정보전략계획 단계에서의 정보시스템 규모 예측 : 기능점수모형을 중심으로

Software Size Measurement from Information Strategy Planning With the Function Point Method

배준수(Joonsoo Bae)*, 정재윤(Jae-Yoon Jung)**

초 록

기업과 정부는 경쟁력 제고와 효율성 확보를 위해 지속적인 정보화사업을 추진하고 있다. 정보화사업의 규모가 커짐에 따라 정보시스템 개발 규모를 측정하고 비용을 예측하는 일이 중요해지고 있다. 본 연구에서는 정보전략계획 단계에서 정보시스템 개발 규모를 합리적으로 측정하기 위하여 기능점수를 활용하는 과정을 설명한다. 정보전략계획의 산출물로부터 프로젝트 종류를 식별하고, 기능점수 산출 범위와 경계를 정의한 후, 데이터 기능과 트랜잭션 기능으로 나누어 기능점수를 계산한다. 특히, 정보전략계획 단계에서 불명확한 기능점수의 복잡도 계산을 하기 위하여 간이기능점수를 활용한다. 합리적인 정보시스템 규모 예측은 투입 공수와 사업 대가를 결정함으로써 성공적인 정보화 사업을 수행하는 밑거름이 될 것이다.

ABSTRACT

Many companies and governments perform information technology (IT) projections repeatedly for higher competitiveness and efficiency. Moreover, as the sizes of IT projects increase, the manpower and cost estimation of the projects are getting more important. In this research, we present a size estimation method of information system development projects on the basis of the function point method at the stage of information strategy planning(ISP), and illustrate the size calculation of an example IT project. According to the method, we first identify the types of projects, the scope and boundaries of size estimation, then count data and transaction functions from the artifacts of the ISP project. The unadjusted function points are adjusted to function point by the table of the Early Function Point. The way of calculating the sizes of IT projects will support successful IT projects by estimating reasonable manpower and cost for the projects.

키워드 : 기능점수법, 소프트웨어 규모 측정, 정보전략계획
Function Point, Software Size Measurement, Information Strategy Planning

* 전북대학교 산업정보시스템공학과
** 경희대학교 산업경영공학과, 산학협력기술연구원
2009년 06월 08일 접수, 2009년 07월 30일 심사완료 후 2009년 08월 07일 게재확정.

1. 서 론

정보시스템 구축과 운영 사업이 증가하고 그 규모가 성장함에 따라, 기업과 공공기관 등의 정보화 기획 부서에서는 정보시스템 도입 및 운영에 관한 예산을 산정하고 집행하는 데 중요성이 높아지고 있다. 정보화사업의 비용과 규모를 정확히 예측하기 위해서는 정보화 사업의 규모를 가능한 한 정확히 파악하고 합리적인 예산을 투입해야 한다. 정보화사업의 인력과 예산 투입은 정보시스템의 납기, 품질, 생산성 등에 결정적인 영향을 미치므로 [9, 14], 수발주 기관 양자가 동의할 수 있는 정보시스템 규모 예측과 예산 산출 방식이 필요하다.

본 연구는 정보전략계획(Information Strategy Planning : ISP)의 산출물로부터 정보화사업의 특성과 범위를 규정하고 기능점수를 측정하여 개발 소프트웨어 규모를 예측하는 것을 목표로 한다. 정보전략계획은 경영상에서 정보기술의 역할과 위상을 파악하여 투자 방향을 결정하는 단계이다. 그러나 정보전략계획은 적용 기간 및 대상에 따라서 다양한 목적과 형태로 진행될 수 있으며, 실제 정보전략계획을 수립할 때에는 수행 기간, 투입 인원, 투자 비용 및 수행 목적 등에 따라서 전략정보계획만 수립하고 구현계획을 수립하지 않을 수도 있다. 본 연구는 정보전략계획에서 구현계획까지 수행된 경우, 그 구현계획 산출물의 기능 및 데이터 목록을 바탕으로 정보시스템 규모를 산정하는 방법을 가정한다.

정보시스템 개발 및 운영 사업의 소프트웨어 규모를 측정하는 것은 개발 및 운영에 투입되는 공수(man-month)를 예측하고 사업 대가를 측정하기 위하여 주로 활용된다. 본 연구에서 제시하는 정보전략계획 단계에서 정보시스템 개발 규모를 예측하는 과정은 기능점수를 통하여 보다 합리적인 정보화 사업 대가를 산정하는 데 도움이 될 것이다.

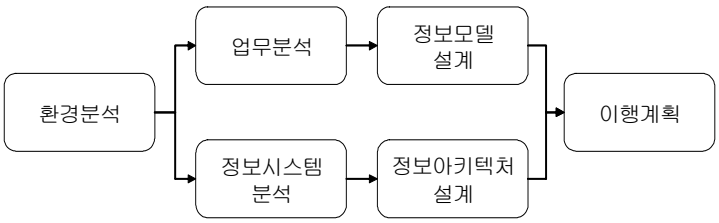
2. 관련 연구

2.1 정보전략계획

정보화전략계획(ISP)의 시작은 IBM에서 정보시스템이 기업전략을 효과적으로 지원하도록 하기 위한 BSP(Business System Planning)라는 방법론을 개발하면서부터라고 할 수 있다. ISP는 조직내의 전략적 정보요구를 식별하여 업무활동과 이에 대한 자료영역을 기술하고, 현행 정보지원 정도를 평가하고, 정보시스템 개발을 위한 통합된 프레임워크를 제공하며, 이에 대한 구현을 위한 정보기술(IT)을 활용한 통합 정보시스템 계획을 작성하는 체계적인 접근활동이다.

