

# 소프트웨어 프로세스 능력 및 성숙 수준에 관한 실증적 비교 분석과 평가 시스템 개발

(Empirical Study on Software Process  
Capability and Maturity Level, and  
Development of the Appraisal System)

홍정유\* 이수정\*\* 경원현\*\*\* 고석하\*\*\*\*

**요약** 1980년대 초기 미 국방성에서는 소프트웨어 개발업체의 소프트웨어 개발능력에 대한 판단 확으로 금전적, 시간적, 성능상의 많은 어려움을 겪어왔다. 따라서 미 국방성은 카네기 멜론 대학의 SEI(Software Engineering Institute)를 통하여 소프트웨어 개발능력을 측정하고, 조직의 소프트웨어 프로세스 성숙수준을 결정하기 위한 CMM(Capability Maturity Model)을 개발하였다. 그러나 CMM에 기반을 둔 소프트웨어 프로세스 평가 방법은 외부 전문가를 이용하여 평가하므로, 시간과 비용이 많이 소모되는 방법이다. 따라서 많은 기업에서 이러한 평가를 수행하는데 어려움을 느끼고 있다. 특히, 중·소규모의 조직에서 사용하기에는, CMM의 평가 방법은 너무 복잡하고 추상적이다. 소규모 기업의 경우에 3년여에 걸친 프로세스 향상노력으로 많은 성과가 이루어졌음에도 불구하고 CMM의 평가방법을 이용한 결과는 1 단계 수준을 벗어 나지 못하였다는 사례들이 많이 발표되고 있다. 이러한 결과들은 CMM의 평가 방법은 중·소 규모 조직에는 적절치 않다는 것을 입증한다. 따라서 중·소 규모 기업이나 조직의 소프트웨어 프로세스의 성숙도 평가와 개선을 위한 모델들을 개발하기 위한 노력이 전 세계적으로 광범위하게 펼쳐지고 있다. 본 연구에서는 최근의 모델인 CMMI(Capability Maturity Model Integration)의 소프트웨어 프로세스 영역 분류에 의거해서, 우리나라의 소프트웨어의 개발과 취득에 관련된 조직들이 각 프로세스 영역을 얼마나 중요하게 평가하는지, 그리고 해당 영역에서의 기준, 절차 그리고 방법을 문서화 시키는 것을 얼마나 중요하게 평가하는지를 조사하였다. 한국의 소프트웨어 개발 관행은 CMM이나 ISO 15504 등에서 바람직한 것으로 제시하고 있는 국제 표준과는 많이 다르며, 나아가서는, 예를 들어, 미국의 일반적인 관행과도 많이 다를 수 있다. 따라서 미국이나 유럽의 대규모 소프트웨어 조직을 위해서 개발된 CMM이나 ISO 15504 등은 우리나라의 중·소 규모 소프트웨어 조직에는 특히 부적절할 수 있다. 본 연구에서는 우리나라 소프트웨어 조직의 종류, 조직의 규모, 그리고 프로젝트 수행 규모에 따라서 각 소프트웨어 프로세스 영역의 중요도가 어떻게 다른지, 그리고 해당 소프트웨어 프로세스 영역의 기준, 절차 그리고 방법을 문서화 시킬 필요성이 어떻게 변하는가를 조사하였다. 본 연구 결과는 우리나라의 소프트웨어 조직들이 프로세스 개선을 위해서 무엇을 정말로 필요로 하는지를 밝힘으로써, 한국 소프트웨어 산업의 현실적인 특수성을 고려한 소프트웨어 프로세스 평가와 개선 모델의 개발을 위한 기초적인 자료를 제공할 것으로 예상된다. 또한, 본 연구 결과는, 우리나라 소프트웨어 조직들이 실제로 무엇을 필요로 하는지를 밝힘으로써, 우리나라의 소프트웨어 산업을 육성하기 위한 실효성 있는 정책 입안을 위한 기초 자료를 제공할 것으로 예상된다.

---

\* 충북대학교 경영정보학과 박사과정  
\*\* 충북대학교 경영정보학과 석사과정  
\*\*\* 충북대학교 경영정보학과 박사과정  
\*\*\*\* 충북대학교 경영정보학과 교수

## 1. 서론

### 1. SPI(Software Process Improvement) 개요와 그 의의

소프트웨어 산업계는 지난 20여 년간 새로운 소프트웨어 개발 방법론과 기술을 적용해왔다. 그럼에도 불구하고 소프트웨어 산업계는 만족할 만한 수준의 생산성 향상과 고품질의 소프트웨어를 얻지 못하고 있다. 그 원인은 소프트웨어 프로세스의 관리가 체계적으로 이루어지지 않고 있기 때문이라고 판단되고 있다. 따라서 소프트웨어 프로세스의 체계적인 관리를 위해 소프트웨어 프로세스 향상에 대한 여러 가지 방법론 및 평가방법과 표준들이 최근까지 제기되고 있다.<sup>1),2)</sup>

프로세스란 하나의 과업을 이룩하는 일련의 단계이다. Software Process 는 프로젝트 계획, 설계서류, 코드, 경우테스트, 사용자 매뉴얼 등의 소프트웨어나 관련제품의 개발과 유지에 이용되는 활동, 방법, 실행, 변환 등을 말한다. 이러한 소프트웨어 프로세스는 매우 중요하게 여겨지고 있다. 소프트웨어 프로세스에서 어떤 문제가 발생할 경우, 발생한 문제만을 해결한다면, 이러한 문제는 추후 다시 발생할 수 있다. 하지만 이러한 문제가 발생하는 프로세스를 해결한다면, 이러한 문제는 영원히 해결 할 수 있기 때문이다. 이러한 소프트웨어 프로세스 향상은 생산성을 향상시키고, cycle time을 줄여주며, 무엇보다도 소프트웨어의 품질을 향상시켜준다.<sup>3),4),5),6),7)</sup>

품질과 생산성에 관한 연구는 산업화 시대 초기에서부터 시작되었다. EQA(European Quality Assurance), Malcolm Baldrige National Quality Award 의 Baldrige, TQC, QMS 같은 여러 형태의 접근방법이 품질 향상을 위해서 추진되어 왔다. 이러한 기업이나 조직의 품질향상을 위한 품질관리 노력은 시스템적 접근 방법으로 발전하여 국제적인 품질 시스템의 기준이 제정되었다. ISO에서 제정한

ISO9000과 같은 품질경영을 위한 일반적인 표준이 대표적인 예이다.

그러나 소프트웨어는 이러한 제조업과는 다른 개발프로세스, 관리 프로세스, 제품 품질 등과 같은 특성을 지니고 있다. 따라서 제조업의 품질관리나 생산성 향상을 위한 기존의 일반적인 품질 시스템과 달리, 소프트웨어만의 품질 향상을 위한 접근 방법이 필요하게 되었다.

1980년대 초기 미 국방성에서는 소프트웨어 개발업체의 소프트웨어 개발능력에 대한 판단착오로 금전적, 시간적, 성능상의 많은 어려움을 겪어왔다. 따라서 미 국방성은 카네기 멜론 대학의 SEI(Software Engineering Institute)를 통하여 소프트웨어 개발능력을 측정하고, 조직의 소프트웨어 프로세스 성숙수준을 결정하기 위하여 SCE(Software Capability Evaluation)를 요청하였다. 그 후, 1986년 IBM에 근무하던 Humphery가 성숙도 프레임워크를 SEI로 가져와서 성숙도 수준 개념을 추가하였다. 마침내, 1990년 SEI는 SW-CMM (Software Capability Maturity Model) 초기 Draft를 시작으로, 1991년 8월 SW-CMM Version 1.0과 1993년 2월 SW-CMM Version 1.1을 개발하였다.<sup>8)</sup>

SW-CMM으로부터 파생된 모델에는 PSP, People-CMM, SA-CMM<sup>9)</sup>, SE-CMM<sup>10)</sup>, SCE 등이 있다. 그리고 FAA는 이 모델들 중에서 SW-CMM, SA-CMM, SE-CMM을 통합한 FAA-iCMM을 개발하였으며, 최근 SEI에서는 기존의 CMM 모델을 통합한 CMMI를 개발하였다. 그 밖에도, BOOTSTRAP, TRILLIUM 등과 같은 Software Process 향상을 위한 모델들이 제시되고, STD, TickIT, STARTS 등과 같은 소프트웨어 프로세스 향상 방법 등이 제시되었다. 또한 기존의 ISO9000시리즈에 소프트웨어 특성을 고려한 ISO9000-3, SPICE 같은 표준도 만들어지게 되었다.

1987년 SW-CMM에 의거한 SPA (Software Process Assessment)가 시행된 이후로 2001년 12월까지 약 14년 간 2164건의 평가가 이루어 졌으며, 45개국 1638개의 조직과 456개

의 회사가 참여를 했으며, 8925개의 프로젝트가 평가되었다. 또한 427개의 회사는 재평가를 받았으며, 재평가를 받는 조직의 수도 계속적으로 증가하고 있다. 1987부터 2001년말 까지, 처음 7년에 비해 최근 7년간의 평가가 급증하였고, 재평가를 수행하는 조직의 수도 점차 증가하였다. 또한, 점차적으로 CMM이 각국으로 확산되고 있다. 조직의 형태별 평가 건수는, 처음에는 미국의 국방부를 위주로 이루어졌지만 1993년부터는 일반 상업적 조직의 평가 건수가 국방부를 추월하여 현재는 국방부의 약 3배 정도까지 증가하였다<sup>11),12),13),14),15),16),17),18),19),20),21),22),23),24),25),26)</sup>

2002년 3월 27일 SEI에서 발표한 성숙 레벨을 획득한 소프트웨어 업체 리스트<sup>27)</sup>에 의하면, 전 세계적으로 레벨 2를 획득한 업체는 52개사(18.24%), 레벨 3을 획득한 업체는 105개사(36.84%), 레벨 4를 획득한 업체는 51(17.90%)개사, 최고수준인 레벨 5를 획득한 업체는 77개사(27.02%)로, 모두 285개사이다. 또한, 우리나라에서도 지난해 포스데이터와 삼성 SDS 첨단소프트웨어센터가 레벨 3을 획득하였다. 그러나 미국 공공기관들은 소프트웨어 개발 입찰에 참여하는 모든 업체에 레벨 3이상을 요구하고 있으므로, 우리나라뿐만 아니라 전 세계적으로 더 많은 소프트웨어 업체들이 상위 CMM 레벨을 획득하기 위해 노력할 것으로 기대되고 있다.

