

기능점수 기반 소프트웨어 개발팀 규모와 개발기간 예측 모델

박 석 규[†] · 이 상 운[†]

요 약

소프트웨어 개발 초기에 개발비용, 소요 인력과 기간을 추정하는 것은 소프트웨어공학 분야의 주요한 요소이며, 어려운 문제이다. 소프트웨어 개발 소요 인력을 추정하기 위한 대부분의 모델들은 소프트웨어 요구사항 명세서로부터 측정된 기능점수를 이용하였다. 본 논문은 기능점수에 기반하여 프로젝트에 대한 가장 실제적이고 생산적인 팀 규모와 개발기간을 선택하는 지침으로 사용될 수 있는 정보를 제공하기 위해 생산성 척도와 비용 척도를 도입하였다. 소프트웨어 개발과 유지보수를 수행한 300개 프로젝트 데이터 집합을 2개의 서브 그룹인 개발 프로젝트와 유지보수 프로젝트로 분류하였다. 생산성과 비용 척도 기준에 의해 분석한 결과 작은 팀 규모와 최소한의 개발기간이 소요되었을 때, 프로젝트가 성공적으로 수행됨을 보였다. 또한, 연구 결과에 근거하여 기능점수 규모에 따른 개발팀의 규모와 개발기간을 추정할 수 있는 모델을 제안하였다. 본 제안 모델은 소프트웨어 규모에 따라 필요한 개발팀의 규모와 개발기간 평가 기준을 제공한다.

A Predictive Model for Software Development Team Size and Duration Based on Function Point

Seok-Gyu Park[†] · Sang-Un Lee[†]

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 models estimate the development effort using the function point that is measured from the requirement specification. This paper presents optimal team size and duration prediction based on function point in order to provide information that can be used as a guide in selecting the most practical and productive team size for a software development project. We introduce to productive metrics and cost for decision criteria of ideal team size and duration. The experimental is based on the analysis of 300 development and enhancement software project data. These data sets are divide in two subgroups. One is a development project; the other is a maintenance project. As a result of evaluation by productivity and cost measured criteria in two subgroups, we come to the conclusion that the most successful projects has small teams and minimum duration. Also, I proposed that predictive model for team size and duration according to function point size based on experimental results. The presented models gives a criteria for necessary team size and duration according to the software size.

키워드: 개발팀 규모(Team Size), 개발기간(Development Duration), 소프트웨어 규모(Software Size), 기능점수(Function Point), 평가 척도(Evaluation Metrics)

1. 서 론

소프트웨어에 대한 개발기간과 비용 추정은 크게 부정확하며, 불충분한 품질을 갖고 있고, 개발 생산성은 사용자가 요구하는 수준보다 매우 느리게 향상되고 있다. Arthur[1]는 이러한 소프트웨어 개발 상황을 소프트웨어 위기(Software Crisis)라고 하였다. 이를 뒷받침하는 사례로, 대형 소프트웨어 프로젝트의 1퍼센트만이 계획된 기간과 예상 비용한도 내에서 고객을 만족시키며 완료되었으며, 대부분의 프로

젝트들은 1년 이상의 일정이 지연되고 초기 예상 비용의 2배 정도가 초과되었다[2]. 이와 같은 이유로 인해, 프로젝트 관리 측면에서 소프트웨어 개발 및 유지보수 비용을 줄이고자 체계적인 연구가 수행되고 있으며, 소프트웨어 비용산정, 소요 인력 및 개발일정을 추정하는 모델을 개발하는 계기가 되었다. 계획단계에서 이들 정보의 보다 정확한 추정은 프로젝트를 관리할 때 발생하는 다양한 의사결정, 소요 예산 및 개발인원 할당과 계약체결 여부에 신뢰할 만한 정보를 제공한다.

소프트웨어 개발에 소요되는 비용, 인력과 개발기간을 추

[†] 정 회 원 : 강원도립대학 컴퓨터응용과 교수
논문접수 : 2003년 5월 6일, 심사완료 : 2003년 7월 21일

정하기 위해서는 소프트웨어 규모(Size)를 측정해야만 한다. 소프트웨어 규모를 측정하기 위한 대표적인 척도로 LOC(Lines Of Code)와 FPA(Function Point Analysis)가 있다. 요구사항 명세서(Requirement Specification)로부터 추정된 기능점수 또는 코딩 결과로부터 정확하게 측정되는 LOC에 대한 소프트웨어 규모(Size)를 이용해 개발에 소요될 인력(Effort, E)과 개발기간(Duration, D)을 추정함과 더불어 개발 소요 비용(Cost)도 추정하였다.

많은 조직에서 제품 출시 시점 또는 구현의 문제로 인해 프로젝트를 시작하기 전에 프로젝트의 개발기간 D 가 확정되는 경우가 빈번히 발생한다. 개발기간을 추정하기 위해 Kitchenham[3]는 21개의 개발일정 모델을 제안하였다. 이 연구는 사용된 표본의 크기가 매우 작고, 1990년대 이전에 개발된 프로젝트들을 대상으로 하고 있어 현대의 복잡하고 다양한 환경에서 개발되는 프로젝트들에 이들 모델들을 적용하기 위한 정확성과 가치를 판단할 수가 없다. 최근 들어 Oigny et al.[4, 5]은 396개 소프트웨어 프로젝트에 대한 데이터를 갖고 있는 ISBSG Database Release 4[6]에 근거하여 개발 플랫폼(Main frame, Midrange와 PC)별로 개발되는 소프트웨어 프로젝트에 대한 개발노력 E 을 이용해 개발기간 D 을 추정하는 모델을 유도하였다.

소프트웨어 개발분야에서의 프로젝트 수행 팀 규모에 관한 연구로 정보시스템 개발에 관한 팀 규모 연구는 Samprevivo[7], LOC를 대상으로 연구한 사례는 Putnam et al.[8]이 있으며, 기능점수(FP, Function Point)를 대상으로 한 연구사례는 ISBSG(International Software Benchmarking Standards Group) Benchmark Release 6[9]가 있다. Putnam et al.[8]은 동일한 LOC를 대상으로 개발팀의 규모를 결정한 사례로 LOC 척도를 사용자 문제점을 가지고 있다. 이에 반해 ISBSG Benchmark Release 6[9]는 기능점수를 소프트웨어 규모 척도로 사용하여 언어와 독립적이며, 일반적으로 적용할 수 있다. 그러나 이 방법은 소프트웨어 규모별로 평균값 또는 중앙값만을 이용하여 개발팀의 규모를 제시하였다.

