

소프트웨어 비용산정을 위한 기능점수 모형 개선 연구

김 현 수[†]

요 약

본 연구는 기능점수 모형의 예측능력과 활용성을 높이는 동시에 소프트웨어 사업대가 기준의 실용성을 제고하는 목적으로 수행되었다. 기능점수 개선모형 연구는 최근에도 몇가지 연구가 수행된 바 있다. 따라서 본 연구에서는 이들 연구 결과를 활용하여 모형을 개선하는 작업을 수행하였다. 우선, 기존의 국내 연구를 보완하고, 전문가의 자문을 통하여 실용성이 높은 후보 모형을 도출하였다. 현재와 같은 프로젝트 환경에서는 실제 데이터에 의한 통계적으로 완벽한 검증을 기대하기 어려우므로, 현업 프로젝트 전문가의 경험에 의한 검증이 매우 중요한 역할을 한다. 따라서 본 연구는 이러한 과정을 충실하게 거쳐 후보 모형을 도출하였다. 특히 기술적 복잡도 판단 기준표는 기능점수 모형에서 제시되지 않은 독자적인 기준으로서 한국형 기능점수 모형의 대표적인 특징이라 할 수 있다.

또한 본 연구는 39개의 프로젝트 실제 데이터를 이용하여 후보 모형의 타당성을 검증하였다. 제안된 후보 모형이 전문가에 의한 경험적인 타당성과 함께 일정 수준 이상의 통계적 유의성을 가지고 있음을 입증하였다. 본 연구의 결과는 산업계와 정부 등의 관련기관 실무 회의를 거쳐 소프트웨어 사업의 대가기준으로 활용될 수 있을 것이다.

An Improvement of Function Point Models for Software Cost Estimation

Hyunsoo Kim[†]

ABSTRACT

There is a strong need to develop a software cost estimation model on economic value perspective. The objective of this research is to improve current software cost estimation method on economic value perspective. We reviewed domestic and foreign researches and practices on software cost estimation with function point method, and derived promising alternative models. Pilot simulation was performed with real project data, and the probable best model was chosen.

We collected data from 39 Korean companies, and assessed statistical significance of the model with those data. Empirical data shows that more practical model has better prediction accuracy. That is, the number of input and output modules, the number of tables, and the number of algorithms are chosen to be the best set of functions. There exists strong correlation between the calculated function points and project effort. And, the revised set of technical complexity factors and evaluation guidelines show practical usefulness. We suggest that the above result be incorporated in a new improved guideline for software cost estimation. By adopting the results of this research to the guideline, we expect that technology innovation will be expedited, and that overall productivity of software industry will be increased.

※ 이 연구는 한국소프트웨어산업협회의 연구비 지원으로 수행되었음.

† 종신회원 : 국민대학교 정보관리학부 교수

논문접수 : 1999년 4월 7일, 심사완료 : 1999년 7월 14일

1. 서 론

소프트웨어 비용산정 방식의 최근 동향은 사용자 가치 중심의 산정 방식이다. 사용자 중심의 경제적 가치를 기준으로 소프트웨어의 규모를 견적하는 대표적인 비용산정 모형이 기능점수(Function Point) 모형인데, 이 모형은 정보처리 규모와 기술적 복잡도 요인에 의하여 소프트웨어의 규모를 산정한다. 정보처리 규모는 사용자 입장에서 본 시스템의 기능을 외부입력, 외부출력, 내부논리파일, 외부인터페이스파일, 외부조회 등의 5가지 유형으로 나누어 각 기능의 복잡도를 고려하여 측정한다. 기술적 복잡도 요인은 데이터통신, 분산처리 등 14가지의 시스템 특성에 의해 계산하며, 이 값이 정보처리규모에서 구한 기능수치를 보정하여 최종 기능점수를 계산한다.

본 연구는 기능점수 모형의 현재 예측능력과 활용성을 제고하는 동시에 소프트웨어 사업대가 기존의 실용성을 제고하는 목적으로 수행되었다. 기능점수 개선모형 연구는 최근에도 일부 연구가 수행된 바 있다. [1][2][3][4][5][12] 본 연구에서는 이들 연구 결과를 활용하여 모형을 개선하는 작업을 수행하였다. 또한 실용성을 높이기 위해서는 개발자 집단의 공통의 이해가 필요하고, 실제 데이터에 의한 후보 모형의 시뮬레이션이 필요하므로, 연구 수행 방법상의 차별화를 추구하였다. 즉 실무 전문가와 공동의 팀을 구성하여 후보모형을 반복적으로 검증하는 방식을 통하여 모형을 개선해가며, 개선된 최종 모형에 대해 광범위한 실제데이터를 수집하여 통계적인 검증을 수행하였다.

제 2 장에서는 기능점수 모형 및 관련 연구를 소개하고, 최신 결과를 요약하여 제시한다. 제 3 장에서는 현 시점에서 타당한 3가지의 후보 모형을 제안한다. 제안된 후보 모형은 대형 SI 업체의 전문가들을 중심으로 구성된 자문팀에 의해 수차례의 수정을 거쳐 검증된 모형이다. 제 4 장에서는 광범위한 프로젝트 데이터를 수집하여 모형을 검증한 결과를 제시하고 토의를 수행한다. 마지막으로 제 5 장에서는 연구결과를 요약하고 향후 연구방향을 토의한다.

2. 기능점수 모형 연구

기능점수는 기능수 계산, 기술적 복잡도 계산, 기능점수 계산 등의 3단계 절차에 의해 계산된다. 기능 수

(Function Count : FC)는 정보처리규모를 계산하는데, 개발 대상 소프트웨어가 명확하게 정의되고, 외부 시스템과의 인터페이스가 분명해진 상태에서 다음과 같이 5가지 유형의 기능을 추출하여 각각의 복잡도를 측정한다. [11]

- ① 외부입력 (external input types) : 시스템(소프트웨어시스템을 의미함)의 경계 밖에서 시스템에 데이터를 입력하는 기능.
- ② 외부출력 (external output types) : 시스템의 경계밖으로 데이터를 출력하는 기능.
- ③ 내부논리파일 (logical internal file types) : 시스템 내에서 논리적으로 유지되는 파일로서 사용자 관점에서 볼때 시스템에 의해 생성되고, 사용되며, 또한 관리되는 논리적인 데이터의 그룹.
- ④ 외부인터페이스파일 (external interface file types) : 응용시스템간에 공유되거나 전송되는 파일로서 시스템 밖의 응용시스템이 논리적으로 유지하는 파일.
- ⑤ 외부조회 (external inquiry types) : 입력이 출력을 즉시 요구하는 입력/출력조합을 외부조회로 판단한다. 입력 또는 출력중의 하나라도 포맷(format)이 다르거나, 내부 처리로직이 다르면 별개의 기능으로 함.

