

소프트웨어 개발기간 추정 모델

(A Software Estimating Model for Development Period)

이 상 운 [†]

(Sang-Un. Lee)

요약 소프트웨어 개발 초기에 개발비용, 소요인력과 기간을 추정하는 것은 소프트웨어공학 분야의 중요하면서도 어려운 문제이다. 기존 모델은 개발업체의 생산성 수준을 고려하지 않고 단순히 기능점수와 노력, 노력과 개발기간 관계에 대한 회귀분석을 통해 개발노력과 기간을 추정하는 모델을 제시하였다. 그러나 동일한 규모의 소프트웨어라도 개발업체의 생산성 수준에 따라 다른 노력의 양이 투입되며, 동일한 노력이 투입되더라도 생산성에 따라 개발기간이 달라진다. 생산성 수준을 고려하지 않은 제안된 모델들은 실제 개발될 프로젝트 적용에 제한점을 갖고 있다. 본 논문은 기존 모델의 단점을 보완하기 위해 생산성을 고려하여 개발기간을 추정할 수 있는 모델들을 제안한다. 생산성에 기반하여 다양한 방법으로 개발기간을 추정할 수 있는 모델을 제안하고 모델들의 성능을 비교하였다. 모델 성능 비교 결과 생산성에 기반하여 소프트웨어 규모로부터 개발기간을 추정하는 모델이 단순하면서도 가장 좋은 결과를 얻었다. 본 모델은 사업 초기에 프로젝트 관리자에게 소프트웨어 개발 기간 의사결정 정보를 제공한다.

키워드 : 개발노력, 개발기간, 소프트웨어 규모, 기능점수, 변수 변환

Abstract Estimation of software project cost, effort, and duration in the early stage of software development cycle is a difficult and key problem in software engineering. Most of development effort and duration estimation models presented by regression model of simple relation function point vs. effort and effort vs. duration instead of considering developer's productivity. But different project have need for different effort according to developer's productivity if the projects are same software size. Also, different duration takes according to developer's productivity if the projects require the same effort. Therefore, models that take into account of productivity have a limited application in actual development project. This paper presents models that can be estimate the duration according to productivity in order to compensate a shortcoming of the previous models. Propose model that could presume development period by various methods based on productivity and compared models' performance. As a result of performance comparison, an estimating model of development period from software size got simple and most good result. The model gives decision-making information of development duration to project management in the early stage of software life cycle.

Key words : Development Effort, Development Duration, Software Size, Function Point, Variable Transformation

1. 서론

프로젝트 관리자에게 가장 중요한 문제인 소프트웨어 개발노력, 일정과 비용에 대한 타당성 있는 추정은 과학보다는 차라리 예술에 가까운 정도로 어렵다[1]. 프로젝트 계획의 성공 여부는 프로젝트를 완료시키는데 요구되는 노력과 달성 가능한 개발기간에 대한 만족스러운 추정에 의존한다. 개발노력과 기간을 추정하는 많은 모

델들이 제안되었으며, 그림 1에서 보는바와 같이 소프트웨어 규모(Size)가 개발노력과 일정 추정에 영향을 미치는 가장 중요한 요소이다. 따라서, 개발 노력, 일정 추정의 타당성은 우선적으로 측정된 소프트웨어 규모 추정의 타당성에 의존한다. 소프트웨어 규모를 추정하는 대표적인 척도로는 LOC(Lines Of Code)와 기능점수(Function Point, FP)가 있다.

LOC 척도는 언어에 종속되어 있으며, 요구분석 또는 설계단계에서 정확한 추정이 어렵고 코딩이 완료된 시점에서 정확한 결과를 얻을 수 있다. 이에 반해, FP는 사용자에게 양도될 시스템의 기능에 기반을 두고 소프

[†] 정 회 원 : 도림강원전문대학 컴퓨터응용과 교수
sangun@gangwon.ac.kr

논문접수 : 2003년 5월 7일

심사완료 : 2003년 9월 5일

트웨어 시스템의 규모와 복잡도를 정량화하는 방법으로 소프트웨어 프로젝트를 개발하기 위해 사용되는 언어 또는 도구와 독립적이며, 개발 생명주기의 초기단계인 요구분석 단계에서 측정 가능한 장점이 있다[2]. 따라서, 본 논문은 소프트웨어 규모 측정 방법으로 기능점수 방법을 고려한다.

측정된 소프트웨어 규모인 기능점수 FP 를 개발하기 위해 투입될 노력(Effort, E)을 추정하는 연구로는 Matson et al.[2], Albrecht[3,4], Albrecht et al.[5], Kemerer[6,7], Low et al.[8] 등이 있다. 또한, 국내에서도 다양한 연구가 수행되고 있다[9-13]. 개발에 투입될 총 노력 E 에 기반하여 개발기간 (Duration, D)를 추정하는 연구로는 Oligny et al.[14,15], Kitchenham[16], Ferens et al.[17] 등이 있다. Oligny et al.[14,15]는 개발 플랫폼별로 구분하여 총 소요 노력 E 와 개발기간 D 의 로그 관계로부터 개발기간을 추정하였다. 제안된 모델들은 다음과 같은 문제점을 갖고 있다: (1) 동일한 노력이 투입되는 프로젝트라 할지라도 개발업체의 생산성 수준에 따라 다른 개발기간이 소요될 수 있으나 Oligny et al.[14,15] 모델은 이들 변수들을 고려하지 않았다; (2) 측정된 소프트웨어 규모인 FP 로부터 직접 개발기간 D 를 추정하는 모델도 제시되지 않고 있다. 왜냐하면 소프트웨어 규모로부터 개발노력 E 을 추정한 후 다시 추정된 개발노력으로 부터 개발기간을 추정하면 추정 단계별로 오차가 누적되어 편향된 결과를 유발시킬 수 있기 때문이다; (3) 다양한 모델들이 제안되었지만 제한적인 프로젝트 수와 특정 업체에서만 개발된 자료들을 대상으로 한정하였기 때문에 다양한 프로젝트들 환경에 에 일반적으로 적용할 수 있는 모델이 없는 실정이다.