이상과 같이 소프트웨어를 개발하는데 요구되는 개발기간과 개발팀 규모를 결정하는 문제를 해결하는데 있어 이론적인 근거에 기초한 기준을 제시하지 못하고 있다. 따라서, 본 논문은 소프트웨어 개발시 가장 실제적이고 생산적인 팀 규모를 선택하는 지침으로 사용될 수 있는 기준을 제시하기 위해 최대의 생산성과 최소의 비용이 소요되는 기준을 적용하여 개발팀과 개발기간을 선정하였다.

2장에서는 소프트웨어 개발팀 규모와 개발기간 추정에 관한 관련 연구를, 3장에서는 생산성과 비용 기준에 근거하여 최소의 개발팀 규모와 개발기간 기준을 제시한다. 마치

막으로 4장에서는 소프트웨어 규모별로 최소의 개발팀 규모와 개발기간을 추정하는 모델을 제시한다.

2. 관련 연구 및 연구동기

소프트웨어 개발기간 D 를 추정하기 위해 프로젝트의 개발노력 E 에 근거한 많은 통계적 모델들이 제안되었다[10-13]. 업체들로부터의 확신에도 불구하고 연구 결과들은 이들 통계적 개발기간 모델들의 정확성에 관한 상반된 결과를 보였다. 그러나 주지해야만 하는 사항은 이들 연구에 사용된 데이터 크기가 매우 작고 오래된 과거 데이터들이라는 점이다. Kitchenham[11]가 거론한 8개 모델 중 7개 연구가 10년 전에 발표되었으며, Ferens와 Daly[10]는 1990년도에 발표된 연구와 1989년 또는 1990년도에 발표된 3종의 다른 평가 보고서에 대해 제시하였다. Kitchenham[11]는 20개 또는 그 이하의 프로젝트 표본으로부터 경험적으로 21개 개발기간 모델 중 12개를 유도하였으며, 6개 모델은 21~33 프로젝트 표본으로부터, 나머지 3개 모델은 46, 63과 81개 프로젝트 표본으로부터 유도하였다. Ferens와 Daly[10]는 개발기간을 추정하는 통계적 모델을 평가하는 다른 3개의 연구결과를 보였다. 이것은 소프트웨어 개발기간 추정은 이들 모델을 적절히 적용하는 것에 대해 보다 숙련을 요구함을 의미한다[4]. Oigny et al[4, 5]는 1989~1996년에 개발 완료된 396개 소프트웨어 프로젝트에 대한 데이터를 갖고 있는 ISBSG Database Release 4[9]에 근거하여 모델을 개발하였다. 개발기간 모델들은 전체 데이터 표본을 사용하는 것보다 보다 동질성을 갖는 서브 데이터에 대해 개발기간을 추정하는 모델을 개발하였다. 이들 모델들을 이용해 개발기간 D 를 추정할 경우 반드시 개발노력 E 의 규모를 추정해야만 한다. 소프트웨어 규모에 대해 투입될 개발노력의 규모를 추정하고 추정된 개발노력 규모에 대해 개발기간을 다시 추정하는 것은 추정과정에서 오차를 크게 유발시킬 수 있다.

소프트웨어 개발과 관련된 프로젝트 팀 규모에 관한 연구로, Samprevivo[7]는 정보시스템 개발에 대한 팀 규모를 결정함에 있어서 팀 구조가 가장 중요한 관점이며, 그룹의 성취도와 문제 해결 능력에 있어서 팀 요원 개개인의 만족도에 영향을 받음을 밝혔다. 팀 규모와 개별 요원의 불만족 관계에 대해 연구한 결과 팀의 규모가 증가함에 따라 팀의 개별 요원의 불만족이 증가하였다. 즉, 작은 그룹(5~7명)에서 팀의 개별요원은 가장 큰 만족을 느끼고 있으며, 큰 그룹(12~15명)에서는 가장 작은 만족을 느낌을 제시하였다. Putnam et al.[8]은 다음과 같은 기준을 제시하였다. ① 적은 규모로 올바로 선택된 팀은 성공적인 프로젝트를 위해 필수적이다.

② 실제로 큰 규모의 개발팀이 개발한 프로젝트보다 적은 팀이 적은 노력으로 보다 빨리 프로젝트를 종료한다. ③ 20 또는 보다 많은 팀 요원으로 개발된 프로젝트는 5 또는 그 이하의 요원으로 개발된 프로젝트보다 많은 노력이 필요하다. 이는 동일한 SLOC(Source Lines Of Code) 프로젝트 비교로부터 유도되었다. ISBSG Benchmark Release 6[9]는 다음과 같은 기준을 제시하였다. ① 적은 팀은 프로젝트 규모와 팀 규모간에 단순 관계를 갖고 있다. ② 2 또는 3명으로 구성된 팀은 팀 인원 당 가장 큰 규모의 프로젝트를 개발한다. ③ 항상 가장 적은 실제 팀 규모를 목표로 삼는다. ④ 팀 규모가 5명을 초과하면 생산성은 감소하며, 적은 팀이 보다 생산적이다. 5 또는 그 이상의 팀 규모로 개발된 프로젝트는 보다 적은 팀으로 개발된 프로젝트보다 기능점수 1개를 개발하는데 특히 많은 노력이 소요된다. ISBSG Benchmark Release 6[9]는 기능점수 *FP*와 개발팀 규모(*TS*, Team Size)를 알고 있는 244개 프로젝트를 대상으로 분석한 결과 <표 1>의 결과를 얻었다. <표 2>는 개발되는 소프트웨어 기능점수 규모에 따른 개발팀의 규모를 선정할 경우에 적용할 수 있는 일반화된 지침을 제공하고 있다.

<표 1> 프로젝트 규모 *FP*에 따른 개발팀 규모 *TS*

기능점수(<i>FP</i>)	표본 수	개발팀 규모(<i>TS</i>)			
		Min	Max	Median	Mean
1~100	34	1	11	3	3.9
101~200	41	1	53	4	5.8
201~300	46	1	14	4	5.0
301~400	29	1	21	5	6.2
401~500	12	2	8	5	5.1
501~600	10	4	17	7	8.4
601~700	4	3	20	4	7.5
701~800	7	3	12	4	5.0
801~900	9	2	7	3	4.4
901~1,000	4	4	17	8	9.0
1,001~1,100	4	3	6	4	4.3
1,101~1,200	4	2	12	4	5.5
1,201~1,400	9	2	16	4	6.9
1,401~1,600	7	3	65	7	15.4
1,601~1,800	4	5	8	7	6.8
1,801~2,000	3	3	25	11	13.0
2,001~3,000	8	3	26	11	12.5
3,001~4,000	3	8	15	10	11.0
4,001~5,000	3	12	50	24	28.7
5,001 이상	3	8	10	8	8.7

