

SW공학 수준과 SW개발 프로젝트의 생산성

김 승 권* , 윤 종 수**

Exploring the Relationship between Software Capability Level and Software Project Productivity

Seung-gweon Kim* , Jong-soo Yoon **

요 약

최근들어 IT 융복합화가 빠르게 진행됨에 따라, SW의 역할과 활용범위가 확대되고 있다. 증가하는 SW 수요와 품질에 대한 요구사항을 만족시키기 위해, 많은 SW 기업들이 SW 제품에 대한 품질과 프로젝트의 성과를 높이기 위해 SW 프로세스 개선을 추진하고 있다. SW 프로세스 개선은 많은 비용과 시간이 요구되는 어려운 작업임에도 불구하고, SW 프로세스 개선의 성과에 대한 객관적인 증거는 여전히 부족한 실정이다. 따라서, 본 연구는 SW 프로세스 개선과 SW 개발 프로젝트 생산성간의 관계를 분석하는데 초점을 두고 있다. 즉, SW 공학수준에 따라 SW 개발 생산성이 차이가 나는지를 통계적 분석방법을 활용하여 검증하고 있다. 분석결과에 따르면, SW 공학수준에 따라 SW 개발 생산성이 다르게 나타나며, SW 공학수준이 높을수록 SW 개발 생산성이 높은 것으로 나타나고 있다.

▶ Keywords : SW 프로세스 개선, SW 개발 생산성, SW 공학수준

Abstract

Recently, as IT convergence has seen rapid growth the role and scope of software has been extended. many software companies are trying to improve the software quality and project outcome using software process improvement approach to cope with increasing software demands and software quality. Although software process improvement is difficult task which requires a lot of cost and time, there is still insufficient objective evidence on business benefits by its deployment. The purpose of this study is to analyze the relationship between software process improvement and software project productivity. That is, we investigated whether there is difference in software development productivity depends on software engineering level using statistical methods. We

• 투고일 : 2012. 3. 23, 심사일 : 2012. 4. 20, 게재확정일 : 2012. 6. 12.

* 제1저자, 정보통신산업진흥원 SW공학센터(SW Engineering Center of National IT Promotion Agency)

** 교신저자, 강남대학교 경영대학(School of Business Administration, Kangnam University)

found that the software development productivity is different across the software engineering levels and the higher software engineering level can lead to improved software development productivity.

▶ Keywords : Software Process Improvement, Productivity of SW Development, Software Engineering Level

I. 서론

IT의 광범위한 보급과 확산은 기업들에게 기업의 운영과 전략적인 측면에서 많은 혜택을 제공하고, 거리적 제약이나 가격 및 제품 관련 정보의 불투명성과 같은 시장의 투명성을 높임으로써 소비자들의 협상력을 높여주었다. 이에 따라 IT 제품이나 서비스에 대한 수요가 빠르게 증가하면서 이를 구성하는 하드웨어, 소프트웨어 및 네트워킹의 장비관련 프로젝트들에 대한 개발 일정에 대한 압력이 과거 어느 때 보다 높아지고 있다[12].

하지만 하드웨어 생산 속도나 네트워크의 성능은 생산 자동화나 기술적 혁신을 통해서 빠르게 향상되고 있으나 SW 개발은 하드웨어나 네트워크처럼 혁신적인 성과를 거두지 못하고 있다. 이로 인해 IT의 중요한 구성요인인 SW 개발 프로젝트에 대한 개발일정 비용의 압박이 시달리게 되고, 최종 SW 제품의 품질을 떨어뜨려서 고객 불만족이 높아지고 있다.

2009년 Standish Chaos Report에 의하면, 고객이 요구하는 특징과 기능을 갖춘 시스템을 주어진 납기와 비용으로 개발한 프로젝트의 성공률(successful)은 32%에 불과하고, 요구사항을 만족시키지 못하거나, 납기 또는 비용을 초과한 프로젝트인 도전(challenged) 프로젝트의 비율이 44%, 중간에 중단되거나 결과물이 실제로 사용되지 못하고 실패한 프로젝트의 비율은 24%에 이르는 것으로 조사되고 있다[15].

SW 개발 프로젝트는 여러 가지 독특한 특성들이 존재한다. 대표적으로 SW는 사람의 창의적인 사고가 반영되어야 하기 때문에 인간의 지속적인 개입이 필요하고, SW 자체의 높은 복잡성으로 인해서 SW 코드의 대량생산을 위한 코드 자동화가 불가능하다. 또한, SW의 가변적이고, 혁신적인 속성들 때문에 프로젝트별 경험이나 지식공유가 어렵다. 마지막으로 SW의 비가시성 때문에 지속적인 생산성을 측정하고, 개선사항을 확인하는 작업이 복잡할 뿐만 아니라 개발현장의 저항이 많다. 이런 요인들이 SW 프로젝트의 관리 및 통제를 어렵게 만드는 요인으로 작용하기도 한다.

SW 개발 생산성은 SW개발 프로젝트의 세 가지 주요 요

소인 기술, 인력, 그리고 프로세스에 대한 균형적인 시각과 접근방법이 필요하다[14]. SW 개발 생산성을 높이기 위한 노력들이 주로 SW구축 관련 기술적인 측면을 강조하고, 관련 대안들이 제시되어 왔다[26]. 예를 들어, 전문적인 컴파일러, 미들웨어 및 스크립트 작성 기술들이 활용됨에 따라 프로그램 속도가 빨라지고, 프로그램 작성 유틸리티가 활용됨에 따라 SW의 결함 식별과 형상관리 활동이 용이해진 것이다. 또한 의사소통 관련 응용 프로그램과 통신망들은 SW개발 프로젝트와 시스템에 대한 정보의 투명성을 높여서 개발자 사이의 원활한 의사소통을 통해 생산성을 높이려고 노력하였다.

