

테스팅 데이터 분석을 통한 소프트웨어 개발 노력 추정

정혜정[†] · 양해술^{††}

요약

소프트웨어의 개발 노력을 추정하기 위한 연구가 지속적으로 이루어지고 있다. 그러나 이러한 개발 노력 추정을 위한 가장 이상적인 모델을 제시하기 위해서 실험자료를 얻어야 하나 이러한 실험자료를 얻는 것은 그리 쉬운 일이 아니다. 또한 이러한 실험자료가 얻어지면 실험자료에 대한 분석 또한 상당히 중요한 연구과제라고 볼 수 있다. 본 연구에서는 1990년대에 개발된 789개의 소프트웨어 개발 프로젝트들에 관련된 데이터를 이용하여 개발 노력에 영향을 미치는 요인별 데이터 분석을 실시하여 그 특징을 추출하고자 한다. 개발 소프트웨어의 규모가 다양하고 소프트웨어의 개발에 참여한 개발 팀의 규모도 차이가 있기 때문이다. 이와 같은 특징을 고려하여 주어진 자료에 대한 분석을 통해서 요인별 특징을 파악하고 파악된 자료를 중심으로 개발 노력을 단계별로 추정하려고 한다. 따라서, 본 연구에 사용된 789개 프로젝트 데이터를 개발형태(Development Type), 개발환경(Development Environment), 개발언어(Development Language) 등으로 분류하여 개발 노력(Development Effort)에 차이가 있음을 증명하고 팀 규모(Team Size)와 기능점수(Function Point)에 따른 교차분석(Crosstabs Analysis)을 실시하여 보았다.

Software Development Effort Estimation for Testing Data Analysis

Hye-Jung Jung[†] · Hae-Sool Yang^{††}

ABSTRACT

The research to estimate development effort of software has been progress. But, it is not easy gain that testing data for estimating of development effort. Also, if we get the testing data, it is important that analysis testing data. In this paper, we study the data analysis of software development effort using the 789 software development projects which developed in the 1990's. Software development scale and software development team size are various. Using the characteristic of factor, we have to study characteristic of data and we estimate the development effort step by step. First, we prove the difference of development effort with the 789 project data according to development type, development environment, the development language etc. Also, we execute the crosstabs analysis that team size and function point.

키워드 : 기능점수(Function Point), 개발형태(Development Type), 개발환경(Development Environment), 개발언어(Development Language)

1. 서론

소프트웨어는 개발 초기단계에서 개발에 투입될 노력과 비용을 추정하는 것이 상당히 중요한 과제이다. 그러나 소프트웨어의 개발 프로젝트에 영향을 미치는 다양한 환경외적인 요인으로 인한 개발 노력(Development Effort)과 비용 추정(Cost Estimation)을 위한 이상적인 모델(Model)이 제시되어 있지 않다. 소프트웨어의 개발 노력과 비용을 추정하기 위해서는 소프트웨어의 규모(Software Size)가 먼저 측정되어야 하며 이러한 규모를 측정하기 위한 척도로 연구되어진 방법은 LOC(Line Of Code)와 FPA(Function Point Analysis)가 있다. 그러나 LOC 방식은[7, 9] 3세대 언

어중심의 메인프레임용 소프트웨어 개발에서는 유효하나 4세대 언어, 오브젝트 라이브러리(Object Library), GUI(Graphical User Interface) 등을 사용하는 개발에서는 적합하지 않아서 이러한 문제점을 해결하기 위한 방안으로 기능점수(FP: Function Point)를 이용하여 소프트웨어의 개발 노력과 비용산정을 하고 있다.

소프트웨어 규모인 기능점수 FP를 이용하여 개발 노력을 추정하기 위한 연구는 대체적으로 회귀분석(Regression Analysis)을 이용하여 이상적인 모델을 제시하였다. 그러나 이 분야에 대한 연구에서 가장 어려운 점이 개발된 소프트웨어를 실험하는 과정에서 얻어진 실험데이터를 구하기 힘들다는 점에 있으며 또한 실험을 통해서 얻어진 데이터를 이용하여 연구되어 제안된 기본 모델들도 조사된 데이터의 숫자가 다소 부족하여 이상적인 모델로 제시하는 데는 문

[†] 종신회원 : 평택대학교 정보통계학과 교수

^{††} 종신회원 : 호서대학교 벤처전문대학원 교수

논문접수 : 2003년 8월 14일, 심사완료 : 2003년 1월 29일

제점을 안고 있다는 것이다. 또한 연구에 이용된 실험데이터는 대체적으로 특정업체의 프로젝트를 대상으로 조사되어진 것이므로 일반적인 소프트웨어의 개발 프로젝트에 적용한다는 것은 어려운 실정이다. 본 연구에서는 1990년대 20여 개국에서 조사된 789개의 소프트웨어 프로젝트에 대한 데이터베이스를 갖고 있는 ISBSG(International Software Benchmarking Standards Group) Benchmark Release 6[12]을 이용하여 조사된 자료에 대한 분석을 실시하여 각 자료별 특징을 파악하려 한다.

이러한 실험 자료의 분석을 통해서 개발 노력을 추정하는데 있어서 기존의 회귀식은 대체적으로 기능점수 FP만을 이용하여 모델을 제시하였으나 본 연구에서는 개발 노력에 영향을 미치는 변인으로는 기능점수 외에도 개발형태(Development Type), 개발환경(Development Environment), 개발언어(Development Language)등 다양한 변인에 의해서 영향을 받고 있음을 확인하고 이에 대한 구체적인 분석을 통해서 개발 노력(Development Effort)을 추정하여 오차(Error)를 줄이고자 한다. 본 연구에 사용된 ISBSG Benchmark Release 6은 여러국가에서 다양한 개발 환경하에 조사된 자료이므로 일괄적으로 데이터 전체를 이용하여 개발 노력을 추정하기 위한 모델을 제시 한다는 데에는 문제점이 있다고 보여진다. 이러한 문제점을 해결하기 위하여 주어진 자료에 대한 세부적인 분석을 통해서 개발 노력을 추정한다면 개발 노력을 추정하기 위한 더 이상적인 모델을 발견할 수 있으리라 기대된다. 그리하여 개발 프로젝트에 관련된 실험데이터를 얻었을 때 고려사항들을 파악하기 위하여 본 연구에서는 789개의 프로젝트데이터를 면밀히 분석하여 보려 한다.

