

성공적인 프로젝트 수행을 위한 FP의 활용방안 검토

황인수°, 오은성, 김홍식

삼성SDS(주) 첨단소프트웨어공학센터

A review of using Function Point for the successful project

Insoo Hwang°, Eun Sung Oh, Heung Shik Kim

Center for Advanced Software Engineering, Samsung SDS Co.

E-mail : {tobehol,eingel,concert7}@samsung.co.kr

요약

전통적으로 소프트웨어 프로젝트는 납기지연, 예산초과, 높은 결함율 등으로 타 산업분야의 프로젝트에 비해 매우 높은 실패율을 기록하고 있는 것으로 알려져 있다. 이 같은 소프트웨어 프로젝트의 실패원인에 대한 많은 연구결과는 소프트웨어가 갖고 있는 범위와 요구사항 정의의 어려움, 비가시성으로 인한 초기견적의 부정확성, 역시 가시성의 부족으로 진행상황파악의 어려움에 따른 진척관리의 애로, 더욱 큰 문제는 변경의 용이성과 변경에 대한 추적의 어려움 등을 지적하고 있다. 실패한 프로젝트들의 내용을 보면 대부분 계획의 부정확성이나 위험에 대한 대처의 부족 또는 진행 중 발생하는 변경에 대한 통제 실패에서 찾아 볼 수 있다. 정확한 예측과 위험 예방 그리고 효과적인 통제대책이 소프트웨어 프로젝트를 성공으로 이끄는 3두 마차라는 지적이다. 정확한 예측의 핵심은 프로젝트 산출물인 제품에 대한 정확한 규모측정에 있고, 위험 예방은 복잡도가 높거나 불확실성이 높은 컴포넌트의 자원소요에 대한 예측과 이에 대한 준비의 소홀에서 찾을 수 있으며, 효과적인 통제대책은 프로젝트 관리 프레임워크가 튼튼하지 못하거나 이의 준수를 위한 노력의 결핍에서 찾을 수 있을 것이다. 본 논문에서는 이 3두 마차 중 가장 근간이 되고 시발점이 되는 제품의 규모에 대한 예측에 초점을 맞추어 규모측정에 가장 합리적이고 객관적이며 실용성이 높다고 현재 국제적으로 높은 평가를 받고 있는 기능점수를 프로젝트 관리에 어떻게 활용해야 프로젝트를 성공시킬 수 있을지의 방법에 대한 검토 결과를 제시하고자 한다.

1. 서론

Mike Nelson, James Clark 등은 "Curing the software requirements and cost estimating blues"에서 지난 1994년부터 2000년까지 미, 국방성이 소프트웨어 프로젝트 수행을 위해 지출한 예산은 450억불이었는데 그 중 40%인 180억불이 프로젝트 실패나 재작업으로 낭비였다면서 납기와 예산을 준수한 프로젝트는 겨우 16%에 불과했다고 주장했다. 그들은 이러한 실패의 원인이 부

적절한 요구사항과 부적절한 소프트웨어 견적에 있었다고 지적했다.

또한 무기체계의 소프트웨어 의존도가 날로 커져서 지난 1960년대에는 8% 수준이던 것이 1970년대는 20%로, 1980년대는 45%로, 2000년대 들어와서는 거의 절대적인 수준인 80%를 초과하고 있다면서 날로 급증하고 있는 무기체계의 소프트웨어 의존도에 비해 소프트웨어 프로젝트의 높은 실패율에 깊은 우려를 나타냈다. 이러한 절망적인 상

황에서 탈피하기 위한 처방의 하나로 그들이 주장한 것은 소프트웨어 프로젝트에 대한 견적의 정확성을 높이자는 것이다. 문제는 소프트웨어 규모측정의 수단으로 미, 국방성이 아직도 개발자와 개발기술 중심적이고 예산 수립당시에는 전혀 가시성이 없는 LOC를 아직도 고집하고 있다면서 이를 하루 속히 사용자 가치 중심적이고 개발 기술과는 무관하게 예산 수립 시점에서도 어느 정도의 가시성을 갖고 있는 기능점수(FP: Function Point)를 사용해야 된다고 주장했다.[1]

Johnson 등은 소프트웨어 프로젝트의 26%만이 성공적이었고, 나머지는 실패이거나 어려움에 처했었다면서 소프트웨어 프로젝트의 규모가 가장 큰 원인이었는데 특히 지난 1994년부터 프로젝트 규모가 기하급수적으로 증가했다고 주장했다.[2,3] Zubrow는 최근의 소프트웨어 프로젝트가 처한 절망적인 상황을 이렇게 요약하고 있다.

