

소프트웨어 규모산정을 위한 기능점수 개선 Micro-FP 모형의 제안

안연식*

An Enhanced Function Point Model for Software Size Estimation: Micro-FP Model

Yeon S. Ahn*

요약

소프트웨어 규모 측정에 국제표준(IFPUG, FPA)에 기반한 기능점수 방식은 개발자 관점이 아닌 사용자 관점의 접근이라는 장점으로 널리 활용되고 있다. 그러나 현재의 기능점수 측정 방식은 복잡도 적용에서의 상한이 존재하는 등 문제점을 내포하고 있다. 본 논문에서는 이러한 복잡도 상한의 문제를 극복하고, 회귀식을 내장하고 있어 쉽게 소프트웨어 규모를 측정할 수 있으며, 특히 대형 애플리케이션에도 적용할 수 있는 개선된 기능점수 측정모형을 제시한다. 조직내에서 운영중인 10개 애플리케이션에 적용하여 적합성을 평가한 결과, 제시한 개선 모형이 기존의 FPA(Function Point Analysis) 방식보다 LOC(Line of Code) 규모를 비교하여 상관관계가 더 높은 장점을 확인할 수 있었다.

Abstract

Function Point Method have been applied to measure software size estimation in industry because it supports to estimate the software's size by user's view not developer's. However, the current function point method has some problems for example complexity's upper limit etc. So, In this paper, an enhanced function point model, Micro FP model, was suggested. Using this model, software effort estimation can be more efficiently because this model has some regression equation. This model specially can be applied to estimate in detail the large application system's size. Analysis results show that measured software size by this Micro FP model has the advantage with more correlative between the one of LOC, as of 10 applications operated in an large organization.

▶ Keyword : 기능점수모형(Function Point), IFPUG, LOC, 소프트웨어규모예측(software size estimation)

• 제1저자 : 안연식

• 투고일 : 2009. 11. 25, 심사일 : 2009. 11. 28, 게재확정일 : 2009. 12. 24.

* 경원대학교 경영학과 부교수

※ 본 논문은 2009년도 경원대학교 연구지원에 의한 결과임

I. 서론

소프트웨어 규모를 추정하는 기법은 통상적으로 스텝수(Line Of Code, 이하 LOC) 방식과 FSM(Functional Size Measurement) 방식이 사용되고 있다[1].

이중에서 FSM 방식은 IBM의 Allan J. Albrecht에 의해 최초로 고안된 기능점수 분석(Function Point Analysis, 이하 FPA)방식이 기원이며, 고객이 요구하는 시스템의 기능을 크기로 해당 소프트웨어의 규모를 측정하는 방식이다[2, 3, 4, 5]. 현재 FSM 방식은 국제 표준(ISO/IEC 14143) 방식이며, 해당 표준을 따르는 방법으로는 FPA, NESMA, MarkII, COSMIC 등이 있다[6, 7, 8, 9]. 우리나라에서도 소프트웨어 사업대가 산정기준에서 분수 방식에서 기능점수 방식으로 전환하였다[10].

그러나 공공사업과 대규모 프로젝트를 시작으로 기능점수 방식을 도입하고 있지만, 기능측정에서 가지고 있는 사용자 주관적이라는 모호성과 다양한 기능의 복잡도를 고정된 복잡도 매트릭스로 표현하는 한계, 그리고 기능점수 측정 경험 부족 등이 문제점으로 지적되고 있는 실정이다[1]. 본 논문에서는 FPA 방식이 가지고 있는 문제점을 분석하고 이를 개선할 수 있는 모델을 제시하여 기존의 FPA 방식보다 효율적이고 효과적인 측정 방식을 구현한다.

II. 관련 연구

2.1 FPA (Function Point Analysis)

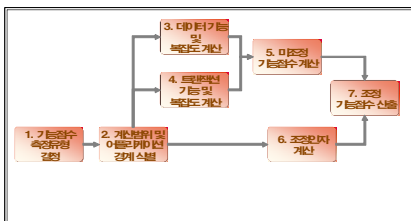


그림 1. FP 측정 절차

Fig 1. Measurement Process of FP

이 과정에서 가장 중요한 측정절차는 데이터 기능과 트랜잭션 기능의 측정이며, 이때 해당 기능의 기능유형 별 복잡도는 데이터 기능의 경우 표 1에서와 같이 레코드 유형의 개수와 DET(Data Element Type)의 개수에 따라, 트랜잭션 기능의 경우는 표 2 표 3에서와 같이 참조하는 데이터 기능인 FTR(File Type Referenced)과 DET의 개수에 따라 '낮음', '보통', '높음'의 세가지로 분류되며, 이어서 표 4에서와 같은 내부 복잡도를 평가하여 가중치를 설정한다.