<표 2> 일반적인 개발팀 규모 선정

기능점수(<i>FP</i>)	개발팀 규모(<i>TS</i>) 선정		
	최 소	최 대	평 균
1~300	1	6	3~6
300~500	2	10	5
500~1,000	4	12	6~8
1,000~1,500	5	15	5~7
1,500 이상	7	25	10

또한, 프로젝트 설계의 “6명은 이상적인 프로젝트 팀 규모가 아니다”라는 주간 칼럼[14]에서는 가장 성공적인 프로젝트는 1명, 2명 또는 3명의 적은 팀으로 이루어지며, 완전한(Perfect) 팀 규모는 2명 내지 3명이다. 또한, 4명 또는 5명도 괜찮은 규모이지만 6명은 혼잡하다고 제시하고 있다. 이상운[15]은 기능점수, 생산성, 개발기간을 입력으로 하여 개발팀의 규모를 추정할 수 있는 RBF(Radial Basis Function) 망 모델을 제시하였으며, 이상운 et al.[16]은 기능점수 당 투입되는 시간과 월 평균 개발되는 기능점수를 기준으로 이상적인 개발팀의 규모를 선정한 결과 개발팀의 규모를 5명 이하로 할 때가 가장 적합함을 보였다. 이상과 같이 팀 선택 및 할당에 대한 연구 결과를 토대로 할 때, 성공을 위한 절대적 필요성보다는 실용적인 지침으로 팀의 규모를 보다 작게 하는 것은 명백한 사실이다. 또한, 실제적으로 5명 이하의 규모로 팀을 구성하면 팀 요원간 원활한 의사소통과 최대의 생산성을 보이며, 적기에, 소요 비용 범위 내에서 성공적으로 프로젝트 개발을 종료할 수 있음을 알 수 있다.

동일한 개발노력의 규모를 투입하더라도 몇 명의 팀 규모로 구성되느냐에 따라 개발기간, 소요 비용과 개발 생산성에 차이가 발생할 수 있다. 그러나 이들 제안된 모델들은 개발노력에 따른 개발기간만을 추정할 수 있으며, 개발팀의 규모에 대한 지침을 제공하지 못하고 있다. 따라서, 소프트웨어 규모별로 몇 명의 팀 규모로 얼마만한 기간내에 개발을 완료하는 것이 최적인가에 대한 기준이 제시되어 있을 경우 새로운 소프트웨어 프로젝트를 착수할 때 지침으로 적용이 가능하다. 3장에서는 이에 대한 평가기준을 제시한다.

3. 최소의 개발팀 규모와 개발기간 평가척도와 기준

소프트웨어를 개발하는데 소요되는 노력(시간)과 개발기간, 개발 팀의 규모간에는 식 (1)과 같이 밀접한 관계가 있어 보인다.

$$\text{노력량} = \text{개발기간} \times \text{개발팀 규모(인원)} \quad (1)$$

이 수식에 의하면 소프트웨어를 개발하는 전 과정에 걸쳐 개발팀의 규모가 일정한 일양분포(Uniform Distribution)를 한다는 것으로 잘못 생각할 수 있다. 그러나 소프트웨어의 개발에 투입되는 인력 분포(Manpower Distribution)는 일양분포를 따르는 것이 아니라 Putnam[8]의 Rayleigh나 Pillai et al.[17]의 Gamma 분포를 따른다. 따라서, 소프트웨어 개발에 투입되는 노력의 양, 개발기간과 개발팀의 관계를 단순히 유도하는데 어려움이 따른다. 본 장에서는 소프트웨어

의 기능성과 복잡도를 고려하여 계산된 기능점수에 기반하여 가장 적게 소요되는 개발기간과 최소의 개발팀의 규모를 판단하는 기준을 제시한다.

3.1 평가척도 및 기준

개발 대상인 프로젝트 규모가 결정되면 이 프로젝트를 개발하는데 소요되는 개발팀의 규모와 개발기간을 결정하는 문제가 발생한다. 소프트웨어를 개발할 경우, 개발요원의 생산성이 높아야 하며, 이와 더불어 개발에 소요되는 기간이 최소가 되면서, 개발에 투입되는 인력도 적어야 개발에 소요되는 직접 인건비도 적게 된다. 이 기준은 개발 제품에 대한 시장성도 확보할 수 있는지를 판단하는 기준이 된다. 따라서, 최대의 생산성과 최소의 개발인원과 기간이 적용될 수 있는 기준 결정이 필요하다. 이상적인 개발팀의 규모를 결정하기 위해 ISBSG Benchmark Release 6[9]는 프로젝트 규모별로 실제 투입된 개발팀 규모의 평균값 또는 중앙값을 취하였다. 이에 비해 본 논문은 생산성 척도와 개발 소요기간 기준을 적용함으로써 보다 이론적인 근거를 바탕으로 최소의 개발팀 규모와 개발기간을 결정한다. 소프트웨어 분야에 대한 대부분의 연구에서는 생산성을 일반적으로 투입된 단위 노력당 생성된 결과물의 비율로 표현된다. ISBSG Benchmark Release 6[9]는 소프트웨어 기능을 사용자에게 양도하는 비율인 프로젝트 인도율(*PDR*, Project Delivery Rate)을 개발노력의 계수로 측정하였다. 즉, 소프트웨어의 규모인 기능점수 *FP*를 개발하는데 투입된 노력 *E*로 정의하였으며, E/FP 로 계산된다. 이는 기능점수 *FP* 1개를 개발하는데 소요되는 개발노력(시간)이다.

소프트웨어 개발에 소요된 시간 *D*와 개발에 참여한 팀 규모 *TS*를 곱하면 이 척도는 프로젝트를 1명으로 수행한 총 개발기간이 되며, 개발에 소요되는 직접 인건비를 산출할 수 있는 기준이 된다. 따라서, 최소의 개발팀 규모와 개발기간을 선정하는 기준으로 생산성 측면에서 *PDR*과 제품의 시장성 확보 측면과 소요비용 측면에서의 1인당 총 개발기간 척도를 사용하여 최소의 개발팀 규모와 개발기간을 평가척도로 적용한다.

평가척도 1 : 생산성 평가척도로 프로젝트 인도율 *PDR*을 적용한다.

평가척도 2 : 소프트웨어를 개발하는데 필요한 1명당 개발기간을 적용한다.

최대의 생산성과 최소의 비용을 투입하여 성공적인 소프트웨어 개발을 위해서는 위에서 제시된 2가지 평가 척도를 이용하여 다음의 평가기준을 적용한다.

평가기준 : 소프트웨어 규모별로 개발팀의 *PDR*과 1명으로 수행한 총 개발기간 모두가 최소가 되는 기준으로 개발팀의 규모와 개발기간을 선정한다.