하지만, SW 개발의 인간 중심적 특성 때문에 앞서 언급한 기술적 개선의 혜택이 SW 개발 과정에 완전하게 실현되기 위해서는 필요한 인력과 개발 과정에 대한 프로세스가 뒷받침되어야 한다. SW 프로젝트의 성공에 대한 궁극적인 의미는 SW프로젝트가 완료된 이후 예상되는 비즈니스 변화와 이에 대응하는 가치를 제대로 전달하고, 이들 가치가 실제로 조직에 반영해야만 완벽한 SW프로젝트 성공이라 할 수 있을 것이다.

많은 SW 기업들이 자신들의 SW 제품의 품질향상과 더불어 주어진 납기 및 비용을 준수하고, 제품을 적시에 출시하기 위해 CMMI나 ISO/IEC 15504(SPICE)와 같은 SW 프로세스 개선을 추진하고 있다. 하지만 SW 프로세스를 개선하기 위해 비용과 인력이 많이 소요된다는 점을 고려하면 SW프로세스 역량수준과 SW프로젝트 성과에 대한 객관적인 증거를 제시하는 일은 매우 중요하다. 따라서 본 연구의 목적은 국내 SW 기업들의 SW공학수준과 SW 프로젝트의 성과변수인 생산성과의 관계를 실증 데이터를 통해 분석하는 것이다.

II. 관련 연구

1. 관련연구

생산성은 특정 활동을 계획하거나 모니터링 하는데 쓰이는 가장 일반적이고 유용한 개념 중의 하나이다. 생산성의 개념은 CMMI[3], ISO/IEC 15504[10]와 같은 SW프로세스

개선 모델과 SW 공학영역에서 SW개발 프로젝트를 계획하거나 관리하기 위해 고려해야 할 중요한 가치요소로 간주되고 있다. 일정 시간동안 투입하는 노력에 대한 산출물의 비율의 개념으로 SW 개발 생산성은 다음과 같이 정의되고 있다.

$$\text{SW 개발 생산성} = \text{크기(size)/노력(efforts)}$$

SW 개발 생산성을 계산하는 수식은 아주 간단해 보이지만 SW 크기와 투입하는 노력을 객관적으로 측정하여 실제 개발 현장에서 유용하게 활용될 수 있게 하는 작업은 단순한 작업이 아니다. SW 크기는 고객에게 제공하는 기능관점에서 기능을 반영하여야 하고, 노력은 새로운 기능을 개발하는 작업의 양을 측정하여야 한다.

SW 개발 프로젝트의 생산성은 SW 규모 측정과 매우 밀접한 관계가 있다. SW규모 대비 투입노력의 비율로 산정되기 때문에 산출된 생산성을 다른 생산성 값과 비교하기 위해서는 측정지표의 표준화가 선행되어야 한다. 이를 구체적으로 살펴보면, 70년 말부터 Albrecht[1][2]의 기능점수 방식이 기존의 라인수 방식(Line Of Code, LOC)을 대신하여 SW의 기능측면 규모를 산정하는 지표로 제안되었다. 이때, 생산성은 기능점수 대비 전체 프로젝트 투입 노력(effort)의 비율로 산정된다[7][8].

SW 공학 커뮤니티에서 SW 규모 산정을 위해 기능점수 방식이 도입되고, 90년대 초기에 기능적 규모 산정 방식(Functional Size Measurement Method, FSM)을 검증하기 위한 일반적인 기준을 만들기 목적으로 ISO/IEC JTCl/SC 7 워킹 그룹(WG12)이 만들어졌다. 이후 FSM은 공식 표준으로 채택되었다.

특히, ISO/IEC 14143-1[9][11]은 기능점수 개념에 대한 ISO의 정의와 FSM의 모든 변종들과 비교할 수 있는 기반을 제공하고 있다. 또한 변형된 FSM 방법들이 ISO/IEC 14143-1의 규정과 호환성이 있는지를 검토할 수 있는 프로세스를 제공하고 있다. ISO는 SW의 규모를 산정하기 위해 세 가지 유형(기능, 품질 및 기술)의 요구사항 중에서 기능적 요구사항만을 FSM 방식의 투입요소로 언급하고 있다. 생산성을 정의하는데 성과(Performance)와는 명확하게 구분되어야 한다. 성과는 프로젝트에서 산출되는 가능한 모든 결과물(outcome)의 상위 부분을 고려한 생산성보다 포괄적인 개념이다.

2. SW 공학(능력)수준

미국 SEI의 CMM(Capability Maturity Model: CMM)이 산업표준으로 인식되면서, SW 프로세스 개선(Software Process Improvement)은 SW공학을 연구하는 학계와 이를 적용하는 개발 현장에서 이슈가 되어왔다. SW 프로세스 개선의 목표는 보다 효과적이고 효율적인 SW 개발과 유지보수가 가능하게 개발 프로세스를 구조화하고 최적화하는 것이다. 이러한 SW 프로세스가 내포하고 있는 기본적인 가정은 사전에 정의된 공학 프로세스를 기반으로 잘 관리되는 조직이 주어진 일정과 예산 내에서 고객의 요구사항을 잘 충족시키는 제품을 만들어낸다는 것이다.

본 연구의 SW공학수준은 기본적으로 미국 SEI(Software Engineering Institute)에서 개발한 CMMI(Capability Maturity Model Integration)[4]을 기반으로 개발되었으며, 프로세스, 인력 및 기술수준으로 구성되었다.

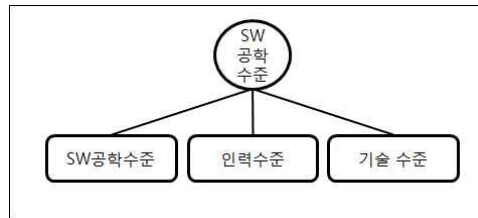


그림 1. SW공학수준의 구성
Fig.1. Structure of SW engineering level

프로세스(Process) 영역은 성공적인 SW개발 활동에 있어서 반드시 수행되어야 하거나 수행이 필요한 활동들이 SW 조직에서 제대로 수행하고 있는지를 확인한다. 또한 이런 활동들이 일회성이 아닌 지속적으로 수행될 수 있는 기반이 되는지를 확인한다. 인력(People) 영역의 지표는 원활한 SW개발을 위하여 필요한 조직을 보유하고 있는지를 확인한다. 조직원들의 역량강화와 필요한 인력을 양성하기 위한 교육과정이 제대로 갖추어져 있는지를 점검한다. 마지막으로 기술(Technology) 영역은 SW기업의 조직원들이 SW 개발을 제대로 수행하기 위하여 필요한 인프라인 프로세스 체계, 자동화 시스템 및 툴 보유 등의 구성과 활용이 어떠한지, 프로세스 자산 및 조직 정보 관리 및 활용이 어떠한지 등 특정 기술에 대한 수준이 아닌 SW개발을 위해 기본적으로 갖춰야 하는 시스템 인프라, 개발 표준 및 기법 등을 확인하는 항목들이다.