각 기능의 복잡도는 단순, 보통, 복잡의 3 단계로 구분하여 평가한다. 복잡도에 주어지는 가중치는 기능유형별로 상이하다.[7][8] 기능수(FC)는 기술적 복잡도를 이용하여 생산성을 측정하는 지표인 기능점수로 변환된다. 기술적복잡도는 아래와 같은 14개의 항목에 대해 영향도를 평가하여 계산한다. 영향도(Degree of Influence : DI)는 0 부터 5까지의 정수로 나타낸다.

- Ⓐ 데이터 통신의 필요 정도
- Ⓑ 분산처리 기능의 정도
- Ⓒ 응답속도나 처리율(throughput)과 같은 시스템의 성능(performance) 요구
- Ⓓ 운용될 시스템의 여유정도 (사용정도)
- Ⓔ 트랜잭션율(Transaction rate)이 높은 정도
- Ⓕ 온라인 자료입력 및 제어기능 요구정도
- Ⓖ 온라인입력이 복수개의 스크린 또는 오퍼레이션을 사용하는 트랜잭션으로 구성되는 정도
- Ⓗ 내부논리파일을 온라인 생산하는 정도
- Ⓘ 처리의 복잡성 정도 (예: 많은 제어 상호작용과 의

사실성 수반, 광범위한 논리 및 수학적 사용, 많은 예외처리로 인한 불완전한 처리)

- ㉑ 프로그램 재사용의 고려 정도
- ㉒ 변환 및 설치 용이성을 고려하는 정도
- ㉓ 효과적인 백업, 복구 등 운용상의 편리함을 고려하는 정도
- ㉔ 복수의 조직을 위한 복수 장소에의 설치를 고려하는 정도
- ㉕ 변경 용이성을 고려하는 정도

14개 항목의 영향도를 평가하여 합산한 총영향도는 0 부터 70 사이의 값이 된다. 기술적 복잡도값인 TCF (Technical Complexity Factor)는 다음 식으로 계산된다.

$$TCF = 0.65 + 0.01 \times \text{총영향도}$$

기능점수(FP)는 FC와 TCF로부터 다음과 같이 계산된다.

$$FP = FC \times TCF$$

기능점수 모형은 사용자 관점에서 소프트웨어를 견적하는 유용한 방법이지만, 적용범위, 측정의 신뢰성, 가중치 적용 등에서 여러가지 문제점을 내포하고 있다. 기능점수 모형을 개선하는 연구는 모형의 구조를 개선하는 방향, 활용성을 향상하는 방안, 적용범위를 확대하는 방향등 3가지 방향으로 진행되고 있다. 각 방향에 대한 주요 연구결과는 다음과 같으며, 상세한 내용은 [2]의 연구를 참조하기 바란다.

모형 구조의 개선 방향으로 대표적인 연구는 Symons [16]와 일본 COSDES(Committee On Software Development Estimation System)의 연구[4,5]이다. Symons는 Albrecht의 기능점수 모형이 기능의 복잡도를 너무 단순화하고 있으며, 기능의 가중치가 많은 토론과 시행착오를 거쳐 결정되었지만, 여전히 많은 의문이 제기되고 있다고 지적한다. 일본 COSDES의 기능점수모형은 보다 많은 구조 변형을 시도한 모형이다. COSDES에서 사용하는 기능 유형은 화면, 대장(출력 보고서), 파일로서 화면은 다시 입력,출력, 입출력, 메뉴로 세분되고, 파일은 입력, 출력, 입출력으로 세분하여 기능점수를 계산한다. 클라이언트/서버 시스템은 하드웨어의 수나 종류, 개발도구, 미들웨어, 데이터베이스 등을 자유롭게 조합시킬수 있어 자유도가 매우 높아진다. 따라서, 시스템을 여러개의 패턴으로 나누어 각 패

턴마다 최적의 견적방법을 개발하는것이 바람직한데, 패턴의 구분 기준으로는 시스템 구조(서버의 종류, 클라이언트의 종류, C/S 시스템의 계층구조 등), 소프트웨어 (OS, 데이터베이스, 클라이언트 개발 도구 등), 개발방법(프로그래밍 언어 중심, 또는 패키지 조합 중심 등) 등 3가지 항목에 기초하는 방안을 모색하고 있다.[5]

모형의 신뢰성 향상 연구로서 Kemerer와 Porter [14]는 기능점수 측정상의 주요 오차요인에 대한 경험적인 분석을 수행하여 백업파일을 세는 방식 등 비교적 소수의 요소만이 기능점수 측정치의 오류수준에 영향을 미친다고 분석하였다. 한편 Low와 Jeffery[15]는 한 조직내에서 기능점수를 사용할때, 측정치의 오차는 30% 이내인 것으로 분석하였다.

