

시스템 복잡도를 적용한 COSMIC-FFP 기반 소프트웨어 개발노력 추정 모델

박상기[†], 박만곤^{**}

요 약

성공적인 프로젝트 수행을 위해서 개발초기 단계에서 소프트웨어 개발노력 등 자원을 예측하는 것은 매우 중요하다. 그리고 이는 소프트웨어 규모를 통해 가능하며, 소프트웨어 규모 산정 방식의 최근 동향은 프로그램 라인 수(LOC)와 같은 개발자 관점의 접근방식 보다는 사용자가치 중심의 산정방식인 기능점수 분석기법(FPA)을 보다 선호하고 있다. 따라서 본 논문에서는 사례연구를 통해 실제 투입되는 개발노력을 측정하고, 완전기능점수 기법 매뉴얼에 따라 COSMIC Function Point를 직접 계산하였다. 그리고 산출된 데이터에 대한 회귀분석을 통해 COSMIC-FFP기반 소프트웨어 개발노력 추정 모델을 제안한다. 그리고, COSMIC-FFP 기법은 규모를 추정하는데 필요한 기능 요소들에 대한 가중치를 적용하지 않는 단점이 있다. 따라서, 본 연구에서는 제안된 모델의 정확성을 높이기 위해 시스템 복잡도를 가중치로 적용하며, 가중치 추정을 위한 모델도 제안한다.

A Model to Estimate Software Development Effort Based on COSMIC-FFP Using System Complexity

Sang-Ki Park[†], Man-Gon Park^{**}

ABSTRACT

It is very important to forecast a back resource of a software development effort at the early stage of development life cycle for successful project processing, and it is carried out through software size estimation. The recent trend of software size estimation method is focused on the user's value such as FPA. We measure the actual development effort through case study and calculate CFP directly according to the cosmic-ffp manual V.3.0. in this paper. We also propose the software development effort estimation model by using the produced data. COSMIC-FFP does not use weights of necessary function elements, and so it has disadvantage in estimating sizes. This paper proposes the estimation model to estimate the precision software size by using system complexity as weight.

Key words: Software Size(소프트웨어 규모), Development Effort Estimation(개발노력추정), COSMIC-FFP(완전기능점수), Complexity-Weighted(복잡도 가중치)

1. 서 론

성공적인 프로젝트란 제한된 인력, 시간 및 비용을 이용하여 사용자가 만족하는 시스템을 개발하는

것을 의미한다. 따라서, 정확한 개발노력, 개발비용, 개발기간의 예측이 필요하며, 이를 위해 소프트웨어 규모를 기본 단위로 채택하였다. 정확한 소프트웨어 규모의 예측은 향후 개발될 소프트웨어에 대한 정확

※ 교신저자(Corresponding Author) : 박만곤, 주소 : 부산광역시 남구 대연3동 599-1(608-737), 전화 : 051)629-6240, FAX : 051)628-6155, E-mail : mpark@pknu.ac.kr
접수일 : 2010년 5월 17일, 수정일 : 2010년 8월 13일

완료일 : 2010년 9월 1일

[†] 정회원, 부산발전연구원 정보화 담당

(E-mail : skpark@bdi.re.kr)

^{**} 종신회원, 부경대학교 IT융합응용공학과 교수

한 자원 배분을 가능하게 하며 이는 소프트웨어 품질에 결정적인 영향을 준다. 따라서 소프트웨어 개발에 소요되는 노력과 비용, 기간 등의 예측은 해당 프로젝트를 성공적으로 수행하기 위한 필수적인 요소라고 할 수 있으며, 이러한 예측 활동은 프로젝트 초반에 수행되어야 보다 효과적일 수 있다. 이러한 이유로 소프트웨어 개발 초기단계에서 개발할 소프트웨어 규모를 정확히 예측하는 활동이 점차로 중요시되고 있다[1,2].

우리나라에서는 소프트웨어의 규모를 산정하기 위해 주로 프로그램 라인수(LOC) 측정 방식을 사용해 왔는데, 이는 프로그램 소스 코드의 라인 수를 통해 소프트웨어의 규모를 산정하는 방법으로 개발자가 이해하기 쉽고 의미가 명확하기 때문에 지금까지 널리 사용되어 왔다. 하지만 프로그램 라인 수 측정 방식은 소프트웨어 개발초기에 라인 수를 예측하기 어렵고 소프트웨어 개발 언어나 환경에 영향을 받음으로 규모예측 방법의 근거가 희박하며, 추산시점에 하자가 발생할 수밖에 없는 등의 여러 가지 문제점을 가지고 있다[3,4].

따라서, 소프트웨어 규모 산정 방식의 최근 동향은 프로그램 라인 수(LOC)와 같은 개발자 관점의 접근 방식 보다는 사용자 가치 중심의 산정 방식인 기능점수 분석 기법(FPA)을 보다 선호하고 있다. 기능점수 기법은 소프트웨어 개발 초기단계에 적은 노력으로 소프트웨어 규모를 예측할 수 있다는 장점은 있으나 지나치게 단순화시킨 기능점수 예측 방식으로 인해 실제 산출결과와 예측된 기능 점수 사이에 측정 오차가 발생하는 단점이 있으며, 경영정보시스템(MIS: Management Information System)에 기반을 두고 개발되어 실시간이나 임베디드, 공학계산 소프트웨어에는 적용이 불가능하다. 하지만, 최근 정보 시스템은 임베디드 시스템이나 실시간 시스템의 비중이 점차적으로 확대되어가고 있는 추세이다. 따라서 이러한 기능점수의 문제점을 보완하기 위해 1998년 COSMIC(Common Software Management International Consortium)이 설립되어 경영정보시스템의 데이터 관리 위주인 기능점수 기법을 제어관리 위주인 실시간 시스템과 내장형 소프트웨어로 적용 범위를 확장하였다. 이를 완전기능점수 또는 COSMIC-FFP(Full Function Point)라 부른다[5].