“인터넷 시대에서 수행되고 있는 소프트웨어 개발 프로젝트의 75%는 완전히 실패하거나 예산을 초과하거나 납기가 지연되거나 기능 또는 성능에 결함이 있거나 하는 등의 문제를 안고 있다.”[4]

2. 본론

2.1 소프트웨어 프로젝트의 성공을 위한 요건

이러한 실패의 가장 큰 원인은 허술한 계획에 있고 다음으로는 엉성한 실행에 있다. 허술한 계획의 원인은 다시 범위에 대한 명확한 정의와 규모에 대한 정확한 측정이 안되어 업무수행에 필요한 노력과 시간의 산정이 부정확했기 때문인 것으로 나타났다. 엉성한 실행의 원인은 목표와 요구사항에 근거한 철저한 관리가 되지 못하고 개발자들에게 방임한 결과였던 것으로 나타났다. 사용자의 요구에 근거한 정확한 계획의 수립과 목표 지향적인 실행이 될 수 있도록 하는 철저한 관리만이 소프트웨어 프로젝트 성공의 관건이다. 정확한 목표의 수립과 철저한 관리는 메트릭의 활용을 통해서만 가능한 일이다. “측정할 수 없는 것은 관리할 수 없다”는 명언을 그대로 입증한 셈이다. 메트릭에는 계획수립 메트릭(Planning Metric)과 진척관리 메트릭(Trackng Metric)이 있다. 가장 유용한 계획수립 메트릭으로는 프로젝트

의 범위와 규모를 양적으로 표시해 주는 규모산정 메트릭이다. 규모산정 메트릭을 제대로만 사용한다면 해야 할 일의 양을 정확하게 알 수가 있다. 한편 가장 유용한 진척관리 메트릭으로는 프로젝트 수행 품질과 진척도를 나타내 주는 메트릭이다. 품질과 진척도를 잘 표시해주는 메트릭을 역시 제대로만 사용한다면 프로젝트의 상태를 계획 대비하여 정확하게 표시해 줄 뿐 아니라 어떠한 수정활동이 필요한지도 표시해 준다.[5]

대상 프로젝트에 유효한 메트릭을 만들어 옳바르만 적용할 수 있다면 그 메트릭을 활용하여 프로젝트 계획을 정확히 수립할 수 있고 또한 같은 메트릭을 활용하여 프로젝트를 완전히 장악할 수 있게 되어 프로젝트의 성공은 확실하게 보장을 받을 수 있을 것이다. 따라서 성공적인 프로젝트 수행을 위한 관건은 효과적인 계획수립 메트릭을 어떻게 만들 것이냐에 있다 할 것이다.

효과적인 계획수립을 위한 메트릭의 가장 근간은 프로젝트 프레임워크에 있다.[5] PMBOK에서는 이것을 작업분류체계(WBS: Work Breakdown Structure)라고 부른다. 작업분류체계는 모든 프로젝트의 근간으로 프로젝트의 계획수립, 범위정의, 규모산정, 원가산정, 자원판단, 산출물정의, 위험식별, 의사소통 등 거의 모든 프로젝트 수행 및 관리 활동의 뿌리를 여기서 찾는다.[6] 이러한 프레임워크가 정의되면 그 구성요소인 컴포넌트들의 정확한 규모의 산정이 계획수립 메트릭의 출발점이다. 컴포넌트들의 정확한 규모의 산정은 해야 할 일의 양을 정확히 판단할 수 있게 되어 투입할 자원의 양과 프로젝트 수행에 소요될 기간을 정확히 예측할 수가 있게 된다. 이것이 계획수립 메트릭의 출발점이고 모든 프로젝트 수행 관련 메트릭의 가장 핵심이다. 그런데 문제는 기준이 되는 규모 메트릭이 측정하는 사람이나 시기에 따라 다르고 적용하는 기술에 따라 다르다면 기준의 일차적 요건인 안정성이 결여되기 때문에 심각한 문제가 아닐 수 없다. 지금까지 소프트웨어 규모산정의 기준으로 사용되어 왔던 스텝수측, LOC는 이러한 안정성의 문제에 결격사유가 있다. Jones 등은 LOC의 문제점을 다음과 같이 지적하고 있다.[7]