본 논문의 2장에서는 개발 노력 추정과 관련된 기존의 연구를 소개하고 3장에서는 본 연구에 이용된 데이터 ISBSG Benchmark Release 6를 세부적으로 분석하여 개발 노력 추정을 위한 데이터의 변인에 대해 설명하고 4장에서는 이것을 이용한 가장 이상적인 모델을 제시하는 방법과 향후 연구 과제를 제시하였다.

2. 기존 연구와 배경

소프트웨어 개발 노력과 비용예측에 대한 연구는 Putman[8]의 SLIM(Software Lifecycle Management) 모델과 Boehm[2]의 COCOMO(COnstructive COst MOdel) 모델 등이 있다. 이러한 모델들은 소프트웨어 개발 노력과 비용산정을 위해서 소프트웨어의 스태프인 LOC(Line Of Code)를 이용하였다. 일반적으로 개발 노력을 추정하기 위한 모델은 두 가지

부분으로 나누어 구성되어진다. 첫 번째는 소프트웨어 규모에 대한 함수로써 추정하는 것으로 기본적인 모델은 식 (1)과 같다

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

식 (1)에서 E는 예측된 개발 노력으로, 공수(Man-Months)로 측정되어지고 a, b, c는 알려지지 않은 추정되어질 상수로 최소자승 추정법에 의해서 추정되어지며 KLOC(Thousands of Line Of Code)는 최종 시스템에 코딩된 라인수를 나타낸다. 둘째는 환경적인 요소의 영향에 대한 것을 추정하는 것이다. 환경적인 요인에는 프로그래머의 능력이나 하드웨어의 구성 등 다양한 요인을 고려할 수 있다. 그 외에도 LOC를 이용하여 개발 노력 E를 추정한 연구로는 Conte, Dunsmore, and Shen 등에 의해서 이루어진 연구로 연구결과는 식 (2)와 같다.

$$\begin{aligned} E &= 5.2 \times (KLOC)^{0.91} && (\text{Walston - Felix model}) \\ E &= 5.5 + 0.73 \times (KLOC)^{1.16} && (\text{Bailey - Basili model}) \\ E &= 3.2 \times (KLOC)^{1.05} && (\text{Boehm simple model}) \\ E &= 3.0 \times (KLOC)^{1.12} && (\text{Boehm average model}) \\ E &= 2.8 \times (KLOC)^{1.20} && (\text{Boehm complex model}) \\ E &= 5.288 \times (KLOC)^{1.047} \text{ for } KLOC > 9 && (\text{Doty model}) \end{aligned} \quad (2)$$

식 (2)에 제시된 모델에서 이들 모델간의 비교를 할 때 KLOC에 대한 정의는 상당히 중요한 역할을 한다. 모델에 따라서 코멘트 라인을 포함하는 경우도 있고 그렇지 않은 경우도 있기 때문이다. 그러나 이러한 LOC에 의한 개발 노력의 추정은 몇 가지 문제점을 가지고 있다. 첫째 경험자의 경험과 감에 의존하고 재현성이 부족하다는 것이다. 둘째 LOC는 코딩이 완료된 단계에서 추정이 이루어지므로 개발 초기에 예측하기가 곤란하다는 것이다. 셋째 개발기술, 개발환경, 개발언어에 매우 의존되어 있다는 것이다. 이것에 비하여 기능점수 FP를 이용한 개발 노력의 추정은 정해진 절차에 의해서 추정이 이루어지므로 신뢰성이 높다는 것이며 개발의 초기단계에서 예측이 가능하다는 것이다. 또한 기존의 연구에서 기능점수 FP는 개발기술이나 개발환경에 의존하지 않고 개발할 소프트웨어의 사양에 고유한 양을 나타낸다는 특징이 있다고 하였으나 본 논문에서 개발환경에 따른 기능점수의 평균적인 차이점정을 실시하여 본 결과 차이가 있음을 확인하였다.

이러한 LOC를 이용한 추정방법의 여러 가지 문제점으로 인하여 개발 노력을 추정하는데 있어서 기능점수를 이용한 소프트웨어 프로젝트 규모추정은 표준화된 척도로 프로젝

트 계획 중에 소요비용과 일정을 추정하는데 유용하다고 연구되어져 있다.

소프트웨어 개발에 관련된 환경외적인 변인으로는 개발 비용(Development Cost), 공수(Man-Months), 길이(LOC), 도큐멘트의 양(Documentation Size), 사용의 용이성, 유효성 등 대단히 많이 이용되고 있다.

기능점수를 이용하여 개발 노력 E를 추정한 연구 결과 몇 가지를 소개하면 식 (3)과 같다. Albert et al.[1]은 IBM Data Processing Service에서 개발된 24개의 응용프로그램에 대해 단순회귀모델을 연구하였고 Matson et al.은 이들 데이터에 대해 비선형 회귀식을 추정하였다.

$$E = -13.39 + 0.054FP \quad (3)$$

$$\sqrt{E} = 1.000 + 0.00468FP$$

Kemerer[3]은 ABC 회사에서 개발된 15개의 프로젝트를 이용하여 회귀식을 제시하였으며 그가 제안한 회귀식은 식 (4)과 같다. 그러나 이 회귀식은 표본의 수가 적고 특정업체를 대상으로 한 프로젝트에서 조사한 자료이므로 다양한 소프트웨어의 개발환경을 반영하지 못한다는 단점을 가지고 있다.

$$E = -12157 + 0.3411FP$$

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

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

Matson et al.은 대형업체로부터 조사한 104개의 프로젝트에 대한 데이터를 이용하여 모델을 제시하였다. 104개의 프로젝트들은 다양한 언어로 구성되어져 있으며 다양한 응용분야로 구성되어져 있는 데이터이다. 그가 제시한 모델은 식 (5)과 같다.

