11

소프트웨어 개발 프로젝트의 제약 도출 및 해결방안에 관한 연구

- 국방부문 Software 개발 project 사례를 중심으로 -

쌍용정보통신(주) ▌박 재 모

1. 서 론

국방분야에서 소프트웨어의 개발 능력은 군의 전투력 수준을 결정짓는 요소가 되고 있다. 예를 들어 1960년대에 생산된 F4 전투기에 들어가는 소프트웨어 비용은 전체 비용의 8%에 불과했지만 2000년대에 생산된 F22 전투기는 전체 비용의 80%가 소프트웨어 관련 비용이다[1]. 시스템 분야에서도 지휘통제통신전산정보(C4I) 체계와 같은 지휘통제시스템, 육군과학화전투훈련체계(KCTC)와 같은 교육훈련시스템, CALS(Computer-aided Acquisition and Logistic Support)와 같은 군수지원시스템 등 군의 작전과 관련된 모든 업무가 소프트웨어를 중심으로 한 시스템으로 빠르게 변화하고 있다.

하지만 유감스럽게도 국방부문 소프트웨어 개발과 관련하여 일정이 지연되고, 예산을 초과하며, 기대한 성능이 구현되지 못하는 경우가 발생할 가능성은 여전히 존재하고 있다. 특히, 국방부문의 소프트웨어 개발은 육군 지휘통제통신전산정보 체계, 육군과학화전투훈련체계, 해군 전술지휘통제체계, 공군 전술방공망 지휘체계 등 수 백억원~수 천억원에 달하는 막대한 국가예산과 장기간의 개발기간에 걸쳐 이루어지는 경우가 많다. 지금까지는 모든 관계자들의 헌신과 노력에도 이러한 대규모 프로젝트가 잘 수행되어 왔지만, 앞으로도 계속 성공하리라는 보장은 없다. 만일 이러한 대규모 소프트웨어 개발 프로젝트가 실패한다면 이는 막대한 국가예산의 낭비뿐만 아니라 수년간 준비해 온 군 전투력 향상 노력이 수포로 돌아간다는 것을 의미한다. 게다가 빠르게 변화하는 작전환경과 한반도 주변 정세 변화에 대응하기 위해 소프트웨어 개발 프로젝트의 규모는 더욱 커지고 복잡도는 증가하는 반면, 개발기간은 단축되고 있는 상황이다. 프로젝트 수행환경 또한 치열한 수주경쟁으로 인한 저가 계약, 무리한 계약조건 등으로 악화되고 있다.

따라서 단순히 생산성을 향상시키는 새로운 기술 또는 도구가 아니라 현재의 문제를 근본적으로 해결하기 위한 접근이 필요하다. 본 연구에서는 소프트웨어 개발 프로젝트의 성공적인 수행(성과 및 수익 개선)을 위해 지금까지 생산성을 획기적으로 개선한다고 알려진 각종 기술, 방법, 도구에 초점을 맞추는 것이 아니라, 프로젝트 성공 및 수익증대를 가로막는 제약이 무엇인가를 도출하고, 이를 해결하기 위한 방법에 초점을 맞추었다. 현황파악 및 제약 도출을 위해 시스템 통합 회사 중 하나인 A社의 국방부문 소프트웨어 개발 프로젝트에 참여하고 있는 관리자, 개발자 47명을 대상으로 설문조사를 실시하였다. 설문을 통하여 제약조건을 파악하고 이를 해결하는데 필요한 프로젝트 관리 기법을 제안하였다.

2. 국방 소프트웨어 개발의 문제점

설문 조사 결과 응답자의 41%가 프로젝트 관리능력 부족이 문제의 가장 큰 원인이라고 답변했으며, 35%가 품질관리능력 부족을 문제의 원인으로 답변하였다. 반면에 S/W개발환경 및 개발능력을 문제의 주요 원인이라고 응답한 비율은 각각 8%, 5%에 그쳤다. 각 분야별 주요 문제를 도출하면 표 1과 같다.

하지만 이러한 문제 중 단순히 설문 응답비율이 높다고 해서 이것이 프로젝트의 주요 제약이라고 할 수는 없다. '무엇이 프로젝트의 throughput를 가장 효과적으로 향상시키는 것인가?'를 찾아내는 것이 중요하다. 이러한 관점에서 프로젝트의 제약을 도출하기 위해 프로젝트 주요 문제간의 특성요인을 분석하였다. 특성요인은 설문결과에 나타난 주요 문제를 중심으로 문제를 발생시키는 원인을 추적하였으며, 설문결과가 도출된 제약과 상관관계를 갖고 있는 가를 파악하기 위해 설문 응답률을 기준으로 선의 굵기를 달리하여 표시하였다. 이는 전통적인 특성요인도의 기법과는 다소 상이하나 현장에서 프로젝트 참여자들이 느끼는 문제점이 실제로 프로젝트의 성과에 얼마만큼

