보면 FP 를 이용한 E 추정 모델, FP 를 이용한 \sqrt{E} 추정 모델과 $\ln(FP)$ 를 이용한 $\ln(E)$ 추정 모델임을 알 수 있다. 기존에 제안된 모델들을 평가하기 위해, ISBSG Benchmark Release 6[8]의 789개 프로젝트에 대해 개발노력 E vs. 기능점수 ZFP , \sqrt{E} vs. FP 와 $\ln(E)$ vs $\ln(FP)$ 관계에 대한 회귀분석을 수행한 결과는 (그림 1)~(그림 3)에 각각 제시되어 있다.

(그림 3) $\ln(E)$ 와 $\ln(FP)$ 와의 관계 선형회귀

E vs. FP 와 \sqrt{E} vs. FP 관계의 경우, FP 가 작은 프로젝트에서 큰 프로젝트 전반에 걸쳐 이상점들이 다수 발견되어

선형으로 적합이 불가능함을 알 수 있다. 이들 제안된 모델들과 더불어 선형(곡선)과 비선형(누승, 지수, 로그) 회귀분석을 수행한 결과 얻은 모델은 <표 2>에, 모델의 성능은 <표 3>에 제시되어 있다.

누승 회귀모델의 정확도가 52.84%로 가장 좋은 결과를 얻었으며, 이 결과는, 기존 제안 모델에서 최고 성능을 나타낸 자연로그인 $\ln(E)$ 추정 모델의 성능(결정계수 $R^2 = 0.5273$) 보다 0.11% 향상된 결과를 보였다. 그러나 이 모델은 다양한 환경과 조건하에서 개발된 대용량의 전체 데이터를 이용하는 경우로, 성능이 낮은 단점을 갖고 있다. 따라서, 다양한 환경과 조건에서 개발된 데이터들을 보다 동질성을 갖는 그룹으로 분할하여 개발노력 추정 성능이 향상된 모델 제시가 필요하다.

<표 2> 개발노력 추정 회귀모델 적용 결과

구분	회귀 구분	개발노력 추정모델
기존 제안 모델	단순(직선)	$E = 13.3848 FP + 975.4649$
	근	$\sqrt{E} = 0.0365 FP + 45.5777$
	자연로그	$\ln(E) = 0.9351 \ln(FP) + 2.4858$
추가 실험 모델	곡선	$E = -0.0011 FP^2 + 22.8833 FP - 3483.5744$
	누승	$E = 11.9167 FP^{0.9367}$
	지수	$E = 1706.2142 e^{0.0007 FP}$
	로그	$E = 11071.1977 \ln(FP) - 54292.3150$

<표 3> 개발노력 추정 회귀모델 성능 비교

여 모델의 성능을 향상시킬 수 있었다.

본 논문은 전체 데이터들을 동질성을 갖는 그룹들로 세분화하는 기준으로 FP 1개를 개발하는데 소요되는 노력의 규모를 고려하였다. 이 속성은 개발업체의 개발능력 수준에 영향을 받는다고 할 수 있다. ISBSG Benchmark Release 6[8]는 소프트웨어 기능을 사용자에게 양도하는 비율인 프로젝트 인도율(Project Delivery Rate, PDR)을 개발노력(시간)의 계수로 측정하였다. 즉, 프로젝트 개발에 소요된 전체 노력(E)를 사용자에게 인도된 소프트웨어의 기능 규모(FP)로 나눈 값 즉, E/FP 로 정의하였다. 이는 기능점수 FP 1개를 개발하는데 소요되는 개발노력의 규모이다.

ISBSG Benchmark Release 6[8]의 789개 프로젝트에 대한 PDR 값은 $0.0 < PDR \leq 268.0340$ 의 범위를 갖고 있었다. 개발업체의 생산성 수준에 따라 PDR 값에 차이가 발생할 수 있다. 따라서, PDR 값을 보다 세분화시킨 그룹으로 분류하였다. $0.0 < PDR \leq 10.0$ 은 1단위로, $10.0 < PDR \leq 20.0$ 은 2단위로 증가시키며 서브 그룹을 형성하였고, $20.0 < PDR \leq 40.0$ 은 10단위로, 나머지는 $40.0 < PDR$ 로 단일 그룹으로 하였다. 이와 같이 PDR 범위를 구분한 이유는 PDR 의 값이 적은 프로젝트들은 표본수가 많고, PDR 의 값이 큰 프로젝트들의 표본수는 적기 때문이다.