COSMIC-FFP기반의 소프트웨어 규모 측정 방법

에 관한 많은 연구가 되고 있으나, COSMIC-FFP로 측정된 소프트웨어 규모에 기반하여 개발노력을 추정할 수 있는 모델 연구는 극소수에 불과함으로 모델 연구가 필요한 실정이다. 또한 완전기능점수는 복잡한 수학기산으로 구성된 알고리즘 위주의 소프트웨어 기능성을 측정하도록 설계되지 못한 단점을 갖고 있다. COSMIC-FFP기법은 소프트웨어 기능 프로세스의 컴포넌트 관점을 기술하고 있으나 규모 추정과 관련된 기능 요소들에 대하여 가중치를 부여하지 않는 문제점을 갖고 있다[6,7]. 소프트웨어의 규모를 보다 정확히 정량화하기 위해서는 기존의 COSMIC-FFP방법에 복잡도를 고려한 새로운 방법으로서의 전환이 요구되고 있다. 따라서, 본 논문에서는 첫째로 기존의 소프트웨어 규모 측정 방법 중 대표적인 기법인 완전기능점수에 매뉴얼에 따라 직접 완전기능점수를 산정해 본다. 그리고 사례연구를 통해 산출된 데이터를 이용하여 완전기능점수를 기반으로 한 개발노력 추정 모델을 제안한다. 또한 제안된 모델의 정확도를 제고하기 위하여 시스템 복잡도를 가중치로 부여하는 가중치 산정 모델도 제안한다. 그리고 제안된 모델에 따라 완전기능점수를 산출하고 가중치를 부여해 봄으로써 완전기능점수를 기반으로 한 개발노력 추정 모델에 대해 평가해 본다.

2. COSMIC-FFP

소프트웨어의 기능성에 기반하여 소프트웨어의 규모를 추정하기 위한 방법은 그림 1과 같이 발전하고 있다.

Common Software Measurement International Consortium (COSMIC)은 1998년 형성되었고, 소프트웨어 기능 규모 측정 방법의 새로운 세대에 기초를 둘 수 있는 기본적인 원리를 제안하였으며, 1999년에 제안된 원리를 구현한 측정방법인 COSMIC-FFP V.2.2을 발표 하였다. 그리고, 소프트웨어의 규모를 정량화하는 것은 소프트웨어 프로젝트의 노력, 비용, 개발기간을 평가하는 키 중의 하나로 일반적으로 인정된다[8,9].

COSMIC-FFP는 데이터 이동형을 4개의 속성으로 분류한다. 소프트웨어 측면에서 바라볼 때, 입구와 출구는 사용자와 데이터의 속성을 주고 받는 것이며, 읽기와 쓰기는 저장장치와 데이터 속성을 주고받

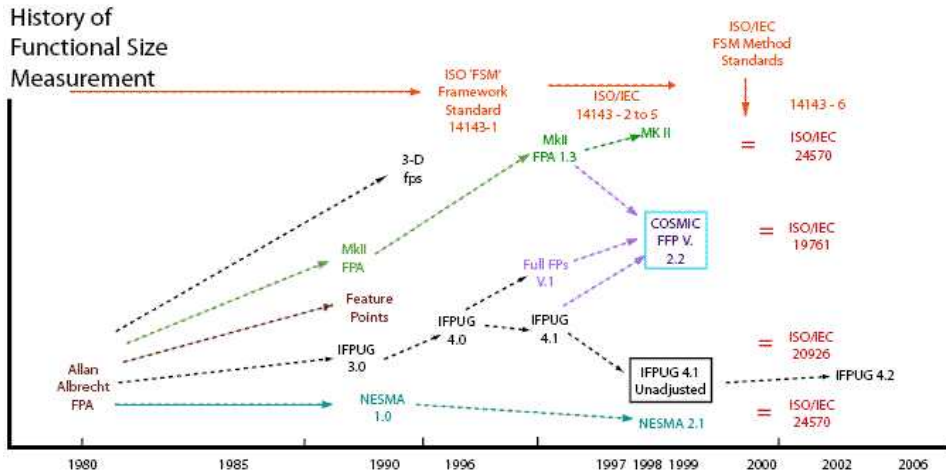


그림 1. 기능적인 규모측정 방법의 변천 과정

는 형태이다. COSMIC-FFP에 기반한 소프트웨어 규모의 단위는 Cfsu(Cosmic Functional Size Unit)로 표기한다. 그리고, COSMIC-FFP 기능 규모는 아래와 같이 ISO/IEC 19761에 제안한 방법에 따라 측정 할 수 있다[8,10,11].

COSMIC-FFP 측정 과정은 그림2와 같은 순서에 따라 진행된다[8].

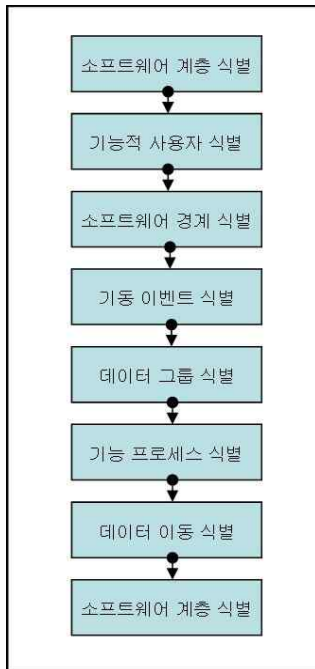


그림 2. COSMIC-FFP 측정 순서

(1) 소프트웨어 계층 식별

소프트웨어는 기능적으로 운영되는 환경 내의 기능 추상화 수준이 다른 여러 요소를 가질 수 있는데, 이들 요소 간의 기능 교환에 기초한 추상화 수준 사이에는 계층적 관계가 있다. 각 계층의 소프트웨어는 사용자에게 기능을 제공해야 한다. 서브계층의 소프트웨어는 이를 사용하는 계층의 소프트웨어에 기능적 서비스를 제공해야 한다.

