

# S/W 규모산정을 위한 개선된 FP 모델

조은숙\* 김영갑\*\*

\*고려대학교 컴퓨터과학기술대학원 소프트웨어공학과

\*\* 고려대학교 컴퓨터학과 소프트웨어 시스템 연구실

e-mail : magaretj@nate.com, ygkim@software.koera.ac.kr

## Improved Function Point Model for S/W Size Estimation

Eun-Sook Jo\* Young-Gab Kim\*\*

\*Dept. of S/W Engineering, Graduate School of Computer Science & Technology, Korea University

\*\* Software System Lab., Dept. of Computer Science & Engineering, Korea University

### 요 약

성공적인 소프트웨어 프로젝트를 목표로 합리적인 계획을 수립하기 위하여 프로젝트의 전체 규모에 대한 신속하고 정확한 견적이 중요해지고 있다. IFPUG의 FPA 견적방식은 이미 검증되어 널리 사용되고 있으나, 복잡성과 모호성으로 인하여 실제 SI 업체의 견적 수행에는 어려움이 있으며 견적작업의 고비용을 초래하고 있다. 본 논문은 기존 FPA 방식의 문제점을 지적하고 이를 개선한 PFP 모델을 제시함으로써 기존보다 쉽고 효과적이며 효율적인 기능점수 견적을 가능하게 하였다.

### 1. 서론

성공적인 소프트웨어 개발 사업의 목표는 고객만족과 경쟁력 확보이다. 높은 수준의 고객 만족과 사업 경쟁력 확보를 위해 고품질의 소프트웨어를 주어진 예산 내에서 주어진 납기일에 맞추어 개발할 수 있는가, 즉 On-Time, On-Budget의 여부는 오랫동안 소프트웨어 관련 업계의 최대 이슈가 되어왔다[1].

이러한 이슈를 해결하기 위한 방법으로 보다 정확한 소프트웨어 개발 규모와 비용, 일정에 대한 견적 모델이 연구되어 왔으며, 또 다른 접근 방법으로써 새로운 프로그래밍 언어와 객체지향-CBD 등과 같은 새로운 방법론 적용과 같은 기술적 방법이 제시되었다. 최근에는 조직의 소프트웨어 개발 절차와 방법에 대해 체계적인 프로세스를 수립함으로써 효과적인 고품질의 소프트웨어를 개발할 수 있다는 가정에서 출발하는 소프트웨어 프로세스 개선 방법(SPI: Software Process Improvement)이 많이 시도되고 있다[2].

기술적인 접근이나 프로세스적인 접근이나 성공적인 소프트웨어 개발 프로젝트를 위해서는 개발하고자 하는 대상 시스템의 사용자 요구사항이 무엇인지 구체적으로 파악하여 범위를 확정하고 계획을 수립하는 일이 선행되어야 하며, CMM, CMMI, SPICE와 같은 소프트웨어 프로세스 모델에서는 사용자의 요구사항을

수치적으로 계량화하여 계획과 실행의 차이를 추적 분석하고 적시에 소프트웨어 개발 계획을 변경 및 통제하도록 강조하고 있다. 사용자의 요구사항을 정량적으로 파악하고 합리적인 소프트웨어 개발 계획을 수립한 후, 진행 중 실적과의 차이를 추적하기 위해서는 소프트웨어 프로젝트의 전체 규모와 현재 진행된 상태를 한눈에 보여줄 수 있는 규모(Size)의 개념이 필요하다[3][4][5][6].