위와 같이 소프트웨어 프로세스 향상을 위한 CMM 모델은 SPI 모델 발전에 핵심적인 역할을 하고 있다. 또한 위와 같이 소프트웨어 프로세스 품질 향상을 위한 CMM은 SPI 모델 발전에 핵심적인 역할을 하고 있다. 또한 CMM으로부터 UMM(Usability Maturity Model), TMM(Test Maturity Model), PMM(Product Maturity Model), RMM(Reuse Maturity Model)과 같은 특수한 소프트웨어 분야를 위한 여러 가지 파생 모델들이 개발되었으며, CMM이 다른 모델에 비해 매우 활발하게 계속적으로 진화하고 있음을 알 수 있다.

국내에서도 소프트웨어 생산성 및 품질을 향상시키려는 목적으로 사단법인 한·카네기멜론대학 기술교류협회(Korea·CMU Technology Transfer Institute : KCTTI) 설립인가가 정보통신부로부터 공식적으로 이루어졌고, 2001년에는 2회에 걸쳐 학술대회가 개최된 바 있다. 이와는 별도로 2000년 8월 19일 SPICE(ISO/IEC TR 15504)와 관련하여, 국제 소프트웨어 프로세서 심사인 협회 소속 우리나라 회원들이 "한국 소프트웨어 프로세서 심사인 협회" 창립총회를 열고 본격적인 활동에 들어갔다.

## II. 문헌연구

### 1. CMM

#### (1) CMM의 개요

1980년대 초기 미 국방성에서는 소프트웨어를 개발하는 업체의 개발능력에 대한 판단 잘못으로 금전적, 시간적, 성능상의 많은 어려움을 겪어왔다. 따라서 미 국방성은 카네기 멜론 대학의 SEI(Software Engineering Institute)를 통하여 Software 개발능력을 측정하고, 조직의 소프트웨어 프로세스 성숙수준을 결정하기 위하여 SCE(Software Capability Evaluation)를 요청하였고, SEI는 IBM에 근무하던 Humphery가 성숙도 프레임워크를 1986년 SEI로 가져와서 성숙도 수준 개념을 추가하여, 1990년 SW-CMM(Software Capability Maturity Model) 초기 Draft를 시작으로 1991년 8월 Version 1.0과 1993년 2월 SW-CMM Version 1.1을 개발하였다.<sup>28)</sup>

일련의 정부, 산업계, SEI 대표자들은 정부 소프트웨어 취득 프로세스를 용이하게 하기 위해 Software Acquisition CMM(SA-CMM)을 만들었으며, 2000년 8월 SW-CMM, SE-CMM을 통합한 CMMI Ver. 1.0을<sup>29),30),31),32)</sup> 시작으로

「표-1」 CMM의 성숙 수준

성숙수준 단계	명칭	내용	프로세스 능력의 유형
5 (상위)	최적화 (Optimizing)	지속적인 프로세스 개선이 프로세스와 혁신적인 아이디어 및 기술로부터의 피드백을 통하여 이루어진다.	지속적인 프로세스 개선
4	관리 (Managed)	소프트웨어 프로세스와 제품 품질에 대한 상세한 측정값을 수집한다. 소프트웨어 프로세스와 제품이 양적으로 이해되고 관리된다.	예측 가능한 프로세스
3	정의 (Defined)	관리 및 공학적인 개발에 필요한 소프트웨어 프로세스가 문서화되고 표준화되어 조직에서 공식적으로 사용하는 표준 소프트웨어 프로세스로 정립되어 있다.	표준화된 프로세스
2	반복 (Repeatable)	비용, 일정, 기능을 추적하기 위한 기본적인 프로젝트 관리가 이루어지고 있다. 필요한 프로세스에 대한 교육은 없고 새로운 프로젝트를 할 때는 이전에 성공한 프로젝트에서 사례를 찾아서 활용한다.	훈련된 프로세스
1 (하위)	초기 (Initial)	소프트웨어 프로세스는 주먹구구식이고 매우 혼란스러운 상태이다. 일부 프로세스가 정의되어 있고 성공은 특정한 개인의 노력에 의존한다.	

2002년 3월에는 SW-CMM, SE-CMM, SA-CMM 등을 통합한 CMMI Ver. 1.1이 발표되어 있다.<sup>33), 34), 35), 36), 37), 38)</sup>

2000년 12월에 발표한, CMMI는 3개의 버전을 가지고 있는데, 단지 소프트웨어와 시스템 공학(SE/SW)을 위한 것과, 소프트웨어와 시스템 공학(SE/SW)과 프로세스개발(SE/ SW/ IPPD)을 포함하는 것과, 취득 측면(SE/SW/IPPD/A)까지를 포함하는 버전으로 구분할 수 있다. 각 버전은 단계적 또는 연속적 표현들을 제공한다.

System Engineering부분을 보면, SW-CMM을 기반으로 Capability Maturity Model (SE-CMM)과 SE-CMM Appraisal Method (SAM)을 개발했으며, INCOSE (International Council on Systems Engineering)는 SECAM (Systems EngineeringCapability Assessment Model)을 개발하였다.

CMM 모델들의 성숙수준은, 「표-1」에서 볼 수 있듯이, 기본적으로 5단계로 구성되어 있다.

각 성숙 수준들은 잘 정의된 점진적 단계들로서, 초기 수준에서 최적화 수준으로 갈수록 조직의 소프트웨어 능력 또는 소프트웨어 프로세스가 더 성숙했다는 것을 의미한다.

CMM에서 성숙 수준은 조직 전체의 성숙 수준과 프로세스 영역의 성숙 수준으로 구분한다. 프로세스 영역은 조직에서 소프트웨어 프로세스를 개선하기 위해 집중해야 하는 영역들을 말한다. 기초 프랙티스들이 각 프로세스 영역의 목표를 달성하기 위해서 수행된다. 각 프로세스 영역들은 서로 독립적으로 그 성속도가 평가된다. 조직 전체의 성속도는 각 프로세스 영역들의 성속도에 따라서 결정된다. 일정한 프로세스 영역들의 성속도가 같은 수준에 도달하면, 조직 전체의 성속도는 해당 수준으로 결정된다. 이를 위해 각 프로세스 영역은 네 개의 성숙수준에 해당하는 집단으로 분류된다. 연속적 평가에서는, 특정 수준과 그 하위 수준에 해당하는 집단의 프로세스 영역의 성속도가 모두 해당 수준

에 이르면 조직 전체의 성숙도도 해당 수준으로 정의된다. 예를 들어, 정의 수준과 반복 수준에 해당하는 프로세스 영역들이 모두 정의 수준에 이르면, 조직 전체의 성숙도도 정의수준으로 평가된다.

## (2). CMM의 평가 체계 (Appraisal Framework)

CMM의 평가는 다음의 2가지 형태로 구성되어 있다. 즉, 평가소프트웨어 프로세스 평가(software process assessment)와 소프트웨어 능력 평가(software capability evaluation) 이다.<sup>39)</sup>

소프트웨어 프로세스 평가는 조직의 현재의 소프트웨어 프로세스의 상태를 결정하고, 소프트웨어 프로세스 관련된 현안 문제를 결정하고, 소프트웨어 프로세스 개선을 위해 필요한 조직적 지원을 획득하기 위해 이루어진다.

소프트웨어 프로세스 평가는 자기 조직의 소프트웨어 프로세스 내에서 우선적으로 개선할 부분을 확인하는 것에 포커스를 맞추고, 평가팀은 CMM을 발견한 것을 확인하고, 우선순위를 결정하는 가이드로 사용한다. CMM에서 key practices에 의해 제공되는 지침에 따라 이러한 발견된 것들은 조직 개선 전략을 계획

하기 위해 사용된다.

반면에, 소프트웨어 능력 평가는 소프트웨어 계약자가 소프트웨어 프로젝트를 수행할 능력과 자격을 갖추고 있는지를 확인하기 위해서 또는 소프트웨어 프로세스의 상태를 감시하기 위해 사용된다.

소프트웨어 능력 평가는 정해진 계획과 주어진 예산 범위 내에서 고품질의 소프트웨어를 개발하기 위해 특정한 프로젝트 또는 계약과 관련된 위험을 확인하는 것에 포커스를 맞추고 있다. 따라서 소프트웨어 획득 프로세스 동안, 소프트웨어 능력 평가는 입찰자들에게 수행될 수 있으며, CMM에 의해 구성된 평가의 결과는 특정한 계약자를 선택할 때 위험을 확인하기 위해 사용될 수 있다.

평가는 또한 기존의 계약에 관해 계약자의 소프트웨어 프로세스의 잠재적인 개선을 확인할 의도로 그들의 프로세스 실행을 감시하기 위해 수행될 수도 있다.

