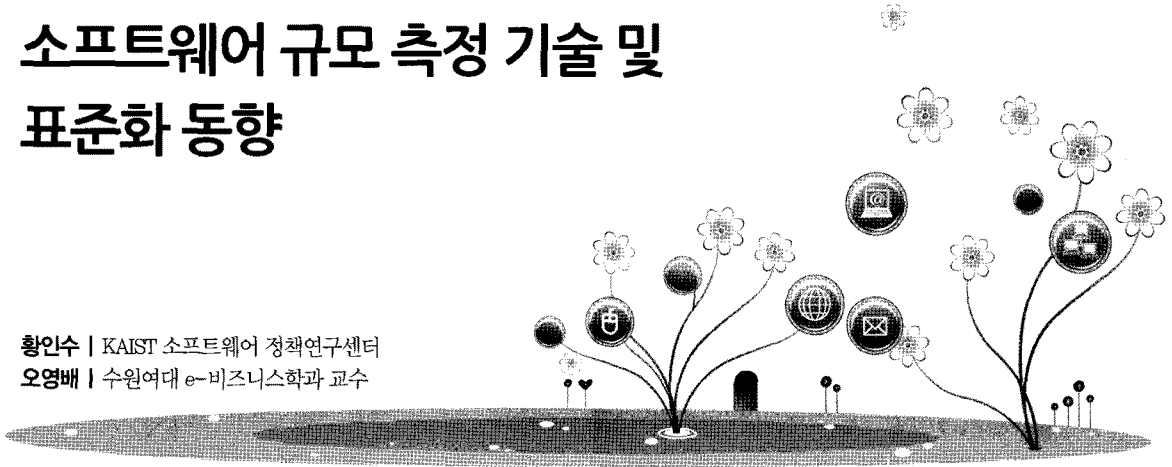


# 소프트웨어 규모 측정 기술 및 표준화 동향

황인수 | KAIST 소프트웨어 정책연구센터

오영배 | 수원여대 e-비즈니스학과 교수



## 1. 머리말

소프트웨어 규모를 알고자 하는 이유는 기술적인 측면보다 관리적인 측면이 크다. 자원과 시간이 얼마나 소요될지를 알아야 예산을 할당할 수 있고 계획을 세울 수 있기 때문이며, 개인은 자신의 시간을 예측할 수 있고 조직은 인력을 체계적으로 관리하는 방법을 강구할 수 있기 때문이다. 이러한 이유에서 규모를 알고자 한다면 아직 개발되지 않은 소프트웨어 규모를 추정해야 하는 것으로 기준 척도를 사용해 실물을 측정할 수 없기에 무언가 명확한 단서를 잡아 예상되는 실제 규모를 추정할 수 있는 방법과 기법을 필요로 하게 된다. 소프트웨어 규모를 알고자 하는 또 다른 이유는 개발자들이 얼마나 성과를 냈는지 알려는 것으로 계획의 차질 또는 초과와 원인분석 결과를 조직원들이 학습을 통해 깨달아 프로세스 개선의 기회를 갖게 할 뿐만 아니라 예측의 정확성을 개선하고자 하려는 것이다. 이러한 이유에서 규모를 알고자 한다면 이미 개발된 소프트웨어 규모를 측정해야 하는 것으로, 기준 척도를 사용해 실물을 직접 측정하면 된다. 이외에 프로젝트 수행 중 진척 정도와 남은 업무량을 알고자 하여 이미

개발한 것에 대한 측정과 개발할 것에 대한 추정을 동시에 하는 것이다. 프로젝트 수행 중 측정과 추정은 앞의 두 가지 경우를 모두 필요로 하는 경우이기에 두 경우에 사용하는 방법과 기법을 그대로 사용하면 된다. 추정이든 측정이든 같은 대상에 대한 수행 시점의 차이만 있을 뿐 측정 기법에 있어서는 근본적인 차이가 있을 수 없다. 다만 실물에 대한 것이냐 그렇지 않느냐가 다를 뿐이므로 굳이 따지자면, 추정에는 대상에 대한 전문 지식과 함께 측정 기술을 필요로 하지만, 측정에는 전문 지식은 부족해도 측정하는 기술만 있으면 된다는 점이 다르다. 그러나 추정은 측정을 바탕으로 이루어져야 타당성과 합리성을 확보할 수 있기에 측정이 없는 추정은 근거 없는 주장이 될 수 있고 소수의 전문가에 의한 직관에 불과하다는 평가를 받을 수 있다. 따라서 명확하고 누구나 인정할 수 있는 객관적인 기준에 따라 측정해야 가시성의 확보가 절박한 소프트웨어 프로젝트 관리도 정상화 되는 길이 열릴 것이다.