규모 개념을 도입하여 프로젝트를 견적하기 위한 노력의 결과로 최근에는 기능점수분석(FPA: Function Point Analysis) 방식이 많이 사용되고 있으며, IFPUG, COSMIC, MK-II, NESMA 등 여러 방법이 있지만 그 중에서도 IFPUG(International Function Point Users Group)에서 제안한 FPA 모델이 가장 많이 사용되고 있다 [7][9]. 정보통신부에서는 공공발주 소프트웨어 사업의 대가 산정 시 IFPUG의 FPA 방식을 이용하도록 관련 법[9]을 개정하였다. 이에 관련 SI 업계에서는 서둘러 공인기능점수전문가(CFPS: Certified Function Point Specialist)를 확보하여 FPA를 이용한 소프트웨어 프로젝트 규모 견적을 시행코자 노력하고 있으나, 경험의 부족과 FPA 방식의 어플리케이션 경계 설정 기준의 모호성 및 가중치 계산방법의 복잡성 등으로 인해 측정작업이 비효율적이고 산정자 간 편차가 크게 나타나는 등 FPA 방식의 견적에 어려움을 겪고 있다.

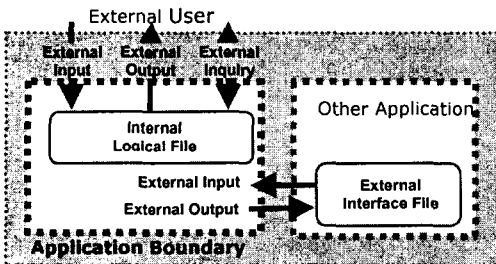
본 논문에서는 관련 연구를 통해 IFPUG의 FPA 방식의 문제점을 구체적으로 지적하고, 이를 극복하고자 하는 노력으로써 기존 방식을 개선한 Plain Function Point(PFP) 모델을 제시한다. PFP 모델을 이용한 견적 수행 결과를 기존 모델과 비교하여 PFP가 기존 모델을 대체하여 보다 효과적·효율적인 견적을 가능하게 하는지 보여주고자 한다.

2. 관련 연구

2.1. IFPUG's FPA

FPA는 사용자 관점에서의 기능적인 크기를 측정하는 것으로써 1980년대 초 IBM의 Alan Albrecht가 최초 고안한 이후 IFPUG의 노력에 의해 가장 보편화된 규모 견적 방법으로 정착되었다. IFPUG는 1990년부터 Counting Practices Manual(CPM)을 주기적으로 발행하여 FPA에 대한 가이드라인을 향상시키고 구체적인 사례를 제공하고 있으며, 최근 2004년 4월에는 CPM4.2를 릴리즈하였다.

기능점수는 크게 데이터 구조와 이를 처리하는 트랜잭션 기능 점수로 구성되어 있으며, 각각은 ILF·EIF, EI·EO·EQ의 5가지 타입으로 다시 세분화되어 산정하게 되어 있다. 어플리케이션에 포함되어 있는 각각의 기능 타입을 정의하기 위해서는 먼저 어플리케이션 경계(Application Boundary)가 설정되어야 하는데, 이는 사용자 관점에서 바라보는 어플리케이션 내부(Internal)와 외부(External) 구분(Border)을 의미하며 이 구분을 바탕으로 하여 어플리케이션 내부에서 유지보수되는 논리적 파일을 ILF(Internal Logical File)라 하고 외부 시스템으로부터 참조만 되는 논리적 파일을 EIF(External Interface File)라고 한다. 또한 어플리케이션 경계를 넘나드는 트랜잭션 기능 타입은 입력, 출력, 단순 조회 등의 고유 목적에 따라 각각 EI(External Input), EO(External Output), EQ(External InQuery)로 정의된다[10]. 그림 1은 이와 같은 기능점수 타입의 정의를 잘 나타내고 있다.



[그림 1] 기능점수 측정 도해

각 기능 타입별로는 각각의 가중치를 갖는데 다음과 같이 ILF·EIF의 경우에는 DET(Data Element Type)과 RET(Record Element Type)의 개수에 따라, EI·EO·EQ의 경우에는 DET와 FTR(File Type Referenced)의 개수에 따라 내부복잡도를 구하고 이에 해당하는 가중치 값을 계산하도록 되어 있다.(표 1,2 참조)

타입	RET	DET		
		1-19	20-50	51+

ILF/EIF	1	Low	Low	Avg	
	2-5	Low	Avg	High	
	6+	Avg	High	High	
타입	FTR	1-4	5-15	16+	
	EI	0-1	Low	Low	Avg
		2	Low	Avg	High
3+		Avg	High	High	
타입	FTR	1-5	6-19	20+	
	EO/EQ	0-1	Low	Low	Avg
		2-3	Low	Avg	High
4+		Avg	High	High	