$$E = 585.7 + 15.12FP \quad (5)$$

$$\ln(E) = 2.51 + \ln(FP)$$

기존의 연구로 제시된 식 (3), 식 (4), 식 (5)의 모두는 기능점수와 개발노력의 관계로 설명되어져 있다. 그리고 이러한 기능점수를 직선회귀와 곡선회귀, 로그회귀로 나누어서 조사를 실시하고 모델을 제안하였다. 그러나 이와 같이 제시된 모델 사이에 로그변환과 같은 데이터의 변환은 일반적인 법칙에 의해서 이루어진 것이 아니라 임의적인 방법으로 변환을 하여 회귀모델을 제안하였다. 이러한 회귀식은 Box-Cox[17]의 변환공식에 의해서 변환되어져야 곡선관계에 있는 회귀관계를 간단히 직선관계로 변형할 수 있을 것이다.

즉 독립변수와 종속변수사이의 회귀관계를 확인하기 위

해서 일차적으로 두 변수사이에 플롯을 그려보고 관계를 파악한다. 두 변수의 관계가 단순 선형 회귀 관계로 나타나면 단순한 일차선형회귀에 적용하여 모델을 설정하면 된다. 그러나 두 변수의 관계가 선형관계가 아닌 곡선관계라면 독립변인과 종속변인의 관계가 오목관계인지 볼록관계인지를 파악하여 선형관계에 맞게 변수를 변환하는 것이 설명력이 높은 회귀식을 찾을 수 있는 방법일 것이다.

본 논문에서는 개발 노력과 비용을 추정하는데 있어서 단순히 기능점수만을 이용하기 보다는 개발 노력과 관계있는 변인을 모두 이용하여 회귀분석을 실시하면 회귀식의 설명력이 높아질 것이라는 가정하에 개발 노력과 관련 있는 변인들을 모두 세밀히 검토하고 단계별 변화를 확인하려 한다. 또한 이러한 결과를 이용하여 개발 노력에 영향을 미치는 변인들을 이용하여 다중회귀분석을 실시한 회귀모델을 제시하려 한다.

본 연구의 3장에서는 본 연구에 이용된 789개의 개발 프로젝트에서 조사된 데이터를 이용하여 자세히 분석하여 보고 분석된 결과를 통해서 개발 노력 추정을 위한 변수를 제시하였다.

3. 데이터에 대한 분석

소프트웨어의 개발노력이나 비용을 산정하는 것은 단순히 주어진 자료만을 이용하여 추정하기에는 상당히 어려움이 많다. 특히 데이터를 수집하였다 하여도 기존의 데이터를 면밀히 분석하지 않고 주어진 데이터를 모두 이용한다는 것도 또한 문제점이 있다고 보여진다. 본 연구에서는 이러한 문제점을 해결하기 위하여 기존에 주어진 데이터를 면밀히 분석하여 특징을 파악하려 한다. 기존의 연구에서 개발 노력을 추정하는데 있어서 LOC를 이용하면 여러 가지 환경외적인 요인에 의해서 변화가 나타나므로 이러한 단점을 극복하기 위하여 기능점수를 이용하여 개발 노력을 추정하는 것으로 되어있다. 본 논문에서는 연구에 이용된 789개 프로젝트를 중심으로 개발 노력에 영향을 미치는 변인에 대하여 세부적인 분석을 실시했고 기능점수가 환경외적인 요인에 의해서 영향을 받고 있다는 것을 확인했다. 본 장에서는 개발 형태, 개발환경, 개발언어에 따라서 기능점수(FP)나 개발 노력(E)이나 팀 규모(TS), 프로젝트인도율(PDR)에 차이가 있는지를 검정하였다. 검정방법은 통계패키지 SPSS/Win 11.0 [16]을 이용하였으며 유의수준 5%에서 검정하였다.

또한 기존의 연구가 개발 노력을 추정하는데 있어서 기능점수만을 이용한 단순회귀모델을 제안하였으므로 본 연구에서는 개발 노력을 추정하는데 있어서 기능점수(FP ; Func-

tion Point)와 프로젝트인도율(PDR ; Project Delivery Rate), 개발기간(D ; Duration), 팀 규모(TS ; Team Size), 팀 레벨(L ; Level)을 이용하여 다중회귀분석(Multiple Regression Analysis)을 실시하여 보았다.

$$E = -6350.802 + 6.13FP + 370PDR + 232D + 54TS + 950L \quad (6)$$

식 (6)에서 제안된 다중회귀모델(Multiple Regression Model)은 회귀식의 설명력이 기존의 연구에 비하여 상당히 높아짐을 결정계수(Determination Coefficient) R^2 을 통해서 확인하였다. 그러므로 소프트웨어를 개발하는데 있어서 관련 있는 외적변인에 대한 분석을 통해서 요인별로 다양한 특징을 찾아 내는 것은 중요한 연구과제로 보여진다. 먼저 개발형태(Development Type)에 따라서 기능점수(FP), 개발시간(D), 개발 노력(E), 팀 레벨(L), 팀 규모(TS)에 차이가 있는지를 검증하기 위하여 평균차이검정을 실시하여 결과를 확인하였다.

본 연구에 이용된 데이터의 개발형태라는 변인은 New Development, Enhancement, Re-development, Other로 나누어져 있다. New Development(ND)라는 것은 새로운 기능을 추가하거나 새롭게 작동하도록 기존의 시스템 전체를 교체하는 것을 의미하며 Enhancement(EH)라는 것은 개발된 소프트웨어를 새로운 기능을 추가한다거나 기존의 기능을 변경하거나 없애는 등의 작업을 하는 것을 의미하여 Re-development(RD)는 같은 소프트웨어나 하드웨어 플랫폼에서 효율성 증진을 위하여 재구성 또는 엔지니어의 재구성을 하는 과정이며 위의 것을 제외한 모든 것을 Other(O)에 포함하였다. 네 가지의 개발형태에 따라서 기능점수(FP), 개발 노력(E), 개발기간(D), 팀 레벨(L), 팀 규모(TS) 모두 평균(Mean)에 차이가 있는 것으로 조사되었다.

