經營科學
第19卷 第2號
2002年 11月

공공부문 정보화사업의 소프트웨어 개발비용 예측에 관한 연구

박찬규* · 구자환* · 김성희* · 신수정* · 송병선**

A Study on the Estimation of Software Development Cost of IT Projects in Public Sector

Chan-Kyoo Park* · Ja Hwan Gu* · Sung Hee Kim*
Soo Jeong Shin* · Byeong Sun Song**

■ Abstract ■

As the portion of information systems (IS) budget to the total government budget becomes greater, the cost estimation of IS development and maintenance projects is recognized as one of the most important problems to be resolved for scientific and efficient management of IS budget. Since IS budget makes much effect on the delivery time, quality and productivity of IS projects, the exact cost estimation is also necessary for the successful accomplishment of IS projects. The primary concern in the cost estimation of IS projects is software cost estimation, which requires the measurement of the size of softwares. There are two methods for sizing software : line-of-code approach, function point model.

In this paper, we propose a function-point-based model for estimating software cost. The proposed model is derived by collecting about fifty domestic IT projects in public sector and analyzing their relationship between cost drivers and development effort. Since the proposed model is developed by simplifying the function point model that can be used only when detailed user requirements are specified, it can be also applied at project planning and budgeting phase.

Keyword : Software Cost Estimation, Function Point, Information Systems, Project Planning

논문접수일 : 2002년 1월 22일 논문게재확정일 : 2002년 10월 24일
* 한국전산원 지식정보기술단 정보기술감리부
** 기획예산처 정보화예산팀

1. 서 론

정보시스템 구축과 운영 사업의 증가로 국가예산에서 정보화예산이 차지하는 비중이 높아짐에 따라, 정보화예산 관리의 과학화와 효율화를 위해 사업의 소요 비용을 사전에 정확히 예측하는 것이 중요한 문제로 대두되고 있다. 또한, 정보화사업의 예산은 정보시스템의 납기, 품질, 생산성 등에 심각한 영향을 미치므로[14, 21, 29], 예산의 과부족을 예방하면서 국가정보화사업을 성공적으로 수행하기 위해서도 정보시스템 구축 및 운영비용의 정확한 예측이 필요하다.

정보시스템 개발사업의 경우 전체 사업비용을 크게 소프트웨어 개발비와 장비 도입비로 구분할 수 있다. 장비 도입비는 하드웨어 시장가격 조사를 통해 쉽게 예측할 수 있으나, 소프트웨어 개발비는 요구사항의 불확실성과 모호성, 소프트웨어 생산성 측정의 어려움, 과거 데이터의 부족, 신기술 및 개발환경 의존성 등으로 인해 예측하기가 쉽지 않다[28].

소프트웨어 개발비용을 예측하기 위해서는 먼저 개발하고자 하는 소프트웨어의 규모를 측정하고, 그 규모로부터 비용을 추정하게 된다. 소프트웨어의 규모를 측정하는 방법으로 여러가지가 있으나 가장 흔히 사용되는 방법으로 프로그램 라인수(line of code) 방식과 기능점수모형(function point model)이 있다[24, 25]. 우리 나라에서는 소프트웨어 개발 및 유지보수 사업의 규모 산정에 주로 라인수 방식을 사용해 왔는데, 라인수 방식은 개발자 중심적이고 사전에 라인수 예측이 어려우며 개발환경에 영향을 받는 등 여러가지 단점을 갖고 있다[9, 15, 20]. 라인수를 이용한 소프트웨어 비용산정 모형으로는 Boehm[6], Putnam & Myers[25] 등의 연구가 있으며, 최근에 기능점수방식을 반영한 수정된 모형이 제안되고 있다[6]. 반면, Albrecht[5]가 제안한 기능점수모형은 정보시스템이 제공하는 기능의 수를 사용자 관점에서 측정하여 소프트웨어 규모를 산정하는 방법으로, 사용자가 이해하기 쉽고 개발환경에 독립적이라는 장점을 갖고 있다[9, 20]. 기능점수를 이용한 비용산정 모형으로는 COCOMO II[6], SPQR/20[14], ISBSG[11] 등이 있다. 그러나, 기능점수모형은 그 장점에도 불구하고 국내에서는 라인수 방식에 비해 활발히 사용되지 못하고 있는데, 이는 제도적 문제와 더불어 국내 적용을 위한 실제 데이터의 축적과 분석이 충분히 뒷받침되지 못했기 때문이라 할 수 있다.

미국을 비롯한 호주, 일본, 유럽 등에서는 1980년대부터 소프트웨어 생산성 측정과 비용/납기 예측을 위해 기능점수방법을 적용하고 문제점을 보완하기 위한 연구가 꾸준히 진행되어 왔다. Symons[26]은 Albrecht가 제안한 기능점수모형에서 내부논리파일(internal logical file) 대신 개체(entity)를 사용할 것과 기능점수 계산을 보다 간편화하고 오차를 줄일 수 있는 Mark II 모형을 제시하였다. Low & Jeffery[19]는 조직, 계측자에 따라 기능점수계수가 일관성이 있는가를 실험적으로 비교하여 기능점수모형이 라인수 방식보다 일관성 있는 척도가 될 수 있음을 보였다. 또한, Kemerer & Porter[15]는 계측자간 기능점수 산정에 편차가 발생하는 이유를 제시하였고, Jeffery, Low & Barnes[12]는 Albrecht의 기능점수방법과 Jones[13]에 의해 제안된 수정된 기능점수 방법에 의한 비용 예측 결과를 비교하였다. 또한, 기능점수방식에서 사용되는 5가지 기능유형(function type)간의 상관성(correlation)에 관한 연구가 Kitchenham & Känsälä[16], Lokan[18] 등에 의해 이루어졌고, Kitchenman[17]이 기능점수 적용 과정에서 나타나는 문제점을 발표하였다. Orr & Reeves[23]은 실제 프로젝트 사례를 통해 소프트웨어 수명단계마다 기능점수계수가 어떻게 변화하는가를 추적·분석하였다. 그 밖에 CASE와 기능점수 모형과의 관계, 프로세스 제어(process control) 시스템에서의 기능점수 적용, 객체지향(object-oriented) 소프트웨어에서의 기능점수 적용 등의 연구가 있었다[7, 8, 22, 27, 29].