우리나라 소프트웨어 사업대가기준은 사업의 규모를 측정하는 것이 대가산정의 대부분을 차지하고 있다. 규모에 단가를 곱하면 보정 전 개발원가가 되고 여기에 규모, 언어, 형태, 품질 및 특성 등 네 가지의 보

정계수만을 적용하면 개발원가를 구할 수 있는 구도로 되어 있기 때문이다. 핵심은 어떻게 아직 시작도 하지 않은 사업의 규모를 측정할 수 있는지의 문제인데 이것이 바로 측정이 아닌 추정이기 때문이다. 안타까운 현실은 추정을 하면서도 측정을 하는 것으로 착각하고 대부분의 발주자들은 측정하는 방법을 왜 이리 어렵게 만들었느냐고 불평을 한다는 것이다. 사실은 측정이 어려운 것이 아니라 측정의 대상을 명확하게 정의하지 못하는데 있고 근본적으로는 측정할 대상물이 실물로 존재하지 않는다는 데 있다.

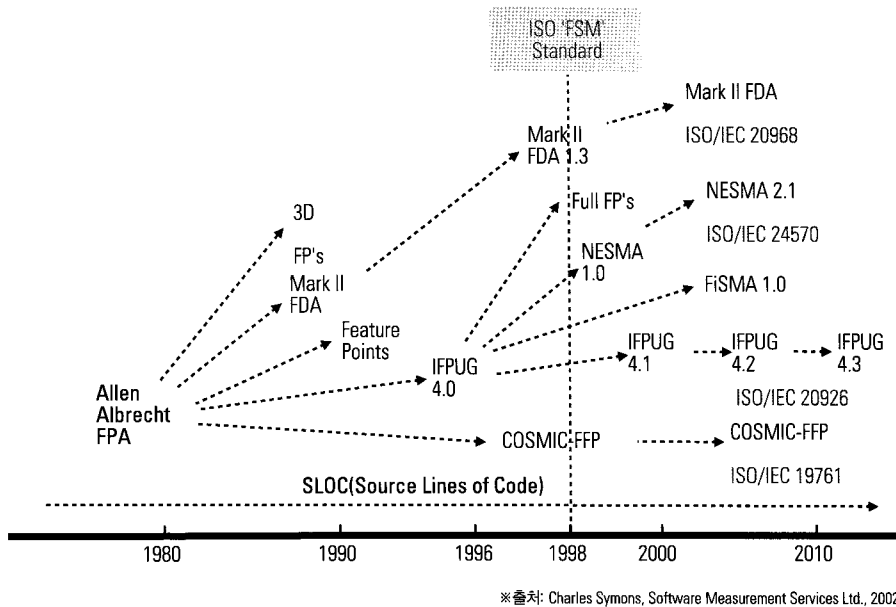
## 2. 소프트웨어 규모 산정 기술 동향

### 2.1 해외 기술동향

소프트웨어 규모 측정은 소프트웨어가 출현한 이후 곧바로 시작되었다 해도 과언이 아닐 것이다. 개발자라면 누구나 자신이 개발한 프로그램 소스의 길이가 얼마나 되는지 궁금할 것이기 때문이다. 당연히 코드라인수(LOC: Line of Code) 방식이 유일한 소프트웨어 규모 측정의 기준일 수밖에 없었다. 문제는 소프트웨어 규모가 커지고 복잡해지면서, 나아가 소프트웨어에 대한 사용자의 요구가 불명확할 뿐만 아니라 변경이 자주 발생하면서 추정이 매우 어렵게 되었다는 데 있다. IBM이 시스템 360 운영체제를 만들면서 겪은 프로젝트 실패 경험은 소프트웨어 규모 추정이 얼마나 어려운지, 그래서 소프트웨어 개발 프로젝트의 정확한 계획수립도 성공적인 관리도 쉽지 않다는 사실을 통감하게 해준 사례였다. 지금도 소프트웨어 프로젝트의 성공률은 스탠디쉬(Standish) 조사에 의하면 30%대를 웃돌지 못하고 있다. 국내의 경우는 더욱 심각해 성공률이 20%대를 밑돌고 있는 것으로 지난 해 소프트웨어사업 선진화 포럼의 조사결과 드러났다. 한 마디로 코드라인수에 의한 단순 소프트웨어 규모 추정은