기능점수 모형은 원래 입출력을 중심으로 하는 사무처리시스템에 적용하는 목적으로 개발되었기 때문에, 적용범위에 대한 확장이 필요하다. 즉 입출력의 양은 많지 않지만, 내부에서 많은 복잡한 계산이 이루어지는 소프트웨어의 견적을 위해 Jones[13]는 기능점수 모형을 변형하여 특성점수(Feature point)을 개발하고, 이를 시스템 및 엔지니어링용 소프트웨어의 견적에 적용할것을 제안하였다. 특성점수를 계산하는 방법은 기능점수를 계산하는 방식과 유사하다.[13]

$$\text{특성점수} = \text{특성수} \times TCF$$

대표적인 국내 연구로서, [2]에서는 가장 바람직한 개선 모형으로서 다음 결과를 제시하였다. 즉 기능수(FC)를 추정하기 위해 입력데이터 유형의 수, 실체유형(Entity type)수, 알고리즘수, 출력데이터 유형의 수 등 4개 변수를 사용하여 다음과 같은 모형을 도출하였다.

$$FC = 692 + 0.426 * N_I - 1.165 * N_E + 6.739 * N_A + 6.232 * N_O$$

(R²=0.9342, Adjusted-R² = 0.9078, Prob>F: 0.0001)

여기서 N_I = 입력데이터 유형의 수
 N_E = 실체유형(Entity type) 수
 N_A = 알고리즘수
 N_O = 출력데이터 유형의 수

위 모형은 소프트웨어 규모와 비용을 초기에 추정할수 있는 실용성있는 모델로 개발되었다. 그러나 이 모형은 기능 요소 각각에 대한 개별적인 검증이 부족하였고, 특히 복잡도를 판단하는 상세한 기준이 제시되

지 않아, 사용에 어려움이 있었다. 본 연구에서는 이 모형에 대한 후속 연구를 수행하였다. 즉 기능 요소 각각의 타당성을 검토하여 보다 실용적이고 예측력이 높은 기능요소를 도출하고, 기술적 복잡도 요소를 보완하고, 각 요소에 대한 영향도 판단기준을 개발하는 것이 본 연구의 주요 내용이다. 아래 제 3 장에서 개선 후보 모형을 제시한다.

3. 기능점수 개선 모형

본 연구에서는 우선 기능수 계산에 대하여 2개의 후보 모형을 도출하였고, 기술적복잡도 계산을 위해서는 3개의 후보 모형을 도출하였다. 이렇게 복수개의 후보 모형을 도출한 이유는 전문가들이 폭넓고 합리적인 의사결정을 할 수 있도록 도와주기 위해서이다. 전문가 자문을 거쳐 기술적 복잡도 모형은 하나로 수렴되고, 세부적인 기준을 수정하여 완성되었다. 본 연구에 적극적으로 참여한 전문가 집단은 8명으로 구성되어 있으며, 대형 SI 업체에서 프로젝트관리와 비용산정 분야에 종사하는 특급기술자 5명과 공공기관의 중견 간부로서 비용산정업무에 다년간 자문 및 연구경력을 가진 전문가 1명, 소프트웨어사업대가기준 주관기관인 한국소프트웨어산업협회의 담당과장 및 주무부처인 정보통신부의 담당사무관 등 8명이다. 전문가 검토회의는 3차에 걸쳐 수행되고, 수시로 전자우편을 통하여 의견수렴을 하였다. 검토회의 기록은 [1]에 요약되어 있으므로 여기에서는 최종 결과만을 제시한다. 검토 결과 모형은 기능수 계산의 경우 3가지 방법을 사용하며, 기술적 복잡도 계산에서는 수정된 단일 모형을 사용한다. 후보 모형에 대한 설명은 다음과 같다.

3.1 기능수 계산

기능수를 계산하기 위한 후보 모형으로 아래 3개 모형을 도출하고, 프로젝트 실패데이터를 수집하였다.

3.1.1. 모형 I :

$$FC_1 \text{ (조정하기 전의 기능점수)}$$

$$= a \times N_I + v \times N_E + w \times N_A + c \times N_O$$

여기서 N_I = 입력데이터 유형의 수
 N_O = 출력데이터 유형의 수
 N_A = 알고리즘의 수
 N_E = 참조되는 실체유형(Entity Type) 수

본 모형은 IFPUG의 기능점수 모형의 기본 원리에 충실하면서, 프로젝트 수행초기에 실용적으로 건적이 가능하도록 개발된 모형이다. 기본 지수 항목의 세부 설명 및 정의는 다음과 같다. 이 부분은 수차례의 전문가 검토회의를 거쳐 확정되었다.

- N_I (입력데이터 유형의 수) : 입력 데이터의 총 자료 항목수. 자료사전(사용하는 방법론에 따라 다른 이름의 문서일 수 있으나, 기본적으로 자료에 대한 서술이 있는 문서의 통칭임)에 나타나있는 항목중에서 입력항목의 수를 의미함. 여러번 나타나도 중복해서 계산(count)해서는 안되며, 고유(unique)한 유형수만을 계산해야함.
- N_O (출력데이터 유형의 수) : 출력데이터의 총 자료항목수. 자료사전에 나타나있는 항목중에서 출력 항목의 수를 의미함. 여러번 나타나도 중복해서 계산(count)해서는 안되며, 고유(unique)한 유형수만을 계산해야함.
- N_E (참조되는 실체유형수) : 트랜잭션에서 생성, 수정, 조회, 삭제를 위해 참조하는 실체유형 수를 의미함. 트랜잭션 전체에 대해 일일이 계산하기 어려운 경우, 단순히 '실체유형수'로 처리할 수 있음.
- N_A (알고리즘수) : 입출력이 아닌, 내부 프로세스의 복잡도를 반영하기 위한 항목임. 알고리즘은 '특정한 컴퓨터 프로그램에 포함된 경계있는 계산문제(a bounded computational problem that is included within a specific computer program) 및 SI 업체에 의해 직접 구현되는 비즈니스 로직(Business Logic) 중 다수의 호출에 의해 사용되는 공통부분(Public Function & Subroutine)'으로 정의됨.

3.1.2 모형 II :

모형 I 의 요소들은 본래의 기능 요소에 충실하기는 하지만, 프로젝트 현장에서 계산하기 쉽지 않을 수 있다. 계산이 보다 용이한 모듈수와 테이블수 등의 유사요소로 대체한 모형은 다음과 같다.

$$FC_2 \text{ (조정하기 전의 기능점수)}$$

$$= a \times N_{IM} + v \times N_T + w \times N_B + c \times N_{OM}$$

여기서 N_{IM} = 입력모듈수
 N_{OM} = 출력모듈 및 장표 수
 N_B = 배치프로그램 수
 N_T = 테이블 수

기능점수를 구성하는 항목 중 모듈은 아래와 같이 정의한다.