표 1. SW공학수준 영역별 측정 내용
Table 1. Measures of SW engineering level

구성요소	설 명
프로세스	SW개발 시 조직차원에서 필요한 활동들의 이행 정도를 확인
	조직 내 프로세스의 지속적인 이행과 개선을 위한 활동의 정도를 확인
인력	원활한 SW개발을 위한 조직차원의 전문 자원조직 체계의 구성 수준을 확인
	SW개발 시 필요한 주요 역량을 보유한 사내 전문가 보유 수준을 확인
	SW개발 시 프로세스 이행을 위한 조직 역할체계가 구성되어있는지 확인
	조직원의 역량 강화를 위한 교육 체계, 경영진 지원 등의 수준을 확인
	외주 인력, 인력 백업체계 및 순환 체계 등 조직 인력 운영 수준을 확인
기술	체계적인 SW개발 활동에 필요한 프로세스 체계 구성의 적절성을 확인
	효율적인 SW개발 및 관리를 위한 자동화 시스템 구성의 적절성을 확인
	SW개발 각 단계별 필요한 개발 표준 및 기법의 보유 수준을 확인
	SW개발 시 참조 가능한 정보의 보유 및 관리 등 정보 공유 수준을 확인

특히, SW 공학수준의 프로세스 분야는 CMMI모델의 능력수준(Capability Level) 2와 3의 18개 주요 프로세스 영역(Key Process Area; KPA)을 바탕으로 개발되었다. CMMI 레벨 2는 프로세스 기반으로 프로젝트가 통제되는 수준으로, 조직이 프로세스에 대한 일정정도의 훈련이 되어있으나, 일정이나 비용과 같은 관리 프로세스 중심이다. CMMI 레벨 3은 프로젝트 수준의 프로세스가 아니라 조직차원의 표준 프로세스가 존재하고, 개별 프로젝트마다 이를 테일러링해서 사용하는 수준을 의미한다(표 2 참조).

표 2. CMMI 레벨2와 3의 주요 프로세스 영역
Table 2. Key process areas in CMMI level 2 and level 3

	프로젝트 관리	공학	지원	프로세스 관리
level 2	프로젝트 계획수립	요구사항 관리	형상 관리	
	프로젝트 모니터링 및 통제		프로세스 및 제품품질 보증	
	공급업체관리		측정 및 분석	
level 3	통합프로젝트 관리	요구사항 개발	의사결정 분석 및 해결	조직 프로세스정의
	위험관리	기술적 해결		조직프로세스 중점관리
		제품통합		조직교육 관리
		검증		
		확인		

CMMI의 개별 프로세스들은 달성해야 할 고유목표(Specific Goal)와 공통 목표(Generic Goal)를 가지고 있으며, 이런 목표를 달성하기 위해 수행되어야 하는 활동으로 구성되어 있다(그림 3 참조). 이를 기반으로 SW공학수준 점수를 산출하기 위해 각 프로세스의 활동들의 수행 여부를 2점 척도(수행/미수행)로 측정하고, 수행하면 1점 아니면 0점을 부여하여 SW공학 수준점수를 산출하였다. 인력과 기술 영역의 SW공학 수준점수도 동일한 척도를 기준으로 개별 프로세스를 수행하는 인력이나 기술이 존재하면 1점을 아니면 0점을 부여하였다.

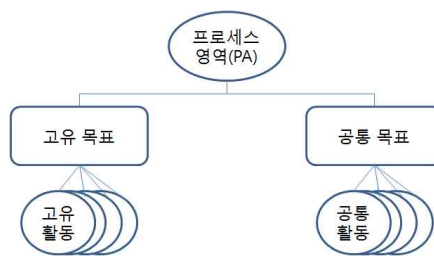


그림 2. CMMI의 프로세스 영역의 구조
Fig.2. Structure of CMMI's process area

이렇게 산출된 SW공학수준점수를 기반으로 기업의 특성을 파악하기 위해 Advanced, Average, Absent의 3가지

등급으로 분류하였다. 이는 SW공학수준과 CMMI의 Level 과 벤치마킹을 위해 CMMI의 Level 체계와 유사한 구조로 구성하였다. 하지만 등급을 부여하는 방식에서 다소 차이가 있다. 예를 들어 CMMI Level2는 CMMI의 Level 2영역인 7개의 영역만을 대상으로 심사가 이루어지지만, SW공학수준 등급은 전체 18개 영역을 대상으로 점수가 산출된다는 차이가 있다. SW공학수준별 SW기업의 전반적인 역량 특성에 대한 설명은 표 3과 같다.