기술이 복잡하고 업무 규모가 큰 프로젝트에서는 도음이 되지 못한다는 평가였다. 이 때부터 본격적으로 보다 객관적인 그러면서도 개발기술과는 무관하게 그것도 사업초기에도 적용할 수 있는 척도를 찾기 시작했고 그 결과 출현한 것이 1979년 IBM의 알브레히트(Allan Albrecht)가 발표한 기능점수분석(Function Point Analysis)법이다. 알브레히트는 사업초기에도 소프트웨어 규모 측정이 가능한 방법을 찾고자 측정의 초점을 개발자 관점에서 사용자 관점으로 옮기고 사용자가 개발자에게 어떠한 소프트웨어 기능을 요구했는지를 측정할 수 있는 방법을 고민했던 것이다. 이러한 알브레히트의 생각은 지금도 존중을 받아 모든 기능점수 측정 방법의 본질이 되었고 보다 편리하고 쉬운 측정방법을 찾기 위한 다양한 노력이 지속적으로 이루어져 오늘날에는 그 종류가 몇 개인지조차 헤아릴 수 없을 정도로 많아지게 되었다. 찰스 사이먼(Charless Symons)에 의하면 소프트웨어 기능 규모 측정 방법의 발전은 크게 코드라인수 측정법과 기능점수법으로 양분되어 발전해 왔다.[그림 1]

코드라인수는 IEEE에서 자기 완결적인 데이터 선언과 처리로직 등 실행에 의미를 부여하는 코드라인 중심으로 측정할 것을 권장하고 있을 뿐 다른 여타의 조건이나 제약사항을 포함하지는 않는다. 따라서 코드라인수는 각기 다른 언어별로 의미를 가질 뿐 서로 다른 언어 간에는 비교의 대상이 될 수 없다. 반면 기능점수는 적용 도메인에 따라 구현되는 소프트웨어의 특성이 달라 해당 도메인에 적합한 기능점수 방법을 달리하다 보니 여러 종류의 기능점수 측정법(3D, Feature Points, Full FP's, COSMIC-FFP 등)이 파생되었고, 기능점수를 보다 편리하게 측정하는 방법을 찾다보니 사용자마다 측정하는 방법이 달라져 여러 종류의 기능점수 측정법(Mark II FPA, NESMA, FISMA 등)이 파생되었다. IFPUG 4.0, 4.1, 4.3 등은 알브레히트의 FPA의



[그림 1] 소프트웨어 기능규모 측정 방법의 발전추이

근간을 유지한 채 측정 기준을 좀 더 구체적이고 명확하게 그리고 좀 더 사용이 편리하게 지속적으로 다듬어 왔다. 특히 IFPUG 4.3(정확하게는 IFPUG CPM 4.3.1)은 기존 IFPUG CPM 4.2.1의 체계를 국제표준인 ISO/IEC 14143의 체계에 맞추어 내용을 재구성하고 일부 용어도 국제표준에 맞게 재정의해 최근(2010년 1월)에 발표하였다. 한편 전 세계 기능점수 시장의 90% 이상을 IFPUG CPM(Counting Practice Manual) 4.x가 점하고 있고, 우리나라 소프트웨어 사업대가기준도 IFPUG 4.3을 기능점수 측정의 표준으로 사용하고 있으며, IFPUG CPM을 사용할 줄 아는 수만 명의 기능점수 측정 전문가(CFPS: Certified Function Points Specialist)가 전 세계적으로 널리 있어서(국내에만 약 1,000명 가량 추산) 당분간 IFPUG 독점체제는 변하지 않을 것으로 생각된다.