소프트웨어 프로세스 평가(software process assessment)와 소프트웨어 능력 평가(software capability evaluation)의 주요 차이점을 살펴보면 다음과 같다.<sup>40)</sup>

소프트웨어 프로세스 평가와 소프트웨어 능력 평가를 수행하기 위해 CMM은 공통된 참조의 프레임(common frame of reference)을 수립

Assessments	Evaluations
Development organization uses to improve processes	Acquirers use in selecting and monitoring suppliers; developers use to measure improvement progress
Results are used within the assessed organization	Results are known to sponsor
Assesses process practice	Substantiates current practice
Acts as improvement catalyst	Evaluates commitment to improve
Provides input to action plan	Analyzes performance potential and improvement input
Collaborative—members of organization must be on team	Members of organization may or may not be on team
Applies to organization, not individual projects or contracts	Organizational data applies to specific sponsor needs
Input for improvement action plan to unfreeze organization	Input for award decision, performance monitoring, risk management, and internal improvement measurement

그림 1 소프트웨어 프로세스 평가(software process assessment)와 소프트웨어 능력 평가(software capability evaluation)의 차이

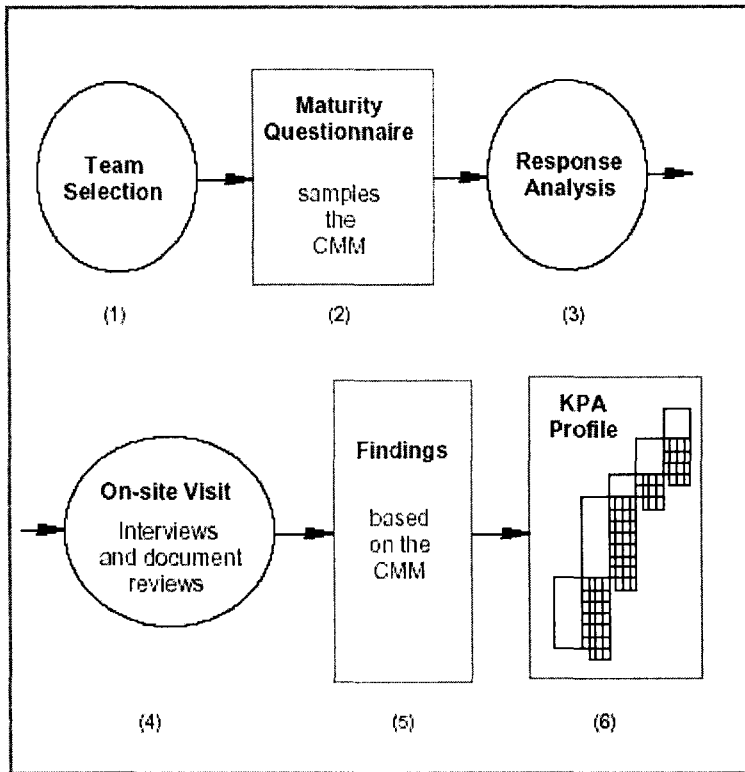


그림 2 소프트웨어 프로세스 평가와 소프트웨어 능력 평가의 공통 스텝

한다. 비록 두 가지 방법이 소프트웨어 프로세스 성숙을 평가하기 위한 목적이 틀리지만, 평가 방법은 CMM을 근거로 사용한다.

다음 그림은 그림 소프트웨어 프로세스 평가와 소프트웨어 능력 평가의 공통 스텝의 요약된 기술을 보여주고 있다.

첫 번째 단계는 평가 팀을 선택하는 것이다. 평가 팀은 소프트웨어 프로세스 평가와 소프트웨어 능력 평가 방법의 특성뿐만 아니라 CMM의 기본적인 개념에 대한 훈련을 이수해야 하며, 팀의 멤버는 소프트웨어 공학과 관리에 지식이 있는 전문가로 구성되어야 한다.

두 번째 단계는 성숙 설문 조사와 다른 진단의 도구를 이용하여 소프트웨어 프로세스와 소프트웨어 능력을 평가 받게 될 사이트로부터 표본을 수집하는 것이다.

세 번째 단계는 표본수집 활동이 완료되면, 소프트웨어 프로세스 또는 소프트웨어 능력 평가 팀은 수집한 응답 분석을 수행하며, CMM의 주요 프로세스 영역과 일치하는 정도를 조사하게 된다.

네 번째 단계는 소프트웨어 프로세스 또는 소프트웨어 능력 평가를 수행할 사이트를 방문하는 것이다. 응답 분석의 결과로 시작하여, 평가 팀은 사이트에서 수행한 소프트웨어 프로세스의 이해를 얻기 위해 인터뷰를 실시하고, 문서를 검토하게 된다.

평가 팀 멤버는 인터뷰와 문서로부터 얻은 정보에 대해 질문, 청취, 재검토, 종합분석 등을 통하여 CMM의 주요 프로세스 영역과 주요 실행 영역(key process areas and key practices)에 대해 판단하게 된다.

주요 프로세스 영역과 평가 사이트의 실행 내

Scale	CMMI			
	Engineering Process Areas	Rating	Other Process Areas	Rating
Level 0		Default Rating		Default Rating
Level 1	Generic goal for capability level 1	Satisfied	Generic goal for capability level 1	Satisfied
	All specific goals including base practices only	Satisfied	All specific goals	Satisfied
Level 2	Generic goals for capability levels 1 and 2	Satisfied	Generic goals for capability level 1 and 2	Satisfied
	All specific goals including specific practices at capability levels 1 and 2	Satisfied	All specific goals	Satisfied
Level 3	Generic goals for capability levels 1, 2 and 3	Satisfied	Generic goals for capability levels 1, 2 and 3	Satisfied
	All specific goals including specific practices at capability levels 1, 2 and 3	Satisfied	All specific goals	Satisfied
Level 4	Generic goals for capability levels 1, 2, 3 and 4	Satisfied	Generic goals for capability levels 1, 2, 3 and 4	Satisfied
	All specific goals including specific practices at capability levels 1, 2 and 3	Satisfied	All specific goals	Satisfied
Level 5	Generic goals for capability levels 1, 2, 3, 4 and 5	Satisfied	Generic goals for capability levels 1, 2, 3, 4 and 5	Satisfied
	All specific goals including specific practices at capability levels 1, 2 and 3	Satisfied	All specific goals	Satisfied

그림 3 Capability Level 측정을 위한 각 Process 평가 방법

용에 명백한 차이가 있을 때 평가 팀은 평가 사이트의 실행 내용이 주요 프로세스 영역의 목적을 만족시키는지에 대해 전문가적인 판단을 수행한다. 이 경우 평가 팀은 주요 프로세스 영역에 대한 판단을 위해 상세하게 기록해야만 한다.

다섯 번째는 현장에서의 기간의 마지막에는, 평가 팀은 조직의 소프트웨어 프로세스의 강점과 약점에 대한 목록을 생산하게 된다. 소프트웨어 프로세스 평가를 통해 수집된 목록은 소프트웨

어 프로세스 개선을 위한 기반이 되며, 소프트웨어 능력 평가를 통해 수집된 목록은 소프트웨어 획득 대행기관에 의해 수행되는 위험 분석의 일부가 된다.

마지막으로, 평가 팀은 조직이 주요 프로세스 영역의 목표를 만족시키는지와 가지고 있는 프로세스 영역과 가지고 있지 못하는 프로세스 영역을 알려주는 프로필을 준비한다.

CMM의 수준을 평가하기 위해서 각 Level 별 프로세스 영역의 rating 기준은 다음과 같다.

## 2. ISO/IEC TR 15504(SPICE)

ISO9001-2000에서 프로세스 어프로치를 도입하기 5년 전인 1995년 국제 표준화 기구와 국제 전기 표준 회의는 공동으로 ISO/ IEC 12207 SLCP(정보 기술 소프트웨어 라이프사이클 프로세스를 위한 기준) 를 발표했다. 그 뒤에, IEEE-EIA 공동 작업 그룹은 ISO/ IEC 표준에 부록을 추가하여 IEEE/ EIA 12207을 만들었으며,

Software Process Improvement Capability dEtermination(SPICE)라고 이름 지어진 ISO/ IEC TR 15504 SPA<sup>41)</sup>는 ISO/IEC JTC 1/SC 7에서 작성한 기술 보고서로서, ISO/IEC JTC 1/SC 7는 국제 표준화 기구와 국제 전기 표준 회의의 합동 기술위원회로 그 가운데 소프트웨어 엔지니어링을 담당하고 있는 것이 S7이다. 공식적인 기술 보고서에는 조직의 성숙과 능력을 평가하기 위한 방법이라고 기술되어있으며, ISO/ IEC TR 15504는 그들이 어떤 표준들에 따르든 간에 CMM같은 모델들을 대체할 수 있는 준거모델들을 포함하고 있다.

ISO/IEC TR 15504 는 다음과 같은 9개의 부분으로 되어 있다.<sup>42),43),44),45),46),47),48),49),50)</sup>,

- Part 1 개념과 개요 지침
- Part 2 프로세스와 프로세스 능력에 대한 기준 모델
- Part 3 평가 실행
- Part 4 평가 실행에 대한 지침
- Part 5 평가 모델과 지도 안내
- Part 6 평가자의 자격에 대한 지침
- Part 7 프로세스 개선에 사용 지침
- Part 8 공급자 프로세스 능력을 결정하는데 사용 지침
- Part 9 용어

평가 모델과 방법의 요구사항은 ISO/IEC TR 15504의 Part 2와 3에 설명되어 있고, 만약 규정 요구사항이 충족되면, 그 평가는 15504 규격에 부합한다고 간주된다.

기준 모델은 다음의 두개의 차원을 규정하고 있다.

- ① 프로세스 차원 - 평가되는 프로세스에 따라 독특하며, 프로세스의 측정가능 목표의 달성과 관련되어 있다.
- ② 능력 차원 - 모든 프로세스에 적용 가능한 일련의 프로세스의 속성임. 그 속성은 프로세스의 능력을 관리하고 향상시키기 위해서 요구하는 측정가능 특성이다.

