

# 컴포넌트기반 개발 프로세스 성숙도 모형설계

나희동°, 김태우

(주)투이컨설팅 CBSE팀

## The Design for Component based Development Maturity Model

Na Hee-Dong°, Kim Tae-Woo

2e Consulting, Co.

E-mail : hdna@2e.co.kr, barahm@2e.co.kr

### 요 약

컴포넌트 기반 개발 프로세스는 그 동안의 구조적 분석/설계 기법과 정보공학, 객체지향 등의 공학적 분석기술을 수용하고 있다. 따라서 CBD를 적용하려는 조직은 다양한 소프트웨어 공학적 기법과 아키텍처 설계 기술들을 습득해야 하였다. 또한 CBD의 핵심 역량인 반복적 개발 프로세스와 아키텍처 설계 역량을 갖추기 위해 각 개발업체 들은 CBD기술의 도입초기에 상당한 혼돈과 어려움을 겪어야 했다.

이에 본 논문에서는 CBD를 조직차원에서 습득하고 적용하는 과정을 5개의 단계로 나누고 각 단계의 목표를 설정한 컴포넌트 개발 프로세스 성숙도 모형을 제시한다. CDMM(Component based Development capability Maturity Model)으로 명명한 본 개발 능력 성숙도 모형은 소프트웨어 개발조직이 CBD를 습득하기 위해 어떤 단계를 거쳐야 하는지 또한 컴포넌트 개발을 위한 핵심기술이 어떤 것들인지를 제시한다. 이는 향후 CBD를 확산시키고 컴포넌트 기술을 통한 생산성향상과 소프트웨어 개발비 절감을 위해 기업과 국가의 정책적 지표로 활용할 수 있을 것이다.

### 1. 서론

컴포넌트 기반 개발방법론은 단순한 소프트웨어 프로세스와는 달리 체계적인 아키텍처와 잘 정제되고 요구변화에 강건한 컴포넌트 설계기술을 전제로 한다. 이밖에도 CBD의 궁극적인 목표인 비즈니스의 다양한 요구에 대한 정량적인 생산능력 통제가 가능하기 위해서는 비즈니스 모델링과 이의 관리기술 확보가 선행되어야 한다. 이러한 CBD의 특징은 일반 업체에서 컴포넌트 기반 소프트웨어 생산기술 확보를 어렵게 하는 주된 요인이 되고 있다. [오영배, 나희동, 박준성, 백두권, "컴포넌트 기반개발프로세스", 한국정보과학회 소프트웨어공학회지 2002. 5.]

또한 컴포넌트 기술을 현대의 경영이 필요로 하는

정보시스템의 신속한 구축, 변경확장의 용이성, 타 시스템과의 호환성 및 유지보수성, 개발과정의 생산성 향상을 달성하고자 하는 소프트웨어 프로세스, 제품구현기술 및 아키텍처 기술의 총체적 개념으로 정의한다. [박준성, "CBD 도입전략", 2001. 한국소프트웨어컴포넌트컨소시엄.] 이는 컴포넌트 기술이 단순한 소프트웨어 생산기술을 넘어 소프트웨어를 매개로 한 현대 IT산업의 발전을 통칭하는 것을 의미한다. 이는 컴포넌트 기술을 익히고 적용하고 그 효과를 측정하는 것은 개발조직이 단번에 매우 많은 기술을 습득해야 하는 것을 의미한다.

이러한 특징으로 실제 대규모 프로젝트에 컴포넌트 기술을 적용하기 위해서는 개발조직이 일정수준 이상의 프로세스 성숙도 수준과 함께 업무적,

기술적 성숙을 필요로 한다. 성숙(Maturity) 수준이란 일정 시간을 통해 일련의 단계를 거쳐 달성해야 하는 정량적 지표로서 최종 제품의 품질과 생산성에 직접적인 영향을 끼친다. 또한 성숙도 수준이란 단순한 능력지표와는 다른 것으로 각 단계간 적정수준의 성장이 상호 유기적으로 연계되어 있음을 의미한다.