본 논문은 기존 모델들의 단점을 보완하기 위하여 개발업체의 생산성 수준 척도에 기반하여 소프트웨어 규모 FP 로부터 직접 또는 개발노력 E 로부터 개발기간 D 를 추정할 수 있는 다양한 모델들을 제안한다. 2장에서는 관련연구와 문제점을 살펴보고, 3장에서는 소프트

웨어 개발기간 D 를 추정할 수 있는 4가지 모델들을 제시하고, 4장에서는 모델의 성능을 평가해 본다.

2. 관련 연구 및 연구동기

2.1 관련연구

개발될 소프트웨어 규모에 기반하여 소요되는 노력의 양을 적정하게 추정하는 것이 소프트웨어 개발계획 작성시 필수적으로 요구되는 사항이다. 기능점수 FP 규모에 기반하여 소프트웨어 프로젝트를 개발하는데 필요한 노력 E 를 추정하기 위해 제안된 모델들은 일반적으로 회귀분석 과정을 거쳐 유도되었으며, $E = a + b \cdot FP$, $\ln(E) = a + b \cdot \ln(FP)$, $\sqrt{E} = a + b \cdot FP$ 또는 $E = a + b \cdot FP + c \cdot FP^2 + d \cdot FP^3$ 형태를 취한다. 여기서, a, b, c, d 는 상수이다.

제품 출시 또는 운영 시점으로 인해 프로젝트를 시작하기 전에 고객이 프로젝트의 개발기간을 요구하는 경우가 빈번히 발생한다. 이 경우, 고객이 요구하는 기간 내에 프로젝트 개발을 완료할 수 있는지를 판단하기 위해서는 측정된 소프트웨어 규모로부터 또는 개발노력으로부터 달성 가능한 현실성 있는 개발기간을 추정하는 것이 필요하다.

프로젝트의 개발노력 E 에 기반하여 소프트웨어 개발기간 D 를 추정하기 위한 많은 통계적 모델들이 제안되었다[14-19]. 연구 결과들은 통계적 개발기간 모델들의 정확성에 관해 상반된 결과를 보였다. 또한, 이들 연구에 사용된 데이터 크기가 매우 작고 오래된 과거 데이터들로 최근 적용되는 개발언어, 기법 등을 반영하지 못하고 있는 점이다. Kitchenham[16]는 20개 또는 그 이하의 프로젝트 표본으로부터 경험적으로 21개 개발기간 모델 중 12개를 유도하였으며, 6개 모델은 21-33개 프로젝트 표본으로부터, 나머지 3개 모델은 46, 63과 81개 프로젝트 표본으로부터 유도하였다. Ferens와 Daly[17]는 개발기간 추정 통계 모델을 평가하는 3종의 연구결과를 보였다. 이것은 소프트웨어 개발기간 추정시 모델들을 적절히 적용하기 위해서는 숙련을 요구함을 의미한다[19]. Oligny et al[14,15]는 1989~1996년에 개발 완료된 396개 소프트웨어 프로젝트에 대한 데이터를 갖고 있는 ISBSG Database Release 4[20]에 근거하여 전체 데이터 표본을 사용하는 것보다 보다 동질성을 갖는 서브 그룹으로 분할하여 개발기간을 추정하는 모델을 개발하였다. 프로젝트들은 개발된 플랫폼인 Main Frame(MF), Mid-range(MR)와 Personnel Computer(PC) 별로 동질성을 갖도록 분류되었다. 주어진 데이터의 개발노력 E 와 개발기간 D 관계가 비선형 형태를 취하며, 편향된 분포를 가져, 데이터를 로그 변환시킨

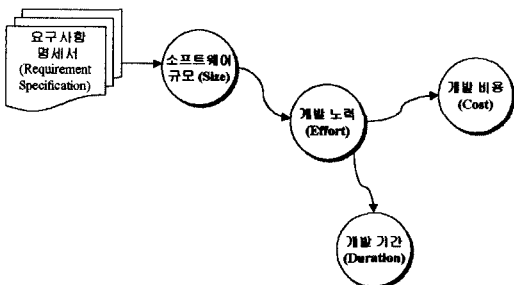


그림 1 소프트웨어 개발노력, 기간과 비용 추정

표 1 Oligny et al.[14,15] 모델 성능

플랫폼	프로젝트 수	개발기간 추정모델	모델 성능 (결정계수)
MF	208	$D_{MF} = 0.458E^{0.366}$	0.4134
MR	65	$D_{MR} = 0.548E^{0.360}$	0.4849
PC	39	$D_{PC} = 1.936E^{0.210}$	0.3874

$\log(E)$ 와 $\log(D)$ 의 관계가 보다 선형적 관계를 가지며, 데이터가 정규분포를 따름으로 이 관계로부터 회귀분석 결과 표 1의 모델을 제안하였다.

2.2 모델 성능 평가 척도

회귀모델의 경우, 종속변수의 총 변동(SST , Total Sum of Squares) 값은 독립변수에 의해 결정되는 부분 (SSR , Regression Sum of Squares)과 미지의 오차의 합(SSE , Residual Sum of Squares)으로 나타난다. 총 변동 중에서 회귀직선으로 설명할 수 있는 비율을 결정 계수(Coefficient of determination, $R^2 = SSR/SST$)라 한다. 따라서, $R^2 (0 \leq R^2 \leq 1)$ 이 0에 가까우면 추정된 회귀직선은 쓸모가 없으며, 값이 클수록 쓸모 있는 회귀직선이 된다[21]. 또 다른 척도인 $MMRE$ 를 살펴보자. 상대오차 (Relative Error, RE)는 (실측치 - 추정치)/실측치 $\times 100(\%)$ 이다. MRE (Magnitude of the RE)는 $|RE|$ 이며, n 개의 데이터에 대한 $MMRE$ (Mean MRE) $= \frac{1}{n} \sum_{i=1}^n MRE_i$ 로 계산된다. 따라서, $MMRE$ 가 작은 값이면 모델은 평균적으로 좋은 모델이 된다. Conte et al.[22]는 $MMRE \leq 0.25 (25\%)$ 이면 개발 노력을 평가하는 모델로 채택 가능한 것으로 고려하였다. 그러나 결정 계수에 대해서는 얼마나 큰 값을 가져야 좋은 모델로 선정 가능한지에 대한 연구 결과가 제시되지 않은 실정이다.