표 3. SW공학 수준별 기업의 특징
Table 3. Organization's characteristics of SW engineering levels

등급	점수 분포	등급별 SW기업 특성
Advanced	80점 이상	프로젝트 관리, 개발, 지원, 프로세스 관리 활동이 균형을 이루며 실제 업무 환경에 잘 적용되어 그 효과가 충분히 발생되고 있는 조직
Average	60점 이상 ~ 80점 미만	프로젝트 관리, 개발, 지원, 프로세스 관리 활동의 일부만이 실제 업무 환경에 적용되어 일부의 효과가 발생되고 있으며, 해당 활동들의 내재화가 일부 수행되고 있는 조직
Absent	60점 미만	프로젝트 관리, 개발, 지원, 프로세스 관리 활동의 대부분이 실제 업무 환경에 적용되지 못하고 있으며, 해당 활동들의 내재화 또한 수행되고 있지 않은 조직

3. SW 프로세스 개선과 성과

SW 개발 프로젝트의 일반적인 목표는 최소 일정과 비용을 투입하여 최상의 품질과 최대의 고객만족을 얻는 것이다. 이러한 SW 프로젝트의 성공은 최신 기술이나 기법, 전문적인 조직이나 인력, 기반 시스템 및 도구의 활용 등 여러 요소들에 영향을 받으며, 프로젝트의 규모, 특성, 상황 등에 따라 적절하게 SW프로젝트의 성공 목표가 설정되어야 한다. 또한, SW프로젝트 수명주기상 경영자, 발주자, 개발자, 관리자 등 여러 계층의 관심사는 다르므로 SW프로젝트의 성공 기준 또한 서로 다르며, 모두를 만족시키는 것은 쉽지 않은 일이다[24].

많은 기업들이 성공적인 SW 개발과 생산성을 높이기 위한 방안으로 SW 프로세스 개선을 추진하고 있다[25]. 실제로 영국의 85개 회사를 대상으로 조사한 자료에 따르면, 전체 55%의 해당하는 기업들이 SW 개발 생산성 개선을 SW 프로세스 개선의 목표로 하고 있는 것으로 제시되고 있다.

SW 프로세스 개선의 혜택은 단지 개발업무의 속도를 올리는데 국한되지 않으며, 결함 수정작업에 소요되는 재작업 시간을 줄일 수 있게 된다. 적절한 개발 프로세스가 없을 경우, 개발 팀 운영은 혼란스러울 수밖에 없고, 이는 낮은 생산성과 품질로 이어지게 된다[23].

SW 프로세스는 고품질 SW를 개발하기 위해 필요한 SW 수명주기 상의 핵심적인 활동들을 제시하고 있는 작업체계를 의미한다. 보다 체계적이고 성숙된 SW프로세스 역량은 보다 높은 품질의 SW제품 생산이 가능하다는 것이 SW프로세스의 기본 사상이다. 이미 많은 SW개발 조직들은 SW 프로세스 역량 개선이 SW 프로젝트 성과 및 SW 제품 품질 향상으로 이어진다는 것을 제시하고 있다.

대표적인 SW 프로세스 개선 모델인 CMMI를 도입한 많은 SW관련 조직들은 프로세스 개선을 통해 비용, 납기, 품질, 고객만족도와 같은 성과를 효율적으로 달성한 것으로 제시되고 있다. CMMI를 적용한 기업들의 적용 효과를 살펴보면 결함 수정 비용이 평균 33% 감소되어 전체 프로젝트 비용을 절감하는 효과가 나타났고, 테스트 이후 재작업을 60% 감소시켜서 프로젝트 전체 일정을 단축하는 효과가 있었다. 또한 고객 만족도는 평균 14%, 품질은 평균 48% 향상되었다고 한다[6].

Gibson et al.[5]은 CMMI를 기반으로 프로세스 개선을 추진한 35개 조직들을 대상으로 조사한 성과를 비용, 스케줄, 생산성, 품질, 고객만족도 및 투자수익율과 같은 6가지 측면에서 분석하였다(표 4 참조). 분석결과는 개선의 정도를 기반으로 상/중/하위 그룹으로 나누어 제시하고 있다. 전반적으로 CMMI를 통한 프로세스 개선의 영향이 긍정적으로 나타나고 있다. 특히, 납기분야를 살펴보면 최저 2%수준까지 최고 95%까지 개선되고 있음을 알 수 있다.

표 4. CMMI 적용 성과분석 결과
Table 4. CMMI performance results

성과	중위 그룹 (median)	하위그룹 (Lowest)	상위그룹 (Highest)
비용	34%	3%	87%
납기	50%	2%	95%
생산성	61%	11%	329%
품질	48%	2%	132%
고객 만족도	14%	-4%	55%
투자대비 수익(ROI)	4:1	1.7:1	22.7:1

III. 데이터 수집 및 측정지표

1. 데이터 수집

본 연구에 사용된 데이터는 정보통신산업진흥원의 SW공학센터에서 수행되는 SW공학수준조사를 통해 수집되었다. 조사기간은 2010년 9월부터 11월에 걸쳐 약 3개월에 걸쳐 수집되었다. 분석에 사용된 자료는 총 75개 Project Data가 활용되었다.

설문 조사 및 분석대상은 SW개발 프로젝트이고, 설문 응답자는 해당 프로젝트를 수행하는 프로젝트 관리자(Project Manager)나 프로젝트의 리더(Project Leader)들로 구성되었다. SW 개발 프로젝트는 IT 서비스, 패키지 SW 및 임베디드 SW 개발 프로젝트가 주류를 이루고 있다.

2. SW공학수준 측정지표

2.1 프로세스 수준 지표

프로세스 수준 지표는 SW개발 수행 시 조직 차원에서 필요한 활동들의 이행 수준과 원활한 이행을 위하여 필요한 내재화 수준의 확인을 목적으로 SW개발 수행 시 조직 차원에서 필요한 활동들의 이행 수준을 확인하기 위한 프로세스 이행수준 지표와 필요한 활동들이 조직 내 내재화하기 위하여 필요한 사항들의 수준을 확인하기 위한 프로세스내재화수준 지표로 구성되었다. 프로세스 수준 측정 지표는 18개 프로세스 영역에 대해 총 145개의 문항으로 구성되었다.

