

# 컴포넌트 품질 관리 프로세스 개발 사례

김길조<sup>†</sup>·장진호<sup>††</sup>·황선명<sup>†††</sup>

## 요약

소프트웨어 컴포넌트 기반 개발을 통해 시스템을 구축하기 위해서는 검증되고 표준화된 컴포넌트가 필수적이다. 본 논문에서는 컴포넌트 개발 시범사업에 적용하기 위해 개발되었던 컴포넌트 품질 관리를 위한 접근방법을 제시한다. 컴포넌트 품질 명세, 품질 계획, 품질 통제, 품질 평가의 4단계로 구성된 품질관리 프로세스는 컴포넌트 개발시 품질 목표를 설정하고 설정된 목표를 달성하도록 유도한다. 이 때 컴포넌트 품질 모형은 품질 요구사항을 측정가능한 품질 목표로 변환시키고 개발된 컴포넌트의 품질을 평가하는 근거로 사용하기 위해 제안되었다.

## A Component Quality Assurance Process and its Application

Gil-Jo Kim<sup>†</sup> · Jin-Ho Jang<sup>††</sup> · Sun-Myung Hwang<sup>†††</sup>

### ABSTRACT

Developing a component-based software requires verified and standardized software components. This paper presents a component quality management (CQM) process. The process was developed and applied to the government-sponsored trial projects that developed software components. The process is composed of four phases : quality specification, quality planning, quality control, and quality evaluation. With this process, we can establish quality goals and focus our efforts on the activities to achieve the goals. A component quality model is also suggested to transform the implicit quality requirements into the measurable quality goals and to be used for the basis when we evaluate the quality of software components against the quality goals.

**키워드 :** 소프트웨어 컴포넌트(software component), 컴포넌트 품질 모형(component quality model), 품질관리 프로세스(quality management process)

### 1. 서론

대부분의 소프트웨어는 품질 미비, 납기 및 비용 초과 등의 문제를 가지고 개발된다. Standish Group[1]의 보고서에 따르면 31.1퍼센트의 개발 프로젝트가 종료 이전에 취소되며, 나머지 프로젝트의 52.7퍼센트는 최초 비용 예측치보다 189퍼센트의 초과비용이 지출되었다는 연구결과는 이러한 사실을 잘 설명해 준다.

소프트웨어 컴포넌트 기술의 등장은 이러한 소프트웨어의 위기를 극복하는 하나의 대안으로 생산성을 높이고 품질을 향상시킬 수 있는 가능성으로 인정받고 있다. 소프트웨어 컴포넌트(이하 컴포넌트)는 컴포넌트 모형을 따르면서 추가적인 변경 없이 독립적으로 생산되고 구매되고, 배치될 수 있는 소프트웨어 요소들을 말한다[2]. 표준화된 컴포넌트 모형으로서 OMG(Object Management Group)의 CORBA(Common Object Request Broker Architecture), MicroSoft사의

COM(Component Object Model) 및 OLE(Object Linking and Embedding), Sun사의 EJB(Enterprise Java Beans) 등이 제시되었다.

컴포넌트 기반 개발(Component Based Development, 이하 CBD)은 표준화되고 검증된 컴포넌트를 활용하여 소프트웨어 시스템을 빠른 시간에 구축하기 위한 것이다. 따라서 CBD에서는 조직의 개발능력의 중요정보보다는 구매 중심 모형의 필요성을 더욱 강조된다[3]. 이러한 컴포넌트의 개발 및 활용을 지원하기 위한 다양한 도구들과 CBD 방법론 등이 개발되었으며, 컴포넌트 기술은 금융시스템, 생산시스템, 통신 시스템 등 다양한 분야에 적용되고 있다.

그러나 컴포넌트의 특성 및 컴포넌트 프레임워크 시스템 특성을 예측하는데 사용될 수 있지만, 컴포넌트가 복잡하거나 다른 컴포넌트를 채택하여 개발된 경우 시스템 특성을 예측하는 것은 매우 어렵다[4]. 따라서 시스템의 특성을 예측하기 위해서는 컴포넌트 개발 프로세스 특성을 포함시키는 것이 바람직하다[5]. 비록 여러 CBD 방법론[6-9]이 제안되었고 일부는 컴포넌트의 품질 보증을 강조하고 있지만 [10], 컴포넌트의 품질 목표를 세우고 그 목표를 달성한다는 측면에서는 부족한 부분이 많다.

