

# 거시수준에서의 응용시스템 개발 생산성에 관한 연구<sup>+</sup>

이선로\* · 이종민\*

## A Study on Application Development Productivity at The Macro Level

Sun-ro Lee · Jong-min Lee

### 〈요 약〉

응용시스템 개발생산성에 대한 연구는 지금까지 프로그래머의 생산성이라는 미시적 수준에 그치는 경향을 보여왔으며 시스템개발 생명주기의 전반적 영향에 중점을 둔 거시적 수준의 생산성의 개념으로 확대되지 못해왔다. 본 연구에서는 거시적 수준의 응용시스템 개발생산성에 초점을 맞추어 생산성 측정치들을 검토하고 또 다양한 생산성 향상 방법을 비교하는데 그 목적이 있다. 응용시스템 개발 백로그가 Nolan의 정보시스템의 확장 단계에 구분없이, 또한 기타의 컴퓨팅 활동들로 특징지어지는 다양한 정보기술 환경을 가진 조직들 내에서 널리 퍼져 있는 것으로 본 연구결과에 나타났다. 또한 본 연구 결과는 '거시적/관리적 차원의 접근이 응용시스템 개발생산성 향상을 통한 개발 백로그의 감소에 긍정적인 영향을 미칠 것이다' 라는 일차적 가설을 지지하는 것으로 나타났다.

오늘날의 정보시스템 관리자들은 생산성 향상을 위하여 미시적/기술적 차원의 접근방법에 주력하는 경향을 보이고 있으나, 본 연구를 통하여 거시적/관리적 차원의 접근방법과 도구의 사용에 더 많은 노력을 기울임으로써 균형잡힌 정보관리정책을 갖게 되기를 바란다.

---

+ 본 논문은 학술진흥재단 1997년도 신진교수지원 연구비로 수행되었음.

\* 연세대학교 경법학부 경영정보학과

## 1. 서 론

최근의 조사에 따르면 소프트웨어 프로젝트의 35%에서 50%가 개발노력과 일정상에서 계획 일정을 넘어서고 있고[5,7] 그에 따른 프로젝트 개발 포기 및 실패는 심각한 상황에 있다고 보고하고 있다[11]. 그와 같은 소프트웨어의 뒤늦은 인도는 현행 소프트웨어 개발 능력의 세 배를 웃도는 응용시스템 개발의 백로그를 초래하고 있으며[2] 부정확한 프로젝트 관련 사용자 요구사항 및 추정치, 부적절한 일정계획, 그리고 위험을 인지하여 그것에 맞게 계획하지 못한 점 등과 같은 많은 요인들이 예산 및 일정을 넘어서는 데에 대한 근본원인으로 생각되고 있다.

이러한 문제들을 해결하기 위하여 상당한 노력이 소프트웨어 프로젝트 관리기법의 연구에 투입되어 왔다[1]. 반면에, 일부 학자들은 헨리포드가 조립공정 노동자들을 더욱 생산적으로 일할 수 있게 콘베어 시스템을 개발했던 것과 같이, 프로그래머들의 생산성을 총체적으로 높일 수 있는 방법을 찾아내야 한다고 제안해왔다. 예를 들면, Boehm[4]은 소프트웨어 생산성을 증진시킬 수 있는 기회들이 존재한다고 믿고, 95년도까지의 소프트웨어 생산성의 20% 향상은 미국에서만 450억불 그리고 세계적으로는 900억불의 가치가 있을 것이라고 예상하고 있다.

따라서 생산성을 향상시키고 결과적으로 프로젝트 백로그 문제를 완화시키는 방안으로 소프트웨어 개발을 위한 새로운 방법론, 도구 및 프로세스 개선안 등이 제시되고 개발되었다. 대표적인 예들로는 정보공학, Rapid Application Development (RAD), 객체지향적 기법, CASE 도구, 4세대 언어(4GLs), Capability Maturity Model (CMM) 등을 들 수 있다.

그러나 이와같은 새로운 방법론과 도구들이 생산성에 미치는 효과는 명료하게 분석되지 못해왔으며, 때로는 그들의 상반된 연구결과는 논

쟁의 대상이 되어왔다[12]. 선행 연구[3,9,10]는 프로그래밍 생산성 향상은 특정의 단위 기술이나 고립된 정책의 실행만으로는 제한적인 성과에 그칠 수 있다는 것을 보여주고 있다. 즉 정보시스템 관리자들은 새로운 시스템 개발 기술의 도입을 통하여 프로그래밍 생산성(미시적 생산성) 향상에 초점을 맞추어 왔다[13]. 그러나 현재까지 새로운 기술이나 정책들이 프로젝트 특성, 프로젝트 팀 및 시스템개발 과정의 통제유형의 특성을 고려하지 않고 확일적으로 도입되는 경향을 보여왔으며 그에 따른 생산성 향상에 관한 연구의 결과도 일과성을 결여하게 되었다. Deephouse et al.[5]는 응용시스템 개발의 생산성 및 질의 향상은 기술적인 방법과 관리적인 방법이 다양한 형태로 통합되어 시스템 개발 생명주기의 전 과정을 통제할 수 있어야 효과를 거둘 수 있다는 것을 보여주고 있다.

따라서 본 연구는 (1) 응용시스템 개발 생산성에 영향을 미치는 기술적인 측면과 관리적 도구나 방법을 체계적으로 분류하고 (2) 미시적 생산성을 생명주기 전 과정의 생산성(거시적 생산성)차원에서 분리하여 (3) 기술적/관리적 도구와 방법의 통합 효과를 미시적/거시적 측면에서 측정하는데 그 목적이 있다.