표 1 분야별 주요 문제점

분야	문 제	응답비율
프로젝트 관리	불합리한 계약조건	31%
	부정확한 프로젝트 공정관리에 따른 현황 파악 애로	23%
	부정확한 예산 측정 및 자원 산정	23%
	고객과의 의사소통 및 협조능력 부족	17%
소프트웨어개발	불명확한 사용자 요구사항 및 잦은 변경	80%
소프트웨어 개발환경	소프트웨어 시험장비 및 도구 부족	23%
	소프트웨어 개발장비 및 도구의 도입지연	21%
	소프트웨어 개발장비 및 도구의 부적합	17%
교육	교육시간 부족	27%
	교육인원 선정기준 및 절차 부적절	19%
	교육예산 부족	17%
품질관리	표준 프로세스의 현실성 부족 및 실무적용 애로	44%
	품질관리의 실질적 성과 미흡	23%

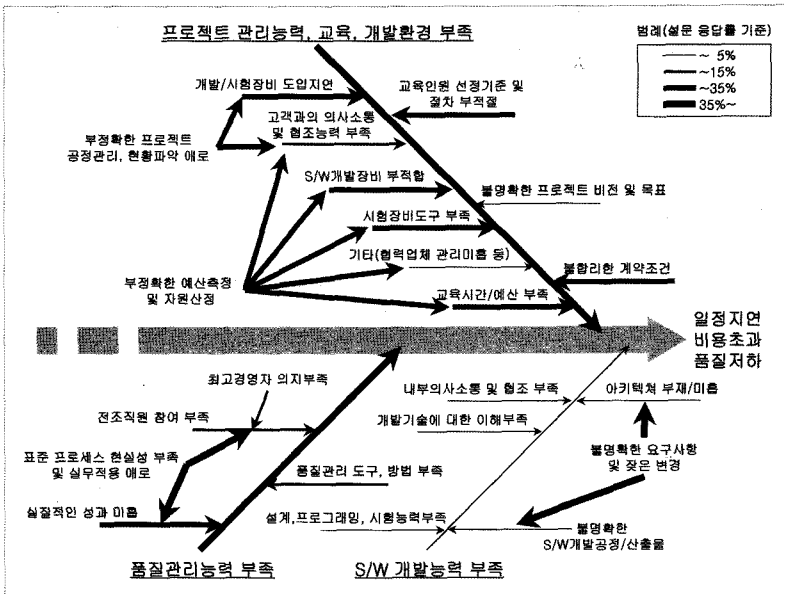

그림 1 프로젝트 주요 문제의 특성 요인도

의 제약을 갖는지를 파악하는데 유용하다.

특성요인 분석 결과 그림 1과 같이 프로젝트 초기 부정확한 예산 측정과 자원 산정은 프로젝트의 가장 큰 문제로 조사된 프로젝트 관리능력 부족뿐만 아니라 교육, 개발환경에 있어 프로젝트 개발 기간 내내 여러 문제를 야기시키는 근본 원인으로 나타났다. 아울러 프로젝트 진행간 부정확한 프로젝트 공정관리로 인하여 적절한 통제 및 조치시기를 실기(失期)하게 되고 고객과의 의사소통 및 협조능력을 떨어뜨리는 것으로 나타났다. 동일한 맥락에서 소프트웨어 개발 초기 불명확한 요구사항 정의 및 이로 인한 잦은 변경은 이후 설계뿐만 아니라 전 체적인 소프트웨어 개발 공정에 영향을 미치는 근본 원인으로 나타났으며, 품질관리에 있어서는 표준 프로세스의 현실성 부족 및 실무적용 애로로 인해 조직원의 참여도가 떨어지고 실질적인 성과 달성에 실패하는 것으로 나타났다.

이러한 특성요인 분석 결과를 바탕으로, '과연 이것이 프로젝트의 통제점으로 유용한가?', '이것을 해소하면 프로젝트의 성과가 개선되는가?'를 기준으로, 프로젝트에서 가장 시급히 해소해야 할 제약을 식별하면,

① 부정확한 예산측정 및 자원산정, 프로젝트 공정관리에 따른 현황 파악 애로
② 불명확한 사용자 요구사항 및 잦은 변경
③ 표준 프로세스의 현실성 부족 및 실무적용 애로, 품질관리의 성과 미흡

이상 세가지로 요약 할 수 있다[2].

이를 해결하기 위한 대안을 마련하기 위해서는 각각 다음이 고려되어야 한다. 첫 번째 제약을 해소하기 위

해서는 소프트웨어 개발 프로젝트의 견적과 진행현황을 가시화 할 수 있는 수단이 마련되어야 한다. 견적은 프로젝트의 목적을 달성할 수 있다는 확신을 심어줄 수 있을 만큼 객관적이고 합리적인 방식으로 이루어져야 한다[3]. 그리고 프로젝트에 참여하는 모든 사람들이 일관된 의사소통을 할 수 있도록 진행현황을 요약하여 가시화 할 수 있는 수단이 마련되어야 한다. 이를 통해 자원활용의 적시성 및 효율성을 확보할 뿐만 아니라, 고객과의 의사소통 수단을 마련하고, 사업의 위험을 조기에 발견, 예방 또는 완화시킬 수 있다[4].