COCOMO II는 소프트웨어 개발에 소요되는 시간과 개발팀의 규모를 선정할 때 개발 팀원의 능력, 소프트웨어 복잡도, 개발환경 등 개발환경변수를 고려하였다. 이는 소프트웨어의 규모를 단순히 길이로서만 측정하는 SLOC를 취하고 있기 때문이다. 그러나 기능점수 기법은 소프트웨어의 기능을 구성하는 5개 요소인 입력, 출력, 조회, 인터페이스와 파일 각각에 대한 복잡도를 고려하고, 여기에 다시 14가지의 기술적 복잡도 조절 요인(COCOMO II 모델의 개발환경변수와 유사)을 곱하여 최종적으로 기능점수를 얻기 때문에 기능점수와 개발기간, 개발팀 규모의 단순한 관계로서도 적합한 모델을 얻을 수 있다. 또한, Briand et al.[18]는 기술적 복잡도 조절요인을 고려하지 않아도 소프트웨어 개발 노력 추정에는 별다른 영향을 미치지 않고 있어 최근 들어서는 기술적 조절요인을 고려하지 않은 기능점수를 계산하는 방식을 채택하고 있다. 그러므로 소프트웨어를 개발하는데 복잡한 환경요인을 고려하지 않고 소요되는 노력의 양과 개발기간, 개발팀 규모의 단순 관계를 유도가 가능할 것이다.

3.2 평가 방법

최소의 개발팀 규모와 개발기간 평가 기준에 따라 다음 순서로 평가한다.

Step 1 : *FP* 규모별로 투입된 개발팀의 *PDR*과 1인당 총 개발기간을 계산한다.

Step 2 : 계산된 *PDR*과 1명당 총 개발기간에 대해, 소프트웨어 규모별로 동일한 개발팀 규모와 개발기간에 대한 데이터들의 평균 값을 취한다.

*PDR*과 1인당 개발기간 *D* 평가척도를 이용할 경우 *PDR*과 *D*값의 범위에 상호 차이가 발생하므로 동일한 범위로 변환시켜 표준화된 비교기준이 필요하다. 따라서, 2개 변수를 0~1사이의 값을 갖도록 정규화(Normalization)시켜 동일한 범위를 갖도록 한다.

Step 3 : 소프트웨어 규모별 개발팀 규모와 개발기간에 대한 *PDR*과 *D*를 정규화시킨 *NPDR*과 *ND*를 구한다.

*NPDR*은 다음과 같이 구해진다. *i*번째 데이터의 *PDR* 값을 PDR_i , $i=1, 2, \dots$ 라 하고, 최대 *PDR* 값을 갖는 데이터를 PDR_{max} 라 하면, *i*번째 데이터의 정규화된 *PDR* 값은

$$NPDR_{m_i} = \frac{PDR_i}{PDR_{max}}$$

로 계산된다. *D*에 대해서도 동일한 방법으로 정규화된 값인 *ND*가 구해진다.

마지막으로, *PDR*이 최소가 되고 1인당 총 개발기간도 최소가 되는 개발기간과 개발팀 규모를 선정하는 평가 기준에 따라 다음의 값을 계산한다.

Step 4: *NPDR*과 *ND*를 합한 값이 가장 작은 값을 가진 개발팀과 개발기간이 이상적인 개발팀의 규모와 개발기간이 된다.

3.3 실험 및 결과

ISBSG Benchmark Release 6[9]는 1990년대에 20여 나라에서 다양한 언어, 개발기법 및 적용분야에서 개발된 789개 프로젝트들에 대한 데이터를 갖고 있는 방대한 데이터베이스이다. 789개 프로젝트 데이터들 중에 *PDR*과 1인당 총 개발소요기간을 계산하기 위해 필요한 데이터를 모두 갖고 있는 프로젝트는 300개로 이들 데이터를 실험에 사용한다. 개발기간은 월(Month) 단위로, 개발노력은 시간(Time) 단위를 사용하고 있다. 300개의 데이터를 분석하여 보면, 신규로 개발되는 프로젝트(New Development Project)와 재개

발 프로젝트(Re-Development Project)로서 개발단계에 적용되는 프로젝트는 212건, 운영단계에서 유지보수 활동 중 새로운 기능을 추가하는 완전 또는 적용 유지보수를 수행한 프로젝트(Enhancement Project)가 88개이었다. 신규로 개발되는 프로젝트를 개발 프로젝트라 칭하며, 완전 유지보수 프로젝트를 유지보수 프로젝트라 하자. 신규로 개발되는 프로젝트와 유지보수 프로젝트는 개발팀의 생산성, 소프트웨어 규모와 개발기간에 차이가 발생할 수 있다. 따라서, 개발 프로젝트와 유지보수 프로젝트로 구분하여 생산성과 비용을 평가함으로써 이상적인 개발팀 규모와 개발기간의 기준을 제시하고자 한다.

Step 1과 Step 2의 평가방법에 의해 기능점수 범위가 1~100인 20개의 유지보수 프로젝트를 대상으로 계산된 *PDR*, 1명당 개발기간 *D*의 예는 <표 3>에 제시되어 있다.

88개 유지보수 프로젝트 전체에 대해 분석한 결과 *PDR*의 최대값은 59.4286, 1인당 총 개발기간 *D*의 최대값은 360.00이었다. <표 3>의 FP 범위가 1~100인 20개 유지보수 프로젝트들에 대해, 동일한 팀 규모와 동일한 개발기간을 가지고 있는 프로젝트에 대해 평균을 취한 결과는 <표 4>에 제시하였다.

<표 3> 유지보수 프로젝트의 *PDR*과 개발기간 예

FP 범위	팀 규모	개발기간 (월)	기능점수 FP	개발노력 E (시간)	생산성 PDR	1인당 개발기간 (월)
1~100	1	2	44	380	8.6364	2.00
		2	64	140	2.1875	2.00
		3	40	285	7.1250	3.00
		4	38	247	6.5000	4.00
		6	25	580	23.2000	6.00
	2	4	49	652	13.3061	8.00
		4	96	318	3.3125	8.00
		4	97	654	6.7423	8.00
	3	1	39	114	0.0944	3.00
		7	42	2,496	59.4286	28.00
	5	3	32	352	5.2321	15.00
		3	56	293	8.1161	15.00
		7	45	281	6.2444	35.00
	6	2	33	145	4.3939	12.00
		12	80	2,595	32.4385	72.00
	7	3	54	348	6.4444	21.00
	10	4	89	442	4.9663	40.00
	11	8	60	471	7.8500	88.00
		9	97	729	7.5155	99.00