기능점수를 이용한 소프트웨어 비용산정에 관

한 국내 연구로는 김현수[1], 이양규[2], 한국전산원[4] 등의 연구가 있었다. 김현수[1]는 클라이언트/서버 환경, 대형 소프트웨어 프로젝트, 사무처리용 및 알고리즘 중심적인 소프트웨어 개발 등에 적합한 수정된 기능점수 모형을 제시하고, 설문조사를 통해 모형간의 적합성, 활용성, 예측정확성 등을 비교하였다. 이양규[2]는 34개의 공공부문 프로젝트를 대상으로 설문을 통해 기능점수와 소요인력을 조사하여, 기능점수와 투입인력간의 회귀식을 제시하였다. 또한, 정보시스템 운영사업 유형별 운영비용 예측에 관한 연구가 한국전산원[4]에 의해 수행되었다. 그러나, 과거 프로젝트 자료가 기능점수에 대한 이해도가 부족한 실무자들을 대상으로 한 설문서에 의해 수집되었고, 주로 소규모 프로젝트가 전체 자료의 대부분을 차지하고 있는 등 소프트웨어 비용예측모형 도출을 위한 실적데이터의 축적과 분석이 절대적으로 부족한 상태이다.

이상의 국내외 기존 연구에서는 대상 프로젝트의 모든 기능유형과 데이터 요소들이 식별되어 정확한 기능점수 산정이 가능하다는 것을 가정하고 있으나, 실제 대부분의 프로젝트 사례에서는 분석/설계단계가 완료되기 이전에는 개략적인 기능유형과 개체들만 파악할 수 있을 뿐이다. 따라서, 소프트웨어 개발비용 예측모형이 정보화사업의 예산 신청 및 예산 결정 과정에 실질적으로 활용될 수 있기 위해서는 사업초기에 획득 가능한 정보를 사용한 비용예측모형이 개발되어야 한다. 또한, 기존 연구에서 사용된 과거 프로젝트 자료는 기능점수 산정 규칙이 상이한 여러개의 조직에서 수집된 자료이기 때문에, 조직간의 편차가 내재된 기능점수 데이터로부터 비용모형이 도출되었다. 이외에도 데이터 개수가 충분치 못하거나, 설문을 통해 수집된 데이터를 이용하여 비용예측모형이 도출되는 등의 문제점이 있었다.

본 연구는 국내 공공기관 소프트웨어 개발사업 결과물을 직접 수집·분석하여 필요한 데이터를 추출하고, 이를 토대로 사업초기 단계에 적용가능한 간이기능점수 비용예측모형을 제시하는데 그 목적을 두고 있다. 따라서, 본 연구에서 제시하는 예측 모형은 통계학적인 관점에서의 엄밀한 검증을 거친 모형이기보다는 실용적이고 간편한 소프트웨어 개발비용 예측모형이라는데 그 의의가 있을 것이다. 예측모형의 개발을 위해 먼저 국내에서 수행된 48개의 정보시스템 신규 구축사업의 최종 결과물로부터 입력화면/출력보고서/내·외부엔티티(entity) 등을 분석하였다. 다음에, 이러한 간이기능점수방식의 비용요소와 투입인력과의 관계를 분석하여 기능점수 모형을 단순화한 소프트웨어 개발비 예측모형을 제시한다. 본 연구의 결과는 공공부문 정보화사업을 기획·심의하거나 직접 수행하는 사람이 정보화사업 착수전에 사업비용을 쉽게 예측하는데 사용될 수 있을 것이다.

본 논문의 구성을 살펴보면, 2절에서 기능점수 방식을 소개하고 그 장단점을 알아보고, 3절에서 사업초기단계에서 활용할 수 있는 소프트웨어 개발비용 예측을 위한 간이기능점수 모형을 제시한다. 4절에서 실제 정보화사업 데이터를 분석하여, 개발비용 예측을 위한 회귀식을 제시한다. 마지막으로, 5절에서는 결론과 추후 연구과제를 제시한다.

2. 기능점수 모형

Albrecht가 기능점수 모형을 제안한 이후, 이를 보완하고 조직에 맞게 수정한 여러가지 변형된 기능점수 모형이 있지만, 그 중에서도 가장 널리 사용되고 있는 국제기능점수사용자그룹(IFPUG)에서 제안한 기능점수 모형을 간략히 알아본다. 기능점수 모형에 대한 자세한 내용은 [10]를 참고하기 바란다.

기능점수방식은 사용자의 관점에서 소프트웨어 규모를 산정하는 방법으로, 주로 논리적 설계를 기초로 소프트웨어가 사용자에게 제공하는 기능의 수를 정량화하여 소프트웨어의 규모를 산정한다. 기능점수방식은 구매하고자 하는 응용패키지의 규모 산정, 소프트웨어의 품질과 생산성 분석, 소프

트웨어 개발과 유지보수를 위한 비용과 소요자원 산정 등에 사용된다. 기능점수방식는 먼저 데이터기능점수와 트랜잭션기능점수(transaction function point)의 합계를 구한 다음, 시스템 특성에 따라 그 값을 보정함으로써 최종적인 소프트웨어의 기능점수 계수를 산정한다.

보정전 기능점수는 데이터와 트랜잭션이라는 두 가지 유형을 갖는다. 이러한 두 가지 유형을 보다 세분화하면 다음 그림과 같다.

데이터기능(data function)은 내부 및 외부 자료 요구사항을 만족시키기 위해 사용자에게 제공되는 기능을 말한다. 데이터기능에는 내부논리파일(Internal Logical File, ILF)과 외부연계파일(External Interface File, EIF)가 있다.

1) 내부논리파일

응용시스템 영역 안에서 유지(maintain)되고 사용자가 식별가능(identifiable)한 논리적으로 연관된 자료 및 제어정보(control information)의 그룹을 말한다. 내부논리파일의 기본적인 용도는 해당 응용시스템의 단위프로세스(elementary process)에서 유지되는 데이터를 저장하는 것이다.

2) 외부연계파일

응용시스템에서 참조되지만 다른 응용시스템에서 유지되며 사용자가 식별가능한 논리적으로 연관된 자료 및 제어정보의 그룹을 말한다. 외부연계파일은 기본적인 용도는 해당 응용시스템내의 단위프로세스에서 참조되는 데이터를 저장하는데 있다. 외부연계파일은 반드시 다른 응용시스템의 내부논리파일이다.

내부논리파일과 외부연계파일은 데이터요소유형(Data Element Type, DET)과 레코드요소유형(Record, Element Type RET)의 개수에 따라 <표 1>과 같이 복잡도를 결정하고, 그 복잡도에 따라 보정전 기능점수를 <표 2>와 같이 산출한다.