1. 스텝 수 방식은 경제적인 의미가 미흡하고 심지어는 언어의 수준에 따라서는 왜곡되기도 한다.
2. 스텝 수 방식은 코드가 없는 요구정의, 설계, 사용자 지침서 등의 결함에 대한 품질 수준을 제시할 수 없다. 그러나 이들 결함은 제거되어야 할 전체 결함의 50% 이상을 차지하고 있다.
3. 개발공수의 50% 이상이 코딩이 없는 공정에 투입되는데도 스텝 수 방식은 이러한 공정 (프로젝트관리, 계획수립, 각종 사양서 작성, 사용자 지침서 작성 등)에 대해서는 측정할 수가 없다.
4. 현재 사용되고 있는 언어만도 500개가 넘는데 이들 언어간 생산성 비교가 표준화 되어 있지 못하다.
5. 스텝 수 방식은 절차식이 아닌 풀 다운 메뉴나 버튼 컨트롤 방식을 사용하는 비주얼 베이직과 같은 언어에서는 사용할 수가 없다.

기능점수(FP: Function Point)는 데이터 처리 시스템을 보다 작은 컴포넌트로 나눔으로 보다 쉽게 이해하여 분석할 수 있도록 해주는 최선의 문제해결 수단인 구조적 방법에 의존하여 소프트웨어 규모를 측정하는 방법으로 논리적 설계에 기초하여 사용자에게 제공되는 기능량(Functionality)을 계량하는 방식으로 물리적 파일이나 화면 또는 프로그램 수가 아닌 사용자 관점에서의 논리적 기능량을 의미한다. [8]

소프트웨어 프로젝트 성공을 위한 요건은 계획수립 메트릭과 진척관리 메트릭을 개발하여 해당 프로젝트에 가장 적합한 형태로 적용하는 것이고, 그 중에서도 특히 계획수립의 생명은 정확성에 있는데 이를 가능하게 해주는 것은 소프트웨어의 규모를 사용자의 관점에서 일관되게 기술과는 무관하게 그것도 프로젝트 초기에 측정이 가능해야 하는데 이러한 요건을 모두 충족시켜 주는 것은 기능점수 밖에 없다.[9]

2.2 FP의 활용가치에 대한 검토

2.2.1 사용자 요구사항 관리

프로젝트 실패의 가장 큰 원인의 하나로 지목되고 있는 사용자 요구사항 관리는 특히 소프트웨어 프로젝트에 있어서는 쉬운 일이 아니다. 사용자 요구사항을 처음부터 구체적으로 명확하게 정

의하기가 쉽지 않을 뿐 아니라 더욱 어려운 점은 프로젝트 진행 중에 대부분의 프로젝트에서 월평균 2~3%의 변경이 발생하기 때문이다.[7] 만일 프로젝트 기간이 10개월이라면 요구사항의 변경이 20~30%에 이른다는 의미이다. 이렇게 될 경우 계약변경을 하지 않는 한 최초의 계약기준으로 프로젝트를 성공시킨다는 것은 결코 쉬운 일이 될 수가 없을 것이다. 그러나 FP를 사용하여 사용자의 요구사항 관리를 하게 된다면 사정은 매우 달라질 수 있다. 호주 빅토리아 주정부는 개발자와의 계약을 FP에 기반하여 총 개발규모를 정해 놓고 기능점수당 단가 계약을 한 후 범위관리자를 두어 프로젝트를 수행한 결과 납기지연이나 예산 초과로 인한 프로젝트 실패율을 50%이상 개선한 사례를 보고하고 있고, 미국의 Nielsen Media 사는 협력업체와의 계약을 FP로 한 결과 상호간의 갈등요소가 많이 해소되어 원만한 협력관계를 유지할 수 있었다는 보고하고 있다.[11,13] 이처럼 소프트웨어 프로젝트의 가장 큰 실패요인의 하나인 사용자 요구사항 관리가 FP를 사용함으로써 원만하게 해결될 수 있음을 알 수 있다.