<표 4> 유지보수 프로젝트의 개발팀 규모와 개발기간 평가

FP 범위	팀 규모	개발기간	평균		평가 결과			순위
			PDR	D	NPDR	ND	계	
1~100	1	2	5.4119	2.00	0.0911	0.0056	0.0966	2
		3	7.1250	3.00	0.1199	0.0083	0.1282	-
		4	6.5000	4.00	0.1094	0.0111	0.1205	-
		6	15.4226	6.00	0.2595	0.0167	0.2762	-
	2	4	7.7870	8.00	0.1310	0.0222	0.1533	4
	3	1	0.9744	3.00	0.0164	0.0083	0.0247	1
	4	7	59.4286	28.00	1.0000	0.0778	1.0778	9
	5	3	8.1161	15.00	0.1366	0.0417	0.1782	6
		7	6.2444	35.00	0.1051	0.0972	0.2023	-
	6	2	4.3939	12.00	0.0739	0.0333	0.1073	3
		12	32.4375	72.00	0.5458	0.2000	0.7458	-
	7	3	6.4444	21.00	0.1084	0.0583	0.1668	5
	10	4	4.9663	40.00	0.0836	0.1111	0.1947	7
	11	8	7.8500	88.00	0.1321	0.2444	0.3765	8
		9	7.5155	99.00	0.1265	0.2750	0.4015	-

표에서 동일한 개발기간, 소프트웨어 규모와 개발팀에 대해 *NPDR*과 *ND*의 합(계)이 가장 작은 값이 최우선 순위가 된다. 따라서, FP 범위가 1~100인 유지보수 프로젝트에 대해서는 3명의 팀 규모로 1개월의 개발기간을 설정하여 소

소프트웨어를 유지보수 하는 것이 가장 적합한 것으로 판단된다. 개발 프로젝트도 유지보수 프로젝트에 적용된 방법과 동일하게 평가되었다. 개발 프로젝트들은 PDR의 최대 값은 77.74, 1인당 총 개발기간 D의 최대 값은 7020.00의 특성을 갖고 있다.

유지보수 프로젝트와 개발 프로젝트에 대한 소프트웨어 규모별로 NPDR과 ND가 계산되었다. 소프트웨어 규모별로 평가된 개발기간과 개발팀의 규모는 <표 5>와 <표 6>에 제시되어 있으며, 표에서 음영으로 처리된 부분이 최소의 개발기간과 개발팀의 규모가 된다. 개발 프로젝트는 19개의 소프트웨어 프로젝트 규모별로 볼 때, 5명 이하 투입(14개), 6명(1개), 8명(2개), 12명(1개)로 1순위를 차지하였다. 따라서, 5명 이하의 개발팀을 투입하는 것이 가장 적합함을 알 수 있다. 개발기간 측면에서는 $1 \leq FP \leq 900$ 는 6개월 이내, $901 \leq FP \leq 1400$ 는 9개월, $1401 \leq FP \leq 5000$ 은 12~18개월, $5001 \leq FP$ 은 3년 정도의 개발기간이 최소가 됨을 알 수 있다.

유지보수 프로젝트에 대해서는 14개의 프로젝트 규모 범위 기준으로 볼 때, 개발팀의 규모를 살펴보면 5명 이하(9건), 7명(1건), 11명(1건), 13명 이상(2건)으로 5명 이하가, 또한 개발기간 측면에서는 1년 이내에 완료하는 것이 가장 적합함을 알 수 있다.

따라서, 일반적으로 다음과 같은 결론을 얻을 수 있다.

결론 1: 개발 대상 소프트웨어의 규모와 개발기간이 확정되었을 경우, <표 5>와 <표 6>의 정보를 이용하면 최고의 생산 성과 개발기간 데이터로부터 개발에 참여할 개발팀의 규모와 개발기간을 판단할 수 있는 기준 정보로 활용된다.

결론 2: 본 제안된 기준을 적용할 경우, 소프트웨어 개발 업체 선정시 개발업체의 생산성 척도인 PDR 값과 투입될 개발팀의 규모만 알 수 있으면 개발 사업에 참여하는 업체의 수준을 평가할 수 있는 기준으로 활용될 수 있다.

4. 최소의 개발팀 규모와 개발기간 추정 모델

<표 5>의 개발 프로젝트와 <표 6>의 유지보수 프로젝트에 대해 기능점수 규모별로 최소의 개발팀 규모와 개발기간을 요약하면 <표 7>과 같다.

개발 프로젝트와 유지보수 프로젝트의 소프트웨어 규모별 개발팀 규모 제안 결과와 ISBSG Release 6[9]의 결과를 (그림 1)에 비교하였다. ISBSG Release 6[9]는 프로젝트 특