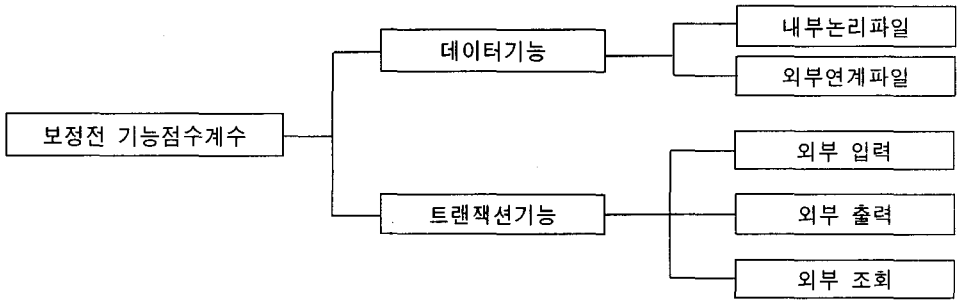
〈표 1〉 데이터기능 복잡도 행렬

		DET 개수		
		1~19	20~50	51개이상
RET 개수	1	낮 음	낮 음	보 통
	2~5	낮 음	보 통	높 음
	6개 이상	보 통	높 음	높 음

〈표 2〉 ILF/EIF의 기능점수

복잡도 등급	보정전 기능점수	
	ILF	EIF
낮 음	7	5
보 통	10	7
높 음	15	10

트랜잭션기능(transactional function)은 데이터를 처리하기 위해 사용자에게 제공되는 기능을 말한다. 트랜잭션기능에는 외부입력(External Input, EI), 외부출력(External Output, EO), 외부조회(External inQuiry, EQ) 등이 있다.



〈그림 1〉 기능점수 유형

3) 외부입력

외부입력은 응용시스템 외부에서 들어오는 데이터 및 제어정보를 처리하는 단위프로세스이다. 외부입력은 주로 내부논리파일을 유지하거나 시스템의 작동상태를 변경하는데 사용된다.

4) 외부출력

외부출력은 응용시스템 밖으로 데이터나 제어정보를 보내는 단위프로세스이다. 외부출력은 주로 데이터 및 제어정보의 조회 외에 처리로직을 통해 사용자에게 정보를 제공하는데 사용된다. 처리로직은 수학적 공식 및 계산식을 포함하거나 유도되는 데이터를 생성해야 한다. 또한, 외부출력은 내부논리파일을 유지하거나 시스템의 작동상태를 변경할 수도 있다.

5) 외부조회

외부조회는 응용시스템 외부에 데이터나 제어정보를 보내는 단위프로세스이다. 외부조회는 주로 데이터나 제어정보의 조회를 통해 사용자에게 정보를 제공하는데 있다.

트랜잭션기능의 복잡도는 참조파일유형(File Type Referenced, FTR)과 데이터요소유형(Data Element Type, DET)의 개수에 의해 <표 3>과 <표 4>와 같이 결정되고, 복잡도에 따라 기능점수가 <표 5>와 같이 결정된다.

보정전 기능점수가 계산되면, 시스템의 특성을 반영하도록 보정요소와 보정계수를 계산하여 기능점수를 보정하게 된다. 보정요소에는 데이터 통신(data communications), 분산 데이터 처리(distributed data processing), 성능(performance), 사용환경(heavily used configuration), 처리율(transaction rate) 등 14개가 있다.

이상의 기능점수 모형이 사용자 중심이고, 프로그래밍 언어 등으로부터 독립적이고 일관성 있는 척도라는 장점이 있지만, 여러가지 보완되어야 할 점도 존재한다.

첫 번째로, 소프트웨어 수명주기 중에서 요구사

〈표 3〉 EI의 복잡도 행렬

		DET 개수		
		1~4	5~15	16이상
FTR 개수	0~1	낮 음	낮 음	보 통
	2	낮 음	보 통	높 음
	3 이상	보 통	높 음	높 음

〈표 4〉 EO/EQ의 복잡도 행렬

		DET 개수		
		1~5	6~19	20이상
FTR 개수	0~1	낮 음	낮 음	보 통
	2~3	낮 음	보 통	높 음
	4 이상	보 통	높 음	높 음

〈표 5〉 트랜잭션기능의 기능점수

복 잡 도	보정전 기능점수	
	EI/EQ	EO
낮 음	3	4
보 통	4	5
높 음	6	7

항 분석단계 이후부터 적용될 수 있는 규모 산정 방법이라는 점이다. 대부분의 정보화사업이 정보전략계획(Information Stratey Planning, ISP) 없이 시작되거나 또는 ISP를 통한 개략적인 기능과 엔티티 수준만 결정된 상태에서 시작된다. 따라서, 사업제안과 예산심의 시점에서 기능점수 모형이 실질적으로 활용될 수 있기 위해서는 기능점수 모형에 대한 보완이 필요하다. 실제로, Symons[26]은 5가지 기능유형대신에 입력자료, 출력자료, 참조되는 엔티티 등 세 가지 요소로 기능점수를 산정하는 MARK II 모형을 제안했다.

두 번째로, 기능점수 모형의 데이터기능, 트랜잭션기능들이 기능점수에 기여하는 정도가 연속적인 척도로 평가되지 않는다는 것이다[17, 26]. 예를 들어, 내부논리파일의 복잡도는 '낮음', '보통', '높음'의 등급으로 평가될 뿐이며 연속적인 값으로 평

가되지 못한다. 또한, 각각의 복잡도에 따라 주어지는 기능점수가 잘 맞지 않을 수도 있다. '낮음'인 내부논리 파일과 '높음'인 내부논리 파일의 기능 점수비가 7 : 15인데 이는 프로젝트 유형이나 조직의 프로세스 수준 등에 따라 달라질 것이다. Symons[26]는 MARK II 모형의 입력자료, 출력자료, 참조엔티티간의 가중치를 연속적인 값으로 설정하고 있다.

세 번째로, 다섯 가지 기능유형(ILF, EIF, EI, EO, EQ)간에 상관관계가 있다는 점이다. Kitchanham & Känsälä[16], Lokan[18]에 의하면, EI와 EO, EI와 EQ, EI와 ILF, EO와 EQ, EO와 ILF, EQ와 ILF간에 상관관계가 있는 것으로 나타났다. 특히, 개발프로젝트의 경우, ILF, EI, EQ, EO간의 강한 상관관계가 존재하는 것으로 나타났으며, EIF는 다른 기능유형과 상관관계가 없는 것으로 나타났다.

네 번째로, 보정계수의 결정은 주관적 판단에 의존하고, 보정요소에 대한 복잡도도 절대적인 값이 아닌 등급으로 결정된다. 실제로, Kemerer[15], Low & Jeffery[19]에 의하면 동일한 조직의 사람들도 동일한 프로젝트에 대해 보정요소 등급을 다르게 판단하고 있으며, 보정요소로 인한 비용모델의 설명력 향상도 크지 않은 것으로 보고되었다.

기능점수 모형이 사업제안단계에 사용하기 어렵고, 기능유형들의 기능점수가 상대적 등급에 의해 결정되며, 기능유형간의 상관관계 및 보정계수의 실효성 문제 등의 문제점을 갖고 있으므로 이에 착안하여 사업초기단계에 적용 가능하도록 단순화한 간이기능점수 모형을 3절에서 제안한다.