- ‘모듈(module)’의 정의: 독립적으로 컴파일되고 독립적인 세부기능 (즉 입력, 조회, 수정, 출력 등)을 수행하는 프로그램 개별단위로서 다른 프로그램에 의해 호출되어 수행되며, 그 프로그램의 구성요소가 될 수 있는 프로그램의 최소단위임. 프로그램은 1개 이상의 모듈로서 구성되며, 모듈은 대개 작성된 프로그램의 프로시저, 함수 또는 부프로그램임.

3.1.3 모형 III:

모형 II에서 배치프로그램수는 계산이 용이한 장점은 있으나, 요소로서의 논리적타당성이 부족하므로, 이를 타당성이 높은 알고리즘수로 대체한 모형 III은 다음과 같다.

$$FC_3 \text{ (조정하기 전의 기능점수)} = a \times N_{IM} + v \times N_T + w \times N_A + c \times N_{OM}$$

- 여기서 N_{IM} = 입력모듈수
- N_{OM} = 출력모듈 및 장표 수
- N_A = 알고리즘 수
- N_T = 테이블 수

기능수에 부여되는 가중치에는 시스템구조가중치와 웹구현가중치가 있다. 기능수에 대한 가중치 부여의 타당성 여부는 실험데이터 분석을 통하여 판단한다. 즉 가중치를 모델에 포함시킨 것의 통계적 유의성이 높으면 가중치가 타당하다고 판단하고, 가중치를 사용하지

않은 모형보다 가중치를 사용한 모형의 통계적 유의성이 크나지 높지 않으면, 가중치가 타당하지 않다고 판단한다. 각 모형에서 기능수는 시스템구조와 웹구현가중치를 곱하여 아래와 같이 계산될 수 있다.

$$FC \text{ (기능수)} = \sum_{i=1,2,3} FC_i \times \text{시스템구조 가중치} \times \text{웹구현 가중치}$$

시스템구조 가중치는 전문가들이 권고하는 가중치로서 호스트 중심구조는 1.0, 2 tier Client/Server 구조는 1.1, 3 tier Client/Server 구조는 1.2를 초안으로 제시하고, 검증한다.

3.2 기술적 복잡도(Technical Complexity) 계산

양식에 제시된 17 개 항목을 통하여 총 영향도를 구하고, 아래의 공식을 사용하여 기술적 복잡도를 계산한다.

$$TCF \text{ (기술적 복잡도)} = 0.50 + 0.01 \times \text{총영향도}$$

기술적 복잡도를 계산하기 위한 영향도 판단기준은 다음 표와 같다. 이 표는 수차례의 전문가 검토회의를 거쳐 조정된 표이다.

기능점수는 아래 식과 같이 기능수와 기술적 복잡도의 곱으로 계산된다.

$$FP \text{ (기능점수)} = FC \times TCF$$

소요노력은 다음과 같이 사용상의 편리함을 고려하

<표 3-1> 기술적 복잡도 판단기준

(요소의 영향이 전혀 없는 경우 '0 점' 처리)

복잡도 요소	단순 (1 점)	보통 (3 점)	복잡 (5 점)
데이터통신 필요성도	데이터의 전송으로 통신처리 거의 불필요	데이터가 일부 분산 매일 1-2시간 통신처리 (1-2시간에 1개정도 트랜잭션 발생)	대량의 데이터가 분산 매일 3시간 이상통신처리 (30분에 1개 이상 트랜잭션 발생)
운용될 시스템의 여유정도 (사용정도)	운영상의 제약이 거의 없음	운영상의 제약이 보통 정도임.	시스템운영상의 제약이 많음
온라인자료입력 및 제어기능 수준	온라인 자료입력과 제어가 거의 없는 경우 (10 % 미만)	온라인 자료입력 및 제어기능이 보통 (20 - 30 %)	온라인자료입력 및 제어가 많은 경우 (50 % 이상)
온라인 입력이 복수개 오퍼레이션/스크린 트랜잭션으로 구성되는 정도	거의 없음 (10% 미만)	보통임 (20 - 30 %)	매우 많음 (50 % 이상)

〈표 3-1〉 계속

(요소의 영향이 전혀 없는 경우 '0 점' 처리)

복잡도 요소	단순 (1 점)	보통 (3 점)	복잡 (5 점)
내부논리파일 온라인 갱신정도	거의 없음 (5 % 미만) 갱신되는 파일의 수가 적은 경우	보통임 (10 % 내외) 갱신되는 파일의 수가 보통인 경우	매우 많음 (20 % 이상) 갱신되는 파일의 수가 많은 경우
프로그램 재사용 고려정도	향후 재사용 대상 프로그램이 거의 없음	일부 프로그램 향후 재사용 (20 % 내외)	다수 프로그램이 재사용 대상 (30 % 이상)
백업/복구등 운용상 의 편리함 고려성도	거의 편리함을 고려하지 않음.	편리함을 많이 고려함	수작업 필요성을 거의 없앴
변경용이성 고려성도	거의 고려하지 않음 (대부분의 검색항목이 고정적임)	보통으로 고려함 (변경 가능한 검색항목이 다수 있음)	크게 고려함 (대부분의 검색항목이 변경가능함)
분산처리가능 정도	거의 분산처리가 수행되지 않음.	2-3 사이트에서 분산처리 수행	5 사이트 이상에서 분산처리
타시스템 인터페이스정도	인터페이스가 1개 이하	3-4 개 시스템과 인터페이스	7개 이상 시스템과 인터페이스
특별한 보안기능 요구정도	낮은 안기부 보안등급 (구체적 보안 요구 없음)	보통의 보안기능요구 (기능단위의 보안요구)	높은 보안가능 요구 (프로그램단위의 보안요구)
변환 및 설치용이성을 고려하는 정도	용이성을 거의 고려하지 않음	보통정도로 고려함	변환 및 설치가 용이하게 되도록 많은 고려를 함.
문서화 요구정도	문서화 요구가 거의 없음	필수 산출물에 대한 문서화 요구 있음.	방법론상의 모든 산출물에 대한 문서화 요구 있음
사용자 요구 하드웨어, 소프트웨어 정의, 선택, 설치 정도	거의 없음	2-3 가지 있음	5가지 이상 있음
처리의 복잡성 정도	매우 적은 제어 상호작용과 의사결정 수반, 논리 및 수학적 사용, 예외처리로 인한 불완전한 처리가 거의 없는 경우	보통의 제어 상호작용과 의사결정 수반, 논리 및 수학적 사용, 보통의 예외처리로 인한 약간의 불완전한 처리	많은 제어 상호작용과 의사결정 수반, 광범위한 논리 및 수학적 사용, 많은 예외처리로 인한 불완전한 처리
복수조직/복수장소 설치고려정도	2개 이하의 사이트에 설치	4-5 개의 사이트에 설치	8개 이상의 사이트에 설치
응답속도/처리를 같은 시스템 성능 요구	별다른 요구가 없는 경우 (예: 급여시스템)	일상적 수준의 성능기준 요구 (예: 생산관리)	매우 높은 성능기준 요구 (예: 은행 온라인, 지휘통제)