[표 1] 각 기능 타입별 복잡도

타입	복잡도		
	Low	Avg	High
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

[표 2] 각 기능 타입의 복잡도별 가중치

FPA는 프로그램 언어나 플랫폼 등 물리적 특성 상관없이 기술 독립적이며 사용자 관점에서 논리적인 기능의 크기를 동일한 척도로 측정 가능하고 소프트웨어 개발 초기 단계에서 사용자 요구사항만으로도 견적이 가능하다는 점에서 기존의 견적 모델보다 우수하다고 평가받고 있으며[7][11], 미국·영국·호주·브라질·일본·캐나다 등을 비롯한 세계 각국에서 널리 채택되고 있다[8].

2.2. FPA 방식의 문제점

2.2.1. 어플리케이션 경계의 모호성

CPM에서는 어플리케이션 경계에 대해 “측정 대상 소프트웨어와 사용자간의 경계(border)”라고 정의하며, 이는 어플리케이션과 사용자와의 경계뿐만 아니라 어플리케이션과 어플리케이션 사이의 경계를 모두 지칭한다. 개발자가 아닌 사용자 중심의 관점에서 결정되어야 하며, 측정 대상 소프트웨어의 범위(Scope) 내에서 하나 이상이 될 수 있지만 처음에 설정된 경계는 그대로 유지되어야 하며 변경되지 않도록 가이드하고 있다[10].

CPM의 정의는 사용자 관점(User View)을 강조함으로써 개발자나 기술적 의존성을 배제하도록 하였으나, 어플리케이션 경계의 정의가 모호하여 사용자들 사이에서도 적지 않은 혼돈을 초래하고 있으며, 어플리케이션들간의 경계 설정을 어떻게 하느냐에 따른 ILF와 EIF의 산정 오차가 크게 발생한다는 점에서 논란이 되고 있다[12]. 예를 들어 시스템 S가 서브시스템 A1, A2, B로 구성되어 있을 때, 사용자 값은 어플리케이션 경계를 A1, A2, B간의 경계로 정하고, 사용자 값은 A1과 A2를 하나의 어플리케이션 영역인 A로 정의하여 A와 B사이의 경계만을 취할 수 있다. 이 경우 값이 계산한 시스템 S의 FP 합과 같이 계산한 FP 합은 같아질 수 없는 모순이 발생하게 된다.

2.2.2. 측정 비효율성

각 타입별로 3개 값으로 대표되는 가중치를 얻기 위해서는 표 1에서와 같이 DET와 RET/FTR 개수를

구한 다음 각 개수가 속하는 그룹에 따라 Low, Average, High 의 내부복잡도를 구하고 이를 표 2 와 같이 전환하는 과정을 거쳐야 한다. 즉, 기능 하나의 가중치를 구하기 위해서 많은 계산을 소비해야 하며, 결국 FPA 를 활용한 건적은 시간이 오래 걸리는 고비용의 작업이 될 수 밖에 없다. CFPS 가 측정 작업에 참여할 경우 FP 산정 시 발생하는 논란을 최소화 하고 산정자별 편차를 줄이는 효과는 있지만 건적에 소요되는 비용을 줄이지는 못한다. 표 3 은 SI 조직의 프로젝트에 대하여 FP 측정 그룹을 CFPS, 숙련자, 비숙련자로 나누어 FP 건적을 실시했을 때 그룹별로 1 일 측정량의 크기와 2,400FP 규모 정도의 프로젝트를 견적하는 데 필요한 총 소요시간을 계산한 것이다. 2,400FP 정도 규모를 가진 프로젝트를 견적하기 위해서는 산정자 역량에 따라 4~10 일 이상의 시간이 소요 되는 것으로 나타나는데, 보통 1~2 주 안에 수행되어야 하는 프로젝트 초기 계획 수립 작업에 상당한 부담이 따름을 보여 준다.