3. 소프트웨어 개발비 예측을 위한 간이기능점수 모형

기능점수 모형이 사업제안단계에 적용할 수 있게 하기 위해서, 본 연구에서는 입력화면, 출력화면, 출력보고서, 내부엔티티, 외부엔티티 등 다섯 가지의 기능유형을 고려하여, 간이기능점수 모형을 개발하였다. 기능점수 모형과 간이기능점수 모형의 차이점은 크게 세 가지로 요약할 수 있다. 첫 번째, 간이기능점수 모형의 기능요소는 기능점수 모형의 기능요소와 약간 다르다. 이는 사업초기단계에서의 획득 가능여부를 기준으로 기능요소들을 선정했기 때문이다. 두 번째, 간이기능점수 모형에서는 각 기능요소의 복잡도를 별도로 구하지 않는다. 왜냐하면, 사업제안단계에서는 내부엔티티 수준만 도출되고 분석/설계단계에서 비로소 엔티티의 속성까지 상세하게 도출되므로, 속성의 개수를 고려하여 엔티티의 복잡도를 결정하는 것은 사업초기단계에 현실적으로 무리가 따르기 때문이다. 세 번째, 간이기능점수 모형은 시스템 특성에 따른 보정 과정이 없다. 시스템 특성에 따른 보정이 소프트웨어 개발비 예측 결과의 정확성을 어느 정도 향상시키지만, 향상되는 정도는 크지 않는 것으로 보고되고 있다[19]. 특히, 사업초기단계에서는 사업의 불확실성으로 인해 분석/설계단계에 비해 요구되는 예측결과의 정확도가 낮다고 볼 수 있기 때문에, 시스템 특성에 따른 보정과정이 없더라도 간이기능점수 모형의 유용성이 크게 훼손되지는 않을 것이다.

〈표 6〉 기능점수 모형 vs. 간이기능점수 모형

기능점수 모형(IFPUG 모형)	간이기능점수 모형
내부논리파일(ILF)	내부엔티티(Internal Entity, IE)
외부연계파일(EIF)	외부엔티티(External Entity, EE)
외부입력(EI)	입력화면(Input Screen, IS)
외부출력(EO) 외부조회(EQ)	출력화면(Output Screen, OS) 보고서출력(Report Output, RO)

기능점수 모형과 간이기능점수 모형의 기능유형 차이점을 비교하면 <표 6>과 같다.

간이기능점수 모형에서는 내부논리파일 대신에 내부엔티티를 사용한다. 내부엔티티는 대상 시스템에 의해 유지되며 사용자가 논리적으로 식별가능한(identifiable) 엔티티를 의미한다. 마찬가지로, 외부엔티티는 외부시스템의 의해 유지되며 대상 시스템에 의해 참조되고, 사용자가 논리적으로 식별가능한 엔티티를 뜻한다.

입력화면은 데이터를 입력/수정/삭제하기 위해 사용되는 화면을 의미하며, 동일한 화면에 입력/수정/삭제 기능이 함께 포함되어 있으면 기능점수 모형과 같이 각각을 하나의 입력화면으로 간주한다. 출력화면은 데이터를 검색/조회/출력하는데 사용되는 화면으로, 입력 기능과 조회기능이 한 화면에 구현되어 있으면, 각각 입력화면과 출력화면으로 산정한다. 또한, 검색 조건을 입력하는 화면과 검색 결과를 출력하는 화면이 별도의 화면으로 구분되어 있더라도 이 두 개를 합쳐 하나의 출력화면으로 산정한다. 출력보고서는 데이터에 대한 조회/처리 결과를 인쇄하는 보고서를 의미한다.

기능점수 모형의 외부출력과 외부조회는 사용자에게 데이터를 전달한다는 점에서 공통점을 가지고 있으나, 유도된 데이터를 포함하거나 수학적 공식 또는 데이터의 변경을 포함하면 외부출력이고 그렇지 않은 경우에는 외부조회로 분류된다. 그러나, 간이기능점수 모형은 출력이 화면상에서 이루어지면 출력화면, 출력이 인쇄물로 나타나는 경우는 보고서출력으로 분류한다. 이는 사업초기단계에서 출력되는 데이터가 내부적으로 저장된 자료인지 아니면 유도되는 자료인지를 결정하기가 현실적으로 어렵다는 점에 착안한 것이다.

그러나, 이러한 기능점수 모형과 간이기능점수 모형간의 차이점에도 불구하고 간이기능점수 모형은 소프트웨어가 제공하는 기능수를 측정한다는 점에서 개념상 동일하다. 간이기능점수 모형은 기능수의 측정을 보다 용이하게 할 수 있도록 기능점수 모형과는 다른 기능요소를 사용하고 있다. 그러나, 간이기능점수 모형의 내부엔티티, 외부엔티티는 기능점수 모형의 내부논리파일, 외부연계파일과 대부분 일대일 관계를 가지고, 기능점수 모형의 일종인 Mark II 모형에서도 데이터기능을 내부엔티티, 외부엔티티를 써서 산정하고 있다. 또한, 기능점수 모형의 외부입력(EI)는 입력화면에 거의 일대일로 대응된다. 다만, 기능점수 모형이 외부출력(EO)과 외부조회(EI)를 구분함에 있어 처리로직의 복잡성을 기준으로 하는 반면에 간이기능점수 모형은 적용 편의를 위해 출력형태를 기준으로 출력화면과 보고서 출력으로 구분한다.

본 연구에서 소프트웨어 개발비를 직접 예측하기보다는 사업에 투입된 개발업체의 인력(Man-Month, M/M)을 예측하였다. 왜냐하면, 소프트웨어사업대가기준에 의해 직접인건비에 일정 비율을 곱한 다음 직접경비를 더하여 실제 소프트웨어개발비를 계산할 수 있기 때문이다. 물론, 실제 소프트웨어 개발비용에는 장비도입비, 소프트웨어 구매비 등은 포함되지 않는다. 실제 사업에서 투입인력과 개발비용간에는 매우 밀접한 선형관계가 있었으며, 소프트웨어 비용산정을 위한 상용도구(Knowledge-PLAN[1], PRICE-S[2] 등)에서도 대부분 개발비용의 예측보다는 필요한 투입인력을 예측하고 있다. 소프트웨어 개발비용을 예측하는데 있어 개발비용을 직접 예측하지 않고 투입인력을 예측하는 방식의 장점은 인플레이션(inflation)이나 노임단가의 변화를 고려하므로 인해 발생하는 분석의 복잡성을 피할 수 있다는 점이다.

