

특집기사

소프트웨어 프로젝트 관리의 체계

이 주 현[†] 고 형 대^{††}

❖ 목

차 ❖

1. 프로젝트 관리의 기능과 범위
2. 프로젝트 관리기술의 체계

3. 프로젝트 관리 방법론과 도구

소프트웨어 개발은 극히 두뇌집약적이고 기술 집약적인 활동이다. 따라서 프로젝트 관리란 소프트웨어 엔지니어들을 조직적으로 관리하며 성공적인 프로젝트의 완성을 위한 활동으로 정의된다.

그러나 품질높은 소프트웨어를 계획되었던 기간내에 성공적으로 개발하기는 너무도 힘들다. 그동안 납기일 지연과 예산초과 현상은 거의 모든 프로젝트의 문제점이었으며 그 결과는 품질보증과 생산성 향상이라는 중대한 목표로 부각되기에 이르렀다. 다양한 개발기법이나 특정 전문가의 경험과 지식보다는 그리고 새로운 언어나 개발환경의 개선보다는 조직력을 강화시켜서 계획을 실행으로 옮기는 구체적인 관리방안이 필연적이라는 현실을 직시하기 이르기까지 너무도 많은 시간이 흘렀다.

특히 우리나라의 경우, 아직도 특정 개발기법이나 도구의 활용을 문제의 해결책으로 삼으려는 경향이 매우 짙다. 그러나 기술과 개발 도구에 의존할뿐 시간과 사람과 돈과 제도의 중요성을

제대로 깨닫지 못하고서는 성공적인 프로젝트 관리는 불가능하다 할 것이다. 관리기술을 발전시키지 않으면 소프트웨어 개발은 여전히 값비싸고 어려운 과제로 남게된다. 본고는 품질높은 소프트웨어를 경제적으로 개발하는데 필요한 프로젝트 관리기술의 체계를 개발적으로 설명함을 그 목적으로 한다.

1. 프로젝트 관리의 기능과 범위

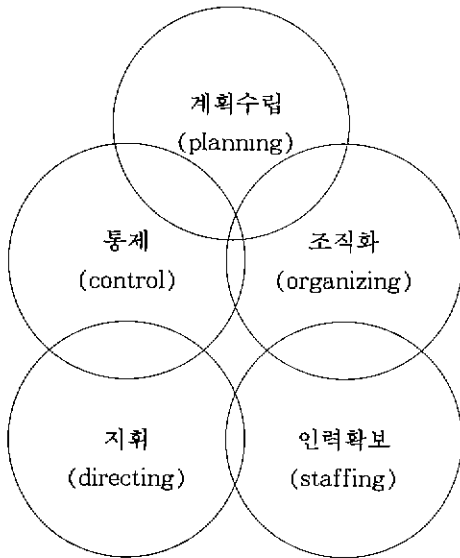
프로젝트 관리란 합의된 시간과 예산 혹은 자원내에서 사용자가 만족할만한 소프트웨어 상품을 개발시키는데 필요한 모든 기술적이며 관리적인 업무이다. 개인보다는 조직의 힘을, 기법보다는 제도를 중시한다. 따라서 프로젝트 관리는 인간관계를 중시하며 관리자의 현명한 의사결정을 필요로하는 사회적, 정치적, 기술적인 문제이다.

경영관리론적 시각에서의 프로젝트 관리는 (그림1)에서도 볼 수 있듯이 다음과 같은 기능을 요구한다.

- ① 계획수립(planning) : 전략(strategic) 및 전술적(tactical) 기능이 약하면 프로젝트 진행의 기초가 흔들린다. 인력,조직, 예산, 자원, 일정계획 등은 미래를 바라보는 (fortelling the future) 힘을 제공한다.