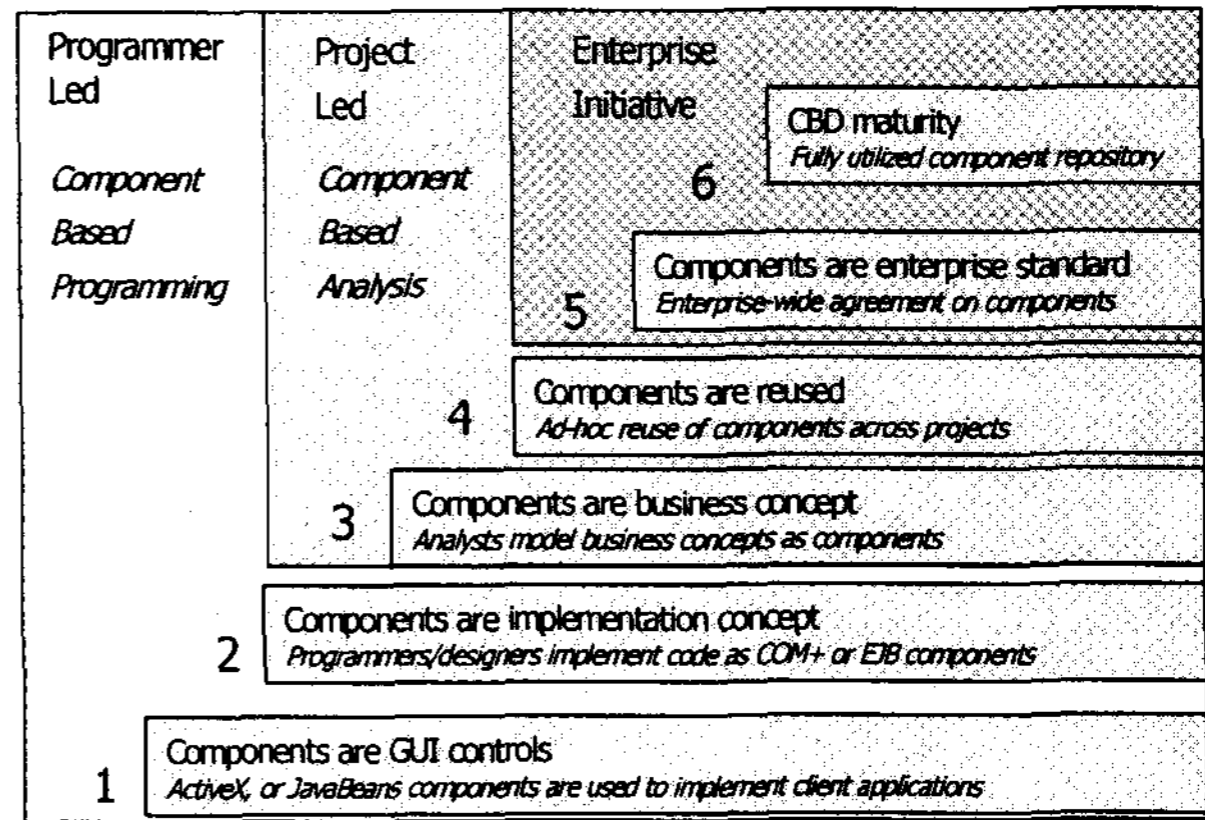
즉 컴포넌트 기반 개발을 통한 개발 생산성 향상을 기하기 위해서는 개발조직이 자신의 조직수준에 맞는 정확한 단계별 목표수준과 구체적인 기술영역에 대한 이해가 필요하다. 이에 본 논문에서는 컴포넌트 기술을 이용한 소프트웨어 개발조직의 컴포넌트 개발 프로세스 성숙도 모형(Component based Development capability Maturity Model)을 정의하고 각 성숙도 수준별 핵심기술영역(Key Technology Area)을 정의하고자 한다.

이는 기존의 컴포넌트 성숙도 모형(Burtler Group's Component Maturity Model)과는 달리 실제 소프트웨어 생산과정에서 컴포넌트 기술을 적용하려는 조직에게 자신의 컴포넌트 개발성숙도 수준을 평가하고, 또 다음 수준으로 성장하기 위해서 어떤 핵심기술영역을 달성해야 하는지에 대한 정량적 지표를 제시하는 것이다.