두 번째 제약을 해소하기 위해서는 사용자 요구사항을 도출하고 가시화하는 수단과 도출된 요구사항을 이해하기 위한 공통된 기준이 제시되어야 한다. 이는 소프트웨어 개발 전략과 무관하게 적용 가능해야 한다. 이를 통해 불명확한 요구사항으로 인한 재개발 및 공정지연을 감소시킬 수 있다.

세 번째 제약을 해소하기 위해서는 프로세스 개선활동의 현실성을 제고하고 프로젝트의 성과 향상에 직접적인 영향을 줄 수 있는 방안이 제시되어야 한다. 프로세스 개선활동은 전사적인 활동이므로 이러한 부분이 고려되어야 한다.

3. 국방 프로젝트 관리 전략

3.1 기성관리 시스템을 통한 국방 프로젝트 통제 기법

기성관리시스템(EVMS, Earned Value Management System)은 최적의 프로젝트 계획과 통제를 위해 일정/비용 요소를 이용하여 프로젝트의 작업을 정의하고 다양한 분석지표를 통해 현황을 분석, 통제하는 관리기법으로써, 1996년 미 국방부에 의해 승인된 32개의 기준으로 구성되어 있다.

모든 일을 분할하여 WBS(Work Breakdown Structure)로 정의하고, WBS에서 정의된 각각의 일을 수행하기 위한 조직(OBS, Organization Breakdown Structure), 일정, 예산 등을 모두 할당한 Control Account(CA)의 통합계획(CAP, Cost Account Plan)으로 프로젝트의 전체 일정계획을 수립한 후, 계획을 측정하기 위한 기준선(Baseline)을 설정하고, 데이터를 측정하여, 프로젝트의 비용, 일정, 생산성 현황 등을 분석하여 적절한 조치를 취하는 절차로 구성되어 있다.

3.1.1 Cost Accounts 산정 및 통합계획(CAP)/Baseline 수립

기성관리시스템을 실무에 적용하는데 있어 가장 어려운 부분은 일정상에 있는 각각의 일(WBS)에 자원(OBS, 시간)을 할당하여 모든 일을 비용화 하는 것(Control Account)이다. 소프트웨어의 분석, 설계, 구현, 시험공정에서 발생하는 모든 일을 Control Accounts로 환산해야 하기 때문이다. 소프트웨어 엔지니어링 활동같이 직접적인 작업시간이 산출 가능한 업무는 적용이 비교적 용이하지만 프로젝트 관리, 품질관리, 형상관리, 위험관리 등과 같은 공통업무에 대한 비용과 일정을 산정하기는 쉽지 않다. 이것이 어렵기 때문에 많이 쓰이는 방법 중에 하나가 투입인력을 기준으로 한 비용계산 방식이다. 하지만 이는 실제 수행될 일에 대해 비용을 계산한 것이 아니고 단순히 투입인력 수(M/M)를 기준으로 비용을 산정한 것이기 때문에 프로젝트 현황 관리 및 통제에 있어 현실적이고 신뢰성 있는 정보를 제공하기 어렵다. 투입인력 수와 일과의 상관관계가 명확하지 않기 때문이다. 따라서 각 일에 대한 조직의 과거 자료를 바탕으로 일에 대한 비용이 산정되어야 한다. 이를 해결하기 위해 여러 방법이 있을 수 있지만, 본 연구에서는 WBS에서 식별된 업무 중 형상항목 개발과 직접적으로 관련된 개발업무를 모두 시간으로 환산하고 각 형상항목에 직접 할당하기 어려운 공통업무는 형상항목 개발과 직접 연관되는 개발업무와의 비율(상관계수)로 시간을 산정하는 방법을 제시한다. 그 절차는 다음과 같다.

① 직접적인 시간산정이 가능한 개발업무를 이용, 공정계획을 수립하고 시간을 산정한다.
② 개발업무를 수행하는 조직에 개발업무별 시간을 할당한다.
③ 개발업무를 수행하는 조직과 공통업무를 수행하는 조직(Cost code)의 시간산정 기준값 및 상관계수를 산정한다.
④ 기준값 및 상관계수를 이용하여 공통업무를 수행하는 조직의 업무별 시간을 할당한다.
⑤ 업무별로 조직에 할당된 시간과 월 비용(Labor rate / month)을 곱해 각 Control Account(CA)를 확정한다.
  - 개발업무 CA = WBS 시간×월비용(Labor Rate)
  - 공통업무 CA = 기준값×상관계수×월비용(Labor Rate)