일관된 ISP 품질을 유지하고 효과적인 커뮤니케이션이 가능하게 하며 경험의 축적에 따른 생산성 향상을 위하여 다양한 ISP 수립 방법론이 개발되어 활용되고 있다.

ISP 수립방법론들은 세부적인 수행절차의 차이는 있으나 전반적인 ISP 수립절차는 <그림 1>과 같이 크게 6단계로 구성된다. 먼저, *환경분석*을 통해 조직의 장래상, 경영전략, 핵심성공요소, 정보화 전략 등을 파악한다. 환경분석을 통해 확인된 사업방향은 업무 및 정보시스템 분석의 기준이 되며 개선모델의 방향성을 제시하는 가이드라인의 역할을 한다.



〈그림 1〉 ISP 수립 절차

다음으로, *업무분석*을 통하여 주요 현안 및 개선기회를 파악하여 현안을 해결하고 경영목표를 달성할 수 있도록 도출된 과제별로 개선모델을 수립한다. 한편, 현재 운영 중인 *정보시스템 분석*을 수행하는데, 기능적, 기술적 품질을 진단하여 개선과제를 파악한 후, *정보아키텍처 설계* 모델을 수립한다. 최종적으로 개선모델을 실제로 구현하기 위한 이행과제들을 정의하고, 이행과제별 위험요소, 투자대비효과 등을 파악하여 우선순위를 설정한 다음 성공적인 이행을 위한 이행조직 구성, 변화관리 방안 등을 수립하여 종합적인 *이행계획*을 완성한다.

2.2 소프트웨어 규모 측정

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.

일반적으로 소프트웨어 개발비용을 예측하기 위해서는 먼저 개발하고자 하는 소프트웨어의 규모를 측정하고, 그 규모로부터 비용을 추정하게 된다. 소프트웨어의 규모를 측정하기 위하여 흔히 사용되는 방법으로 프로그램 라인수(line of code : LOC) 방식과 기능점수법(function point : FP)이 있다. 우리 나라에서는 소프트웨어 개발 및 유지보수 사업의 규모 산정에 주로 LOC 방식을 사용해 왔으나, LOC 방식은 개발자 중심적이고 사전에 예측이 어려우며 개발환경에 영향을 많이 받기 때문에 기능점수법을 도입하고자 하는 시도가 많으며[6], 본 연구에서도 소프트웨어 개발 비용을 예측하는 데 기능점수법을 사용하고자 한다.

소프트웨어 비용산정 방법은 LOC와 기능점수를 이용한 모형으로 나눌 수 있는데, 초기에 Boehm, Putnam and Myers 등이 라인수를 이용한 모형을 제시하였으며[4, 12], 이후에 기능점수방식을 반영한 수정된 모형이 많이 개발되었으며, COCOMO Ⅱ, ISBSG, NESMA 등의 표준모형이 여기에 해당된다[5, 8, 11]. 국내에서는 라인수 방식에 비해 기능점수 모형이 활발히 사용되지 못하고 있는데, 이는 제도적 문제와 더불어 국내 환경을 적용한 데이터 축적 및 분석이 부족하기 때문이다.

LOC 기반 모형을 살펴보면, Putnam이 고안한 Software Lifecycle Model(SLIM)은 매개변수를 이용하여 아래와 같이 규모를 산정한다[12].

〈표 1〉 데이터기능의 복잡도 행렬

RET \ DET	1~19	20~50	51개 이상
1	낮음	낮음	보통
2~5	낮음	보통	높음
6개 이상	보통	높음	높음

〈표 2〉 보정전 데이터 기능점수

복잡도 \ 데이터	ILF	EIF
낮음	7	5
보통	10	7
높음	15	10

$$L = C_k \times K^{1/3} \times T_d^{4/3}, \quad K = D_0 \times T_d^3$$

위에서 L 산정식은 Putnam 곡선이라고 부르는데, L은 스텝 수(LOC), K는 소요공수, T_d는 개발기간, C_k는 환경변수(2000 : 프로젝트 난이도가 높음, 8000 : 양호, 11,000 : 쉬움)를 의미한다. 소요공수 K는 아래와 같이 조정된다. D_0는 인력구성 매개변수로 신규 소프트웨어와 많은 인터페이스인 경우 8의 값을, 기존 소프트웨어 변경인 경우 27의 값을 사용한다.

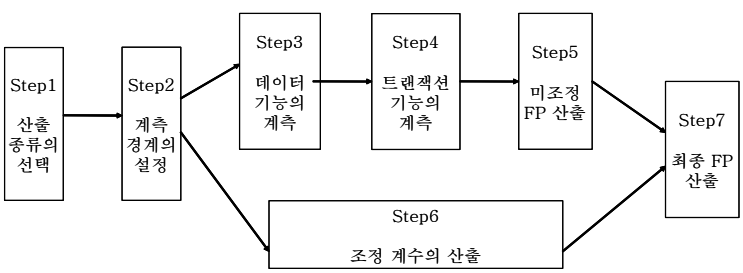
이외에도 한편, COCOMO 모형은 1981년 Boehm에 의하여 연구된 것으로 미국방성 및 NASA에서 비용 산정을 위한 기본 모형으로 사용하고 있으며, 현재 COCOMO Ⅱ로 발전하여 15개 보정요소를 가지고 있다[5]. 기능점수 모형은 제 3장에서 자세히 설명한다.