## 2. 관련연구

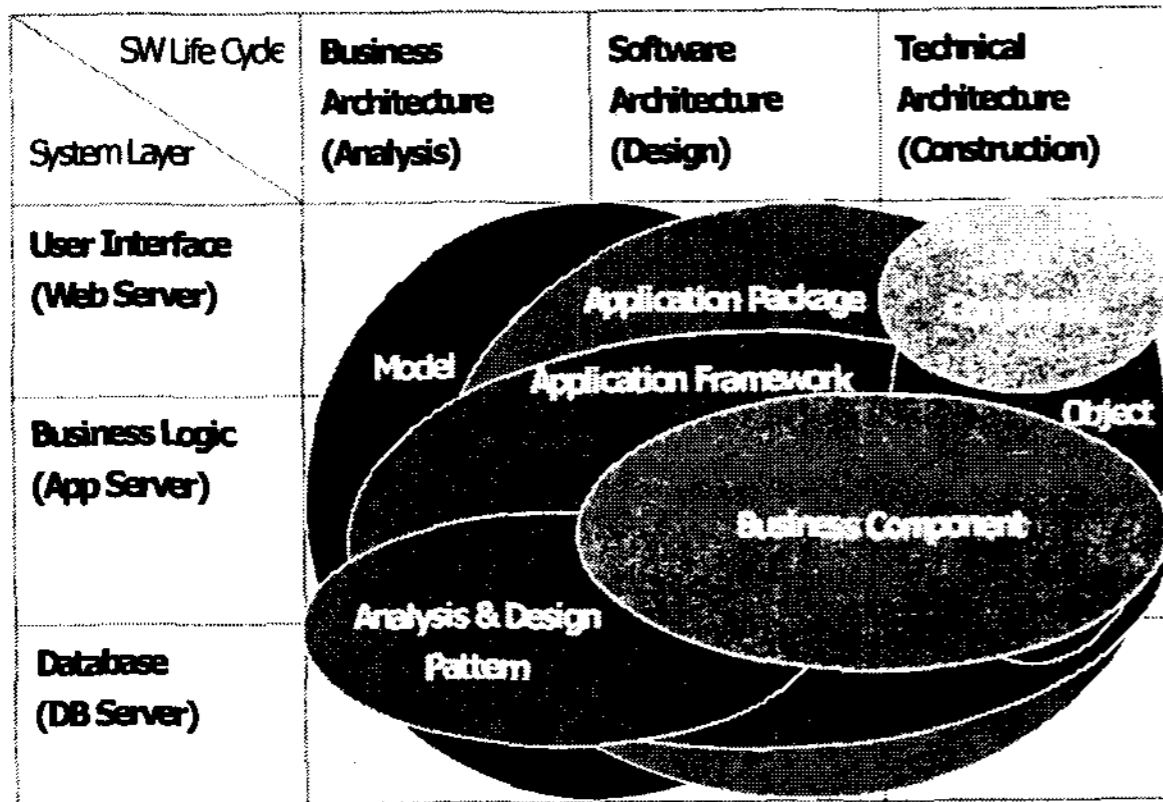
컴포넌트 기술의 성숙도 모형으로는 개별 기업들이 컴포넌트 기술을 적용하면서 겪게 되는 몇 가지 단계를 명시한 영국의 Burtler Group에서 제안한 기업별 Component Maturity Model이 있다.<그림 1>

본 컴포넌트 성숙도 모형은 컴포넌트 기술을 적용하는 기업이 최초 GUI컨트롤 등의 구현수준 컴포넌트를 위주로 컴포넌트 단위의 개발을 하는 것으로 시작하여 단계적으로 비즈니스 로직을 컴포넌트화 하는 컴포넌트 중심의 분석, 그리고 최종적으로 컴포넌트를 전사적으로 확산하여 적용하는 단계를 거치게 되는 과정을 설명하고 있다. 또한 5 단계 이후에 컴포넌트를 전사적으로 확산시키기 위해서는 전사적인 컴포넌트 표준을 정의하고 이를 적용하기 위한 정치적인 저항(Political Hurdle)에 부딪힌다고 정의한다.