* 본고의 미흡한 부분에 대한 설명은 「실용 프로젝트 관리론」(이 주현, 법영사, 증보판 발행 1994년 8월 1일)과 「전략적 소프트웨어 기술개발계획 수립연구」(과학기술처, 1994년)을 참고하기 바람.
[†]정 회 원 : 한국외대 경영정보대학원 교수
^{††}중 심 회 원 : 목포대 전산통계학과 부교수

- ② 조직화(organizing) : 소프트웨어 개발은 조직적으로 진행시켜야 한다. 개발팀과 개발조직의 균형과 발전은 중요하다. 책임과 권한의 정의는 개발자에게 효율적으로 근무할 수 있는 울타리를 만들어 주는 것(creating the shell)과 다름없다.
- ③ 인력확보(staffing) : 전문기술 인력의 부족 현상은 우수개발자의 확보와 유지를 강조하고 있다. 적합한 인물을 확보하는 것(acquiring the talent)이란 관리의 핵심이다.
- ④ 지휘(directing) : 관리자는 개발자들을 활용하여 프로젝트를 진행 완료시킨다. 개발자들과의 의사소통을 통하여 그들을 효율적으로 이끌어 나가는 것(lead the way)이야말로 두뇌집약적인 소프트웨어 개발의 성과를 좌우하는 열쇠이다.
- ⑤ 통제(contol) : 계획된 작업이 순조롭게 진행되도록 만드는 기능이 통제기능이다. 통제란 진단적, 치료적, 적시적, 경제적, 그리고 정확한 의사결정을 요구한다. 계획으로부터의 이탈을 폐쇄시키는 것(closing the loop)이 통제의 기본이라고 할 수 있다.



(그림1) 프로젝트 관리기능

그러나 과거 대부분의 소프트웨어 개발 프로젝트는 이와 같은 관리기능들의 중요성을 도외시하여 왔다. 각 기능들에 관한 보다 상세한 설명은 다음과 같다.

1.1 개발계획의 수립

계획수립은 무엇을 할 것인가, 어떻게 할 것인가, 언제 할 것인가, 그리고 누가 할 것인가를 사전에 결정함을 말한다. 만약 계획을 수립하지 않는다면 모든 작업은 우연에 맡겨지고 경영이나 관리는 비합리적으로 되고 만다. 따라서 절차를 조심스럽게 결정하고 목적, 지식, 예측에 대한 의사결정의 기준을 신중하게 결정해야 한다. 특히 소프트웨어 개발처럼 불확실성이 큰 프로젝트의 관리는 계획수립이 가장 중요하다. 계획수립의 범위는 다음과 같다.

- ① 개발절차계획(process planning) : 조직과 개발할 시스템에 적합한 생명주기모형(life-cycle model)의 정의와 이와 관련된 산출물(deliverable)의 내역 및 검토회(inspection)의 진행계획을 포함한다. 모든 작업들은 일정계획(schedule)으로 도표화시키는 것도 필요하다.
- ② 문서화계획(document planning) : 각 산출 문서들의 작성방법론을 제시하는 것이다. 각 단계에서 사용될 기법, 표기법, 표준제도를 정의하고 모든 개발자가 이를 준수하도록 명시한다. 개발계획서를 포함한 요구분석, 설계사양서, 시험계획서 등의 문서작성은 작성자가 자신의 모든 지식을 남에게 전달한다는 목표에 충실해야 한다. 또한 문서화계획에는 모든 문서의 결재절차, 보관규정, 검색 및 복사방법 등이 명시되어야 프로젝트의 능률과 보완성을 함께 유지할 수 있다.
- ③ 인원계획(manpower planning) : 소요예측되는 개발자들의 수를 기술수준, 직급, 담당분야별로 산정하는 것이다. 월별로 집계하여 인월(man-month)로 표현하는 것

도 좋다. 책임과 권한을 명확히 하기 위해 개발조직도를 만들어 관리자와 개발자들 간의 상호관계를 정립시켜야 한다.

- ④ 예산계획(cost planning) : 인원계획으로부터 인건비를 주축으로 한 필요예산을 산출하는 것이다. 인건비외에도 직접경비와 간접경비가 함께 고려되어야 한다. 구체적인 예산 계획은 경제적 타당성 분석(economic feasibility analysis)을 요구하기도 한다. 모든 예산 내역은 월별,조직별,항목별로 세분하여 검토하고 계획과 실적이 면밀히 비교될 수 있도록 준비함이 바람직하다.