3.1.2 Earned Value 데이터 측정 및 분석

통합계획 수립 후에는 계획 대비 성과를 측정하고 분석한다. 기성관리시스템의 분석 데이터 및 공식 중 실적진도(BCWP), 계획진도(BCWS), 투입비용(ACWP)은 기성(Earned Value)측정을 위한 기초 데이터이며, 비용(CV, CPI), 일정(SV, SPI), 생산성(TCPI) 데이터는 가

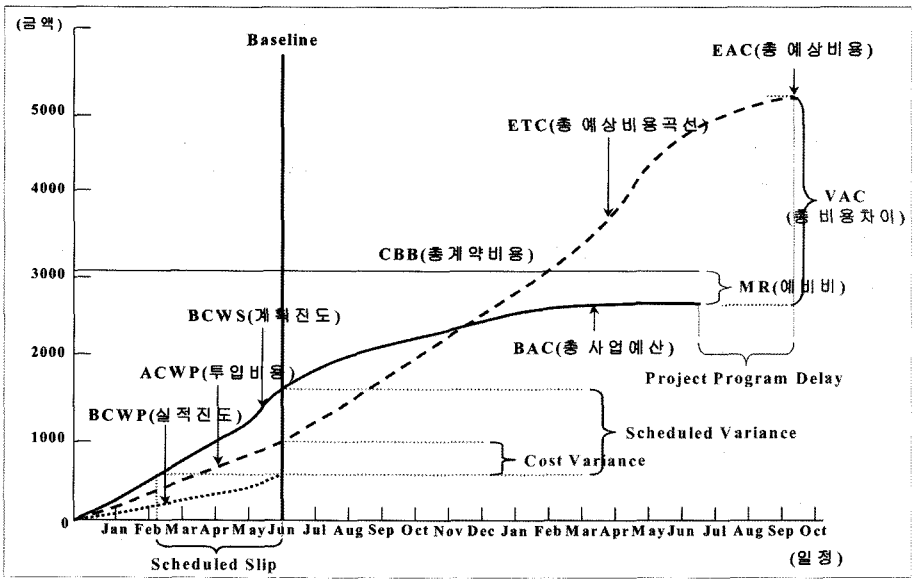

그림 2 Earned Value 데이터 곡선

장 기본적인 데이터이다. 이를 통해 총 사업예산(BAC), 총 예상비용(EAC), 총 비용차이(VAC) 등 나머지 데이터를 추가로 계산하여 프로젝트의 진행현황 및 향후 전망을 진단할 수 있다. 측정 및 분석 데이터는 프로젝트 현황에 대한 직관적인 이해를 제공할 수 있도록 표현한다. 프로젝트와 관련된 현황이 게시되지 않고 있어 프로젝트 참여자가 현재의 상황을 이해하는데 어려움을 겪고 있다. 이는 프로젝트 참여자의 프로젝트 현황에 대한 공통된 이해를 어렵게 하고 이는 문제해결 노력을 결집시키는데 장애가 된다. Earned Value 데이터 곡선은 현황 및 추세 분석을 가시적으로 표현함으로써, 프로젝트의 현황과 전망을 이해하는데 도움을 준다. 측정 및 분석 데이터는 프로젝트 현황에 대한 직관적인 이해를 제공할 수 있도록 표현한다. 설문조사에서도 나타난 바와 같이 프로젝트와 관련된 현황이 게시되지 않고 있어 프로젝트 참여자가 현재의 상황을 이해하는데 어려움을 겪고 있다. 이는 프로젝트 참여자의 프로젝트 현황에 대한 공통된 이해를 어렵게 하고 이는 문제해결 노력을 결집시키는데 장애가 된다. 그림 2 Earned Value 데이터 곡선은 현황 및 추세 분석을 가시적으로 표현함으로써, 프로젝트의 현황과 전망을 이해하는데 도움을 준다.

3.2 스토리보드를 利用한 要求事項 分析

요구사항이 명확히 정의되지 않고 소프트웨어 개발 기간 내내 변경되는 데에는 많은 이유가 있겠으나 다음 두 가지를 주된 이유로 생각할 수 있다.

첫째, 고객은 자신의 업무가 어떻게 소프트웨어로 개발될 지 알 수 없기 때문에 어떻게 만들어 달라고 해야 할 지 명세화 하기가 힘들다. 따라서 고객은 업무 용어를 사용하여 자신의 일을 말한다. 반면에 개발자는 프로젝트 초기 기술적 용어를 사용하여 어떻게 만들 것인가를 고객에게 얘기한다. 즉, 서로 다른 언어로 의사소통을 하는 셈이다.

둘째, 소프트웨어를 개발함으로써 새로운 업무(비즈니스)가 만들어지거나 기존 업무가 변경되는 경우, 프로젝트 초기 고객과 개발자 모두 정확한 요구사항을 알지 못한다. 즉, 개발하고자 하는 대상은 있지만 무엇을 개발할지는 모르는 상태이다.