3. 기능점수(Function Point)

3.1 기능점수 계산

Albrecht가 기능점수모형을 제안한 이후 변형된 여러 가지 기능점수모형 중에서 가장 널리 사용되고 있는 국제기능점수사용자그룹(IFPUG)에서 제시한 기능점수모형을 간략히 알아본다[7]. 기능점수모형은 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 기능 유형을 측정한다. 데이터기능(data function)은 내부 및 외부 자료 요구사항을 만족시키기 위해 사용자에게 제공되는 기능을 말하는데, 내부논리파일(Internal Logical



〈그림 2〉 기능점수 계측 흐름도

File : ILF)과 외부연계파일(External Interface File : EIF)로 나뉜다. ILF와 EIF의 복잡도는 레코드요소유형(Record Element Type : RET)과 데이터요소유형(Data Element Type : DET)의 개수에 따라 <표 1>과 같이 복잡도를 결정하고, 그 복잡도에 따라 데이터 기능의 보정전 기능점수(UnAdjusted Function Point : UAF)를 <표 2>와 같이 결정된다.

트랜잭션기능(transactional function)은 데이터를 처리하기 위해 사용자에게 제공되는 기능을 말하는데, 트랜잭션기능에는 외부입력(External Input : EI), 외부출력(External Output : EO), 외부조회(External inQuiry : EQ)로 나뉜다. EI, EO, EQ의 복잡도는 참조파일유형(File Type Referenced : FTR)과 데이터요소유형(Data Element Type : DET)의 개수에 의해 <표 3>, <표 4>와 같이 결정되고, 복잡도에 따라 보정전 기능점수(UAF)가 <표 5>와 같이 결정된다.

〈표 3〉 EI의 복잡도

FTR \ DET	1~4	5~15	16개 이상
0~1	낮음	낮음	보통
2	낮음	보통	높음
3개 이상	보통	높음	높음

〈표 4〉 EO/EQ의 복잡도

FTR \ DET	1~5	6~19	20개이상
0~1	낮음	낮음	보통
2~3	낮음	보통	높음
4개 이상	보통	높음	높음

〈표 5〉 보정전 트랜잭션 기능점수

복잡도 \ 트랜잭션	EI/EQ	EO
낮음	3	4
보통	4	5
높음	6	7

보정전 기능점수(UAF)가 계산되면, 시스템의 특성을 반영하도록 보정요소와 보정계수를 계산하여 기능점수를 보정하여 최종적인 기능점수를 산정한다. 보정요소에는 데이터 통신(data communications), 분산 데이터 처리(distributed data processing), 성능(performance), 사용환경(heavily used configuration), 처리율(transaction rate) 등 14가지가 있다.

3.2 간이기능점수(Early Function Point)

기능점수 분석은 정보화 예산의 적정성을 판단하는 합리적인 기준으로 사용될 수 있다. 특히, 정보전략계획 단계에서는 프로젝트의 정의가 명확하게 이루어지지 않으므로 정확한 비용을 산정하는 것은 매우 힘들며, 다음과 같은 어려움이 있다.

- 기초요구분석 없이 정보화 계획 단계에 전체 프로세스를 조사하기에 현실적으로 한계가 있음
- 예산신청 단계에서 단위 프로세스의 유형(EO, EQ)을 구분하기가 쉽지 않음
- 단위 프로세스의 복잡도 산정에 필요한 DET/FTR를 모두 식별하기 어려움

이러한 이유로 정보전략단계에서는 데이터 기능과 트랜잭션 기능에 기반한 정확한 복잡도 산출 방식 대신에 간소화된 간이기능점수 방식을 사용한다.

- RET(데이터기능) 또는 DET/FTR(트랜잭션기능)를 식별하지 않고, 평균복잡도를 사용함
- 계산 로직의 포함여부, 유도된 데이터의 포함여부 등 세부 로직까지 정의하지 않아도 됨
- 입력화면 내에 있는 포함되어 있는 Pick List 등을 식별하지 않아도 됨

이와 같은 간이기능점수 모형을 위하여 한국정보사회진흥원에서 사용하는 평균 복잡도 가중치 기준은 <표 6>과 같다.

〈표 6〉 기능유형별 평균 복잡도 가중치

구분	ILF	EIF	EI	EO	EQ
가중치	8.1	5.3	3.8	4.9	3.7

4. 정보전략계획 단계의 기능점수 적용

본 연구에서는 앤더슨컨설팅에서 개발한 정보전략계획 방법론인 Method/1을 확장하여 기능점수를 산정하는 방식을 제시하였다. 아래 <그림 3>과 같이 200G 다음 단계로 210G "비용산정을 위한 기능점수 계측" 단계를 추가하였으며, 210G의 구체적인 단계는 <그림 4>와 같이 수행된다.