2.2.2 원가 관리

앞에서 지적한 바와 같이 전통적으로 소프트웨어 프로젝트는 예산초과가 비일비재하여 Standish Group의 "Chaos"에 의하면 42%의 프로젝트 만이 주어진 예산 내에 프로젝트를 끝낼 수 있었다고 한다. 예산을 초과하여 일을 한다는 것은 사업의 지속성을 보장할 수 없다는 의미이기 때문에 심각한 문제가 아닐 수 없다. 소프트웨어 프로젝트에서 예산초과가 발생하는 주요 원인은 크게 2가지로 하나는 견적의 부정확성에 있고 다른 하나는 변경관리의 실패로 최초 예산을 초과하여 일을 하게 되는 경우이다. 전자의 경우는 견적의 정확성을 기해야 하는 일이고 후자의 경우는 사용자 요구사항에 대한 변경관리를 철저히 하는 일로 모두가 FP를 사용하면 쉽게 해결할 수 있는 사안이다. 견적의 정확성은 물론 요구사항이 확정되어야 하고 명확해야 하겠지만, 일단 이것이 전제 된다면 남는 것은 요구사항을 정확하게 측정하는 일과 개발자 자신의 생산성과 품질 수준을 아는 일이 될 것이다. 여기서 어려운 문제는 요구

사항을 구체적인 수치로 계량화하는 일인데 사용자의 시각에서 개발기술과는 무관하게 일관된 측정치를 제공해 줄 수 있는 FP야말로 가장 확실한 답이 될 것이다. 요구사항의 변경에 대한 통제는 앞 절에서 설명한 바와 같이 FP에 기반하여 계약을 하고 FP를 가지고 통제를 해나간다면 설명 변경이 발생한다 하더라도 계약관리가 용이하기 때문에 FP야말로 예산초과의 문제를 근본적으로 해결해 줄 수 있는 가장 확실한 대안이 될 수 있을 것이다.[14]

2.2.3 일정 관리

전통산업의 프로젝트에서는 대체적으로 작업분류 체계가 분명하고 각 워크패키지에 대한 명확한 역할과 책임 및 일정과 예산의 할당이 패턴화 되어 있으나 소프트웨어 프로젝트에서는 작업분류 체계도 워크패키지에 대한 식별이나 일정과 예산의 할당도 용이하지가 않는 것이 특징이다. 가장 큰 원인은 역시 사용자 요구의 불명확성과 프로젝트 진행 중 빈번한 변경으로 인한 안정성의 부족 등에 있는 것으로 알려져 있다. 따라서 소프트웨어 프로젝트 일정관리의 관건은 요구사항의 명확성과 안정성에 기반 한 프로젝트 산출물의 통합체계 구축이 우선이다. 산출물 통합체계 구축을 위한 가장 합리적이고 효율적인 작업단위를 구분하고 이의 개발절차 수립과 최적의 담당자 할당 그리고 작업단위별로 실현 가능한 시간의 할당이 다음이다. 작성된 일정을 효과적으로 통제하기 위한 메트릭의 개발과 메트릭의 운영절차 수립이 필요하고 나아가서 수정에 따른 일정통제 대책의 수립이 또한 필요하다. 요컨대 최적의 스케줄 작성과 일정통제를 위한 메트릭의 활용이 일정관리의 핵심이라는 의미이다. 최적의 스케줄 작성을 위해서는 작업단위에 대한 정확한 업무량의 판단과 작업단위의 고유 특성을 감안한 알고리즘의 개발이 필요하고 메트릭의 활용을 위해서는 일정 통제에 가장 효과적이고 현실적인 메트릭의 개발과 이의 운영절차의 개발이 필요하다. 역시 여기에서도 FP가 작업단위의 업무량 판단에 결정적인 기여를 할 수 있을 뿐 아니라 획득가치를 계량화하는데 매우 유용하게 사용될 수 있다. 또한 일정의 진척도표시를 하는데 없어서는 안 되는 규모

산정의 가장 합리적인 기준이 FP라는 사실은 여러 차례 강조를 했다.[9]

2.2.4 품질 관리

소프트웨어의 가장 큰 특징의 하나는 사용자에게 보이지 않을 뿐 아니라 손으로 만질 수도 없다는 것이다. 이것이 바로 소프트웨어 품질관리를 어렵게 하는 요인이다. 관리의 핵은 가시화에 있는데 이렇게 비가시적인 소프트웨어 품질을 여하히 가시화 할 것이냐가 소프트웨어 품질관리의 관건이다. 일반적으로 품질의 문제는 올바른 일을 올바르게 하지 않아서 생기는 문제로 치료를 통하여 확보되는 것이 아니라 설계를 통하여 확보할 수 있는 것이다. 따라서 품질관리는 예방에 있지 발견과 치료에 있지 않다는 의미이다. 이러한 품질보증 활동의 효과성은 FP당 결함수나 소프트웨어 결함에 의한 시스템의 다운 주기 등으로 표현할 수 있다. 여기서 FP가 갖는 의미는 다른 프로젝트나 조직의 품질관리 활동의 효과성을 서로 비교해 볼 수 있도록 해주는 것이다. 다시 말해서 품질의 좋고 나쁨의 평가기준이 될 수는 있는 기준 수치를 제공해 줄 수 있다는 점이다.