표 1. ILF, EIF 유형 기능점수 내부 복잡도
Table 1. Inner Complexity of ILF, EIF Type

레코드요소	데이터요소유형의 개수		
유형의 개수	1~19	20~50	51이상
1	낮음	낮음	보통
2~5	낮음	보통	높음
6이상	보통	높음	높음

표 2. EI 유형 기능점수 내부 복잡도
Table 3. Inner Complexity of EI Type

참조파일	데이터요소유형의 개수		
유형의 개수	1~4	5~15	16이상
0~1	낮음	낮음	보통
2	낮음	보통	높음
3이상	보통	높음	높음

표 3. EO, EQ 유형 기능점수 내부 복잡도
Table 3. Inner Complexity of EO, EQ Type

참조파일	데이터요소유형의 개수		
유형의 개수	1~5	6~19	20이상
0~1	낮음	낮음	보통
2~3	낮음	보통	높음
4이상	보통	높음	높음

표 4. 각 기능점수의 내부 복잡도 별 가중치
Table 4. Inner Complexity of each FP Type

참조파일 유형의 개수	복잡도		
	낮음	보통	높음
ILF	x 7	x 10	x 15
EIF	x 5	x 7	x 10
EI/EQ	x 3	x 4	x 6
EO	x 4	x 5	x 7

2.2 관련 규모 측정 기법

NESMA(Netherlands Software Metrics Association)은 Detailed, Estimated, Indicative의 3가지 측정 방식을 정의하고 있다(9, 14). Detailed Count 방식은 일반시스템특성(GSC, General System Characteristics)을 적용하지 않는다. Estimated Count 방식은 각 기능 유형에 default 가중치를 부여한다. Indicative Count 방식은 데이터 기능 유형(ILF, EIF)의 개수만을 고려하므로 개략적인 추정에 적합한 방식이다.

Charles Symons는 기능 측정의 단점들을 고려하여 수정된 Mark II 방식의 측정기법을 제안했는데(15), 주요 내용은 다음과 같다.

- ① 엔티티의 수와 그 성능을 측정하여, 파일을 다루는 데 있어서의 주관성을 줄인다.
- ② 애플리케이션 경계와 상관없이 단일 시스템으로서 또는 관련된 서브시스템의 집합으로 동일한 합계를 계산하도록 기능점수 기법을 수정한다.
- ③ 사용자에게 전달되는 가치보다는 그 기능을 구현하는데 소요되는 노력에 초점을 맞춘다.
- ④ 14개의 일반 시스템 특성에 추가로 6개의 복잡도 요소를 고려한다.

III. 기존 FPA 모형의 분석

본 장에서는 기존 FPA 방식의 문제점들을 문헌을 기초로 분석한다(16, 17, 18, 19).

3.1 논리파일 측정

논리파일의 기능점수를 측정 과정에서 이슈가 되는 부분은 논리파일의 복잡도를 산정하는 과정이다. 식별된 모든 ILF와 EIF에 대해 각각의 데이터요소유형(DET: Data Element Type)과 레코드요소유형(RET: Record Element Type)의 수에 따라 복잡도를 설정하도록 되어있는데(12), 사용자의 관점에서 RET를 식별하는 과정에서 많은 해석의 차이로 논란이 있다. RET는 ILF나 EIF 안에서 사용자 식별 가능한 데이터 요소의 서브그룹으로서 ① 선택적 (Optional), ② 필수적 (Madatory) 서브그룹이 있다. 여기에서 선택적 서브그룹이란 사용자가 데이터의 인스턴스를 추가 또는 생성하는 단위 프로세스에서 서브 그룹을 사용할 수도 있고 사용하지 않을 수도 있는 데이터 그룹을 의미하며 필수적 서브그룹은 사용자가 적어도 하나 이상을 사용해야하는 서브그룹을 의미한다.

다. 그림 2에서는 같은 유형의 RET산정에서 결과가 상이할 수 있음을 보여주고 있다.