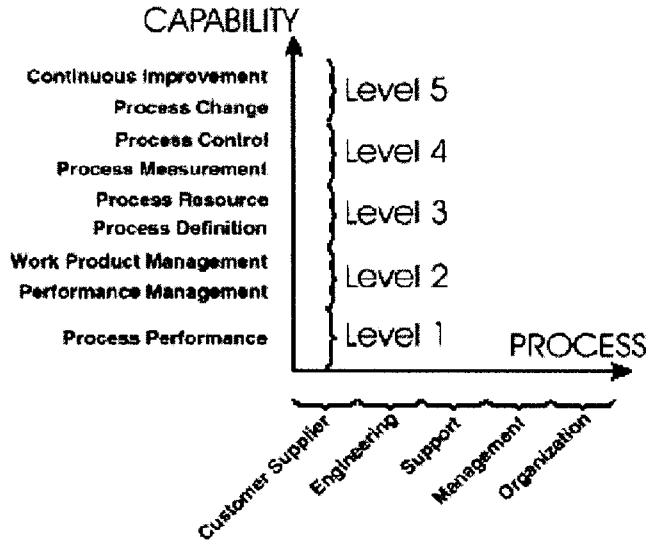
2차원적인 구성은 아래와 같으며, 최소한 평가 모델은 규정된 요소들을 포함하고 있어야 한다. 아래에 보여준 것 같이 프로세스는 다섯 개의 범주로 그룹 되어 있다. 측정된 프로세스 능력 달성의 다섯 개의 단계로 나누어진 아홉 개의 프로세스 속성은 수직 축에 주어졌다.

능력 수준은 0부터 5까지 6단계로 나누어서 개발자 스스로가 자기의 수준을 인지할 수 있는 프로세스 속성을 제시하고 있다. 적용 범위도 소프트웨어 개발에 관한 계획, 관리, 실행 및 제어를 통해서 개선과 능력 평가를 수행하고, 그 대상 업무를 소프트웨어 개발뿐만 아니라 개발 요청, 공급, 운영, 유지보수 및 지원 영역까지 포함 할 수 있도록 제정하고 있다. SPICE는 소프트웨어의 프로세스를 심사하여 소프트웨어를 개발하는 프로세스를 개선하고, 개발하는 개발자의 능력을 결정하므로써 개발 리스크를 통제할 수 있도록 하여준다.

Level 2 ~ 5 의 프로세스 속성(PA)에는 보통 Support, Management, Organization process category의 프로세스들이 관련되어 있으며. Level 2는 management & support category 의 일부 프로세스들과 관련이 있고 Level 3, 5 는 organization process category의 일부 프로세스들과 관련이 있다.

프로세스 속성(PA)의 rating scale을 보면 PA 는 프로세스의 측정 가능한 특성(characteristics)을 나타내고 Rating scales은 PA의 성취도를 백분율(0-100%)로 나타내며,





성취도의 백분율(0-100%))을 ordinal로 다음과 같이 이용한다.

- (1) 미달성 (Not achieved: N) : 0% - 15%
  - 해당 프로세스 PA 달성에 관한 증거가 거의 없을 경우에 해당되며,
- (2) 부분 달성 (Partially achieved: P) : 16% - 50%
  - 해당 프로세스 PA에 대한 양호한 체계적 접근방법과 달성 증거가 있거나
  - 일부 측면에서의 달성은 예측 불가능할 수도 있을 경우
- (3) 대부분 달성 (Largely achieved: L) : 51% - 85%
  - 해당 프로세스 PA에 대한 양호한 체계적 접근방법과 뚜렷한 달성(significant achievement) 증거가 있을 경우
  - 프로세스 수행은 분야(areas) 및 작업 단위(work unit)에 따라 다를 수 있을 경우
- (4) 완전 달성 (Fully achieved: F) : 86% - 100%
  - 해당 프로세스 PA에 대한 완전하고 체계적인 접근방법과 완전한 성취 증거(full achievement) 가 있을 경우
  - 조직 단위(Org. Unit) 전체에 걸쳐 뚜렷한 약점이 없을 경우 위와 같이 분류한다.

프로세스의 능력 수준은 PA Rating에 근거하여 결정되며 각 프로세스의 Rating을 결정하는 기준은 다음 표와 같다.

SPICE의 프로세스 심사(평가) 형태는 자체 심사(self-assessment)로 조직 자신의 프로세스 수행능력을 심사하는 경우로서 스폰서는 조직 내부에 있는 경우가 있고, 독립 심사(independent assessment)로 조직 단위와 제 3의 공인 심사 기관의 독립적인 심사원에 의해 수행되는 형태로 스폰서는 조직 외부에 있을 수도 있다.

SPICE의 프로세스 심사(평가) Framework은 a) 자체 심사(self assessment)를 장려하고, b) 심사 대상 프로세스의 context를 고려하고, c) 프로세스 ratings(프로세스 프로파일)을 생성하며, d) 프로세스 속성을 사용하여 프로세스 능력을 표현하며 e) 모든 application domains이나 조직 규모에 사용할 수 있다.

심사 프로세스는 개발되어 문서화되어야하고 최소한 다음과 같은 활동을 포함하여 이루어진다.

- (가) 계획(planning) : 심사 계획서에는 최소한 다음이 포함되어야 함
  - 심사 입력
  - 심사 시 수행될 활동

Scale	SPICE	
	Process Attributes	Rating
Level 0		
Level 1	PA1.1 Process Performance	Largely or fully
Level 2	PA1.1 Process Performance	Fully
	PA2.1 Performance Management	Largely or fully
	PA2.2 Work Product Management	Largely or fully
Level 3	PA1.1 Process Performance	Fully
	PA2.1 Performance Management	Fully
	PA2.2 Work Product Management	Fully
	PA3.1 Process Definition and Tailoring	Largely or fully
	PA3.2 Process Resource	Largely or fully
Level 4	PA1.1 Process Performance	Fully
	PA2.1 Performance Management	Fully
	PA2.2 Work Product Management	Fully
	PA3.1 Process Definition and Tailoring	Fully
	PA3.2 Process Resource	Fully
	PA4.1 Process Measurement	Largely or fully
	PA4.2 Process Control	Largely or fully
Level 5	PA1.1 Process Performance	Fully
	PA2.1 Performance Management	Fully
	PA2.2 Work Product Management	Fully
	PA3.1 Process Definition and Tailoring	Fully
	PA3.2 Process Resource	Fully
	PA4.1 Process Measurement	Fully
	PA4.2 Process Control	Fully
	PA5.1 Process Change	Largely or fully
	PA5.2 Continuous Improvement	Largely or fully

그림 5 프로세스의 Rating 결정 기준

- 자원 및 일정
- 심사원 선정 및 책임 할당
- Part 3의 요구사항 수행에 대한 검증 기준
- 심사 결과물(output)에 대한 설명

(내) 데이터 수집 : 심사 범위 내의 프로세스 평가에 필요한 데이터는 체계적으로 수집되어야 하며 최소한 다음 사항을 포함하여야 한다.

- 데이터의 선정/수집/분석과 ratings의 justification을 위한 전략 및 기법은 명시적으

로 식별되고 보여줄 수 있어야 함

- 심사대상 프로세스와 Part 2의 프로세스와 관계를 보여주어야 함
- 각 프로세스는 객관적 증거를 바탕으로 심사되어야 함

- 각 프로세스의 PA 을 위해 수집된 객관적 증거는 심사 목적과 범위에 맞게 충분하여야 함

- ratings의 검증을 위한 기초로 제공하기 위해 객관적 증거(지표에 기반한)는 기록되고 유

모델의 구성

Process Category Dimension					
구분	CMMI 구조		SPICE		
	Staged Representation	Continuous Representation	구분	ISO/IEC TR 15504(SPICE) 구조	
			(1)Process Group	Primary Life Cycle Process Group Supporting Life Cycle Process Group Organizational Life Cycle Process Group	
(1)Process Category	Process Management Process Category	좌동	Organizational Life Cycle Process Group	Management Process Category	
	Project Management Process Category	좌동		Organization Process Category	
	Engineering Process Category	좌동	Primary Life Cycle Process Group	Customer - Supplier Process Category Engineering Process Category	
	Support Process Category	좌동	Supporting Life Cycle Process Group	Support Process Category	
(2)Process Area			(2)Process	Top-Level Process Type	basic(identical in intent to the processes in ISO/IEC 12207) new(expansions of ISO/IEC 12207) extended(outside the scope of ISO/IEC 12207)
				Second-Level Process Type	component(a group of one or more ISO/IEC 12207's activities from the same extended component(one or more of ISO/IEC 12207's activities from the same process, with additional
(3)Goal	Specific goals Generic goals	좌동 좌동	(3)Process Purpose		
Common feature (Generic Goal)	Commitment to Perform (CO)				
	Ability to Perform (AB)				
	Directing Implementation (DI)				
	Verifying Implementation (VE)				
(4)Practice	Specific Practices	Base Practice(capability level of 1)	(4)Base Practice		
		Advanced Practice(capability level of 2 or higher)			
	Generic Practices	좌동			
Generic/Specific Practices	Typical Work Product	좌동	work product & WP characteristics	input	
		좌동		output	
	SubPractice	좌동			
Process Maturity/Capability Dimension					
(1)Maturity Level		0 Incomplete	(1)Capability Level	0 Incomplete Process	
	1. Initial	1 Performed		1 Performed Process	
	2. Managed	좌동		2 Managed Process	
	3. Defined	좌동		3 Established Process	
	4. Quantitatively Managed	좌동		4 Predictable Process	
	5. Optimizing	좌동	5 Optimizing Process		
(2)Process Area			(2)Process Attribute		
(3)Goal	Specific goals Generic goals	좌동 좌동	(3)Process Purpose		
Common feature (Generic Goal)	Commitment to Perform (CO)				
	Ability to Perform (AB)				
	Directing Implementation (DI)				
	Verifying Implementation (VE)				

지되어야 한다.

(㉔) 데이터 확인: 수집된 데이터는 확인되어야

함. 확인된 데이터가 심사 범위를 충분히 커버

함을 보장할 수 있는 조치를 취해야 한다.

(㉠) rating : 각 프로세스 속성(PA)은 확인된 데이터에 근거하여 rating되어야 한다.