2.2.5 생산성 지수

생산성이란 투입대 산출로 경제적 의미가 크다. 생산성 측정에서 기준이 되는 것은 산출로, 어떻게 그 크기를 아느냐는 것이다. 생산성이란 비교를 통하여 경쟁력 고취와 수익의 향상이 목적이기 때문에 그 생명은 일관성과 단순성에 있다. 따라서 산출물을 측정하는 기준은 단순해야 하고 누가 언제 어떠한 환경에서 측정을 해도 같은 결과가 나와야 하며, 개발자의 관점이 아닌 사용자의 관점에서 측정이 되어야 한다. 소프트웨어 규모를 측정하는 기준에는 크게 물리적인 척도와 논리적인 척도가 있는데 물리적인 척도란 화면수나 프로그램 본수 또는 스텝수와 같은 것으로 육안으로 쉽게 측정할 수 있으나 개발환경과 구현 상태에 따라 심지어는 측정하는 사람의 기준에 따라서도 다를 수 있다는 단점이 있다. 반면 논리적 척도는 논리스텝수나 기능점수와 같은 것들로 육안으로 식별할 수 있는 것이 아니고 논리적인 판단기준에 따라 심층적인 안목으로 측정되는 어

려움이 있으나 개발환경이나 구현상태와는 무관하게 측정하는 사람과는 무관하게 같은 측정결과가 나온다는 장점이 있다. 소프트웨어 산업의 경쟁력 제고를 위해서는 업체간, 프로젝트간, 개발도구간, 적용 방법론간, 시기별 등의 생산성 비교가 반드시 필요하기 때문에 시간과 장소, 기술, 사람 등과는 무관하게 일관된 측정이 가능한 FP와 같은 논리적 척도를 사용하는 것이 바람직하다.[7,9]

2.2.6 소프트웨어 프로세스 개선

벤치마킹을 통한 기업의 경쟁력 제고 노력이 어느 때보다도 활발한 오늘날 조직의 능력향상이 초미의 관심사가 되었다. 조직의 능력을 향상시킨다는 의미는 조직의 생산성과 품질 및 수익성이 개선된다는 의미로 구성원의 자질 향상은 물론 조직차원의 업무표준화와 재활용이 활발해진다는 의미이다. 조직도 하나의 유기체로서 성장 수명주기가 있고 발전단계가 있게 마련이다. 이러한 사고에 근거하여 발전시킨 개념이 조직의 프로세스 정의와 개선이다. 조직은 자신만이 갖고 있는 프로세스를 통하여 성장하게 되는데 조직에 따라 많은 차이를 보인다. 그 결과는 조직의 프로세스가 내는 생산성과 품질로 측정이 되고 이러한 프로세스를 통하여 산출한 프로덕트의 생산성과 품질로 측정이 된다. 그런데 후자의 생산성과 품질은 전자의 그것에 의해서 결정되기 때문에 모든 기업은 자신의 프로세스 능력을 개선하기 위한 각고의 노력을 경주하고 있다. 소프트웨어 산업분야에서의 이러한 노력의 일환으로 대표되는 것이 바로 SEI의 CMM과 ISO/IEC의 SPICE이다. 근자에 국내 많은 기업들이 CMM이나 SPICE 심사를 받아 자신의 프로세스 능력을 과시하려는 경쟁이 치열하게 벌어지고 있는데 그것은 한마디로 조직차원의 생산성과 품질개선을 위한 노력의 일환으로 보아도 좋을 것이다. 여기서의 초점은 지속적으로 프로세스가 개선되어야 한다는 것이고, 그것은 바로 프로세스의 생산성과 품질능력이 향상된다는 것을 의미하는 것이다. 과거에는 CMM이나 SPICE 모두 생산성이나 품질수준의 측정을 위한 측정기준으로 스텝수(LOC)를 사용했으나 근자에 와서는 LOC의 한계를 인정하고 비교와 벤치마킹

의 강점이 있는 FP를 모두 수용하고 있다. 그럴 수 밖에 없는 것은 LOC로는 다른 언어 다른 개발환경 다른 개발자 다른 조직간의 일관된 측정치를 제공해 줄 수 없기 때문이다. 역시 FP만이 여기에 대한 확실한 답을 주기 때문인 것으로 풀이된다.[5,7,8,9,10]

2.3 프로젝트 성공을 위한 FP 활용방안

일반적으로 프로젝트 계획 수립에 필요한 정보는 대략 다음과 같은 것들로 간주된다.