구분	단위	CFPS	숙련자	비숙련자
1 일측정량 (8 시간)	FP	430~520	410~670	220~430
총소요시간 (2,400 FP)	Day	4.6~5.6	3.6~5.8	4.6~10.9

[표 3] 그룹별 FP 측정 결과 비교

2.2.3. FPA 의 구조적 약점(Structural Weakness)

FPA 가 IFPUG 의 노력에 의해 지속적으로 향상되어 보편화되고 있음에도 불구하고, 이미 90 년대 초 Alain Abran 과 Pierre N. Robillard 이 지적했던 바와 같이 FPA 방식은 구조적인 약점을 아직 극복하지 못하고 있다. 이는 각 기능 타입별로 제시된 내부 복잡도에 따라 가중치를 구하는 방법이 수학적으로 풀이되는 견고한 알고리즘을 갖고 있지 못한 데서 기인한다. High, Average, Low 등급별로 결정되는 가중치는 FPA 고안 당시 Albrecht 가 임의로 정한 것이며[12], 각 단위 기능(Elementary Function)마다 복잡도를 결정하기 위해 사용되는 DET, RET/FTR 형렬(표 1,2 참조)에서의 Scale 표현 단위는 Nominal, Ordinal, Interval 과 같은 어떠한 수학적 규칙도 적용되지 않는다[13].

3. Plain Function Point(PFP)

3.1. PFP 모델 제안

PFP 모델은 위에서 지적한 기존 FPA 방식의 문제점을 해결하기 위하여 다음의 3 가지 원칙을 적용하였다.

- ① 어플리케이션 경계의 명료화
- ② 내부복잡도 계산의 단순화
- ③ 구조적 알고리즘의 사용

이 외에 나머지 규칙은 IFPUG 의 FPA 방식을 그대로 따르도록 한다.

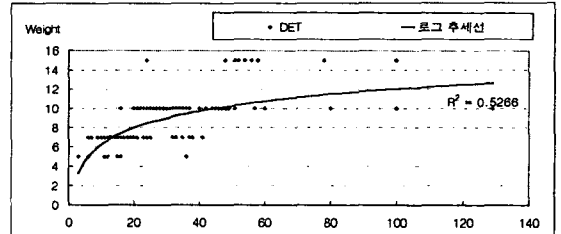
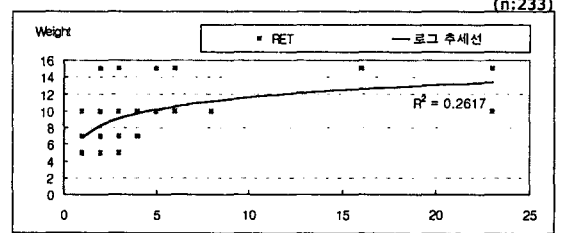
3.1.1. 어플리케이션 경계의 명료화

먼저 논란이 되고 있는 어플리케이션 경계를 시스템과 사용자간의 경계로만 한정하고 어플리케이션들간의 경계는 구분하지 않도록 한다. 이로써 어플리케이션 간의 경계 설정에 따르는 혼란과 FP 합계의 불

일치를 방지할 수 있다. 어플리케이션 경계를 시스템-사용자 경계로 국한할 경우, 측정 대상이 되는 시스템 내에서 어플리케이션 간에 참조되는 EIF 는 산정대상에 포함되지 않으나, 측정 대상 시스템이 유지보수하지 않는 외부로부터 인터페이스되는 파일, 즉 예를 들어 CMS 를 통한 EDI 파일과 같은 경우는 EIF 로써 계속 유지된다.

3.1.2. 내부복잡도 계산의 단순화

ILF 와 EIF 의 내부복잡도를 계산할 때 식별하기가 어려우며, 산정하기 위해 필요한 자료를 수집하는 데 오래 걸리는 부분이 RET 의 측정이다. 마찬가지로 EI EO EQ 의 경우에도 각 기능이 참조하는 FTR 의 개수를 측정하는 데에 상대적으로 오랜 시간이 소요된다. 반면에 그림 2 와 같이 내부복잡도에 대한 RET 결정력은 오히려 작으며, FTR 도 마찬가지이다.