4.1 기능점수 산출 종류 선택

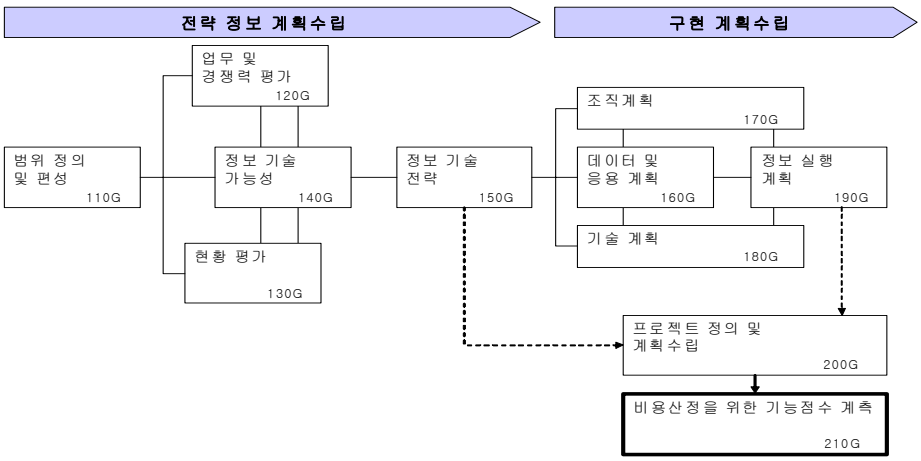
기능점수를 산출하기 위해서는 먼저 대상 프로젝트의 종류를 확인해야 한다. 기능점수에서 고려하는 프로젝트의 종류는 신규개발, 기능확장, 어플리케이션의 세 가지로 나눈다. 이 세 가지는 서로 독립적인 것이 아니라 <그림 5>와 같이 서로 연관되어 있다.

만약 프로젝트 A가 신규개발 프로젝트이고, 프로젝트 B가 기능 확장 프로젝트라고 하면, 프로젝트 초기에는 정확한 기능점수 산정이 힘들기 때문에 추정 기능점수를 산출한다. 그러나, 프로젝트가 진행되면서 산출 범위가 명확해지고, 기능이 개발되면서 점점 정확한 기능점수를 산정할 수 있다. 결국 프로젝트가 완료되면 최종 기능점수를 산출할 수 있다. 만약 신규개발 프로젝트인 프로젝트 A가 먼저 완료되면 어플리케이션 기능점수가 처음으로 계산된다. 그러다 확장 프로젝트인 B가 완료되면 어플리케이션 기능점수가 변경된다.

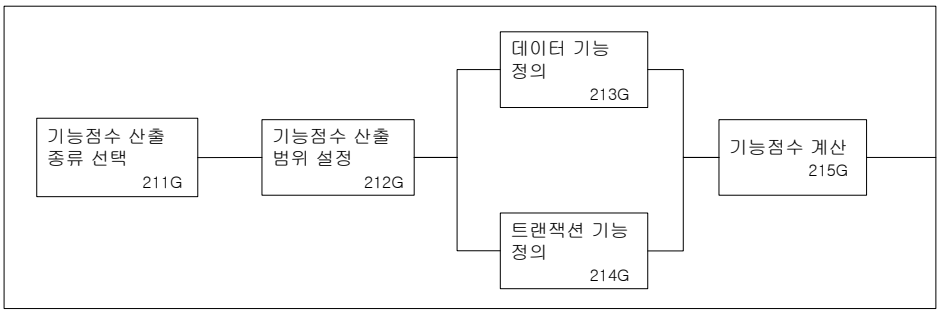
이와 같이 프로젝트의 종류에 따라서 그 기능점수를 계산할 때 방법이 달라지기 때문에 가장 먼저 대상 프로젝트를 종류별로 구분한다. 예를 들어 <표 7>과 같이 단일한 정보전략계획의 대상으로 신규개발과 기능확장 프로젝트가 동시에 포함되어 있을 수 있다.

4.2 기능점수 산출 범위 설정

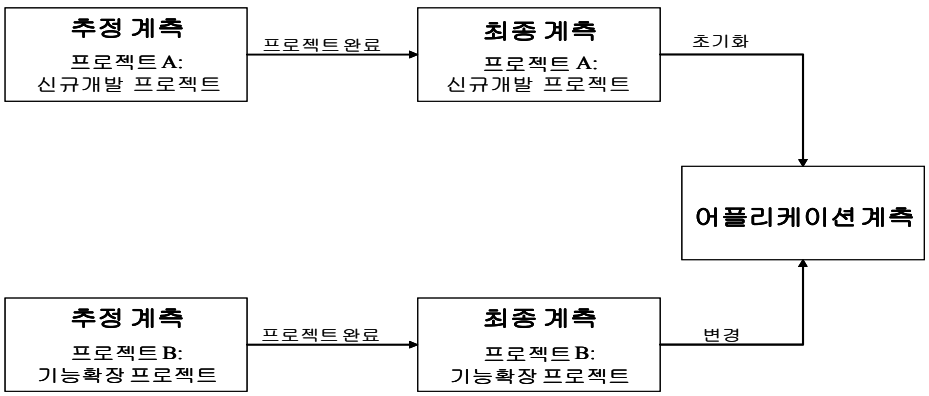
두 번째로는 기능점수 산출 범위(counting scope)와 어플리케이션 경계(application boundary)를 결정해야 한다. 기능점수 산출 범위란 특정한 기능점수 산정에 포함될 기능을 의미하는데, 앞서 결정한 프로젝트 종류별로