- 업무범위: WBS로 표현됨
- 개발규모: 국내에서는 본수/스텝수를 주로 사용하여 왔으나 선진국에서는 이미 일반화 되어있고 현재 국제표준으로 자리잡아가고 있는 FP를 사용하자는 요구가 점차 강하게 일고 있음
- 투입공수: 주로 Man Month를 사용하고 있으나 관리의 정확도를 높이기 위하여 Man Day나 심지어는 Man Hour를 사용하기도 함
- 납기: 주로 Calendar Month로 표현됨
- 실행예산: ₩/MM or ₩/FP 등의 단가를 사용하여 표현하고 직접비와 간접비 및 이익율로 표시됨
- 품질수준: 일반적으로 결함율(Defects/FP)로 표현됨

한편 소프트웨어 견적을 할 때 나오는 데이터의 종류에는,

- 업무범위 정의서: 보통 구조적인 프로덕트 구성을 나타내는 계층적 구조도로 나타남
- 개발대상 규모: FP로 나타남
- 개발소요 공수: MM/MD/MH 등
- 적정소요 기간: 카렌다 개월 수
- 품질수준: 결함수/FP
- 개발원가: ₩/FP, 직접인건비, 간접비, 이익율 등 프로젝트 계획수립에 필요한 데이터와 견적결과로 나오는 데이터들 간에 거의 차이가 없음을 볼 수 있다. 그리고 견적결과로 나오는 수치의 핵은 규모에서 나오는 FP임을 쉽게 알 수가 있다. 보통견적활동을 통하여 얻게 되는 수치는 프로젝트 관리 시스템에 입력되어 프로젝트의 계획 수립은 물론 프로젝트의 통제에도 유용하게 활용되어야 의미가 있다고 Jones는 주장하였다.[10] 한편 그

견적의 핵심은 소프트웨어 규모 측정으로, 가장 일관성이 있고 합리적이며 사용자의 요구만 명확하다면 프로젝트 초기단계에서도 얼마든지 측정이 가능한 FP가 프로젝트의 성공적인 관리에 매우 유용한 수단임을 강조하였다. FP에 의한 견적 결과는 정확한 계획 수립이외에도 소프트웨어 프로젝트 관리에 있어서 가장 어려운 변경관리의 기준으로도 매우 유용하게 사용할 수 있음을 제시해 준다. 견적결과로 나온 업무범위와 규모를 바탕으로 계약을 했다면 그것이 변경관리의 기준선이 되어야 함은 너무도 당연하다 할 것이다. 미국의 Nielsen Media 사의 CIO인 Kim Ross는 파트너사인 CTS사와 프로젝트 계약시 총 개발규모를 정해 놓고 기능점수당 단가 계약을 하고 쌍방간에 변경관리자를 지정하여 프로젝트를 수행한 결과요구변경을 막지는 못했으나 변경에 대한 상호간의 명확한 인지를 바탕으로 원만하게 프로젝트를 완료할 수 있었다고 보고하였다.[11] 견적결과가 최대한 정확해야 하겠지만 설령 그렇지 못하더라도 계약 당사자간에 명확히 인지되고 그것을 관리하기 위한 지속적인 노력(중간견적)이 있다면 최소한 파국은 면할 수 있다는 이야기다. FP기반의 견적은 또한 프로젝트를 데이터 중심으로 관리할 수 있게 함으로서 진행상황을 정확히 파악할 수 있게 해줄 뿐 아니라 성공을 위한 다양한 대책을 강구할 수 있도록 도와주는 역할을 한다. 견적이야말로 데이터가 전부다. 합리적인 근거가 제시되지 않는 데이터도 있을 수 있으나 이런 데이터는 차라리 없느니만 못하다. 거듭 지적한 바와 같이 견적은 합리적인 근거를 생명으로 하기 때문에 견적 데이터에 기반한 프로젝트 관리야말로 가장 합리적인 관리라 할 수 있을 것이다. 프로젝트가 진행되는 과정에서의 FP측정과 견적은 지금까지 진행된 실적을 바탕으로 프로젝트가 끝날 때까지 소요될 기간과 공수와 원가 등을 예측해 주는 기능을 한다. 계획 대비 지금까지의 실적을 평가할 수가 있을 뿐 아니라 이러한 추세로 계속될 경우 향후에 소요될 자원과 그 결과를 알 수 있게 해주는 것이다. 그래서 진척관리에서 견적기능을 잘 활용하면 프로젝트가 어디쯤 가고 있는 지, 어디로 향하고 있는지, 목표 달성을 위해서는 얼마나 패도 수정을 해서 더 가야