여 선형모형으로 구축한다.

$$\text{소요노력(MM)} = a + b \times \text{FP}$$

여기에서 계수 a, b 는 프로젝트 실 데이터에 의한

시뮬레이션을 거쳐 수치가 확정된다.

4. 실데이터 분석 결과

위의 모형을 검증하기 위하여 조사서 500부를 배포하고, 실제 프로젝트 데이터 39건을 수집하였다. 조사서 배포후 2주동안 한국소프트웨어산업협회와 본 연구팀에서 작성과정을 전화로 상세히 안내하여 수집되는 데이터의 정확성을 최대한 확보하도록 노력하였다. 수집된 데이터의 상세내역은 [1]에 정리되어 있다. 실제 프로젝트 데이터를 이용한 모형 검증 결과는 다음과

같다. 기능수를 계산하기 위한 3개의 후보 모형을 중심으로 실데이터 분석 결과를 제시한다.

4.1 모형 I에 대한 분석

4.1.1 기능점수와 소요노력간의 관계분석

실 소요노력(MM)을 종속변수로 하고 입력자료유형수, 출력자료유형수, 참조 실체유형수와 알고리즘수로 계산된 기능점수(FP)를 독립변수로 하여 산출된 모형 I에 대한 기능점수 모형의 계수 추정 결과는 다음과 같다. 여기서는 각 요소의 가중치를 1로 부여하였다.

$$\text{소요노력 (표준 MM)} = 15.773 + 0.286 * \text{FP}$$

$$(R^2 = 0.728, \text{Adj-}R^2 = 0.719, \text{Prob}>F : 0.000)$$

즉 1 기능점수당 0.286 MM가 소요되는 것으로 분석되었다. R²가 충분히 높아 모형의 설명력이 있는 것으로 판단되며, 가중치를 1로 할 경우에도 기능점수로

소요노력을 비교적 정확하게 추정할 수 있는 것으로 나타났다.

4.1.2 기능수 산출에 관한 분석

여기에서는 실데이터를 이용하여 기능 요인 각각에 대한 적정 계수를 추정하였다. 실데이터 조사시에 사용된 기능점수와 소요노력과의 관계식은 대기업의 실제 자료를 사용하여 어느정도 검증된 값이므로, 타당성이 상당히 있다고 할 수 있다. 따라서, 실제노력으로부터 기능요소의 가중치를 아래 식을 이용하여 도출하였다.

- 식1 : 기능수 = a*입력데이터 유형의 수
+ b*출력데이터 유형의 수
+ c*참조되는 실체유형 수 + d*알고리즘 수
- 식2 : 소요노력 = 10 + 0.15*기능점수
- 식3 : 기능점수 = 기능수*가중치*기술적 복잡도

위의 식을 기능수에 대해 정리하면 다음과 같다.

$$\begin{aligned} \text{기능수} &= (\text{소요노력} - 10) / \\ & (0.15 * \text{가중치} * \text{기술적 복잡도}) \quad \dots \textcircled{1} \\ &= a * \text{입력데이터 유형의 수} \\ &+ b * \text{출력데이터 유형의 수} \\ &+ c * \text{참조되는 실체유형 수} \\ &+ d * \text{알고리즘 수} \quad \dots \textcircled{2} \end{aligned}$$

①을 종속변수, ②를 독립변수로 하여 중회귀분석을 수행한 결과는 다음과 같다.

$$\begin{aligned} \text{기능수(FC)} &= 91.740 + 0.08558 * \text{입력데이터 유형수} \\ &+ 0.209 * \text{출력데이터 유형수} + 0.206 * \\ &\text{실체유형수} - 0.211 * \text{알고리즘수} \quad (R^2 \\ &= 0.275, \text{Adj-R}^2 = 0.164, \text{Prob} > F: 0.070) \end{aligned}$$

즉 모형의 설명력이 낮고, 알고리즘수와 같이 유의 계수를 가지는 요인도 있어 실무에서 사용하기에 적절하지 않은 것으로 나타났다. 본 연구에서 사용한 기능수 가중치는 실제노력보다 많게 소요노력이 계산되도록 부여되었다는 의견이 있다. 따라서, 가중치를 고려하지 않은 모형에 대해서 계수를 추정하는 모형을 도출하고 회귀분석을 수행하였다. 그 결과는 다음과 같다.

$$\text{기능수} = (\text{소요노력} - 10) / (0.15 * \text{기술적 복잡도})$$

$$\begin{aligned} &a * \text{입력데이터 유형의 수} \\ &+ b * \text{출력데이터 유형의 수} \\ &+ c * \text{참조되는 실체유형 수} \\ &+ d * \text{알고리즘 수} \end{aligned}$$

$$\begin{aligned} \text{기능수(FC)} &= 105.906 - 0.00458 * \text{입력데이터 유형수} \\ &+ 0.389 * \text{출력데이터 유형수} \\ &+ 0.122 * \text{실체유형수} - 0.258 \\ &* \text{알고리즘수} \quad (R^2 = 0.854, \text{Adj-R}^2 \\ &= 0.834, \text{Prob} > F: 0.000) \end{aligned}$$