- rating의 결과는 프로세스 프로파일로서 기록됨

- rating을 수행하는 심사원의 판단을 돕고 심사에서 반복성을 유지하기 위해 호환 모형에서 정의된 심사 지표를 사용함

- rating을 위한 의사결정 프로세스(즉 심사팀의 consensus나 다수결 투표)는 기록되어야 함

(㉡) 보고 : 다음과 같은 최소한의 내용을 포함한 심사결과는 문서화하고 스폰서에게 제출함

- 심사 날짜
- 심사 입력
- 수집된 객관적 증거의 식별자
- 사용된 심사 접근 방법(self-assessment or independent assessment)
- 심사 결과인 프로세스 프로파일
- 심사 중 수집된 추가 정보의 식별

### 3. 소규모 조직을 위한 변형 모델

Karl 과 Doris(2000)<sup>51)</sup>는 CMM에 기반을 둔 내부 Process Improvement와 Software Process Assessment는 시간과 비용을 많이 소모하는 방법으로서 많은 기업에서 이러한 평가를 자주 수행하는데 어려움을 느끼고 있다하며, 정기원(1997)<sup>52)</sup>의 경우 CMM 모델은 대규모 조직에 적합하다고 주장하고 있으며, Roman, Ivan, Jozsef(2000)<sup>53)</sup>의 경우 Key Process Areas 와 Activities 는 작은 기업에서 사용하기에는 너무 복잡하고 추상적이므로 단순화하여야 한다고 주장하고 있다.

또한, Paul Grunbacher(1997)<sup>54)</sup>는 CMM, BOOTSTRAP, ISO 15504(SPICE)등의 성숙개발프로세스(Mature Development Process) 방법은 모두 높은 비용이 수반되며, 외부전문가를 이용하고, 단계 조직에 적합한 복잡한 방법이며, 기존 방법은 중, 소규모 조직의 필요와 문제에 대해서는 소홀히 하고 있다고 주장하고, 자기평가 방법 등을 대안으로 제시하고 있다.

Otoya and Cerpa(1999)<sup>55)</sup>의 연구결과를 보면 소규모 기업에서 프로세스 평가 3년만에 걸친 프로세스 향상노력으로 많은 성과가 이루어졌음에도 불구하고 CMM의 단계 평가 결과는 1단계 수준을 벗어나지 못하고 있다는 결과를 볼 수 있다. 따라서 소규모 기업이나 조직 등에 적합한 변형된 모델들이 다양하게 제기되고 있다.

Hareton 등(2001)<sup>56)</sup>의 연구결과를 보면 월 3인 이하만을 투입하고, 6개월 이하의 프로젝트 기간을 가진 소규모 프로젝트가 80% 이상을 이루고 있으며, 소규모 프로젝트에는 문서관리에 40%이상의 오버헤드가 발생하고 있다고 한다.

Milissa등(2000)의 연구결과를 보면 소규모 프로젝트일수록 구성원간 인터랙션이 크며, 이에 따라 small project 는 coordination 의 필요성이 대규모 project에 비해 적다고 한다.

#### (1) Tailored Model

Otoya and Cerpa(1999)<sup>57)</sup>의 연구결과를 보면 CMM은 소프트웨어 개발 프로세스를 향상시키기 위한 framework 로서 많은 조직에서 사용하여 왔다. 그러나 소규모 조직은 CMM의 key practice의 상당부분은 소규모 조직의 프로젝트에 부적절하여 CMM을 적용시키는데 많은 어려움을 겪어왔다. 따라서 LOGOS international 은 CMM을 소규모기업, 조직, 프로젝트에 적합한 맞춤형버전을 개발하였다. 소규모 조직은 CMM의 단계에서 대부분 1단계에 머무르고 있으며, 혼란스러운 소프트웨어 프로세스를 가지고 있다고 한다.

Brodman and Johnson은(1996)<sup>58)</sup> 향상된 프로세스 성숙은 생산성을 증가시키고, 더 향상될 질적 내용을 제공하고, 더 정확한 예상시간을 지킨다는 것이다.

그러나 2단계 도달에도 대부분의 조직에서는 어려운 상황이므로 각 레벨의 도달을 위해서는 실제적인 실행 전략이 필요하다.

소규모 조직, 소규모 조직의 프로젝트는 CMM의 상의 단계에 도달하기가 상당히 어렵다. 왜

냐하면 많은 CMM 모델이 제시하는 key practice의 대부분은 소규모 조직, 소규모 조직의 프로젝트에 부적절하기 때문이다(Brodman and Johnson, 1994)<sup>59)</sup>

소규모 프로젝트 조직은 CMM에 의해서 요구하는 많은 양의 서류문서를 생산해야 하는 Overhead를 감당할 수 없기 때문이며, 또한 소규모 프로젝트 조직은 시간을 절약하기 위해 일반적으로 복잡한 서류를 이용해야 하기 때문이다.

소규모 프로젝트에서 팀은 일반적으로 횡적 조직구조를 가지게 되고, 자원의 부족에 의해서 일반적으로 개인은 여러 가지 역할을 부여받게 된다. 이러한 상황은 CMM practice에 의해서 요구하는 팀 구조와 지위(position)와 정면 배치되는 상황이며, 여러 practice의 이행을 더욱 어렵게 만든다는 것이다.

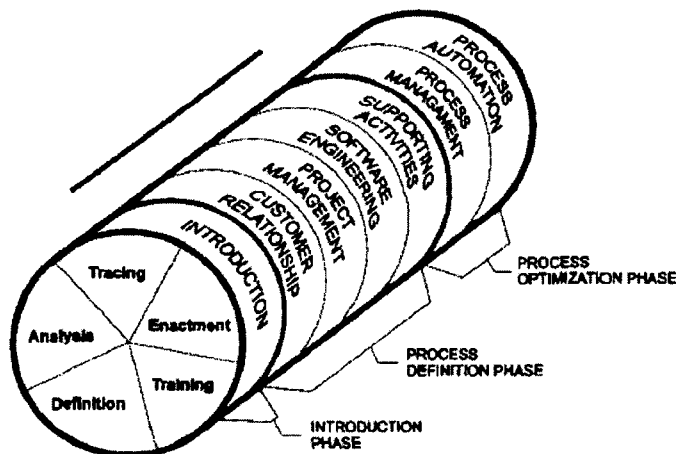
따라서 Otoy and Cerpa는 소규모 조직, 소규모 조직의 프로젝트는 CMM의 맞춤형 버전(tailored version)이 필요하다고 주장하고 있으며, LOGOS를 이용하여 호주의 Winapp 사를 대상으로 2단계의 6가지 key practice area 중 소프트웨어 프로젝트 계획 과 소프트웨어 프로젝트 추적 및 예측을 통합하고, 1개영역을 추가하여 (1) 요구사항 관리 (2) 소프트웨어 계획과 추적 (3) 소프트웨어 하청 계약자관리 (4) 소프트웨어 품질보증 (5) 소프트웨어 형상관리 (6)

훈련 등 6개영역으로 분류하였다.

Romana 등(2000)<sup>60)</sup> CMM, ISO, 기타 방법들은 대부분 중규모나, 대규모 조직에 적합한 방법이라 주장하고, 이러한 방법을 특정한 특성을 가진 소규모 조직에 이용하기 위해서는 소규모 기업의 특정한 특성에 맞추는 방법이 필요하다고 하며 PROCESSUS SPISE(Software Process Improvement for Small Company)모델을 제시하고 있다.

ISO, CMM등은 소규모 기업에게는 너무 복잡하므로 단순화가 필요한데, 이러한 단순화에 영향을 주는 주요한 이슈들은 다음과 같은 요소를 들 수 있다.

- (㉠) 작은 기업의 조직 구조는 간단하고, SPI 프로젝트 수행시 역할이 적기 때문에 SPI 프로젝트 수행 초기에 책임에 대해 분명히 해야 한다는 것이고,
- (㉡) 절차에 대한 설명과 정의 등을 위한 문서의 형태나, 필요한, 또는 도출된 문서의 형태 등도 SPI 프로젝트 수행 초기에 정의되어야 한다.
- (㉢) 각 활동(activities)들은 절차(procedure)에 포함된 영역을 고려해야하는 것처럼 절차 구현에 대한 교육은 기술적인 문제를 강조해야하며,
- (㉣) 프로젝트의 규모가 작고, 직원의 수가 적기 때문에 모든 직원이 SPI 목적에 전념하여 향상된 작업 절차에 대한 도입이 빠르다는 것이며, 그 결과 품질 보증과 같은 어떤 문제는 다른 절차의 시작에 이미 포함되어 진행될 수도 있다는 것이다.



## (2) Mini Assessment Model

Karl 과 Doris(2000)<sup>61)</sup>는 CMM에 기반을 둔 내부 프로세스 향상과 소프트웨어 프로세스 평가는 시간과 비용을 너무 소모하며, 많은 기업에서 이러한 평가를 자주 수행하기는 어렵다고 주장하며, Eastman Kodak 회사에서 개발한 MMA(modular mini-assesment) 방법을 소개하고 있다.

Eastman Kodak은 3가지 형태의 모듈식 방법을 개발하였는데 다음과 같다

첫째는 SPA의 2일간의 미니 버전이며 둘째는 다양한 세션으로 구성된 12시간이 소요되는 방법으로 8시간의 토의시간과 의견의 일치 를 위한 시간으로 구성되어 있으며,

셋째는 매우 압축시킨 방법으로 1-3명이 관련 세션에 대해 의견일치방법으로 성숙도에 대한 질문을 완료시키는 방법이다.

다음 그림은 MMA에 대한 전반적인 프로세스의 흐름을 보여주고 있는데 여러 분리된 사건(event)들을 감소시키기 위해 다양한 방법으로 이러한 단계(step)들을 결합할 수 있다.