이를 해결하기 위해서는 첫째, 고객과 개발자가 함께 이해할 수 있는 가장 쉽고 명확한 의사소통 수단을 마련해야 하고, 둘째, 정의한 요구사항을 통해 업무수행이 가능한지를 사전에 검증할 수 있는 수단을 마련하는 것이 필요하다. 초기에 명확하고 완전하게 요구사항을 정의한다면, 개발 기간 중 변경을 최소화할 수 있다는 것은 당연할 것이다.

스토리보드(Story board)를 이용한 요구사항 분석 방법은, 사용자 화면을 이용한 스토리보드를 구성함으로써, 고객과 개발자가 함께 이해할 수 있는 가장 쉽고 명확한 의사소통 수단을 제공하며, 이를 바탕으로 시험절차를 작성하여 요구사항의 완전성을 사전에 검증하는 방법을 말한다. 요구사항 분석 단계에서 완전한 화면설계를 목표로 하며, 이에 따른 시험 절차까지도 기술한다. 여기서 '스토리보드'란 사용자의 업무흐름(business flow)에 따른 화면설계 흐름을 말한다. 예를 들어, 소프트웨어 개발을 위해 객체지향 방법론을 사용한다면, 사용자의 사용사례(Use case)에 대한 화면

설계 흐름을 만드는 것이 이에 해당된다.

최종 사용자 화면은 고객이 자신의 업무가 어떻게 소프트웨어로 구현될지를 살펴볼 수 있는 가장 쉽고 명확한 수단이며, 방법론, 개발언어에 관계없이 일관된 의사소통 수단을 제공하는 장점이 있다. 아울러 개발자 입장에서는 최종 구현 모습을 조기에 확정함으로써 소프트웨어의 논리적 설계, 데이터 설계를 용이하게 접근할 수 있다. 또한 사용자 화면을 이용 스토리보드를 구성하면 업무 흐름을 명확히 파악할 수 있기 때문에 시험절차를 조기에 기술할 수 있으며, 이를 통해 사용자, 개발자 모두 요구사항의 완전성을 쉽게 판단할 수 있다.

대표적인 객체지향 방법론인 RUP(Rational Unified Process)에 본 방법을 결합하면, 화면설계 및 스토리보드 구성은 사용사례 명세서 작성 후 이루어진다. 아울러 화면설계 및 스토리보드 구성 후 각 사용사례에 대한 시험 절차서를 작성한다. 스토리보드와 시험 절차서를 통해 문제기술서와 사용사례 모델(사용사례 다이어그램 및 사용사례 명세서)이 명확하고 완전한가를 사용자와 함께 검증하고 검증 결과를 반영하여 문제기술서 및 사용사례 모델을 수정한다. 스토리보드 및 시험기술서 기술이 완료되면, 이를 이용하여 소프트웨어 요구사항에 대한 합동검토회의를 진행한다. 합동검토회의의 목적은 요구사항의 명확성과 완전성을 검증하는 것이다. 여기서 요구사항의 명확성은 요구사항 내용이 소프트웨어로 구현 가능하고 시험 가능한 가를 의미하며, 완전성은 요구사항을 통해 업무를 완전하게 구현할 수 있는가를 의미한다.

3.3 성과 위주의 프로세스 개선 구현

설문조사의 대상이 되었던 A사의 경우 SPICE(Software Process Improvement Capability dEtermination), CMM (Capability Maturity Model) 심사에서 Level 3을 받은 바 있다. 하지만 설문 결과에 따르면, 조직구성원의 대다수가 소프트웨어 프로세스 개선(SPI)을 통한 품질 향상의 필요성과 효과에 대해 공감하고 있음에도 불구하고, 전체의 67%가 표준 프로세스의 현실성 부족 및 실무적용 애로, 품질관리의 실질적 성과 미흡을 가장 큰 문제로 응답하였다. 이를 해결하기 위해서는 소프트웨어 프로세스 개선활동이 전조직원의 참여 속에 현실적이고 실질적인 성과를 달성하는 방향으로 수정되어야 한다.