작은 수에서 발생한 오차의 값과 큰 수에서 발생한 오차의 값이 동일할 경우, 결정계수에 사용되는 절대오차 개념인 잔차제곱합은 동일한 값을 나타낸다. 이에 반해, 작은 수에서 상대오차는 상당히 큰 값을 보이는 반면 큰 수에서 상대오차는 무시할 만한 미미한 값을 가진다. 따라서, 분포 상태가 극도로 비대칭인 경우 잔차제곱합을 적용하는 결정계수보다는 평균 절대 상대오차 ($MMRE$)가 모델의 적절성을 평가하는 척도로 더 큰 의미를 갖는다. 따라서, 본 논문에서는 모델의 성능을 평가하는 척도로 $MMRE$ 를 적용한다.

2.3 개발기간 추정 모델의 적용 한계

Oligny et al[14,15]이 제안한 방법을 최근의 데이터 들인 ISBSG Benchmark Release 6[23]의 프로젝트들에 적용하여 보았다. 789개 프로젝트 데이터 중에 개발

표 2 ISBSG Release 6[23] 적용 결과

플랫폼	프로젝트 수	개발기간 추정 모델	모델 성능	
			결정계수	MMRE
MF	199	$D_{MF} = 0.458E^{0.366}$	0.5104	454.31 %
MR	76	$D_{MR} = 0.548E^{0.360}$	0.5328	324.73 %
PC	60	$D_{PC} = 1.936E^{0.210}$	0.5393	535.53 %

플랫폼 (MF, MR 또는 PC)이 기술된 335개 프로젝트가 선정되었다. 플랫폼별로 추정된 개발기간 모델의 성능은 표 2에 제시하였다. 표 2는 표 1의 데이터들에 비해 보다 최근에 개발된 프로젝트들로 최근 적용된 개발 언어, 기법 등을 반영하고 있다. 최근의 개발 프로젝트들에 적용한 결과 결정계수 측면에서는 Oligny et al.[14,15] 제시 모델에 비해 향상된 결과를 보였으나 $MMRE$ 가 25% 이하지 예측모델로 적합하다는 Conte et al.[22]의 기준과 비교시 적합한 모델로 채택이 불가능함을 알 수 있다.

개발 플랫폼별로 분류된 데이터들에 대해 개발노력 E 에 기반하여 개발기간 D 을 추정하는 모델은 현실성이 없어 보인다. 따라서, 개발기간에 영향을 미치는 다른 요인들을 파악하고, 이들 요인들은 고려하여 보다 타당성 있는 추정 모델을 제시할 필요가 있다. 소프트웨어 개발시 소요되는 총 개발노력(Man-Months)을 총 소요된 개발기간(Months)으로 나누면 월 평균 투입인력 (E/D)이 된다. 이를 \bar{E} (Man-Months per Month)라 하자. ISBSG Release 6[23]의 640개 프로젝트를 대상으로 \bar{E} 를 분석한 결과는 그림 2에 제시되어 있으며, 대부분의 프로젝트들이 월 평균 1-5명의 인력을 투입하여 개발을 수행함을 알 수 있다.

그림 3은 1 MM 부터 2 MM 까지 소요되는 프로젝트들에 대해 실제 소요된 개발기간을 보여주고 있다.

이론상으로, 2 MM의 노력이 소요되는 프로젝트의 경우 최소한 1개월에 1명의 노력을 투입하더라도 2개월

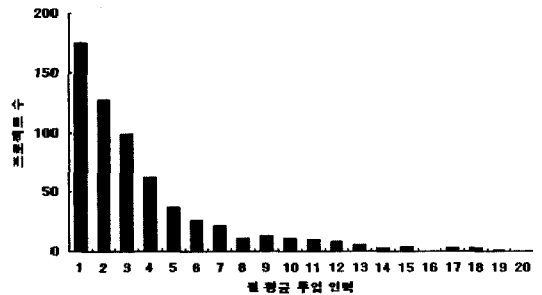


그림 2 월 평균 투입 인력별 프로젝트 수

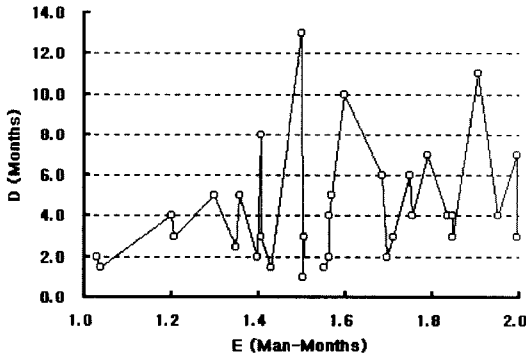


그림 3 1-2 MM 투입 프로젝트의 개발기간

이내에는 개발이 완료될 수 있다. 그러나, 실제로 2개월 이내에 개발이 완료된 프로젝트는 32개중 8개로 26.7%에 불과하였으며, 심지어는 13개월까지 소요되었음을 알 수 있다. 이와 같이 투입될 개발노력의 양이 유사하더라도 인력 투입의 효율성이 심한 변동을 갖고 있어 개발기간에도 상당한 차이를 나타낸다. 따라서, Oligny et al.[14,15]나 다른 연구 결과들이 제시한 단순한 방법으로는 개발노력에 기반하여 추정된 개발기간의 정확성에 한계가 있다. 이러한 한계점을 극복하기 위해서는 보다 동질성을 갖는 데이터들로 세분화하는 기법이 필요하다.

3. 소프트웨어 개발기간 추정 모델

동질성을 갖는 그룹들로 세분화하기 위해, 기능점수 1점을 개발하는데 소요되는 시간(Hours)을 고려하였다. 이 속성은 개발업체의 개발능력 수준에 영향을 받는다고 할 수 있다. ISBSG Benchmark Release 6[23]는 이 척도를 ‘프로젝트 인도율(Project Delivery Rate, PDR)’로, Longstreet[24]는 ‘생산성(Productivity)’이라는 용어로 정의하였다. PDR 값은 1부터 25까지 갖고 있다. 즉, 기능점수 1점을 개발하는데 어떤 프로젝트는 1시간이 소요되는 반면에 어떤 프로젝트는 25시간이 소요됨을 의미한다. 따라서, 개발업체의 생산성에 따라 동일한 규모의 프로젝트라도 소요되는 개발기간이 달라짐을 알 수 있다. 본 장에서는 생산성 수준별로 세분화시킨 프로젝트들에 기반을 두고 그림 4와 같이 개발기간을 추정할 수 있는 4개의 모델을 제안한다.