본 논문의 구조로 다음 장에서 본 연구의 기본 개념인 소프트웨어 개발 생산성의 개념이 거시적 관점에서 정의되었고, 3장에는 연구 모형 및 가설과 연구 방법이 제시되었다. 4장에서는 조사대상 기업의 정보시스템 환경 분석과 연구 변수가 설명되었고, 5장에서는 분석결과가 정리되었으며 본 논문의 결론은 6장에 제시되었다.

## 2. 생산성의 개념

소프트웨어개발 생산성에 대한 이전의 연구들은 프로그래머 생산성에 초점을 맞추어져 왔으며 많은 연구자들은 시스템 개발 생명 주기 중

한 단계인 코딩 단계의 생산성에 집중하여 연구를 수행하였다. 그러한 프로그래머 생산성에 대한 연구는 전형적으로 한 프로그래머가 시간당 코딩한 코드의 수(Lines of Code: LOC)나 시간당 완료한 기능점수(Function Points: FP)의 수 등과 같은 노력 단위당 생성된 원천 코드의 줄 수 또는 기능점수라는 견지에서 생산성을 측정하였다[10]. 이와 같은 측정치들은 소프트웨어 개발의 다른 단계들에 포함되는 노력을 직접적으로 설명하지 못하기 때문에, 본 논문에서는 그러한 측정치들을 생산성에 대한 미시적 수준의 측정치라고 간주하였다.

더욱이 시스템 분석과 설계 기술이 발전하고 프로그래밍 경험이 증가됨에 따라 프로그래밍(코딩) 자체는 점차 기계적인 과업으로 그 의미가 축소되고 있다. 특히 복잡해진 현대 기업의 전략적 정보 욕구를 만족시킬 응용시스템 개발에 있어서 정보시스템의 전략적 계획단계에서 요구되는 지적인 도전과 창의적 기회는 그 중요성이 더욱 증가하는 반면 단순 프로그래밍은 컴포넌트화된 패키지 도입(예를 들면, ERP 패키지), 코드의 재사용과 표준화된 인터페이스 설치 등으로 외주외탁의 대상이 되고 있다. 과거에는 팀에 기초한 프로그래밍에 프로젝트 노력의 50%이상을 소비했으나 프로그래밍 노력은 최근에는 30%이하로 내려갔으며, 나머지(70%)의 프로젝트 수행 노력은 시스템 개발 생명주기의 다른 단계로 방향을 바꿔 소비되고 있다. 이와 같이 프로젝트 노력의 대부분(70%)이 미시적 수준의 생산성 측정방법에서는 무시되었다.

최근의 조사에 따르면 프로그래머의 시간당 LOC의 수로 측정된(미시적) 프로그래밍 생산성이 획기적으로 증가했는데도 불구하고 연간 백로그는 지난 10년간 줄어들지 않고 있다고 한다[7]. 따라서 전체 개발 생명주기에 걸쳐 생산성 향상의 이유를 밝히기 위해서는 응용시스템 개발 생산성을 거시적(통합적)수준에서 정의하고 또 측정할 필요가 있는 것이다. 또한 생산성 향상을

특정 소프트웨어 개발환경 및 응용형태 등을 고려하여 측정하기 위하여 생산성 측정치들을 상황에 맞게 조정할 필요가 있다. 특히, 주어진 조직 내에서 응용시스템 개발 집단의 연간 수용능력(공급능력)은 조직 안에서의 시스템 개발 및 유지보수에 대한 연간 요구량(수요량)과 비교될 수 있으며 연간 수용능력(공급 능력)의 부족은 연간 백로그로 표현될 수 있다. 따라서 연간 백로그가 줄어드는 장기적 추세를 보일 때 거시적 생산성 향상이 이루어지고 있다고 볼 수 있다.

본 연구에서는 응용시스템 개발 생산성의 향상을 위한 여러 가지 기술적 도구와 관리적 방법의 유용성을 통합적인 관점에서 검증하고자 한다. 따라서 본 연구에서 정의된 거시적 수준의 생산성 측정의 기본 전제는 효과적으로 사용된 방법론과 도구들은 시스템 개발 주기 전체에 긍정적인 영향을 미치게 되어 단위 프로젝트의 생산성 향상으로 연결되고 개발 집단의 프로젝트 수행 주기도 높아져 응용 시스템 개발 백로그의 감소로 이어질 것이다 라는 것이다. 만약 과거 연구에서와 같이 미시적 수준의 생산성 측정치(LOC 또는 기능 점수)를 사용한다면 현재 사용 중인 방법론들과 도구들의 생산성에 미치는 기여도를 심각하게 잘못 판단할 우려가 있다. 그러한 이유는 소프트웨어 개발 생명 주기상 지엽적인 생산성 향상은 생명 주기의 다른 단계에서의 비생산성을 극복하지 못하고 상쇄될 수 있기 때문이다.