[그림 2] DET 와 RET 의 가중치 결정력 비교

이러한 점에 착안하여 PFP 에서는 RET/FTR 를 산정에 포함하지 않고 단지 DET 개수만으로 내부복잡도를 결정하게 함으로써 측정의 효율성을 높이며 RET/FTR 식별 차이에 따른 논란을 방지할 수 있다.

3.1.3. 구조적 알고리즘 사용

어플리케이션 경계와 내부복잡도 계산 방식을 단순화하고 기존의 비구조적 가중치 계산 방식을 구조적 방식으로 개선하여, 표 4 와 같이 DET 만을 독립변수로 사용하는 가중치 알고리즘을 가정할 수 있다. 알고리즘의 형태는 여러 가지 방법을 시도한 결과 가중치에 대해 결정계수가 가장 높은 로그 형태를 취하였다.

$$W_{PFP} = f_{PFP}(DET) = a + LOG_b DET$$

\* a=보정계수, b=기능타입별 상수.

[표 4] PFP 가중치 모형

여기서 a, b 의 값은 새로운 회귀분석을 통해 추출되어야 하는 계수이다. 다음 3.2 장에서 회귀분석을 사용하여 a, b 값을 추정하여 PFP 모델을 완성하고, 3.3 장에서는 결정된 PFP 모델을 이용하여 프로젝트 규모

를 산정한 결과와 기존 방식의 결과를 비교하여 PFP 방식의 적합성을 검증하도록 한다.

3.2. PFP 모형의 계수 결정

기존 FPA 방식에 의한 FP 산정 결과는 조직 내에서 차기 프로젝트 견적을 위해 사용되어야 하는 Data 들이다. 즉, PFP 모형에 의해 계산식이 바뀌더라도 결과 값은 기존 방식에 의한 결과값의 분포에 유의하여야 한다. 따라서 기존 가중치를 종속변수 Y 로 두고 PFP 가중치 모형 수식을 종속변수인 X 값으로 두어 0.5% 유의수준에서 회귀모형을 가정한다.

- Y 값 : 기존 FP 방식에 의한 가중치 결과
- X 값 :  $f_{PFP}$  (DET)에 의한 가중치 결과

회귀모형을 완성하기 위한 b 값은 각각의 기능타입 별로 여러 회 시뮬레이션하여 다음과 같이 결정하였다. EIF의 경우에는 어플리케이션 경계 설정 정의의 변경으로 인해 개수가 많이 감소되므로 대신에 오히려 가중치가 높게 나오도록 도출하였다(b가 작을수록 가중치는 커짐).

보정계수인 a 는 회귀모형에서 산정하고자 하는 결정계수에 크게 영향을 미치지 않는 상수이므로 0으로 가정하여 적용하였다.

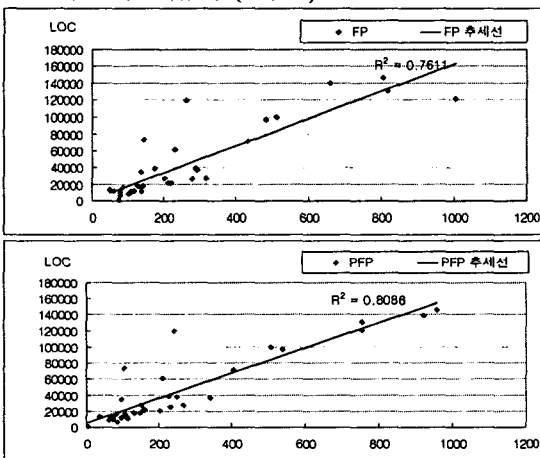
ILF	EIF	EI	EO	EQ
1.35	1.28	1.65	1.64	1.71

[표 5] PFP 가중치 모형의 기능타입별 계수(b)

4. 검증 및 기대효과

PFP 모형의 적합성과 효과성을 분석하기 위하여 기존 FPA 방식과 PFP 방식의 상관성과 견적 일일 측정량의 두 가지 측면에서 검증하였다.