\* 본 연구는 한국과학재단 목적기초연구 (R01-2001-00343)지원으로 수행되었음.

† 정희원 : 한국전자통신연구원 컴퓨터소프트웨어기술연구소 선임연구원

†† 정희원 : 한국전자통신연구원 컴퓨터소프트웨어기술연구소 책임연구원

††† 종신희원 : 대전대학교 컴퓨터공학과 교수

논문접수 : 2001년 10월 5일, 심사완료 : 2001년 12월 5일

본 논문에서는 컴포넌트의 품질 관리를 위한 프로세스를 제시한다. 컴포넌트로 개발된 소프트웨어 시스템은 전통적인 소프트웨어와는 다른 프로세스에 따라 개발된다. 따라서 품질 관리 프로세스에는 컴포넌트 개발을 위한 프로세스와 전체 시스템을 위한 프로세스 모두를 포함시켜야 한다. 그러나 본 논문에서는 컴포넌트 개발을 위한 품질 관리 프로세스만을 제시한다. 전체 시스템의 품질 보증은 전통적인 소프트웨어 품질 보증 차원에서 다룰 수 있기 때문이다.

본 논문에서 제시하는 품질관리 프로세스는 컴포넌트 개발 시범사업에 적용된 것으로 개발 과정에서 컴포넌트의 품질 목표를 설정하고 이러한 목표를 달성하도록 품질 활동을 수행하도록 유도한다. 2장에서는 컴포넌트 품질 보증 요구사항 및 접근방법을 설명하고 있으며, 3장에서는 컴포넌트 품질모형을 제시한다. 4장에서는 컴포넌트 품질 관리 프로세스를 설명하고 있으며, 5장에서 결론을 맺는다.

**2. 컴포넌트 품질 관리 요구사항 및 접근방법**

한국전자통신연구원 주관의 컴포넌트 개발 시범사업이 2000년도에 수행되었다. 시범사업을 통해 수요가 많은 컴포넌트를 개발하여 보급함으로써 컴포넌트 산업을 활성화 할 수 있는 가능성을 타진해 보기 위한 것이었다. 시범사업은 다음 <표 1>와 같이 4개 분야에 걸쳐 이루어졌으며 각 분야 별로 서로 다른 개발업체에 의해 수행되어 총 28개의 컴포넌트가 개발되었다. EJB로 개발된 각 컴포넌트는 1개 이상의 세션빈(session bean) 또는 엔터티빈(entity bean)으로 구성되며, 각 빈(bean)은 하나 이상의 클래스(class)로 구현되었다.

<표 1> 시범 사업의 컴포넌트 개발 요약

분야	컴포넌트 수	빈(Bean) 수		클래스 수
		세션빈	엔터티빈	
XML/EDI 서버 공통 컴포넌트	12	3	11	47
B2B가상시장 공통 컴포넌트	6	7	27	8
B2C 쇼핑몰 공통 컴포넌트	5	5	7	51
비즈니스 공통 컴포넌트	5	4	12	60
총 계	28	19	57	166

시범사업을 통해 개발되는 컴포넌트의 품질 관리를 위한 프로세스는 다음과 같은 요구사항을 수용하여야 한다.

- 주관기관(한국전자통신연구원)과 개발업체의 역할이 명확히 구분되어야 한다.
- 주관기관은 개발업체에게 컴포넌트가 달성해야 할 측정가능한 품질 목표를 제시하여야 하고 이 목표에 비추어 평가를 하여야 한다.
- 시범사업에서 컴포넌트는 단기간에 개발되므로 품질 관리 단계는 간결한 형태로 제시될 필요가 있다.
- 개발이 끝난 컴포넌트가 품질 목표의 달성여부를 평

가받기 위해 개발업체는 필요한 산출물을 제출하고, 주관기관은 품질 평가를 수행하여야 한다.

