

CASE Tool의 기능과 생산성에 관한 실증적 연구 —우리나라 시스템전문가들의 인식을 중심으로—

김 호 석*, 김 병 철**

A Study on Functions and Productivity of CASE Tool

Computer-Aided Software Engineering or CASE is the technology for automating the software development and maintenance process. Many firms are turning to CASE in hopes of realizing improvement in productivity and system quality. Although expenditures on CASE were reported to be rapidly rising, the performance gains have been very difficult to measure.

The empirical results in this article are intended to provide some evidence about (1) the major objective of using CASE, (2) the gains from CASE, (3) the difference in productivity among CASE tools. They are based on an analysis of data collected from 61 software engineers from five corporations who have the experience of using CASE.

I. 서 론

1. 연구의 목적

최근 전세계적으로 컴퓨터 관련산업은 빠른속도와 대용량의 처리능력 그리고 네트워크 관련기술의 발전으로 과거와는 다른 형태의 환경을 맞

이하고 있다. 그러나 H/W는 상당한 속도로 발전되고 있으나 그에 따른 S/W 개발의 발전속도가 뒤따르지 못하고 있어 '소프트웨어의 위기'라는 용어가 등장하고 있다. S/W의 수요는 연간 25% 이상 증가하는 추세에 있고 요구사항을 반영하기 위한 기능추가가 많이 되어 하나의 시스템에 필요로 되는 프로그램의 평균코드수는 매년

* 중앙대학교 경영대학

** 중앙대학교 대학원 경영학과

25% 이상 증가되어 3년마다 그 수요는 배로 증가되는 반면에 지난 20년간 프로그래머의 생산성은 연간 4-6%의 성장에 머물고 있었다[조선형, 1988].

이러한 S/W의 생산성 문제를 해결하기 위한 방안의 연구로서 소프트웨어 공학의 개념이 나타났고, 그러한 개념을 적용하여 소프트웨어 개발의 자동화를 실현시킬 목적으로 CASE(Computer-Aided Software Engineering)가 출현하였다. 많은 기업들이 S/W 비용을 낮추고 생산성 향상을 위해 CASE에 눈을 돌리고 있으나 CASE의 효과에 대하여는 일관된 결과를 제시하지 못하고 있다. 예를 들면 IBM은 자사의 AD/Cycle을 사용할 경우 20% 내지 30%의 생산성 향상을 주장하는가 하면[Sperling, et al, 1989] SONY는 제한된 응용업무에 적용한 결과 600%의 효과를 보았다고 주장한다[Gabel, 1989]. 그러나 반면 CASE를 도입하여 전혀 생산성에 기여할 수 없었다는 연구결과도 발견할 수 있다[McGuff, 1989]. 미국의 Software Magazine이 CASE를 사용하고 있는 196개의 미국 기업을 대상으로 조사한 결과 생산성을 측정하는 프로그램이 없는 기업이 74%나 되는 것을 보면[Knight, 1989] 아직도 많은 기업이 CASE의 효과에 대한 측정조차 시작하지 못하고 있는 실정이다.

본 연구는 현재 우리나라에서 사용중인 CASE를 대상으로 하여 사용실태를 조사한 후 CASE에 대해 다음과 같은 핵심적인 질문에 답을 제시하고자 함에 있다.

첫째, CASE의 도입목적은 무엇인가?

둘째, CASE는 과연 S/W의 생산성 향상에 기여할 수 있는가? 있다면 어떤 기능이 생산성 향상에 기여하고 있는가?

셋째, CASE Tool 별로 생산성 기여도에 차이가

있는가?

넷째, 앞으로 CASE는 어떤 방향으로 발전되어야 하는가?

2. 연구방법론

본 연구는 CASE의 도입의 효과에 대한 기초적인 실증자료를 제공하는데 있다. 대부분의 기업이 생산성 측정수단이 없는 상황에서 CASE를 도입함에 따른 생산성 효과를 직접 측정하기가 어려워 본 연구에서는 실제 CASE를 이용하여 시스템 개발을 담당하는 시스템전문가를 대상으로 이들이 인식하고 있는 CASE의 생산성을 간접적으로 측정하였다.

표본선정을 위하여 국내에 CASE를 도입한 업체를 조사한 결과 30여개 업체로 파악되었으며 대부분의 이들 업체들은 CASE를 도입하여 사용빈도가 얼마되지 않은 상태이었다. 본 연구에서는 실증분석 대상으로 최소한 1개 이상의 프로젝트를 개발한 경험이 있는 업체만을 대상으로 하였으며, 응답자의 경우도 CASE의 각 기능을 모두 사용한 경험이 있는 자로 한정하여 표본으로 선정된 업체는 6개 업체에 총응답자는 83명이었다.

설문방법은 CASE를 사용하는 팀의 실무책임자에게 설문의 취지를 설명한 후 설문내용의 의문점에 대하여 질문을 받은 후 설문을 전달하고, 최소한 일주일의 여유를 주어 설문을 회수하였다.

앞에서 언급된 연구목적을 달성하기 위해 다음과 같은 접근방법을 이용하였다.

첫째, 설문과정에서 응답한 사용자들을 CASE의 종류에 따라 분류하고 CASE가 제공하는 기능들에 있어서의 생산성 기여도에 대한 인식이 분류에 따라 차이가 있는지를 Ho-

telling의 T²을 이용하여 검증한다. 또한 개별적인 기능에서의 차이도 검증하여 본다.

둘째, 같은 종류에 속하는 제품일지라도 CASE가 제공하고 있는 기능들에 있어서의 생산성 기여도에 대한 인식이 제품에 따라 차이가 있는지를 Hotelling의 T²을 이용하여 검증한다. 또한 개별적인 기능에서의 차이를 검증하여 본다.

셋째, 여러종류의 CASE가 공통적으로 제공하고 있는 세부적인 기능이 보다 큰 기능으로 대별될 수 있는지를 알아 본다. 이를 위해 각 기능의 생산성 기여도에 대한 비교우위의 상관관계를 분석한 후에 높은 상관관계를 가지는 기능을 묶어 이들에 대한 정의를 한다. 분석방법은 인자분석(Factor Analysis)을 사용한다.