전통적인 LOC나 FP와 같은 미시적 생산성 측정치는 정보시스템 부서내의 단위 조직의 단기적 생산성 측정에 제한적으로 사용될 수 있으나 백로그와 같이 정보시스템의 수요와 공급 측면에서 정보시스템 전 부서의 거시적 생산성 및 장기적 효과를 나타내지 못하고 있다. 예를 들어, 정보시스템 관리자가 분석, 설계, 혹은 코딩에서의 향상된 기법을 사용했다고 가정하자. 새로운 기법은 좀더 노동집약적이기 때문에 LOC/시간은 더 나빠져 미시적 수준의 측정치로는 프

로그래머의 생산성이 감소했다고 나타날 것이다. 그러나 새로운 기법은 잘 설계된 소프트웨어를 산출하기 때문에 그 소프트웨어를 사용하는 초기 수년 동안에는 유지보수/개선에 대한 요구를 획기적으로 줄일 수 있을 것이다. 만일 코딩과 유지보수에 사용되는 노동시간의 합계가 기존의 방법하에서 보다 작다면 순수효과는 백로그에서의 감소가 될 것이고 미시적 생산성의 저하에도 불구하고 거시적 생산성의 향상으로 정보시스템 부서 전반적인 생산성은 향상되었다고 보아야 할 것이다.

### 3. 연구 설계

#### 3.1 연구 모형 및 가설

정보시스템 구축 실패는 다양한 요인이 복합적으로 작용하여 프로젝트 추진을 중도에 포기하거나 사용자가 사용하기를 거부하는 결과를 의미한다. 실패의 기술적인 요인으로 저조한 시스템 신뢰성, 기존 시스템과의 호환성 문제, 새로운 기술의 학습 부족, 기존 하드웨어의 과부하 발생 등을 들 수 있고 조직과 관련된 문제로 최고 경영층의 지원 부족, 사용자와의 부적절한 의사소통과 그에 따른 부정확한 요구사항 도출, 사용자의 정보시스템 도입후 권한 변화에 대한 거부감, 사용자의 비현실적인 시스템에 관한 기대 등을 들 수 있다[16]. 또한 프로젝트의 급진적 확대(Project Escalation)는 심리적, 사회적, 그리고 조직상의 장애 요인들에 의하여 프로젝트 관련 의사결정의 일관성을 상실하는 것으로 그러한 급진적 확대는 정보자원의 지속적인 투입에도 불구하고 프로젝트 추진을 중단 해야하는 결과를 초래하는 경향이 있다[11].

또한 프로젝트 실패 및 포기 요인과 급진적 확대 요인의 형태는 상황요인적 성격을 나타내고 있으며 프로젝트 추진 단계에 따라 치명적인 요인이 달라지는 것으로 나타났다. 백로그가 증

가하는 이유로 [그림 1]에서와 같이 두가지 유형을 들 수 있다. 첫째: 정보관리 부서 내 정보시스템 개발 및 유지보수 프로세스상의 기술적 또는 관리적 문제로 프로젝트 실패, 포기 및 확대가 증가하면서 정보 전문 인력과 자원을 문제의 프로젝트에 소모함으로써 새롭게 발생하는 시스템 개발 요구나 유지보수 요구의 백로그를 증가시킬 수 있으며, 둘째: 정보관리 부서 외적인 요소로 기업환경의 불확실성이 증가하면서 응용시스템의 수요가 증가하면서 발생하는 시스템 개발 공급부족으로 백로그가 증가할 수 있다.

프로젝트 추진 실패 및 포기의 결정적인 요인은 기술적이기 보다는 관리적인 측면이 강하게 나타나고 있으며 이를 보완하기 위한 체계적인 프로젝트 관리 기법 도입이 필수적이고[5] 또한 시스템 개발 생명 주기에 통합적으로 영향을 줌으로써 프로젝트 단위의 생산성을 향상시킬 수 있는 거시적 방법이 요구되고 있다. 또한 그러한 거시적 관리적 접근방법[1]이 응용시스템 개발 백로그를 감소 시키는데 보다 효과적이라는 것은 부분적 사례연구를 통하여 나타났다[14].

본 연구의 초점은 시스템 개발 후 사용자 그룹에 인도하고 성공적으로 정착시키는데 필수적인 요인(예를 들면, 정치적 갈등 해소 요인, 최고 경영층의 지원 등) 분석이나 응용시스템 개발 및 유지보수의 수요를 증가시키는 요인의 분석보다는 새로운 응용시스템 개발 및 유지보수 과정에서의 정보 시스템 부서의 생산성을 거시적 차원에서 측정하는데 있다. 따라서 본 연구에서는 응용 시스템 개발 및 유지보수의 백로그의 수준을 거시적 생산성 수준을 측정할 수 있는 대리 변수로 정의하면서 아래와 같이 기본 가설을 설정하였다.

H1 : 거시적 접근방법의 사용과 더 낮은 백로그 수준은 양의 상관관계가 있을 것이다.

1) 생산성 향상을 위한 접근 방법은 본 연구의 독립변수로서 4장에서 상세히 설명되고 있음.

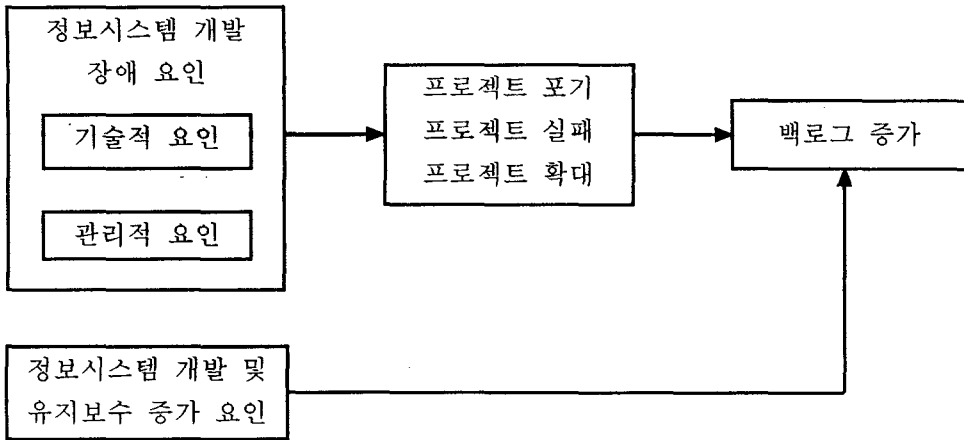


그림 1: 백로그 증가 요인

- H2 : 관리적 접근방법의 사용과 더 낮은 백로그 수준은 양의 상관관계가 있을 것이다.
- H3 : 거시적/관리적 접근방법의 결합사용은 백로그 수준에 가장 큰 양의 효과를 가질 것이다.