첫번째로, 제안 모델들이 개발노력 E 로부터 개발기간 D 를 추정하는데 기반하고, 월평균 투입인력에 차이가 발생한다는 사실을 반영하여 개발노력으로부터 월평균 투입인력을 산출하고, 이로부터 개발기간을 추정하는 모델을 제시한다.

① 모델 1 : 개발노력 E 에 기반하여 월 평균 투입 개발인력 \bar{E} 를 구한다. 이와 같이 구해진 월 평균 개발

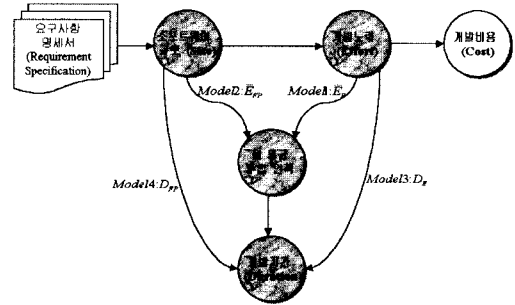


그림 4 개발기간 추정 모델 패밀리

인력의 규모를 \bar{E}_E 라 하자. 제안된 모델을 이용하여 \bar{E}_E 을 측정하고, 투입될 총개발인력 E 를 \bar{E}_E 으로 나누면 개발기간 D 를 구할 수 있다. 이와 같이 구한 개발기간을 $D_{\bar{E}_E}$ 라 하자.

두 번째로, 모델 1과 유사하지만 소프트웨어 개발노력이 아닌 소프트웨어 규모로부터 직접 월 평균 투입인력을 산출하여 이로부터 개발기간을 추정할 수 있는 모델을 제시한다.

② 모델 2 : 소프트웨어 규모 FP 에 기반하여 월 평균 투입 개발인력 \bar{E} 를 구한다. 이와 같이 구해진 월 평균 개발인력의 규모를 \bar{E}_{FP} 라 하자. 제안된 모델을 이용하여 \bar{E}_{FP} 를 측정하고, 투입될 총 개발인력 E 를 \bar{E}_{FP} 으로 나누면 개발기간 D 를 구할 수 있다. 이와 같이 구한 개발기간을 $D_{\bar{E}_{FP}}$ 라 하자.

세 번째로, 기존의 개발노력으로부터 개발기간을 추정하는 모델 대신 소프트웨어 규모로부터 직접 개발기간을 추정할 수 있는 모델을 제시한다.

③ 모델 3 : 소프트웨어 규모 FP 로부터 직접 개발기간 D 를 구한다. 이와 같이 구한 개발기간을 D_{FP} 라 하자.

마지막으로, 기존 제안 모델들과 동일하게 개발노력으로부터 개발기간을 추정하는 모델을 제안한다.

④ 모델 4 : 소프트웨어 개발노력 E 로부터 직접 개발기간 D 를 구한다. 이와 같이 구한 개발기간을 D_E 라 하자.

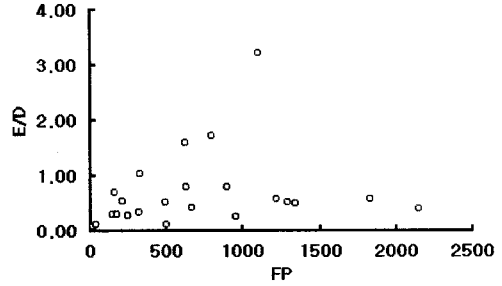
기존 제안 모델들은 개발업체의 생산성 변수를 고려하지 않았지만 새롭게 제안되는 모델들은 모델 4와 같이 기존 제안 모델들과 동일한 방법으로 개발기간을 추정하였더라도 개발업체의 생산성에 따라 동질성을 갖는 프로젝트들을 분류하였다는 차이점을 갖고 있다.

3.1 월 평균 개발인력 추정

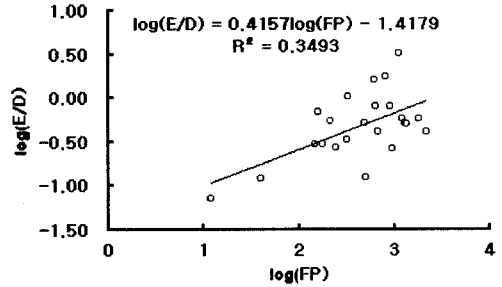
PDR별로 프로젝트들을 분류하고, 분류된 PDR 수준별로 $\log(FP)$ vs. $\log(\bar{E}_{FP})$, $\log(E)$ vs. $\log(\bar{E}_E)$ 관계를 그려 회귀분석 후 로그를 제거하면 표 3의 모델들을