넷째, 설문조사 내용을 검증하고 해석하기 위하여 각 CASE 사용자별로 비교적 CASE 사용경력이 오래된 기관을 선정하여 면접을 실시하였다. 또한 설문조사와 면접내용을 기초로 하여 앞으로 CASE가 어떤 방향으로 발전되어야 할 것인가를 제시한다.

II. 이론적 배경

1. 시스템 개발 스펙트럼의 변화

정보시스템 관리업무중 가장 어려운 일종의 하나가 새로운 시스템의 개발로 인식되고 있으며 1960년대 이래 시스템 개발을 지원할 수 있는 도구(tool)와 방법론에 대하여 많은 연구가 진행되어 왔다.

지난 30여년 동안의 발전과정을 요약해 보면 1970년대에 들어 오늘날까지 가장 널리 이용되

고 있는 전통적인 “시스템 개발 수명주기”가 등장하여 개발과정을 현저하게 개선하였다. 이와 같은 전통적인 접근방법은 용어와 각 단계별 활동내용에 있어 방법론에 따라 약간의 차이가 있긴 하나 이들의 공통적인 특징은 3세대 언어를 이용하여 전문적인 프로그래머에 의해 주로 메인 프레임 응용시스템을 중심으로 “구조화 프로그래밍” 개발방법론에 의해 개발된다는 점이다.

70년대 후반에는 시스템 개발에 있어 또 다른 추세가 나타났는데 이는 시스템 개발 수명주기의 앞단계에 보다 많은 노력을 투입하려고 하는 시도였다. 이와 같은 현상은 시스템 개발에 있어 두가지의 새로운 접근방법에서 나타나고 있는데 하나는 데이터중심 개발방법론들의 등장이고 또 다른 하나는 사용자들의 요구사항을 보다 빨리 정확하게 정의하기 위한 그룹설계 방식의 출현이었다.

80년대 초에는 시스템 개발에 있어 크나큰 변화의 움직임이 있었는데 과거의 COBOL이나 PL/1 중심의 3세대 언어로부터 4세대 언어(4GLs)의 사용이 증가하기 시작하였고 전통적인 시스템 개발 수명주기 이론으로부터 반복적인 개발을 지향하는 프로토타이핑(prototyping) 기법이 4세대 언어와 함께 사용되기 시작하였다. 과거 6-7년 동안 4세대 언어와 프로토타이핑에 관하여 많은 논란이 있어 왔으며 과연 이러한 기법들이 전반적인 생산성을 증가시킬 수 있는지에 관하여는 많은 의문이 제기되었다[McNurlin, et al, 1989]. 결국 소프트웨어 개발과 유지보수의 생산성은 개발과정과 오류의 체크 자동화를 가능하게 하는 표준화된 방법론에 의존하지 않으면 안된다는 사고하에서 탄생한 것이 80년대 말에 나타나기 시작한 CASE 이다. 따라서 CASE의 기본사상은 실무에서 적용하기 어려운 구조화 기법을 현실적으로 가능하게 하는데 있다.

2. CASE의 분류

CASE란 'Computer-Aided Software Engineering'의 약어로서 소프트웨어 개발 수명주기(SDLC : Software Development Life Cycle)의 전과정을 자동화하기 위한 방법을 제시한 것이다. 즉, 구조화 기법의 개념을 적용하여 프로그램의 구현 및 유지보수를 위한 도구의 역할을 하고 분석 및 설계단계에서의 수동적인 방법론들을 결합시켜 소프트웨어 개발주기를 자동화하는 것이다.

CASE의 기본사상은 SDLC의 전 단계에 걸쳐서 상호연결하여 소프트웨어의 생산성을 높이는 데에 있다. 즉, SDLC의 시작부터 끝까지 되풀이되는 분석과 설계업무를 자동화함과 동시에 프로그래밍과 유지보수 및 문서화 업무의 자동화를 통하여 소프트웨어의 생산성을 제고시키고자 하는 것이다.

현재 사용되고 있는 CASE 툴은 상위 CASE(Upper CASE), 하위 CASE(Lower CASE) 그리고 통합 CASE(I-CASE : Integrated CASE)로 나누어 볼 수 있다. 상위 CASE란 시스템 개발수명주기의 초기단계인 분석과 설계를 지원하는 툴을 말하며 하위 CASE는 개발수명주기의 후기단계인 코드생성과 테스트를 지원한다. 반면 통합 CASE(I-CASE)는 수명주기의 초기와 후기단계를 동시에 지원한다.

Ⅲ. 연구설계

CASE의 기능과 생산성에 관한 실증적 분석을 위하여 우리나라 기업중 CASE를 도입하여 사용하고 있는 업체에 대한 설문조사와 면접을 실시하였는데 그 내용은 다음과 같다.

1. 표본설계

CASE가 제공하는 기능을 문헌 연구와 현장실무자들과의 면접과정을 통하여 조사하여 본 결과 다음같은 23개의 기능으로 나눌 수 있었다. 설문지의 내용을 분석함에 있어서 설문지에 있는 23개 기능을 편의상 <표 1>과 같이 약어로 표시하였다.

<표 1> 설문분석시 CASE 기능별 표기

약어	용어 설명
DFD	Data Flow Diagram
ERD	Entity Relation Diagram
SC	Structure Chart-Yourdon, Constantine
SD	Structure Diagram-Jackson
LDMD	Logical Data Model Diagram
ADD	Activity Decomposition Diagram
SP	Screen Painter
RP	Report Painter
ECW	Error Checking within Each Diagram
ECA	Error Checking across Diagrams
DMA	Data Model Analysis
DM	Data Modeling
DMLR	Data Model List Report
ESC	Evolving System by Changing
PMS	Project Management Support
DG	Document Generation
CG	Code Generation
RE	Reverse Engineering
I/O	Import/Export
LAN	LAN 등 사용자 접속기능
O/A	Open Architecture
BR	Bridge 기능
P/M	PC, Mainframe 등 운영환경

CASE의 생산성을 측정하기 위하여 표본대상자에게 위의 각 기능이 수작업에 비하여 가지는 비교우위의 정도를 독립적으로 판단하여 7점 척도로 표시하도록 하였다. 표본대상 기업은 현재 국내에서 CASE를 도입하여 사용하고 있는 30여개 업체중 최소한 1개 이상의 프로젝트 개발 경험이 있는 기업에서 CASE tool의 모든 기능을 사용해 본 경험이 있는 시스템전문가를 대상으로 하였다. 그 결과 통합 CASE는 6개 기업에 응답자는 59명이었으며, 상위 CASE는 1개 기업에 응답자는 10명, 그리고 하위 CASE는 1개 기업에 14명 이었다. 그러나 하위 CASE 대상 업체와 통합 CASE의 경우 2개 기업은 응답자들이 충분한 지식이 없이 응답한 것으로 판단되어 분석대상에서 제외하였다. 최종 표본조사 대상은 <표 2>와 같다.