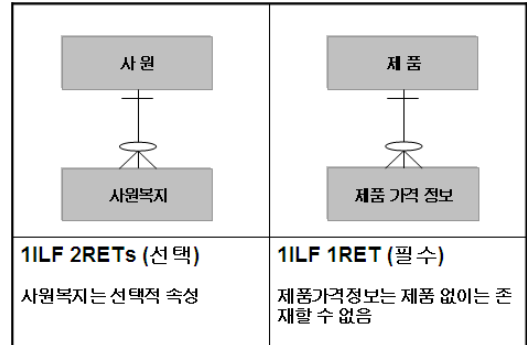


그림 2. RET 산정 예제
Fig 2. Calculation Example of RET

3.2 VAF (Value Adjustment Factor)

조정인자(VAF)는 애플리케이션의 기능에 등급을 부여한 14개의 일반시스템특성(General System Characteristics, 이하 GSC)에 기반을 두고 있으며 각 특성에는 영향도를 평가하여 0 ~ 5까지의 6 등급을 부여한다. 14개의 영향도를 합하여 총영향도(TDI)를 산정하고, 다음 식을 통해서 그 규모를 -35% ~ +35%까지 조정한다.

$$VAF = (\text{총영향도} * 0.01) + 0.65$$

VAF의 적용에도 다음과 같은 논란이 제기되어왔다(20, 21).

- ① 영향도 등급 판정이 주관적이다.
- ② 모든 시스템특성을 평균적인 영향으로 산정하여도 +7%까지만 조정된다.
- ③ 산정하는 과정에서 비율척도(ratio)와 서열척도(ordinal)를 혼용하고 있다.
- ④ 14가지 시스템특성이 최근의 다양하고 동적인 소프트웨어 환경을 제대로 반영하지 못한다.

3.3 복잡도 가중치

Albrecht의 FP 방식에서 가중치 적용의 획일성도 논란의 대상이 되어왔다(3, 7, 10). 각 기능 별 가중치가 표 5에서와 같이 single matrix에 의해 정해지는 방식으로서, 다양한 기능에 대해 차별화된 가중치 적용이 어렵고, 특히 공학용 계산이나 복잡한 알고리즘이 탑재된 경우 단점으로 작용한다(1). 따라서 항공, 교통통제, 실시간 시스템 등에는 COSMIC v.3 등 다른 FSM 방식으로 전환하는 사례가 늘고 있다(22).

표 5. 기능점수 복잡도에 따른 가중치 부과표
Table 5. Weight for FP Complexity

참조파일 유형의 개수	복잡도		
	낮음	보통	높음
ILF	x 7	x 10	x 15
EIF	x 5	x 7	x 10
EI/EQ	x 3	x 4	x 6
EO	x 4	x 5	x 7

IV. Micro Function Point 모형

본 연구에서는 기존 FP 방식의 문제점을 해결하기 위하여 그림 3과 같은 Micro Function Point 기능점수 모형(이하 Micro-FP)을 제안한다.

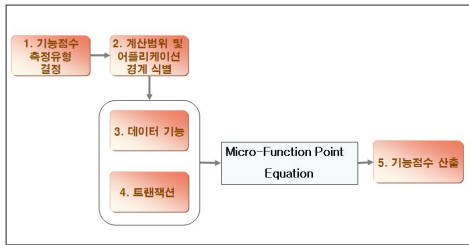


그림 3. Micro-FP 제안 모형
Fig. 3. Suggested Micro-FP Model

4.1 Micro-FP Model의 복잡도 측정

표 6에서와 같이 기존에 사용되는 IFPUG 모형에서는 외부 입력(EI) 유형의 경우에만 보더라도, 0개의 FTR과 1개의 DET를 갖는 'EI'의 미조정기능점수는 3점으로 1개의 FTR과 15개의 DET를 갖는 'EI'의 미조정 기능점수와 그 값이 같다. 이는 기능간의 세밀한 규모차이를 측정할 수 없음을 의미한다.