3.3.1 Push-Pull approach를 통한 전 조직원의 참여 유도

소프트웨어 프로세스 개선 활동을 SEPG(Software Engineering Process Group)가 아닌 전조직이 참여하는 개선 활동으로 추진해야 한다. 이를 위해서는 소프트웨어 프로세스 개선 활동을 전개하는 접근전략이 중요하다. 만일 최고경영자가 SEPG 등 프로세스 개선 활동의 변화 책임을 담당하는 조직에 비용감소, 결함감소, 라이프싸이클 단축 등과 같은 개선 목표를 제시하고 다시 변화를 책임지고 있는 조직에서 프로젝트 또는 사업팀에 이러한 목표를 달성할 것을 요구한다면, 실제 프로세스 개선활동을 수행해야 하는 각 프로젝트 관리자 및 기술진은 개선 활동에 대한 직접적인 동기를 느끼지 못하게 된다. 이에 따라 시간이 지날수록 개선 활동은 품질경영팀 또는 SEPG만의 활동이 되어간다. 이를 Push approach라 한다. 이러한 접근 방법은 Push-Pull approach로 개선되어야 한다. SEPG에 개선활동을 요구(push)하는 것이 아니라 프로젝트나 사업팀에서 오히려 개선 활동을 위한 도움을 요청(pull)하게 되므로, 개선 활동에 참여를 높이고, 개선활동이 지속적인 활동이 되도록 하는데 효과적이다[5]. 이 때 중요한 것은 SEPG 구성원의 역량이다. 프로젝트나 사업팀의 요청에 대해 프로젝트 및 사업의 현재 프로세스를 이해하면서 실질적인 도움이 될 수 있는 개선방향을 제시하고 도와줄 수 있는 지식과 기술, 경험을 갖춘 인원이 준비되어 있어야 한다.

3.3.2 비즈니스 목표와 관련된 개선 목표의 제시

소프트웨어 프로세스 개선 활동이 cost center가 아닌 비즈니스 이익을 창출하는 활동임을 인식시키기 위해 조직의 비즈니스 목표와 관련된 개선 목표가 명확히 제시되어야 한다. 이를 위해서는 현재 수준에 대한 진단이 선행되어야 하며, 이를 바탕으로 성과를 측정하기 위한 수단을 통해 개선 성과를 꾸준히 관리하여야 한다. 현재 많은 기업이 소프트웨어 프로세스 개선 활동을 진행하는 절차를 보면, 비즈니스 목표와 관련된 개선 목표 대신 도입되는 모델의 능력수준을 목표로 설정한다. 이로써 능력수준을 달성함으로써 얻어지는 이익이 무엇인지 명확히 알 수 없다. 먼저 소프트웨어 프로세스 개선 활동을 통해 어떤 성과를 얻을 것인가를 명확하게 설정하여야 한다. 가장 명확한 개선 목표의 대상은 개선 활동을 통한 비용 절감, 라이프싸이클 단축, 결함 감소 등이다. 개선의 목표값을 설정하기 위해서는 현재의 상태를 먼저 파악해야 한다. 따라서 목표와 관련된 현재 상태를 측정하기 위한 측정값을 설정하고 이에 따라 현재 상태를 측정한다. 현재 상태를 바탕으로 개선 목표값을 설정하고 이를 달성하기 위한 계획을 수립, 시행한다.

3.3.3 현실적인 표준 프로세스의 정의

회사의 비즈니스 환경 및 현재 수행 중인 프로세스를 바탕으로 한 현실적인 표준 프로세스를 정의해야 한다. 프로세스란 주어진 목표를 달성하기 위한 각 단계들의 연속이라고 할 수 있다. 소프트웨어 프로세스는 사람들이 소프트웨어 및 관련된 제품(계획, 설계서, 코드, 시험절차, 매뉴얼 등)을 개발하고 관리하기 위해 사용하는 활동, 방법, 사례, 변환 규칙(transformation)의 집합이다.

CMM, SPICE 등은 소프트웨어 프로세스 개선을 도와주는 모델이지 요구사항이 아니다. 소프트웨어 프로세스 개선이란 비즈니스 목표를 달성하기 위해 소프트웨어 프로세스를 향상시키는 기술과 원리를 적용하는 것이다. 표준 프로세스는 SEPG에 의해 CMM, SPICE 등과 같은 모델을 참조로 하여 쓰여지는 것이 아니라, 회사의 비즈니스 목표를 달성하기 위해 현재 수행 중인 프로세스를 바탕으로 기술자들에 의해 작성되어야 한다[6]. SEPG의 역할은 문서화가 아니라 프로세스 개선을 조율(coordinate)하고, 촉진(facilitate)시키며, 장려(promote)하는데 있다. SEPG는 표준 프로세스를 모든 프로젝트에서 공유할 수 있는 기반을 마련하고 각 프로젝트로부터 표준 프로세스에 사용될 만한 베스트 프랙티스 즉, 자산(assets)을 수집한 후 이를 바탕으로 회사의 표준 프로세스를 정의해야 한다.

3.3.4 지속적인 프로세스 개선 활동의 정착

소프트웨어 프로세스 개선 활동이 일회성, 구호성 활동이 아니라 지속적인 개선 활동으로 정착될 수 있도록 하여야 한다. 또한 개선된 프로세스가 지속성을 갖기 위해서는 내재화(institutionalization)의 시간이 필요하다. 내재화가 이루어지기 위해서는 프로세스가 정의되고(defined), 문서화되고(documented), 사용되고(used), 측정되고(measured), 검증되고(verified), 유지되고(maintained), 지속적으로 개선되는 과정이 이루어져야 한다[7]. 프로세스가 정의된다는 것은 조직원이 해당 프로세스에 대해 설명할 수 있어야 하며, 프로세스 입출력, 업무(tasks), 측정 및 확인이 구성된 상태를 말한다.