PDR 각각의 그룹에 대한 개발노력을 추정하는 모델을 개발하여보자. 먼저, 기능점수 FP 에 따른 개발노력 E 변화에 대한 산점도를 그리고 선형(단순, 곡선)과 비선형(누승, 지수와 로그) 회귀분석을 수행한 결과는 <표 4>에 제시되어 있다. 5개 모델들 중에 최고의 성능을 나타내는 모델을 <표 5>에 제시하였다.

<표 4> PDR 별 회귀 모델 성능

4. 프로젝트 인도율 이용 개발노력 추정 모델

4.1 모델 제안

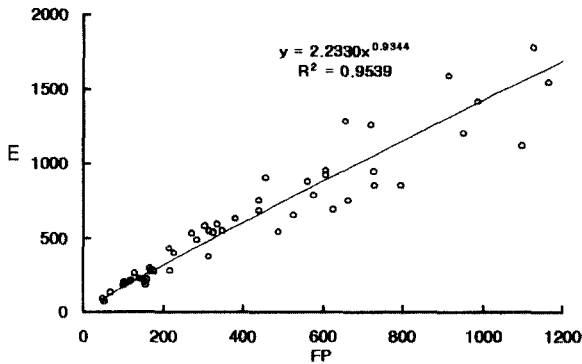
프로젝트 개발노력에 영향을 미치는 속성들로 ISBSG Benchmark Release 6[8]는 개발형태(신규개발, 재개발, 성능개량), 개발 플랫폼(Main Frame, Mid-Range, Personnel Computer), 언어 형태, DBMS 사용 여부, 개발기법 사용 여부, 프로그램 언어, 적용분야, 응용 형태 등을 제시하고 있다.

보다 적합한 개발노력 추정 모델을 유도하기 위해 다양한 환경과 조건하에서 개발된 전체 데이터를 사용하는 대신 보다 동질성을 갖는 그룹으로 세분화하여 개발노력을 추정하는 모델을 유도하여 보자. Oligny et al.[14, 15]는 개발노력의 규모에 따라 개발기간을 추정하는 모델을 제안함에 있어, 전체 데이터들을 개발 플랫폼별로 세분화시켜 개발기간을 추정하

예로, $1.0 < PDR \leq 2.0$ 에 대해 기능점수 FP 에 따른 개발노력 E 추정 모델은 (그림 4)에 제시되어 있으며, 분산분석은 <표 6>에, 제시되어 있다. 제안된 모델의 유의성 검정결과 $F > F(\alpha) = 448.1676 > 1.29E-26$ 로 회귀 분석이 유의함을 알 수 있다.

<표 5> PDR별 최고 성능 회귀 모델

PDR	최고 성능 회귀 모델
$0.0 < PDR \leq 1.0$	$E = 0.3210FP^{1.0666}$
$1.0 < PDR \leq 2.0$	$E = 2.2330FP^{0.9344}$
$2.0 < PDR \leq 3.0$	$E = 2.4717FP^{1.0022}$
$3.0 < PDR \leq 4.0$	$E = -0.0005FP^2 + 4.0141FP - 309.4885$
$4.0 < PDR \leq 5.0$	$E = 4.7950FP^{0.9886}$
$5.0 < PDR \leq 6.0$	$E = 5.4866FP^{1.0034}$
$6.0 < PDR \leq 7.0$	$E = 0.0001FP^2 + 6.6019FP - 15.8775$
$7.0 < PDR \leq 8.0$	$E = 0.0001FP^2 + 7.2914FP + 42.8385$
$8.0 < PDR \leq 9.0$	$E = -0.0001FP^2 + 8.5825FP - 26.4959$
$9.0 < PDR \leq 10.0$	$E = 9.2256FP^{1.0049}$
$10.0 < PDR \leq 12.0$	$E = 11.0035FP^{0.9990}$
$12.0 < PDR \leq 14.0$	$E = 0.0002FP^2 + 12.6347FP + 85.6458$
$14.0 < PDR \leq 16.0$	$E = 0.0001FP^2 + 14.6651FP + 128.6060$
$16.0 < PDR \leq 18.0$	$E = 17.1447FP^{0.9989}$
$18.0 < PDR \leq 20.0$	$E = -0.0008FP^2 + 19.8351FP - 126.1199$
$20.0 < PDR \leq 30.0$	$E = 25.5613FP^{0.9856}$
$30.0 < PDR \leq 40.0$	$E = 32.4073FP^{1.0085}$
$40.0 < PDR$	$E = 27.5804FP^{1.1482}$