이러한 요구사항을 충족시키기 위해 본 논문에서 제시하고 있는 품질관리 프로세스는 다음의 두 가지 접근방법을 통해 개발되었다.

(1) 국제표준에 근거한 컴포넌트 품질 평가 모형의 제시  
 컴포넌트에 대한 품질 평가는 ISO/IEC 9126[11]과 같은 품질 모형에 근거하여야 한다. 그러나 컴포넌트의 특성을 고려할 때 컴포넌트 품질모형은 ISO/IEC 9126에 정의되어 있는 품질 속성(quality attributes)들의 의미를 달리 해석할 필요가 있다. Woodman[12] 등은 컴포넌트 사용자(즉, 컴포넌트 기반 시스템 설계자)를 고려하여 11개의 CBD 품질 속성으로 재사용성(reusability), 유지보수성(maintainability), 정확성(accuracy), 명확성(clarity), 대체가능성(replaceability), 상호운영성(interoperability), scalability, 성능(performance), 유연성(flexibility), 적응성(adaptability), 신뢰성(reliability)을 제시하고 있다. 이 중 명확성, 유연성, 재사용성, scalability는 ISO/IEC 9126에 정의되어 있지 않아 추가적인 정의 및 해석이 필요하다. 컴포넌트 품질 모형은 ISO/IEC 9126을 소프트웨어 패키지의 평가를 위해 제시된 ISO/IEC 12119 [13]를 고려하여야 한다. 소프트웨어 패키지가 최종산출물을 고객에게 인도한다는 측면에서 컴포넌트와 비슷하며, 향후 국제표준화의 방향 또한 ISO/IEC 12119에서 컴포넌트의 평가를 포함하도록 하고 있기 때문이다.

(2) 품질평가에 초점을 맞춘 컴포넌트 품질관리 활동 개발  
 소프트웨어 품질관리 프로세스 또는 프로젝트의 작업산출물 및 프로세스가 명시된 요구 사항에 부합하고 수립된 계획에 따르고 있음을 관리하기 위한 프로세스이다. 컴포넌트 품질 관리 프로세스는 구체적인 품질 관리 활동을 제시하여 개발자에게 제공하여야 하며, 각 활동 및 작업산출물은 적절한 시점에서 검토되어야 한다. 이러한 검토는 개발자 또는 별도의 조직에 의해 수행되는 것이 바람직하다. 그러나 시범 프로젝트가 짧은 기간에 이루어져야 하며, 개발자는 별도의 품질관리 조직을 운영하지 않고 주관기관이 품질관리의 많은 부분을 담당하여야 한다. 또한 개발된 컴

<표 2> 컴포넌트 품질관리 프로세스와 ISO/IEC 15504 품질관리 프로세스 단계의 비교

컴포넌트 품질관리 프로세스 단계	SPICE의 품질보증 프로세스의 관련 활동
1. 품질 명세 : 컴포넌트의 품질요구사항을 품질달성목표로 표현한다.	1. 품질보증 전략 개발 2. 품질 표준 수립
2. 품질 계획 : 품질달성목표를 수행하기 위한 품질활동을 계획한다.	3. 품질기록 정의
3. 품질 통제 : 품질활동 및 품질목표에 비추어 품질 활동을 통제한다.	4. 프로세스 활동 품질보증 5. 작업산출물 품질보증
4. 품질 평가 : 개발된 컴포넌트를 품질 목표에 비추어 평가한다.	6. 품질결과 보고 7. 부적합 사항 처리

포넌트는 반드시 품질 평가를 위해 통해 품질을 검증 받도록 하고 있다. 이러한 상황을 고려하여 품질 관리 프로세스는 품질 평가를 위한 핵심 단계만으로 구성된 프로세스의 형태가 필요하였다. 다음 <표 2>는 본 논문에서 제시하고 있는 컴포넌트 품질관리 프로세스의 단계와 ISO/IEC 15504 표준[14]에서 제시하고 있는 활동을 비교한 것이다.