(2) 기능적 사용자 식별

개발될 시스템과 상호작용하는 기능적 사용자가 누구인지를 식별하는 과정이다. 먼저 개발될 시스템에 데이터를 전송하는 기능적인 사용자가 있는지, 개발될 시스템으로부터 데이터를 받아들이는 기능적 사용자가 존재하는 지 그리고, 마지막으로 시스템과 상호작용하는 인간 사용자 있는지를 확인한다.

(3) 소프트웨어 경계 식별

기능 규모 측정(FSM, FUNCTIONAL USER REQUIREMENTS)의 범위 안에 있는 것으로 식별된 계층 내의 개별 소프트웨어의 경계는 반드시 식별되어야 하며, 일단 경계가 식별되면, 기능 규모 측정의 범위 안에서 식별된 각 기능적 사용자 요구(FUR, Functional User Requirements)은 개별 소프트웨어에 반드시 할당되어야 한다.

(4) 기동 이벤트 식별

기동 이벤트는 소프트웨어 경계 밖에 있는 기능적인 사용자에게 의해 발생하는 이벤트로써 이 이벤트들에 의해 하나 이상의 기능 프로세스들이 발생시킨다.

(5) 데이터 그룹 식별

FSM 범위 안에서 식별된 각 데이터 그룹은 데이터 속성에 대한 고유한 수집 활동을 통해 유일해야 하고 구별이 가능해야 하며, 소프트웨어의 기능적 사용자 요구 사항 내에 기술된 관심 대상의 하나와 직접적으로 관련되어야 한다. 관심 대상(object of interest)은 기능적 사용자 요구 사항의 관점에서 식별되는 것으로 물리적인 것일 수도 있고, 소프트웨어로 데이터를 처리하거나 또는 처리하여 저장하도록 요구하는 사용자 세계에서의 어떤 개념적 대상 또는 그 일부일 수도 있다.

(6) 기능 프로세스 식별

기능적 사용자 요구 사항 집합의 단위 요소로서 유일하고 응집력이 있으며 독립적으로 실행이 가능한 데이터 이동의 집합을 포함한다. 또한, 하나 이상의 기동 이벤트에 의해 직접 기동되거나 액터를 통해 간접적으로 기동되며, 기능 프로세스는 기동 이벤트에 반응하여 이루어져야 할 모든 요구 사항이 실행되어야 완료된다. 식별된 각 기능 프로세스는 최소한 하나의 식별된 기능적 사용자 요구로부터 추출되어야 하며, 기동 이벤트에 의거하여 시작되어야 한다. 그리고 들어가기 하나에 나가기 또는 쓰기와 같이 최소한 두 개의 데이터 이동을 포함해야 한다. 측정 단위로 1 Cfsu를 사용하며, 단위 소프트웨어에 대한 가장 작은 이론적 기능 규모는 2 Cfsu가 된다.

(7) 데이터 이동 식별

식별된 각 기능 프로세스는 구성되는 요소 데이터 이동들로 분할되어야 하며, 각 데이터 이동들은 들어가기, 나가기, 읽기 또는 쓰기 중의 하나이어야 한다. 또한 식별된 기능 프로세스 내에서 단 한번만 계산되어야 하며, 다른 데이터 이동형을 포함하지 않는다. COSMIC-FFP에는 들어가기(Entry, E), 나가기(Exit, X), 읽기(Read, R), 쓰기(Write, W) 4가지 유형의 데이터 이동이 있다. 들어가기(E)는 데이터 그룹이 사용자로부터 경계를 넘어 그것을 필요로 하는 기능 프로세스로 이동하는 데이터 이동형이다. 나가기(X)는 데이터 그룹이 기능 프로세스에서 경계를 넘어 그것을 요청한 사용자에게 이동하는 데이터 이동형이다. 읽기(R)는 데이터 이동을 요구한 기능 프로세스가 미치는 영향 범위 내의 영구 저장소로부터 데이터 그룹이 이동하는 데이터 이동형이다. 쓰기(W)는 기능 프로세스에 있는 데이터 그룹을 영구 저

장소로 이동하는 데이터 이동형이다.

(8) COSMIC-FFP 규모 계산

측정 단위인 1 Cfsu는 식별된 각 데이터 이동 즉, 식별된 각 데이터 이동의 사례 (들여가기, 나가기, 읽기, 쓰기)마다 할당되어야 하며, 식별된 기능 프로세스 내 모든 식별된 데이터 이동에 적용된 것으로 다음 절차에 의거하여 해당 기능 프로세스에 대한 단일 기능 규모 값으로 취합되어야 한다. 모든 데이터 이동형에 대해 단위 규모는 1이다. 먼저, 각 데이터 이동형 내 데이터 이동의 수에 해당 단위 규모를 곱한 후 각 데이터 이동형의 규모를 합산하여 기능 프로세스의 규모를 산정한다. 그 결과, 해당 기능 프로세스의 기능 규모는 다음 공식을 이용하여 Cfsu로 계산된다.

$$C_{fsu} = \sum (N_e * E_{us}) + \sum (N_x * X_{us}) + \sum (N_r * R_{us}) + \sum (N_w * W_{us})$$