표 3 월 평균 개발인력 추정 모델

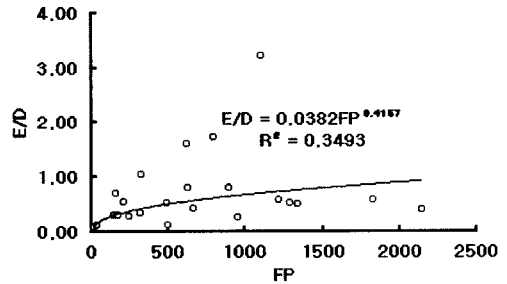
PDR	$\overline{E_{FP}}$ 모델	$\overline{E_E}$ 모델
1	$\overline{E_{FP}} = 0.3082FP^{0.4157}$	$\overline{E_E} = 0.3301E^{0.3861}$
2	$\overline{E_{FP}} = 0.0129FP^{0.7097}$	$\overline{E_E} = 0.3091E^{0.7285}$
3	$\overline{E_{FP}} = 0.0265FP^{0.6202}$	$\overline{E_E} = 0.3394E^{0.6063}$
4	$\overline{E_{FP}} = 0.0599FP^{0.5176}$	$\overline{E_E} = 0.4260E^{0.5138}$
5	$\overline{E_{FP}} = 0.0321FP^{0.6752}$	$\overline{E_E} = 0.3421E^{0.6858}$
6	$\overline{E_{FP}} = 0.0489FP^{0.6075}$	$\overline{E_E} = 0.3624E^{0.6242}$
7	$\overline{E_{FP}} = 0.0617FP^{0.5606}$	$\overline{E_E} = 0.3746E^{0.5575}$
8	$\overline{E_{FP}} = 0.0296FP^{0.7486}$	$\overline{E_E} = 0.2960E^{0.7488}$
9	$\overline{E_{FP}} = 0.0620FP^{0.5909}$	$\overline{E_E} = 0.3455E^{0.6052}$
10	$\overline{E_{FP}} = 0.0246FP^{0.7606}$	$\overline{E_E} = 0.2155E^{0.7608}$
11	$\overline{E_{FP}} = 0.1192FP^{0.5604}$	$\overline{E_E} = 0.5646E^{0.5579}$
12	$\overline{E_{FP}} = 0.0538FP^{0.6633}$	$\overline{E_E} = 0.3158E^{0.6645}$
13	$\overline{E_{FP}} = 0.1120FP^{0.5398}$	$\overline{E_E} = 0.4469E^{0.5458}$
14	$\overline{E_{FP}} = 0.0006FP^{1.6181}$	$\overline{E_E} = 0.0383E^{1.5907}$
15	$\overline{E_{FP}} = 0.0216FP^{0.8013}$	$\overline{E_E} = 0.1501E^{0.8077}$
16	$\overline{E_{FP}} = 0.0585FP^{0.7024}$	$\overline{E_E} = 0.3090E^{0.7062}$
17	$\overline{E_{FP}} = 0.0411FP^{0.7933}$	$\overline{E_E} = 0.2550E^{0.9001}$
18	$\overline{E_{FP}} = 0.0006FP^{1.5824}$	$\overline{E_E} = 0.0225E^{1.5881}$
19	$\overline{E_{FP}} = 0.2287FP^{0.4339}$	$\overline{E_E} = 0.5956E^{0.4349}$
20	$\overline{E_{FP}} = 0.0598FP^{0.6828}$	$\overline{E_E} = 0.2545E^{0.6904}$
21	$\overline{E_{FP}} = 0.0394FP^{0.7253}$	$\overline{E_E} = 0.1843E^{0.7224}$
22	$\overline{E_{FP}} = 0.0100FP^{1.0824}$	$\overline{E_E} = 0.0952E^{1.0767}$
23	$\overline{E_{FP}} = 0.1401FP^{0.5817}$	$\overline{E_E} = 0.4839E^{0.5661}$
24	$\overline{E_{FP}} = 0.0567FP^{0.6660}$	$\overline{E_E} = 0.2164E^{0.6609}$
25	$\overline{E_{FP}} = 0.0308FP^{0.8280}$	$\overline{E_E} = 0.1495E^{0.8356}$



(a) 기능점수와 월 평균 개발인력 관계



(b) 로그변환에 의한 회귀 모델



(c) 변수 역변환으로 최종적으로 얻어진 회귀분석 모델
그림 5 기능점수에 근거한 월평균 개발인력 추정 모델 유도

구할 수 있다. 예로, 기능점수 FP 에 근거한 월 평균 개발인력 $\overline{E_{FP}}$ 는 그림 5의 순서로 구해진다.

3.2 소프트웨어 개발기간 추정

월 평균 투입 개발인력 추정 모델과 동일한 방법으로, PDR 별로 프로젝트들을 분류하고, 분류된 PDR 수준 별로 $\log(FP)$ vs. $\log(D_{FP})$, $\log(E)$ vs. $\log(D_E)$ 관계를 그려 회귀분석 후 로그를 제거하면 표 4의 모델들을 구할 수 있다. 예로, 기능점수 FP 에 근거한 개발기간 D_{FP} 는 그림 6의 순서로 구해진다.

4. 모델 평가

다수의 생산성 수준에 따른 모델의 성능을 종합적으로 비교하기 위해 Karunanithi, Whitley와 Malaiya

[25]가 제시한 모델의 성능 척도를 사용하고자 한다. 이 방법은 i 번째 생산성 수준에 대해 m 번째 모델의 정규화된(Normalized) MMRE인 $NMMRE_i^m = \frac{MMRE_i^m}{MMRE_{max}^m}$ 를 구한 후 각 모델에 대해 $R_m = \sum_{i=1}^n MMRE_i^m$ 을 구하여 모델의 순위를 결정한다.

4.1 월 평균 개발인력 추정

모델 평가기준에 의해 월 평균 개발인력을 추정하는 $\overline{E_{FP}}$ 와 $\overline{E_E}$ 모델의 성능을 비교한 결과는 표 5에 제시되어 있다. 생산성 수준별 MMRE가 작은 경우 수는 $\overline{E_E} : \overline{E_{FP}}$ 가 12 : 14로 $\overline{E_{FP}}$ 가 약간 우세한 성능을 보이나 R_m 측면에서는 유사한 성능을 가지고 있음을 알