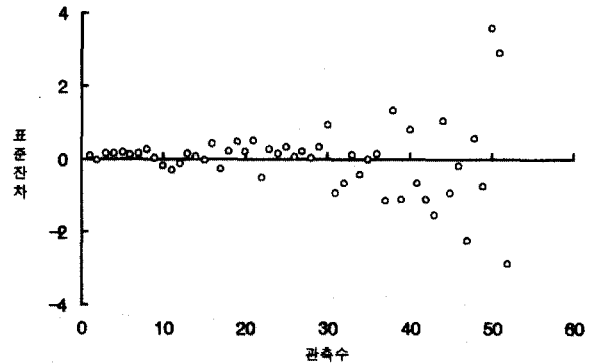
(그림 4) 기능점수 규모별 개발노력 추정 ($1.0 < PDR \leq 2.0$)

<표 6> 분산분석

요인	제곱합	자유도	평균제곱	F	F(α)
회귀	23158817	1	23158817	448.1676	1.29E-26
잔차	1119217	50	22384.34		
계	24278035	51			

다음으로 제안된 모델의 적합성을 검증하기 위해 잔차분석을 하여보자. 잔차(Residual)는 실제 값과 추정된 값과의 차이로, 잔차분석은 단순회귀모델에서 설정한 등분산성, 독립성, 정규성과 직선관계의 가정이 옳은가를 검토할 때 가장 많이 사용되는 방법이다. 따라서, 잔차가 어떤 일정한 형태를

취하지 않고 랜덤하게 분산되어 있는 경우가 좋은 모델이 된다. 그러나 (그림 5)의 잔차분석 결과 잔차들이 랜덤한 결과를 보이지 않고 관측수가 증가함에 따라 표준잔차들이 넓은 폭을 가지면서 흩어져 있어 분산이 증가함으로써 등분산성의 가정이 옳지 않아 좋은 모델로 선정할 수 없다. 따라서, <표 5>에 제안된 모델들은 적합하지 않아 쓸모가 없다.



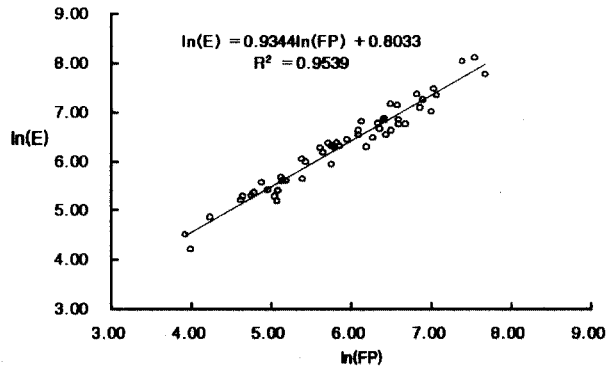
(그림 5) 잔차분석

이 경우, 간단한 변수 변환(Variable Transformation)을 통해 직선관계로 변형시킬 수 있으며, 직선회귀 방법을 사용하여 분석한 후 변환되기 이전의 변수간의 관계를 규명하는 경우도 있다[16]. 본 논문은 이러한 방법을 이용하여 개발노력을 추정하는 일반화된 모델을 제시한다. FP 와 E 에 대해 자연로그를 취한 변수변환을 통해 위에서 제안된 모델들을 선형으로 변환시킬 수 있다. $\ln(FP)$ 규모에 따른 $\ln(E)$ 를 PDR 범위별로 추정한 모델의 회귀분석 결과는 <표 7>에 제시되어 있다.