단, N_e = 해당 기능 프로세스에 대한 들어가기의 수,

E_{us} = 들어가기의 단위 규모

N_x = 해당 기능 프로세스에 대한 나가기의 수,

X_{us} = 나가기의 단위 규모

N_r = 해당 기능 프로세스에 대한 읽기의 수,

R_{us} = 읽기의 단위 규모

N_w = 해당 기능 프로세스에 대한 쓰기의 수,

W_{us} = 쓰기의 단위 규모

3. 사례연구를 통한 COSMIC-FFP 측정

3.1 사례연구환경

이 절에서는 사례연구로 수행된 개발 프로젝트의 구성 및 각 서브시스템에 대해 소개하고, 시스템 개발을 위한 환경 및 투입인원, 개발기간에 대해 설명한다. 통합행정업무 시스템의 개발 기간은 총 4명의 인원이 요구사항 분석에서 구현까지 5개월의 기간 동안 투입 되어 수행되었으며, 개발 공정 중 시험은 이 기간에서 제외되었다. 프로젝트의 기능점수는 각각의 서브시스템별로 계산되었으며, 실제 프로젝트 투입시간 또한 서브시스템별/공정별로 측정되었다. 프로젝트에 참여한 모든 구성원은 한국소프트웨어 산업협회에서 제시하는 기술자 등급 및 자격기준에 따라 기사 10년 이상의 특급기술자로 구성되었으며, 개발 플랫폼은 Windows의 웹기반으로 개발언어는

ASP(Active Server Page), DBMS는 MS-SQL 2000, 웹서버는 IIS(Internet Information Server)5.0, 서버 운영체제는 Windows 2000 서버로 구성된다. 하드웨어는 Dell사의 Poweredge-SC1 420 Server로 메모리 1G, SCSI Hard 146G Dual, Xeon(TM) Processor 2.8GHz 사양으로 구성된다. 모든 프로그램 사용자는 웹을 통해서 접근하며, 각 서브시스템의 기능별로 사용자 권한을 부여 하여 개인별 접근 내용이 차별화 된다. 통합행정업무 시스템은 총 8개 서브시스템으로 구성되며, 새로운 시스템 개발을 목적으로 하며, 각 서브시스템별 개요는 다음과 같다.

첫째, 예산관리 시스템은 예산 항목별로 예산 편성, 배정 및 자금 배정과 반납의 기능을 가지며, 연도별 예산에 대한 집행현황 및 예산서를 관리한다. 둘째, 수입관리 시스템은 수입과 관련하여 발생하는 모든 결의서를 생성하며, 승인된 결의서에 대한 통계자료를 관리한다. 셋째, 계약관리 시스템은 지출과 관련하여 발생하는 모든 결의서를 생성하며, 법인카드 및 수탁관련 자료를 관리하며, 승인된 결의서에 대한 통계자료를 생성한다. 넷째, 회계관리 시스템은 수입과 계약관리 시스템으로부터 요청된 결의서에 대해 승인 또는 반려를 하며, 예산대비 지출 내역에 대한 통계자료를 생성 한다. 다섯째, 인사관리 시스템은 인사정보, 인사고과, 인사발령을 관리하며, 각종 증명서 출력 및 구성원에 대한 출신교별, 직급별 등 통계자료를 관리한다. 여섯째, 급여관리 시스템은 급여 및 수당에 대한 기본정보를 관리하고, 월별 급여 및 수당 계산과, 급여내역서 조회 및 연말정산, 퇴직금, 연가보상비를 계산한다. 일곱째, 총무관리 시스템은 출장관리, 복무관리, 차량관리, 상조회관리, 동호회관리, 우편물관리의 기능을 가지며, 각각은 독립적으로 수행된다. 여덟째, 연구실적관리 시스템 : 연구과제관리, 일정관리, 자료관리, 출판물관리, 위원회관리로 구성되며, 연구과제DB 구축을 통해 연구실적에 대한 통계자료를 생성한다.

3.2 COSMIC-FFP 측정 결과

이 절에서는 COSMIC-FFP 측정 매뉴얼Version 3.0에 따라 통합행정업무시스템의 완전기능점수를 직접 산정하여 결과 표 1과 표 2와 같은 결과 값을 얻었다. 먼저, 표 1은 예산관리시스템을 예로 들어 각 프로세스별 데이터 이동 수와 완전기능점수 및 예산

관리시스템 전체의 완전기능점수를 나타내고 있다. 표 2는 통합행정업무시스템 전체 및 각 서브시스템별 데이터 이동 수와 완전기능점수를 표시하고 있다.

4. COSMIC-FFP 기반 개발노력 추정 모델

본장에서는 사례연구를 통해 직접 산출한 Cfsu와 실제로 소프트웨어 개발에 투입된 개발노력 사이의 회귀분석을 통해 개발노력 추정 모델을 제안한다. 표 3은 통합행정업무시스템의 8개 서브시스템별로 산출된 완전기능점수와 실제 투입된 개발시간을 제시하고 있다.

그림 3은 사례연구를 통해 직접 산출한 Cfsu값을 독립변수로, 프로젝트개발에 실제 투입된 시간을 종속변수로 두어 선형회귀 분석 결과를 그래프로 나타내고 있으며, 표 4는 COSMIC-FFP기반 개발노력 추정 모델을 제시 하고 있다.

제안된 모델을 평가하기 위해 결정계수(Coefficient of determination, R^2) 와 MMRE(Mean Magnitude of Relative Error)를 적용하였다. 주어진 데이터들의 변동을 적절히 표현할 수 있는 모델을 찾기 위해 결정계수를 이용한다. 종속변수의 값(E)은 독립변수의 값(Cfsu)에 의해 결정되는 부분과 미지의 오차의 합으로 나타내며, 총 변동(주어진 데이터의 실제 개발노력 값)중에서 회귀직선(E)에 의해 설명되는 변동 비율을 결정계수($0 \leq R^2 \leq 1$)라 하며, 값이 클수록 신뢰성 있는 회귀직선인 모델이 얻어진다[12].