<그림 1> Burtler's Component Maturity Model 또한 이 컴포넌트 성숙도 모형을 기반으로 유럽에서는 몇몇 대학들과 기업들이 함께 OOSPICE (Software Process Improvement and Capability determination for Object Oriented / component based software development)라는 프로젝트를 진행하고 있다. [ OOSPICE Project Home Page, <http://www.oospice.com>] OOSPICE는 객체지향 기술을 토대로 컴포넌트를 개발하는 CBD프로세스와 프로세스 평가방법론을 만들고, 또한 CBD 방법론과 통합CBD 프로세스 메타모형(Unified CBD Process Meta Model)을 만들려는 프로젝트이다. OOSPICE는 컴포넌트 생산자의 개발능력과 컴포넌트 구축프로세스의 불일치를 해결하고자 [Herbsleb., J. and Zubrow, D. and Goldenson, D. and Hyes, W. and Paulk, M., Software Quality and the Capability Maturity Model, Comm. ACM vol. 40 (1997), no. 6. pp. 30--45] Butler그룹의 컴포넌트 성숙도 수준을 달성하기 위해서 컴포넌트 프로세스의 품질을 높이고자 한다. 이는 소프트웨어 제품(Product)의 품질은 소프트웨어 프로세스(Software Process)의 품질에 기인한다는 Humphrey의 이론에 따라 궁극적인 컴포넌트의 기술수준 향상과 관련한 컴포넌트 기반 프로세스(Component based Software Process)의 품질을 향상코자 하는 것이다. [ Humphrey, W.S., Managing the Software Process, Addison-Wesley Reading Mass. 1989] [Cignoni, G.A.:Software Process Technologies and the Competitiveness Challenge, Conradi, R. (ed.): Software Process Technology, 7th European Workshop EWSPT 2000, Springer 2000, pp. 151--155] 그러나 Burtler Group의 컴포넌트 성숙도 모형은

실제 조직이 컴포넌트 기술을 적용할 때 UI수준의 컴포넌트와 함께 비즈니스 컴포넌트를 동시에 적용하거나 또는 UI컴포넌트 등과 같은 구현수준 컴포넌트보다 데이터모형의 재사용성이 높은 현실성을 반영하지 못한다. 예를 들어 과거 클라이언트/서버 환경에서 비 객체지향으로 개발된 시스템을 웹 기반의 컴포넌트 환경으로 전환하고자 할 때 재사용성이 높은 순서는 데이터모형, 화면설계 그리고 핵심 비즈니스 알고리즘 순이다.



<그림 2> 재사용 컴포넌트의 분류

즉 버틀러 그룹의 컴포넌트 성숙도 모형에서 제시한 바와 같이 컴포넌트 기술을 적용하는 초기단계에서는 UI컨트롤과 같은 프로그래밍 관점의 컴포넌트에서 점차 분석수준과 전사적(Enterprise Wide) 수준의 확산단계를 거친다는 것과 실제 상황이 다른 것이다. 이는 기본적으로 버틀러 그룹의 컴포넌트 성숙도 모형에서 전제한 기본적인 전제조건인 컴포넌트 기술의 확산순서가 잘못된 것이다.

<그림2>와 같이 초기 컴포넌트 기술의 적용단계와는 달리 최근에는 전통적인 실행 가능한 컴포넌트 외에 패턴이나 비즈니스모형 등을 재사용 컴포넌트의 영역에 포함하고 있다.[ Coad, P. Object Models: Strategies, Patterns and Applications. Englewood Cliffs, NJ: Yourdon Press, 1995][ Fowler, M. Analysis Patterns: Reusable Object Models. Reading, MA: Addison Wesley, 1997.][ Gamma, E. et al. Design Patterns: Elements of Reusable Object-Oriented Software. Reading, MA: Addison Wesley, 1995.] 따라서 전통적인 관점의 컴포넌트 성숙도 분류에서는 이러한 다양한 관점의 컴포넌트 기술의 단계적 적용모형을 제시하지 못한다.

또한 기술의 성숙도 모형에서 반드시 제시해야 할 조직의 기술성숙도 평가와 단계별 기술성숙도 발전모형을 제시하고 있지 못하다. 더군다나 초기 컴포넌트 기술의 해석과는 달리 최근에는 컴포넌트 기반 개발 프로세스(CBD)를 Component 단위의 개발, 조립, 유지보수를 통해 현대 경영이 필요로 하는 정보시스템의 신속한 구축(Time to Market), 변경 확장의 용이성(Flexibility)와 타 시스템과의 호환성(Interoperability)를 달성하고자 하는 SW Engineering 프로세스, 방법론 및 기술의 총체적 개념으로 정의하고 있다. [ Park, J.S. "A New Revolutionary Paradigm of Software Development for Mainstream Business Operations,"

International Journal of Technology Management Vol. 20, No. 3/4, 2000. pp. 272-286.]