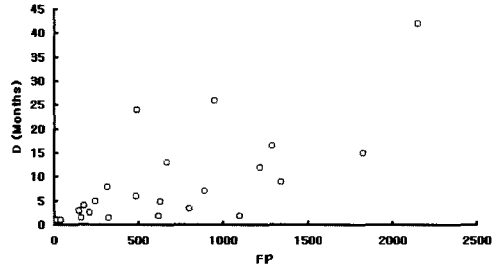
표 4 개발기간 추정 모델

PDR	D_{FP} 모델	D_E 모델
1	$D_{FP} = 0.1247FP^{0.6216}$	$D_E = 3.0292E^{0.6138}$
2	$D_{FP} = 1.1370FP^{0.2883}$	$D_E = 3.2349E^{0.2115}$
3	$D_{FP} = 0.6030FP^{0.3809}$	$D_E = 2.9463E^{0.3937}$
4	$D_{FP} = 0.3414FP^{0.5019}$	$D_E = 2.3473E^{0.4862}$
5	$D_{FP} = 0.9420FP^{0.3177}$	$D_E = 2.9231E^{0.3142}$
6	$D_{FP} = 0.8122FP^{0.3888}$	$D_E = 2.7591E^{0.3758}$
7	$D_{FP} = 0.6490FP^{0.4420}$	$D_E = 2.6696E^{0.4425}$
8	$D_{FP} = 1.5434FP^{0.2531}$	$D_E = 3.3786E^{0.2512}$
9	$D_{FP} = 0.9430FP^{0.3855}$	$D_E = 2.8941E^{0.3948}$
10	$D_{FP} = 2.3077FP^{0.2422}$	$D_E = 4.6407E^{0.2392}$
11	$D_{FP} = 0.5172FP^{0.4445}$	$D_E = 1.7711E^{0.4421}$
12	$D_{FP} = 1.2889FP^{0.3357}$	$D_E = 3.1664E^{0.3355}$
13	$D_{FP} = 0.7053FP^{0.4498}$	$D_E = 2.2375E^{0.4542}$
14	$D_{FP} = 123.7045FP^{-0.6032}$	$D_E = 26.1230E^{-0.5907}$
15	$D_{FP} = 4.1687FP^{0.1920}$	$D_E = 6.6617E^{0.1923}$
16	$D_{FP} = 1.5925FP^{0.2853}$	$D_E = 3.2365E^{0.2838}$
17	$D_{FP} = 2.4729FP^{0.1992}$	$D_E = 3.9218E^{0.1999}$
18	$D_{FP} = 167.7102FP^{-0.5867}$	$D_E = 44.3720E^{-0.5881}$
19	$D_{FP} = 0.4817FP^{0.5647}$	$D_E = 1.6791E^{0.5651}$
20	$D_{FP} = 2.0049FP^{0.3105}$	$D_E = 3.9288E^{0.3096}$
21	$D_{FP} = 3.0318FP^{0.2761}$	$D_E = 5.4245E^{0.2766}$
22	$D_{FP} = 12.4474FP^{-0.0789}$	$D_E = 10.5042E^{-0.0767}$
23	$D_{FP} = 0.7657FP^{0.4527}$	$D_E = 2.0664E^{0.4338}$
24	$D_{FP} = 2.3349FP^{0.3409}$	$D_E = 4.6221E^{0.3391}$
25	$D_{FP} = 4.9073FP^{0.1626}$	$D_E = 6.6870E^{0.1644}$

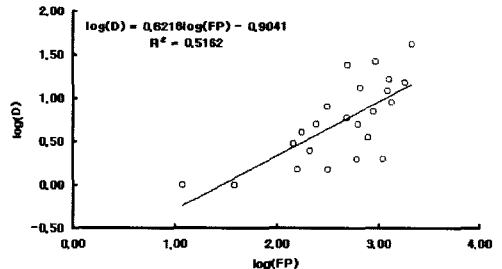
수 있다.

4.2 소프트웨어 개발기간 추정

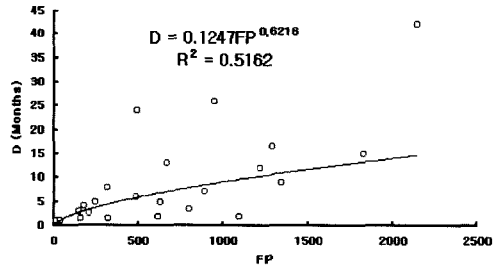
월 평균 개발인력 추정 모델인 $\overline{E_E}$ 와 $\overline{E_{FP}}$ 를 이용해 추정된 개발기간 $D_{\overline{E_E}}$, $D_{\overline{E_{FP}}}$ 와 D_E , D_{FP} 모델로부터 측정된 개발기간의 성능을 비교한 결과는 표 6에 제시하였다. 표 6에서 MMRE 비교시 $\overline{E_{FP}}$ 모델로 추정된 개발기간이 실측치에 가장 근접된 결과를 보였으며, 1위인 $D_{\overline{E_{FP}}}$ 와 4위인 D_{FP} 간의 편차는 그림 7의 (a)에 제시되어 있다. 그림에서 '-'이면 $D_{\overline{E_{FP}}}$ 가 D_{FP} 보다 상대 오차가 적음을, '+' 이면 이와 반대인 경우를 의미한다. 따라서, PDR 값이 적은 부분(생산성이 증가)에서는 $D_{\overline{E_{FP}}}$ 가 D_{FP} 에 비해 상대오차가 상당히 적음을 알 수 있다. 그러나 PDR 값이 증가(생산성 저하)할수록 D_{FP} 가 $D_{\overline{E_{FP}}}$ 모델 보다 좋은 결과를 얻음을 알 수 있다. 그러나 NMMRE로 변환된 결과를 비교시 $D_{\overline{E_{FP}}}$ 는 1위에



(a) 기능점수와 개발기간 관계



(b) 로그변환에 의한 회귀 모델



(c) 변수 역변환으로 최종적으로 얻어진 회귀모델

그림 6 기능점수에 근거한 개발기간 추정 모델 유도

서 2위로, D_{FP} 는 4위에서 1위로 변경되었다. NMMRE로 변경된 $D_{\overline{E_{FP}}}$ 와 D_{FP} 의 성능을 비교한 결과는 (b)에 제시되어 있다.

전반적으로 PDR 값이 적은 부분에서는 $D_{\overline{E_{FP}}}$ 모델이 우세하나 PDR이 5-13, 23에서는 D_{FP} 가 크게 우세함을 알 수 있다. 4개 모델 모두 PDR 값이 1, 2, 4-7, 9, 11, 15와 16의 10개 영역에서 Conte et al.[17] 기준을 초과한다. 그러나 불만족하는 경우를 살펴보면 대부분이 25-30% 범위의 상대오차를 보이며, 40%대가 2건, 70%대가 1건으로 양호한 결과를 보이고 있다. 이는 표 2의 플랫폼별 추정된 개발기간 모델의 성능과 비교하면 월등한 성능 향상을 보임을 알 수 있다.

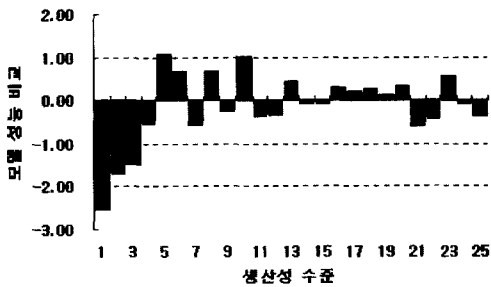
결론적으로, 4개 모델 모두 25개 생산성 수준에 대한 R_m 은 유사한 결과를 보이고 있어 어느 모델이 특히 좋다고 할 수 없으며, 모두 좋은 모델로 선정될 수 있다.