비록 결정계수에 의해 좋은 모델이 얻어졌더라도, 주어진 데이터들이 값이 큰 부분에 약간의 이상점 등이 분포한 상태에서 회귀직선이 이 이상점들을 잘 표현하고 값이 작은 표본에 대해서는 표현하지 못할 경우에도 결정계수는 큰 값을 나타낼 수 있다. 이 경우 좋은 모델이라고 판단할 수 없다. 따라서, 모델 정확도를 평가하는 측도로 MMRE가 적용된다.

$$RE = \frac{\text{추정치} - \text{실측치}}{\text{실측치}} \tag{1}$$

$$MRE(\text{Magnitude of RE}) = |RE| \tag{2}$$

$$MMRE(\text{Mean MRE}) = \frac{100}{n * \sum_{i=1}^n MRE} \tag{3}$$

MMRE가 작은 값이면 모델은 평균적으로 좋은 모델임을 알 수 있다. Cont et al.[13]는 $MMRE \leq$

표 1. 예산관리 COSMIC-FFP 측정 결과

번호	기능 프로세스 설명	데이터 이동 수				Cfsu
		E	X	R	W	
1	예산정보 등록	1	0	5	1	7
2	예산정보 수정	1	0	5	1	7
3	예산정보 삭제	1	0	5	1	7
4	예산정보 조회	1	1	5	0	7
5	예산편성 등록	1	0	1	1	3
6	예산편성 삭제	1	0	1	1	3
7	예산편성 열람	1	1	2	0	4
8	예산배정 등록	1	0	2	1	4
9	예산배정 수정	1	0	2	1	4
10	예산배정 삭제	1	0	2	1	4
11	예산배정 열람	1	1	3	0	5
12	자금배정 등록	1	0	2	1	4
13	자금배정 수정	1	0	2	1	4
14	자금배정 삭제	1	0	2	1	4
15	자금배정 열람	1	1	3	0	5
16	예산반납 등록	1	0	2	1	4
17	예산반납 수정	1	0	2	1	4
18	예산반납 삭제	1	0	2	1	4
19	예산반납 열람	1	1	3	0	5
20	예산집행 조회	1	2	4	0	7
합 계		19	5	51	14	89

표 2. 서브시스템별 COSMIC-FFP

번호	서브 시스템명	데이터 이동 수				Cfsu
		E	X	R	W	
1	예산관리 시스템	20	7	55	14	96
2	수입관리 시스템	8	5	10	3	26
3	계약관리 시스템	22	14	10	17	63
4	회계관리 시스템	6	1	4	1	12
5	인사관리 시스템	15	9	11	52	87
6	급여관리 시스템	28	9	22	27	86
7	총무관리 시스템	22	5	23	36	88
8	연구실적관리 시스템	23	6	54	16	96
합 계		0	0	0	0	0

0.25(25%)이면 개발노력을 예측하는 모델로 채택 가능한 것으로 고려하였다.

제안된 모델의 분석 결과 결정계수는 0.986으로 신뢰성 있는 값을 나타냈으며, MMRE은 약 21%로

Cont et al.[13]이 좋은 모델로 제시한 25% 보다 적은 값을 나타내고 있다. 따라서 본 논문에서 제시한 개발노력 추정 모델은 소프트웨어 개발노력을 예측하기에 상당히 정확성 있는 모델로 판명되었다.

표 3. 서브시스템별 COSMIC-FFP 및 실제개발시간

번호	서브 시스템명	Cfsu	실제개발 투입시간
1	예산관리 시스템	96	399
2	수입관리 시스템	26	279
3	계약관리 시스템	63	444
4	회계관리 시스템	12	216
5	인사관리 시스템	87	376
6	급여관리 시스템	86	462
7	총무관리 시스템	88	439
8	연구실적관리 시스템	96	420
합계		458	2,615

표 4. COSMIC-FFP기반 개발노력 추정 모델과 성능

모델	결정계수(R^2)	MMRE
$E = 5.367 \cdot Cfsu + 13.709$	0.986	20.88

5. 복잡도 가중치를 적용한 개발 노력 추정 모델

5.1 시스템 복잡도 추정모델

소프트웨어 기능을 기반으로 규모를 측정하는 방법들은 대부분 기능을 측정하는 과정에서 복잡도를 함께 고려하고 있다. 소프트웨어의 복잡도는 시스템 복잡도와 컴포넌트 복잡도로 구분할 수 있다. 하지만, COSMIC-FFP 기법은 소프트웨어 기능 프로세스의 컴포넌트 관점을 기술하고 있으나, 규모 추정과 관련된 기능요소들에 가중치를 부여하고 있지 않다 [14,15]. 따라서 본 장에서는 COSMIC-FFP 기반 개발노력 추정 모델의 정확성을 높이기 위해 시스템

복잡도를 추정할 수 있는 모델을 제안한다. 시스템 복잡도는 추정된 개발노력 수치에 가중치로 적용함으로써 시스템 개발 초기단계에서 보다 정확한 개발노력을 예측할 수 있게 된다.

시스템 복잡도는 4가지의 데이터 이동 타입(들어가기, 나가기, 읽기, 쓰기)을 기반으로 측정 되어진다. 각 레이어에서 데이터 이동은 COSMIC-FFP의 규칙과 원칙을 적용하여 식별하였다.