표 6. EI의 복잡도 및 가중치
Table 6. Complexity and Weight for EI Type

FTR	DET 갯수		
	1 ~ 4	5 ~ 15	16이상
0 ~ 1	낮음(3)	낮음(3)	보통(4)
2	낮음(3)	보통(4)	높음(6)
3이상	보통(4)	높음(6)	높음(6)

따라서 이런 점의 개선이 필요함을 알 수 있다. 제안된 Micro-FP 모형에서는 '낮음', '보통', '높음'으로 제한되어있는 현재의 복잡도의 문제점을 개선하기 위해서 Matirx를 2배로 확장시켜 표 7과 같은 좀 더 세밀한 Matirx를 얻는다. 즉 DET 개수를 최대 16이상에서 30으로, FTR은 최대 3이상에서 6까지 확장하는 방법을 적용하도록 개선한다.

표 7. 2배로 확장된 EI의 복잡도 및 가중치
Table 7. Complexity and Weight by Doubled Measure for EI Type

FTR	DET 갯수			
	1	2	...	30
0	낮음	낮음	...	보통
1	낮음	낮음	...	보통
2	낮음	보통	...	높음
3	보통	높음	...	높음
4	보통	높음	...	높음
5	보통	높음	...	높음
6	보통	높음	...	높음

본 연구에서 제안한 모형은 위와 같이 외부입력(EI) 유형 이외에도 다른 기능 유형에서도 같은 방식으로 세밀한 구간까지 범위를 2배로 확장하여 복잡도 및 가중치를 부여할 수 있도록 개선하였기 때문에 Micro-FP 모형이라고 명명하였다.

4.2 가중치 수치모형

표 6, 표 7을 토대로 (1 DET - 0 FTR)에서 (30 DETs - 6 FTRs)까지의 모든 경우의 수를 고려한 후 미조정기능점수를 얻어낼 수 있다. 그리고 DET 개수와 FTR 개수를 독립 변수로 설정하고, 미조정기능점수를 종속변수로 하는 다중선형회귀 분석을 통하여 아래와 같은 회귀식을 도출할 수 있다.

$$\text{Unadjusted FP(EI)} = 2.78 + (0.058 * \text{DET}) + (0.430 * \text{FTR})$$

위의 외부입력(EI) 유형과 동일한 방식으로 외부출력(EO), 외부조회(EQ), 내부논리파일(ILF), 그리고 외부연계파일(EIF) 유형의 기능유형에서 적용시켜서 표 8에서와 같은 기능유형별 기능점수 회귀식을 얻을 수 있다.

또한 Micro-FP모형에 의한 각 기능 유형별 기능점수 회귀식을 그래프 형식으로 나타내면 그림 4와 같다.

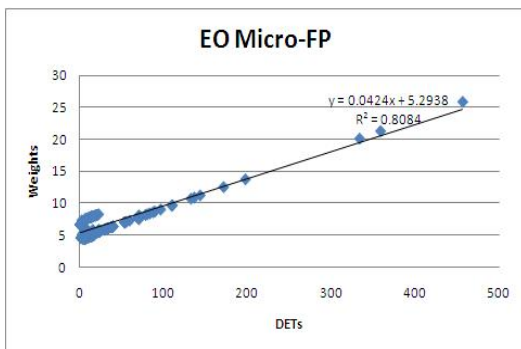
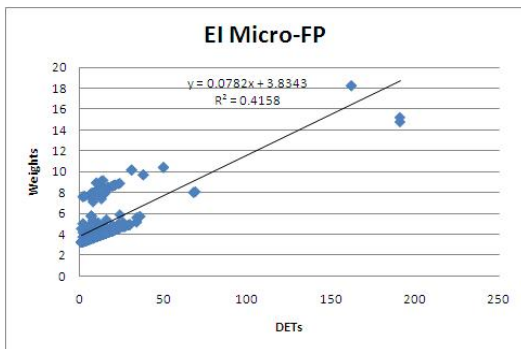
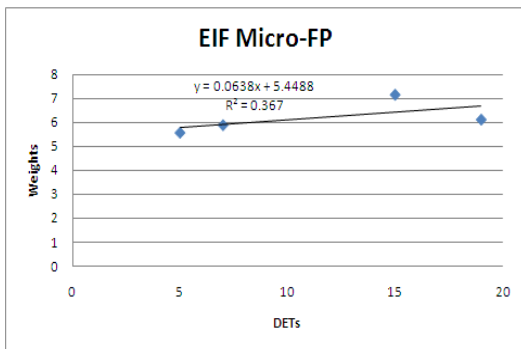
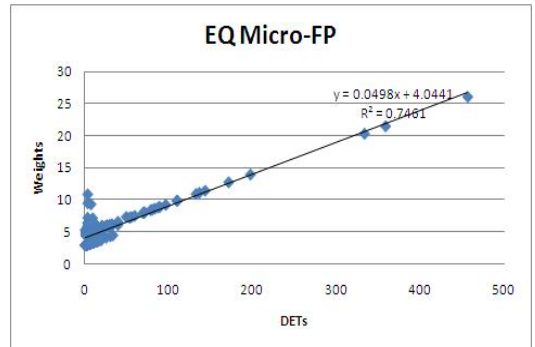
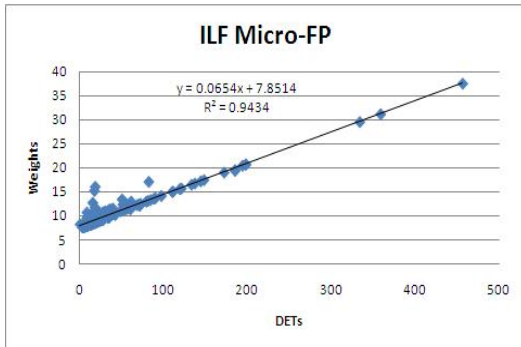