<표 1> 개발형태에 따른 분석분석표

개발형태 (Development Type)	F	Significant Level
기능점수 (FP)	12.054	0.000
개발 노력 (E)	2.817	0.038
개발기간 (D)	23.360	0.000
팀레벨 (L)	5.523	0.001
팀규모 (TS)	4.256	0.006

<표 1>의 결과에서 개발 형태(Development Type)에 따라서 기능점수(FP), 개발 노력(E), 개발기간(D), 팀 레벨(L), 팀 규모(TS) 모두 평균(Mean)에 차이가 있는 것으로 나타났다. 개발 형태에 따른 평균값은 <표 2>에 수록되어져 있다.

<표 2>는 개발형태 4가지에 따라서 5개 변인의 평균차이 검증 결과이다. 표에서 팀 레벨(L ; Resource Level)은 4단계로 데이터를 수집한 사람에 따라서 분류하였다. 레벨 1은 개발팀에서 업무를 담당한 것으로 구성은 Project Team, Project Management, Project Administration이며, 레벨 2는 개발팀 보조자로서 팀구성은 Data Base Administration, Data Administration, Quality Assurance, Data Security, Standards Support, Audit & Control, Technical Support이며, 레벨 3는 작동보조로서 작동환경의 유지관리를 하는 책임을 담당하는 것으로 구성은 Software Support, Hardware Support, Information Centre Support와 레벨4는 사용자 보조로서 Application Users/Client, User Liaison, User Training으로 구성 되어져 있다.

<표 2> 개발형태에 따른 평균(Mean)값

개발형태 (Development Type)	Mean	
	레벨	Mean
기능점수 (FP:Function Point)	EH	388.64
	ND	845.73
	O	864
	RD	1062.62
개발노력(E:Effort)	O	2087
	EH	5074
	RD	13160
	ND	13408
개발기간(D:Duration)	EH	7.33
	O	10.50
	ND	11.87
	RD	19.79
팀레벨 (L:Resource Level)	EH	1.89
	ND	1.99
	RD	2.64
	O	3.00
팀규모(TS:Team Size)	RD	4.52
	EH	5.74
	ND	5.75

다음은 개발 환경(Development Environment)에 따라서 평균에 대한 차이 검정을 실시하여 보았다. 여기서 개발 환경은 Mainframe(MF), Midrange(MR), PC로 3가지 종류로 나누어져 있다. 이러한 개발환경에 따라서 각 변인들은 수준별로 어떤 차이가 있는지를 검증한 결과는 아래와 같다.

<표 3>은 개발 환경(Development Environment)인 PC, Mainframe(MF), Midrange(MR)에 따라서 각 변인간의 평균 차이 검정을 실시한 결과이다. 기능점수는 개발 환경에

따라서 평균에 차이가 있는 것으로 나타났는데 개발 노력을 산정하는데 사용되어지던 기능점수는 MR인 경우에 가장 크게 조사되었으며 PC인 경우에 가장 작은 것으로 조사되었다. 개발 노력도 개발 환경에 따라 차이가 있는 것으로 나타났는데 MR인 경우에 가장 크고 PC인 경우에 가장 작다는 것을 알 수 있다. 그리고 팀 규모(TS)에 있어서도 차이가 있는 것으로 조사되었으며 대체적으로 PC에서는 4명 정도 MR인 경우는 5명, MF인 경우는 6명 정도로 조사되었다. 그리고 프로젝트인도율(PDR)의 경우도 개발 환경에 따라서 평균에 차이가 있는 것으로 조사되었으며 MR이 가장 크고 PC가 가장 작다는 것을 알 수 있다. 개발 환경에서는 PC에서 하는 개발작업이 가장 소규모라는 것을 알 수 있으며 MR에서 가장 큰 개발 프로젝트가 이루어지고 있음을 알 수 있다.

<표 3> 개발환경에 따른 평균차이 검정

개발환경 (Development Enviroment)	F	Significant Level
기능점수 (FP)	5.135	0.016
개발노력 (E)	9.711	0.000
개발기간 (D)	1.923	0.147
팀레벨 (L)	0.913	0.402
팀규모 (TS)	6.316	0.002
프로젝트인도율 (PDR)	7.145	0.001

<표 4> 개발환경에 따른 평균(Mean) 값

개발환경 (Development Enviroment)	Mean	
기능점수 (FP ; Function Point)	PC	548.26
	MF	643.02
	MR	964.71
개발노력(E ; Effort)	PC	4413.88
	MF	7531.30
	MR	23965.06
개발기간(D ; Duration)	PC	9.02
	MF	10.71
	MR	11.56
팀레벨(L ; Resource Level)	PC	2.08
	MF	2.02
	MR	2.18
팀규모(TS ; Team Size)	PC	4.33
	MR	5.54
	MF	6.13
프로젝트인도율 (PDR ; project Delivery Rate)	PC	8.04
	MF	12.98
	MR	17.22

다음은 소프트웨어를 개발하는 개발 언어(Development Language)에 따라서도 각 변인간에 평균적인 차이가 있을 것으로 예상하여 평균차이검정을 실시하였다.