- ⑤ 개발 환경계획(development environment planning) : 개발자의 가장 큰 관심은 개발 표준제도나 돈보다는 개발 여건의 확립이기 쉽다. 개발장비나 도구, 그리고 기술정보가 훌륭하게 갖추어져 있을 때 비로소 개발에 자신감을 얻기 때문이다. 개발 환경계획은 필요한 장비, 언어, 도구 및 참고문헌 등의 내용 및 범위와 이의 조달계획을 칭한다.

개발계획을 수립함에 있어서 가장 어려운 과제는 흔히 개발 비용과 개발기간을 산정하는 것으로 알려져 있다. 어쩌면 소프트웨어 프로젝트를 실패시킨 주된 원인이 바로 부정확한 비용과 기간의 예측이라고 해도 과언이 아닐 것이다. 따라서 그동안 산정방법들이 제한되어 왔고 최근에는 규모예측(size estimation)을 통한 COCOMO(constructive cost model)이나 기능점수(function point)기법 등이 활발하게 이용되고 있는 실정이다. 국내에서도 이들 기법을 토대로 개발비 산정 기준을 고시하여 정부 및 공공기관에서 활용중이다. 그러나 각 개발조직마다 나름대로의 프로젝트 데이터를 축적하고 이를 이용한 통계적 모형을 발견해야만 현실적인 대안으로 사용가능할 것이다.

1.2 개발조직 관리

소프트웨어 개발은 조직력을 토대로 한다. 프로젝트의 규모가 방대할수록 많은 개발자들이 필요하며 이들의 협동심과 단합된 추진력은 프로젝트의 성패에 큰 영향을 미친다. 조직이란 경영자와 관리자와 개발 담당자들의 상하구조를 정의하고 하부조직간의 관계를 이해시키며 개인과 하부조직들의 책임과 권한을 명확히 하는데 그 목적이 있다. 소프트웨어 개발에 있어 조직이란 개발 관리기능을 활성화시키고 예산과 경비개념을 강조시키며 진행중에 중간 결과가 자연스럽게 보고되어 경영관리자들의 의견이 쉽게 반영될 수 있도록 편성되어야 한다. 또한 각 개발팀내에서의 개발자들간의 단결력이 강화되도록 운영되어야 개발의 능률이 보장된다.

조직화는 일반적으로 분업의 원칙, 책임과 권한의 원칙, 권한이양의 원칙, 계층제의 원칙, 관리와 기술의 이원화 원칙 등을 지켜야 한다. 통합과 행동자유의 원칙, 혁신성과 업무흐름의 원칙과도 합치되면 좋다. 개발조직을 보는 시각은 크게 두가지로 나뉜다.

- ① 개발조직의 유형 : 많은 소프트웨어 프로젝트들을 동시에 추진하는 조직의 효율적인 조직체계는 매우 중요하다. 전문성을 강조하는 직능식 구조, 프로젝트 하나하나를 중시하는 프로젝트 구조가 널리 알려진 가운데 최근에는 두 조직화 기법의 장점만을 살리자는 행렬식 구조(matrix organization)가 인기를 얻고 있다. 국내의 경우, 대부분의 조직은 프로젝트 팀원들로 구성하는 프로젝트 구조체이나 앞으로는 분석, 설계, 테스트, 품질보증등의 전문성을 확보하기위해 직능식이나 행렬구조를 따라야 할 것이다.

- ② 개발팀의 운영 : 프로젝트팀의 일심단결한 협동심을 도모하기 위해서는 효과적인 팀 운영이 필요하다. 팀원위주의 민주주의적 팀(democratic team)보다는 팀장위주의 (leader-oriented team)운영이 일반적이지만 운영의 묘를 찾는 슬기가 없으면 특

정개인의 능력에 너무도 크게 의존하는 상황을 맞게된다.