그러나, 사업에 투입되는 개발인력은 소프트웨어사업대가기준[3]에 의해 특급/고급/중급/초급기술자, 고급/중급/초급기능사 등으로 구분되므로, 소프트웨어사업의 노임단가[3]를 기준으로 서로 다른 등급의 개발인력을 중급기술자수로 환산한 표준투입인력수를 구하였다. 예를 들어, 특급기술자, 중급기술자, 초급기술자의 1일 노임단가는 각각 153,525

1) http://www.spr.com.
2) http://www.pricesystems.com.

원, 97,571원, 70,963원이므로, 특급기술자 1M/M는 중급기술자 1.573 M/M로 변환되고, 초급기술자 1 M/M는 중급기술자 0.727 M/M로 변환된다.

간이기능점수 모형의 기능요소와 표준투입인력수의 상관관계를 구하여 소프트웨어개발비 예측모형을 만들어 본다. 이 때, 다음과 같은 세 가지 소프트웨어 개발비 예측모형을 고려하였다. 기능점수(FC)는 시스템의 특성을 반영하지 않은 보정 전의 기능점수를 의미한다.

모형 1과 모형 2는 간이기능점수 모형의 기능요소들과 기능점수와의 상관관계를 구하는 식과, 기능점수와 표준투입인력수의 상관관계를 구하는 식으로 이루어져 있다. 기능점수와 표준투입인력수의 상관관계를 선형으로 가정한 것이 모형 1이고, 지수함수 관계로 가정한 것이 모형 2이다. 모형 1과 모형 2와 같이 소프트웨어의 규모를 기능점수로 산정한 다음, 기능점수와 소프트웨어 개발비간의 관계를 추정하는 방식은 김현수[1], Mukhopadhyay [22], Symon[26] 등에서 사용된 방법이다. 다만, 각 연구에서 사용한 기능점수요소들은 차이가 있다. 그러나, 이러한 방식을 적용하는 데는 두 가지 중요한 문제점이 있다. 첫 번째, 기능점수(FC)와 간이기능점수 모형의 관계식 계수들을 구하기 위해서는 동일한 소프트웨어사업에서 기능점수(FC)와 간이기능점수요소들을 모두 계산해 내야 하기 때문에, 사례 수집에 많은 시간이 소요된다. 두 번째, 기능점수 계측자에 의한 기능점수 오차가 약 30%에 달하므로[15, 19], 식 (1)에 오차가 발생하게 되고 이는 식 (2)의 회귀모형에 영향을 미쳐 전체적으로 간이기능점수 모형의 소프트웨어 개발비에 대한 설명력을 저하시킨다. 즉, 식 (1)과 식 (2)를 거쳐 간이기능점수 모형의 기능요소와 소프트웨어 개발비간의 상관관계를 구하기 때문에 예측오차가 커지는 문제점이 있다. 덧붙여, 기능점수라는 한 개의 독립변수로 표준인력투입수를 예측하기 때문에 회귀모형의 설명력이 식 (4)에 비해 떨어지게 된다.

모형 3은 간이기능점수 모형의 기능요소들과 표준투입인력수의 상관관계를 바로 구하는 방식을 취한다. 모형 3은 회귀절차가 두 단계에서 한 단계로 줄어 모형 1과 모형 2에 비해 회귀식의 오차를 줄일 수 있기 때문에, 간이기능점수 모형의 설명력을 높일 수 있다. 이러한 접근방법은 Kitchenham & Känsälä[16]에 의해 사용되었으며, 이 때 기능점수 모형의 기능요소와 투입인력간의 상관관계를 구하였다.

4. 정보화사업 적용사례

간이기능점수 모형과 표준투입인력간의 회귀식을 구하기 위해, 정보화사업 결과를 분석하였다. 간이기능점수 모형의 기능요소별 개수는 <표 8>에 나타난 산출물을 참조하여 산정하였다.

내부엔티티는 ERD(Entity Relationship Diagram), 자료흐름도(Data Flow Diagram), 논리테이블명세서 등을 참고로 하여 추출해 냈으며, 외부엔티티는 자료흐름도, 파일명세서, 또는 외부 인터페이스 설계서 등을 참조하여 파악하였다. 입력화면과 출력화면은 기능분해도, 화면 및 윈도우 설명서 등을 토대로 산출해 낼 수 있었고, 출력보고서는 작업흐름도, 보고서 서식 등을 참고하여 식별해 낼 수 있었다. 또한, 등급별 투입인력은 사업

〈표 7〉 소프트웨어 개발비 예측 모형

모형 1	기능점수 (FC) = $a + b \times IE + c \times EE + d \times IS + e \times OS + f \times RO$ 표준투입인력수 (SMM) = $\alpha + \beta(FC)$	(1) (2)
모형 2	기능점수 (FC) = $a + b \times IE + c \times EE + d \times IS + e \times OS + f \times RO$ 표준투입인력수 (SMM) = $\alpha(FC)^{\beta}$	 (3)
모형 3	표준투입인력수 (SMM) = $a + b \times IE + c \times EE + d \times IS + e \times OS + f \times RO$	(4)

〈표 8〉 간이기능점수 모형과 관련 산출물

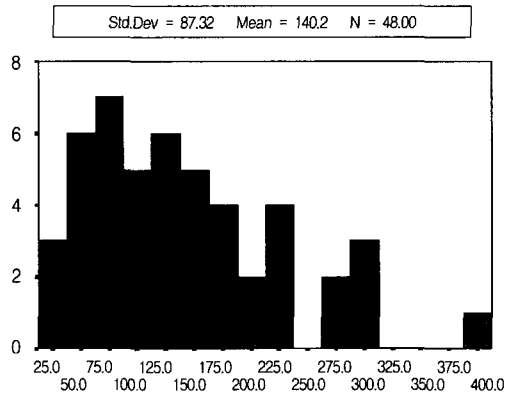
간이기능점수 모형의 기능요소	참고한 산출물
내부엔티티/외부엔티티	자료흐름도, ERD, 엔티티설명서, 논리테이블 설명서, 파일설명서, 외부인터페이스 설계서
입력화면/출력화면	기능분해도, 화면 및 윈도우 설명서
보고서출력	작업흐름도, 보고서 서식 및 설명서

계획서, 주월간공정보고 및 계약관련자료 등을 참조하여 산출한 후 표준투입인력으로 환산하였다. 기능점수 모형에 따른 기능점수를 산출하기 위해서는 상당한 인력 및 시간이 필요하기 때문에, 본 연구에서는 기능점수 모형의 기능점수 산출까지는 수행하지 못했다. 따라서, <표 7>에 제시된 세 가지 모형 중에 적용 가능한 것은 모형3뿐이다.

본 연구에서는 산출물과 투입인력 자료를 획득할 수 있는 총 48개의 공공기관 정보화사업을 선정하여, 간이기능점수 모형의 기능요소와 표준투입인력수를 산출했다. 각 기능요소와 표준투입인력수의 분포는 <표 9>와 같다. 소프트웨어사업대가기준에 의거 표준투입인력수로부터 소프트웨어 개발비를 산출해 보면, 약 25억원 이내에 사업들이 본 연구의 분석대상임을 알 수 있다. 단, 소프트웨어 개발비에 직접경비는 포함되지 않는다.