<표 7> PDR별 개발노력 추정 모델

PDR	모 델	성 능
$0.0 < PDR \leq 1.0$	$\ln(E) = 1.0556 \ln(FP) - 1.1362$	0.7213
$1.0 < PDR \leq 2.0$	$\ln(E) = 0.9344 \ln(FP) + 0.8033$	0.9539
$2.0 < PDR \leq 3.0$	$\ln(E) = 1.0002 \ln(FP) + 0.9049$	0.9791
$3.0 < PDR \leq 4.0$	$\ln(E) = 0.9984 \ln(FP) + 1.2596$	0.9958
$4.0 < PDR \leq 5.0$	$\ln(E) = 0.9886 \ln(FP) + 1.5676$	0.9965
$5.0 < PDR \leq 6.0$	$\ln(E) = 1.0034 \ln(FP) + 1.7023$	0.9983
$6.0 < PDR \leq 7.0$	$\ln(E) = 1.0088 \ln(FP) + 1.8319$	0.9985
$7.0 < PDR \leq 8.0$	$\ln(E) = 0.9957 \ln(FP) + 2.0448$	0.9989
$8.0 < PDR \leq 9.0$	$\ln(E) = 0.9903 \ln(FP) + 2.1873$	0.9989
$9.0 < PDR \leq 10.0$	$\ln(E) = 1.0049 \ln(FP) + 2.2220$	0.9993
$10.0 < PDR \leq 12.0$	$\ln(E) = 0.9990 \ln(FP) + 2.3982$	0.9974
$12.0 < PDR \leq 14.0$	$\ln(E) = 1.0059 \ln(FP) + 2.5356$	0.9985
$14.0 < PDR \leq 16.0$	$\ln(E) = 0.9973 \ln(FP) + 2.7284$	0.9989
$16.0 < PDR \leq 18.0$	$\ln(E) = 0.9989 \ln(FP) + 2.8417$	0.9986
$18.0 < PDR \leq 20.0$	$\ln(E) = 0.9982 \ln(FP) + 2.9552$	0.9988
$20.0 < PDR \leq 30.0$	$\ln(E) = 0.9856 \ln(FP) + 3.2411$	0.9939
$30.0 < PDR \leq 40.0$	$\ln(E) = 1.0066 \ln(FP) + 3.4956$	0.9939
$40.0 < PDR$	$\ln(E) = 1.1482 \ln(FP) + 3.3171$	0.9180

모델의 적합성을 평가하기 위해, $1.0 < PDR \leq 2.0$ 에 대해 기능점수 $\ln(FP)$ 에 따른 개발노력 $\ln(E)$ 추정 모델은 (그림 6)에 제시되어 있으며, 분산분석은 <표 8>에 제시되어 있다. 제안된 모델의 유의성 검정결과 $F \geq F(\alpha) = 1035.507 > 4.4E-35$ 으로 회귀 분석이 유의함을 알 수 있다.

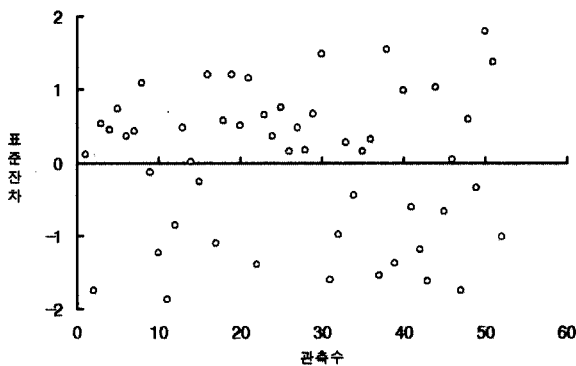


(그림 6) 기능점수 규모별 개발노력 추정 ($1.0 < PDR \leq 2.0$)

<표 8> 분산분석

요 인	제곱합	자유도	평균제곱	F	F (α)
회 귀	36.4026	1	36.4026	1035.507	4.4E-35
잔 차	1.7577	50	0.0352		
계	38.1603	51			

(그림 7)의 잔차분석 결과 잔차들이 일정한 경향을 보이지 않고 랜덤한 결과를 보여 좋은 모델로 판명되었다.