최근에는 소프트웨어 개발업체나 연구기관의 조직화에 과거의 관료적인 집권적(centralized) 조직보다는 하위계층으로 권한을 이양하여 자주적으로 결정을 내릴수 있도록 하는 분권적(de-centralized) 조직이 발전하고 있다. 또한 사업부제(devisional)조직으로 각 조직이 '기업내의 기업'으로 책임을 지는 양상도 눈에 띄고 있다. 구태의연한 계층적.중앙집권식 조직으로 소프트웨어 개발조직을 바라보던 시각에서 이제 조직력을 강화시키기 위한 보다 현명한 조직구조를 구상하는 시각으로 바뀌어야 할때이다.

1.3 개발인력의 인사관리

소프트웨어는 두뇌집약체이기 때문에 환경보다는 사람이 중요하고 사람이 곧 개발기술을 뜻한다. 그러나 우리나라처럼 우수한 소프트웨어 엔지니어가 부족한 경우 실력있는 개발자들을 배양하기 위한 교육, 훈련, 경험들의 기술축적은 장기간의 많은 투자를 필요로하므로 인력확보 당 시부터 경영관리자의 큰 관심사가 된다. 특히 이직률(attrition rate)이 높은 전산분야에서 업무에 적합한 개발자를 찾고 그들을 유지(retain)함은 고도의 관리능력을 필요로 한다.

인사관리란 전산실 혹은 소프트웨어 개발업체의 목적을 달성시키기 위하여 필요한 인원을 확보하고 이들의 능력을 최대한 발전시킴과 동시에 이를 효율적으로 경영활동에 이용하기 위한 계획적이고 조직적인 모든 절차를 뜻한다. 인사관리는 크게 개별적인 내용과 집단적인 내용으로 구분되는데, 개별적 내용의 핵심관리기능은 다음과 같다.

- ① 신규채용 : 조직계획에 따라 요구되는 인원의 종류와 수에 준하여 새로운 개발자를 모집(recruitment)하고 선발(selection)하는것으로써 직무분석과 직무평가가 우선되어야 한다.
- ② 업무배정 : 채용된 개발자들의 지식과 경험

과 희망에 적합한 일감을 배정하고 하부조직에 배치(placement)시키는 것이다.

- ③ 자기개발 : 많은 프로젝트를 수행해가며 풍부한 경험을 쌓도록 도와줌은 물론 새로운 능력개발을 위해 교육훈련을 지속적으로 실시함을 말한다. 또한 자신의 직업경로를 순탄하게 밟아가며 직장인으로서의 만족감을 찾을 수 있도록 만드는 것도 중요하다.
- ④ 근무평점 : 개발자들의 상대적 가치와 조직내의 유용성을 공정하게 평가하여 노력에 대한 보상을 하기위한 기초로 삼는 인사과과를 말한다.
- ⑤ 승진 및 인사이동 : 자질있는 유능한 개발자들을 관리자로 승진(promotion)시키며 효과적인 인력활용을 위한 인사이동조치를 말한다.

인사관리의 집단적 내용으로는 임금관리, 성과배분제, 복리후생관리, 인간관계관리, 노사관계관리, 종업원제안제도 등이 있다.

일반제조업체의 직능직 사원들을 다루는 문제가 아닌, 소프트웨어 프로젝트의 성공을 보장해주는 우수한 전문가들을 관리한다는 일은 인사과에만 일임할 수는 없다. 경영관리자는 좋은 개발자를 발굴하고 지속적으로 교육훈련시킴과 아울러 일하고자하는 동기부여에 충실해야 할것이다.

1.4 개발인력의 지휘

지휘란 대인관계를 통한 관리기능이다. 하부의 개발자들이 계획에 따라 개발활동을 의욕적이고 적극적으로 수행하도록 동기를 부여하고 감독하는 관리기능을 말한다. 따라서 지휘는 관리자와 피관리자와의 상호인간관계(interpersonal relations)라고 할 수 있다. 피플웨어(perpleware)라고 칭해지기도 한다. 사람이 사람을 다룬다는 것처럼 어려운 일은 없다는 것이 통설이다.

프로젝트 관리자는 개발자들이 집단의 목표를 달성하도록 지휘하기 위해 우선 일하고자 하는 동기를 부여할 수 있어야 하고, 또한 통솔력(leadership)을 발휘하여야 한다. 따라서 동기부

여자(motivator), 통솔자(leader), 평가자(evaluator)의 역할에 충실하기 위해 의사소통 기술을 익혀야 한다.