하는지 등을 보다 정확하게 알 수 있게 해준다. 진척관리를 가장 효율적으로 할 수 있게 해주는 기법으로 잘 알려진 기성고 관리기법 (EVMS: Earned Value Management System)은 프로젝트 진행상황을 한 눈에 알아 볼 수 있도록 데이터로 표시해 준다. FP기반의 견적의 활용은 기성고 관리기법과 결합하여 지금까지의 실적을 기준으로 향후의 소요자원을 예측할 수 있게 해줄 뿐 아니라 보다 적극적으로는 현상황에서 납기나 예산 및 품질을 준수하기 위한 향후의 투입자원 변경이나 내부 동기요소, 개발기법 및 도구의 추가 투입 등의 소요를 판단해 주는 기능을 한다. 프로젝트 종료단계에 가서는 프로젝트의 성과를 평가해 줄 뿐 아니라 견적의 문제점을 파악하여 보다 정확한 견적을 할 수 있도록 데이터와 방법을 제공해준다. 프로젝트 성과 평가란 계획시 견적 데이터를 기반으로 프로젝트 성공의 기준으로 잡았던 기준선에 비하여 얼마나 잘했는지 또는 못했는지를 데이터로 정확히 표시해준다는 의미다. 그 결과는 전체 프로젝트 단위로 알 수 있을 뿐 아니라 서브 프로젝트나 외주업체 그리고 필요하면 개인단위까지 알 수 있게 해준다. 또한 종료단계에서 실적을 카운트하여 견적의 정확도와 문제점을 파악할 수 있게 해준다. 프로젝트 착수 전의 견적결과와 종료 후의 실적카운트 결과가 얼마나 차이가 있는지, 그 원인은 무엇인지, 왜 그 원인을 처음에 예측을 못했는지, 향후의 견적에서 고려해야 할 점은 무엇인지 등을 자세히 알 수 있게 해준다. 더구나 종료시 카운트한 실적 데이터를 견적 데이터베이스에 축적해 놓을 수 있게 됨에 따라 향후에 각종 유용한 정보를 추출할 수 있는 점도 간과되어서는 안 된다. 지금까지 견적이 프로젝트 관리에 어떻게 기여할 수 있는지에 대해서 설명을 하였다. 견적은 제안이나 프로젝트 착수 전에 하는 일회성의 활동이 아니라 프로젝트 수명주기 동안에 지속적으로 실시하여 그 결과를 프로젝트 관리에 유용하게 활용할 수 있음도 알았다. 기획단계에서는 프로젝트 예산을 수립하기 위함이고, 제안/계약 단계에서는 우리의 원가를 파악하여 고객의 예가 대비 얼마나 손익이 발생할 지를 알아 제안여부와 제안가를 알고자 함이며, 계획단계에서는 앞에서 설명한 바 프로젝트

성공을 위한 정확한 실행예산을 편성하고자 함이고, 진행단계에서는 변경에 따른 원가변경을 파악하고자 함이며, 종료단계에서는 건적의 정확도와 프로젝트의 성과 등을 알고자 함이다. 이러한 단계별 건적행위는 프로젝트가 계약 목표로부터 이탈하지 못하도록 할 뿐 아니라, 프로젝트팀이 목표 지향적으로 일을 하도록 촉구하는 효과를 발휘하기 때문에 프로젝트 성공을 향한 꺾대 역할을 한다고 할 수 있다 할 것이다.