<표 2> 설문분석 대상 개요

제품분류	제품명	업체명	응답수
통합 CASE	'가'	A사	16
		B사	11
	'나'	C사	15
		D사	9
상위 CASE	'다'	E사	10
합 계		5 개사	61

설문을 분석하기에 모집단의 성격에 대한 적합성 여부를 판정하기 위하여 각 기능항목에 대한 응답분포가 정규분포인지와 각 집단간에 등분산 여부를 확인한 결과 대체적으로 분석을 하기에는 무리가 없을 정도의 정규성과 등분산을 보였다. 또한 본 연구의 주요분석대상인 생산성 향상에 관한 CASE 기능의 항목과 기타 보조기능에 대한 응답에 대해 신뢰성 여부를 판정하기 위하여 크론박-알파(Cronbach-Alpha) 계수를 산정한 결과 모든 값이 0.86 이상으로 신뢰성에 문제가 없었다.

2. 표본의 특성

표본조사대상이 되는 시스템전문가 61명중 전산경력 항목에 응답한 60명의 전산경력 분포를 보면 <표 3>과 같다. 3년 내지 6년의 전산경력을 가진 사람이 전체의 31.6%로 가장 많았고 나머지 계급에도 고른 분포를 나타내고 있다.

<표 3> 전산경력별 응답자 분포

전산경력	응답자수	비율 %
3년 이하	16명	26.7
3-6년 이하	19명	31.6
6-9년 이하	12명	20.0
9년 초과	13명	21.7
총 계	60명	100.0

CASE의 사용경력에 대하여는 57명이 응답을 하였는데 이들의 분포는 <표 4>와 같다. 그 내용을 보면 1년 이상 사용한 사람은 19명으로 전체의 33.3%에 불과하고 나머지는 사용경력이 1년 미만으로 대부분 아직 사용경력이 짧음을 알 수 있다.

<표 4> CASE 사용경력별 응답자 분포

사용경력	응답자수	비율 %
6개월 이하	20명	35.1
6-1년 이하	18명	31.6
1년 초과	19명	33.3
총 계	57명	100.0

IV. 실증조사 결과

1. CASE 도입목적에 대한 분석

먼저 CASE를 도입하게 된 목적이 무엇인가에 대해 응답한 4개 기업의 응답내용은 <표 5>에

나타나 있다. 이들 4개의 기업은 모두 통합 CASE를 사용하고 있었으며 기업간에 도입목적 순위가 일치하는가를 검정하기 위하여 Kendall의 순위일치계수를 산정한 결과 0.3542로 유의수준 0.05에서 일치한다는 가설은 기각되었다. 따라서 기업별로 도입목적의 순위는 통계적으로 일치한다고 할 수 없겠으나 평균의 순위를 살펴보면 품질향상 목적이 가장 높은 순위를 나타냈고 그 다음이 개발시간 단축, 표준화 지원, 데이터 최적관리, 방법론 지원, 비용감소 및 프로젝트 관리, 유지보수 개선, 프로토타이핑의 순으로 나타났다.

〈표 5〉 CASE 도입목적에 대한 기관별 순위

도입목적\기관	A	B	C	D	평균순위
품질향상	1	2	2	4	2.25
개발시간단축	2	3	1	6	3.00
표준화 지원	6	5	4	2	4.25
데이터 최적관리	3	9	5	1	4.50
방법론 지원	4	6	3	9	5.50
비용감소	9	1	7	7	6.00
프로젝트관리 증진	5	7	9	3	6.00
유지보수의 개선	7	4	6	8	6.25
프로토타이핑	8	8	8	5	7.25

각 기관들이 CASE를 실제 도입하여 사용하여 본 결과 본래의 도입목적에 얼마나 만족하고 있는가를 알아보기 위하여 조사한 자료를 나타낸 것이 〈표 6〉이다. 상위 도입목적에 해당하는 품질향상의 목적과 개발시간의 단축 목적에 대해서는 전체적으로 만족을 하고 있는 것으로 나타났으며 표준화와 방법론에 있어서는 3개의 기업이 상당히 만족하고 있는 것으로 나타났으나 한개의 기업은 불만을 표시하였다. 이유를 분석한 결과 그 기관의 경우 표준화와 방법론에 대한 자체의 방안이 이미 정립되어 있었는데 그 내용이 CASE에서 지원하는 방법론과 달라 낮게 평가한

것으로 나타났다. 그밖에 전기관이 비용감소의 면에 있어서는 좋은 반응은 보이지 않고 있는데, 주된 이유는 대부분의 사용자들이 CASE 툴을 학습하는 초기단계에 있기 때문에 오히려 비용이 증가하는 경향이 있기 때문으로 해석된다. 이 결과로 부터 CASE는 S/W 비용의 절감보다는 품질향상이나 개발기간의 단축면에서 더 의의가 있다고 해석할 수 있을 것이다.

〈표 6〉 CASE 도입목적에 대한 기관별 만족도

도입목적\기관	A	B	C	D	평균만족도
방법론 지원	6	6	6	2	5.00
표준화 지원	6	6	6	2	5.00
품질향상	4	6	5	5	5.00
개발시간단축	5	4	4	5	4.50
데이터 최적관리	3	4	5	5	4.25
유지보수의 개선	3	6	3	4	4.00
프로젝트관리 증진	4	5	4	3	4.00
프로토타이핑	5	4	4	2	3.75
비용감소	3	4	4	4	3.75

2. CASE의 생산성 기여도 분석

CASE가 과연 S/W의 생산성 향상에 기여할 수 있는가, 또 있다면 어떤 기능이 기여하고 있는가를 분석하기 위하여 표본대상자에게 수작업에 비하여 CASE가 가지는 비교우위정도를 7점 척도로 나타내도록 한 결과 〈표 7〉과 같은 결과를 얻었다.