이 모형의 설명력은 매우 높아, 가중치를 적용하지 않는 것이 타당함을 보여주고 있다. 즉, 시스템구조 및 웹인터페이스 가중치는 기술적 복잡도 요소로 고려될 수 있는 성질의 작은 요인일 수 있다. 따라서 가중치를 적용하지 않은 모형에 대하여 추가 분석을 수행하였다. 가중치를 고려하지 않고 소요노력 추정 계수를 현재의 0.15에서 0.1로 바꾼 모형에 대한 분석결과는 다음과 같다.

$$\begin{aligned} \text{기능수(FC)} &= 158.859 - 0.00688 * \text{입력데이터 유형수} \\ &+ 0.583 * \text{출력데이터 유형수} \\ &+ 0.183 * \text{실체유형수} - 0.387 \\ &* \text{알고리즘수} \quad (R^2 = 0.854, \text{Adj-R}^2 \\ &= 0.834, \text{Prob} > F: 0.000) \end{aligned}$$

또한 가중치를 고려하지 않고 계수를 0.05로 바꾼 모형에 대한 분석결과는 다음과 같다.

$$\begin{aligned} \text{기능수(FC)} &= 317.718 - 0.0138 * \text{입력데이터 유형수} \\ &+ 1.166 * \text{출력데이터 유형수} + 0.366 \\ &* \text{실체유형수} + 0.774 * \text{알고리즘수} \quad (R^2 \\ &= 0.854, \text{Adj-R}^2 = 0.834, \text{Prob} > F: 0.000) \end{aligned}$$

4.2 모형 II에 대한 분석

4.2.1 기능점수와 소요노력간의 관계분석

실 소요노력(MM)를 종속변수로 하고 입력 모듈수, 출력 모듈수(및 장표수), 테이블 수, 배치프로그램수로 계산된 기능점수를 독립변수로 하여 산출된 모형 II에 대한 기능점수 모형의 소요노력 계수 추정 결과는 다음과 같다.

$$\text{소요노력 (표준 MM)} = -1.249 + 0.113 * \text{FP}$$

$$(R^2 = 0.893, \text{Adj-}R^2 = 0.889, \text{Prob}>F:0.000)$$

이 모형의 경우, 모형의 설명력도 높고, 계수의 크기 도 기준치와 큰차이가 없는 것으로 나타나 매우 바람 직한 모형으로 판단된다.

4.2.2 기능수 산출에 관한 분석

실데이터를 이용하여 기능 요인 각각에 대한 적정 계수를 추정하는 계산식을 기능수에 대해 정리한 식과 회귀분석 수행결과는 다음과 같다.

$$\begin{aligned} \text{기능수} &= (\text{소요노력} - 10) / (0.15 * \text{가중치} * \text{기술적 복잡도}) \\ &= a * \text{입력모듈 수} + b * \text{출력모듈 수} \\ &\quad + c * \text{테이블 수} + d * \text{배치프로그램 수} \end{aligned}$$

$$\begin{aligned} \text{기능수(FC)} &= -44.5 + 1.619 * \text{입력모듈 수} \\ &\quad + 0.515 * \text{출력모듈 수} \\ &\quad + 0.101 * \text{테이블 수} \\ &\quad + 0.375 * \text{배치프로그램 수} \\ (R^2 &= 0.912, \text{Adj-}R^2 \\ &= 0.899, \text{Prob}>F:0.000) \end{aligned}$$

이 모형의 설명력은 매우 높고, 계수의 부호도 모두 양수로 나와 바람직한 모형으로 판단된다. 가중치를 고려하지 않은 모형에 대해 회귀분석을 수행한 결과는 다음과 같다.

$$\begin{aligned} \text{기능수(FC)} &= -70.033 + 2.342 * \text{입력모듈 수} \\ &\quad + 0.593 * \text{출력모듈 수} \\ &\quad + 0.196 * \text{테이블 수} - 1.596 \\ &\quad * \text{배치프로그램 수} \\ (R^2 &= 0.910, \text{Adj-}R^2 \\ &= 0.896, \text{Prob}>F:0.000) \end{aligned}$$

이 모형도 설명력이 매우 높아 통계적으로 의미있는 모형으로 판단된다. 가중치를 고려하지 않고 계수를 0.1로 바꾼 모형에 대한 분석결과는 다음과 같다.

$$\begin{aligned} \text{기능수(FC)} &= -105.05 + 3.513 * \text{입력모듈 수} \\ &\quad + 0.89 * \text{출력모듈 수} \\ &\quad + 0.295 * \text{테이블 수} - 2.394 \\ &\quad * \text{배치프로그램 수} \\ (R^2 &= 0.910, \text{Adj-}R^2 \end{aligned}$$

$$= 0.896, \text{Prob}>F:0.000)$$

마지막으로 가중치를 고려하지 않고 계수를 0.05로 바꾼 모형에 대한 분석결과는 다음과 같다.

$$\begin{aligned} \text{기능수(FC)} &= -210.099 + 7.027 * \text{입력모듈 수} \\ &\quad + 1.78 * \text{출력모듈 수} \\ &\quad + 0.589 * \text{테이블 수} - 4.788 \\ &\quad * \text{배치프로그램 수} \\ (R^2 &= 0.910, \text{Adj-}R^2 \\ &= 0.896, \text{Prob}>F:0.000) \end{aligned}$$

4.3 모형 III에 대한 분석

4.3.1 기능점수와 소요노력간의 관계분석

실 소요노력(MM)을 종속변수로 하고 입력 모듈수, 출력 모듈수(및 장표수), 테이블 수, 알고리즘수로 계산된 기능점수를 독립변수로 하여 산출된 모형 III에 대한 기능점수 모형의 계수 추정 결과는 다음과 같다.

$$\begin{aligned} \text{소요노력 (표준 MM)} &= -1.316 + 0.1 * \text{FP} \\ (R^2 &= 0.844, \text{Adj-}R^2 \\ &= 0.839, \text{Prob}>F:0.000) \end{aligned}$$

즉, 배치프로그램수를 알고리즘수로 대치한 결과는 모형의 설명력이 약간 떨어지는 것으로 나타났으나, 실용적으로 사용하기에 매우 적합한 모형으로 판단된다.