사업별 표준투입인력수 분포는 <그림 2>와 같다. 표준투입인력수가 50M/M~150M/M인 사업이 전체 사업의 52.1%로 과반수 이상을 차지하고 있으며, 300M/M이상인 사업은 총 4개이고 그 중 1개의 사업은 400M/M를 초과하고 있다.

본 연구에서는 기존 정보화사업에 투입된 인력은 규모에 맞게 적정하게 투입되었다고 가정한다. 왜냐하면, 원가분석과 손익계산이 선행되어야만 기수행된 정보화사업의 적정 투입인력이 계산될 수 있으나, 현실적으로 개별 정보화사업의 정확한 원가분석과 손익계산 정보는 획득하기 어렵기 때문이다.



〈그림 2〉 표준투입인력수 분포

모형3에 의해 간이기능점수 모형의 기능요소와 표준투입인력간의 상관관계를 분석하였다. 모형3에서 상수항을 포함하는 경우와 포함하지 않는 경우에 회귀식의 계수와 결정계수 R^2 은 <표 10>과

〈표 9〉 기능요소/투입인력 분포

기능요소/투입인력	범 위	평 균	표준편차
내부엔티티	8~336	89.40	71.08
외부엔티티	0~145	7.40	25.28
입력화면	2~434	119.71	97.45
출력화면	8~453	145.19	107.10
보고서출력	0~105	19.88	25.79
표준투입인력	14.94~402.84	140.23	87.32

〈표 10〉 모형 3에 의한 회귀결과

	회 귀 식	R^2	계수값 및 유의확률					
			상수	IE	EE	IS	OS	RO
상수항이 있는 경우	표준투입인력수 = $a + b \times IE + c \times EE + d \times IS + e \times OS + f \times RO$	0.528	65.409 (0.001)	0.352 (0.033)	0.911 (0.023)	0.550 (0.001)	-0.116 (0.385)	-0.624 (0.140)
상수항이 없는 경우	표준투입인력수 = $b \times IE + c \times EE + d \times IS + e \times OS + f \times RO$	0.827	-	0.541 (0.003)	1.039 (0.023)	0.555 (0.003)	0.058 (0.683)	-0.435 (0.359)

같다. 두 개의 회귀식 모두 유의한 것으로 나타났다.

<표 10>에서 알 수 있듯이 상수항 존재 여부에 상관없이 회귀식의 계수에 음수가 나타나고 있다. 이러한 현상은 기능요소가 증가할수록 비용이 증가한다는 일반적인 인식과 배치되는 것으로서, 기능요소간에 상관관계가 존재함을 암시한다. 기능요소간의 상관관계는 <표 11>과 같이 분석되었다. 기능점수 모형에 대한 기존 연구들에서 입증된 바와 같이, 간이기능점수 모형의 기능요소간에도 상관관계가 존재함을 알 수 있다.

모형 3의 회귀식 계수들이 현실 상황에 부합되도록 하기 위해서는 간이기능점수 모형의 기능요소간의 상관관계를 제거할 필요가 있다. 그러나, 대부분의 기능요소간에 상관관계가 존재하여 상관관계를 완전히 제거하기는 현실적으로 어렵다. 따라서, 간이기능점수 모형의 기능요소 중에서 상관관계가 가장 큰 것으로 분석된 입력화면의 수와 출력화면의 수를 더해 하나의 변수로 표현함으로써, 추정된 회귀식의 계수가 현실 상황과 배치되는 현상을 완화해 보기로 하자. <표 3>, <표 4>, <표 5>, <표 6>에서 알 수 있듯이 입력화면 하나와 출력화면 하나에 투입되는 노력은 거의 비슷하다고 할 수 있다. 입력화면의 수와 출력화면의 수를 하나의 변수(IOS, Input/Output Screen)으로 나타내어 표준투입인력과 간이기능점수 모형의 기능요소간의 회귀식을 구해보면 <표 12>와 같다.

<표 12>에서 볼 수 있듯이, 여전히 회귀식에 음수가 나타나고, 내부엔티티(IE), 입출력화면(IOS), 보고서출력(RO)간에 상관관계가 큰 것으로 나타났다. 또한, 상수항이 있는 회귀식은 상수항의 값이 약 58인데, 이는 소프트웨어 규모와 상관없이 58M/M가 투입됨을 의미하므로 실제 상황과 부합되지 않는다고 볼 수 있다. 실제로 상수항이 있는 경우의 R^2 값이 상수항이 없는 경우의 R^2 값보다 작아, 회귀식의 설명력면에서도 상수항이 없는 경우가 더 좋음을 알 수 있다. 그리고, <그림 2>에 나타난 바와 같이 하나 밖에 없는 400M/M이상인 사업에 의해 회귀식의 정확성이 떨어짐이 발견되

〈표 11〉 기능요소간 상관계수 및 유의확률

	IE	EE	IS	OS	RO
IE	1.000 (-)	0.352 (0.007)	0.790 (0.000)	0.766 (0.000)	0.681 (0.000)
EE		1.000 (-)	0.244 (0.048)	0.221 (0.066)	0.091 (0.269)
IS			1.000 (-)	0.907 (0.000)	0.657 (0.000)
OS				1.000 (-)	0.596 (0.000)
RO					1.000 (-)

〈표 12〉 모형'3에 의한 회귀결과(입력/출력화면수 통합)

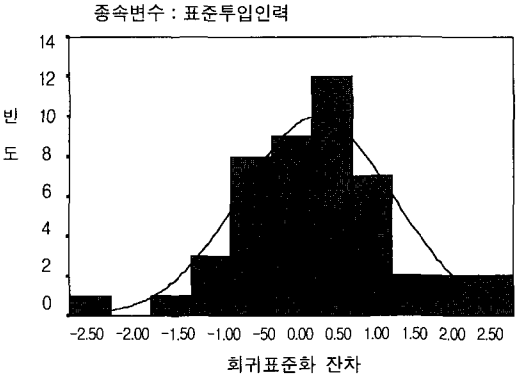
	회 귀 식	R^2	계수값 및 유의확률				
			상수	IE	EE	IOS	RO
상수항이 있는 경우	표준투입인력수 = $a + b \times IE + c \times EE + d \times IOS + f \times RO$	0.451	58.856 (0.002)	0.385 (0.029)	0.988 (0.021)	0.181 (0.003)	-0.434 (0.324)
상수항이 없는 경우	표준투입인력수 = $b \times IE + c \times EE + d \times IOS + f \times RO$	0.813	-	0.554 (0.003)	1.089 (0.020)	0.271 (0.000)	-0.313 (0.514)