정보시스템 전문가들이 인식하는 CASE의 전반적인 생산성은 4.53점이며 2개의 기능을 제외한 모든 기능이 4점 이상으로 비교적 긍정적으로 나타났다. 특히 CASE의 기능중 실체관계도(ERD), 다이어그램내 예러체크(ECW), 자료구조정의(DM), 자료구조보고(DMLR), 자료흐름

〈표 7〉 CASE의 기능별 생산성 기여도

기능	평균	표준편차	기능	평균	표준편차
DFD	4.79	1.57	ECA	4.59	1.30
ERD	5.22	1.34	DMA	4.47	1.42
SC	4.84	1.19	DM	4.88	1.40
SD	4.29	1.21	DMLR	4.72	1.33
LDMD	4.63	1.44	ESC	4.30	1.61
ADD	4.64	1.36	PMS	4.13	1.49
SP	4.57	1.57	DG	4.64	1.64
RP	3.90	1.55	CG	4.16	1.96
ECW	4.92	1.19	RE	3.81	1.70
전체 평균			:	4.53	

도(DFD)는 비교적 높게 나타난 반면 재공학(RE), 출력화면설계(RP), 프로젝트 관리지원(PMS), 코드생성(CG)은 상대적으로 낮게 나타났다.

생산성 기여도에 대한 인식은 전산경력에 따라 달라질 수 있으며 또한 CASE 제품에 따라 차이가 있을 수 있기 때문에 다음에는 이에 대한 분석을 하기로 한다.

3. 전산경력과 CASE 기능간의 분석

전산경력과 CASE의 생산성 기여도에 대한 인식의 차이를 알아보기 위하여 전산경력과 각 기능간의 상관관계를 분석한 결과 〈표 8〉과 같았다. 자료흐름도(DFD)와 Yourdon과 Constantine의 구조도(SC)에 있어서 유의적으로 상관관계를 나타내고 있는데 전산경력이 많을수록 DFD와 SC의 S/W 생산성에 대한 기여도가 큰 것으로 응답하였다. 나머지 기능에 대하여는 전산경력과 생산성 기여도에 대한 인식에 있어 유의적인 차이를 발견할 수 없었다. 그러나 〈표 8〉을 살펴보면 비록 통계적으로 유의성을 발견할 수는 없었으나 프로젝트 관리기능에 있어서는

전산경력이 높은 사람일수록 생산성 기여도가 큰 것으로 응답한 반면 코드생성(CG) 기능과 입력화면설계(SP)에 있어서는 전산경력이 적은 사람들이 생산성 기여도가 큰 것으로 응답하였다. 이들의 결과를 종합하여 볼 때에 전산경력이 많은 사람일수록 CASE의 그래픽 기능이 과거 수작업에 비해 생산성을 높일 수 있다고 인식하고 있는 반면 코드생성과 입력화면설계에 있어서는 이미 숙달되어 있기 때문에 CASE의 기능이 크게 도움이 되지 않는 것으로 응답한 것으로 해석할 수 있을 것이다.

4. 지원단계별 CASE와 제품별 CASE의 생산성 기여도 분석

최근의 CASE 동향은 상위 또는 하위 CASE에서 점차 통합 CASE로 이행되고 있는 현상을 볼 수 있는데 여기에서는 상위 또는 하위 CASE와 같은 단일 구성 CASE(Single Component CASE)와 통합 CASE간에 생산성 기여도에 있어 차이가 있는지를 분석하려고 한다. 또한 통합 CASE의 경우 표본에 선정된 2개의 통합 CASE는 세계적으로 가장 많이 이용되는 대표적

〈표 8〉 전산경력과 각 기능간의 상관관계표

기 능	상관계수	P 값	기 능	상관계수	P 값
DFD	0.3183	0.013	DMLR	-0.0064	0.962
ERD	0.2011	0.130	ESC	0.0353	0.797
SC	0.2781	0.037	PMS	0.2172	0.118
SD	0.0202	0.891	DG	-0.0189	0.888
LDMD	0.2447	0.086	CG	-0.1795	0.181
ADD	0.1699	0.214	RE	-0.0195	0.893
SP	-0.2008	0.127	I/O	0.1572	0.251
RP	-0.0467	0.729	LAN	-0.0514	0.709
ECW	-0.0492	0.713	O/A	-0.0380	0.778
ECA	-0.0121	0.931	BR	-0.0399	0.790
DMA	0.0194	0.886	P/M	-0.0070	0.959
DM	0.1080	0.432			

〈표 9〉 CASE 분류별 생산성 기여도

기 능	통합 CASE		상위 CASE		t 값
	통합	표준편차	통합	표준편차	
DFD	4.92	1.85	4.10	1.49	-1.5281
ERD	5.29	1.28	4.75	1.34	-1.0692
SC	4.96	1.15	4.00	1.15	-2.0520*
SD	4.27	0.81	4.33	1.25	0.1020
LDMD	4.68	0.95	4.28	1.50	-0.6715
SP	4.70	1.72	3.90	1.51	-1.4896
RP	3.85	1.10	4.10	1.63	0.4523
ECW	4.80	1.01	5.55	1.19	1.7812
ECA	4.53	1.36	4.88	1.28	0.7481
DMA	4.40	1.64	4.87	1.38	0.8629
DM	5.00	1.64	4.12	1.33	-1.6601
DMLR	4.79	1.41	4.33	1.32	-0.9457
ESC	4.43	1.85	3.50	1.54	-1.5480
PMS	4.19	1.79	3.71	1.45	-0.7863
DG	4.82	1.92	3.50	1.53	-2.1817*

인 CASE 제품인바 이들간에 생산성 기여도에 있어 차이가 있는가를 분석하려고 한다.

4.1 지원단계별 생산성 차이분석

첫번째의 가설은 다음과 같이 설정된다.

H₁: 통합 CASE와 상위 CASE간에 생산성 기

여도에 있어 차이가 있을 것이다.

<표 9>는 통합 CASE와 상위 CASE가 공통적으로 가지고 있는 기능에 대하여 생산성 기여도를 나타내고 있다.