수 있었다. 그 중 9개의 설문서는 정의 부족 및 미완성 때문에 사용할 수가 없었고 분석에 사용된 58개 기관의 산업별 분류는 [표 1]에 나타나 있다.

### 3.2 연구방법

본 연구의 선행 연구[14]에서 생산성과 관련 있는 모든 요인들이 포함된 설문서의 초안이 작성되었고 (관심 있으신 분은 연구자에게 연락하여 설문서 사본을 받아볼 수 있음). 예비 설문조사 후 수정된 설문서는 정보시스템의 자체개발 능력을 갖고 있는 무작위 표본인 100개 기관에 우송되었다. 설문서는 회수율을 높이기 위하여 각 조직의 특정한 정보시스템 관리자에게 전달되었다.

그러나 회수율이 매우 낮아서(11개 회수 : 11 퍼센트), 두 번째 설문조사는 본 연구자 소속 대학의 경영대학원 야간 프로그램에 다니고 있는 80명의 학생들(회사의 중견간부사원)을 통하여 80개 기관의 정보시스템 관리자 앞으로 전달되었다. 직간접적인 접촉으로 두 번째 시도에서는 56개의 설문서를 회수할 수 있었고 일차 조사 응답자와 함께 최종 67개의 설문서를 회수할

표 1: 조사응답기관의 산업별 분류

산업구분	기관수
제조	20
금융	14
정보통신	11
유통	6
교육	4
의료	3
계	58

설문 조사 응답자는 정보시스템 관리자(전산실장: 28명), 프로젝트 관리자(팀장: 16명), 시스템분석 및 설계자(11명), 프로그래머(3명)로 전체적으로 대상 기관의 백로그 현황과 기술 및 관리 동향을 파악할 수 있는 직위에 있었다. 설문조사에서 응답자에게 그들의 현재 응용 시스템 개발 백로그의 수준을 개월 수로 표시하도록 하였고 백로그의 증가에 대한 주된 이유가 될 수 있는 요인을 그 정도에 따라 7점 리커트 형태

척도를 사용하여 확인할 것을 요청하였다. 또한 응답자에게 선행 연구에서 생산성 향상에 영향을 주는 것으로 확인되었던 미시적/거시적 수준의 방법들 중 현재 그들의 조직에서 사용하고 있는 것을 확인하도록 하였으며 그들이 사용하는 기법들이 백로그 감소에 어느 정도 영향을 주었는지 7점 리커트 형태의 척도를 사용하여 표기할 것을 요청하였다.

### 4. 실증분석

#### 4.1 정보 시스템 환경 분석

정보시스템 환경을 통제하고 본 조사의 배경을 설명하기 위해서 Nolan[15]의 정보시스템 성장모형을 사용하여 각 조직의 정보시스템의 성장단계를 구분하고 각 조직의 정보시스템 프로파일과 세련화 정도를 검토하였다. 정보시스템의 프로파일과 세련화 요소에는 정보자원 활용방식, 정보관리 지출액, 사용기술, 응용시스템 포트폴리오, 정보관리 조직, 정보시스템 계획수립과 통제, 그리고 사용자 의식 등이 포함되었다. 설문 에 응한 회사의 약 절반이 자사의 정보시스템 지출액이 매출액 성장률과 같은 비율로 성장하고 있다고 대답하였으며 또 그들 중 25% 정도는 자사의 정보시스템 지출액이 매출액 성장률 보다 낮다고 응답하였다.

[그림 2]에 나타나 있는 것과 같이, 소프트웨어 사용에 있어서 대부분 전통적인 프로그래밍

과 데이터베이스 관리에 치중하는 경향을 보이고 있으며, 일부 회사들은 4세대 언어를 사용하고 있으나, 의사결정 모형이나 통계 패키지 사용은 많이 하지 않는 것으로 나타났다. 또한 응답기관의 65%는 공유된 데이터베이스 접근방법을 사용하여 자료를 처리하고 있는 반면에 35%는 아직도 개별화일 접근방법을 사용하고 있는 것으로 나타났다. 데이터베이스를 구축한 기업도 전사적 차원의 정보 공유보다는 특정 사업 또는 업무 분야에 제한적으로 데이터베이스를 활용하고 있었으며 특히 조사대상의 의료 및 교육 분야에서는 전혀 데이터베이스를 구축하지 않고 있었다.