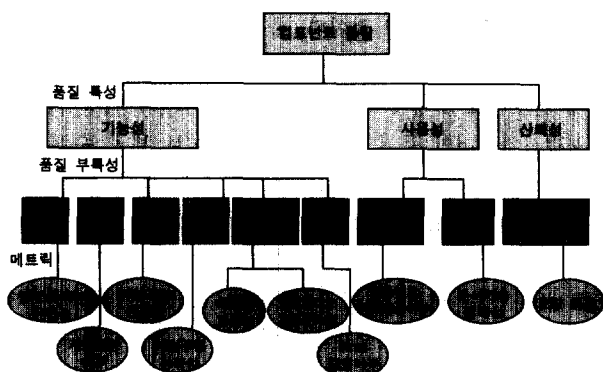
컴포넌트 평가를 위해 컴포넌트 개발자는 다음 <표 3>과 같은 구성항목으로 된 시험패키지를 제출하도록 하였다. 시험패키지는 컴포넌트 품질평가에 필요한 정보를 제공할 뿐 아니라 시험패키지 자체의 품질을 평가하기 위해 사용된다.

<표 3> 시험패키지의 구성항목

구성항목	설명
컴포넌트 설치 설명서	컴포넌트가 설치되기 위한 환경에 대한 정보와 설치를 위한 작업에 관한 설명
클래스파일 및 자바 소스 코드	EJB 빈을 이루고 있는 클래스 파일
Deployment Descriptor	컴포넌트를 어플리케이션 서버에 설치하기 위해 필요한 내용에 대한 설명
컴포넌트 설명서 및 개발문서	컴포넌트 사용자 매뉴얼, 컴포넌트 기능 설명서 등의 사용자 참고 문서
시험 설명서	제삼자가 시험을 수행할 수 있도록 시험에 필요한 환경을 구성하는 방법과, 시험 데이터 별로 시험을 수행할 수 있는 시험의 절차에 대한 설명
주요 기능 요약서	시험 대상이 되는 컴포넌트의 주요 기능을 시험 데이터와의 연관성을 통해 시험 데이터 별로 시험하게 되는 컴포넌트의 기능을 알아볼 수 있도록 설명한 것
시험 오라클	시험 데이터를 입력으로 시험을 수행하였을 경우 예상되는 출력에 대한 설명
도메인에 대한 동치영역 분석표	개발자를 위한 컴포넌트 시험 지침에서 제안하고 있는 동치 영역 분할 기법을 적용한 동치영역 분석표
시험 드라이버	개발자가 시험에 사용한 시험 드라이버
시험 스텝	개발자가 시험에 사용한 시험 스텝

### 3. 컴포넌트 품질 모형

본 논문에서 제시하고 있는 (그림 1)의 컴포넌트 품질 모형은 ISO/IEC 9126과 같이 품질특성-부특성-메트릭의 체계로 구성되어 있다.



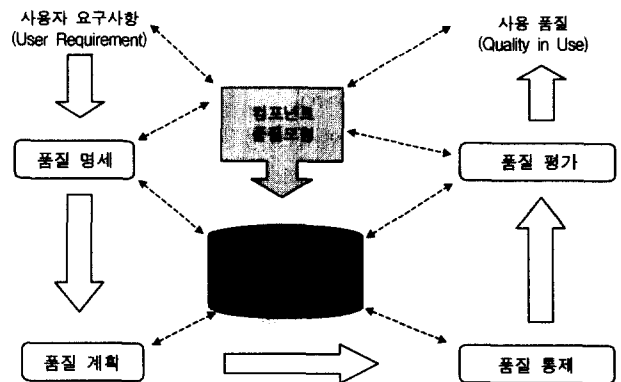
(그림 1) 컴포넌트 품질 모형의 구성요소

위 그림에서 보는 바와 같이 본 컴포넌트 모형은 ISO/IEC 9126에서 제시하고 있는 6개의 품질특성 중 기능성, 사용성 및 신뢰성은 포함시켰지만 효율성, 유지보수성, 이식성은 시간제약 및 효율적인 평가방법의 부재 등에 의해 제외시켰다. 매트릭은 컴포넌트의 품질을 평가하기 위한 구체적인 방법을 제공하는 것으로 본 모형에서는 10개의 매트릭만을 정의하였다. 각 매트릭은 ISO/IEC 14598-6[15]에서 정의된 평가모듈(Evaluation Module)로 정의되었다[16]. 부록에서는 기능구현완전성 매트릭에 대한 평가모듈의 예를 보여준다.