두 CASE의 전체적인 기능에 차이가 있는가를 분석하기 위해 Hotelling의 T² 검정을 한 결과

0.7133(p값은 0.7477)으로 연구가설은 기각되

<표 10> '가' CASE 와 '나' CASE 의 각 기능에 있어 생산성 기여도

기 능	'가' 통합CASE		'나' 통합CASE		전체 평균	t 값
	평균	표준편차	평균	표준편차		
DFD	4.29	1.46	5.62	1.20	4.92	3.5092*
ERD	4.77	1.36	5.87	1.07	5.29	3.1552*
SC	4.85	1.26	5.08	1.04	4.96	0.7107
SD	3.92	1.22	4.77	1.16	4.27	2.3142*
LDMD	4.15	1.40	5.44	1.33	4.68	3.0530*
ADD	3.96	1.58	5.37	0.92	4.64	3.8843*
SP	5.00	1.68	4.34	1.22	4.70	-1.5378
RP	3.81	1.98	3.90	1.09	3.85	0.2001
ECW	4.51	1.34	5.13	0.91	4.80	1.8482
ECA	4.19	1.23	5.00	1.24	4.53	2.1594*
DMA	4.00	1.46	4.90	1.10	4.40	2.3990*
DM	4.66	1.41	5.42	1.12	5.00	2.0221*
DMLR	4.30	1.51	5.36	0.72	4.79	3.1478*
ESC	4.16	1.83	4.70	1.16	4.43	1.2229
PMS	4.56	1.66	3.77	1.06	4.19	-1.9565
DG	5.00	1.64	4.60	1.40	4.82	-0.8969
CG	3.92	1.71	4.43	1.34	4.16	1.1503
RE	3.50	1.81	4.14	1.49	3.81	1.2633

었다.

그러나 각각의 기능에 대한 t 검정을 한 결과 0.05의 유의수준에서 Yourdon과 Constantine의 구조도(SC)와 문서화(DG)에 있어 유의적인 차이가 있는 것으로 나타났는데 평균값을 살펴본 결과 이 두기능에 있어서는 통합 CASE가 상위

CASE보다 생산성이 높은 것으로 인식되었다.

4.2 제품별 CASE의 차이분석

같은 종류의 CASE라 할지라도 제품에 따라 생산성 기여도에 있어 차이가 있나 알아보기 위

하여 두개의 통합 CASE 제품에 대한 자료를 비교하였다. 따라서 두번째의 가설은 다음과 같이 설정된다.

H₁: '가' 통합 CASE와 '나' 통합 CASE간에 생산성 기여도에 있어 차이가 있을 것이다.

두 제품의 생산성 기여도에 관한 자료를 나타낸 것이 <표 10>인데 전체 평균값을 보면 전체적으로 수작업에 비해 비교우위가 있는 것으로 응답하였으며 특히 자료흐름도(DFD), 실체관계도(ERD), Yourdon과 Constantine의 구조도(SC), 자료구조정의(DM), 다이어그램내 에러체크(ECW)에 있어 생산성 기여도가 큰 것으로 나타나고 있다.

전체 기능간에 차이가 있나를 알아보기 위해 Hotelling 검정 결과 T² 값이 1.844로 연구가설은 기각되었다(p 값=0.1324). 그러나 각각의 기능에 대한 추가적인 t 검정을 한 결과 일부 기능에 있어서는 차이가 있는 것으로 나타났다. 즉 자료흐름도(DFD), 실체관계도(ERD), Jackson의 구조도(SD), 논리적 자료모형도(LDMD), 비지니스 기능의 계층도(ADD), 방법론들간에 일관성 및 완전성 검사(ECA), 자료모형분석(DMA), 자료구조정의(DM), 자료모형보고(DMLR) 등의 기능에서는 두 종류의 CASE간에 유의성이 있는 것으로 나타났다. 특히 '나' 통합 CASE가 상대적으로 비교우위가 높게 나타나고 있다.

5. CASE 기능에 대한 인자분석

CASE가 제공하고 있는 23개의 기능중 생산성에 관련된 18개의 기능을 보다 큰 몇가지의 기능으로 대별할 수 있는지를 알아보기 위하여 인자분석을 하고자 한다. 설문을 통해 얻은 CASE의 각 기능항목의 생산성 기여도들을 상관관계를 통하여 각각 유사성이 있는 기능으로 묶

음으로써 이들을 대표할 수 있는 인자들을 찾아낸다. 만약 설명할 수 있는 인자들이 발견된다면 이들 인자들이 지원단계별로 차이가 있는지 또는 같은 지원단계를 지원하더라도 제품간에 차이가 있는지를 검증한다. 인자분석에 사용된 설문은 통합 CASE 중에서 미응답 부분이 없는 부분으로 하였는데 32개의 설문이 분석되었다.

먼저 주성분분석(Principal Component Analysis)을 통하여 각각 고유값(Eigen Values)이 1 이상으로 산출된 성분을 선택하였는데 5개로 결정되었다. <표 11>은 5개의 성분을 대상으로 적재행렬표를 만든후 VARIMAX 방식으로 회전하여 얻은 인자행렬표이다. 이들 인자들의 고유값의 합은 14.399로 계산되며 5개의 인자가 전체분산 중 약 80%를 설명하는 것으로 나타났다. 공통성(communality)을 살펴보면 출력화면 설계(RP)와 재공학(RE)을 제외하고는 다섯개의 인자가 각 기능의 분산에 대하여 70-90% 정도를 설명하고 있다. 각 인자들의 내용을 살펴보면 다음과 같다.

5.1 인자 1 에 대한 분석

인자 1의 고유값은 4.365이며 전체분산 중약 24% 정도를 설명하고 있다. 적재값이 높게 나타난 변수는 5개로 나타났는데 실체관계도(ERD), Yourdon과 Constantine의 구조도(SC), Jackson의 구조도(SD), 논리적 자료모형도(LDMD), 자료구조정의(DM)이다. 예상했던 대로 이들의 부호는 모두 양으로 나타났다. 이들의 성격은 자료구조 지향(Data Structure Modeling Oriented)의 변수로 표현할 수 있을 것이다. 결국 자료구조추출의 세부기능은 상당히 높은 양의 상관관계로 연관되어 있고 생산성의 비교우위도 유사하게 나타난다는 것이다.