〈그림 3〉 ISP 방법론 Method/1 프로세스



〈그림 4〉 기능점수 계측 프로세스 210G



〈그림 5〉 세 가지 프로젝트 종류(신규개발, 기능확장 및 어플리케이션 프로젝트)의 관계

〈표 7〉 기능점수 산출을 위한 프로젝트 종류

프로젝트 명	프로젝트 종류
프로젝트 1	신규개발
프로젝트 2	기능확장
프로젝트 3	신규개발
프로젝트 4	신규개발
프로젝트 5	신규개발 후 기능확장

산출 범위의 의미는 다음과 같다.

- 신규개발 기능점수의 산출 범위 : 프로젝트 활동에 의해 영향을 받는 모든 기능(신규 구축 혹은 맞춤화)을 포함함
- 기능확장 기능점수의 산출 범위 : 추가, 변경, 삭제되는 모든 기능을 포함함
- 어플리케이션 기능점수의 산출 범위 : 산출 목적에따라서 특정 사용자에 의해서 사용되는 기능만을 의미할 수 있고, 제공되는 모든 기능을 의미할 수도 있음

어플리케이션 경계란 기능점수를 산출하고자 하는 대상 소프트웨어와 사용자간의 구분을 의미하며, 어플리케이션 경계를 설정할 때는 다음과 같은 규칙을 따라야 한다.

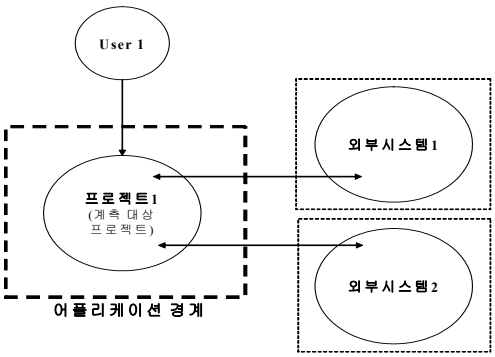
- 경계는 사용자의 관점에서 결정되어야 하며, 사용자가 이해하고 기술하는 것이 중요함
- 관련 어플리케이션 간의 경계는 기술적 사항이 아니라, 사용자에게 제공되는 기능에 바탕을 두어야 함
- 수정되는 어플리케이션의 초기에 설정된 경계는 산출 범위에 의해 영향을 받지 않음

정보전략계획을 수행하게 되면 어떠한 프로젝트가 필요하고 어떤 순서로 진행되어야 하는지를 결정하게 된다. 예를 들어, <표 8>과 같이 총 5개의 프로젝트가 도출되었는데, 그 중 프로젝트 1과 프로젝트 2는 1차년도, 프로젝트 3과 프로젝트4는 2차년도, 프로젝트 5는 3차년도에 시행하기로 결정되었다 하자. 여기에서 1차년도 비용 산정을 위해서는, 프로젝트 1과 프로젝트 2를 하나의 산출 범위(Scope1)로 정하고, 산출 범위별로 기능점수가 산정하게 된다.

또한, 산출 경계를 설정해야 하는데, 위에서 정해진 각 프로젝트 별로 사용자 관점에서 시스템을 독립된 구조적 단위로 분리해내는 단계이다. <그림 6>은 단독 프로젝트 1 어플리케이션과 외부 어플리케이션(외부 시스템 1과 외부 시스템 2)의 경계를 보여준다. 또한 사용자(User 1)와 프로젝트 1 어플리케이션과의 경계도 보여준다.

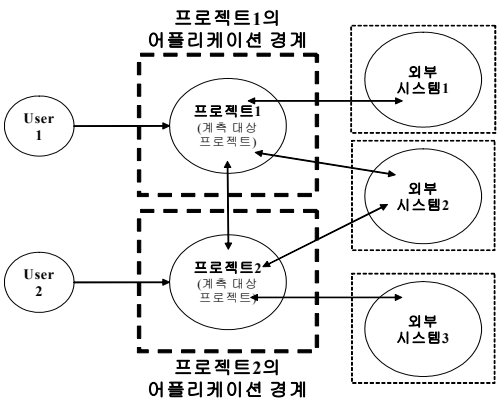
〈표 8〉 대상 프로젝트의 기능점수 산출 범위 설정

프로젝트 명	프로젝트 종류	산출 범위
프로젝트 1	신규개발	Scope1
프로젝트 2	기능확장	
프로젝트 3	신규개발	Scope2
프로젝트 4	신규개발	
프로젝트 5	신규개발 후 기능확장	Scope3