지휘자의 유형은 다양하다. 독재형(autocratic)이 있는가하면 금전적인 보상을 이용하는 보호형(custodial)이 있다. 그러나 통솔력에 의존하여 개발자들의 목표를 달성할 수 있도록 도와주는 지원형(supportive)이 가장 이상적이다. 물론, 복잡성 극복능력, 조직이해능력, 갈등해소능력, 변화관리능력들이 결비되어야 한다고 한다.

대표적인 지휘방법들은 다음과 같다.

- ① 관리자위주 지휘방법: 한명의 관리자가 프로젝트의 총괄책임자로서 공식적인 위치를 대내외적으로 인정받는 경우에 가능하다. 모든 의사결정을 관리자가 내리며 하부 개발자들에 대한 상벌을 줄 권한도 지닌다.
- ② 팀위주 지휘방법: 팀원 개개인의 의견을 최대한 존중하면서 업무도 배정하고 의사결정도 한다. 즉, 소프트웨어 개발자와 같은 전문가들을 지휘하려면 그들의 개인적 발전을 위해 우선 노력하고 상관과 부하의 관계가 업무효율로 직결된다는 철학이다.
- ③ 목표위주 지휘방법: 관리자에게 업무의 배당권을 일임하되 상하간 대화의 필요성을 강조시키는 절충식 방법이다. 개발자의 업무목표를 관리자가 검토하고 그 결과를 집행시키도록 하는 이 지휘방법은 선진국에서 가장 인기가 있다.

1.5 프로젝트의 통제

소프트웨어 프로젝트를 통제(control)한다 함은 계획에 따라 프로젝트의 목표가 달성될 수 있도록 계획과 대비하여 주기적으로 진행(progress)을 감시하고 성과를 측정하여 만일 계획과 성과사이에 편차가 있으면 이를 수정하는 활동을 말한다.

관리자가 감시하고 통제해야 할 사항들은 개발의 진척관리, 예산과 비교한 비용집행실적관리, 그리고 개발문서와 프로그램의 질을 관리하는 품

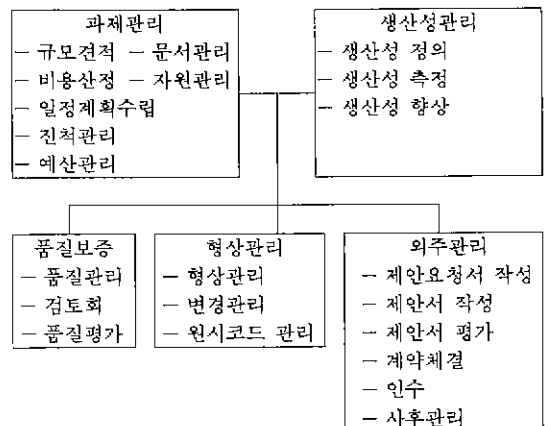
질관리등이 대표적이다. 조직과 개발자의 능률 및 소프트웨어 형상의 변화되는 상태도 감시와 통제의 대상에 속한다.

프로젝트 통제를 위해 한때는 PERT/CPM 기법등의 적용이 시도되었으나 스케줄에 따른 이정표(milestone) 관리, 검토회를 통한 품질관리 및 진척관리등은 이제 특정 기법보다는 방법론의 필요성을 강조하고 있다.

2. 프로젝트 관리기술의 체계

프로젝트 관리의 목표는 넓게는 생산성 향상과 품질보증으로 집약된다. 예산초과와 납기지연이라는 문제점은 궁극적으로는 관리기술로 풀어야 하며 이를 위해서는 소프트웨어의 특수성을 면밀히 검토할 필요가 있다. 프로젝트 관리기술은 (그림 2)에서 보듯이 아래 다섯가지로 분류된다.