〈표 11〉 VARIMAX 방식으로 회전한 인자행렬표

	인자 1	인자 2	인자 3	인자 4	인자 5	공통성
DFD	0.453	0.762	0.093	0.186	-0.238	0.888
ERD	0.880	0.250	-0.055	0.174	-0.015	0.872
SC	0.825	-0.252	0.189	0.218	-0.192	0.865
SD	0.897	0.152	-0.063	0.062	0.074	0.843
LDMD	0.864	0.349	-0.177	-0.061	0.026	0.904
ADD	0.449	0.797	0.050	-0.069	0.104	0.856
SP	0.088	-0.043	0.000	0.102	0.942	0.909
RP	-0.071	0.387	0.361	0.069	0.516	0.557
ECW	0.288	0.131	0.104	0.868	0.090	0.874
ECA	0.105	0.555	0.029	0.715	0.243	0.891
DMA	0.471	0.632	0.043	0.116	0.313	0.736
DM	0.695	0.275	-0.001	0.115	0.370	0.710
DMLR	0.018	0.830	0.195	0.204	0.011	0.769
ESC	-0.068	0.545	0.548	0.246	0.203	0.705
PMS	-0.042	-0.006	0.724	0.494	-0.266	0.841
DG	-0.319	0.058	0.833	0.053	-0.082	0.809
CG	0.072	0.027	0.861	0.032	0.103	0.759
RE	0.141	0.214	0.705	-0.104	0.171	0.603
고유값	4.365	3.480	3.026	1.795	1.729	14.399
VAF	24.25%	19.33%	16.81%	9.97%	9.60%	79.99%

인자분석의 결과 나타난 자료구조 지향에 관한 변수에 대하여 '가'와 '나'의 통합 CASE간에 생산성 기여도의 차이를 분석하기 위하여 가설을 다음과 같이 설정하였다.

H_1 : 자료구조 지향 변수의 생산성 기여도에 있어 '가' 통합 CASE와 '나' 통합 CASE간에 차이가 있을 것이다.

Hotelling T^2 를 통하여 검증한 결과 1.8269로 유의수준 0.05에서 가설이 기각되었다 (p 값=0.1330). 따라서 자료구조 지향의 생산성 기여도는 '가' 통합 CASE와 '나' 통합 CASE간에 차이가 있다고 할 수 없다.

5.2 인자 2에 대한 분석

인자 2의 고유값은 3.48이며 전체분산중 약 19% 정도를 설명하고 있다. 적재값이 높게 나타난 변수는 4개로 나타났는데 자료흐름도(DFD), 비즈니스 기능의 계층도(ADD), 자료모형분석(DMA), 자료모형보고(DMLR)의 변수이다. 이들의 성격은 자료흐름 지향(Data Flow Modeling Oriented)의 변수로 표현할 수 있을 것이다. 또한 이들 변수들은 양의 상관관계로 연관되어 있다.

자료흐름 지향 변수에 관하여 '가'와 '나'의 통합 CASE간에 생산성 기여도의 차이를 분석하

기 위하여 두 CASE간에 차이가 있다는 가설을 세워 검증한 결과 유의수준을 0.05로 하였을 때 유의성이 있음이 검증되었다($p값=0.012$). CASE 전반적인 면에서의 차이는 보이지 않았으나 자료흐름 지향 변수의 경우만 비교하였을 때에는 상당한 차이가 있는 것으로 나타났는데 결국 CASE 제품에 따라서 자료흐름을 지향하는 기능의 생산성 기여도는 차이가 있으므로 선택에 있어서 신중한 고려가 있어야 할 것으로 생각된다.

5.3 인자 3 에 대한 분석

인자 3의 고유값은 3.026이며 전체분산중 약 17% 정도를 설명하고 있다. 적재값이 높게 나타난 변수는 5개로 나타났는데 요구사항 변화시 자동교정(ESC), 프로젝트 관리지원(PMS), 문서화(DG), 코드생성(CG), 재공학(RE)의 변수이다. 이들의 성격은 유지보수 지향(Maintenance Oriented)의 변수로 표현할 수 있을 것이다. 여기서 요구사항 변화시 자동교정기능은 정보저장소(Repository)의 주요기능이면서 유지보수와 밀접한 기능을 갖는 기능이다.

유지보수 지향 변수에 관하여 '가'와 '나'의 통합 CASE간에 생산성 기여도의 차이를 분석하기 위하여 종속변수를 인자 3에 높게 적재된 변수들로 정한 다음 두 CASE간에 차이가 있다는 가설을 세워 검증한 결과 유의수준 0.05에서 가설이 기각되었다($p값=0.1335$). 따라서 유지보수 지향 변수의 생산성 기여도는 '가' 통합 CASE와 '나' 통합 CASE간에 차이가 있다고 할 수 없다.

5.4 인자 4에 대한 분석

인자 4의 고유값은 1.795로 전체분산중 약 10% 정도를 설명하고 있다. 적재값이 높게 나타난

변수는 2개로 나타났는데 다이어그램내 에러체크(ECW), 방법론들 간에 에러체크(ECA) 등의 변수가 높게 적재되었다. 여기에서도 예상했던 대로 이들의 부호는 양으로 나타났으며 이들의 성격은 품질관리 지향(Quality Oriented)의 변수로 표현할 수 있을 것이다.

품질관리 지향 변수에 있어 '가' CASE와 '나' CASE간에 생산성 기여도에 있어 차이가 있다는 가설은 유의수준 0.05에서 가설이 기각되었다($p값=0.0567$). 따라서 품질관리 지향 변수의 생산성 기여도는 '가' 통합 CASE와 '나' 통합 CASE간에 차이가 있다고 할 수 없다.

5.5 인자 5에 대한 분석

인자 5의 고유값은 1.729이며 전체분산중 약 10% 정도를 설명하고 있다. 적재값이 높게 나타난 것은 2개의 변수로서 입력화면설계(SP), 출력화면설계(RP)의 변수이다. 여기에서도 예상했던 대로 이들의 부호는 양으로 나타났으며 이들의 성격은 프로토타이핑 지향(Prototyping Oriented)의 변수로 표현할 수 있을 것이다.