그림 4. 기능유형별 기능점수 회귀식
Fig. 4 Regression of FP by Function Type

표 8. 기능 유형 별 기능점수 회귀식
Table 8. FP Regression Equation by Each Function Types

기능 유형	기능점수 회귀식	회귀분석 통계량			
		S	R2	DF 범위	F
ILF	$7.06 + (0.0649 \cdot \text{DET}) + (0.391 \cdot \text{RET})$	0.706	66.9	2~207	20.948
EIF	$5.12 + (0.0398 \cdot \text{DET}) + (0.242 \cdot \text{RET})$	0.709	64.2	2~357	320.36
EI	$2.78 + (0.0584 \cdot \text{DET}) + (0.430 \cdot \text{FTR})$	0.709	64.2	2~357	320.36
EO	$3.91 + (0.0468 \cdot \text{DET}) + (0.301 \cdot \text{FTR})$	1.815	61.9	2~1197	971.66
EQ	$2.91 + (0.0468 \cdot \text{DET}) + (0.301 \cdot \text{FTR})$	1.114	62.0	2~1197	975.14

V. 검증 및 평가

본 연구에서 제시된 모형의 검증을 위하여 동일한 업무 도메인에 속해있는 10개의 애플리케이션에 기능점수를 산정하여 검토하였다. 기능점수 산정은 국제공인 기능점수전문가(CFPS)를 통하여 산정하였다. 첫번째 방식은 IFPUG의 산정지침을 적용하는 것으로 미조정 기능점수를 산정하였고, 두번째 방식은 본 연구에서 제안한 Micro-FP 회귀식을 적용하여 미조정 기능점수를 산정하였다. 이 두가지 방식에 의해서 산정된 소프트웨어 규모는, 표 9에서와 같이 현재 운영중인 10개 애플리케이션의 소프트웨어 규모인 LOC와 서로 상관성이 높은지를 비교함으로써 적합성을 평가하였다.

표 9. Micro-FP 모형과 LOC와의 상관관계
Table 9. Comparison of Micro-FP Model and LOC Model

시스템	LOC	IFPUG-FPA	Micro-FP
A	46,115	694	801.17
B	18,552	416	432.04
C	13,869	217	261.03
D	120,644	1,248	1593.31
E	30,837	332	357.01
F	98,778	1,635	1668.23
G	189,554	1,375	1775.68
H	146,786	1,246	1392.15
I	18,023	324	366.03
J	37,264	555	388.99

위의 결과로부터 기존 IFPUG FP 방식과 제안된 Micro FP 모형에 의한 기능점수 산정결과가, 운영중인 애플리케이션의 LOC 규모와의 상관관계를 SPSS프로그램(Ver. 12.0)을 이용하여 분석한 결과가 표 10에 나타나 있다. 분석결과, Pearson 상관계수값의 크기가 Micro-FP에 의한 기능점수와 LOC에서 0.926으로서, IFPUG FP 모형에 의한 기능점수와 LOC에서의 0.874보다 크게 나타나서, Micro-FP 모형이 우수한 결과를 보이고 있다고 판단할 수 있다.