표 5. 프로젝트 계획 프로세스의 측정내용
Table 5. Measurement of project planning process

프로세스	측정 내용
프로젝트 계획	프로젝트 범위 건적 이행여부
	작업산출물 및 작업 속성 건적 이행여부
	프로젝트 개발방법론 정의 여부
	노력 및 비용 건적 이행여부
	예산 및 일정수립 이행여부
	프로젝트 위험요소 식별 이행여부
	데이터 관리 계획 수립 이행여부
	프로젝트자원계획수립 이행여부
	필요지식 및 기술 확보 계획수립 이행여부
	관련 이해관계자 참여계획 수립 이행여부
	프로젝트 계획 수립 이행여부
	프로젝트에 영향을 주는 타 계획 검토 이행여부
	직업 및 지원 수준조정 이행여부
계획에 대한 공동합의 확보 여부	

2.2 인력수준지표

원활한 SW개발을 위하여 필요한 인력 보유, 지원 조직의 보유, 인력 육성을 위한 교육 수준 등 인력 측면에서 필요한 사항이 제대로 갖추어 졌는지를 확인하기 위한 지표로 국내 SW기업의 조직 및 인력육성에 대한 현황을 이해할 수 있으며, SW 개발을 적절히 수행하기 위하여 필요한 조직의 역할, 인력, 교육 등을 측정하였다. 전문 지원 조직 체계의 측정항목을 살펴보면, 총 9개의 측정항목으로 구성되었다. 내용은 아래의 표와 같다(표 6 참조).

표 6. 전문지원조직체계의 측정항목
Table 6. Measurement of organizational support

구분	측정 내용
전문 지원 조직 체계	개발관련 지원 전문조직 보유여부
	테스팅 지원 전문조직 보유여부
	프로젝트 관리 전문조직 보유여부
	간접 지원 전문조직 보유여부
	업체(외주업체 포함)관리 조직 보유여부
	품질보증 전문조직 보유여부
	데이터 측정 및 분석 지원조직 보유여부
	프로세스 개선 전문 조직
	교육훈련 전문조직 보유여부

2.3 기술수준 지표

SW개발 기업의 조직원들이 SW개발을 제대로 수행하기 위하여 필요한 인프라(프로세스 체계, 자동화 시스템 및 툴 보유)의 구성과 활용 여부, 프로세스 자산 및 조직 정보 관리 활용 정도 등으로 특정 기술에 대한 수준이 아닌 SW개발에 있어서 기본적으로 갖추어지거나 적용되어야 하는 기술(인프라, 체계, 기법, 시스템, 툴 등) 수준을 확인하기 위한 지표로 국내 SW기업의 기본 인프라(프로세스, 자동화 시스템, 툴, 정량적 데이터 체계 등), 개발에 필요한 기본 기법 및 표준 등으로 구성되었다. 프로세스 수준 측정 지표는 총 42개의 문항으로 구성되었다.

표 7. 시스템 인프라의 측정내용
Table 7. Measurement of system infrastructure

구분	측정내용
시스템	조직 표준 프로세스 및 산출물의 통합

인프라	시스템 보유 여부 프로젝트 관리 시스템 보유여부 공수 수집관리 시스템 보유여부 형상관리 툴 및 시스템 보유여부 모델링 전용툴 보유여부 요구사항 관리툴 보유여부 테스트 관련 시스템 보유여부
-----	--

2.4 SW 프로젝트 생산성

SW개발 개발 생산성은 앞에서 언급한 바와 같이 SW 개발 프로젝트에 단위 투입인력(1MM)을 통해 얼마나 많은 SW의 기능을 산출하는지를 측정하였다. 즉, 전체 SW개발 프로젝트의 규모를 기능점수 방식을 통해 산출하고 이를 해당 프로젝트에 투입되는 총 M/MM로 나누어 산출하였다.

$$\text{SW개발 생산성} = \frac{\text{프로젝트 전체 FP}}{\text{프로젝트 투입 MM}} \quad (\text{단위 FP/MM})$$

IV. 데이터 분석방법

1. 일원 분산분석

SW공학수준에 따른 생산성의 차이를 검증하기 위해서, 두 개 이상 다수의 집단 간 평균의 차이를 검증하는 통계분석 방법으로, 독립변수가 하나일 경우에 사용하는 일원분산분석을 실시하였다. 이 분석방법은 표본의 독립성, 모집단의 정규성, 모집단의 등분산성과 같은 3가지의 가정 사항(assumption)을 가지고 있으며, 귀무가설은 집단 간 평균은 차이가 없다는 것이다. 독립변인(처치효과)과 종속변인 사이의 일원분산의 통계적 모형은 다음과 같다.

$$Y_{ij} = \mu + \alpha_i + \varepsilon_{ij}$$

Y_{ij} 는 i 개의 표본 집단과 j 개의 관찰치를 의미하고, α_i 는 i 개의 표본 집단의 효과를, ε_{ij} 는 i 개의 표본 집단과 j 개의 관찰치의 오차를 의미한다[13].

2.붓스트랩과 모델 안정성

Efron[16, 17, 18, 19]의 붓스트랩(bootstrap)은 컴퓨터 기반의 비모수적 방법으로 샘플 추정치의 정확성을 평가하기 위한 컴퓨터 기반 통계기법이다. 이 기법은 매우 단순한

방법을 활용하여 거의 모든 통계적 표본의 분포를 추정할 수 있게 해주는 기법이다[20]. 일반적으로 붓스트랩은 재표본방법(resampling technique)을 의미한다. 주어진 확률표본 x_1, x_2, \dots, x_n 으로부터 복원 추출방법으로 붓스트랩 표본의 붓스트랩 표본 $x_1^*, x_2^*, \dots, x_n^*$ 을 얻는다. 추출 가능한 붓스트랩 표본의 전체 수는 n^n 개이며, 붓스트랩 표본으로부터 n^n 개의 $\hat{\theta}^*$ 를 얻는다. 각각의 θ 의 추정 값이 되며 또한 이들은 $\hat{\theta}$ 의 붓스트랩 분포를 구성한다.

본 연구에서는 조사된 SW공학수준에 따른 품질비용의 차이에 대한 붓스트랩 신뢰구간을 도출하기 위해 1000개의 붓스트랩 표본으로부터 $\hat{\theta}$ 를 구하고 순서대로 정렬하였다. Efron[24]는 정확한 신뢰구간 추정을 위해서는 최소 1,000개의 붓스트랩 표본이 적절하다고 제안하였다. 수집된 SW품질비용 데이터가 전체 국내 SW개발 기업을 대상으로 무작위 추출된 것이 아니기 때문에 분석결과의 안정성(stability)을 검증하는 작업이 필요하다. 이를 위해 본 연구에서는 수집된 데이터에서 1000번 이상 복원 추출하는 스트랩 표본 재추출 방법을 활용하였다.