프로토타이핑 지향 변수에 있어 '가' CASE와 '나' CASE간에 차이가 있다는 가설은 유의수준 0.05에서 가설이 기각되었다($p값=0.1210$). 따라서 프로토타이핑을 지향하는 기능에 대한 생산성 기여도는 '가' 통합 CASE와 '나' 통합 CASE간에 차이가 있다고 할 수 없다.

이상 인자분석을 통하여 각 기능을 분석하여 본 결과 현재의 통합 CASE는 자료흐름지향, 자료구조지향, 유지보수지향, 품질관리지향, 프로토타이핑지향의 다섯가지의 큰 기능으로 묶어짐을 알 수 있었다. 또한 각 인자들 내의 변수들이 모두 양의 상관관계를 나타냄을 보았다.

6. CASE의 진화 모형

여기에서는 조사대상 시스템 전문가들에 대한 표본조사와 면접과정을 통하여 CASE가 소프트웨어 품질과 생산성을 높이기 위하여 해결되어야 할 과제를 정리함으로써 미래의 CASE 환경을 예측하고 향후 CASE Tool의 연구방향을 제시하고자 한다.

먼저 초기 단계에서는 상위 또는 하위 CASE와 같은 단일구성 CASE로 출발하여 점차 통합 CASE로 이행되고 있으나 기존시장에 나와있는 통합 CASE들도 “통합”(integrated)이라는 용어를 사용하고는 있지만 통합의 정도에 있어서는 많은 차이가 있을 뿐만 아니라 대부분이 상위와 하위 CASE 도구들을 묶어 놓은 것에 불과하며 시스템 개발수명주기의 모든 단계(Full Life Cycle)를 지원하는 통합된 CASE로 발전되어야 할 것이다. IBM의 AD/Cycle이나 DEC의 Cohesion은 아직 구체적으로 구현되지는 않았으나 공통적인 플랫폼(platform)을 통하여 통합 정도를 높이려고 하고 있으나 개발수명주기 단계간의 변환에 있어(Cross-Phase Transformations) 문제가 있을 뿐만 아니라 정보저장소(Repository)가 아직 정립되어 있지 않고 방법론에 있어서도 통일이 되어있지 않아 이러한 문제를 어떻게 해결해 나갈지 귀추가 주목된다.

두번째로는 개방형 CASE(Open CASE)로의 이행이 이루어져야 할 것이다. “개방”(open)이라고 함은 공급처(vendor)와 독립적이면서 다수의 공급업자의 표준(standard)에 따라 설계되고 구현된 테크놀러지와 제품으로 구성된 소프트웨어 환경을 말하는데 현재의 CASE와 도구들은 개방형이라는 이름을 사용하는 것들도 특정한 공급처의 운영체제나 DBMS 하에서만 운영이 가능할 뿐만 아니라 전유코드(Proprietary Code)

만 생성해 냄으로써 사용과 선택에 신중을 요한다.

세번째로는 객체지향형 CASE(Object-Oriented CASE) 기술의 발전이 이루어져야 할 것이다. 현재의 CASE 도구들이 기초를 두고 있는 구조화 기법은 현재 로직의 큰 비중을 차지하고 있는 휴먼 인터페이스와 관계된 GUI나 펜 기반 컴퓨팅(Pen-Based Computing)에 대한 아무런 지침이 없을 뿐만 아니라 재이용 가능성에 있어서도 객체지향방법론에 비해 뒤떨어진다고 할 수 있다. 최근에는 시스템의 규모가 방대해져 재이용에 대한 관심이 점차 증대되고 휴먼 인터페이스의 비중이 증가됨에 따라 객체지향형 CASE를 지원할 수 있는 기반시설이 강화되어야 할 것이다.

네번째로는 기존 시스템을 다시 구축하기 위한 재공학(Reverse Engineering) 기능이 보완되어야 한다. 초기단계의 최대 관심사는 새로운 시스템을 개발할 때의 CASE 사용이었으나 점차 사용이 확산되어감에 따라 종래의 시스템을 계속 살려서 이용하는 문제가 대두됨에 따라 CASE가 그런 작업을 도울 수 있어야 할 것이다.

그 외에도 클라이언트-서버 방식의 구축환경을 지원한다든지 그룹웨어 CASE 툴 등이 제기되었으며 외국제품으로 인한 문화적인 정착을 위한 연구들이 진행되어야 할 것으로 지적되었다.

V. 맺 음 말

소프트웨어 위기를 극복하기 위해 등장한 CASE에 대한 관심이 고조되기 시작하면서 과연 CASE가 소프트웨어의 품질과 생산성을 향상시킬 수 있으며 또한 어떤 CASE 툴을 도입할 것인가가 정보관리자의 중요한 의사결정의 하나가 되었다. 정보시스템 전문가들에 대한 설문조사결

과 CASE의 도입목적은 품질향상, 개발시간단축, 표준화 지원, 데이터 최적관리, 방법론 지원, 프로젝트 관리증진의 순으로 나타났고 비용감소의 목적은 도입순위가 크게 떨어져 소프트웨어 개발 비용을 줄이기 위해 CASE를 도입하려고 하는 기업은 적었다.

정보시스템 전문가들이 인식하는 CASE의 전반적인 생산성은 4.53으로 비교적 긍정적으로 나타났으며 특히 2개의 기능을 제외한 모든 기능에 대하여 4점 이상으로 응답하였다. 또한 CASE 툴의 생산성 효과는 기능에 따라서는 전산경력에 영향을 받고 있음을 발견하였다. 지원단계별 CASE와 제품별 CASE간의 생산성 차이는 전체적으로 볼때는 차이가 있다고 할 수 없었으나 기능에 따라서는 차이가 나타나 이러한 차이가 제품자체에서 오는 것인지 기업의 개발환경에서 오는 것인지가 분명치는 않으나 면접과정

에서 기업간의 개발환경의 차이를 발견할 수 없어 제품자체에서 오는 차이로 인식할 수 밖에 없었다. 따라서 CASE를 도입할 때에는 도입목적과 일치하는 툴을 선정하는데에 유의하여야 할 것이다. CASE가 제공하고 있는 기능들은 크게 자료흐름 관련기능, 자료구조 관련기능, 유지보수 관련기능, 품질관리 관련기능, 프로토타입 관련기능으로 대별할 수 있으며 이러한 인자내의 소기능들은 양의 상관관계로 결합되어 있었다.