시스템 복잡도 (SC_x)는 내부 복잡도($intraC_x$)와 외부 복잡도($interC_x$)로 구성된다. 내부 복잡도는 같은 레이어에서 컴포넌트 사이의 복잡도를 나타내며, 외부복잡도는 레이어 사이의 복잡도를 의미한다. 따라서, 본 논문에서는 G.Zayaraz, and P. Thambidurai[16]가 제안한 시스템 복잡도 식(4), (5), (6)을 이용하여 본 논문 4장에서 제안된 개발노력 추정 모델에 대한 가중치로 사용하였다. 내부 복잡도, 외부 복잡도 모두 범위는 0에서 1사이의 값을 가진다.

$$intraC_x = \sum_{i=1}^L \frac{(E_i * X_i)^2}{(N_i^2 * (N_i * (N_i - 1)))^2} \tag{4}$$

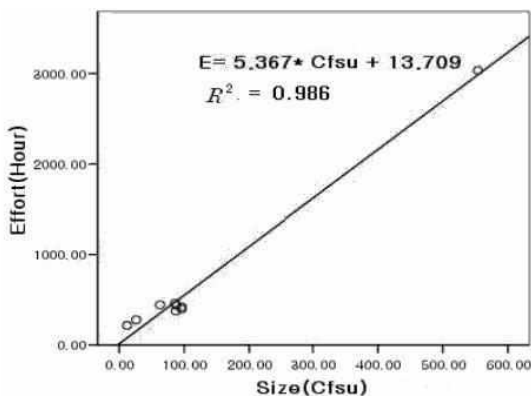


그림 3. 개발노력 추정 선형 회귀분석 모델

표 5. 시스템 복잡도에 사용되는 매개변수

매개변수	심볼
Number of components	N
Layer	L

$$interC_x = \sum_{i=1}^{L-1} \frac{(E_{(i,i+1)} * X_{(i,i+1)})^2 + (R_{(i,i+1)} * W_{(i,i+1)})^2}{(N_i * N_{i+1})^4} \quad (5)$$

시스템 복잡도 = 내부 복잡도 + 외부 복잡도 (6)

위에서 제안된 식(4), 식(5), 식(6)을 사례연구에 적용하여 시스템 복잡도를 계산한 결과 표 6과 같은 값을 도출하였다. 데이터 이동 수는 CFFP V3.0을 기반으로 측정하였으며, 컴포넌트 수는 각 서브 시스템에서 기능별로 그룹을 구성하여 하나의 모듈(컴포넌트)로 간주하였다.

5.2 시스템 복잡도를 이용한 개발노력 추정치 보정

본 절에서는 시스템 복잡도(독립변수)와 실제개발 투입시간과 본 논문에서 제안된 개발노력 추정모델에 의해 산출된 개발노력 예측치 사이의 차이값(종속변수)에 대한 회귀분석을 통해 개발노력 오차를 보정할 수 있는 추정모델을 제안한다. 개발노력의

오차를 추정함으로써 COSMIC-FFP기반 개발노력 추정모델에 의해 추정된 개발 노력에 오차를 보정해 줌으로써 보다 정확한 소프트웨어 개발노력이 예측 가능하다. 표 7은 서브시스템별 시스템 복잡도와 본 논문에서 제안된 COSMIC-FFP기반 개발노력 추정모델에 의해 산정되어진 개발노력 추정치와 실제 투입된 개발노력 사이의 오차를 나타내고 있다.

표 7의 8개 서브시스템을 대상으로 시스템 복잡도와 개발노력 오차 사이의 회귀분석 결과는 그림 5에 제시하였다. 그림 4에 나타난 시스템 복잡도를 이용한 개발노력 오차 추정 모델의 성능 분석 결과는 표 8에 제시되어 있다.

표 7에서 제시된 데이터를 이용하여 COSMIC-FFP기반으로 산출된 시스템 복잡도를 독립변수로, 실제개발 투입시간과 본 논문에서 제안된 개발노력 추정모델에 의해 산출된 개발노력 예측치 사이의 차이값을 종속변수로 두어 회귀분석을 통해 선형회귀 모델과 지수 회귀모델을 표 8과 같이 구하였다. 본 논문에서 제안한 개발노력 추정모델에 의해 산정된

표 6. 서브시스템별 시스템 복잡도

번호	서브 시스템명	데이터 이동 수				컴포넌트 수	Cfsu	시스템 복잡도
		E	X	R	W			
1	예산관리 시스템	20	7	55	14	4	96	0.53
2	수입관리 시스템	8	5	10	3	3	26	0.54
3	계약관리 시스템	22	14	10	17	5	63	0.37
4	회계관리 시스템	6	1	4	1	2	12	0.56
5	인사관리 시스템	15	9	11	52	4	87	0.49
6	급여관리 시스템	28	9	22	27	6	86	0.05
7	총무관리 시스템	22	5	23	36	4	88	0.32
8	연구실적관리 시스템	23	6	54	16	4	96	0.52

표 7. 서브시스템별 시스템 복잡도와 개발노력 오차

번호	서브 시스템명	복잡도	개발노력 오차 (절대치)
1	예산관리 시스템	0.53	129
2	수입관리 시스템	0.54	126
3	계약관리 시스템	0.37	93
4	회계관리 시스템	0.56	138
5	인사관리 시스템	0.49	104
6	급여관리 시스템	0.05	13
7	총무관리 시스템	0.32	47
8	연구실적관리 시스템	0.52	108

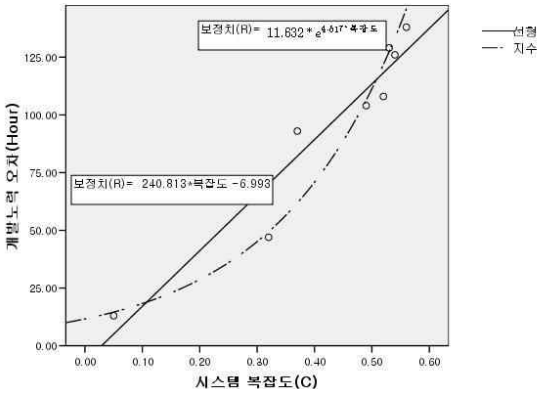


그림 4. 개발노력 오차 추정 회귀분석 모델

개발노력 추정치에 시스템 복잡도를 적용하여 추정된 개발노력 오차를 보정하면 실제 프로젝트에 투입될 개발노력을 보다 정확히 예측할 수 있으므로 본 논문 5.2절에서 제안된 모델에 의해 산출되는 값을 보정치(R)라고 정의하였다. 두 모델의 성능평가 결과 지수 모델이 선형 모델에 비해 결정계수 및 MMRE 모두에서 좋은 것으로 판명되었다. 따라서 본 논문에서는 지수회귀 모델에 시스템 복잡도를 적용하여 개발노력 추정치를 보정하기 위한 보정치 추정 모델로 선정하였다.