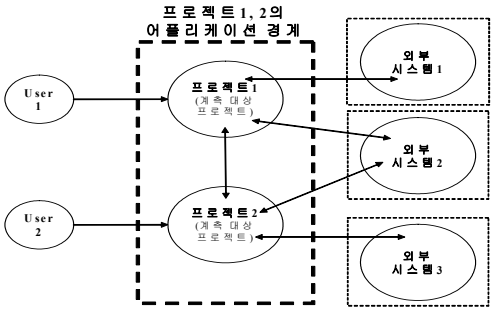


〈그림 6〉 단독 프로젝트의 어플리케이션 경계 결정

둘 이상의 프로젝트가 서로 연결되어 있는 경우 어플리케이션 경계 설정은 다음과 같이 두 가지로 나눌 수 있다. 첫째, 두 프로젝트가 <그림 7>과 같이 독자적인 경계를 가지는 경우로, 사용자 측면에서 두 프로젝트가 별개의 업무인 경우이며, 또한 시스템적으로도 두 프로젝트 간에 사용하는 데이터베이스 서버가 달라서 별도로 운영되는 경우이다. 둘째, <그림 8>과 같이 사용자 입장에서는 두 프로젝트가 동일한 업무로 인지되지만, 두 프로젝트가 하나의 어플리케이션 경계에 속하



〈그림 7〉 프로젝트 간의 독립 어플리케이션 경계



〈그림 8〉 프로젝트의 공통 어플리케이션 경계

는 경우이다. 이 경우는 대개 데이터베이스를 공유하여 사용하는 환경이라고 할 수 있다.

4.3 데이터 기능 정의

세 번째 단계는 이전 단계에서 선택한 각 프로젝트 별로 사용되는 데이터를 찾아내는 것이다. 기능점수에서 구별하는 데이터의 종류는 제 3장에서 언급했듯이 내부논리화일(ILF)과 외부연계화일(EIF)로 나눈다.

- 내부논리화일(ILF) : 사용자가 알 수 있는 정보이고 어플리케이션 경계 내부에서 관리되면서 논리적으로 관련된 데이터 혹은 통제 정보를 말한다.
- 외부연계화일(EIF) : 어플리케이션에서 참조하는 논리적으로 관련된 데이터 혹은 통제 정보이지만, 다른 어플리케이션의 경계 내에서 관리된다. 이것의 주요 목적은 기능점수를 산정하는 어플리케이션 내에서 필요한 데이터가 존재하는지 확인하고 그 정확성을 파악하기 위한 것이다. 다시 말하면, 기능점수 산정 어플리케이션에서 EIF이면 다른 어플리케이션에서는 ILF로 관리되고 있어야 한다.

〈표 9〉 데이터 기능을 정의하기위한 화일 목록

어플리케이션			화일 목록		
전체 프로젝트	단위 프로젝트	단위 프로세스	화일	속성 개수	ILF/EIF 구분
전자정부	프로젝트 1	프로세스 1	파일 1	10	ILF
			파일 2	12	EIF
		프로세스 2	파일 3	20	ILF
			파일 4	30	ILF
	프로젝트 2	프로세스 3	파일 5	25	ILF
			파일 6	22	EIF
		프로세스 4	파일 7	10	ILF
			파일 8	40	ILF

이 단계의 결과로는 데이터 화일 목록이 도출되어야 한다. 각 프로젝트 별로 필요한 화일과 그 화일의 속성 개수가 결과로 나와야한다. 또한 각 화일이 내부논리화일인지 외부연계화일인지 구분해야 한다.

4.4 트랜잭션 기능 정의

트랜잭션이란 데이터 처리를 위하여 어플리케이션에 의해 사용자에게 제공되는 기능을 의미한다. 이것은 크게 외부입력(EI), 외부출력(EO), 외부조회(EQ)의 세 가지로 나눈다.

- 외부입력(EI) : 어플리케이션 바깥에서 들어오는 데이터와 통제 정보를 처리하는 기본적인 프로세스이다. 외부입력의 주목적은 내부논리화일(ILF)를 유지하고 시스템의 상태를 바꾸기 위한 것이다.
- 외부출력(EO) : 어플리케이션 바깥으로 데이터와 통제 정보를 내보내는 기본적인 프로세스이다. 이것의 주요 목적은 단순한 정보 검색이 아니라 특정 처리로직을 가지고 사용자에게 정보를 제공하고자 하는 것이다. 여기서 처리로직이란 적어도 하나 이상의 수학적인 공식을 포함하거나 새로운 데이터를 유도해내는 것이다. 외부출력에 의해서 내부논리화일(ILF)를 유지하고 시스템의 상태를 바꾸게 된다.
- 외부조회(EQ) : 어플리케이션 바깥으로 데이터와 통제 정보를 내보내는 기본적인 프로세스이다. 이것의 주요 목적은 내부논리화일(ILF)와 외부연계화일(EIF)에서 데이터와 통제 정보를 검색하여 사용자에게 제공하는 것이다. 여기에서의 처리로직에는 수학적인 공식이나 없어야 하고, 새로운 데이터를 유도해서도 안 된다. 즉, 외부조회에 의해서는 어떠한 내부논리화일도 변경되지 않고, 시스템의 상태도 바뀌지 않는다.

이 세 가지 종류의 차이에 대해서는 다음 <표 10>과 같이 요약할 수 있다.

〈표 10〉 트랜잭션 종류별 기능 비교

기 능	트랜잭션 종류		
	EI	EO	EQ
시스템의 상태 변경	주요 기능	부가 기능	N/A
하나 이상의 ILF 변경	주요 기능	부가 기능	N/A
사용자에게 정보제공	부가 기능	주요 기능	주요 기능

이 단계의 결과로는 각 어플리케이션별로 필요한 트랜잭션의 개수를 산출해 내는 <표 11>과 같은 표이다. 예를 들어, 단위업무1은 EI 트랜잭션이 총 12개가 있고 단위업무2는 EO와 EQ가 각각 하나씩 존재한다.