4.3.2 기능수 산출에 관한 분석

실데이터를 이용하여 기능 요인 각각에 대한 적정 계수를 추정하는 식을 기능수에 대해 정리한 식과 분석결과는 다음과 같다.

$$\begin{aligned} \text{기능수} &= (\text{소요노력} - 10) / (0.15 * \text{가중치} \\ &\quad * \text{기술적 복잡도}) \\ &= a * \text{입력모듈 수} + b * \text{출력모듈 수} \\ &\quad + c * \text{테이블 수} + d * \text{알고리즘 수} \end{aligned}$$

$$\begin{aligned} \text{기능수(FC)} &= -43.845 + 1.664 * \text{입력모듈 수} \\ &\quad + 0.576 * \text{출력모듈 수} \\ &\quad + 0.253 * \text{테이블 수} - 0.306 * \text{알고리즘 수} \\ (R^2 &= 0.913, \text{Adj-}R^2 = 0.900, \\ &\quad \text{Prob}>F:0.000) \end{aligned}$$

위 모형의 설명력은 매우 높으나, 알고리즘수에 대한 계수가 음수로 나타나, 직접 실무에서 사용하기에

다 다소 어려움이 예상된다. 가중치를 고려하지 않은 모형에 대한 회귀분석 결과는 다음과 같다.

$$\begin{aligned} \text{기능수(FC)} &= -57.682 + 1.742 \times \text{입력모듈 수} \\ &+ 0.631 \times \text{출력모듈 수} \\ &+ 0.058 \times \text{테이블 수} + 0.177 \\ &\times \text{알고리즘 수} \\ (R^2 &= 0.865, \text{Adj-}R^2 \\ &= 0.846, \text{Prob}>F:0.000) \end{aligned}$$

이 모형은 설명력도 높고, 기능요인에 대한 가중치 계수도 모두 양수로 나와 비교적 양호한 모형으로 판단된다. 그러나 테이블수에 대한 계수값이 매우 작은 등 기능요인간의 균형성 측면에서 다소 부족함이 있다. 위 모형의 계수를 0.1로 바꾼 모형에 대한 분석결과는 다음과 같다.

$$\begin{aligned} \text{기능수(FC)} &= -86.524 + 2.613 \times \text{입력모듈 수} \\ &+ 0.946 \times \text{출력모듈 수} \\ &+ 0.088 \times \text{테이블 수} + 0.266 \\ &\times \text{알고리즘 수} \\ (R^2 &= 0.865, \text{Adj-}R^2 \\ &= 0.846, \text{Prob}>F:0.000) \end{aligned}$$

모형의 계수값이 다소 증가되었다. 따라서, 이 모형의 계수를 0.05로 바꾸고 다시 분석을 수행한 결과는 다음과 같다.

$$\begin{aligned} \text{기능수(FC)} &= -173.047 + 5.226 \times \text{입력모듈 수} \\ &+ 1.893 \times \text{출력모듈 수} \\ &+ 0.175 \times \text{테이블 수} \\ &+ 0.531 \times \text{알고리즘 수} \\ (R^2 &= 0.865, \text{Adj-}R^2 \\ &= 0.846, \text{Prob}>F:0.000) \end{aligned}$$

이상의 분석에서 보는 바와 같이 모형 III은 모형의 설명력도 높고, 기능요소를 측정하기도 용이하여 대체로 실용성이 높은 것으로 나타났다.

4.4 적정계수 시뮬레이션 분석

소프트웨어 사업대가기준 개정을 위해서는 보다 단순화된 가중치 값이 필요하다. 즉 사용자의 이해도를 높이기 위해 소수점 이하 1자리 정도에서 가중치 값이 설정되는 것이 바람직하다. 따라서 기능수 증대 요인

에 대한 단순화된 가중치를 도출하기 위하여 가중치 변형 시뮬레이션을 실시하였다. 여기에서의 시뮬레이션은 기능요인간의 가중치 수준을 찾아내는데 목적이 있으므로, 아래와 같이 기술적복잡도와 기능수 가중치를 배제하고 시뮬레이션을 수행하였다. 시뮬레이션은 전문가직 판단에 의하여 가중치의 후보를 찾아내고, 이들 후보 중에서 가장 모형 적합도가 높은 계수 집합을 찾아내는 방식으로 진행하였다.

$$\begin{aligned} \text{소요노력} &= 10 + e \times (a \times \text{입력모듈수} + b \times \text{출력모듈수} \\ &+ c \times \text{테이블수} + d \times \text{알고리즘 수}) \quad \dots \text{①} \\ \text{소요노력} &= 10 + e \times (a \times \text{입력모듈수} + b \times \text{출력모듈수} \\ &+ c \times \text{테이블수} + d \times \text{매치프로그램수}) \quad \dots \text{②} \end{aligned}$$

① 번 모형에 대해 계수를 시뮬레이션한 결과는 다음과 같다.

가 중 치					평균오차 1	평균오차 2
a	b	c	d	e	1	2
1.5	1.1	0.5	0.7	0.12	123.9 %	49.4 %
1.3	1.1	0.5	0.7	0.10	106.9 %	43.4 %
1.5	1.0	0.5	0.7	0.12	125.8 %	49.7 %
1.3	1.0	0.5	0.7	0.10	108.2 %	43.2 %

비교 1 : 평균오차1은 전체데이터를 사용한 경우의 오차이고, 평균오차2는 크게 차이나는 4개의 자료를 제외한 경우의 평균오차임.

비교 2 : 평균오차식은 다음과 같음.

$$\begin{aligned} \text{평균오차} &= \text{평균} \left(\frac{\text{절대치}((\text{계산노력} - \text{실제노력})}{\text{실제노력}} \times 100) \right) \end{aligned}$$

② 번 모형에 대해 계수를 시뮬레이션한 결과는 다음과 같다.