4. 결 론

국방부문 소프트웨어는 국가방위의 근간이 되는 Mission critical 소프트웨어로서, 이의 품질은 국가 군사전력에 영향을 미치게 된다. 더구나 지휘통제시스템, 과학화전투훈련시스템 등 국방부문 지휘통제, 작전, 교육 시스템과 관련된 프로젝트는 대부분 그 규모와 개발기간이 매우 크고 길다. 따라서 이러한 프로젝트를 관리하고 개발하는 기술은 민간부문의 그것과는 다른 차원의 중요성을 갖는다.

본 연구에서는 이러한 배경 하에 지금까지 국방부문 소프트웨어 개발 프로젝트의 제약이 무엇이고 이를 어떻게 해결할 수 있을 것인가에 대해 살펴보았다. 설문조사를 통해 식별된 여러 가지 문제 중, 지적활동으로 이루어지는 소프트웨어 개발 프로젝트의 견적(estimation)과 진행현황을 어떻게 가시화할 것인가, 눈으로 보고 만지고 느낄 수 없는 소프트웨어의 요구사항을 어떻게 도출하고 가시화할 것인가, 그리고 소프트웨어의 품질을 향상시키기 위해 소프트웨어 프로세스를 어떻게 개선할 것인가 이상 세가지를 가장 시급히 해결해야 될 과제로 선정하였다.

첫번째 제약인 프로젝트 견적과 공정관리 문제를 해소함으로써 우리는 프로젝트의 견적 및 일정 그리고 비용을 가시적으로 관리할 수 수단을 마련할 수 있다. 특히, 본 연구에서는 Control Account를 산정하여 견적을 수행한 후 '프로젝트 현황판' 을 EVMS 데이터를 프로젝트 일정 및 비용 관리에 효과적으로 활용하는 방안을 제시하였다. 육군 K프로젝트에 적용된 EVMS 효과분석 결과(2000년~2002년)에 따르면, EVMS의 도입은 다음 여섯 가지 측면에서 프로젝트 관리의 개선 효과를 나타냈다[8].

① 범위관리 측면

WBS 개발 및 활용을 통해 프로젝트 관리상의 베이스라인 확보가 가능해졌으며 개발 범위의 한계를 명확히 함으로써 계약요구사항이 충실히 이행될 수 있는지 시기별로 통제할 수 있는 기준을 갖게 되었다. 또한 핵심 공정(Critical Path)을 집중적으로 관리하게끔 하였으며, WBS의 상세한 분해를 통해 작업자의 임무와 역할을 상세하게 부여함으로써 프로젝트 생산성 향상을 위한 기반이 마련되었다.

② 기성관리 측면

기성산정 시 고객과의 불필요한 마찰과 오해를 불식시켰으며, 효과적이고 합리적인 진척율 측정을 가능하게 했다. 이를 통해 프로젝트 성과에 대한 투명한 대가산정이라는 측면에서 개발사와 고객 모두가 만족할 수 있었다.

③ 일정 및 비용예측 측면

매월 월말보고 시 EVMS를 통해 잔여 공정의 일정과 비용을 작업활동 수준까지 예측함으로써 프로젝트 성공의 열쇠인 예산 범위 내 프로젝트 종료 가능여부를 추적, 관리하게 되었다.

④ 개인성과 측정과 평가의 연계

EVMS를 통해 개인별 성과측정 기준이 마련되었다. 이는 개인의 성과가 곧 비용이라는 원가인식의 전기를 제공하였으며, 인사평가 시 객관적인 평가를 위한 기준자료로 활용되었다.

⑤ 고객의사소통 측면

진척율에 따른 기성산정은 근본적으로 프로젝트팀의 성과에 대한 고객의 승인여부에 측정기준, 기성판단기준)를 극복함으로써 궁극적으로 고객 의사소통과 고객만족을 추구하여 수·발주자간 공동의 목표인 프로젝트 성공에 대한 확신을 더욱더 공고히 했다는 측면에서 긍정적으로 이해할 수 있다.

⑥ 위험관리측면

EVMS를 통해 일정, 비용에 대한 현황과 예측이 가능해짐에 따라 프로젝트 관리자는 예상되는 위험(비용, 일정, 기술에 영향을 끼치는 요소)을 사전에 제거 또는 완화 할 수 있는 관리능력을 확보하게 되었다.