표 10. Pearson 계수에 의한 상관분석 결과
Table 10. Correlation Analysis

	LOC	IFPUG_FPA	Micro_FP
LOC	1	.874(**)	.926(**)
IFPUG_FPA	.874(**)	1	.976(**)
Micro_FP	.926(**)	.976(**)	1

(주) SPSS프로그램 실행결과로, 수치는 Pearson 계수를 표시함. (유의수준 0.01 이하)

다음 그림 5에서는 IFPUG FP모형에 의한 기능점수와 LOC의 관계를, 그림 6에서는 Micro-FP 모형에 의한 기능점수와 LOC의 관계를 그래프로 나타낸 것이다.

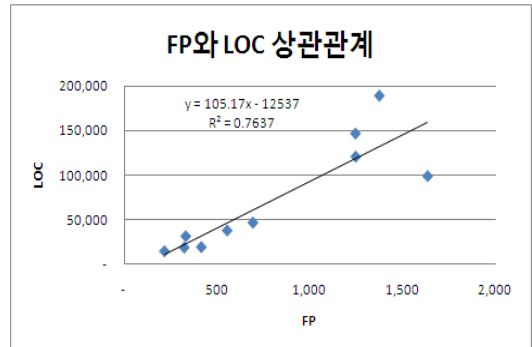


그림 5. IFPUG FPA와 LOC 상관관계
Fig. 5 Correlation of IFPUG FPA and LOC

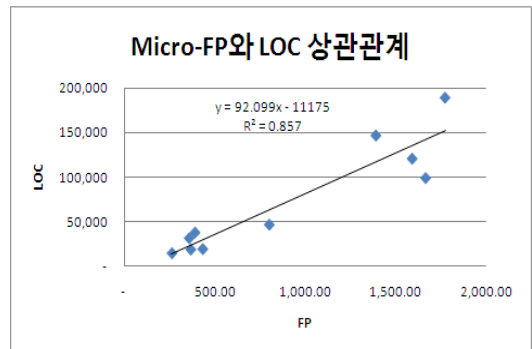


그림 6. Micro-FP 모델과 LOC 상관관계
Fig. 6 Correlation of Micro-FP and LOC

VI. 결론 및 향후 과제

FPA를 통한 소프트웨어 건적은 구현기술, 언어, 플랫폼과 독립적으로 규모를 산정할 수 있으며 사용자 언어로 요청된 사항을 규모로 산정하기 때문에 프로젝트 초기에 비교적 구체적인 요구사항을 도출할 수 있다는 것과 널리 쓰이는 국제 표준을 준용한 측정 방식이기 때문에 국가간 소프트웨어 경쟁력 비교가 가능하다는 것이 장점으로 부각되고 있다. 그러나 사용자 관점에서 오는 측정기준의 모호성과 주관적 복잡도 매트릭스의 적용문제는 논란의 대상이 되어왔다.

이에 본 논문에서는 기존에 제시된 이슈를 분석하여 그 해결책으로 Micro-FP 모델을 제시하였다. 제시한 모델은 함수식을 통하여 기존의 방식보다 세밀한 기능점수를 얻을 수 있는 장점이 있다. 또한 평가 결과 기존의 FPA 방식을 대체할 수 있음이 입증되었으며, 특히 복잡도의 상한이 존재함으로써 Real-Time, Embedded, 과학/공학분야 등의 애플리케이션

과 일부 복잡한 기능을 포함하고 있는 애플리케이션에서 적용하기 어려웠던 기존 FPA의 문제점을 해결할 수 있을 것으로 평가된다.

본 논문에서 제안하는 개선된 Micro-FP 기능점수 방식은 회귀식을 이용하므로 측정에서 용이한 장점을 가지면서도, 시스템 규모가 복잡화, 대형화되는 추세를 고려하여 더욱 정교하고 확장성을 갖춘 장점을 활용할 수 있게 되었는데, 다만 제한된 조직의 데이터를 통한 결과이므로 본 논문에서는 제안하는 모델의 성능 평가에만 그 의의를 두고자 한다. 그러나 애플리케이션을 운영하는 사용자의 조직에서는 이와 같은 기법을 적용하면 본 연구에서 제안하는 모형의 장점과 적용효과를 그대로 기대할 수 있다.