## 2.2 국내 기술동향

우리나라의 경우도 마찬가지로 초창기 소프트웨어 규모 측정 방법은 대부분 코드라인수 방식이었고, 아직

도 제품이나 장치 제어 관련 소프트웨어 분야에서는 여전히 코드라인수를 고집하고 있다. 전통적으로 측정이란 측정 대상의 물리적 특성을 기준척도로 계량하는 것으로 인식하고 있어서 소프트웨어 측정을 코드라인수로 하는 것은 어찌보면 당연한 것이기도 하다. 문제는 코드라인수가 언어별로 다르다는 것과 개발자마다 다르다는 것이고 더구나 같은 개발자라도 알고리즘을 어떻게 구사하느냐에 따라 달라진다는 데 있다. 더구나 규모에 따라 가격을 지불해야 할 고객의 입장이라면 어디에 기준을 두어야 할 지 난감한 상황에 이르게 된다. 그럼에도 불구하고 초창기 우리나라 소프트웨어 사업대가기준에서는 사업대가 산정의 기준으로 코드라인수를 사용했다. 하지만 사업초기에는 코드라인수 측정이 불가능하다고 판단해 프로그램 본 수를 측정하게 한 후 여기에 일률적으로 80을 곱해 코드라인 수를 계산했다. 코드라인수를 사용하는 이유가 소스를 눈으로 보고 측정이 가능하기 때문인데 소스는 커녕 사양서도 없는 사업 초기에 요구사항만 가지고 코드라인수를 측정

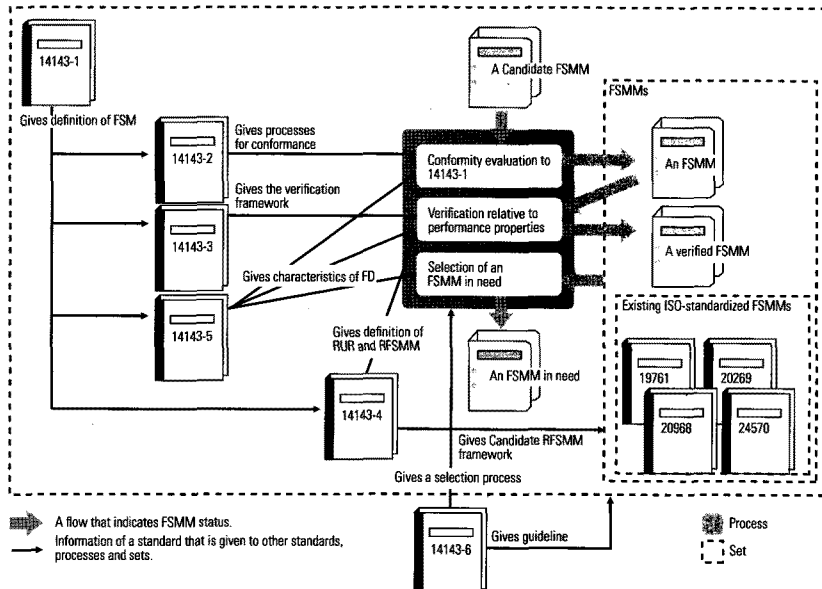
해 사업대가를 산정해야 하는 발주자의 입장에서는 현실성이 없는 상황이었다. 합리적인 척도로 바꾸기 위한 노력들이 이어졌으나 이렇다 할 성과를 거두지 못하던 차, 기능점수 방식이 국제 표준으로 채택되었다는 소식이 들려왔다. 국제 표준의 기능점수 방식을 도입하기 위한 검토가 2002년부터 꾸준히 이루어져 왔으나 정작 소프트웨어 사업대가기준의 기본 척도로 결정이 된 것은 2004년의 일이다. 그 당시만 해도 코드라인수를 버릴 수가 없어서 함께 고시되었으나 점차 기능점수 사용자가 늘어나면서 코드라인수 사용자가 급격하게 줄게 되자 지난 2009년도 고시에서는 코드라인수가 사라지게 되었고 그 결과 기능점수가 유일한 소프트웨어 규모 산정의 기준이 되어 오늘에 이르고 있다.