컴포넌트 품질 평가는 컴포넌트 문서(component documents), 컴포넌트 프로그램(component program)를 대상으로 이루어진다. 컴포넌트 문서(Component documents)는 컴포넌트의 속성을 설명하는 문서 또는 컴포넌트의 사용을 위해 제공되는 문서들로서 그 자체가 제품의 중요한 부분이다. 시험사업 평가를 위한 컴포넌트 문서는 시험패키지이다. 컴포넌트 프로그램(Component program)은 추가적인 컴파일 없이 사용될 수 있는 컴포넌트 프로그램으로서 다른 컴포넌트와 결합하여 시스템을 구성하는데 사용되는 것으로 일반적으로 소스코드를 포함하지 않는다.

### 4. 컴포넌트 품질 관리 프로세스 및 그 적용

본 논문에서 제시하고 있는 품질관리 프로세스는 (그림 2)와 같이 품질 명세, 품질 계획, 품질 통제 및 품질 평가의 4 단계로 구성되어 있다. 이 때 컴포넌트 품질모형은 품질을 명세하고 품질을 평가하는 근거로 활용된다. 경험베이스는 개발과정에서 축적되는 경험을 향후 활용할 수 있도록 이를 조직화하기 위한 제안된 것으로 실제 구현되지는 않았다.



(그림 2) 컴포넌트 품질모형

품질관리 프로세스의 각 단계 중 품질 명세, 품질 평가는 주관기관에서 담당하고, 품질 계획 및 품질 통제는 개발업체에서 주로 담당하였다. 각 단계에 대한 자세한 설명은 다음과 같다.

#### 4.1 품질 명세(Quality Specification)

컴포넌트 품질명세 단계에서는 사용자의 요구(needs)와

컴포넌트 품질 모형에 기반 하여 컴포넌트의 품질 목표가 정의하고 이를 품질 명세로 표현한다. 컴포넌트 품질 요구 사항은 컴포넌트 품질 프로파일의 형태로 나타낼 수 있다. 품질 요구사항은 컴포넌트 품질 모형을 기반으로 수립된다. 이러한 요구사항에 근거하여 컴포넌트의 품질 달성 목표(targets)가 수립된다. 이 때 품질 목표는 측정 가능한 형태로 표현되는 것이 바람직하다.

이 때 목표가 도달가능하지는 과거 프로젝트 수행 결과를 고려하여 평가되는 것이 바람직하고, 달성 불가능한 도 달목표는 수정되어야 한다. 이러한 분석 및 의사결정 프로 세스는 문서화되어 경험 베이스에 저장할 수 있다.

시험사업에서 개발된 컴포넌트는 다음 <표 4>과 같은 품질 달성 목표를 설정하였다. 그러나 컴포넌트의 신뢰성은 그 구체적인 평가방법이 가용하지 않아 제외시켰다. 이 품질 달성목표는 컴포넌트 품질모형의 매트릭으로 쉽게 표현 할 수 있다.

<표 4> 시험사업의 컴포넌트 품질 목표(가능성)

품질 요구사항	품질 달성 목표
컴포넌트로써 구성요소를 갖출 것	컴포넌트 사용에 필요한 정보를 시험패키지에서 제공함 시험패키지의 내용이 시험평가에 사용할 수 있도록 충분함
컴포넌트가 안전하고 설치가 용이할 것	컴포넌트가 바이러스에 감염되어 있지 않음 사용자가 직접 운영 환경에 설치/제거할 수 있음
컴포넌트 기능이 제대로 구현될 것	컴포넌트 문서에 기능이 제대로 설명되어 있음 컴포넌트 문서에 명시된 기능이 모두 구현됨 컴포넌트 문서에 명시된 기능이 모두 정확하게 구현됨
컴포넌트가 상호 운영성을 만족할 것	컴포넌트가 인터페이스 표준을 준수하여 구현됨 입출력 데이터 표준이 명시된 경우 이를 준수 하여 구현됨