이와 같이 비즈니스와 IT를 고려한 확장된 컴포넌트 기술영역을 고려할 때 버틀러 그룹의 성숙도 모형은 컴포넌트 기술을 습득하는 조직이 달성해야 할 단계별 비즈니스의 목표수준을 제시하지 못하고 있는 것이다. 더 나아가서 이러한 초기 수준의 컴포넌트 성숙도 모형을 기반으로 연구중인 OOSPICE에서도 CBD를 기반으로 한 방법론과 평가모형에 대한 연구만을 할 뿐 조직의 성숙수준에 따른 CBD 프로세스의 변화를 반영하지 못하고 있다.

### 3. 컴포넌트 개발 프로세스 성숙도 모형

본 논문에서는 재사용 컴포넌트 위주의 초기 컴포넌트 성숙도 모형의 단점을 보완하면서 향후 컴포넌트를 개발하려는 조직이 단계적인 목표를 설정하여 컴포넌트 개발기술과 개발능력의 성숙도 수준향상을 위한 컴포넌트 개발 프로세스 성숙도 모형(Component based Development capability Maturity Model : 이하 CDMM이라 명명함)을 제시하고자 한다.

기존의 컴포넌트 성숙도 모형이 컴포넌트를 적용하는 조직이 전사적인 컴포넌트 표준수립과 이를 적용하는 단계를 다루는 반면 본 논문에서 제안하는 컴포넌트 개발 성숙도 모형은 컴포넌트를 개발하는 조직의 성숙단계에 관한 것이다. 이는 궁극적으로 각 단계별 목표수준을 달성함으로써 최종적으로 컴포넌트 기술을 통한 생산성 향상과 재사용성, 적시생산(Time to Market) 및 변경확장의 용이성과 호환성의 목표를 이룰 수 있게 할 것이다.

### 3.1 컴포넌트 개발 프로세스 성숙도 단계 정의

CDMM에서는 컴포넌트 개발 프로세스 성숙도 수준의 최종적 목표를 미국의 카네기멜론 대학의 SEI에서 제시한 소프트웨어의 Software Product Line으로 설정하였다. Software Product Line 프로그램은 아키텍처 기반 개발을 통해 다양한 소프트웨어 제품라인을 구축하려는 목표를 가지고 있다. [Carnegie Mellon University's Product Line Program Homepage, [http://www.sei.cmu.edu/programs/pls/pl\\_program.html](http://www.sei.cmu.edu/programs/pls/pl_program.html)]

제품계열(Software Product Line)프로그램은

5단계 (Organization)	발생가능한 비즈니스 요구에 대한 정량적인 생산능력 통제(Product Line) 요구 조정에 따른 아키텍처의 최적화(동적 컴포넌트 영향 평가) 신규 어플리케이션 아키텍처 예측 생산
4단계 (Business)	재사용 컴포넌트 리파지토리의 구축 정량적인 아키텍처 평가 및 컴포넌트 적응성, 안정성 통제 비즈니스의 요구에 대한 아키텍처 영향도 평가 및 반영
3단계 (Architecture)	팀간 기술적 협업이 가능한 수준의 컴포넌트 분할 프로젝트 템플릿을 통한 지식의 재사용 체계적인 아키텍처 관리 (Framework & Architecture Style 적용) 최적화 된 반복 프로세스를 적용할 수 있음.
2단계 (Component)	점진적 프로세스를 통한 핵심 요구관리를 수행함. 비 정형적 아키텍처를 구현할 수 있음. 패턴기술을 적용한 체계적 설계 기술 습득이 주 목표임.
1단계 (Process)	산발적인 기술의 적용으로 결과를 보장하지 못함. 명확한 아키텍처가 없으며, 컴포넌트를 식별하지 못함. 모델링 기법 및 개발 프로세스 습득이 주 목표임.