### 3. SW 규모산정 표준화 동향

최초로 소프트웨어 규모 측정 표준화 작업이 시작된 것은 1993년에 도쿄에서 열린 ISO/IEC JTC1 SC7 총

회에서 14143-1의 작업문서(WD: Working Document)가 WG6에 제출되면서부터였다. 당시 소프트웨어 규모 측정 표준화 작업은 별도의 워킹그룹이 없이 소프트웨어 제품 품질 관련 표준화 워킹그룹인 WG6의 한 파트로 시작되었다. 그러나 2년 후 캐나다 오타와에서 열린 총회 때 14143-1 프로젝트를 별도의 워킹그룹으로 독립시킬 것을 결의했고 그 결과 WG12가 정식으로 발족하게 되었다. WG12를 처음 이끈 사람은 호주출신의 Hugo Rehessaar(1994~1997)였고 다음은 역시 호주출신의 Pam Morris(1997~2004)였으며 마지막으로는 아일랜드의 Marie O'Neill(2005~2007)이 이끌었다. 지금은 소프트웨어 규모 측정에 대한 표준화 작업이 종료되어 표준 문서에 대한 유지보수 권한을 원래 소속인 WG6에 넘겨 WG6의 한 파트로 일본이 중심이 되어 유지보수를 하고 있는 중이다.

ISO/IEC JTC1 SC7 WG12에서 지난 15년에 걸쳐 작업한 소프트웨어 규모 측정 표준은 기능점수 규모 측정에



※출처: ISO/IEC 14143 - part 6: Guide for use of ISO/IEC 14143 series and related international standards

[그림 2] 기능 규모 측정 국제표준화문서들 간 관계

대한 내용이다. 코드라인수는 국제표준화 대상으로 적합하지 않다는 판단을 했기에 제외되었다. WG12에서 그동안 국제표준으로 발표한 기능점수 규모 측정에 대한 표준은 총 10개이다. 그 중 6개는 기능점수 개념 정의와 기능점수 측정 모델에 대한 평가방법 및 기능점수 국제표준의 사용방법 등에 관한 문서이고 나머지 4개는 국제표준의 기능점수 측정 표준에 부합되는 기능점수 측정 모델에 대한 것으로, 미국에서 제시한 IFPUG CPM 4.x, 캐나다에서 제시한 COSMIC-FFP, 영국에서 제시한 Mark II, 네덜란드에서 제시한 NESMA 등이다. 이들 10개의 표준문서들 간의 관계는 [그림 2]와 같다.

기능점수의 개념을 정의한 문서인 ISO/IEC 14143-1(파트 1)은 1998년에 표준화 되었고 각 국가에서 국제표준으로 제시한 소프트웨어 규모 측정 방법이 파트 1에서 정의한 기능점수 개념에 적합한지의 여부를 평가하는 문서인 ISO/IEC 14143-2(파트 2)는 2002년에 표준화 되었다. 기능점수 측정 방법에 대한 검증 방법을 제시한 문서인 ISO/IEC 14143-3(파트 3)은 2004년에 표준화 되었다. 기능점수 측정에 대한 참조 모델을 제시한 문서인 ISO/IEC 14143-4(파트 4)는 2004년에 표준화 되었고 어떤 기능 규모 측정 방법이 어떤 도메인에 적합한지를 평가하기 위해 모델별 적합한 기능 도메인을 결정하는 문서인 ISO/IEC 14143-5(파트 5)는 2005년에 표준화 되었으며, 기능 규모 측정 표준에 대한 사용방법에 대한 지침을 제시하는 문서인 ISO/IEC 14143-6(파트 6)은 2007년에 표준화되었다. 한편 각국에서 국제표준으로 인정해 줄 것을 요구한 기능 규모 측정 모델들에 대한 검토는 14143-2(파트 2)가 기술문서로 발표된 2000년부터 시작되어 2002년에는 MK II FP analysis method가 ISO/IEC 20968:2002로, 2003년에는 IFPUG 4.1 unadjusted FPA method가 ISO/IEC 20926:2003으로, COSMIC FFP가 ISO/IEC 19761:2003으로, NESMA FPA method V2.1 unadjusted가 ISO/IEC 24570:2003으로 각

각 표준으로 채택되었다.