표 5 월 평균 투입 개발인력 추정 모델 성능 비교

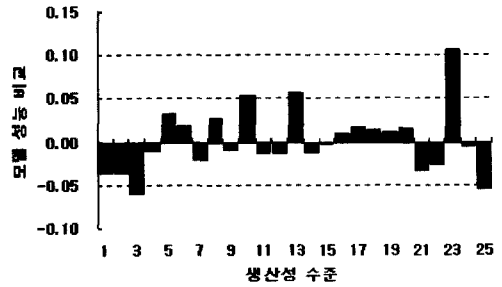
PDR	MMRE (%)		NMMRE	
	\bar{E}_E	\bar{E}_{FP}	\bar{E}_E	\bar{E}_{FP}
1	59.80		1.0000	0.9975
2	49.07		1.0000	0.9863
3	26.04		1.0000	0.9508
4		55.02	0.9904	1.0000
5		33.06	0.9731	1.0000
6		45.44	0.9850	1.0000
7	28.83		1.0000	0.9795
8		26.36	0.9727	1.0000
9	25.31		1.0000	0.9838
10		19.85	0.9602	1.0000
11	24.74		1.0000	0.9976
12			1.0000	1.0000
13		8.21	0.9440	1.0000
14	5.85		1.0000	0.9385
15	29.62		1.0000	0.9973
16		34.37	0.9907	1.0000
17		13.24	0.9841	1.0000
18	19.02		1.0000	0.9206
19		11.91	0.9924	1.0000
20		24.59	0.9813	1.0000
21	18.91		1.0000	0.9730
22	17.14		1.0000	0.9790
23		5.31	0.9510	1.0000
24	20.99		1.0000	0.9929
25	6.46		1.0000	0.9644
R_m	629.38	628.43	24.7249	24.6612
순위	2	1	2	1

표 6 개발기간 추정 모델 성능 비교

PDR	MMRE				NMMRE			
	$D_{\bar{E}_E}$	$D_{\bar{E}_{FP}}$	D_E	D_{FP}	$D_{\bar{E}_E}$	$D_{\bar{E}_{FP}}$	D_E	D_{FP}
1	68.88		68.88	71.38	0.9650	0.9646	0.9650	1.0000
2	45.37		45.37	46.22	0.9816	0.9634	0.9816	1.0000
3	23.54		23.54	24.35	0.9667	0.9400	0.9667	1.0000
4	47.05		47.05	47.30	0.9947	0.9888	0.9947	1.0000
5	33.41	34.15	33.41		0.9783	1.0000	0.9783	0.9684
6	36.32	36.73	36.31		0.9888	1.0000	0.9886	0.9820
7	27.15		27.15	27.43	0.9898	0.9789	0.9898	1.0000
8	24.65	25.22	24.65		0.9774	1.0000	0.9774	0.9734
9	25.02		25.02	25.10	0.9968	0.9904	0.9968	1.0000
10	18.08	18.88	18.08		0.9576	1.0000	0.9576	0.9460
11	26.36		26.36	26.52	0.9940	0.9864	0.9940	1.0000
12	24.73		24.73	24.83	0.9960	0.9863	0.9960	1.0000
13	7.55	7.93	7.55		0.9521	1.0000	0.9521	0.9433
14	6.12		6.11	5.63	1.0000	0.9085	0.9984	0.9199
15	32.73		32.72	32.75	0.9994	0.9979	0.9991	1.0000
16	31.34	31.52	31.34		0.9943	1.0000	0.9943	0.9908
17	12.29	12.46	12.29		0.9864	1.0000	0.9864	0.9839
18	21.09	21.08	21.11		0.9991	0.9986	1.0000	0.9853
19	10.91	10.97	10.91		0.9945	1.0000	0.9945	0.9891
20	22.09	22.38	22.09		0.9870	1.0000	0.9870	0.9853
21	17.76		17.75	17.97	0.9883	0.9661	0.9878	1.0000
22	16.54		16.54	16.45	1.0000	0.9680	1.0000	0.9946
23	5.03	5.30	5.02		0.9491	1.0000	0.9472	0.8943
24	20.24		20.24	20.25	0.9995	0.9956	0.9995	1.0000
25	6.75		6.76	6.83	0.9883	0.9458	0.9898	1.0000
R_m	611.00	610.25	610.98	613.97	24.6247	24.5792	24.6224	24.5563
순위	3	1	2	4	4	2	3	1



(a) MMRE 비교 ($D_{\bar{E}_{FP}} - D_{FP}$)



(b) NMMRE 비교 ($D_{\bar{E}_{FP}} - D_{FP}$)

그림 7 개발기간 추정 모델 성능 비교

그러나 FP에 근거하여 \bar{E} 를 구하고 다시 D를 구하는 $D_{\bar{E}_{FP}}$, FP에 근거하여 \bar{E} 를 구한 후 D를 구하는 $D_{\bar{E}_E}$ 와 FP에 근거하여 E를 구한 후 D를 구하는 D_E 의 복잡한 단계를 거치는 과정에서 누적되는 오차가 발생할 수 있다. 따라서, FP에 근거하여 직접 개발기간

D를 구하는 D_{FP} 가 실제 적용 측면에서는 오차가 가장 적을 수 있고, 가장 손쉽게 적용할 수 있을 것이다.

5. 결론 및 향후과제

소프트웨어 개발기간을 추정하기 위해, 기존 연구들은

개발노력이 개발기간에 가장 큰 영향을 미치는 요인이라 생각하여 개발노력 E 에 기반하여 개발기간 D 를 추정하였다. 그러나 개발기간은 단순히 개발노력이나 개발 플랫폼에만 영향을 받지 않는다. 이들 요인들보다는 업체의 생산성, 월 평균 투입 인력, 개발팀의 규모 등 다른 요인들에 의해 보다 큰 영향을 받을 수 있다. 따라서, 본 논문은 업체의 생산성 수준에 기반을 두고 개발기간을 추정하는 모델을 제안하였다. 기존 모델들은 개발노력에 기반한 모델인데 비해 본 제안 모델들은 개발노력으로부터 월 평균 투입 인력을, 소프트웨어 규모로부터 월 평균 투입 인력을, 개발노력으로부터 개발기간을, 소프트웨어 규모로부터 개발기간을 추정하는 모델들을 제안하였다.