그러나 본 연구는 아직 국내에서 CASE의 도입기관이 충분치 않은 상황에서 진행된 결과 표본대상업체와 응답자수에 있어서 제한적일 수 밖에 없었다. 따라서 CASE의 사용이 좀 더 정착된 후에 이와같은 연구가 다시 이루어져야 할 것이다. 또한 소프트웨어의 생산성을 직접 측정할 수 있는 모형의 개발이 동시에 진행되어야 할 것이다.

참 고 문 헌

류성렬, 여호영, "CASE의 선정 및 평가방법," 정보과학회지, 1989.8, pp.30-40.

송성면, "소프트웨어 유지보수관리의 효율화를 위한 지원도구의 설계 및 구현," 성균관대 석사학위논문, 1991.5.

이상연, "국내소프트웨어 개발도구 이용현황에 대한 조사연구," 한국외대 경영정보대학원 석사학위논문, 1990.4.

이주현, 프로젝트 관리론, 법영사, 1991.

조선희, "소프트웨어 생산과 엔지니어링 공학 : 엔지니어의 질적향상과 개발의 자동화," 컴퓨터월드, 1988.4, pp.150-160.

Banker, R.D. and Kauffman, R.J., "Reuse and Productivity in Integrated Computer-Aided Software Engineering : An Empirical Study," *MIS Quarterly*, September 1991, pp.374-401.

Baram G. and Steinberg G., "Selection Criteria for Analysis and Design CASE Tools," *ACM SIGSOFT Software Engineering Notes*, Vol.14, No.6, Oct. 1989, pp.73-80.

Boehm, B. W., "Improving Software Productivity," *IEEE Reprinted with permission from Computer*, 1987, pp.43-57.

Carlyle, R., "Is Your Data Ready For the

- Repository?," *Datamation*, January 1, 1990, pp.43-47.
- Falk, H., "Software Vendors serve up varied palette for CASE Users," *Computer Design*, Jan. 1989, pp.70-80.
- Gabel, "A Yen for Just-in time Decisions Aids Sony's Drive for Coprocessing," *Computerworld*, April 10, 1989.
- Gane, C., *COMPUTER-AIDED SOFTWARE ENGINEERING*, Prentice-Hall, 1990.
- Gibson, M.L., "A Guide To Selection CASE Tools," *Datamation*, Vol.34, No.13, July 1, 1988, pp.124-130.
- Gibson, M.L., "The CASE Philosophy," *BYTE*, APRIL 1989, pp.209-213.
- Hanson, S.J. and Rosinski, R.R., "Programmer Perceptions of Productivity and Programming Tools," *Communication of the ACM*(28:2), February 1985, pp.180-189.
- Hazzah, A., "Making Ends Meet : Repository Manager," *Software Magazine*, December 1989, pp.59-71.
- Kalakostas, V., "Requirement for CASE tools in Early Software Reuse," *ACM SIGSOFT Software Engineering Notes*, Vol.14, No.2., Apr. 1989, pp.39-54.
- Kerr, J.D., "Systems design : Users in the hot seat," *COMPUTERWORLD*, February 27, 1989, pp.86-91.
- Knight, J., "CASE Up On MIS Agenda," *Software Magazine*, August 1989, pp.32-36.
- Loh, M. and Nelson, R. R., "Reaping CASE Harvests," *Datamation*, July 1, 1989, pp.31-34.
- Martin, J. and McClure, C., *Structured Techniques for computing*, Prentice-Hall, 1985.
- McClure, C., *CASE IS SOFTWARE AUTOMATION*, Prentice-Hall, 1989.
- McGuff, F.P., "Cost Cutting Revisited," *Computerworld*, July 17, 1989.
- McNurlin, B.C. and Sprague, R.H., Jr., *Information Systems Management in Practice*, 2nd ed., Englewood Cliffs, N.J.: Prentice-Hall, 1989.
- Naur, P., and B. Randell(eds.), *Software Engineering: A Report on a Conference Sponsored by NATO Science Committee*, NATO, 1969.
- Norman, R.J. and Nunamaker, J.F., Jr., "CASE Productivity Perceptions of Software Engineering Professionals," *Communications of the ACM*(32:9), September 1989, pp.1102-1108.

Pressman, R.S., *Software Engineering*, McGraw-Hill, 1987

Sommerville, I., *Software Engineering*, Addison Wesley, 1989.

Sperling, E., Schwartz, E., and Gerber, C., "IBM Makes CASE for Coding," *Computer Systems News*, September 25, 1989.

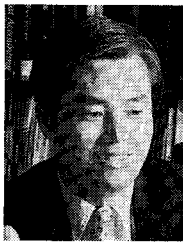
Thadani, A.J., "Factors Affecting Programmer Productivity During Application Development," *IBM Systems Journal*, Vol.23, No.1, 1984.

Troy, D.A., "An Evaluation of CASE Tools," *CONAC/IEEE COM. SOC., COMP. Software & Applications Conf. 10th-11th*, 1986-87, pp.124-130.

Zelkowitz, M., Shaw, A., and Gannon, J., *Principles of Software Engineering and Design*, Englewood cliffs, N.J.: Prentice-Hall, 1979.

Zucconi, L., "Selecting a CASE Tool," *ACM Software Engineering Notes*, Vol.14, No.2, Apr. 1989, pp.43-44.

◇ 저자소개 ◇



공동저자 김효석은 현재 중앙대학교 경영대학 교수로 재직중이다. 그는 서울대학교 상과대학 경영학과를 거쳐 미국 조지아 대학에서 경영학 석사학위 및 경영학 박사학위를 취득하고 미국 UCLA에서 Visiting Scholar로 활동하였다. 저서로는 '경영마인드와 PC와의 만남', '경상계열을 위한 통계학' 등이 있으며 주요 관심분야는 MIS 전략과 정보기술, 시스템 분석 및 설계, 소프트웨어 공학 등이다.



공동저자 김병철은 중앙대학교 경영대학 경영학과를 졸업하고(경영학사) 동 대학원 경영학과에서 경영학 석사학위를 취득하였고 현재 중앙대학교 경영대학 전산실에서 근무하고 있다. 그의 주요 관심분야는 시스템 분석 및 설계, 전문가시스템, 전략정보시스템 등이다.