국제 표준문서는 매 5년마다 개정의 필요성을 검토하기에 그동안 여러 차례 기존 표준 문서에 대한 개정작업이 이루어졌고 지금도 계속되고 있다. 최근에 이슈화 되어 개정되거나 검토되고 있는 내용으로는 14143-1(파트 1)에서 비기능에 대한 정의를 어떻게 할 것인지에 대한 것과 COMIC-FFP에서 핵심개념으로 사용되고 있는 용어 중 계층(Layer)이란 용어를 어떻게 할 것인지에 관한 것이다. 비기능적 요구사항은 이미 정리되어 개정 문서에 반영되었으나 계층이란 정의에 대해서는 논란을 거듭하고 있는 중이다.

#### 4. 정부 및 산업계의 활용현황

이미 우리는 국제표준의 기능점수를 국가표준으로 받아들였을 뿐만 아니라 매우 활발하게 사용하고 있는 중이다. 현행 소프트웨어 사업대가기준에서 유일한 소프트웨어 규모 산정 표준으로 고시되어 있고 국가 정보화 사업 예산은 그 산출근거로 기능점수 산정내역을 제시하도록 의무화하고 있다. 국내에는 이미 기능점수 측정전문가(CFPS)가 약 1,000명 가량 있는 것으로 추산되고 있고 대부분의 IT 서비스 업체들은 소프트웨어 규모 측정의 기준으로 기능점수 측정을 당연시 하고 있다. 또한 공공기관 발주자들을 대상으로 하는 발주관리 교육에는 기능점수가 20시간 정도 배정되어 있다. 국내의 유명한 교육기관에서는 기능점수 기본과정과 전문가 양성과정이 개설되어 있어서 거의 매달 운영되고 있는 중이다. 대학에서도 소프트웨어 공학 과정에 기능점수에 대한 강의가 포함되어 있다. 우리나라는 기능점수에 관하여는 이미 많은 전문가들이 양성되어 있고 교육과정도 많이 개설되어 있으나 기능점수가 제대로 활용되고 있지 못한 실정이다. 지난 해 감사원에서 소프트웨어 산업 관련 법제도의 시행 실태에 대해

감사를 벌인 적이 있다. 감사결과는 그동안의 사업대가기준의 기능점수당 단가 고시가 비현실적이었다는 것이다. 그 결과 금년도 기능점수 단가가 소프트웨어 사업대가기준 고시 이후 20여 년 만에 처음으로 16.5%나 떨어지는 사태가 발생했다. 관련 업계에서 단가 고시에 활용되는 비용자료를 랜덤 샘플링을 해서 제출하지 않고 단가가 높은 것 위주로 골라 제출한 잘못이 인지되었기 때문이다. 측정의 생명은 대표성, 정확성 및 적시성에 있는데 기본 조건을 갖추지 못한 데이터는 오히려 진실을 왜곡하여 더 큰 부작용을 초래할 수도 있다는 사례라 할 수 있다. 우리의 문제는 측정이 잘 이루어지지 않을 뿐만 아니라 이루어진다 해도 사실을 정확히 반영하는 데이터의 수집을 소홀히 한다는 것이고 지속적으로 이루어지지 않는다는 것이며, 프로세스와 프로젝트에 대한 상호 유기적 연관성을 고려한 측정이 이루어지 않는다는 것이다. 또한 측정은 별로 하지 않으면서 측정의 결과로 나오는 추정에만 관심을 보이고 그것도 범위와 확률에 의한 상식적인 수준의 추정치보다는 거두절미 유일 값 하나만을 고집하는 매우 불합리한 행태를 보이고 있다는 것이다. 측정이 없는 한 추정도 없다는 지극히 평범한 진리를 우리는 아직도 외면하고 있어 안타까운 실정이다. CMMI 레벨 5를 획득한 기업이 여럿 있지만 이들 기업들의 프로젝트 현장의 프로세스 수준은 평균 레벨 2가 안 된다는 것이 외부 전문기관의 평가에서도 드러났고 자체평가 결과에서도 드러났다. 한마디로 측정이 이루어지지 않고 있어서 측정 기반의 프로젝트 관리가 안 되고 있다는 것으로 이는 우리나라의 기능점수 활용은 측정에는 별로 사용되지 않고 있다는 것을 의미한다.