예를 들어, 일원분산분석으로 F값, 평균, 분산과 같은 값들이 산출되며, 이때 산출되는 개별 값들에 대한 신뢰구간은 95% 수준에서 결정된다. 경험적 기준분포(empirical reference distribution)를 기반으로 산출되는 신뢰구간은 경험적 신뢰구간(Empirical Confidence Interval; ECI)이라고 한다. 붓스트랩 방법은 정규분포나 동질성(homogeneity)과 같은 제약사항에서 자유로운 방법이다.

원 수집자료의 분석을 통해 산출된 통계 값들이 경험적 신뢰구간의 중앙에 위치하면 안정적인 것으로 분류된다. 즉, 경험적 신뢰구간의 상/하위 경계에 존재하지 않아야 한다는 것을 의미한다. 이러한 안정성 편향(stability bias)을 평가하는 식은 다음과 같다.

$$\text{Bias} = \frac{\sum_{b=1}^B t_b^*}{B} - \hat{\theta}$$

t_b^* 는 b 번째 샘플의 통계 값이고, b 는 $1, \dots, B$ 를, $\hat{\theta}$ 는 원 수집자료에서 분석된 통계 값을 의미한다. 편향의 정도는 B 번의 반복을 통해 나온 분포의 표준오차(standard error)를 기반으로 평가되며, 이때 표준오차는 다음과 같이 계산된다.

$$SE = \sqrt{\frac{\sum_{b=1}^B (t_b^* - \bar{t}^*)^2}{B-1}} \quad \text{단, } \bar{t}^* = \frac{\sum_{b=1}^B t_b^*}{B}$$

만약 편향(bias)이 표준오차에 비해 크다면, 안정성에 문제가 있는 것이다. 안정성을 판단하는 기준(criteria)은 편향의 절대값이 표준오차의 1/4보다 적다면 편향은 무시될 수 있다[19].

붓스트랩 방법은 실증적 SW공학연구에서 이전부터 활용되어왔다. El-Emam과 Garro[21]는 포획/재포획(capture-recapture)방법을 활용하여 ISO/IEC 15504 심사 횟수를 추정하였다. Jung과 Hunter[22]는 개별 ISO/IEC 15504 프로세스의 능력 수준(capability level)의 신뢰구간을 계산하는데 붓스트랩을 활용하였으며, Jung과 Goldenson[23]은 SW-CMM의 내적 일관성(Internal consistency)에 대한 안정성을 평가하기 위해 붓스트랩을 활용하였다.

V. 실증적 분석결과

1. 기술 분석

분산분석 이전에 SW공학수준을 기준으로 SW 개발 프로젝트 평균 개발 생산성과 빈도를 분석해 보았다. SW공학수준별 프로젝트의 분포를 살펴보면, Absent 등급이 14개, Average 수준 등급이 38개, Advanced 등급이 23개로 Average 등급이 차지하는 비율이 높을 것을 알 수 있다. Advanced와 Average 등급이 Absent 등급보다 높은 비율을 차지하는 이유는 SW 공학수준이 낮을 경우 SW 개발 프로젝트에 대한 기본적인 측정 및 분석이 이루어지지 않기 때문이다.

SW공학 수준별 SW 개발 프로젝트의 생산성을 살펴보면, Absent 수준 등급이 16.8 FP/MM, Average 수준등급이 23.5 FP/MM, Advanced 수준 등급이 23.7 FP/MM로 나타나고 있다. 이는 SW 공학수준 등급이 높을수록 SW 개발 프로젝트의 생산성이 높다는 것을 간접적으로 보여주고 있다.

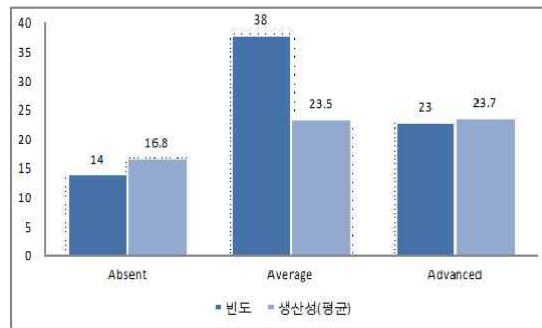


그림 3. SW공학수준별 분포
Fig.3. Distribution of SW engineering levels

2. 상관관계 분석

일원분산분석에 앞서 변수간의 관계에 대한 사전정보를 얻기 위하여 SW공학 수준점수와 SW개발 생산성간의 피어슨 상관관계를 분석하였다. SW공학수준과 SW개발생산성의 상관관계수가 0.529로, 이들 간에 양의 상관관계가 있음을 알 수 있다(표 8 참조). 즉, SW공학수준이 높아질수록 SW개발 생산성도 높아지고 있다는 것을 알 수 있다.

표 8. 상관관계 분석결과
Table 8. Correlation analysis result

	SW공학수준	생산성
SW공학수준	1	0.529**
생산성	0.529**	1

**유의수준 <0.01

모델의 안정성을 검증하기 위해 SW개발 생산성에 대해 1,000번의 재표본 추출을 통해 1,000개의 SW개발 생산성 세트를 추출하고, 이를 기반으로 붓스트랩을 통해 추출된 데이터 세트의 평균값(boot.means)과 표준오차(Standard Error; SE)값을 구하였다. 분석결과를 살펴보면, 원래 표본의 평균은 22.33이고 편차는 0.003이고, 표준오차는 0.59로 분석되었다.

표 9. 붓스트랩 평균과 편차
Table 9. Bootstrap mean and standard error

원래 평균 (Original)	편차 (Bias)	표준오차 (Standard Error)
22.33	0.003	0.59

편차 0.003은 표준오차인 0.59의 1/4 값인 0.15보다 작다는 것을 알 수 있다. 이는 앞서 살펴본 모델의 안정성 기준을 만족시키고 있음을 나타낸다. SW 개발 생산성에 대한 평균값의 분포는 아래와 같다(그림 7참조).