<표 5> 개발언어에 따른 평균차이 검정

개발언어 (Development Language)	F	Significant Level
기능점수(FP)	1.046	0.372
개발노력(E)	3.674	0.012
개발기간(D)	0.825	0.480
팀레벨(L)	4.324	0.005
팀규모(TS)	0.157	0.925
프로젝트인도율(PDR)	15.476	0.000

<표 6> 개발언어에 따른 평균(Mean) 값

개발언어 (Development Language)	Mean	
기능점수 (FP ; Function Point)	2GL	304.25
	3GL	619.64
	4GL	627.72
	ApG	871.23
개발노력(E ; Effort)	2GL	1397
	4GL	4927
	ApG	7508
	3GL	8172
개발기간(D ; Duration)	2GL	5.67
	4GL	9.20
	3GL	10.18
	ApG	10.23
팀레벨(L ; Resource Level)	2GL	1.75
	4GL	1.76
	3GL	2
	ApG	2.32
팀규모(TS ; Team Size)	2GL	4
	4GL	5.16
	3GL	6
	ApG	6.325
프로젝트인도율 (PDR ; Project Delivery Rate)	4GL	8.82
	2GL	11.21
	ApG	11.81
	3GL	14.699

2GL은 2세대 언어를 의미하고 3GL, 4GL은 각각 3세대 4세대 언어를 의미한다. 그리고 ApG는 응용생성언어를 의미한다. <표 5>의 결과를 보면 프로젝트 개발 언어에 따라서 개발 노력(E), 팀 레벨(L), 프로젝트인도율(PDR)에 차이가 있는 것으로 조사되었다. <표 6>의 결과를 통해서 알 수 있는 것은 2세대 언어인 2GL로 이루어진 프로젝트인 경우

는 샘플 수가 적어서 분석에서 제외한다면 4세대 언어인 4GL이 프로젝트인도올(PDR)이나 개발 노력(E), 개발기간(D), 팀 규모(TS) 모두에서 평균값이 가장 최소인 것으로 조사되어 가장 이상적인 언어로 보여지고 있다. 다음은 프로젝트를 하는데 있어서 가장 이상적인 팀 규모(TS)를 예측하기 위하여 이상적인 팀의 평균에 대한 차이점정을 실시하여 보았다. 본 연구에 사용한 789개의 프로젝트는 팀 규모(TS)에 있어서 1명부터 최고 468명 규모로 구성되어 있으므로 13명 이상으로 구성되어진 팀의 규모를 하나로 묶어서 분석을 실시하여 보았다. 먼저 팀의 규모에 대한 빈도분석(Frequency Analysis)의 결과이다.

<표 7> 팀 규모(TS)에 대한 빈도분석표

팀 규모	빈도	유효퍼센트(%)
1	22	6.5
2	52	15.4
3	47	13.9
4	42	12.4
5	37	10.9
6	24	7.1
7	21	6.2
8	19	5.6
9	9	2.7
10	12	3.6
11	14	4.1
12	6	1.8
그외	33	9.8
무응답	451	

주) 유효퍼센트는 무응답을 제외하고 응답자의 빈도를 중심으로 응답자의 비율을 산정한 것임.

본 연구 분석에 이용된 789개 프로젝트는 팀의 규모에 응답한 338개의 데이터를 중심으로 팀 규모(TS)에 대한 빈도분석을 실시하여 본 결과 대체적으로 2~5명이라는 응답 비율이 가장 높은 것으로 조사되었다. 이것으로 프로젝트를 하는데 있어서 팀 구성은 대체적으로 2~5명으로 이루어지고 있음을 알 수 있다. 프로젝트는 일반적으로 프로젝트의 규모에 따라서 팀 규모(TS)가 달라질 것이다. 그러므로 프로젝트의 크기를 알 수 있는 기능점수 FP에 대한 빈도분석을 실시하여 보았다. 또한 기능점수에 따라서 팀 규모(TS)에 대한 평균(Mean)과 중앙값(Median)을 확인하여 보았다. 이와 같이 프로젝트의 크기를 나타내는 기능점수에 따른 팀 규모(TS)의 평균(Mean)과 중앙값(Median)을 조사함으로써 가장 이상적인 팀의 크기를 프로젝트의 크기에 따라서 예측할 수 있을 것이다.

<표 8>은 기능점수(FP)에 따라서 789개 프로젝트에 대한 빈도분석을 실시한 결과이다. 기능점수(FP)를 나누기 위한 기준이 따로 연구되어 있지 않으므로 샘플수를 고려하여 기능점수의 구간을 나누었다. 또한 <표 8>에서 제시한 기능점수의 구간을 이용하여 각 구간별 기능점수(FP)와 팀 규모(TS) 사이에 어떤 관계가 있는지를 확인하기 위하여 교차분석(Crosstabs Analysis)을 실시하여 보았다.

<표 8> 기능점수(FP)에 따른 빈도분석표

기능점수(FP)	빈도	퍼센트
1~100	103	13.6
101~200	156	19.8
201~300	128	16.2
301~400	86	10.9
401~500	50	6.3
501~700	56	7.1
701~1000	64	8.1
1001~2000	85	10.8
2001~	57	7.2
합계	789	100

<표 9> 기능점수(FP)와 팀 규모(TS)에 대한 교차빈도분석

FP \ SIZE	1	2	3	4	5	6	7	8	9	10	11	12	13	합계
0~100	11	9	7	6	4	4	2	1		2	2		1	49
101~200	5	12	7	5	7	6	1	2	1	3	1		2	52
201~300	3	12	11	10	3	3	5	3	2	2	2	2	1	59
301~400	2	8	4	4	9	4	1	4	3	1	1	0	3	44
401~500	1	2	4	3	1	2	3	2					1	19
501~700			2	4	2	3	1	1	1		3	1	3	21
701~1000		5	5	4	3	1	5			1		1	2	27
1001~2000		4	6	6	7	1	3	1	2	1	5	1	9	46
2001~			1		1			5		2		1	11	21
합계	22	52	47	42	37	24	21	19	9	12	14	6	33	338

<표 10> 기능점수(FP)와 팀 규모(TS)의 교차분석표

χ^2	p-value
183.430	0.000

<표 10>의 기능점수(FP)와 팀 규모(TS)에 대한 교차분석의 결과를 보면 유의수준 0.05에서 두 변수간에는 상관관계가 있다는 것을 확인할 수 있다. <표 9>의 결과표를 살펴보면 기능점수(FP)가 작은 경우에 팀 규모(TS)는 대체적

으로 작은 값 쪽에 분포되어 있으나 팀 규모(TS)는 기능점수(FP)의 크기에 비하여 상대적으로 큰 변화가 없음을 확인할 수 있다. <표 9>를 참조하면 기능점수 FP의 값에 따라서 대체적으로 팀의 규모는 2~5명 정도가 가장 이상적으로 구성되어져 있음을 확인할 수 있다.