제안된 4개 모델들은 모두 유사한 성능을 보였으며, Oigny et al.[9,10]이 제안한 플랫폼별 개발노력에 기반한 개발기간 추정 모델들에 비해 월등한 성능 향상을 보였다. 이들 모델 중에서 소프트웨어 규모 FP 에 근거하여 직접 개발기간 D 를 구하는 D_{FP} 모델이 추정 과정이 가장 단순하며, 추정 과정 중에 발생하는 누적 오차를 줄일 수 있는 장점이 있어 실무에 적용하기가 쉬울 것이다. 따라서, 개발업체의 생산성 수준과 개발될 소프트웨어의 규모만 측정하였다면, 제안된 모델을 이용하여 타당성 있는 개발기간을 추정할 수 있을 것이다.

본 논문에서 적용된 프로젝트들은 3세대, 4세대 언어로 구현되었으며, 구조적, 객체지향 기법 등을 적용하여 실제와 괴리가 발생하지 않아 현 시점에서 적용하기에는 어려움이 없을 것이다. 그러나 소프트웨어 개발방법론 중 새롭게 각광받고 있는 CBD(Component-Based Development)로 기존에 운영중인 컴포넌트 들을 결합시켜 프로젝트들을 개발하는 형태는 현재의 구조적이나 객체지향 개발 방법론과는 완전히 다른 패러다임으로 현재 제안된 모델을 적용할 수 없다. 따라서, 추후로는 이 경우에 대한 개발노력과 기간을 추정하는 모델에 대한 연구가 수행될 것이다.

참고 문헌

- [1] 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.
- [2] Matson, J. E., Barrett B. E. and Mellichamp, J. M., "Software Development Cost Estimation Using Function Points," IEEE Trans. on Software Eng., Vol.20, No.4, pp. 275~287, 1994.
- [3] Albrecht, A. J., "Measuring Applications Development Productivity," Proceedings of IBM Application Dev., Joint SHARE/GUIDE Symposium, Monterey, CA., pp. 83~92, 1979.
- [4] Albrecht, A. J., "Measuring Application Development Productivity," In Programming Productivity : Issues for the Eighties, C. Jones, ed. Washington, DC : IEEE Computer Society Press, 1981.
- [5] Albrecht, A. J. and Gaffney, J. E., "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.
- [6] Kemerer, C. F., "An Empirical Validation of Software Cost Estimation Models," Communication ACM, Vol. 30, No. 5, pp. 416~429, 1987.
- [7] Kemerer, C. F., "Reliability of Functional Point Measurement-A Field Experiment," Communications of ACM, 1993.
- [8] Low, G. C. and Jeffery, D. R., "Function Points in the Estimation and Evaluation of the Software Process," IEEE Trans. on Software Eng., Vol. 16, pp. 64~71, 1990.
- [9] 이상운, "신경망을 이용한 소프트웨어 개발노력 추정", 한국정보처리학회 논문지(D), 제8-D권, 제3호, pp. 241~246, 2001.
- [10] 이상운 et al. "RBF 망을 이용한 소프트웨어 개발노력 추정 성능향상", 한국 정보처리학회 논문지, 제8-D권, 제5호, pp. 581~586, 2001.
- [11] 이상운, "주성분분석을 이용한 소프트웨어 개발노력 추정능력 향상", 한국 정보처리학회 논문지, 제9-D권, 제1호, pp. 75~80, 2002.
- [12] 이상운 et al. "프로젝트 인도율 그룹 분할 방법을 이용한 소프트웨어 개발노력 추정", 한국정보처리학회 논문지(D), 제9-D권, 제2호, pp. 259~266, 2002.
- [13] 이상운 et al. "기능점수를 이용한 소프트웨어개발노력 추정", 한국정보처리학회 논문지, 제9-D권, 제4호, pp. 603~612, 2002.
- [14] Oigny, S., Bourque, P. and Abran, A., "An Empirical Assessment of Project Duration Models in Software Engineering," In The Eight European Software Control and Metrics Conference (ES-COM'97), Berlin Germany, 1997.
- [15] Oigny, S., Bourque, P., Abran, A. and Fournier, B., "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] Kitchenham, B. A., "Empirical Studies of Assumptions That Underlie Software Cost-estimation Models," Information and Software Technology, Vol. 34, No. 4, pp. 211~218, 1992.
- [17] Ferens, D. V. and Daly, A., "A Comparison of Software Scheduling Methods," In Reifer D, ed. Software Management. 4th ed. Washington : IEEE Computer Society Press, 1993.
- [18] Jones, C., "Determining Software Schedules," Computer Vol. 28, No. 2, pp. 73~75, 1995.
- [19] Park, R. E., Goethert, W. B. and Webb, J. T., "Software Cost and Schedule Estimating : A

- Process Improvement Initiative," Pittsburgh, PA Software Engineering Institute, 1994.
- [20] ISBSG, "Worldwide Software Development - The Benchmark Release 4," Victoria, Australia International Software Benchmarking Standards Group, 1997.
- [21] 김우철 et al., "현대통계학", 영지출판사, 1994.
- [22] Conte, S. D., Dunsmore, H. E. and Shen, V. Y., "Software Engineering Metrics and Models," Menlo Park., CA: Benjamin Cummings, 1986.
- [23] ISBSG, "Worldwide Software Development-The Benchmark Release 6," Victoria, Australia International Software Benchmarking Standards Group, 2000.
- [24] Longstreet, D., "Estimating Software Development Effort Using Function Points," <http://www.softwaremetrics.com/Articles>.
- [25] Karunanithi, N., Whitley, D. and Malaiya, Y. K., "Prediction of Software Reliability Using Connectionist Models," IEEE Trans. Software Eng., Vol. 18, pp. 563~574, 1992.



이 상 운

1983년~1987년 한국항공대학교 항공전자공학과(학사). 1995년~1997년 경상대학교 컴퓨터과학과 (석사). 1998년~2001년 경상대학교 컴퓨터과학과(박사). 1992년~2002년 국방품질관리소 항공전자장비 및 소프트웨어 품질보증 담당. 2003

년~현재 도립 강원전문대학 컴퓨터응용과 전임강사. 관심 분야는 소프트웨어 공학(소프트웨어 계획, Metrics, 개발방법론, CBD, 시험 및 품질보증, 신뢰성), 신경망, 뉴로-퍼지