〈표 13〉 모형 3에 의한 회귀결과(상관관계가 있는 독립변수 제거)

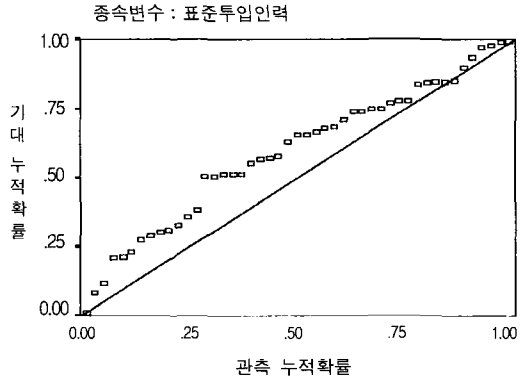
회 귀 식	R^2	계수값 및 유의확률	
		EE	IOS
표준투입인력수 = $c \times EE + d \times IOS$	0.749	1.155 (0.031)	0.392 (0.000)

었다. 따라서, 400M/M이상인 사업 한 개를 제거하고 통계적으로 상관관계가 없는 입출력화면(IOS)과 외부엔티티(EE)만을 독립변수로 하여 상수항 없이 표준투입인력과의 회귀식을 구하면 <표 13>과 같다. 이 때, 입출력화면과 외부엔티티간의 상관계수는 0.171, 유의확률은 0.125로 유의수준 0.05보다 크므로 독립변수인 입출력화면과 외부엔티티간의 상관관계는 없다고 볼 수 있다. 또한, 독립변수 각 계수의 유의확률이 0.05보다 작음을 알 수 있다.

본 연구의 목적이 소프트웨어 개발비용 예측을 위한 정밀한 통계적 모형의 제시하는 것보다는 국내 상황에 맞는 실용적 예측 모형을 제시하는데 있지만, 제시된 모형의 정확성 확인을 위해 <표 13>에 제시된 회귀식이 중회귀모형의 기본가정에서 크게 벗어나지 않는지를 검증해 볼 필요가 있다. 먼저 독립변수간의 상관관계을 의미하는 다중공선성(multicollinearity)가 존재하는가를 확인하기 위해 공선성 통계량을 보면, 공차한계(tolerance)가 0.971, 0.971로 다중공선성은 낮다고 볼 수 있다. 두 번째로, 잔차간에 자기상관성(autocollelation)이 존재하는가를 알아보기 위해 Durbin- Waton 테스트의 검정통계량을 보면, 그 값이 1.578로 자기상관성이 존재하지 않는다고 볼 수 있다. 세 번째로, 잔차가 정규분포를 따르는가를 알아보기 위해 잔차의 히스토그램과 산포도를 보면 <그림 3>, <그림 4>와 같다. <그림 3>과 <그림 4>를 보면 잔차가 정확히 정규분포를 따르지는 않지만 정규분포 가정에 크게 어긋나지 않음을 볼 수 없다.

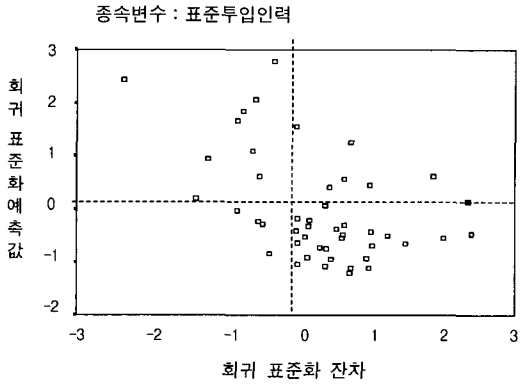


〈그림 3〉 잔차의 히스토그램



〈그림 4〉 잔차의 정규 P-P 도표

마지막으로, 종속변수가 독립변수의 변화에 따라 다른 분산을 보이는 이분산성(heteroscedasticity)이 존재하는가를 확인하기 위해, 잔차와 예측값의 산포도를 그려보면 <그림 5>와 같으므로, 이분산성이 존재한다고 할 수 없다. 따라서, <표 13>에 제시된 회귀모형은 중회귀분석의 기본가정을 어느 정도 만족하고 있다고 볼 수 있다.



〈그림 5〉 잔차의 예측 산포도

<표 13>에 제시된 회귀모형은 입출력화면수와 외부엔티티수라는 두 개의 독립변수만을 포함하고 있다. 그러나, 제시된 모형의 독립변수는 기능점수 모형에서 고려한 기능요소들을 대부분 포함하고 있다. 예를 들어, 기능점수 모형의 트랜잭션기능인 외부입력, 외부출력, 외부조회는 입출력화면수에 모두 포함되어 있다고 볼 수 있다. 다만 기능점수 모형에서 고려하고 있는 데이터기능 중 내부논리파일에 대응하는 내부엔티티수가 누락되어 있는데, 이는 Kitchenham & Känsälä[16]와 Lokan [18]의 연구에서 확인되었듯이 상관관계가 큰 변수를 제거하였기 때문이다. 이는 내부엔티티수는 입출력화면수와 밀접한 관계를 갖고 있어, 입출력화면수로도 내부엔티티수의 변화에 의한 투입인력의 변화를 충분히 설명할 수 있음을 의미한다.

<표 13>에 제시된 회귀모형를 살펴보면 외부엔티티의 계수가 입출력화면의 계수보다 약 4배정도 큼을 알 수 있다. 이는 외부엔티티 1개에 소요되는 M/M가 입출력화면 1개에 소요되는 M/M보다 4배 많음을 의미한다. 한편, <표 2>과 <표 5>에서 알 수 있듯이 기능점수 모형에서 외부참조파일은 외부입력/외부조회에 비해 약 1.7배정도 기능점수가중치가 높다. 이렇게 간이기능점수 모형과 기능점수 모형 사이에 차이가 발생하는 것은 국내 정보화사업의 특성에서 그 이유를 찾을 수 있다. 외부엔티티가 존재한다는 것은 기관/시스템간 연계가 필요한 사업임을 의미하고, 타기관 또는 타 정보시스템의 정보를 참조하기 위해서는 공공기관 특성상 많은 협의와 복잡한 절차를 요하게 된다. 이로 인해 인력자원을 상대적으로 많이 소모하게 되므로, 선진국의 상황을 기반으로 만들어진 기능점수 모형과 차이가 발생한다고 볼 수 있다.