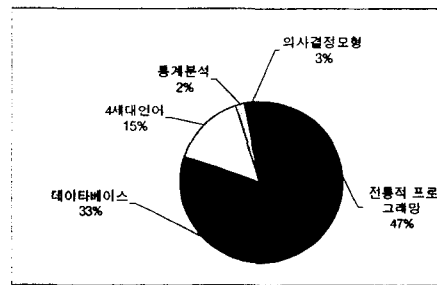


그림 2: 전사적 소프트웨어 사용분포

[그림 3]에 나타나 있는 바와 같이 정보시스템이 사무용 자료 입력 그리고 운영, 통제 등에 대부분 사용되고 있고 전략계획 수립활동 등에는 거의 사용되지 않는 것으로 나타났다. 정보시스템 작업부하는 운영 및 유지보수에 54%가 그리고 시스템 개발에 46%로 나뉘어 걸리는 것

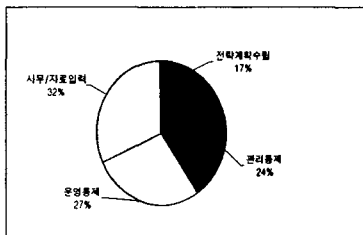


그림 3: MIS가 지원하는 사업활동

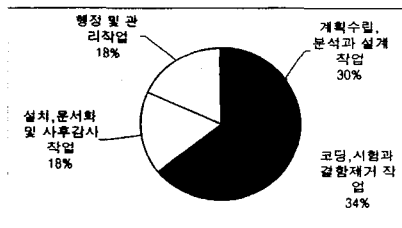


그림 4: 응용시스템 개발 노력분포

으로 나타났다. 개발 노력 면에서 볼 때 [그림 4]에 나타나 있는 것과 같이, 시스템 개발 생명주기 단계에서 코딩과 시험이 여전히 가장 노력이 많이 드는 과업으로 나타났다.

정보관리 지출액 향을 제외한 Nolan의 5개의 수준점지표(benchmarks)가 균등가중선형조합(equally-weighted linear combination)방식에 기초하여 합성점수가 도출되었으며 이를 이용하여 설문에 응답한 조직들의 정보시스템 성장단계를 조사하였다. 그 결과 조사된 조직들은 성장단계 중 3단계와 4단계에 걸쳐 있었으며 어떤 조직도 제1단계에 위치하지는 않았다. 이 사실은 조사된 기관들은 정보시스템 자원을 통합하기 시작하는 위치에 있다는 것을 시사하고 있다.

### 4.2. 독립변수

본 연구에서는 생산성 향상을 위한 방법과 기술을 2가지 형태로 - 기술적/ 관리적 - 분류하였다. 기술적 범주에 속하는 것은 4GL, CASE, 오픈 시스템 등을 들 수 있으며 이들은 새로운 프로그래밍 언어, 정보공학적 기법, 새로운 정보 인프라 등을 통하여 생산성을 향상시키는 것을 목표로 하는 반면 관리적 범주에는 아웃소싱(Outsourcing), 최종 사용자 컴퓨팅(EUC), 과정 통제(Process control) 등 비기술적 수단을 통하여 생산성 문제를 해결하는 형태들이 속할 수 있다.

또한 본 연구에서는 생산성 관련 방법론들과 기술들을 시스템 개발 생명 주기에 영향을 주는 단계에 따라 좀더 세분화시켜 다시 2가지로 분류하였다. 한 범주는 자동코드생성기와 같이 생명주기의 특정 단계의 생산성 향상을 도모하는 미시적(micro) 기법들을 포함하고, 다른 범주에는 새로운 시스템 개발 방법론이나 통합CASE와 같은 생명주기 전반에 영향을 주어 프로젝트 단위의 생산성 향상을 지향하는 거시적(macro) 기법들이 해당된다.

	미시적	거시적
기술적	4GLs Code Reuse Packaged Software Interactive Debugger Screen Editor Testing Tools Upper & Lower CASE	I-CASE Open Systems Object-Oriented Approach Structured Programming
관리적	Outsourcing Contract Programmers EUC Dual-track Careers Configuration Manager	Prototyping RAD Process Control IT Planning Charge-back Systems

그림 5: 현 사용기법의 2x2 행렬에 따른 분류

결과적으로, 본 연구에서는 현재 사용되고 있는 방법론이나 기술 또는 도구들을 4가지 범주로 - 기술/미시적, 기술/거시적, 관리/미시적, 관리/거시적 - 분류하였다. [그림 5]는 본 연구의 선행 연구[14]로부터 도출된 항목들을 4가지 범주로 나누어 배치시킨 것이다. 이와 같은 방법으로 각 항목들을 검토해 봄으로써, 본 연구에서는 각 항목들의 개별효과 및 범주효과를 통합적으로 고찰하는 것을 시도하였다.

일련의 잠재적 외생 변수들은 선행 연구[14]에서 제시된 바 있으며 장단기 계획수립, 정보시스템 부서의 조직구조, 이직률, 프로그래머와 시스템 분석가의 교육수준 및 경험년수, 사용하는 프로그래밍 언어, 분석/설계/코딩/시험 사이에서의 노력의 배분 정도 등 이었다. 그러나 정보기술의 장기계획수립 만이 생산성 향상에 대한 긍정적인 영향을 주는 것으로 나타났다. 또한 과거 연구에서 정보기술의 장기계획수립이라는 요인이 새로운 개발 방법과 도구들을 성공적으로 도입하고 사용하는데 영향을 주는 것으로 나타났다. 따라서 본 연구에서는 장기계획수립의 존재 여부 및 방법에 따라 발생할 수 있는 각 기법과의 상호작용 효과에 대하여도 검증은 수행하였다.