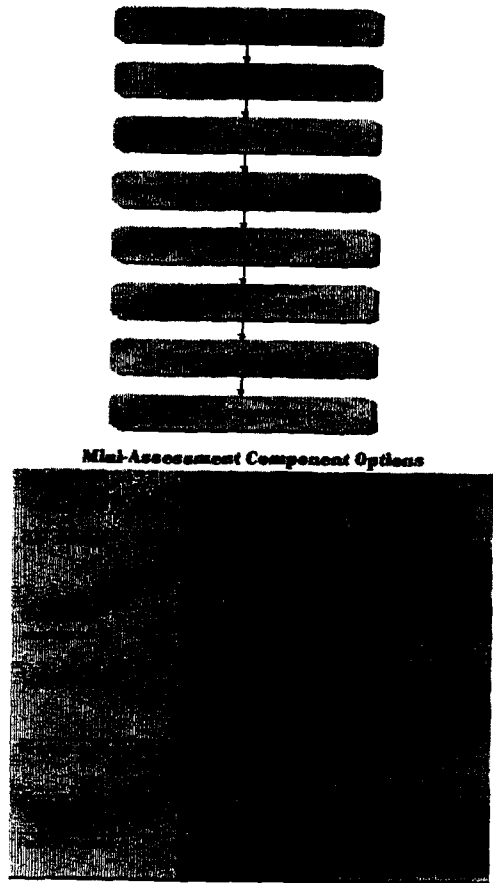
MMA에서 사용하는 주요 데이터 모 집 방법은 프로세스 성숙도에 관한 질문서와 프로젝트에 참여하는 사람들의 토의내용이다.

MMA에 참가하는 모든 프로젝트 팀 구성원은 CMM Orientation을 받아야 한다. MMA가 CMM에 기반을 두고 있지만 CMM 평가 체계를 따르는 것은 아니다. 즉 CMM과는 달리 성숙수준에 대한 공식적인 rating은 하지 않는데, 이것은 적절한 향상 기회를 확인하는데 더 관심을 두기 때문이다.

MMA 방법의 유연성은 대부분의 평가 단계에 유용한 여러 가지 옵션들을 제공하기 때문이다. 옵션들은 그림과 같이 요약되어 있다.

## (3) Self Assesment Model

Paul(1997)<sup>62)</sup>은 개발 프로세스에 대한 정의가 되어 있는, 성숙된 개발 프로세스는 제품의 질을 향상시키



는 중요한 수단인데, CMM, BOOTSTRAP, ISO 15504(SPICE) 등의 개발프로세스 방법은 소규모, 또는 중기업 규모의 기업이 당면한 문제와 요구에 대해서는 소홀히 하고 있으며, 높은 비용이 수반되고 외부 전문가를 이용하여야 하는 복잡한 방법이라고 규정하고 있다.

프로세스 향상 프로젝트의 초기 단계로서 현재 조직의 능력을 결정하는 평가가 필요하며, 이러한 평가를 통해 확인된 강, 약점을 분석하여 기업의 활동계획(action plan)을 수립하는데 이용할 수 있다고 주장한다.

그러나 소규모 기업에서는 이러한 평가에는 여러 가지 장애물이 존재하는데, 주요 장애물은 광범위한 평가와 이에 따른 여러 외부 전문가의 동원 등으로 인한 비용과 노력의 문제이다. 또한 평가 방법이 분산된 책임과 다단계 수준의 조직에 적합한 평가 방법이라는 것이며, 소규모 기업에 있어서는 특정 개인에 대한 의존성이 크므로, 질적 향상 프로그램에 대한

장기간의 투자에 대한 필요성을 느끼지 못한다는 것이며, 직원들도 ISO 9001 같은 품질관리 표준 등에 대한 이해가 부족하다는 것이다. 따라서 초기에는 손쉽게 평가할 수 있는 자기평가방법을 실시한 후 그 결과를 이용하여 기업 내에서 구조적인 인터뷰 등을 하는 것이 바람직하다는 것이다. Technical University Graz, 호주기업 HM&R, 독일/스위스 기업 Synspace 등이 컨소시엄을 구성하여 자기 평가 툴인 SynQuest를 개발하였는데 소프트웨어 프로세스에 관한 37개의 질문으로 구성된 질문서로 이루어져 있다. 프로세스 영역은 CMM이나 ISO 9001의 2-3단계와 관련된 질문으로 주로 구성되어 있고, 프로세스 영역의 8개 보조 속성에 대해 총 330개의 대담으로 구성되어 있다

### III. 연구내용 및 가설의 도출

기존의 소규모 조직, 소규모 프로젝트 등에 적합하도록 변형된 Tailored 모델, 자기평가 모델, 미니평가모델 등과 일반적인 기업을 대상으로 한 일반적 모델에 관한 기존의 문헌연구 결과를 토대로 다음과 같은 가설을 도출하여 검증하고자 하며, 여러 가지 CMM 모델과 ISO/IEC TR 15504(SPICE) 등의 자가 평가 시스템을 구현하고자 한다.

가설 1-1  
조직의 형태에 따라 각 Process의 중요성, 우선순위에 차이가 있다

가설 1-2  
조직의 크기에 따라 각 Process의 중요성, 우선순위에 차이가 있다

가설 1-3  
프로젝트의 수행규모에 따라 각 Process의 중요성, 우선순위에 차이가 있다

가설 2-1  
조직의 형태에 따라 각 Process의 문서화 수준이 프로젝트의 성공에 미치는 영향력의 정도에 차이가 있다

가설 2-2  
조직의 크기에 따라 각 Process의 문서화 수준이 프로젝트의 성공에 미치는 영향력의 정도에 차이가 있다

가설 2-3  
프로젝트의 수행규모에 따라 각 Process의 문서화 수준이 프로젝트의 성공에 미치는 영향력의 정도에 차이가 있다

가설 3-1  
조직의 형태에 따라 각 Process의 문서화 수준이 조직의 장기적인 능력 개선에 미치는 영향력의 정도에 차이가 있다

가설 3-2  
조직의 크기에 따라 각 Process의 문서화 수준이 조직의 장기적인 능력 개선에 미치는 영향력의 정도에 차이가 있다

가설 3-3  
프로젝트의 수행규모에 따라 각 Process의 문서화 수준이 조직의 장기적인 능력 개선에 미치는 영향력의 정도에 차이가 있다

### IV. 자료수집방법

1. 자료 수집 대상 기업 및 기관  
자료수집 대상 기업 및 기관은 다음과 같다.

- (1) 한국 소프트웨어 산업협회 소프트웨어 관련 기업 자료의 5,673개 기업
- (2) 광역시 이상의 주요 정부부처 60여개 정부 기관
- (3) 정부출연기관, 투자기관 등의 50여개 연구 기관
- (4) 정부투자 공공기업 및 기관 등의 120개 기업
- (5) 종합대학, 전문대학 등 200여개의 교육기관

2. 설문자료의 수집방법  
설문지 배포 및 수집은 인터넷을 이용하여 e-mail로 해당기업이나 기관의 IT부서 담당자에게 보내 e-mail에서 응답하여 회수할 수 있

도록 하였으며, ① 동일인, 동일 IP, E-mail 반복 방지, 또는 미 응답 문항 확인하여 갱신 내용 허용 ② 중간 임시 저장방법, 재작업을 위한 불러오기 기능 ③ 타인에게 전달 가능 등의 기능을 이용하여 응답률을 높이고, 회수도 간편하게 할 수 있도록 설계하였다.

## V. 자료 분석

수집된 자료에 대해 일반적인 각종 현황자료 및 통계자료를 도표와 빈도 등의 비교를 통해 소프트웨어 개발조직 및 기관 등의 형태, 규모, 개발프로젝트의 규모 등에 따른 차이를 살펴보고,

가설들의 검증을 위해 ANOVA, MANOVA 분석을 통해 각 그룹 간 차이에 대한 유의적인 수준을 확인하고, Cluster 분석이나, MDS 분석을 통하여 각 그룹 간 유사성의 정도 등을 확인하고 데이터에 내재된 패턴이나 구조를 찾아내어 정책적 자료 등에 활용할 수 있도록 제공하고자 한다.

확인하고자하는 주요 사항은 다음과 같다.

① 조직의 형태에 따라 각 프로세스의 중요성에 차이가 발생할 것이다

○ IT전문독립기업 이나 대기업의 IT 관계사 등은 주로 3단계, 4단계, 5단계 등의 프로세스 카테고리에 많은 관심을 나타낼 것이며, 일반기업이나 기관의 IT조직이나 일반 단위조직 주로 2단계, 3단계의 프로세스 카테고리에 많은 관심을 나타낼 것이다

② 조직의 크기에 따라 각 프로세스의 중요성에 차이가 발생할 것이다

○ 대규모기업이나 조직은 소규모 기업이나 조직과는 달리 주로 3단계, 4단계, 5단계 등의 프로세스 카테고리에 많은 관심을 나타낼 것이며, 소규모기업은 주로 2단계, 3단계의 프로세스 카테고리에 많은 관심을 나타낼 것이다

③ 조직의 프로젝트 규모에 따라 각 프로세스의 중요성에 차이가 발생할 것이다

○ 프로젝트를 수행하는 규모가 큰 기업이나 조직은 프로젝트를 수행하는 규모가 작은 기업이 비해 주로 3단계, 4단계, 5단계 등의 프로세스 카테고리에 많은 관심을 나타낼 것이며, 프로젝트를 수행하는 규모가 작은 기업은 주로 2단계, 3단계의 프로세스 카테고리에 많은 관심을 나타낼 것이다

④ 조직의 형태에 따라 각 Process의 문서화 수준이 프로젝트의 성공과 조직의 장기적인 능력 개선에 미치는 영향력의 정도에 차이가 있다

○ IT전문독립기업 이나 대기업의 IT 관계사 등은 협업 프로젝트가 많으므로 문서화 수준이 프로젝트 성공에 중요한 영향을 주며 장기적인 능력 개선에 미치는 영향력도 클 것이나, 일반기업이나 기관의 IT조직이나 일반 단위조직은 협업 프로젝트가 적으므로 문서화 수준이 프로젝트 성공에 중요한 영향을 주지 못하며 장기적인 능력 개선에 미치는 영향력도 작을 것이다.