참고문헌

- [1] 정인용 외 3인, "소프트웨어 규모 산정을 위한 개선된 기능 점수 측정 모델," 한국인터넷정보학회, 인터넷정보학회논문지, 제10권 제4호, 115-126쪽, 2009년 8월.
- [2] Albrecht, A. J, "Measuring Application Development Productivity," Proceedings of Joint SHARE, GUIDE, and IBM Application Development Symposium, IBM Corp., Monterey, CA, pp. 83-92, Oct. 1979.
- [3] Demirors, O. and Gencel, C., "Conceptual Association of Functional Size Measurement Methods," Software, IEEE, Vol. 26, No. 3, pp. 71-78, May-June 2009.
- [4] 안연식, "소프트웨어유지보수 프로젝트의 투입인력 규모 예측 모형," 한국컴퓨터정보학회논문지, 제4권, 제2호, 61-69쪽, 1999년 6월.
- [5] 이원호, "IT프로젝트에서 개발방법이 기능점수에 미치는 영향: 신규개발과 보강개발을 중심으로," 한국컴퓨터정보학회논문지, 제14권, 제4호, 131-139쪽, 2009년 4월.
- [6] Abran, A., J. M. Desharnais and F. Azziz, "Measurement Convertibility: From Function Points to COSMIC," 15th International Workshop on Software Measurement(IWSM 2005), Montr éal (Canada), Shaker Verlag, Vol. 12-14, pp. 227-240, Sept. 2005.
- [7] Gencel, C., "How to Use COSMIC Functional Size in Effort Estimation Models?," Springer Berlin: Heidelberg, Vol. 5338, pp. 196-207, Nov. 2008.
- [8] Molokken, K., and Jorgensen, M., "A Review of Software Surveys on Software Effort Estimation," In International Symposium of Empirical Software Engineering, 2003.
- [9] NESMA, "Functional Sizing in Contemporary Environments: Introduction of a Functional Sizing Reference Model," NESMA Workgroup NT, Oct. 2002.
- [10] 정보통신부, "소프트웨어 사업대가 기준," 정보통신부 고시, 제 2004-8호, 2004.
- [11] IBM CIS & a Guidelines 313, "AD/M Productivity Measurement and Estimate Validation," New York, 1984.
- [12] IFPUG, "Function Point Counting Practices Manual, version 4.2," International Function Point Users Group, 2004.
- [13] IFPUG, "Function Point Counting Practices Manual R4.2.1," International Function Point Users Group, 2006.
- [14] NESMA, "Definitions and Counting Guidelines for the Application of Function Point Analysis: A Practical Manual, version 2.2," Netherlands Software Measurement user Association, 2004.
- [15] Symons, Charles, "Function Point Analysis-Difficulties and Improvements," IEEE Transactions on Software Engineering, Vol. 14, No. 1, pp. 2-11, 1988.
- [16] Al-Hajri, M. A., A. A. A. Ghani, M. S. Sulaiman, and M. H. Selamat, "Modification of standard function point complexity weights system," Journal of Systems and Software, Vol. 74, pp. 195 - 206, 2005.
- [17] Jorgensen, M. "Evidence-based Guidelines for Assessment of Software Development Cost Uncertainty," IEEE transactions on software engineering, Vol. 31, No. 11, pp. 942 - 954, 2005.
- [18] Jorgensen, M., "A Review of Studies on Expert Estimation of Software Development Effort," Journal of Systems and Software, Vol. 70, No. 1-2, pp. 37 - 60, 2004.
- [19] Xia W., L. F. Capretz, D. Ho, and F. Ahmed,"

A New Calibration for Function Point Complexity Weights," *Information and Software Technology*, Vol. 50, Issues 7-8, pp. 670-683, June 2008.

- [20] Fenton, N. E., Pfleeger, S. L., "*Software Metrics A Rigorous & Practical Approach, -second ed-*" PWS, Boston. 1997.
- [21] 허승현, 최은만, "동적 웹 어플리케이션의 특성을 반영한 조정 기능 점수 산정 방안," 한국정보과학회 학술발표논문집, 제31권, 제2호, 355-357쪽, 2004.
- [22] Heeringen, H. S. Van, "Changing from FPA to COSMIC: A Transition Framework," *RPM-AEMES*, Vol. 4, pp. 104-115, 2007.

저 자 소개



안 연 식

1982: 전북대학교 이학사

1989: 연세대학교 공학석사

2002: 국민대학교 경영정보학박사

현 재: 경원대학교 경영학과 부교수

관심분야: 기술경영, 정보시스템평가 등