### 4.3 종속 변수

본 연구의 종속변수는 백로그 수준인데 연구자들은 백로그 수준을 요원 당 연중 개발 노력 (Man-years/Employee) 수로써 조작화하였다. 즉 연구자들은 정보시스템 관리자에게 백로그를 없애는데 걸리는 달 수로 표현할 것을 요청하였다. 이 숫자를 년 수로 바꾼 후, 응용개발에 직접 관련된 조직내의 종업원수를 곱하고, 그 숫자를 다시 조직 내에 있는 총 종업원수로 나누었다. 이와 같은 방법으로 본 연구에서는 백로그의 절대적 크기에 조직의 규모가 미치는 영향을 최소화시키기 위하여 백로그 수준을 표준화시켰다.

## 5. 발견사항(Findings)

조사대상 기업들은 거시 혹은 미시 수준의 방법들에 따라 또한 각각의 방법의 사용여부에 따라 분류되었고 선행연구에서 식별되었던 22개항 ([그림 5]에 열거된 항목들 참조) 각각에 대한 단순효과를 일원적 ANOVA 검증으로 분석하였으며 장기계획수립의 각 범주에 대한 상호작용을 통한 생산성향상의 기여도를 측정하기 위하여 이원분석 ANOVA를 추가적으로 사용하였다. [표 2]에서는 ANOVA 분석의 결과가 나타나 있고 그 표에는  $\alpha < .01$  수준에서 유의적인 4개의 항목이 고딕체 활자로 강조되어 있다. 고딕체로 표시된 IT planning, Prototyping, Outsourcing, 4GLs라는 4항목 중 2개는 거시적 범주에 다른 2개는 미시적 범주에 속하기 때문에 "거시적 접근방법이 백로그 수준을 낮추는데 더 효과적일 것이다"라는 첫번째 가설은 현재 단계로는 결론을 낼 수 없는 것으로 나타났다. 4개의 항목 중 3개는 분류기준에 따라 관리적 방법이라고 분류했기 때문에 이것은 두 번째 가설(H2)을 지지해 주고 있다고 볼 수 있다. 또한 백로그 감소에 가장 강한 효과는 4항목 중 2개를 차지하는 거시적/관리적 접근방법(IT planning과 Prototyping)

으로부터 나오는 것으로 볼 수 있으며, 이것은 세 번째 가설(H3)을 지지해 주고 있다. 그러나 이러한 두 결과들은 작은 표본 수에 의존하고 있기 때문에 아직은 임시적이다 라는 것을 지적하지 않을 수 없다. 분명히 이와 같은 결과를 확인하기 위해서 더 많은 검증이 진행되어야 할 것으로 여겨진다.

IT 계획수립은 [표 2]에서와 같이 백로그를 감소시키는데 가장 영향력 있는 방법으로 나타났다. 따라서, 우리들은 각각의 항목과 IT 계획수립 간에 있을지도 모를 상호작용을 이원 ANOVA분석으로 검토하였으나 어느 상호작용도  $\alpha < .01$  수준에서 유의적이지 않았다. 이는 전체 표본 수가 작고 또 분류기준에 따라 분류된 범주에 속한 표본 수가 달라서 신뢰도에 문제가 있고 보다 광범위한 표본 수집과 범주에 따른 표본 수의 균등화는 새로운 결과를 제시할 수도 있다고 보여진다.

다음 단계로 Earl[6]의 정보시스템 계획수립 방법의 분류 - 사업주도형 (business-led), 방법주도형 (method driven), 관리강조형 (administrative), 기술강조형 (technological), 조직강조형 (organizational) - 에 따라 각 계획수립 접근방법의 조직에서의 사용빈도수를 조사하여, 백로그에 대한 영향을 검증하였다. 사용방법의 빈도수 조사 결과는 사업주도형 접근방법을 사용하는 조직이 16개로 가장 많았으며 기술강조형이 14개 조직으로 그 뒤를 이었다. 반면에 조직주도형은 6개 기업으로 가장 관심이 없는 방법으로 나타났다. 그러나, Earl의 5가지 IS계획수립 접근방법들은 모두 백로그상에 유의적인 차이를 보이지 않는 것으로 판명되었다. 또한, Nolan의 단계별 성장모형에 나오는 수준점지표(benchmarks)에 따라 조직을 분류하여 백로그에 대한 그들의 영향을 검토하였으나 각 수준점지표들은 백로그와 유의적인 상관관계를 보이지 않는 것으로 나타났다. 따라서 본 연구자들은 백로그가 정보기술 성장의 단계상 성숙된 조직이나 그 중간에 있는 조직이나



표 2: 각 방법의 백로그에 대한 영향을 일원 ANOVA 로 분석한 표

사용된 방법	F	P	df	사용한 회사의 수
Integrated debugger	0.00057	0.9811	1,56	18
Screen editor	2.24673	0.1395	1,56	33
Testing tools	0.94037	0.3364	1,56	15
Configuration manager	0.57270	0.4524	1,56	14
I-CASE	0.66863	0.4170	1,56	17
Upper CASE	0.23049	0.6330	1,56	11
Lower CASE	0.54890	0.4619	1,56	9
4GL	11.14503	0.0015	1,56	30
Open systems	0.06768	0.7957	1,56	32
Structured programming	0.29832	0.5871	1,56	43
Prototyping	27.89042	<0.0001	1,56	19
RAD	0.00064	0.9799	1,56	11
OOD	0.07458	0.7858	1,56	20
Contract programmers	0.22558	0.6367	1,56	14
Packaged software	0.56408	0.4558	1,56	30
Outsourcing	70.32949	<0.0001	1,56	11
Code reuse	1.81934	0.1828	1,56	13
EUC	0.11763	0.7329	1,56	32
Process Control	0.97910	0.3267	1,56	12
IT Planning	27.72737	<0.0001	1,56	22
Career Alternatives	0.00402	0.9496	1,56	12
Charge-back	0.20460	0.6528	1,56	10

구분 없이 널리 퍼져 있는 일반화된 문제라고 믿게 되었다.