가 중 치					평균 오차1	평균 오차2
a	b	c	d	e	오차1	오차2
1.5	1.1	0.5	0.7	0.10	111.0 %	44.5 %
1.3	1.1	0.5	0.7	0.07	91.8 %	45.2 %
1.5	1.0	0.5	0.7	0.10	112.6 %	44.6 %
1.3	1.0	0.5	0.7	0.07	92.5 %	44.7 %

이 결과를 통해 볼 때 입력모듈수, 출력모듈수, 테이블수, 알고리즘수 에 대해 각각 1.3, 1.1, 0.5, 0.7 의 가중치를 부여하는 것이 오차가 낮아지는 것으로 판단된다.

5. 결 론

본 연구는 다음과 같은 측면에서 소프트웨어 규모 및

비용산정 모형의 정확성과 실용성 증대에 기여하였다.

우선, 기존의 국내 연구를 보완하고, 전문가의 자문을 통하여 실용성이 높은 후보 모형을 도출하였다. 현재와 같은 프로젝트 환경에서는 실제 데이터에 의한 통계적으로 완벽한 검증을 기대하기 어려우므로, 현업 프로젝트 전문가의 경험에 의한 검증이 매우 중요한 역할을 한다. 따라서 본 연구는 이러한 과정을 충실하게 거쳐 후보 모형을 도출하였으므로, 도출된 모형은 매우 타당성이 높은 모형이라 할 수 있다. 특히 기술적 복잡도 판단 기준표는 기능점수 모형에서 제시되지 않은 독자적인 기준으로서 한국형 기능점수 모형의 대표적인 특징이라 할 수 있다.

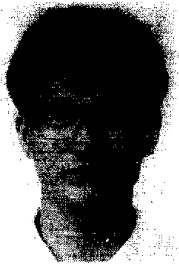
또한 본 연구는 39개의 프로젝트 실제 데이터를 이용하여 후보 모형의 타당성을 입증하였다. 즉 제안된 후보 모형이 전문가에 의한 경험적인 타당성과 함께 일정 수준 이상의 통계적 유의성을 가지고 있음을 보여준데 본 연구의 의의가 있다.

본 연구의 결과는 산업계와 정부 등의 관련기관 실무 회의를 거쳐 소프트웨어 사업의 대가기준으로 활용될 수 있을 것이다. 향후 연구과제로는 프로젝트 환경 변수의 도입에 의한 기능점수 모형의 세분화와 통계적 데이터 검증의 확대 등을 들 수 있다. 즉 개발 프로젝트의 유형, 유지보수 프로젝트의 성격 등을 고려한 개별적인 기능점수 모형을 확대하고, 프로젝트 수행중에 실제 데이터를 수집할 수 있는 체계의 마련이 필요하다.

참 고 문 헌

[1] 김현수, 소프트웨어 비용산정 대가기준 연구, 한국 전자통신연구원/한국소프트웨어산업협회, Dec., 1998.
 [2] 김현수, 기능점수를 이용한 소프트웨어 규모 및 비용산정 방안에 관한 연구, 경영과학, 제14권 1호, pp.131-149, May., 1997.
 [3] 김현수의 1인, 소프트웨어 개발 및 구축기술 대가 기준에 관한 연구, 한국소프트웨어산업협회, June, 1996.
 [4] 西山茂, 정보처리학회 역, 소프트웨어 규모전적 기술의 동향, 정보처리, 제1권 제3호 (1994), pp.95~104
 [5] 田中 淳, "클라이언트/서버의 어려운 문제 - 개발 공수 전적에 도전-", 일경컴퓨터, pp.114-125, 20

Mar., 1995.
 [6] 유성열, 백인섭, 김하진, 유지보수관리 체계의 정형화 및 비용 예측 모델에 관한 연구, 한국정보처리 논문지, 제3권 제4호, pp.846-854 July, 1996.
 [7] Albrecht, Allan J., "Measuring Application Development Productivity," in Proc. IBM Application Develop. Symp., Monterey, CA, GUIDE Int. and SHARE, Inc., IBM Corp., pp.83-92, 14~17, Oct., 1979.
 [8] Albrecht, Allan J., and John Gaffney, "Software Function, Source Lines of Code, and Development Effort Prediction : A Software Science Validation", IEEE Transactions on Software Engineering, Vol.9, No.6(1983), pp.639-648, 1983.
 [9] Bassman, M.J., McGarry, F., and Pajerski, R., Software Measurement Guidebook, NASA/GSFC/SEL-94-102, June, 1996
 [10] Florac, W.A., R.E. Park, and A.D. Carleton, Practical Measurement : Measuring for Process Management and Improvement, CMU/SEI-97-HB-003, April, 1997.
 [11] IFPUG, Function Point Counting Practice anual, Release 4.0, IFPUG, Atlanta, Georgia, 994.
 [12] Jones, C., Applied Software Measurement, McGraw Hill, 2nd Edition, 1996.
 [13] Jones, C., A Short History of Function Points, Software Productivity Research Inc., Burlington, MA, 1986.
 [14] Kemerer, Chris F., and Benjamin S. Porter, "Improving the Reliability of Function Point Measurement : An Empirical Study," IEEE ransactions on Software Engineering, Vol.18, No.11(1992), pp.1011-1024
 [15] Low, Graham C. and Ross 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
 [16] Symons, Charles R., "Function Point Analysis: Difficulties and Improvements," IEEE Transactions on Software Engineering, Vol.14, No.1(1988), pp.2-11



김 현 수

e-mail : hskim@kmu.kookmin.ac.kr

1982년 서울대학교 공대 원자핵
공학과 졸업(학사)

1985년 한국과학기술원 졸업(석사)

1992년 University of Florida 경
영정보학과 졸업(박사)

1985년~1988년 (주) 데이콤 주임연구원

1994년~1999년 국민대학교 정보관리학부 조교수

관심분야 : 정보시스템 진단과 감리, 소프트웨어공학,
데이터마이닝 등