<표 1> 컴포넌트 개발 프로세스 성숙도 단계(CDMM)

Product Line Practice (PLP)와 Architecture Tradeoff Analysis(ATA)를 주축으로 한다. PLP는 소프트웨어 제품의 공통성을 극대화하여 재사용성을 증진시키고 개발위험을 최소화 시키는 개발조직의 체계를 정의하려는 것이며, ATA는 소프트웨어 아키텍처를 통해 여러 시스템들에서 지속적으

로 재사용할 수 있는 추상모델을 정의하고자 하는 것이다. [The Product Line Practice (PLP) Initiative, [http://www.sei.cmu.edu/plp/plp\\_init.html](http://www.sei.cmu.edu/plp/plp_init.html)] [Software Architecture and the Architecture Tradeoff Analysis Initiative, [http://www.sei.cmu.edu/ata/ata\\_init.html](http://www.sei.cmu.edu/ata/ata_init.html)] 위와 같은 제품계열의 핵심 요소인 PLP와 ATA의 목표는 컴포넌트 기반 개발의 핵심적인 목표와 같다. 또한 컴포넌트 기반 개발능력을 축적하기 위해서 달성해야 하는 핵심기술 요소인 재사용 컴포넌트 기반 개발조직과 아키텍처 개발능력에 대한 기술요소를 정의하고 있다.

컴포넌트 기술을 이용한 개발의 최종적인 목표는 발생 가능한 외부의 비즈니스 요구에 대한 정량적 생산능력 통제와 아키텍처의 최적화이다.

컴포넌트 기반 개발방식을 적용하려는 조직에서 이러한 최종적인 컴포넌트 기반 개발능력을 보유하는 과정을 [표1]과 같이 5단계로 나누었다.

첫 번째 단계에서는 통합모델링언어(UML)와 비교적 정형화 된 개발 프로세스를 습득하는 것을 주목표로 한다. 이는 일반적으로 컴포넌트 기술을 프로세스(Process) 측면, 제품설계기술(Product) 측면, 업무기술(Business) 측면으로 나눌 때 가장 먼저 CBD로 이행하려는 조직에서 습득해야 하는 기술이 프로세스 기술임을 반영한 것이다.

2단계에서는 개발위험관리 중심의 점진적 개발 프로세스를 통한 핵심요구관리를 수행할 수 있는 수준으로서, 정형적인 CBD프로세스를 정의하고 각 개발공정에 대한 절차와 산출물을 정의하는 단계이다. 또한 별도의 핵심 설계팀(SA팀)을 구축하게 되고 이 팀을 중심으로 패턴과 아키텍처 스타일과 같은 제품설계기술을 습득해야 한다.

3단계에서는 제품설계기술이 안정화되어 조직수준의 프레임워크와 아키텍처가 구축된 상태이며 또한 최적화 된 반복적 프로세스를 적용할 수 있게 된다. 여기서는 개발프로젝트의 규모와 일정 자원에 따른 프로젝트의 동적인 통제를 할 수 있는 수준이 된다.

4단계부터는 아키텍처와 비즈니스와의 관련성 평가 및 통제가 가능한 단계이다. 이 단계에서는 아키텍처 평가기법(Evaluation Software Architecture)과 비즈니스 모델의 연계성을 정의하는 것이 주된 목표이며 최종적으로 5단계에서는 제품계열(Product-Line) 조직이 완성되어 체계적으로 고객의 요구를 소프트웨어 제품의 품질과 생산능력에

반영할 수 있게 된다. 또한 고객의 요구에 부응하는 예측생산을 하기 위한 기술적 발전방향을 예견할 수 있게 된다.

CDMM에서 제시한 단계는 그 동안 필자의 경험을 통해 컴포넌트 기반 소프트웨어 개발기술을 확보하고자 하는 개발조직 들이 필연적으로 거치는 단계를 정의한 것이다. 다른 성숙도 모형과 마찬가지로 CDMM에서도 2단계나 3단계와 같이 더 높은 성숙도 수준의 개발능력을 일부 먼저 시작할 수는 있으나 조직 전체에 역량으로 확보되기 위해서는 위의 프로세스컴포넌트아키텍처비즈니스조직과 같은 일련의 단계를 따르게 된다는 것이다.