3. 결론

오늘날과 같이 컴퓨터가 생활의 일부가 되어 있는 "Ubiquitous Computing" 시대에 살고 있는 우리는 알게 모르게 소프트웨어에 우리의 삶을 의지하고 있다고 해도 과언이 아니다. 소프트웨어에 대한 수요의 급증과 그에 따른 소프트웨어 인력의 부족사태는 우리 나라는 물론 전 세계적으로 너무나 당연한 일상의 현상이 되어버렸다. 일찍이 NATO는 이러한 문제를 예견하고 소프트웨어 위기를 경고하였다. 그 경고가 나온지 40년이 흐른 지금에도 소프트웨어 위기는 지속되고 있고 획기적인 문제해결의 기미는 보이지 않고 있다. 서두에서 본 참담한 소프트웨어 프로젝트의 성공률이 이를 말해 주고 있다. 그 동안 소프트웨어 위기를 극복하기 위한 기술 및 관리적인 여러 가지 대책이 제시되었고 많은 경우 효과를 보았으나 무엇보다 중요한 것은 소프트웨어 프로젝트가 성공을 해야 한다는 점이었다. 프로젝트가 성공하려면 정확한 계획의 수립이 무엇보다 중요하고 계획과 프로젝트 수행환경을 보호하기 위한 위험예방활동이 또한 중요하며 계획을 준수하려는 효과적인 통제대책이 있어야 함을 강조하였다. 이 중에서도 프로젝트의 베이스라인이 되는 정확한 계획의 수립이 프로젝트 성공의 관건임을 거듭 강조하였다. Weaver 등은 프로젝트 수명주기의 10~20%에 해당되는 초기단계가 전 프로젝트에 절대적인 영향을 미치는 반면 가장 적은 노력으로 손쉽게 프로젝트에 영향을 끼칠 수 있다면서 Galorath가 지적한 프로젝트 프레임워크의 구축을 위한 프로젝트 착수 전문가의 육성과 초기투입을 적극 권장하였다. [12]

Weaver, Galorath 등이 주장한 초기단계의 노력은 정확한 계획의 수립을 통한 베이스라인의 설정에 역점을 둔 것으로 FP가 바로 여기에 충분히 기여할 수 있는 조건을 갖추었음을 본 논문은 보여주었다. 베이스라인이란 고객과 개발자간의 합의가 전제되어야 하고 메트릭을 통하여 가시화되어야 하는 것으로 사용자 관점의 일관되며 기술과 무관한 측정 기준인 FP야말로 이러한 베이스라인의 설정에 필요충분조건을 갖춘 것으로 평가된다. 실제로 Pam 등은 그들의 소프트웨어 개발 프로젝트에서 FP당 단가 기준의 계약과 납품 규모를 FP로 통제하는 방식으로 90%에 가깝던 예산초과 프로젝트 비율을 10% 미만으로 줄일 수 있었다고 보고하였다.[13]

따라서 FP로 소프트웨어 프로젝트 범위와 규모를 산정하여 계획을 수립하고 프로젝트관리 베이스라인을 설정할 뿐 아니라 고객과의 계약도 FP에 기반 한 계약방식 즉, 납품규모를 FP로 통제하고 비용을 FP당 단가로 계약함으로써 소프트웨어 프로젝트의 성공율을 획기적으로 개선할 수 있음을 알 수 있다.

[참고문헌]

- [1] Curing the software requirements and cost estimating blues: The fix is easier than you might think, Mike Nelson & James Clark, 2001
- [2] Turning chaos into success, Jim Johnson, CIO Magazine, 1999.12
- [3] The secret to software success, Scott Berinato, CIO Magazine, 2001.7
- [4] Beating the odds: measurement and project success, Dave Zubrow, SEI, 2001.10
- [5] Effectively utilizing software metrics: Project metrics, Daniel Galorath, IT Measurement, 2002
- [6] PMBOK: Project Management Body of Knowledge, PMI, 2000
- [7] What it means to be "Best In Class", Capers Jones, 1996
- [8] "Function Point counting Practices Manual" Release 4.1, IFPUG, 1999

- [9] "Function Point Analysis" Measurement Practices for Successful Software Projects, David Gamas, Addison-Wesley, 2001
- [10] KnowledgePlan Education Materials, C. Jones 1999
- [11] "Nielsen Media Research's Software Estimation System," CIO, March 15, 1999.
- [12] The project start-up conundrum, Patrick Weaver & Lynda Bourne, PMI Melbourne chapter meeting, 2002.3
- [13] Measurement now and in the future, Pam Morris, (사)한국정보기술원가표준원 초청세미나 자료, 2002.5
- [14] On time in budget, how did you do it?, The David Consulting Group, Inc., 2001