다음은 <표 8>에서 제시한 기능점수(FP) 구간에 따른 구간별 팀 규모(TS)에 대한 평균(Mean)과 중앙값(Median)을 조사하였다. 일반적인 자료분석의 측면을 자료가 어떤 형태로 있는가에 대한 분석으로 연구한 존 튜키의 EDA적 측면에서 본 자료를 분석한다면 이상치에 영향을 받지 않는 중앙값을 팀 규모(TS)를 예측하는 통계적인 척도로 사용하는 것이 더욱 정확한 예측을 할 수 있을 것으로 보여진다. 존튜키의 연구는 일반적인 기술통계학적(Descriptive Statistics)인 측면이 아니므로 자료분석에 있어서 상당히 다른 견해를 줄 수 있다. 본 연구에서도 이러한 측면을 고려하여 팀 규모(TS)에 대한 기술통계값을 구해본 결과 최저 1에서 최고 468명까지로 분포되어 있음을 확인하였으며 팀의 규모에 있어서 상당히 크게 조사되어진 프로젝트의 경우에 이러한 값의 출처가 코딩상의 오류인지 응답자의 기술상 오류인지를 밝힐 수 없었다. 본 연구에 이용된 데이터만 하더라도 이와 같이 이상치에 의해서 많은 영향을 받게 되므로 대표값으로 중앙값(Median)을 이용하는 것이 분석에 있어서 오류를 최소로 줄일 수 있을 것으로 보여진다.

<표 11> 기능점수(FP)에 따른 팀의 평균(Mean)과 중앙값(Median)

FP	Mean	Median	N
1~100	4.04	3	49
101~200	5.38	4	52
201~300	4.91	4	59
301~400	5.81	5	44
401~500	5.42	4	19
501~700	8.61	6	21
701~1000	5.56	4	27
1001~2000	9.76	5.5	46
2001~	38.23	13	21
합 계	8.02	5	338

<표 11>에서 기능점수(FP)가 2000 이상인 구간은 다른 구간에 비하여 상당히 팀 규모(TS)가 큰 것으로 조사되었다. 이것은 기능점수 FP가 2000이상인 경우에 팀 규모(TS)에 대한 값으로 본 데이터에서 50, 55, 468이란 이상치 값

이 검색되었기 때문에 나타나는 현상으로 보여진다. 위의 이상치 값을 고려하면 이 구간에서의 중앙값 13은 평균값 38.23보다 팀 규모의 대푯값으로 상당히 유효한 값으로 보여진다.

다음은 프로젝트인도율(PDR)에 따라서 팀 규모(TS)에 대한 평균(Mean)과 중앙값(Median)을 조사하여 보았다. 프로젝트인도율(PDR)에 따른 팀 규모(TS)는 상당한 차이가 있을 것이라는 예상하에 조사한 결과표이다. <표 12>에 나타난 결과를 살펴보면 프로젝트인도율(PDR)이 3이하인 경우의 프로젝트는 대체적으로 팀 규모(TS)가 2, 3명 정도가 적당한 것으로 조사되었으며 프로젝트인도율(PDR)이 4~6 정도에서는 팀 규모(TS)가 4명 정도가 이상적인 것으로 조사되었고 프로젝트인도율(PDR)이 6이상에서는 팀의 규모가 6명정도가 적당한 것으로 보여진다. <표 12>의 결과에서 프로젝트인도율(PDR)의 변화값에 비하여 팀 규모(TS)의 변화는 크지 않다는 것을 쉽게 알 수 있으며 기존의 연구에서 제안한 팀의 규모는 2~5명 정도가 적절한 것으로 보여진다. <표 12>의 결과에서도 프로젝트인도율(PDR) 각 구간별 평균(Mean)은 차이가 크게 나타나고 있으나 중앙값(Median)은 차이가 없음을 알 수 있다.

<표 12> 프로젝트인도율(PDR)에 따른 팀 규모(TS)의 평균(Mean)과 중앙값(Median)

PDR	Mean	Median	N
0 < PDR ≤ 1	2.92	2	13
1 < PDR ≤ 2	2.73	3	26
2 < PDR ≤ 3	3.29	2.5	24
3 < PDR ≤ 4	4.5	4	32
4 < PDR ≤ 5	6.92	4	28
5 < PDR ≤ 6	4.88	4	25
6 < PDR ≤ 7	6.2	5	20
7 < PDR ≤ 8	5.58	6	17
8 < PDR ≤ 9	7.86	6	15
9 < PDR ≤ 10	7.55	6	18
10 < PDR ≤ 12	7.58	5	17
12 < PDR ≤ 14	8.91	7	24
14 < PDR ≤ 16	35.26	7	19
16 < PDR ≤ 18	7.09	5	11
18 < PDR ≤ 20	7.92	8	13
20 < PDR ≤ 30	10.19	6	21
30 < PDR ≤ 40	8.22	8	9
40 < PDR	17.83	18.5	6