정보시스템 관리자들이 답한 현재까지 백로그의 계속적 증가에 대하여 중요 요인을 파악하고 그러한 요인이 어느 정도 백로그 문제를 야기시켰는지를 분석하였다. 응답자들이 백로그의 증가를 초래했다고 지목한 항목들을 중요도의 순서에 따라 열거하면 다음과 같다.

1. 상이한 아키텍처로 전환할 경우
2. 시스템 개발 중 요구사항의 변경이 있을 경우
3. 시스템 인도후 많은 부수적 변경요구가 있을 경우  
(상황변화적 질의 및 새로운 보고서 형식의 잦은 변경)
4. 최종 사용자 컴퓨팅의 증가로 인하여 정보 관리 부서의 지원 수요가 증가한 경우
5. 사업환경이 크게 변화했을 경우
6. 정보기술에 대한 향상된 인식으로 더 정교한 시스템 개발 요구가 증가한 경우
7. 사내에서 업무재설계(process reengineering) 등의 조직 재구조화가 이루어질 경우

8. 예산삭감이 있을 경우

위에 제시된 첫 번째 요인은 조사에 응한 조직이 구(舊) 정보기술 인프라 구조에 근거해서 새로운 응용을 개발하기 보다는 자사의 정보기술 인프라 구조를 새롭게 변경하고 있으며, 조직의 인력을 새로운 정보기술 하부구조 개발과 시스템 통합 작업에 집중 투입하고 있음을 나타내 주고있다. 그럼에도 불구하고 거시적/관리적 방법이 백로그 감소에 보다 긍정적인 효과가 있다는 본 연구 결과는 거시적 방법과 관리적 방법의 통합이 정보시스템 관리자들이 지적한 대부분의 문제들을 백로그 해소 차원에서 해결할 수 있는 방법인 것을 보여주고 있다. 특히 내부적/외부적 시스템 요구사항의 변경과 관련된 문제가 발생했을 경우에는 그러한 통합적 접근방법이 좋다는 것을 보여주고 있다.

정보시스템 관리자들은 현재 사용되고 있는 여러 가지 방법에 표시할 것을 요청받았으며, 앞으로 사용될 방법에도 표시를 하여 기존방법과 혼동되지 않도록 분리시켰다. [그림 6]은 설문에 응답한 조직에서 현재 사용되고 있는 각각의 방법에 대한 빈도수 및 순위가 나타나 있다.

	미시적	거시적
기술적	4GLs(30,5) Code Reuse(13,15) Packaged Software(30,5) Interactive Debugger(18,10) Screen Editor(32,2) Testing Tools(15,12)	I-CASE(17,11) Open Systems(32,3) Object-Oriented Approach(20,8) Structured Programming(43,1)
관리적	Outsourcing(11,18) Contract Programmers(14,13) EUC(32,3) Dual-track Careers(12,16) Configuration Manager(14,13)	Prototyping(19,9) RAD(11,18) Process Control(12,16) IT Planning(22,7) Charge-back Systems(10,21)

그림 6: 현재사용되고 있는 방법의 빈도수 및 순위

현재 사용중인 방법 중 오직 세가지 관리적

방법 「EUC(3), IT계획수립(7), 프로토타이핑(9)」만이 상위 10위 안에 포함되었다. 이것은 아직도 설문대상 조직들이 관리적 방법 보다는 기술적 접근방법을 선호한다는 것을 암시하고 있다. 본 연구에서 정보시스템 관리자들은 아웃소싱(outsourcing)이 백로그 감소에 긍정적인 영향을 줄 것으로 인식하고 있음에도 불구하고실상 그들은 아웃소싱을 조직내에 적극적으로 정착시키기를 주저하고 있다는 것은 흥미미있는 결과이다. 요약하면, 응용시스템 개발 생산성에 대해 관리자의 인식과 실제로 사용하는 방법 사이에는 커다란 차이가 있다는 것을 알 수 있다. 정보시스템 관리자들은 관리적 해법을 통하여 생산성 향상을 추진해야 한다는 것을 충분히 이해하고 있으면서도 그들이 가시적인 기술적 접근방법을 채택하여 단기적 해답을 얻고 있다는 것은 조직 문화적 측면에서 연구가 필요한 점이다.

6. 토의 및 결론

응용시스템 개발 생산성에 관한 기존의 연구는 단편적인 기술 또는 방법론의 도입에 따른 효과 측정에 국한되어왔다. 더욱이 그들의 생산성 측정은 미시적 측정단위를 사용함으로써 조직 내 응용시스템 개발요구의 백로그의 증가라는 거시적 문제 해결과는 거리가 있어 왔다. 본 연구는 정보시스템 조직의 생산성 향상에 기여할 수 있는 도구 및 방법이 실무적으로 어떻게 사용되고 있고 그 효과가 어느 정도인지를 관찰하는 탐색적 연구로써 응용 시스템 개발 생산성에 기술적 방법과 관리적 방법의 통합 효과를 체계적으로 분석하는데 그 목적을 두었다.