4.5 기능점수의 계산

마지막 단계인 215G 기능점수의 계산에서는 정확한 비용산정의 예비단계로 사용이 보다 간편한 간이기능점수 방식을 사용한다. 이

〈표 11〉 트랜잭션 기능 정의를 위한 트랜잭션 목록

어플리케이션				트랜잭션				
				EI			EO	EQ
전체 프로젝트	단위 프로젝트	단위 프로세스	단위업무	추가	수정	삭제	출력	조회
전자정부	프로젝트 1	프로세스 1	단위업무 1	4	4	4		
			단위업무 2				1	1
			단위업무 3					2
			단위업무 4	1	1			
		프로세스 2	단위업무 5				2	2
			단위업무 6		1	1		
			단위업무 7				1	1
			단위업무 8	3	3	3		
	프로젝트 2	프로세스 3	단위업무 1	2	2			
			단위업무 2		1	1		
			단위업무 3				2	2
			단위업무 4	1	1	1		
		프로세스 4	단위업무 5				1	1
			단위업무 6	1	1	1		
			단위업무 7				1	1
			단위업무 8	2	2			

것의 특징은 데이터기능과 트랜잭션기능에서 복잡도를 산출하지 않고, <표 6>에서 제시된 한국정보사회진흥원에서 사용하는 평균 복잡도 가중치를 사용한다.

5. 기능점수 계산 예제

본 장에서는 정보시스템 구축을 위한 정보전략계획 사업의 산출물을 가정하여, 제 4장에서 제시한 방법론에 따라서 정보시스템 개발 비용을 예측하는 과정을 설명한다.

5.1 기능점수 산출 종류 및 산출 범위 설정

정보전략 계획의 결과로 <표 8>과 같이 다섯 개의 프로젝트가 도출되었고, 신규개발인 프로젝트 1과 기능확장인 프로젝트 2가 기능점수 산정의 범위(Scope1)로 결정되었다고 가정하자.

5.2 데이터 기능의 계측

데이터에 대한 분석결과 <표 9>와 같이 8개의 화일이 도출되었으며, 그 중 내부논리화일(ILF)가 6개이고 나머지 2개가 외부연계화일(EIF)라고 하자. 프로젝트1의 경우, 신규개발 프로젝트이므로 미조정 데이터 기능(UAF)은 제 3장에서 설명한 대로 다음과 같이 계산된다. 여기서 가정은 프로젝트 1을 위해서 다른 시스템에서 이행해 와야 되는 업무는 없다고 생각한다.

UAF1 = (ILF 개수) × (ILF의 평균복잡도)
+ (EIF 개수) × (EIF의 평균복잡도)
= 3 × 8.1 + 1 × 5.3 = 29.6

또한 프로젝트 2의 경우에는 기능확장 프로젝트이므로, 변경된 부분, 삭제된 부분, 추가된 부분, 이행 부분으로 나누어 기능점수를 계산해야 하지만, 여기서는 단지 추가, 변경되는 경우만 있다고 가정하고 삭제되거나 이행되는 업무는 없거나 무시해도 좋을 만큼 적다고 가정하였다. 그러면 프로젝트 2의 데이터 기능은 다음과 같이 계산된다.

UAF2 = (ILF 개수) × (ILF의 평균복잡도)
+ (EIF 개수) × (EIF의 평균복잡도)
= 3 × 8.1 + 1 × 5.3 = 29.6

5.3 트랜잭션 기능의 계측

트랜잭션의 분석결과 프로젝트 1의 경우에 프로세스 1, 프로세스 2로 나누어져서 단위업무 8개가 수행되면서 트랜잭션을 발생시킨다.

프로젝트 1은 신규개발 프로젝트이므로 <표 11>에 제시된 트랜잭션 목록으로부터 다음과 같이 계산된다.

UAF3 = (EI 개수) × (EI 평균복잡도) + (EO 개수) × (EO 평균복잡도) + (EQ 개수) × (EQ 평균복잡도) = 25 × 3.8 + 4 × 4.9 + 6 × 3.7 = 136.8

프로젝트 2에 대한 트랜잭션 기능도 동일한 방법으로 다음과 같이 계산된다.

UAF4 = 16 × 3.8 + 4 × 4.9 + 4 × 3.7 = 95.2

〈표 12〉 미조정 기능점수의 계산

기능유형 \ 복잡성		프로젝트 1 (a)	프로젝트 2 (b)	Scope 1 (a + b)
데이터 기능	내부논리화일(ILF)	3 × 8.1 = 24.3	3 × 8.1 = 24.3	48.6
	외부연계화일(EIF)	1 × 5.3 = 5.3	1 × 5.3 = 5.3	10.6
트랜잭션 기능	외부입력(EI)	25 × 3.8 = 95.0	16 × 3.8 = 60.8	155.8
	외부출력(EO)	4 × 4.9 = 19.6	4 × 4.9 = 19.6	39.2
	외부조회(EQ)	6 × 3.7 = 22.2	4 × 3.7 = 14.8	37.0
미조정 기능점수		166.4	124.8	291.2

5.4 보정전 기능점수의 합산(UAF : Unadjusted Function Point)

일반적인 시스템 특성(General System Characteristics : GSC)을 고려하지 않은 보정전 기능점수를 산출하는 것은 두 프로젝트의 데이터 기능과 트랜잭션 기능을 합하면 된다. Scope1에 해당하는 프로젝트 1과 프로젝트 2의 미조정 기능점수는 <표 12>와 같이 계산된다.