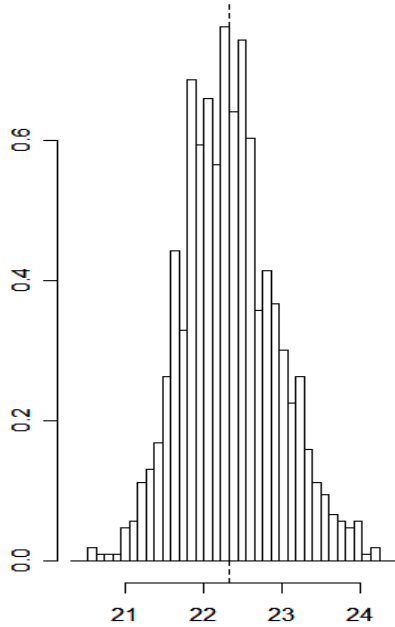


그림 4. 붓스트랩 평균값의 분포(복원 추출 1000회)
Fig. 4. Distribution of bootstrap means(1000 resampling)

수집된 자료를 SW공학수준별로 분류하고, 개별 SW공학 수준별로 10,000번의 복원 추출한 샘플을 기반으로 일원분산분석을 실시하기 위한 경험적 기준분포인 새로운 F* 값들의 분포를 구하였다(그림 8 참조).

새로운 F값의 분포를 기준으로 유의수준 0.01%수준의 F값(2, 72)은 4.91로 분석되고 있다. 앞에서 살펴본 것처럼 수집된 자료의 F값은 13.3으로 붓스트랩을 통해서 분석된 유의수준 1%의 F(2, 72)의 값인 4.91보다 크기 때문에 유의수준 1%에서 통계적 유의성이 있다는 것을 보여주고 있다. 즉, SW공학수준에 따라 SW개발 생산성의 평균값들의 차이가 존재한다는 것을 알 수 있다.

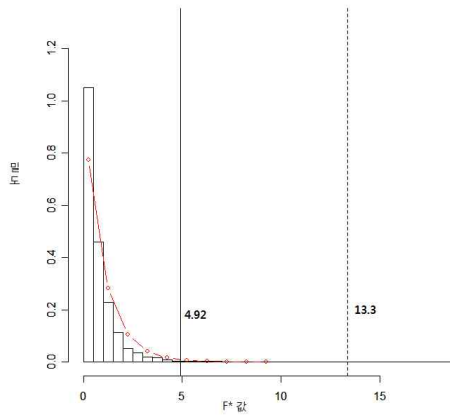


그림 5 . 붓스트랩 F값의 분포
Fig.5. Distribution of bootstrap F values

분산 분석의 F-검정은 SW공학수준에 따른 전체적인 차이를 검증결과를 보여준다. 그러므로 수준별 차이(예: Absent 평균= Average 평균)가 있는지 혹은 수준별 결합 대비(contrast)차이가 있는지 검정해야 하는데 이를 사후검증이라고 한다. 여러 가지 사후분석 방법 중에서 가장 보수적인 방법인 Tukey HSD(Honestly significant difference) 방법을 사용하였다.

Tukey HSD 분석결과를 살펴보면, Absent 수준등급과 Advanced 수준 등급 간, Absent 수준등급과 Advanced 수준 등급의 SW 프로젝트 평균 생산성 차이는 존재하지만 Average 수준등급과 Advanced 수준 등급의 SW 개발 생산성의 차이는 유의하지 않는 것으로 분석되었다.

표 10. Turkey HSD 분석 결과
Table 10. Results of Turkey HSD analysis

공학수준(I)	공학수준(J)	평균차(I-J)	유의확률
Absent	Average	-6.717*	.000
	Advanced	-6.894*	.000
Average	Absent	6.717*	.000
	Advanced	.177	.987
Advanced	Absent	6.894*	.000
	Average	.177	.987

Average 수준이 Advanced 수준사이의 SW개발 생산성의 차이가 유의하지 않은 것은 SW개발 생산성에 영향을 미치는 요인들 때문인 것으로 보인다. 즉, 개발하는 SW개발의 복잡도와 난이도 등이 영향을 미쳤을 것으로 보인다. 즉, 개발하는 SW 개발 프로젝트의 난이도나 복잡도가 낮을 경우 SW

개발 생산성의 차이가 두드러지게 나타나지 않을 수 있다.

VI. 결론 및 한계점

본 연구는 IT 용·복합화가 가속화되면서 기존 제품의 부가가치 및 경쟁력을 높이는데 중요한 역할을 수행하는 SW 개발 프로젝트의 생산성을 높이기 위해 SW공학 적용 수준을 높이는 노력의 투입이 타당한 것인지를 확인하는 것이다.

이를 위해 국내 SW개발 프로젝트를 대상으로 SW공학수준 현황과 SW개발 생산성 자료를 수집하여 SW프로젝트 조직의 SW공학수준에 따라 품질비용에 차이가 존재하는지를 살펴보았다. 본 논문의 연구 결과를 요약하면 다음과 같다.

SW공학수준별 SW 개발 프로젝트의 평균 생산성의 차이를 분석하기 위한 일원 분산분석 결과, SW공학 수준에 따라 SW 개발 프로젝트의 생산성 차이가 존재하는 것으로 분석되었다. SW공학 수준별 생산성을 구체적으로 살펴보면, Absent 수준 등급이 16.8 FP/MM, Average 수준등급이 23.5 FP/MM, Advanced 수준 등급이 23.7 FP/MM로 나타나고 있다. 이 결과는 SW공학수준에 따라 SW개발 생산성이 다를 뿐 아니라 SW공학 수준이 높을수록 개발 생산성이 높다는 것을 보여주고 있다. 조직의 SW공학 적용수준이 높아지면서, 제작업 비율이 줄어들면서, 자연스럽게 조직의 SW개발 생산성 향상으로 이어진 것으로 보인다.