### 3.2 핵심기술 영역 정의

각 단계별로 달성해야 할 기술요소를 명확히 하기 위해 핵심기술영역(Key Technology Area : 이하 KTA)을 정의하였다. 소프트웨어 개발능력의 향상에 영향을 주는 기술영역을 프로세스, 아키텍처, 조직으로 나누었다. 각 기술영역은 각기 컴포넌트 기반 소프트웨어 개발에 밀접한 영향을 미치는 기술영역이지만 상호 독립적인 기술적 성숙단계를 갖는 특성을 갖고 있다.

각 핵심기술영역별로 몇 개의 달성할 핵심기술항목(Key Technology Item)을 정의하였다. 각 핵심기술항목은 독립적으로 습득해야 하는 기술요소이다.

	Process	Architecture	Organization
5단계			Dynamic Production Organization Planning Market Planning
4단계		Architecture Repository ATAM	Business Case Enterprise Modeling Analysis Pattern
3단계	Process Control Iterative Process	ABAS Architecture Restructuring Architecture Representation	
2단계	Incremental Process Formalized Process	Design Pattern Refactoring	
1단계	Analysis & Modeling	Commercial Framework	

<그림 3> CDMM 핵심기술항목 정의

프로세스 영역에서는 초기단계에는 UML 중심의 모델링 기법과 분석기법을 조직에 습득하는 것이 주된 목표이다. 이 단계가 지난 후에는 비교적 정형화된 CBD방법론을 조직에 적용하는 과정을 거치게 된다. 또한 개발과정의 정량적 목표제시와 평가를 위해 완전한 반복 프로세스 보다는 점진적

개발 프로세스를 중심으로 한 프로세스를 적용하게 된다. 프로세스 영역의 최종 목표는 완전한 반복프로세스의 적용이다. 기존의 정형화 된 산출물이나 개발 프로세스에 비해서 비교적 자유로운 반복 프로세스를 통해 조직문화와 개발자의 능력에 맞는 동적인 개발 프로세스 통제를 실시하게 된다. 두 번째로 아키텍처 영역에서는 1단계에서 상용화된 소프트웨어 아키텍처인 프레임워크에 대한 기술습득을 통해 우회적인 소프트웨어 아키텍처 기술을 습득한다. 예를 들어 J2EE나 .NET과 같은 표준 프레임워크나 응용 프레임워크를 이용하기 시작한다. 개발능력의 성숙도 향상과정으로 해석하면 아키텍처 영역에서 달성해야 하는 1단계 목표는 기존의 아키텍처에 대한 이해라고 할 수 있겠다. 그 다음에 2단계에서는 아키텍처 설계를 위한 설계패턴과 리팩토링에 대한 기술습득이 필요하다. 또한 3단계 이후에는 아키텍처의 정의와 설계 그리고 평가하는 기술항목을 필요로 한다. 아키텍처 영역의 최종목표는 해당조직의 제품계열을 구성하는 아키텍처 리퍼지토리를 구성하여 등록하는 것이다.

세번째로 조직영역에서는 4단계의 성숙도 수준부터 기술항목을 정하였다. 이는 1,2,3단계와 같이 아키텍처가 확정되지 않은 성숙수준에서는 비즈니스와 아키텍처의 동적인 상관관계를 명확히 하기가 어려움을 고려한 것이다. 조직영역의 최종 목표는 해당 소프트웨어 제품의 대상시장 분석을 통한 제품기획 및 생산이다. 조직영역에서 가장 먼저 달성해야 할 기술항목은 분석패턴과 엔터프라이즈 모델링을 포함한 고급 비즈니스 모델링이다. 이러한 비즈니스 모델링 기술을 이해함으로써 소프트웨어 아키텍처를 고려한 비즈니스 컴포넌트를 작성할 수 있게 된다. 다음은 비즈니스케이스(Business Case)를 만들고 이를 통해 시스템의 공통성을 식별하고 소프트웨어 아키텍처를 정의하는 기술을 습득하는 것이다. 개발조직이 이러한 비즈니스 케이스를 통한 커뮤니케이션 기술을 습득하는 것을 통해 비로소 컴포넌트 기반 개발기술이 정량적인 비즈니스의 통제를 가능하게 하는 것이다. 최종적으로 비즈니스 컴포넌트 설계 기술과 비즈니스 케이스를 통한 커뮤니케이션 기술 그리고 시장분석 기술을 토대로 컴포넌트 기반의 소프트웨어 개발 조직 체계를 수립할 수 있게 된다.