(그림 7) 변수변환으로 유도된 모델의 잔차분석

4.2 모델 평가

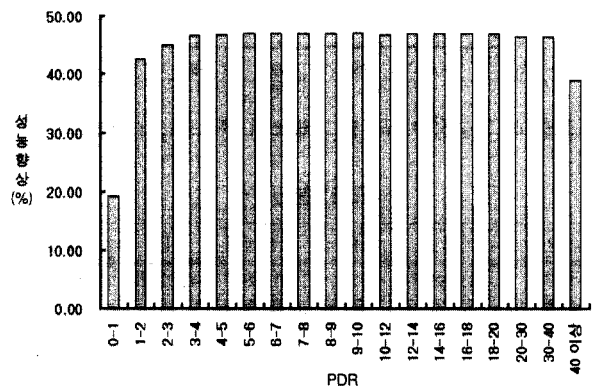
PDR 을 고려하지 않고 전체 데이터를 대상으로 하여 개발노력을 추정한 경우의 최고 성능을 나타내는 <표 3>의 누승 회귀모델과 PDR 그룹으로 분류하여 제안된 <표 7>의 모델 성능을 비교한 결과는 <표 9>와 (그림 8)에 제시되어 있다.

PDR 을 고려한 경우의 최고 성능을 갖는 회귀모델을 적용할 경우, PDR 을 고려하지 않은 경우의 누승 회귀모델 보다

PDR 범위 전체에 대해 성능이 향상되는 결과를 얻었다. 본 제안 모델은 기존 회귀 모델의 최고 성능에 비해 약 46%(99%~53%)의 성능이 향상된 모델이며, 다양한 개발환경과 조건에서 개발된 대용량의 최근 데이터로부터 유도된 모델로 개발업체의 PDR 수준만 알 수 있으면 본 제안 모델을 활용하여 개발초기에 프로젝트 개발에 소요되는 개발노력을 추정하고자 할 때 일반적으로 적용이 가능하다.

<표 9> 모델 성능 비교

PDR	모델성능 (R^2)		성능비교 (① - ②)
	PDR 을 고려한 모델 (①)	PDR 을 고려하지 않은 모델 (②)	
$0.0 < PDR \leq 1.0$	0.7213	0.5284	+0.1929
$1.0 < PDR \leq 2.0$	0.9539		+0.4256
$2.0 < PDR \leq 3.0$	0.9791		+0.4507
$3.0 < PDR \leq 4.0$	0.9958		+0.4674
$4.0 < PDR \leq 5.0$	0.9965		+0.4681
$5.0 < PDR \leq 6.0$	0.9983		+0.4699
$6.0 < PDR \leq 7.0$	0.9985		+0.4701
$7.0 < PDR \leq 8.0$	0.9989		+0.4705
$8.0 < PDR \leq 9.0$	0.9989		+0.4705
$9.0 < PDR \leq 10.0$	0.9993		+0.4709
$10.0 < PDR \leq 12.0$	0.9974		+0.4690
$12.0 < PDR \leq 14.0$	0.9985		+0.4701
$14.0 < PDR \leq 16.0$	0.9989		+0.4705
$16.0 < PDR \leq 18.0$	0.9986		+0.4702
$18.0 < PDR \leq 20.0$	0.9988		+0.4704
$20.0 < PDR \leq 30.0$	0.9939		+0.4655
$30.0 < PDR \leq 40.0$	0.9939	+0.4655	
$40.0 < PDR$	0.9180	+0.3896	



(그림 8) 모델 성능향상

5. 결론 및 향후 연구과제

기능점수 FP 를 이용해 개발노력 E 를 추정하는 기존 제안

된 모델들은 사용된 표본의 개수가 적고, 특정 업체에서 개발되었던 프로젝트들에만 적용된 관계로 다양한 형태의 개발환경과 개발방법 등을 적용한 프로젝트들에 적합하지 않는 문제점을 갖고 있었다. 이의 근거로, ISBSG Benchmark Release 6[8]의 최근 다양한 개발환경과 방법론으로 개발된 789개 프로젝트에 적용한 결과 모델의 성능이 현저히 저하된 결과를 얻었다. 따라서, 본 논문은 표본의 크기가 상당히 크고, 다양한 최근의 소프트웨어 개발 환경과 방법론을 적용한 프로젝트에 적합한 소프트웨어 개발노력을 추정하는 모델을 제안하였다.