이는 성공적인 SW개발 프로젝트 수행을 위해 SW공학을 적용하여, SW공학수준을 높이는 작업이 필요하다는 것을 보여주고 있다. 따라서 향후 국내 SW기업들의 SW품질향상과 SW개발 생산성향상을 위해서는 SW공학 수준의 향상과 이에 대한 효율적인 적용을 위한 노력을 기울여야 함을 알 수 있다.

일반적인 실증적인 연구들과 마찬가지로 본 연구도 몇 가지 한계점을 가지고 있기 때문에 이를 극복하기 위한 추가적인 연구가 반드시 필요하다. 먼저, 표본 집단을 국내 SW개발 프로젝트를 대상으로 하였지만 SW 개발 유형별 다양한 분석을 위한 충분한 샘플을 확보하지 못했다. 또한 데이터 수집단계에서 철저한 검증을 거쳤음에도 불구하고, 수집 및 분석에 활용된 데이터가 자동화된 도구나 시스템을 통해 직접 측정된 데이터가 아니라 응답자들이 스스로 작성하여 제출한 자료이기 때문에 자료에 다소간의 편차가 있을 수 있다는 점이다.

참고문헌

- [1] A.J. Albrecht, "Measuring Application Development Productivity," Proceedings of the IBM Applications Development Symposium, GUIDE/SHARE, October 14-17, 1979, Monterey, CA, 1979, pp.83-92.
- [2] A.J. Albrecht, "AD/M Productivity Measurement and Estimate Validation, IBM Corporation. New York, 1984.
- [3] CMMI, CMMI for Development Version 1.2, CMU/SEI-2006-TR-008, Technical Report, Software Engineering Institute, 2006
- [4] CMMI, CMMI Product Team CMMI for Development, Version 1.3, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 2010.
- [5] D.L. Gibson, D.G. Goldenson and K. Kost "Performance Results of CMMI-Based Process Improvement" ,Technical Report CMU/SEI -2006-TR-004, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 2006.
- [6] R. Hefner, "Achieving the Promised Benefits of CMMI", presentation file, CMMI Technology Conference & User Group, 14-17 November 2005.
- [7] ISBSG, Repository R10 Field Description, January 2007.
- [8] IFPUG, Function Points Counting Practices Manual (release 4.2), International Function Point User Group, Westerville, Ohio. January, 2004.
- [9] ISO/IEC 14143-1:1998 Information Technology - Software Measurement Functional Size Measurement - Part 1: Definitions of Concepts: International Organization for Standardization, 1998
- [10] ISO/IEC TR 15504-4, Information technology -- Process assessment-Part 5: An exemplar Process Assessment Model, 2006
- [11] ISO/IEC 14143-1:2007 Information Technology - Software Measurement -Functional Size Measurement - Part 1: Definitions of Concepts: International Organization for Standardization, 2007

- [12] C. Jones, "Applied Software Measurement: Global Analysis of Productivity and Quality," 3rd ed, McGraw-Hill, 2008.
- [13] M.K. Kerr, M. Martin, G.A Churchill, "Analysis of Variance for Gene Expression Microarray Data", Journal of Computational Biology, Vol.7, pp. 819 -837, 2000.
- [14] R. Ponce, "Steps to On-Time, On-Budget Project Delivery", ITworld, May 2009.
- [15] Standish Group, Chaos: A Recipe for Success, tech. report, Standish Group Int'l, 2009.
- [16] B. Efron, "Bootstrap Methods: Another Look at the Jackknife," The Annals of Statistics, Vol.7, pp.1-26, 1979.
- [17] B. Efron, The Jackknife, the Bootstrap and Other Resampling Plans, Society for Industrial and Mathematics, 1987.
- [18] B. Efron, R. Tibshirani, "Bootstrap Methods for Standard Errors, Confidence Intervals, and Other Measures of Statistical Accuracy", Statistical Science, Vol.1, No.1, pp.54-75, 1986.
- [19] B. Efron, R. Tibshirani, "An Introduction to the Bootstrap," Chapman and Hall, 1994.
- [20] K. El-Emam, & I. Garro, "Estimating the Extent of Standards Use: The Case of ISO/IEC 15504", Journal of Systems and Software, Vol.53, No.2, pp.137-143, 2000.
- [21] H-W. Jung, & R. Hunter, "An Evaluation of the SPICE Rating Scale with Regard to the Internal Consistency of Capability Measurement", Proceedings of the Second International SPICE Conference, pp.105-115, 2002.
- [22] H-W Jung, & D. Goldenson, The Internal Consistency of Key Process Areas in the Capability Maturity Model® (SW-CMM®) for Software, Carnegie Mellon University / Software Engineering Institute, CMU/SEI-2002-TR-037, 1994.
- [23] Paulk, M.C., Curtis, B., Chrisses, M.B., Weber, C.V. Capability Maturity Model for software. CMU/Software Engineering Institute, Technical Report, CMU/SEI-93-TR-024, (Feb. 1993)
- [24] CMMI Product Team CMMI for Development, Version 1.3, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 2010.
- [25] In-Oh Song, Sung-Yul Rhew, Sung-Eung Lee, "A Software Process Certification Model of Small sized Software Development Using Scrum," Journal of the Korea society of computer and information, Vol. 16, No. 4, 2011.
- [26] Byungha Choi, Sung-kyo Choi, Kyungsan Cho "A Design of Framework based on Distributed System for Enhancing Productivity", Journal of the Korea society of computer and information, Vol 17, No. 1, 2012.

저 자 소 개



김 승 권

2008 : 고려대학교 대학원 경영학과, 경영학박사(MIS)

현 재 : 정보통신산업진흥원 SW공학센터 수석

관심분야 : 소프트웨어 프로세스, 소프트웨어 품질, 전자상거래, 정보시스템 성과 분석

E-mail : sgkim@nipa.kr



윤 종 수

2000 : 고려대학교 대학원 경영학과, 경영학박사(MIS)

현 재 : 강남대학교 경영대학 교수

관심분야 : e-Business/EC 기획 정보시스템 활용 및 평가, 모바일 인터넷 서비스, CRM/BSC 응용

E-mail :

jongsoo@kangnamackjion