표 7에 나타난 8개의 서브시스템별로 시스템 복잡도를 적용하여 실제 소프트웨어 개발노력이 투입된 시간과 제안 모델에 의한 추정치 사이의 오차를 비교해 본 결과 Cfsu 값이 85이상일 경우는 제안 모델에 적용한 개발노력 추정치가 실제 투입되는 개발시간보다 더 많이 투입될 것으로 예측되었으며, 85미만인 경우 추정치가 실제투입시간보다 적은 값을 나타냈다. 따라서 시스템 복잡도를 적용하여 모델 추정치에 대한 개발노력을 보정할 경우 Cfsu값이 85이상인 경우 오차 보정치를 개발노력 추정치에서 마이너스하고, 85미만인 경우 반대로 오차 보정치를 합산해 주면 보다 정확한 개발노력을 예측할 수 있다.

5.3 제안된 개발노력 추정 모델에 따른 개발노력 추정 과정

표 8. 시스템 복잡도를 이용한 개발노력 보정치 추정 모델

구분	모델	결정계수(R^2)	MMRE
선형	보정치(R) = 240.813 · 복잡도 - 6.993	0.921	6.64
지수	보정치(R) = 11.632 * e ^{4.517 * 복잡도}	0.955	5.28

제안된 모델을 통해 투입될 개발노력 추정치를 산정하고 여기에 시스템 복잡도를 적용한 보정치를 부여함으로써 보다 정확한 개발노력 예측이 가능하다. 따라서, 다음과 순서로 개발노력을 추정하면 소프트웨어 개발 초기단계에서 보다 정확한 결과를 얻을 수 있다.

step 1 :

본 논문에서 제안한 COSMIC-FFP 기반의 개발노력 추정 모델

$E = 5.367 \cdot Cfsu + 13.709$ 을 통해 개발노력을 추정한다.

step 2 :

$$intraC_x = \sum_{i=1}^L \frac{(E_i * X_i)^2}{(N_i^2 * (N_i * (N_i - 1)))^2} \tag{4}$$

$$interC_x = \sum_{i=1}^{L-1} \frac{(E_{(i,i+1)} * X_{(i,i+1)})^2 + (R_{(i,i+1)} * W_{(i,i+1)})^2}{(N_i * N_{i+1})^4} \tag{5}$$

식(4)을 통해 시스템 내부 복잡도, 식(5)를 통해 외부 복잡도를 산출한다.

step 3 :

시스템 복잡도 = 내부 복잡도 + 외부 복잡도 (6)

식(4)(5)(6)을 이용하여 시스템 복잡도를 산출한다.

step 4 :

본 논문에서 제안한 복잡도를 이용한 개발노력 보정 모델

보정치(R) = 11.632 * e^{4.517 * 복잡도} 을 이용하여 개발노력 보정치를 산출한다.

step 5 :

Cfsu >= 85 이면 개발노력 추정치(E)-보정치(R) (7)

Cfsu < 85 이면 개발노력 추정치(E)+보정치(R) (8)

추정된 Cfsu 값에 따라 식(7) 또는 식(8)을 적용하여 보정된 최종 개발노력 추정치를 산출한다.

6. 결 론

성공적인 프로젝트 수행을 위해서 개발초기 단계

에서 소프트웨어 개발노력 등 자원을 예측하는 매우 중요하다. 그리고 이는 소프트웨어 규모를 통해 가능하며, 소프트웨어 규모 산정 방식의 최근 동향은 사용자가치 중심의 산정방식인 기능점수 분석기법을 보다 선호하고 있다. 하지만, FPA는 MIS영역 이외의 분야에서는 적용하기가 어렵다는 취약점이 있다 [17,18]. 따라서 본 논문에서는 MIS영역인 데이터 관리 위주에서 제어관리 위주인 실시간 시스템과 내장형(Embedded)소프트웨어에도 적용할 수 있는 COSMIC-FFP를 기반으로 개발노력 추정 모델을 제안하였다.

본 논문은 소프트웨어 개발 초기단계에 개발노력을 추정할 수 있는 모델을 제안하였다. 추정된 개발노력을 통해 개발에 소요될 비용이나 개발기간까지도 예측이 가능하다. 사례연구를 통해 실제 프로젝트를 수행하면서 소요되는 개발노력을 측정하였으며, 이 데이터를 통해 COSMIC-FFP기반 개발노력 추정 모델을 제안하고 성능 평가를 하였다. 그리고 제안된 모델의 정확성을 높이기 위해 시스템 복잡도를 이용하여 제안된 모델에 의해 추정된 개발노력을 보정할 수 있는 모델 또한 제안하고 평가 하였다. 제안된 모델들은 실제 프로젝트 수행을 통해 얻어진 결과를 기반하고 있으므로, 현재 프로젝트 개발현장에서 사용하면 보다 정확한 개발기간, 개발비용, 개발노력을 예측할 수 있다[19].

소프트웨어 개발에 영향을 주는 요소로서 시스템 복잡도 뿐만 아니라, 개발자의 스킬, 개발경험, 개발환경 등 다양한 것들이 있다. 소프트웨어 개발노력 예측시 이러한 요소들을 다 반영한다면, 보다 정확한 예측이 가능할 것이다. 따라서 소프트웨어 개발에 영향을 주는 다양한 요소의 발굴 및 이를 개발노력 예측에 반영할 방법에 대한 연구가 수행될 것이다. 또한 완전 기능점수는 최근 들어 국제 표준화가 되었으며, 현재는 개발현장에서 적합성이 검증되고 있는 단계로 실제적으로 많은 데이터가 존재하지 않는다. 따라서 추후 다양한 종류의 실험데이터 수집을 통해 범용성을 높일 수 있는 모델에 대한 연구가 수행될 것이다.