먼저 PFP 방식이 기존 FPA 방식을 대체할 수 있는지 보기 위하여 동일한 프로그램 언어를 사용한 4개의 프로젝트에서 39개의 어플리케이션을 선정하여 각 어플리케이션별 기능점수 합계와 LOC(Line of Code)와 상관관계를 비교하였다. (그림 3).



[그림 3] 기존 FP와 PFP 모형의 LOC 상관관계 비교

그림 3의 상단 그래프에서도 볼 수 있듯이 보통 어플리케이션별로 기능점수와 LOC는 상관관계를 갖고 있는데, PFP 모형을 사용할 경우 LOC에 대한 상관관계가 계속 유지되며 이는 PFP 모델이 기존 방식을 대체할 수 있음을 보여주는 것이다.

다음의 측정 결과는 CFPS와 숙련자의 FP 견적시 1일 측정량을 비교한 것이다.

구분	방식	CFPS	숙련자
1일 측정량 (8시간)	FP	430~520	410~670
	PFP	550~610	540~620

[표 6] 기존 FP와 PFP 방식의 일일 측정량 비교

PFP 모형을 사용한 견적 결과에서는 CFPS와 숙련자 간에 측정량의 편차가 적고, 1일 측정량에 있어서는 기존 방식보다 높게 나타난다. 이는 산정자의 역량에 구애받지 않으며, 수행 시간 단축으로 인해 견적 비용을 감소시킬 수 있다는 측면에서 PFP 모델의 효율성·효과성을 입증한다.

5. 결론

IFPUG의 FP 견적방법은 프로젝트 초기 단계에서 활용 가능하며 사용 언어와 플랫폼에 독립적인 장점에 불구하고, 수행 기준의 복잡성과 모호성으로 인하여 전문성이 요구되는 등 견적이 어렵고 고비용을 초래한다.

이에 본 논문에서는 견적 비용을 최소화하고 고도의 지식이 요구되지 않는 간결하고 명료한 방법으로 견적을 수행할 수 있으며 기존 방식의 구조적인 약점을 해결한 PFP 모형을 제시하였다. 새로운 PFP 모형을 적용하여 프로젝트 규모 견적을 실시한 결과 PFP 모델이 기존 FPA 방식을 대체 가능하며 효율적·효과적 측면에서 보다 나아진 견적 모델임을 검증하였다.

참고문헌

- [1] Roger S. Pressman, "Software Engineering - A practitioner's Approach", McGraw Hill, 2001
- [2] 유재구, 이은서, 장윤정, 이경환, "SPI를 위한 개선 항목과 프로세스와의 연관성 연구", 2003
- [3] PMI Standards Committee, "A Guide to the Project Management Body of Knowledge", 2000
- [4] Mark C Paulk, Charles V.Weber, Bill Curtis, "The Capability Maturity Model", 1995
- [5] Mary Beth Chrissis, "CMMI: Guidelines for Process Integration and Product Improvement", Addison Wesley, 2003
- [6] KSPICE, "SPICE Assessments in Korea", 2002
- [7] Hans van Vliet, "Software Engineering Principles and Practice 2nd Ed.", John Wiley & Sons, 2000
- [8] 이현욱, 정화자 외 7명, "소프트웨어 사업대가기준 교육자료 개발", 한국전산원, 2002
- [9] "소프트웨어 사업대가 기준", 정보통신부 고시 제 2004-8호, 2004.2
- [10] IFPUG, "Function Point Counting Practices Manual R4.1.1", 2000
- [11] D. Garmus, D. Herron, 조춘식 역, "기능점수분석", Addison & Wesley, 2002
- [12] Ray Boehm, "Function Point FAQ", <http://ourworld.compuserve.com/homepages/softcomp/fpfaq.htm>, 1997
- [13] Alain Abran, Pierre N.Robillard, "Identification of the structural weaknesses of Function Point metrics", 1991