⑤ 조직의 크기에 따라 각 Process의 문서화 수준이 프로젝트의 성공과 조직의 장기적인 능력 개선에 미치는 영향력의 정도에 차이가 있다

○ 대규모기업이나 조직은 많은 인력과 많은 프로젝트 수행으로 표준화가 필요하며 이에 따라 문서화 수준이 프로젝트 성공에 중요한 영향을 주며 장기적인 능력 개선에 미치는 영향력도 클 것이나, 일반기업이나 기관의 IT조직이나 일반 단위조직은 적은 인력으로 프로젝트 개발 업무 수행에 더 노력을 집중하여야 하므로 문서화 수준이 프로젝트 성공에 미치는 영향은 크지 않을 것이나 장기적인 능력 개선을 위해서는 문서화가 필요하므로 이에 미치는 영향력은 클 것이다.

⑥ 프로젝트의 수행규모에 따라 각 Process의 문서화 수준이 프로젝트의 성공과 조직의 장기적인 능력 개선에 미치는 영향력의 정도에 차



이가 있다

○ 대형 프로젝트 수행에는 많은 인력과 단위 프로젝트 간 협력이 중요하므로 문서화 수준이 프로젝트 성공에 중요한 영향을 준 것이며 장기적인 능력 개선에 미치는 영향력도 클 것이나, 소형 프로젝트에서는 적은 인력으로 단위 프로젝트 개발 업무 수행에 더 노력을 집중하여야 하므로 문서화 수준이 프로젝트 성공에 미치는 영향은 크지 않을 것이나 장기적인 능력 개선을 위해서는 문서화가 필요하므로 이에 미치는 영향력은 클 것이다.

## VI. 결론 및 기여 내용

본연구의 의의와 기여내용은

첫째, 우리나라 기업이나 조직의 소프트웨어 프로세스의 성숙 수준에 대한 전반적인 조사를 통해 현재 우리나라 기업이나 조직의 능력 수준에 대한 이해 정도와 수준에 대한 개략적인 현황을 살펴보았다는 것이며,

이를 통해, 각 소프트웨어 프로그램을 개발하는 조직이나 기업의 입장에서는,

(가) 현재 각 기업이나 조직이 속해있는 분야나 형태에 따라, 각 기업이나 조직 능력의 성숙수준에 대한 현재의 위치를 파악함으로써, 조직이나 기업의 경쟁력 등을 파악하는데 도움을 줄 수 있으며,

(나) 소프트웨어 프로세스의 능력 향상을 위한 조직이나 기업의 강점과 약점, 기회와 위기 등의 분석을 위한 자료를 제공하여 줄 수 있다는 것이다. 프로세스 카테고리별로 어떠한 측면에 어떠한 강점과 약점이 있으며, 이를 보완하고, 발전시킬 부분은 무엇이며, 또한, 능력 수준에 존재하는 프로세스에서 조직이나 기업의 수준별 위치는 어떠한지, 문제점은 무엇인지에 대한 해답을 도출하는데 도움을 줄 수 있다는 것이다.

(다) 분석된 자료를 이용하여 기업의 프로세스 향상을 위한 장, 단기 계획 수립과 이를 수행하기 위해 다음과 같은 전략을 도출하여 이용할

수도 있을 것이다.

① 아직 불완전한 초기 단계에 있는 소프트웨어 개발을 수행하는 기업이나 조직에게는 성숙 능력 모델에서 제시하는 각각의 프로세스와 프랙티스 등을 이용하여, 조직의 소프트웨어 프로세스 향상을 위해 계획을 수립할 때 수행하여야 할 각종 내용을 조직의 특성에 맞도록 계획안에 반영할 수 있으며,

② 어느 정도 성숙단계에 있는 소프트웨어 개발을 수행하는 기업이나 조직에게는 또한, 경쟁기업이나 기관에게 뒤지지 않고 기업이나 조직을 생존시키기 위해 필요한 프로세스를 확인하여 수행 계획을 수립할 수 있으며,

프로그램을 공급자에게서 획득하려는 조직이나 기관에게는

① 현재의 소프트웨어 개발을 수행하는 기업이나 조직의 전반적인 성숙수준의 파악으로 소프트웨어 개발기업이나 조직의 성숙수준에 적절한 프로젝트 수행내역에 대한 계획을 사전에 수립하여 대응 할 수 있으며,

② 개발하여 획득하려는 소프트웨어 프로젝트에 특별히 필요하거나 요구되는 수준이 무엇인지를 사전에 파악하여 이러한 프로세스를 완전하게 수행하고 있는 적절한 형태의 기업이나 조직을 선정할 기준을 수립할 수 있다.

국가적으로는

(가) 소프트웨어 전체 산업과 각 기업들의 소프트웨어 개발 프로세스의 성숙 수준 현황을 파악할 수 있으며,

(나) 소프트웨어 산업발전이나 경쟁력 강화를 위한 다음과 같은 정책제안을 도출할 수 있다

① 소규모의 전문 소프트웨어 개발 기업에서는 대규모 기업과 달리 인력의 부족과 시간과 금전적인 제약으로 인해 문서화 작업에 상당한 애로사항을 느끼고 있으며, 이에 대한 오버헤드

가 크므로 소규모 소프트웨어 기업이나 조직을 활성화하기 위해서 문서화 작업을 보조할 수 있는 프로그램의 개발을 통해 지원 할 수 있는 노력이 필요하며,

② 또는 문서화 작성방법과 절차 등의 표준화 방법 등의 개발을 통해서 소규모 소프트웨어 기업이나 조직에서 신속하고, 저렴하게 활용할 수 있도록 지원이 필요하다.

③ 또한, 소프트웨어 프로세스 향상을 통한 경쟁력 제고를 위해 정부기관 및 공공기관의 소프트웨어 개발 프로젝트 발주 시에 반드시 각종 프로세스의 일정 부분에 대해서는 단계적으로 문서화를 요구하고, 지키도록 제도적으로 시행 하여, 장기적으로 기업의 경쟁력 향상을 유도할 수 있으며,

소프트웨어 프로세스의 성숙 수준에 관한 모델을 개발하는 기관이나 평가기관의 입장에서는  
(가) 모델 개발 시 각 프로세스의 구성 등에 대해 다음과 같은 정책적 대안을 도출 할 수 있겠다.

① 대부분의 소프트웨어 프로세스 성숙수준이 1-2 단계를 벗어나고 있지 못함으로서 각 기업 및 조직의 소프트웨어 프로세스의 성숙수준에 대해 정확한 측정에 애로사항이 발생하며,

② 또한 각 기업 및 조직 간에 차이를 나타내기가 어려우므로, 1-2단계의 성숙수준에 대하여는 각 프로세스에 대해 좀더 세부적인 분류를 통해, 차이가 명확하게 분류되어 평가가 이루어 질 수 있도록 추가 분류 작업이 필요하다.

둘째, 조직의 형태, 조직의 규모, 프로젝트 수행 규모에 따라 조직이나 기업의 성숙수준에 차이가 있으며, 조직의 형태, 조직의 규모, 프로젝트 수행 규모에 따라 각 프로세스가 조직이나 기업에서 느끼는 중요성의 정도가 다르며, 조직의 형태, 조직의 규모, 프로젝트 수행 규모에 따라 가장 개선할 필요성이 있는 각 프로세스의 내용이 다르다는 것이다.

따라서 각 프로세스의 중요성과 달성이 어려운 프로세스 등에 따른 각 성숙수준의 구성 프로세스의 재배치 등에 이용할 수 있으며, 소규모의 조직이나 기업과 소규모 프로젝트를 수행하는 기업이나 조직에 적합하도록 현재의 성숙수준 내에서 단계를 상세하게 구분하여 소규모의 조직이나 기업과 소규모 프로젝트를 수행하는 기업이나 조직의 성숙수준을 더 정교하고 상세하게 표현하도록 이용할 수 있겠다.

셋째, 소프트웨어 프로세스를 자가 평가를 이용하여 평가할 수 있는 시스템은, 각 소프트웨어 개발 조직이나 기업에서 실제로 소프트웨어 프로세스 향상을 위한 여러 모델들을 직접 평가받기 위해서는 외부의 전문 평가기관을 이용함으로써 많은 시간과 비용이 수반되어 중소기업에서는 이로 인한 부담을 느끼고 있으며, 개발업무의 진행에도 영향을 줄 수 있지만 본 모델을 이용하면, 소프트웨어 프로세스 향상을 위해 간편하게 각 모델에 따른 질문에 응답함으로써 각 기업이나 조직의 현재 수준을 알 수 있으며, 외부의 도움 없이 간편하게 SPI 모델의 수준을 자체적으로 평가할 수 있다는 것이다.

넷째, 또한 여러 기업이나 조직이 현재 각 SPI 모델에 따라 SWOT 분석 등을 통해 각 기업이나 조직의 문제점의 확인에 따른 향후 추진 방향 등을 수립하는데 도움을 줄 수 있으며, 여러 다른 기업이나 조직의 평가된 자료를 이용하여 다른 기업이나 조직 또는 동등한 위치에 있는 기업이나 조직과 비교하여 자기의 기업이나 조직의 위치 등을 분석하여 이용할 수도 있으며,

타 기업이나 조직이 기간의 경과에 따라 성숙수준의 변화나 프로세스의 방향 등의 변화를 통해 이러한 자료를 이용하는 기업이나 조직의 정책 자료로 이용할 수 있다.

다섯째, 조직의 형태 및 조직의 규모 등에 따라 기존의 프로세스 카테고리를 재분류하여, 조직의 형태 및 조직의 규모 등에 알맞은 카테고리

를 제공하여 기업이나 조직의 프로세스 향상을 위해 필요한 카테고리를 쉽게 파악할 수 있도록 제공하고 있다.