참 고 문 헌

- [1] A.J.Albrecht and J.E.Gaffney, "Software Function, Source Line of Code and Development Effort Prediction : A Software Science Validation," *IEEE Trans. on Software Eng.*, Vol. SE-9, No. 6, pp. 639-648, 1983.
- [2] A.Abran, C. Symons, and S. Olgyi, "An Overview of COSMIC-FFP Field Trial Results," ESCOM 2001, London, England, 2001.
- [3] Kamer, Chris F., and Benjamin S. Porter "Improving the Reliability of Function Point Measurement:An Empirical Study," *IEEE Transactions on Software Engineering*, Vol. 18, No. 11, pp. 1011-1024, 1992.
- [4] Stephen H. Kan , *Metrics and Models in Software Quality Engineering*, Pearson Education Publications, Low Price Edition, pp. 311-328, 2003.
- [5] F.Bootsma, "Applying Full Function Points to Drive Strategic Business Improvement with the Real-Time Software Environment," Annual IFPUG Conference, New Orleans, 1999.
- [6] 박주석, 정기원, "소프트웨어 개발 비용을 추정하기 위한 FFP기반 모델," 정보처리학회논문지 D, 제 10-D권 제 7호, pp. 1137-1144, 2003.
- [7] G. Levesque and V. Bevo, "Comparing COSMIC-FFP and SLIM Back-Firing Function Points Size Measurements," ICSSEA, 2001.
- [8] C. Symons, "COSMIC-FFP Measurement Manual, Version 3.0 (The COSMIC Implementation Guide for ISO/IEC 19761:2003)," Common Software MeasurementInternational Consortium, 2003.
- [9] Albrecht, A. J. and J. E. Gaffney, "SoftwareFunction, Source Line of Code and Development Effort Prediction : A Software Science Validation," *IEEE Trans. on Software Eng.*, 1983.
- [10] F. Bertoa and A. Vallecillo, "Usability Metrics for Software Components," Proceedings of the 8th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering(QAOOSE 2004), Oslo, Norway, June 2004.
- [11] G.Zayaraz, P.Thambidurai, Madhu Srinivasan and Paul Rodrigues, "Software Architectural

Quality Assessment Through COSMIC FFP," Proceedings of the National Conference on Product Development with Mechatronic Systems for Global Quality, PMGQ 2005, May 2-3, pp. 211-216, 2005.

- [12] B. W. Boehm, "Software Engineering Economics," *IEEE Trans. on Software Eng.*, Vol. 10, No. 1, pp. 7-19, 1984.
- [13] ISO/IEC FDIS 19761, "Software Engineering COSMIC-FFP-A Functional Size Measurement Method," 2002.
- [14] S. D. Conte, H. E. Dunsmore and V. Y. Shen, *Software Engineering Metrics and Models*, Menlo Park., CA:Benjamin Cummings, 1986.
- [15] N. Monaka, A. Kakural, E. Bukhary, and M. Azuma, "A Complexity-Weighted Functional Size Metric for Interactive Software," Advanced Institute for Science & Eng., Waseda University, 2002.
- [16] Garlan. D and Shaw. M, "An Introduction to Software Architecture," *Advances in Software Engineering and Knowledge Engineering*, Vol. 2, pp. 1-39, 1993.
- [17] G. Zayaraz, and P. Thambidurai, "Quantitative Measurement of Software Architectural Qualities through COSMIC FFP," *IEEE*, 2006.
- [18] F.Bootsma, "Applying Full Function Points to Drive Strategic Business Improvement with the Real-Time Software Environment," Annual IFPUG Conference, New Orleans, 1999.
- [19] Cote and St-Pierre, "A Model for Estimating Perfective Software Maintenance projects," Proceedings of Conference on Software Maintenance, Vol. 11, pp. 328-334, 1990.



박 상 기

1995년 울산대학교 미생물학과 이학사
 2005년 부산대학교 전산학 공학석사
 2008년 부경대학교 정보시스템학과 박사수료
 1999년~2000년 (주)대흥전기부설연구소 연구원
 2000년~현재 부산발전연구원 정보화 담당
 관심분야: 소프트웨어 공학, 소프트웨어 매트릭스, 소프트웨어 개발노력, 정보시스템 감리



박 만 곤

경북대학교 전산통계학(이학사, 교육학석사, 이학박사)
 Philippine Women's University (국제행정학석사)
 University of Rizal System, Philippines (명예 기술학박사)
 Dept. of Electrical & Computer Engineering, University of Kansas (Post Doc.)
 1981년~현재 부경대학교 IT융합응용공학과 교수
 1997년~현재 한국멀티미디어학회(KMMS) 부회장, 회장 및 명예회장
 2002년~2007년 정부간 국제기구 CPSC (콜롬보폴렌기술교육대학) 총재 (Director General and CEO)
 2004년~2007년 Asia Pacific Accreditation and Certification Commission 아태지역 인증 및 검증위원회 위원장
 1990년~1991년 영국 Univeristy of Liverpool 전자계산학과 객원교수
 1996년~1997년 호주 University of South Australia 컴퓨터정보과학부 객원교수
 2005년~현재 유네스코 (UNESCO-UNEVOC) 자문위원, 아시아개발은행(ADB) 자문관
 관심분야: 소프트웨어신뢰성공학, 비즈니스 프로세스 재공학 (BPR), 소프트웨어 공학 및 재공학, 멀티미디어정보처리기술, 정보시스템성능평가, ICT 기반 HRD System