<표 5> 개발 프로젝트의 개발팀 규모와 개발기간 선정 결과

FP	구분	개발팀 규모(명)														
		1	2	3	4	5	6	7	8	9	10	11	12	13	13이상	
1	D	1	1	1	2	3	8	5			7					
	PDR	0.539	3.279	8.976	1.241	5.286	10.855	24.125			12.958					
	결과	0.0071	0.0425	0.1159	0.0171	0.0701	0.1466	0.3153			0.1766					
	순위	1	3	5	2	4	6	8			7					
101	D	1	2	6	7	7	2	10	8	6	8					6
	PDR	0.291	1.438	4.385	10.962	13.353	6.979	20.099	18.723	15.636	8.214					77.747
	결과	0.0039	0.0191	0.0591	0.0450	0.1787	0.0915	0.2885	0.2442	0.2088	0.1170					1.0111
	순위	1	2	4	3	7	5	10	9	8	6					11
201	D	5	3	6	7		4	7	4	4		6	6			
	PDR	0.783	1.277	3.330	4.383		6.298	3.645	10.8495	5.439		18.986	8.084			
	결과	0.0108	0.0173	0.0654	0.0604		0.0844	0.0539	0.1440	0.0750		0.2536	0.1442			
	순위	1	2	3	5		7	4	8	6		10	9			
301	D	7	6	8	11	11	2	7	9			4				7
	PDR	1.566	1.754	1.183	2.027	6.057	0.833	7.174	3.833			4.914				11.867
	결과	0.0211	0.0243	0.0186	0.0323	0.0859	0.0124	0.0892	0.0596			0.0885				0.1706
	순위	3	4	2	5	8	1	9	6			7				10
401	D				13	12	12		8	24	8					
	PDR				1.972	5.705	4.562		5.493	21.289	19.099					
	결과				0.0291	0.0785	0.0555		0.0775	0.2375	0.2548					
	순위				1	4	2		3	6	5					
501	D					5		6	18	14		6	10			10
	PDR					1.354		4.258	6.679	18.518		17.416	32.672			8.432
	결과					0.0203		0.0599	0.1039	0.2541		0.2334	0.4373			0.1441
	순위					1		2	3	6		5	7			4
601	D				13	3		8				18				11
	PDR				1.129	1.510		3.410				7.560				14.874
	결과				0.0201	0.0211		0.0537				0.1253				0.2226
	순위				1	2		3				4				5
701	D		24	26	16	12										
	PDR		4.252	4.335	3.861	5.772										
	결과		0.0615	0.0669	0.0588	0.0830										
	순위		2	3	1	4										
801	D		6	18			20	13								
	PDR		2.391	3.396			9.304	19.124								
	결과		0.0325	0.0514			0.1368	0.2598								
	순위		1	2			3	4								
901	D					30					11					27
	PDR					3.757					9.511					29.989
	결과					0.0897					0.1390					0.4865
	순위					1					2					3
1001	D		15	8	12	9	9							16	6	
	PDR		5.706	1.317	3.456	3.129	6.645							9.954	13.937	
	결과		0.4496	0.0777	0.0513	0.0467	0.0832							0.1531	0.1885	
	순위		7	3	2	1	4							5	6	
1201	D		9	5	11	18					25	12	15			38
	PDR		0.484	0.934	2.344	4.080					18.939	12.809	8.237			54.456
	결과		0.0087	0.0141	0.0364	0.0554					0.2757	0.1818	0.1294			0.8736
	순위		1	2	3	4					8	7	6			5
1401	D		18	27	18	23			4					16	12	
	PDR		5.646	4.441	3.410	12.282			4.008					10.147	15.674	
	결과		0.0777	0.0367	0.0541	0.1744			0.0555					0.1601	0.2409	
	순위		4	3	1	6			2					5	7	
1601	D						16	12			30					
	PDR						4.614	4.321			0.243					
	결과						0.0753	0.0692			0.0815					
	순위						2	1			3					
1801	D			15							10					7
	PDR			0.663							0.243					6.765
	결과			0.0149							0.0815					0.1070
	순위			1							2					3
2001	D			42		9				11				16	9	
	PDR			1.119		8.194				3.847				6.982	9.966	
	결과			0.0323		0.1118				0.0620				0.1192	0.1500	
	순위			1		3				2				4	5	
3001	D									42		17				35
	PDR									14.721		3.292				10.386
	결과									0.2372		0.0666				0.2105
	순위									3		1				2
4001	D													15	16	
	PDR													5.789	13.568	
	결과													0.1001	0.2884	
	순위													1	2	
5001	D									37		27				
	PDR									3.687		9.667				
	결과									0.0897		0.1628				
	순위									1		2				

<표 6> 유지보수 프로젝트의 개발팀 규모와 개발기간 선정 결과

FP	구분	개발팀 규모(명)												
		1	2	3	4	5	6	7	8	9	10	11	12	13 이상
1~100	D	2	4	1	7	3	2	3			4	8		
	PDR	2.186	6.742	0.974	59.429	8.116	4.394	6.444			4.966	7.850		
	결과	0.0966	0.1533	0.0247	1.0778	0.1782	0.1073	0.1688			0.1947	0.3766		
101~200	D	5	3	2	2	4	3		45		2	5		2
	PDR	3.889	1.456	1.748	14.484	3.771	8.896		2.047		4.727	4.913		4.655
	결과	0.0793	0.0412	0.0461	0.2859	0.1190	0.1997		1.0344		0.1351	0.2564		0.3694
201~300	D	3	3	3	3	3		4	12		12	10	12	7
	PDR	1.459	0.870	3.186	6.101	3.761		13.624	2.853		6.000	12.146	16.286	8.064
	결과	0.0329	0.0313	0.0786	0.1360	0.1049		0.3070	0.3147		0.4343	0.5089	0.6740	0.4079
301~400	D		5		6		6			4	10			24
	PDR		3.740		4.889		8.224			9.800	6.421			24.125
	결과		0.0907		0.1489		0.2384			0.2649	0.3858			1.4079
401~500	D			8		9		11						
	PDR			9.482		3.119		12.865						
	결과			0.2262		0.1775		0.4004						
501~600	D					4								
	PDR					2.921								
	결과					0.1047								
601~700	D			18	7	11							8	
	PDR			1.286	4.251	5.632							16.321	
	결과			0.1718	0.1493	0.2475							0.5413	
701~800	D							10						
	PDR							7.296						
	결과							0.3172						
801~900	D				4									
	PDR				1.430									
	결과				0.0665									
901~1000	D				7	4								
	PDR				2.347	1.573								
	결과				0.1173	0.0820								
1001~1200	D													7
	PDR													9.081
	결과													0.5028
1201~1400	D													1
	PDR													15.424
	결과													1.1151
1401~1600	D					8								
	PDR					5.020								
	결과					0.1956								
1601~1800	D													
	PDR													
	결과													
1801~2000	D													10
	PDR													5.546
	결과													0.4822
2001~3000	D													
	PDR													
	결과													
3001~4000	D													
	PDR													
	결과													
4001~5000	D													
	PDR													
	결과													
5001 이상	D													
	PDR													
	결과													

성을 개발 또는 유지보수로 구분하지 않고 평균적으로 개발팀의 규모를 제시하였으며, 제안된 결과와 많은 차이를 보이고 있다. 특히 기능점수가 2,000~3,000과 4,000~5,000 범위에서는 많은 차이를 보이고 있음을 알 수 있다.

<표 7>의 결과에 대해 기능점수 규모별로 개발기간, 개발팀 규모와 생산성에 대한 그래프에 추세선을 그려 개발팀 규모, 개발기간과 PDR을 추정하는 모델을 제시하였다. 주어진 데이터에 대해 가장 적합한 모델을 선정하기 위해 다양한 회귀분석을 수행한 결과 가장 적합한 추세선을 그린 결과는 각각 (그림 2), (그림 3)과 (그림 4)와 같다.