1 J. Batista and A. Dias de Figueiredo, "SPI in a very small team : a case with CMM". Software Process Improvement and Practice, Vol. 5, 2000, pp. 243-250.

2 정기원, 소프트웨어 프로세스와 품질, 홍릉과학, 1997, pp. 157.

3 H. Saiedian, R. Kuzara, "SEI Capability Maturity Model's Impact on Contractors", IEEE Computer, Vol. 28, Issue 1, Jan. 1995, pp. 16 -26.

4 James Herbseb, "Software Quality and the Capability Maturity Model", Communication of ACM, Vol.40, No. 6, June, 1997, pp. 30-40.

5 Butler, K.L. "The Economic benefit of software process improvement" CrossTalk, July 1995, pp. 14-17.

6 Bradford K. Clark, "Quantifying the Effects of Process Improvement on Effort" IEEE Software, Nov./Dec. 2000, pp. 65-70.

7 Karlheinz Kautz, "Software Process Improvement in Very Small Enterprises : Does It Pay Off?", Software Process Improvement and Practice, Vol. 4, 1988, pp. 209-226.

8 Mark C. Paulk, Bill Curtis, Mary Beth Chrissis, Charles V. Weber, Capability Maturity Model for Software, Version 1.1, SEI Technical Report MU/SEI-93-TR-024, ESC-TR-93-177, February, 1993.

9 Jack Cooper, Matthew Fisher, S.Wayne Sherer, Software Acquisition Capability Maturity Model® Version 1.02, SEI Technical Report CMU/SEI-99-TR-002, ESC-TR-99-002, April 1999.

10 Software Engineering Institute, A

Systems Engineering Capability Maturity Model Version 1.1, CMU/SEI-95-MM-003, November 1995.

11 CMU-SEI, Process Maturity Profile of the Software Community 1994 MID Year Update, Oct. 1994, pp. 1-27.

12 CMU-SEI, Process Maturity Profile of the Software Community 1994 Year End Update, April, 1995, pp. 1-28.

13 CMU-SEI, Process Maturity Profile of the Software Community 1995 Mid-Year Update, Nov. 1995, pp. 1-33.

14 CMU-SEI, Process Maturity Profile of the Software Community 1995 Year End Update, April, 1996, pp. 1-33.

15 CMU-SEI, Process Maturity Profile of the Software Community 1996 Mid-Year Update, Nov. 1995, pp. 1-34.

16 CMU-SEI, Process Maturity Profile of the Software Community 1996 Year End Update, May, 1997, pp. 1-32.

17 CMU-SEI, Process Maturity Profile of the Software Community 1997 Mid-Year Update, Oct. 1997, pp. 1-32.

18 CMU-SEI, Process Maturity Profile of the Software Community 1997 Year End Update, May, 1998, pp. 1-33.

19 CMU-SEI, Process Maturity Profile of the Software Community 1998 Mid-Year Update, Dec. 1998, pp. 1-33.

20 CMU-SEI, Process Maturity Profile of the Software Community 1998 Year End Update, March, 1999, pp. 1-33.

21 CMU-SEI, Process Maturity Profile of the Software Community 1999 Mid-Year Update, Dec. 1999, pp. 1-33.

22 CMU-SEI, Process Maturity Profile of the Software Community 1999 Year End Update, March, 2000, pp. 1-33.

23 CMU-SEI, Process Maturity Profile of the Software Community 2000 Mid-Year

- Update, Aug. 2000, pp. 1-33.
- 24 CMU-SEI, Process Maturity Profile of the Software Community 2000 Year End Update, March, 2001, pp. 1-33.
- 25 CMU-SEI, Process Maturity Profile of the Software Community 2001 Mid-Year Update, Aug. 2001, pp. 1-33.
- 26 CMU-SEI, Process Maturity Profile of the Software Community 2001 Year End Update, March, 2002, pp. 1-34.
- 27 Published Maturity Levels, <http://www.sei.cmu.edu/sema/published.ml.html>, 2002. 03. 27.
- 28 Mark C. Paulk, Bill Curtis, Mary Beth Chrissis, Charles V. Weber, Capability Maturity Model for Software, Version 1.1, Technical Report CMU/SEI-93-TR-024 ESC-TR-93-177, February 1993.
- 29 CMMI Product Development Team, Capability Maturity Model Integrated for Systems Engineering/Software Engineering, Version 1.0 Continuous Representation, CMMI-SE/SW, V1.0, CMU/SEI-2000-TR-019 ESC-TR-2000-019, Aug. 2000.
- 30 CMMI Product Development Team, Capability Maturity Model Integrated for Systems Engineering/Software Engineering, Version 1.0 Staged Representation, CMMI-SE/SW, V1.0, CMU/SEI-2000-TR-019 ESC-TR-2000-019, Aug. 2000.
- 31 CMMI Product Development Team, CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development/Acquisition, Version 1.02d DRAFT CMMI-SE/SW/IPPD/A, V1.02d, DRAFT. Staged Representation, December 2000.
- 32 CMMI Product Development Team, CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development/Acquisition, Version 1.02d DRAFT CMMI-SE/SW/IPPD/A, V1.02d, DRAFT Continuous Representation, December 2000.
- 33 CMMI Product Development Team, CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development/Supplier Sourcing, Version 1.1, Continuous Representation (CMMI-SE/SW/IPPD/SS, V1.1, Continuous), Mar. 2002.
- 34 CMMI Product Development Team, CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development/Supplier Sourcing, Version 1.1, Staged Representation (CMMI-SE/SW/IPPD/SS, V1.1, Staged), Mar. 2002.
- 35 CMMI Product Development Team, CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development, Version 1.1, Staged Representation (CMMI-SE/SW/IPPD, V1.1, Staged), Jan. 2002.
- 36 CMMI Product Development Team, CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development, Version 1.1, Continuous Representation (CMMI-SE/SW/IPPD, V1.1, Continuous), Jan. 2002.
- 37 CMMI Product Development Team, CMMI for Systems Engineering/Software Engineering, Version 1.1, Staged Representation (CMMI-SE/SW, V1.1, Staged), Jan. 2002.
- 38 CMMI Product Development Team, CMMI for Systems Engineering/Software Engineering, Version 1.1, Continuous Representation (CMMI-SE/SW, V1.1, Continuous), Jan. 2002.

- 39 Mark C. Paulk, Bill Curtis, Mary Beth Chrissis, Charles V. Weber, Capability Maturity Model for Software, Version 1.1, CMU/SEI-93-TR-024, Feb. 1993. pp.43-49.
- 40 Paul Byrnes, Mike Phillips, Software Capability Evaluation Version 3.0 Method Description Technical Report CMU/SEI-96-TR-002 ESC-TR-96-002, April. 1996. pp. 8 -10.
- 41 ISO/IEC, SPICE, Software Process Assessment , Version 1.00, June 1995.
- 42 ISO/IEC, TECHNICAL REPORT ISO/IEC TR 15504-1 First edition, Aug.1998.
- 43 ISO/IEC, TECHNICAL REPORT ISO/IEC TR 15504-2 First edition, Aug.1998.
- 44 ISO/IEC, TECHNICAL REPORT ISO/IEC TR 15504-3 First edition, Aug.1998.
- 45 ISO/IEC, TECHNICAL REPORT ISO/IEC TR 15504-4 First edition, Aug.1998.
- 46 ISO/IEC, TECHNICAL REPORT ISO/IEC TR 15504-5 First edition, Aug.1998.
- 47 ISO/IEC, TECHNICAL REPORT ISO/IEC TR 15504-6 First edition, Aug.1998.
- 48 ISO/IEC, TECHNICAL REPORT ISO/IEC TR 15504-7 First edition, Aug.1998.
- 49 ISO/IEC, TECHNICAL REPORT ISO/IEC TR 15504-8 First edition, Aug.1998.
- 50 ISO/IEC, TECHNICAL REPORT ISO/IEC TR 15504-9 First edition, Aug.1998.
- 51 Karl E. Wieners and Doris C. Sturzenberger, A Modular Software Process Mini-Assesment Method, IEEE Software, 2000, January/February, pp. 62-69.
- 52 정기원, 소프트웨어 프로세스와 품질, 흥룡 과학, 1997, pp. 242.
- 53 Roman Vajde Horvat, Ivan Rozvan and Jozsef Gyorkos, Managing the Complexity of SPI in Small Companies, Software Process Improvement Practice, 2000, Vol. 5, pp. 45-54.
- 54 Paul Grunbacher, "A Software Assesment Process for Small Software Enterprise", IEEE, 1997, pp. 123-128.
- 55 Sergio Otoyá and Narciso Ceroa, "An Experience : A Small Software Company Attempting to Improve its Process", Software Technology and Engineering Practice Conference, Sep. 1999.
- 56 Hareton K. N. Leung, Terence C. F. Yuen, "SOFTWARE PROCESS IMPROVEMENT AND PRACTICE", 2001, Vol.6, pp. 67-83.
- 57 Sergio Otoyá and Narciso Ceroa, "An Experience : A Small Software Company Attempting to Improve its Process", Software Technology and Engineering Practice Conference, Sep. 1999.
- 58 Brodman, J. G. and Johnson, D. I, 1996. Return on investment from software process improvement as measured by U.S. industry. Crosstalk, 9(4).
- 59 Brodman, J. G. and Johnson, D. I. 1994. What Small Business and Small Organizations Say About the CMM. In Proceedings of the 16 International Conference on Software`Engineering, Sorrento Italy.
- 60 Romana Vajde Horvat, Ivan Rozman and Jozsef Gyorkos, "Managing the Complexity of SI in Small Companies", Software Process Improvement Practice, 2000; Vol. 5 : pp. 45-5.
- 61 Karl E. Wieners and Doris C. Sturzenberger, "A Modular Software Process Mini-Assessment Method", IEEE Software, January/February, 2000, pp. 62-69.
- 62 Paul Grunbacher, "A Software Assessment Process for Small Software Enterprises", IEEE, 1997, pp. 123-128.