## 5. 맺음말

소프트웨어 사업은 성공률이 30%를 넘지 못하는 매

우 비정상적인 사업이다. 그 이유는 계획이 불합리하게 설정되고 요구변경이 많으나 계획수정에 제대로 반영이 되지 않으며, 프로젝트의 가시성이 확보되지 않아 관리가 제대로 이루어지지 않거나 동일 것이다. 계획이 불합리한 이유는 정확한 추정에서 비롯된 수치에 기반하지 않고 정해진 예산에 맞추어 계획을 수립하기 때문이고, 요구변경이 많음에도 계획변경이 이루어지지 않는 이유는 기준선이 애매하여 변경에 대한 식별이 잘 되지 않기 때문으로 이 역시 정확한 추정에 의한 기준선 설정이 안 되기 때문이며, 프로젝트의 가시성이 확보되지 않는 것은 정확한 측정이 이루어지지 않기 때문이다. 이 모두가 측정의 부재에서 비롯되었음을 인지하지 못하는 데 있다. 우리나라는 정부차원에서 기능점수의 사용을 적극 권장하고 있고 전 세계적으로도 기능점수를 가장 광범위하게 활용하고 있는 나라의 하나임에도 불구하고, 이렇게 측정이 미흡한 이유는 기능점수를 측정에 사용하기 보다는 추정에 사용하고 있기 때문이다. 기능점수를 올바르게 활용하려면 상용화된 비용산정 패키지에서 보는 바와 같이 수많은 프로젝트에 대한 다양하고 상세한 측정을 통해 데이터를 축적하는 일을 먼저 해야 한다. 그런 다음에 축적된 데이터를 기반으로 다양한 지식을 활용해 추정을 할 때야 기능점수의 진가가 발휘된다는 사실을 깨달아야 할 것이다.

## [참고문헌]

- [1] 정기원, 황인수, 오영배 외, 소프트웨어선진화사업 보고서, 정보통신산업진흥원, 2009.12
- [2] 황인수, 기능점수 표준화 동향 조사연구 보고서, 기술표준원, 2004.1
- [3] 소프트웨어사업 대가의 기준, 지식경제부고시 제2010-52호
- [4] '소프트웨어사업 계약제도 운영실태' 감사결과 발표, 감사원, 2009. 12. 23

- [5] The Standish Group, Chaos Report, The Standish Group Report, <http://www.standishgroup.com>, 1995
- [6] Software Measurement Services Ltd, <http://www.measuresw.com/aboutus/index.html>
- [7] ISO/IEC 14143-1 Information technology – Software measurement – Functional size measurement – Part 1: Definition of concept
- [8] ISO/IEC 14143-2 Information technology – Software measurement – Functional size measurement – Part 2: Conformity Evaluation of Software sizing methods to ISO/IEC 14143-1:1998
- [9] ISO/IEC 14143-3 Information technology – Software measurement – Functional size measurement – Part 3: Verification of functional size measurement method
- [10] ISO/IEC 14143-4 Information technology – Software measurement – Functional size measurement – Part 4: Reference Model
- [11] ISO/IEC 14143-5 Information technology – Software measurement – Functional size measurement – Part 5: Determination of Functional Domains for Use with Functional Size Measurement
- [12] ISO/IEC 14143-Part 6: Guide for use of ISO/IEC 14143 series and related international standards
- [13] ISO/IEC 19761 Information technology – Software Engineering – Functional Size Measurement – COSMIC – FFP Functional Size Measurement Method
- [14] ISO/IEC 20968 Information technology – Software Engineering – Functional Size Measurement – Mk II Function Point Analysis – Counting Practices Manual
- [15] ISO/IEC 20926 Information technology – Software Engineering – Functional Size Measurement – Function point counting practices manual Release 4.1 Unadjusted
- [16] ISO/IEC 24570 Information technology – Software Engineering – Definitions and counting guidelines for the application of Function Point Analysis(NESMA) **TTA**