6. 결 론

정보시스템의 다양성 및 환경 요인으로 인하여, 보정전 기능점수(UAF)를 산정한 후에는 조정계수를 이용하여 개별적인 시스템의 특성을 반영시킨다[3, 10]. 왜냐하면 정보시스템 개발에 소요되는 기능에는 데이터 기능과 트랜잭션 기능에 의해 충분히 표현되지 않는 요인들, 즉, 일반 시스템 특성(GSC)이 존재하기 때문이다. 그러나 보정에 따른 차이가 -35%~+35%로 조정전 규모에 크게 영향을 미치지 못할 뿐 아니라 현재 국제 표준화 기구에서 기능점수를 표준화 하면서 이 부분을 국제 표준에서 제외하고 있기 때문에, 본 연구에서는 일반 시스템 특성(GSC)을 고려하지 않은 보정전 기능점수까지 산출하는 것이 목표로 하였다.

소프트웨어 규모 측정은 소프트웨어 개발 규모를 예측하거나, 기존의 소프트웨어를 유지보수, 기능 추가, 운영하기 위한 투입공수와 소요비용을 산정하는 정량적 방법으로 활용되고 있다[2, 13]. 예를 들어, 2007년 10월에

〈표 13〉 기능점수당 사업비용 단가

(단위 : 원)[1]

단 계	분 석	설 계	구 현	시 험	합 계
기능점수당 단가	108,857	137,504	183,339	143,233	572,933

재개정된 정보통신부 고시 '소프트웨어 사업 대가의 기준'은 <표 13>과 같이 기능점수당 사업비용 단가를 보여준다.

실제로 소프트웨어 투입공수와 소요비용를 예측하는 것은 사업 수발주 기관의 외부적인 환경 및 시장경쟁 정도, 개발 언어 종류, 정보시스템 특성 등에 따라 다양한 영향을 받기 때문에 훨씬 예측하기 힘들고 정성적인 요인들이 상존한다. 그럼에도 불구하고 기능점수를 이용하여 정량적으로 정보시스템 규모를 측정하는 것은 현재로서는 사용자 중심에서 합리적인 정보시스템 대가 산정에 활용될 수 있기 때문이다.

참 고 문 헌

[1] 소프트웨어사업 대가의 기준, 정보통신부 고시 제2007-39호, 2007년 10월 30일 개정.

[2] F. Ahmed, S. Bouktif, A. Serhani, I. Khalil, "Integrating Function Point Project Information for Improving the Accuracy of Effort Estimation," 2nd International Conference on Advanced Engineering Computing and Applications in Sciences, 2008, pp. 193-198.

[3] A. K. Bharadwaj, T. R. G. Nair, "Mapping General System Characteristics to Non-Functional Requirements," IEEE International Conference on Advance Computing, 2009, pp. 1634-1638.

[4] B. W. Boehm, "Software and Its Impact : A Quantitative Assessment," Datamation, Vol. 19, No. 5, 1973, pp. 48-59.

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

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

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

[8] ISBUG (International Software Benchmarking Standards Group), http://www.isbsg.org.au/macroest.htm.

[9] C. Jones, and T. C. Jones, Estimating Software Costs, McGraw-Hill, 1998.

[10] G. C. Low, and 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.

[11] NESMA, "Function Point Analysis for Software Enhancement," 2001, C. R. Symons, "Function point analysis : difficulties and improvements," IEEE Transactions on Software Engineering, Vol. 14, No. 1, 1988, pp. 2-11.

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

[13] J. Wu, and X. Cai, "A Software Size Measurement Model for Large-Scale Business Applications," 2008 International Conference on Computer Science and Software Engineering 2008, pp. 39-42.

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

저 자 소 개

배준수	(E-mail : jsbae@chonbuk.ac.kr)
1993년	서울대학교 산업공학과 (학사)
1995년	서울대학교 산업공학과 (석사)
2000년	서울대학교 산업공학과 (박사)
2000년~2002년	LG CNS 전문과장
2002년~2004년	성결대학교 전자상거래학부 전임강사
2005년~2006년	미국 조지아공대(GIT) 박사후과정
2004년~현재	전북대학교 산업정보시스템공학과 조교수, 부교수
관심분야	경영정보시스템, BPM, 전자상거래

정재윤	(E-mail : jyjung@khu.ac.kr)
1999년	서울대학교 산업공학과 (학사)
2001년	서울대학교 산업공학과 (석사)
2005년	서울대학교 산업공학과 (박사)
2005년~2006년	네덜란드 아인트호벤공대 초빙연구원
2006년~2007년	서울시 유비쿼터스 컴퓨팅 원천기술개발지원센터
2007년~현재	경희대학교 산업경영공학과 전임강사
관심분야	정보시스템 분석, 인터넷 비즈니스, 소프트웨어 공학