4. 연구 결과

개발노력을 추정하기 위한 기존의 연구에서는 기능점수를 이용한 단순회귀 모델을 제시하였다. 그러나 본 연구에서는 데이터 분석을 통해서 개발 노력에 영향을 미치는 변인은 기능점수(FP ; Function Point)를 제외한 프로젝트인도율(PDR ; Project Delivery Rate), 개발기간(D ; Duration), 팀 규모(TS ; Team Size), 팀 레벨(L ; Level) 등의 변인이 통계적으로 유의함을 증명하였다. 또한 개발 노력 추정을 위해서 사용된 독립변인들은 개발형태, 개발환경, 개발언어에 따라서 통계적으로 유의한 평균 차이를 보이고 있음을 확인하였다. 또한 기능점수나 프로젝트 인도율에 따라서 팀의 규모에 평균적인 차이를 보이고 있음을 확인하였다. 즉 본 연구를 통해서 첫째, 개발 노력에 영향을 미치는 변인들을 찾아 내어 통계적으로 유의함을 증명하고 유의성이 검증된 변수를 이용하여 다중회귀분석(Multiple Regression Analysis)을 실시하여 보았다. 유의성 검증을 통해서 개발 노력(E)을 설명하기 위한 설명변수로는 기능점수(FP), 프로젝트인도율(PDR), 팀규모(TS), 개발기간 등을 고려해야 함을 제시하였다. 실시결과 회귀분석의 설명력이 향상되어짐을 확인하였다. 둘째, 개발 노력을 추정하기 위해서 이용되어지는 독립변인들은 개발형태, 개발환경, 개발언어에 따라서도 평균적인 차이를 보이고 있음을 확인하였다. 즉 개발 노력에 유의한 변인을 일차적으로 선택하고 나면 선택된 변인에 대한 값의 변화를 고려하여 모델을 제시하는 것이 개발 노력 추정에 있어서 더욱 유의한 모델을 제시할 수 있을 것임을 확인하였다.

셋째 기능점수에 따라서 팀규모를 예측하였다. 대체적으로 기능점수의 크기 변화에 비하여 팀규모는 2명에서 5명 정도가 가장 적합한 것으로 가장 높은 빈도를 보이고 있었다. 그러나 이것을 이용하여 기능점수에 따른 팀규모의 평균(Mean)과 중앙값(Median)을 추정하여 본 결과 평균값은 팀 규모 예측에 있어서 상당히 변화가 심한 것으로 조사되었다. 이것은 각 기능점수별 이상치에 상당히 영향을 많이 받고 있음을 알 수 있었다. 여기에서 팀규모를 산정하는데에는 일반적으로 사용하는 평균(Mean)이란 값을 통하여 팀 규모(TS)나 기타 프로젝트에 관련된 여러 가지 변인을 예측하기 보다는 중앙값(Median)을 이용하는 것이 이상치(Outlier)에 영향을 받지 않으므로 정확한 예측값을 추정할 수 있다는 것을 확인하였다.

본 연구를 통해서 2가지를 제시하려 한다

첫째 개발 노력을 추정하고자하는 경우에 기존의 연구에 따라서 단순히 기능점수만을 고려한 단순회귀 모델 보다는 프로젝트의 성격에 따라 개발 노력에 영향을 미치는 유의한 변수를 먼저 검증하여야 한다는 것이다.

둘째 기능점수, 프로젝트 인도율, 개발기간, 팀 규모, 팀 레벨 등 본 연구에서 다중 회귀 분석의 독립변인으로 제시한 변인들은 개발형태, 개발환경, 개발 언어에 따라서도 평균적인 차이를 보이고 있으므로 프로젝트의 성격에 따라서 선택된 독립변인에 대한 평균적인 차이검정을 통해서 세부적인 값에 따른 모델을 제시하는 것이 필요하다는 것이다. 소프트웨어에 관련된 데이터는 다소 수집하기 어려운 면이 있어 소수의 자료를 가지고 분석하는 경우가 많다. 적은 데이터를 가지고 정확한 분석을 위해서는 변인에 따른 값의 변화가 심하므로 먼저 데이터에 대한 철저한 분석을 통해서 모델을 제시하여야 만 오차가 적은 범위내에서 회귀계수를 추정할 수 있을 것이다.

5. 향후 연구과제

본 논문에서는 1990년대 20여 개국에서 조사된 789개의 소프트웨어 프로젝트에 대한 데이터베이스를 갖고 있는 ISBSG(International Software Benchmarking Standards Group) Benchmark Release 6을 이용하여 조사된 자료에 대한 분석을 실시하여 각 자료별 특징을 파악하였다.

기존 연구에서 프로젝트를 하는 과정에서 조사된 기능점수(FP)만을 이용하여 단순선형회귀모델(Simple Linear Regression Model)이나 단순곡선회귀모델(Simple Curve Regression Model)을 통해서 개발 노력(E)을 추정하였다. 그러나 본 논문에서는 개발 노력을 추정하는데 있어서 개발 노력의 설명력에 유의한 반응을 보이는 변인을 기능점수(FP) 외에 개발기간(D), 팀레벨(L), 팀규모(TS), 프로젝트인도율(PDR) 등이 통계적으로 유의함을 발견하고 유의한 변인들을 이용하여 다중회귀 모델을 제시하였다. 또한 이러한 자료를 개발 노력을 추정하는데 있어서 분석 없이 사용하였던 기존의 방법에 대한 모순을 밝히고 기능점수(FP)와 개발기간(D), 팀레벨(L), 팀규모(TS), 프로젝트인도율(PDR)등은 개발환경, 개발형태, 개발언어등의 변인에 따라서 평균적인 차이를 보이고 있음을 세분화된 분석을 통해서 밝혀냈다.

개발 노력이나 비용을 예측하는데 있어서 최소의 오차 범위내에서 정확한 추정이 되어지기 위한 모델 설정에서

이와 같이 데이터에 대한 세부적인 분석이 먼저 선행적으로 이루어져야 함을 제시하였다.

대체적으로 개발 노력을 추정하기 위해서 사용되어지는 데이터의 수가 극히 소수이기 때문에 개발 노력의 유의한 변수에 따라서 하위 구조에 따라 데이터를 구간별로 분류한다는 것은 쉬운 일이 아니다. 그러나 데이터는 여러 가지 환경요인이나 프로젝트의 성격에 따라서 다른 값을 가질 수 있으므로 구간별 세분화된 자료를 이용하는 것이 개발 노력을 추정하는데 있어서 정확한 예측 모델을 찾을 수 있다는 것이다.