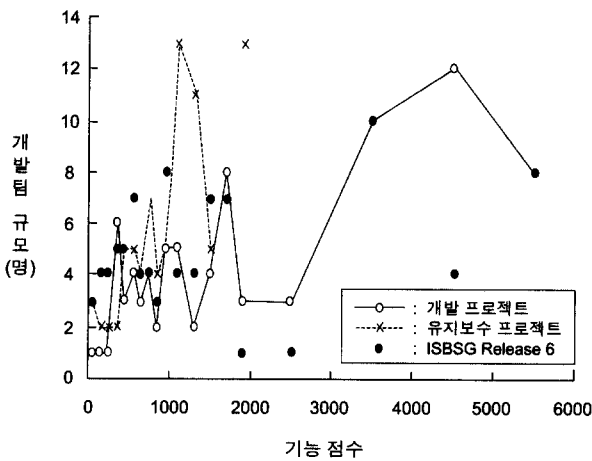
<표 7> 기능점수 규모별 최소의 개발팀 규모와 개발기간

FP	개발 프로젝트			유지보수 프로젝트		
	D	TS	PDR	D	TS	PDR
1~100	1	1	0.539	1	3	0.974
101~200	1	1	0.291	3	2	1.456
201~300	5	1	0.783	3	2	0.870
301~400	2	6	0.833	5	2	3.740
401~500	13	3	1.972	9	5	3.119
501~600	5	4	1.354	4	5	2.921
601~700	13	3	1.129	7	4	4.251
701~800	16	4	3.861	10	7	7.296
801~900	6	2	2.391	4	4	1.430
901~1000	30	5	3.757	4	5	1.573
1001~1200	9	5	3.129	7	13 이상	9.081
1201~1400	9	2	0.484	28	11	15.424
1401~1600	18	4	3.410	8	5	5.020
1601~1800	12	8	4.321	-	-	-
1801~2000	15	3	0.663	10	13 이상	5.546
2001~3000	42	3	1.119	-	-	-
3001~4000	17	10	3.292	-	-	-
4001~5000	15	12	5.789	-	-	-
5001 이상	37	8	3.697	-	-	-

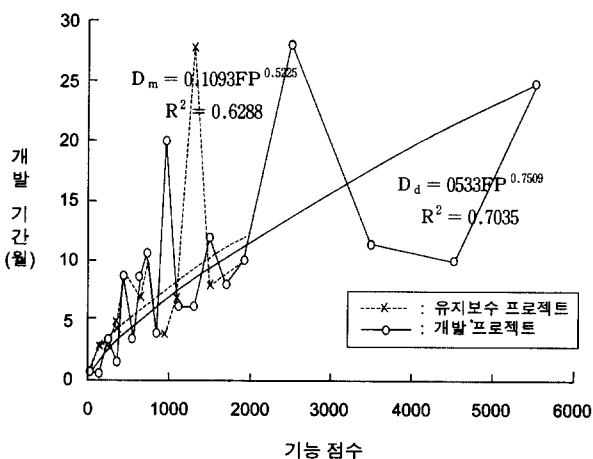
개발 프로젝트를 d , 유지보수 프로젝트를 m 이라 하자. 그림의 결과에 따라 개발과 유지보수 프로젝트에 대한 기능점수 규모별로 최소의 개발기간, 개발팀 규모와 최대의 생산성을 추정할 수 있는 모델과 성능을 <표 8>에 제시하였다.

적합한 통계적 회귀모델을 선정하기 위해 결정계수(Coefficient of determination, R^2)[19]를 적용하며, 모델의 성능을 평가하기 위해 MSE(Mean Squared Error)와 Briand et al.[20]이 적용한 MMRE(Mean Magnitude of Relative Error)를 이용한다. 종속변수의 값(E)은 독립변수(UUCP)에 의해 결정되는 부분과 미지의 오차의 합으로 나타나며, 총 변

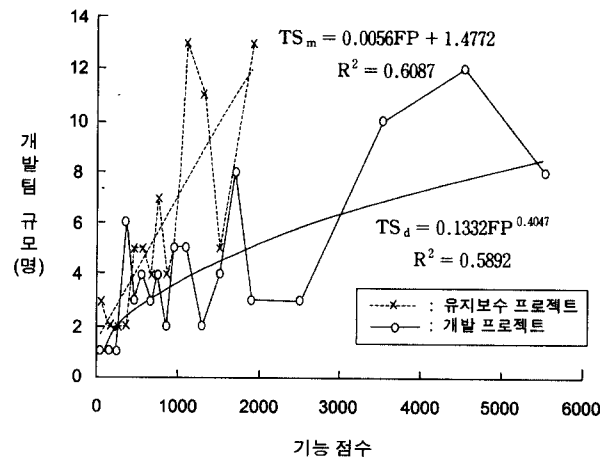
동을 설명하는데 있어서 회귀직선에 의해 설명되는 변동 비율을 결정계수라 하며, 값이 클수록 쓸모 있는 회귀직선이 된다. 모델의 정확도(Accuracy)를 평가하는 MMRE는 다음과 같이 측정된다. $RE = (\text{추정치} - \text{실측치}) / \text{실측치}$, $MRE(\text{Magnitude of RE}) = \text{ABS}(RE)$, $MMRE(\text{Mean MRE}) = 100/n \times \sum_{i=1}^n MRE$. MMRE가 작은 값이면 평균적으로 좋은 모델임을 알 수 있다. 개발기간 예측 모델은 결정계수가 커 상당히 적합한 모델로 평가되지만 MSE와 MMRE가 비교적 큰 값을 나타내어 상당한 오차를 갖고 있음을 알 수 있다. 이에 비해 생산성을 추정하는 모델은 결정계수도 작고 MMRE가 큰 값을 나타내어 생산성을 정확히 예측하는데 많은 어려움이 있음을 알 수 있다. 이에 비해 팀 규모 추정 모델은 결정계수도 어느 정도는 높고 MSE, MMRE도 적은 값을 나타내어 본 제안 모델로 예측한 팀 규모와 실제 투입되는 규모간에는 상당히 근접된 결과를 얻을 수 있음을 알 수 있다.



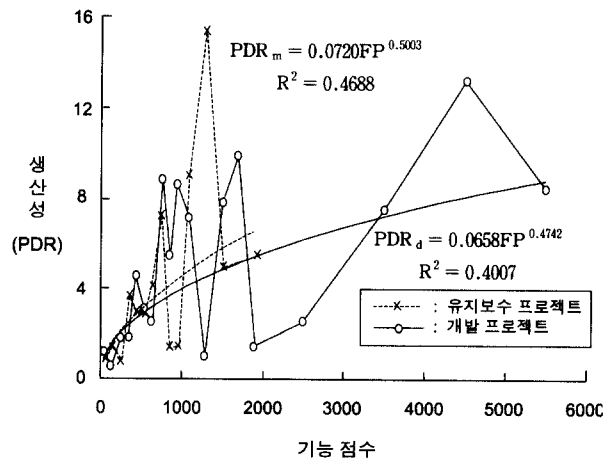
(그림 1) 개발팀 규모 추정 비교



(그림 2) 기능점수 규모별 최소의 개발기간 추정 모델



(그림 3) 기능점수 규모별 최소의 개발팀 규모 추정 모델



(그림 4) 기능점수 규모별 최대의 PDR 추정 모델

<표 8> 모델 성능

구분	모델	모델 성능		
		결정계수	MSE	MMRE
기간	개발 $D_d = 0.533FP^{0.7509}$	0.7035	76.80	45.52
	유지보수 $D_m = 0.1093FP^{0.5225}$	0.6288	49.32	41.61
팀 규모	개발 $TS_d = 0.1332FP^{0.4047}$	0.6087	10.60	39.96
	유지보수 $TS_m = 0.0056FP + 1.4772$	0.5892	5.38	36.19
생산성	개발 $PDR_d = 0.0658FP^{0.4742}$	0.4007	1.66	68.65
	유지보수 $PDR_m = 0.0720FP^{0.5003}$	0.4688	18.94	52.68