- ① 과제 관리: 단일 프로젝트 과제를 성공적으로 진행시키기 위한 계획수립 및 통제 기술을 의미한다.
- ② 품질 보증: 개발상의 중간산출물과 결과물의 품질을 보증하기 위한 조직적이고 제도적인 관리기술이다.
- ③ 형상 관리: 소프트웨어의 변화상을 추적하고 통제의 효과를 누리자는 관리기술이다.
- ④ 외주 관리: 용역업체에 발주하는 프로젝트



(그림 2) 프로젝트 관리기술의 체계

의 성공을 기하기위한 제반관리기술이다.

- ⑤ 생산성 관리 : 개발조직의 생산성을 측정하고 이의 향상을 위한 기술을 도입하는 관리기술이다.

각 기술들에 관한 개요는 다음과 같다.

2.1 과제관리 기술

과제관리(편의상 '프로젝트 관리'라 부를 수 있음)는 단일과제의 성공적인 집행을 위한 활동을 뜻한다. 따라서 규모견적, 비용산정, 스케줄작성등에 필요한 계획수립 기술, 진척관리 및 예산관리등의 통제 기술, 그리고 기타 문서관리와 각종 자원관리 기술등이 과제관리 기술의 체계에 속한다. 타당성분석, 투자효과분석이나 위험(risk)관리기술도 과제관리 기술에 속한다고 할 것이다.

2.2 품질보증 기술

소프트웨어 품질보증(quality assurance)은 어떠한 소프트웨어 제품이 이미 설정된 요구사항과 일치하는가를 확인하는데 필요한 계획적이면서도 체계적인 작업이다. 품질보증은 설정된 표준, 준수해야하는 규칙, 그리고 방법론 및 절차의 확립으로부터 시작한다. 즉 개발환경에 질서와 통제력을 가져오도록 만드는 것이다.

품질보증의 기능은 개발착수 이전의 개발 및 프로그래밍 표준지침서 작성, 관리 표준 지침서 작성, 그리고 지침 준수를 위한 훈련으로부터 시작하여 개발과정중의 요구분석 및 설계안의 검토, 소프트웨어의 시험 및 개발품의 평가로 이어진다. 개발 완료 시험에는 생산에 임하기 전의 공인 기능과 유지보수 및 변경관리를 위한 검토가 필요하다. 이와같은 검토회나 점검목록을 이용한 평가는 품질관리 활동이라 불린다.

2.3 형상관리

소프트웨어 개발관리가 힘든 가장 큰 이유로서 소프트웨어의 무형성을 들 수 있다. 눈에 보이지 않기 때문에 현재를 정확히 파악하기 힘들고 이

에 따라 통제가 불가능하므로 관리가 실패하고 만다. 즉 소프트웨어 특성이 갖는 가시성의 결핍이 문제의 원인이라는 것이다.

소프트웨어 형상관리(configuration management)는 개발과정에 변화되어가는 소프트웨어의 짜임새를 질서있게 통제하고 또한 개발과 유지보수기간동안 나타나는 소프트웨어 변경요구를 관리하는 목적을 지닌 중요한 관리 기술이다. 형상의 식별, 통제, 감사, 기록을 통한 문서 및 원시코드관리는 이제 프로젝트관리기술의 핵심으로 부각되고 있다.

2.4 외주관리 기술

전문기관으로의 외주용역을 통한 위탁개발은 개발기간 단축, 비용절감, 전문기술 활용등의 목적을 위해 광범위하게 실시되고 있다. 그러나 외주개발의 문제점은 관리부재로 말미암아 경제적, 시간적 손실과 정신적 고충으로까지 귀결되는 실패로 너무도 자주 나타나고 있다. 즉, 발주자에게는 발주기술이 절실히되고 개발업체들도 수주기술을 발전시켜야 할 상황에 이르렀다.

대표적인 수발주기술은 다음과 같다.

- ① 제안요청서(request for proposal)작성기술 : 자체 전산화 계획서와는 달리 제안요청서는 용역대상업체들에게 발부해야 하는 문서라는 점에서 우선 외형적 차이가 있다. 내면적으로는 자신의 문제를 적나라하게 설명하여 전문업체들에게 해결책을 제시받아야 되므로 그 구조는 매우 체계적이고 내용은 매우 상세해야 한다.
- ② 제안서 작성 기술 : 제안서란 기술력을 판매하기 위한 영업활동의 기본이다. 발주자측의 이해를 촉구하며 수주를 끌어내기 위해서라면 사용자의 입장에서 모든 사고를 정리해야 한다. 단순한 문서작성능력이 아니라 문제이해력, 대안 평가력, 해결책 제시력등이 필요하다. 조직적인 대응책이나 제안요청서에 대한 철저한 반응전략으로서의 다양한 기법들이 있다.