본 논문을 통해서 앞으로 연구하여야 할 분야에 대하여 정리하면 첫째, 각 변인의 대표값으로 사용할 수 있는 여러 가지 요인들 중에서 소수의 자료를 보다 잘 설명할 수 있는 대표치에 대한 값을 연구하려 한다. 둘째, 본 연구와 관련된 자료는 무응답 처리된 데이터가 많으므로 이러한 자료에 대한 처리방법도 연구되어야 한다고 본다. 일반적으로 무응답 처리된 자료의 경우는 조사된 자료의 평균값이나 기타 값들로 대체하는 경우도 많이 활용되어진다. 이와 같은 방법을 통해서 무응답 처리된 자료에 대한 처리방법을 보다 깊게 연구해야 할 것이다. 셋째, 이상치에 대한 처리문제도 고려되어야 한다고 본다. 일반적으로 이상치에 대한 연구는 회귀분석분야의 연구에서 데이터의 분석을 실시할 경우에 필수적으로 시행하고 있으나 개발 노력이나 비용산정의 예측 연구에 있어서는 소수의 표본을 이용하여 예측하여야 하므로 이상치에 대한 영향을 고려하지 않고 있다고 볼 수 있다. 이러한 조사 데이터에 대하여 이상치에 대한 연구는 앞으로 계속 진행되어야 할 과제이며 이러한 이상치에 대한 처리방법도 연구되어야 할 것이다. 일단 데이터의 처리 방법에 대한 일차적인 연구가 심도있게 진행되어야 할 것으로 보여진다. 선행된 연구를 통해서 각 변인별 특성뿐만 아니라 개발 노력과 비용산정에 영향을 미치는 변인에 대한 연구도 계속적으로 이루어져야 할 것이다.

참 고 문 헌

- [1] 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.
- [2] B. W. Boehm and P. N. Papaccio, "Understanding and controlling software cost," *IEEE Trans. Software Eng.*, Vol. 14, pp.1462-1477, 1988.
- [3] C. F. Keremer, "An Empirical Validation of Software Cost Estimation Models," *Communications of ACM*, Vol.30, No. 5, pp.416-429, 1987.
- [4] C. B. Tayntor, "SIX SIGMA SOFTWARE DEVELOPMENT", Auerbach publications, 2002.
- [5] D. Meyerhoff, B. Laibarra, R. V. D. Pouw Kraan and A. Wallet, 'Software Quality and Software Testing in Internet Times,' Springer, 2002.
- [6] G. C. Low and D. R. Jeffery, "Function Point in the Estimation of the Software Process," *IEEE Trans on Software Eng.*, Vol.16, pp.64-71, 1990.
- [7] I. Jacobson, M. Christerson, et al., "Object-oriented Software Engineering. A Use Case Driven Approach," Addison-Wesley, 1992.
- [8] ISBSG, "Worldwide Software Development -The Benchmark Release 6," Victoria, Australia International Software Benchmarking Standards Group, 2000.
- [9] 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.
- [10] K. Johnson, "Software Cost Estimation : Metrics and Models," Department of Computer Science University of Calgary, Alberta, Canada, <http://sern.ucalgary.ca/courses/seng/621/W98/Johnsonk/cost.htm>, 1998.
- [11] L. A. Laranjeira, "Software Size Estimation of Object-Oriented System," *IEEE Trans. on Software Eng.*, Vol.16, pp.64-71, 1990.
- [12] L. A. Laranjeira, "Software size estimation of object-oriented systems," *IEEE Trans. Software Eng.*, Vol.67, pp.10-18, 1990.
- [13] L. Putnam and W. Myers, "Selecting the Right Team Size : Small is Beautiful," Cutter Consortium, <http://www.cutter.com/consortium/research/1998/crb981222.html>, 1998.
- [14] P. F. Velleman, "Definition and Comparison of Robust Nonlinear Data Smoothing Algorithms," *American Statistical Association*, Vol.75, pp.609-715, 1980.
- [15] P. F. Velleman and D. C. Hoaglin, 'ABC of EDA,' Duxbury Press, 1981.
- [16] 이상운, 노명옥, 이부권, "프로젝트 인도올 그룹 분할 방법을 이용한 소프트웨어 개발 노력 추정", 정보처리학회논문집, 제9-권 제2호, pp.259-266, 2002.
- [17] 원태연, 정성원, '통계조사분석,' SPSS 아카데미(고려정보산업), 2000.
- [18] 허명희, 문승화, '탐색적자료분석(EDA)', 자유아카데미, 2000.



정혜정

e-mail : jhjung@ptuniv.ac.kr

1988년 경북대학교 통계학과(이학사)
1991년 경북대학교 통계학과(이학석사)
1994년 경북대학교 통계학과(이학박사)
1995년~현재 평택대학교 정보통계학과
부교수

2001년~현재 경기도정보화촉진위원, 한국산업표준심의회 S/W
공학분야전문위원(ISO/IEC JTC1/SC7/WG6 국제표준
전문가 활동), 평택시정보화추진위원, 한국정보통신기
술협회 인증심의회위원 등

관심분야 : 소프트웨어공학, 소프트웨어 모형결정, 소프트웨어 품
질특성에 따른 평가, 소프트웨어 신뢰도 측정, 영상
처리



양해술

e-mail : hsyang@office.hoseo.ac.kr

1975년 홍익대학교 전기공학과(학사)
1978년 성균관대학교 정보처리학과(석사)
1991년 日本 오사카대학교 정보공학과
S/W공학 전공(공학박사)
1975년~1979년 육군중앙경리단 전산실
시스템분석장교

1980년~1995년 강원대학교 전자계산학과 교수
1986년~1987년 日本 오사카대학교 객원연구원
1994년~1995년 한국정보처리학회 총무이사, 논문편집위원장
1995년~2002년 한국 S/W품질연구소 소장
1999년~현재 호서대학교 벤처전문대학원 교수
2001년~현재 한국정보처리학회 부회장
2003년~현재 미국 ACIS 학회 Vice President

관심분야 : 소프트웨어공학(특히, S/W 품질보증과 품질평가,
품질감리와 컨설팅, OOA/OOD/OOP, CASE, SI),
컴퍼넌트관리, CBD기반기술, IT품질경영