기능점수 FP와 개발노력 E 사이의 관계가 명확한 산점도를 얻기 위해 먼저 동질성을 갖는 그룹을 형성하였으며, 이를 위해 프로젝트 인도를 PDR을 고려하였다. PDR 그룹별로 회귀분석을 통해 모델을 제안하였다. 그러나 잔차분석 결과 제안된 모델들이 좋지 않은 결과를 나타내어, 다시 자연로그 형태의 변수변환을 통해 적합한 모델들을 얻었다. 본 제안 모델의 성능이 기존의 가장 성능이 좋은 회귀분석 모델보다 약 46% 향상된 99% 이상의 정확성을 가지고 개발노력을 추정할 수 있었다. 따라서, 업체의 프로젝트 인도를 PDR 값과 개발되는 소프트웨어 프로젝트의 규모인 기능점수 FP 값만 계산되면, 본 모델을 이용해 최적의 소프트웨어 개발노력을 추정하여 사업관리와 개발인력의 배분에 적용이 가능하다.

본 연구는 생산성 척도인 PDR만을 고려하여 동질성을 갖는 그룹으로 세분화하여 개발노력을 추정하는 능력을 향상시키는 모델을 제안하였다. 그러나 프로젝트의 특성, 적용분야, 또는 개발방법 등에 의해 투입되는 개발노력의 규모에 영향을 미칠 수 있을 것이다. 따라서, 추후 이 분야에 대한 연구를 수행할 것이다.

참 고 문 헌

[1] L. A. Laranjeira, "Software Size Estimation of Object-Oriented Systems," IEEE Trans. on Software Eng., Vol.16, pp.64-71, 1990.
 [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] A. J. Albrecht, "Measuring Applications Development Productivity," Proceedings of IBM Application Dev., Joint SHARE/GUIDE Symposium, Monterey, CA., pp.83-92, 1979.
 [4] 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.

[5] T. Demarco, "Controlling Software Projects: Management Measurement & Estimation," New York: Yourdon Press, 1982.
 [6] C. F. Kemerer, "An Empirical Validation of Software Cost Estimation Models," Communication ACM, Vol.30, No.5, pp.416-429, 1987.
 [7] C. F. Keremer, "Reliability of Functional Point Measurement - A Field Experiment," Communications of ACM, 1993.
 [8] ISBSG, "Worldwide Software Development - The Benchmark Release 6," Victoria, Australia International Software Benchmarking Standards Group, 2000.
 [9] C. Jones, "Programming Productivity," New York, McGraw-Hill, 1986.
 [10] G. C. Low and D. R. Jeffery, "Function Points in the Estimation and Evaluation of the Software Process," IEEE Trans. on Software Eng., Vol.16, pp.64-71, 1990.
 [11] R. D. Emrick, "In Search of a Better Metric for Measuring Productivity of Application Development," Int. Function Point Users Group Conf. Proc., 1987.
 [12] 이상운, "신경망을 이용한 소프트웨어 개발노력 추정", 정보처리학회논문지, 제8-D권 제3호, pp.241-246, 2001.
 [13] 이상운, 박영목, 박재홍, "RBF 망을 이용한 소프트웨어 개발노력 추정", 정보처리학회논문지, 제8-D권 제5호, pp.581-586, 2001.
 [14] S. Oigny, P. Bourque and A. Abran, "An Empirical Assessment of Project Duration Models in Software Engineering," In The Eight European Software Control and Metrics Conference(ESCOM'97), Berlin Germany, 1997.
 [15] S. Oigny, P. Bourque, A. Abran, and B. Fournier, "Exploring the Relation Between Effort and Duration in Software Engineering Projects," World Computer Congress 2000, August 21-25, Beijing, China, pp.175-178, 2000.
 [16] 김우철, et al., "현대 통계학", 영지출판사, 1994.

이 상 운

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 매트릭스, 신경망, 뉴로-퍼지

노명옥

e-mail : nomeok@Lycos.co.kr

1989년 경상대학교 전산통계학과 학사

1996년 고려대학교 교육대학원 전산교육
전공 석사

1998년~현재 경상대학교 대학원 컴퓨터과
학과 박사과정

1989년~현재 산청여자고등학교 교사

관심분야 : 정보통신, 소프트웨어 공학

이부권

e-mail : bklee@nongae.gsnu.ac.kr

1972년 경상대학교 농경제학과 학사

1978년 미시건 주립대학 대학원 시스템공학
전공 석사

2001년 경남대학교 대학원 컴퓨터공학과 박사

1980년~현재 경상대학교 컴퓨터공학과 교수

관심분야 : 시뮬레이션, 멀티미디어 통신, 소프트웨어 공학