5. 결론

소프트웨어 프로젝트를 원활히 개발하기 위해 개발팀 규모와 개발기간을 결정하는 것은 프로젝트 관리자에게 필수적으로 요구되는 정보이다. 소프트웨어 개발분야에 대해 일반적으로 이상적인 팀 규모는 5~7명 또는 5명 이하가 가장 적합함을 알 수 있었다. ISBSG Benchmark Release 6 [9]는 FP 규모별 개발팀 규모에 대한 평균 또는 중앙 값을

이용하여 이상적인 개발팀 규모를 제시하였으나 FP 규모가 작거나 큰 모든 범위에서 이상적인 개발팀의 규모를 5~10 명까지 제시하였다.

본 논문은 개발팀의 생산성 척도인 PDR 과, 비용측면에 서 1인당 총 개발기간을 고려하였다. 1명당 PDR이 최소가 되고, 1인당 개발기간이 최소가 되는 기준을 적용하여 최소의 개발팀 규모와 개발기간을 선정하였다. 실험에는 ISBSG Benchmark Release 6[9]의 300개 프로젝트가 사용되었으며, 이들 데이터 집합을 개발 단계와 유지보수 단계에 적용된 프로젝트들로 분류되어 이상적인 개발팀 규모와 개발기간을 선정하였다.

실험 결과 개발 단계나 유지보수 단계 프로젝트 모두 소프트웨어의 규모에 따라 대부분이 1년 또는 1년 6개월내에 프로젝트가 종료되며, 5명 이하의 팀 구성이 요구되었다. 본 제시된 기준과 모델들은 소프트웨어 규모에 따라 최소의 개발팀 규모와 개발기간을 결정하는 지침으로 활용이 가능하다.

소프트웨어 규모에 따라 최소의 개발팀 규모와 개발기간을 설정하는데 적용된 기준인 생산성과 비용 요소는 개발팀 개인의 숙련도에 많은 영향을 받을 수 있으나 본 논문은 이 부분을 고려하지 못하였다. 따라서, 추후 이 분야에 대한 연구를 수행할 것이다.

참 고 문 헌

- [1] L. J. Arthur, "Measuring Programmer Productivity and Software Quality," New York, John Wiley, 1985.
- [2] K. H. Möller and D. J. Paulish, Software Metrics-A Practitioners Guide to Improved Product Development, Chapman & Hall Co., New York, 1993.
- [3] B. A. Kitchenham, "Empirical Studies of Assumptions That Underlie Software Cost-estimation Models," Information and Software Technology, Vol.34, No.4, pp.211-218, 1992.
- [4] S. Oligny, 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.
- [5] S. Oligny, P. Bourque, A. Abran and B. Fournier, "Exploring the Relation Between Effort and Duration in Software Engineering Projects," World Computer Congress 2000, Beijing, China, pp.175-178, August, 2000.
- [6] ISBSG, "Worldwide Software Development - The Benchmark Release 4," Victoria, Australia International Software Benchmarking Standards Group, 1997.
- [7] P. C. Semprevivo, "Teams in Information Systems Development," Yordon, p.85, 1980.
- [8] 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.
- [9] ISBSG, "Worldwide Software Development - The Benchmark Release 6," Victoria, Australia International Software Benchmarking Standards Group, 2000.
- [10] D. V. Ferens, B. A. Daly, "A Comparison of Software Scheduling Methods," In Reifer D, ed. Software Management. 4th ed. Washington : IEEE Computer Society Press, 1993.
- [11] B. A. Kitchenham, "Empirical Studies of Assumptions That Underlie Software Cost-estimation Models," Information and Software Technology, Vol.34, No.4, pp.211-218, 1992.
- [12] C. Jones, "Determining Software Schedules," Computer Vol.28, No.2, pp.73-75, 1995.
- [13] R. E. Park, W. B. Goethert and J. T. Webb, "Software Cost and Schedule Estimating : A Process Improvement Initiative," Pittsburgh, PA Software Engineering Institute, 1994.
- [14] "Six, not the Ideal Project Team Size," A Weekly Column on Software Design, <http://www.sum-it.nl/en199947.html>, 1999.
- [15] 이상운, "소프트웨어 개발팀 규모 추정 모델," 한국정보과학회논문지B, 제29권 제12호, pp.873-882, 2002.
- [16] 이상운, 박종양, 박재홍, "개발과 유지보수 프로젝트의 이상적인 팀 규모," 정보처리학회논문지D, 제10-D권 제1호, pp. 77-84, 2003.
- [17] 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.
- [18] L. C. Briand and I. Wiecek, "Resource Estimation in Software Engineering," International Software Engineering Research Network, Technical Report, ISERN 00-05, 2000.
- [19] A. Abran, C. Symons and S. Oligny, "An Overview of COSMIC-FFP Field Trial Results," ESCOM 2001, London, England, 2001.
- [20] L. C. Briand, K. E. Elmam, D. Surmann, I. Wiecek and K. D. Maxwell, "An Assessment and Comparison of Common Software Cost Estimation Modeling Techniques," International Software Engineering Research Network, Technical Report, ISERN-98-27, 1998.



박 석 규

e-mail : sgpark@gangwon.ac.kr

1988년~2001년 진주산업대학교 전산실장

1992년 경남대학교 대학원 컴퓨터공학과
(공학석사)

2001년~2003년 강원도립대학 컴퓨터응용과
전임강사

2003년~현재 강원도립대학 컴퓨터응용과 조교수

관심분야 : 시스템 분석 및 설계, 멀티미디어, 정보검색



이 상 운

e-mail : sangun@gangwon.ac.kr

1983년~1987년 한국항공대학교 항공전자
공학과(학사)

1995년~1997년 경상대학교 컴퓨터과학과
(석사)

1998년~2001년 경상대학교 컴퓨터과학과
(박사)

1992년~2003년 국방품질관리소 항공전자장비 및 소프트웨어
품질보증 담당

2003년~현재 강원도립대학 컴퓨터응용과 전임강사

관심분야 : 소프트웨어 공학(소프트웨어 시험 및 품질보증, 소프트
웨어 신뢰성), 소프트웨어 프로젝트 관리, 신경망,
뉴로-퍼지, Use-Case, RUP, CBD