5. 결 론

본 연구에서는 정보화예산의 합리적 · 객관적 산정을 위해 해외에서 널리 사용되고 있는 기능점수 모형의 국내 정보화사업 적용을 모색하였다. 이를 위해, 사업초기단계에 사용할 수 있도록 기능점수 모형을 수정한 간이기능점수 모형을 제시하였고, 기존 공공정보화사업 분석 결과를 토대로 간이기능점수 모형의 기능요소와 표준투입인력간의 회귀식을 제시하였다. 본 연구에서 제시된 모형은 사업초기 단계에 적용할 수 있다는 점 외에도 기능요소와 투입인력간의 상관관계를 바로 구했다는 점에서 기존 연구와 구별된다. 또한, 본 연구에서 제시된 회귀모형은 해외 선진국의 경우와는 다른 결과들을 제시하였다. 이는 선진국의 현실을 토대로 개발된 기능점수 모형를 국내에 적용하기 위해서는 국내 정보화사업의 특성에 맞게 수정 · 보완하는 연구가 필요함을 의미한다.

본 연구결과는 정보화예산 심의에 실질적으로 활용되고 있으며, 추후 예산 심의 결과를 토대로 모형을 보다 발전시킬 계획이다. 첫 번째로, 본 연구에서 사용한 정보화사업들에 대해 정확한 기능점수를 산정하여 기능점수 모형의 기능요소와 간이기능점수 모형의 기능요소간의 상관관계를 규명함

으로써, 간이기능점수 모형의 유용성과 기능점수모형과의 호환성을 체계적으로 검증할 필요가 있다. 또한, 기존에 개발된 예측모형들에서 사용되는 독립변수에 관한 실제 자료를 수집하여 기존 모형과의 비교·분석과 정확성 비교도 이루어져야 할 것이다. 두 번째로, 사업특성을 반영할 수 있는 보정계수를 개발하고 보정계수 도입으로 인한 모형의 정확도 향상 여부를 검증할 필요가 있다. 마지막으로, 보다 큰 규모의 정보화사업을 분석하고, 운영 및 유지보수사업에 대한 모델을 추가하여 모형의 활용범위를 확장시켜야 할 것이다.

참 고 문 헌

[1] 김현수, "기능점수를 이용한 소프트웨어 규모 및 비용산정 방안에 관한 연구", 「경영과학」, 제14권, 제1호(1997. 5), pp.131-149

[2] 이양규, "기능점수 모형을 이용한 소프트웨어 개발 비용 산정", 「경영연구」, 제6권(1997), pp. 241-261.

[3] 한국소프트웨어산업협회, http://www.sw.or.kr.

[4] 한국전산원, "정보시스템 운영을 위한 비용산출 방안에 관한 연구", 1999.

[5] Albrecht, Allan J., "Measuring application development productivity," in *Proc. IBM Application Deveoplment Symp.*, GUIDE Int. and SHARE Inc., IBM Corp., Monterey, CA, Oct. 14-17, 1979, pp.83-92.

[6] Boehm, B.W., et al., *"Software cost estimation with COCOMO* II," Prentice Hall PTR, 2000.

[7] Caldiera, G., G. Atoniol, R. Fiutem, C. Lokan, "Definition and experimental evaluation of function points for object-oriented systems," *Proceedings of the 1998 5th International Software Metrics Symposium*, Bethesda, MD, USA, Nov 20-21, 1998. pp.167-178,

[8] Freeman, R.J., "An empirical study of the linkage of CASE, function points and systems development," *Proceedings of the 5th International Workshop on Computer-Aided Software Engineering*, Montreal, Quebec, Canada, Jul. 6-10, 1992, pp.254-257,

[9] Furey, Sean, "Why we should use function points," *IEEE Software*, Vol.4, No.3(1997), p.28

[10] IFPUG, *Function Point Counting Practices Manual(Release 4.1.1)*, International Function Point Users Group, 2000.

[11] International Software Benchmarking Standards Group, *http://www.isbsg.org.au/macroest.htm.*

[12] Jeffery, D.R., G.C. Low, M. Barnes, "A comparison of function point counting techniques," *IEEE Transactions on Software Engineering*, Vol.19, No.5(1993), pp.529-531

[13] Jones, T. Capers, "A short history of function points and feature points," Software Productivity Research Inc., 1988.

[14] Jones, T. Capers, *Estimating Software Costs*, McGraw-Hill, 1998.

[15] Kemerer, C.F., B.S., Porter, "Improving the reliability of function point measurement : An empirical study," *IEEE Transactions on Software Engineering*, Vol.18, No.11(1992), pp.1011-1024

[16] Kitchenham, B., Kari Kansala, "Inter-item correlations among function points," *Proceedings of the 15th International Conference on Software Engineering*, Baltimore, MD, CA, USA, May 17-21, 1993, pp.477-480.

[17] Kitchenham, B., "The problem with function points," *IEEE Software*, Vol.4, No.3(1997), p.29.

[18] Lokan, Chris J., "Empirical study of the cor-

relations between function point elements," *Proceedings of the 1999 6th International Software Metrics Symposium*, Nov. 4-6, 1999, pp.200-206.

[19] Low, G.C., D.R., Jeffery, "Function points in the estimation and evaluation of the software process," *IEEE Transactions on Software Engineering*, Vol.16, No.1(1990), pp.64-71.

[20] Matson, J.E., B.E. Barrett, J.M. Mellichamp, "Software development cost estimation using function points," *IEEE Transactions on Software Engineering*, Vol.20, No.4(1994), pp.275-287

[21] Mizuno, O., T. Kikuno, K. Inagaki, Y. Takagi, K. Sakamoto, "Analyzing effects of cost estimation accurracy on quality and productivity," *Proceedings of the 1998 20th International Conference on Software Engineering*, Kyoto, Japan, Apr 19-25, 1998. pp.410-419.

[22] Mukhopadhyay, T., S. Kekre, "Software effort models for early estimation of process control application," *IEEE Transactions on Software Engineering*, Vol.18, No.10(1992), pp.915-924

[23] Orr, G., T.E. Reeves, "Function point counting : one program's experience," *Journal of Systems and Software*, Vol.53. No.3(2000), pp.239-244

[24] Pressman, R.S., *Software engineering : A practitioner's approach*, 4th ed., McGraw-Hill, 1997.

[25] Putnam L., W. Myers, *Measures for excellence*, Yourdon Press, 1992.

[26] Symons, C.R., "Function point analysis : difficulties and improvements," *IEEE Transactions on Software Engineering*, Vol.14, No.1(1988), pp.2-11

[27] Uemura, T., S. Kusumoto, K. Inoue, "Function point measurement tool for UML design specification," *Proceedings of the 1999 6th International Software Metrics Symposium*, Nov. 4-6, 1999, pp.62-69.

[28] Vidger, M.R., A.W. Kark, *Software cost estimation and control, Institute for Information Technology*, National Research Council Canada, 1994.

[29] Yokoyama, Y., M. Kodaira, "Software cost and quality analysis by statistical approach," *Proceedings of the 1998 20th International Conference on Software Engineering*, Kyoto, Japan, Apr. 19-25, 1998, pp.465-467.