- ③ 제안서 평가기술 : 접수한 제안서를 공정하게 평가하기 위한 발주기술이다. 객관성을 확보하기 위해서는 조직적인 기술전략이 필요하다. 또한 업체선정을 위한 계량화 방법도 있어야 한다.
- ④ 계약 체결 기술 : 계약서를 단순한 형식적 문서가 아닌 집행가능한 법적문서로서 효력을 갖게하려면 계약서를 작성하고 계약에 임하는 기술이 필요해진다.
- ⑤ 개발 관리 기술 : 개발을 용역업체에 일임하지 않고 철저한 관리를 해야 외주관리에 성공한다. 공동관리를 위한 제도가 바람직하며 형상관리의 철학이 활용될 수도 있다.
- ⑥ 인수 기술 : 접수기준을 설정하고 품질평가에 만족한 결과물로서의 인수작업은 철저한 준비를 필요로 하며 그 기술도 다양하다.
- ⑦ 사후 유지보수 기술 : 자체적으로 개발되지 않은 소프트웨어 사후관리는 개발과정에

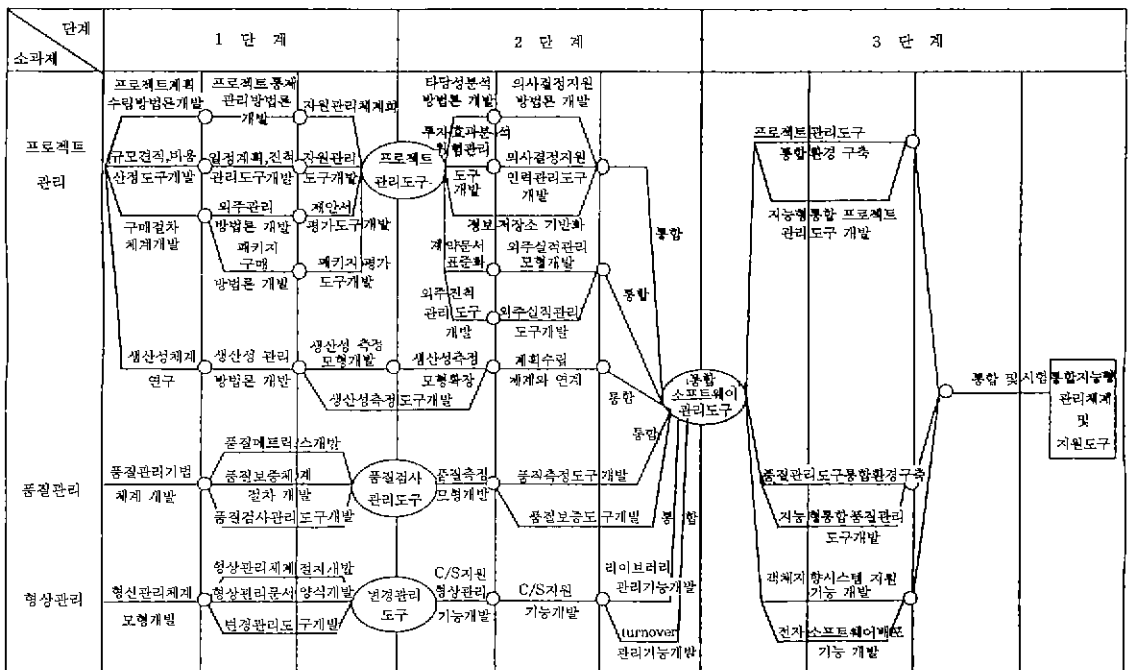
못지않게 힘들다. 사후관리를 위한 대응책을 강구하려면 사전준비를 통한 계획적인 관리기술의 활용이 필연적이다.