이상과 같이 CDMM의 핵심기술영역과 핵심기술항목에 관해 살펴보았다. 구체적인 핵심기술항목별

설명은 본 논문에서 생략하기로 한다. 다만 4단계 이상의 핵심기술항목은 실제 현재의 컴포넌트 기술의 발전수준이 미흡하여 해당수준을 달성한 개발조직을 찾을 수 없었다.

#### 4. 결론

컴포넌트 기술은 향후 소프트웨어 산업을 진정한 산업화로 전환하게 하는 기초 기술임에 틀림없다. 따라서 그 동안 다양한 측면에서 연구해 온 소프트웨어 생산기술을 총체적인 접근이 필요하며 컴포넌트 기반 개발방법론이 이에 해당한다. 이러한 이유로 국내 업체들은 보다 포괄적인 차원에서 CBD/CBSE를 접근해야 하며 CDMM은 컴포넌트 기술을 도입하여 소프트웨어 개발능력을 향상시키려는 조직에 체계적인 컴포넌트 기반 개발기술 도입을 위한 참조모형이 될 것이다.

최초 CDMM은 컴포넌트 기반 개발기술을 습득하려는 개발조직을 위해 만들어진 단계적 학습모델을 확장한 것이다. 따라서 3단계 이상의 상위 수준에서의 명확한 기술항목이 정의되지 못한 한계가 있다. 그리고 각 성숙도 수준을 측정하거나 평가할 수 있는 평가지표의 개발도 아직은 이루어지지 못하였다.

그러나 컴포넌트 기반 소프트웨어 개발기술을 향후 핵심역량으로 확보하려는 국내외 개발업체들이 늘어나고 있어 각 개발업체들간의 개발능력 및 성숙도 평가나 단계별 컴포넌트 기술 확보방안 수립에 지침이 될 수 것으로 판단한다. 아무쪼록 CDMM에 대해 보다 많은 의견과 비판이 있기를 기대한다.

#### [참고문헌]

- [1] 오영배, 나희동, 박준성, 백두권, "컴포넌트 기반개발프로세스", 한국정보과학회 소프트웨어공학회지 2002. 5.
- [2] 박준성, "CBD 도입전략", 2001. 한국소프트웨어컴포넌트컨소시엄.
- [3] OOSPICE Project Home Page, <http://www.oospice.com>
- [4] Herbsleb., J. and Zubrow, D. and Goldenson, D. and Hyes, W. and Paulk, M., Software Quality and the Capability Maturity Model, Comm. ACM vol. 40 (1997), no. 6. pp. 30--45

- [5] Humphrey, W.S., Managing the Software Process, Addison-Wesley Reading Mass. 1989
- [6] Cignoni, G.A.:Software Process Technologies and the Competitiveness Challenge, Conradi, R. (ed.): Software Process Technology, 7th European Workshop EWSPT 2000, Springer 2000, pp. 151--155
- [7] Coad, P. Object Models: Strategies, Patterns and Applications. Englewood Cliffs, NJ: Yourdon Press, 1995
- [8] Fowler, M. Analysis Patterns: Reusable Object Models. Reading, MA: Addison Wesley, 1997.
- [9] Gamma, E. et al. Design Patterns: Elements of Reusable Object-Oriented Software. Reading, MA: Addison Wesley, 1995.
- [10] Park, J.S. "A New Revolutionary Paradigm of Software Development for Mainstream Business Operations," International Journal of Technology Management Vol. 20, No. 3/4, 2000. pp. 272-286.
- [11] Carnegie Mellon University's Product Line Program Homepage, [http://www.sei.cmu.edu/programs/pls/pl\\_program.html](http://www.sei.cmu.edu/programs/pls/pl_program.html)
- [12] The Product Line Practice (PLP) Initiative, [http://www.sei.cmu.edu/plp/plp\\_init.html](http://www.sei.cmu.edu/plp/plp_init.html)
- [13] Software Architecture and the Architecture Tradeoff Analysis Initiative, [http://www.sei.cmu.edu/ata/ata\\_init.html](http://www.sei.cmu.edu/ata/ata_init.html)