본 연구에서 응용개발 백로그가 Nolan의 수준점지표와 기타의 컴퓨팅 활동으로 구별할 수 있는 다양한 IT환경에 처해 있는 조직들에 넓게 퍼져있음을 확인하였다. 본 연구의 결과는 일차적인 가설이었던 거시적/관리적 수준의 접근방법이 응용시스템 개발 생산성 이라는 문제를 다

루는데 더 효과적이라는 것을 지지하고 있다. 그러나 관심의 대상이 되었던 접근방법에 응답한 회사의 수가 작았기 때문에 백로그 감소에 긍정적인 영향을 주는 정보기술의 장기 계획 수립, 아웃소싱, 프로토타이핑, 4GL 등이 기타의 접근방법에 상호작용을 통하여 어떻게 추가적 기여를 하는지 분석하기 어려웠다. 불행하게도 아직도 연구대상이 되었던 많은 조직들은 현업에서 기술적 미시적 접근방법을 관리적 거시적 접근방법보다 선호하는 것으로 드러났다. 따라서 본 연구를 통하여 정보시스템 관리자들이 거시적 방법 및 도구들을 사용하고 정보시스템 계획 수립에 대한 더 많은 관심을 갖도록 확신시킬 증거를 제시할 것을 기대하고 있다.

본 연구에서 선행 연구[14]에서 수행했던 정성적인 분석의 결론을 정량적 분석을 통하여 확인할 수 있었으나 이 분야의 연구에서 다음과 같은 영역은 더 많은 연구가 진행되어야 한다고 생각한다. 첫째로, 응용시스템 개발의 백로그에

대하여 각 범주안에 있는 각 사용기법들이 주는 영향을 좀더 공식적으로 검증할 수 있는 더 많은 자료가 요구된다. 둘째로, 응용시스템 개발 및 시스템 통합 파라다임에 획기적 영향을 끼치고 있는 ERP패키지가 응용개발의 백로그에 주는 영향을 분석할 필요가 있다. 셋째로, 관리적 거시적 방법의 효과가 외국 및 다른 문화권에 서도 일관성 있게 나타나는지를 확인해볼 필요가 있다고 생각한다. 현재 본 연구의 후속 연구는 한국과 홍콩 사이의 비교 단계에 와 있다. 그러나 이 비교는 북미 또는 유럽 국가들로 확장될 필요가 있다고 생각한다. 후속 비교 연구를 통하여 실무에서 사용하고 있는 접근방법에 대한 데이터베이스를 넓혀감에 따라, 본 연구의 발견 사항에 대해 확신을 갖게 될 것이며, 또 이것은 생산성 향상에 기여하는 실무적 접근방법의 효과성에 대하여 지역 및 문화가 갖고 있을지도 모를 영향에 대한 이해를 촉진시킬 것이다.

### 〈참고문헌〉

- [1] Abdel-Hamid, T., and S. E. Madnick, *Software Project Dynamics: An Integrated Approach*, Englewood Cliffs, NJ: Prentice-Hall, 1991.
- [2] Alloway, R. M. and J. A. Quillard, "User managers' systems needs", *MIS Quarterly* 7(2), 1983, pp. 27-41.
- [3] Banker, R., Datar, S., and Kemerer, C., "A model to evaluate variables impacting the productivity of software maintenance projects," *Management Sciences*, Vol. 37, No. 1, 1991, pp. 1-18.
- [4] Boehm, B. W., "Improving software productivity", *IEEE Computer* 20(8), 1987, pp. 43-58.
- [5] Deephouse, C., Mukhopadhyay, T., Goldenson, D., and Kellner, M., "Software processes and project performance," *Journal of Management Information Systems*, Vol. 12, No. 3, 1995-1996, pp. 187-205.
- [6] Earl, M., "Experiences in strategic information systems planning," *MIS Quarterly*, Vol. 17, No.1, 1993, pp. 1-24.
- [7] Genuchten, M. V., "Why is software late? An empirical study of reasons for delay in software development", *IEEE Transactions on Software Engineering* 17(6), 1991, pp. 582-590.
- [8] Henderson, J. C. and S. Lee, "Managing I/S design teams: A control theories perspective", *Management Science* 38(6), 1992, pp. 757-777.

- [9] Jeffrey, D. R., "Software engineering productivity models for management information system development", in R. L. Boland and R. A. Hirscheim (eds.), *Critical Issues in Information Systems Research*. New York: Wiley, 1987, pp. 113-134.
- [10] Jones, C., *Programming Productivity*, New York: McGraw-Hill, 1986.
- [11] Keil, M., "Pulling the plug: software project management and the problem of project escalation," *MIS Quarterly*, December 1995.
- [12] Kemerer, C. F., "How the learning curve affects CASE tool adoption", *IEEE Software* 9(3), 1992, pp. 23-28.
- [13] Klepper, R. and Bock, D., "Third and fourth generation language productivity differences, *Communications of The ACM*, Vol. 38, No. 9, 1995, pp.69-79.
- [14] Lee, S. and Schmidt, R., "Improving application development productivity: A Hong Kong study", In J. Burn and M. G. Martinsons (eds.) *Information Technology and the Challenge for Hong Kong*, University of Hong Kong Press, 1997.
- [15] Nolan, R., "Managing the crisis in data processing," *Harvard Business Review*, March-April, 1979, pp. 115-126.
- [16] Poullymenakou, A. and Homes, A., "A contingency framework for the investigation of information systems failure," *European Journal of Information Systems*, Vol. 5, 1996, pp. 34-46.