2.5 생산성 관리기술

소프트웨어의 생산성은 조직에 따라 큰 차이를 보인다. 그러나 보편적으로 값비싼 인건비에 비해 너무 저조한 까닭에 소프트웨어 위기 의식을 불러일으키고 있다. 그러나 더욱 더 심각한 문제는 대부분의 개발조직이 자신들의 생산성을 파악하지 못하고 있으며 이에따라 생산성 향상은 막연한 목표로만 삼고 있다는 것이다.

생산성 향상 기술은 크게 두가지로 분류된다.

- ① 생산성 측정 기술 : 생산성 척도를 정의하고 이 척도에 준하여 측정하는 기술이다. 안타깝게도 많은 조직들이 척도조차 정의하지 못하고 있는것이 현실이다. 과거 인월당 산출되는 원시코드 스텝수의 개념에서 최근엔 기능점수로 일반화되어가고 있으나



(그림 3) 프로젝트 관리 방법론과 도구 개발 전략

각 조직에 적합한 측정모형이 필요하다.

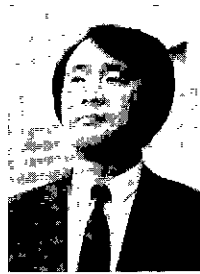
- ② 생산성 향상 기술: 생산성에 영향을 미치는 요인들은 매우 다양하다. 프로그래밍 언어, 개발자의 경험, 개발도구 및 환경, 재사용 정도, 프로토타이핑등의 신기술은 물론, 개발팀의 규모, 이직률, 교육훈련, 개발기간, 팀워크등 계량화시키기 어려운 요인들도 많다. 무슨 기술과 제도가 생산성을 얼마만큼 향상시켜줄 것인지를 밝혀내고 그 결과를 평가하는 것은 프로젝트 관리기술의 핵심이라 할 것이다.

3. 프로젝트 관리 방법론과 도구

개발방법론은 개발자의 작업을 표준화시켜주는 반면 프로젝트 관리의 효율성을 제고시켜주는 시각에서 관리방법론으로 불릴수도 있을 것이다. 따라서 제임스 마틴의 정보공학 방법론을 위시해서 NAVIGATOR, METHOD/1 등 대형정보시스템 구축 방법론등은 다분히 관리의 효율성을 강조하고 있다. 또한 미국 국방성의 방법론을 개조한 국방부의 사업관리 지침이나 TDX 프로젝트 관리를 위한 '국산' 방법론들도 관리의 효과를 중시한다.

그러나 프로젝트 관리방법론을 개발방법론과 굳이 분리시켜 고려했을 경우에도 나름대로의 영역을 정의해 볼 수 있다. 앞서 설명한 프로젝트 관리 기능을 구체적인 지침으로 만들어 관리기술

의 활용을 극대화할 수 있는 전략을 정리하면 (그림 3)과 같다. 우리식의 관리방법론을 정착시키고 이를 지원하는 CASE 도구를 적극 활용하는 것만이 프로젝트의 성공을 보장하는 길이 될 것이다.



이 주 현

- 1974년 (미) 남미시피 주립대학교 전자계산학과 졸업(학사)
- 1977년 (미) 버지니아 주립공과대학교 산업공학과(석사)
- 1983년 (미) 일리노이 공과대학교 경영정보학과(박사)

- 1978년~84년 (미) 벨 연구소 연구원
- 1984년~85년 금성반도체(주) 연구본부장
- 1985년~86년 금성소프트웨어(주) 사업본부장 및 연구소장
- 1986년~현재 한국의국어대학교 경영정보대학원 부교수, 교수
- 관심분야: 소프트웨어 공학, 데이터베이스



고 형 대

- 1982년 전남대학교 계산통계학과(학사)
- 1984년 전남대학교 대학원 전산통계학과(석사)
- 1991년 전남대학교 대학원 전산통계학과(박사수료)
- 1985년~현재 목포대학교 전산통계학과 부교수

- 1992년~현재 목포대학교 정보산업연구소 소장
- 관심분야: 소프트웨어공학, 프로젝트 관리