두 번째 제약인 요구사항 문제를 해소함으로써 우리는 소프트웨어 개발공정에서 나타나는 재작업과 불필요한 노력, 모호한 제품 및 품질기준에 대한 문제를 해결할 수 있다. 모호한 요구사항을 가시화하고 이를 통해 제품 및 품질기준을 개발 초기에 수립하는 것은 요구사항 단계가 이후 모든 공정에 영향을 미친다는 점을 고려할 때 매우 의미 있는 활동이 될 것이다. 코드를 만들지 못해 실패하는 프로젝트가 많은지 요구사항을 제대로 이해하고 정의하지 못해서 실패한 프로젝트가 많은지 생각해보면 답은 분명해진다. 지난 4년간 육군 K프로젝트를 수행하면서 반복확장개발을 통해 요구사항 문제를 해결할 수 없음을 경험해 왔다. 소프트웨어 개발전략과 관계없이 요구사항을 가시화하기 위한 방법이 필요하다. 본 연구에서 소개한 스토리보드를 이용한 요구사항 분석 방법은 초기에 요구사항을 가시화하고 인수 기준을 수립하는 수단을 제공할 것이다.

세 번째 제약인 소프트웨어 프로세스 개선 활동의 문제를 해결함으로써 우리는 소프트웨어 프로세스 개선을 통한 품질향상 및 비즈니스 목표 달성에 기여할 수 있다. 이는 소프트웨어 프로세스 개선의 실질적인 성과를 통해 본연의 목적을 달성하는 것을 의미한다. 소프트웨어 프로세스 개선 활동은 더 이상 비용을 소비하는 cost center가 아니라 새로운 경쟁력과 이윤을 창출하는 핵심역량이 되어야 한다.

본 연구에서 제시한 개선안은 미 보잉(Boeing)사의 사례를 바탕으로 작성되었다. 보잉(Boeing)사는 1991년 CMM을 도입한 이래 소프트웨어 프로세스 개선 활동을 위해 10,000명의 인력을 교육시켰으며, 200여 팀이 프로세스 심사에 참여하였다[9]. 더욱 중요한 것은 이러한 과정에서 소프트웨어 프로세스 개선 활동의 문제를 인식하고, 개선 활동의 목표가 무엇인지 다시 생각하고, 개선 활동의 주체가 누구인지를 올바로 인식하고, 이를 위해 표준 프로세스를 어떻게 정의할 것인가에 대해 살펴봄으로써, 현재 진행되고 있는 개선 활동이 일회성, 구호성 활동이 아니라 프로젝트의 제약을 해소하는 기초(fundamental)가 되고 새로운 경쟁력을 창출하는 동인(動因)이 되도록 하는데 초점을 맞추었다는 점이다. 이를 통해 보잉(Boeing)사는 상업용 비행기, 위성발사대, 전자상거래 등에 있어 세계 최고의 품질로 비즈니스 목표를 달성하고 있다.

지금까지 살펴본, 이상의 세가지 제약은 현재의 문제를 해결하는 통제점으로, 이를 해결함으로써 다른 문제에 대한 해결방향 또한 손쉽게 정립할 수 있을 것이다. 무엇보다 중요한 것은 항상 프로젝트의 제약이 무엇인가를 이해하고, 프로젝트의 성과와 관련하여 통제점을 관리해야 한다는 점이다. 모든 의사결정을 이를 기준으로 할 때 문제를 효율적, 효과적으로 해결하고 성과를 개선시킬 수 있다.

참고문헌

[1] Watts S. Humphrey(최종섭외 번역), 「소프트웨어로 승부하자」, 학현사, 2004, p. 5.

[2] 박재모, 「국방부문 소프트웨어 개발 프로젝트의 제약 도출 및 해결방안에 관한 연구」, 고려대학교 경영대학원, 2004, pp. 24-50(재구성).

[3] CMMI product team, Capability Maturity Model Integration(CMMI) Version 1.1, Carnegie Mellon Software Engineering Institute, Pittsburgh PA USA, 2002, p. 97.

[4] CMMI product team, Capability Maturity Model Integration(CMMI) Version 1.1, Carnegie Mellon Software Engineering Institute, Pittsburgh PA USA, 2002, p. 127.

[5] John Vu, S/W Process Improvement Journey, KIPA, 2004, pp. 25-26(재구성).

[6] John Vu, S/W Process Improvement Journey, KIPA, 2004, pp. 9-10(재구성).

[7] John Vu, Process Improvement Journey, Carnegie Mellon ISRI, Pittsburgh PA USA, 2003, p. 81.

[8] 김인환 외, 「효과적인 진척관리를 위한 관리체계 구축사례에 대한연구 : WBS구축과 기성고(EVMS)구축을 중심으로」, 쌍용정보통신, 2002, pp. 35-36(재구성).

[9] John T. Grasso, Business Models for Process Improvement, Carnegie Mellon ISRI, Pittsburgh PA USA, 2003, p. 36.

박 재 모

1997~현재 쌍용정보통신 근무 중
1999~2005 육군과학화전투훈련장(KCTC) 시스템 개발 사업
2005 고려대학교 경영대학원(MBA) 졸업
2003 Carnegie Mellon University, Master of Software Engineering(MSE) SEEK course 수료
SPICE(ISO15504) 심사원
E-mail : kenzomo@paran.com