4.2 품질 계획(Quality Planning)

컴포넌트 품질 계획 단계에서는 적절한 컴포넌트 개발 프로세스를 결정하고 개발 프로세스 내에서 도달 목표값을 설정하는 것을 포함한다. 컴포넌트 품질 달성 목표와 근거하여 개발 프로세스에서 측정하여야 할 매트릭의 집합들이 이 단계에서 정의되고, 측정 계획이 만들어진다. 측정 계획에는 각 매트릭에 대해 다음 사항들이 반드시 포함되어야 한다.

- 매트릭의 공식적인 정의
- 매트릭에 대한 설명
- 매트릭 값의 범위
- 데이터를 수집하는 사람(엔지니어, 관리자 등)의 역할
- 데이터가 수집 시기
- 데이터가 수집되는 수단(도구, 데이터 수집폼)

예를 들어 컴포넌트 개발시 컴포넌트의 기능점점을 위한 공식검토 시점을 사전에 정의하고, 검토시 발견되는 결함의 수를 매트릭으로 사용할 수 있다. 이러한 매트릭은 사전에

정의되어 품질 통제 단계에서 활용할 수 있도록 준비한다.

시험사업에서는 특정 개발프로세스가 채택되지 않았기 때문에 개발업체 나름대로의 품질계획단계를 수행하였다.

<표 5> 가능성에 대한 매트릭스

매트릭	측정공식	의미	척도	측정형태
기능적 이해능력	명세서 대비 구 기능 구현율 $X = A/B$	$0 \leq x \leq 1$ 1.0에 가까울 수록 좋음.	X = 절대치	A = 실행된 기능수 B = 명세서에 기술된 기능수
기능적 안정성	운영이후 수정이 필요한 비율 $X = 1 - (A/B)$	$0 \leq x \leq 1$ 1.0에 가까울 수록 좋음.	X = 절대치	A = 운영중 오류 발견 기능수 B = 명시된 기능수 X = 개수/크기

4.3 컴포넌트 품질 통제(CQC : Component Quality Control)

컴포넌트 품질 통제 단계는 컴포넌트 개발 프로세스 내 중요 검토 시점에서 중간산출물 및 활동과 관련된 측정을 통해 프로세스를 감시하는 것이다. 컴포넌트가 구현되는 동안 측정 계획에 따라 매트릭이 수집되면 이를 분석하여 필요한 통제를 하는 것을 포함한다. 예를 들어 컴포넌트 품질 달성 목표와 실제 측정값을 비교한 차이를 분석하여 문제 영역을 식별하고 필요한 조치를 취할 수 있다. 다음 <표 6>은 특정 컴포넌트의 기능의 구현여부와 구현 정확성을 기록한 예로서 시험사례를 보완하거나 코드 재작성 등이 필요하다는 조치 내용을 보여준다.

<표 6> 컴포넌트의 기능 구현 현황 및 조치의 예

기능 id	기능	구현 여부	구현정확성	조치 내용
F1-1	addAuction	O	X	시험사례 보완
F1-2	modifyAuction	O	X	시험사례 보완
F1-3	deleteAuction	O	X	시험사례 보완
F1-4	getAuctionInfo	O	O	-
F1-5	getAuctionInfo	O	X	코드 재작성
F1-6	getSuccessfulBidInfo	O	X	코드 재작성
F1-7	checkRemainsTime	O	O	-
F1-8	decideWinner	O	X	시험사례 보완
F1-9	decideLoser	O	X	시험사례 보완

4.4 컴포넌트 품질 평가(CQE : Component Quality Evaluation)

컴포넌트 품질 평가 단계의 목적은 개발된 컴포넌트의 품질을 평가하기 위한 것이다. 품질 평가는 각 컴포넌트별로 이루어진다. 이 때 문서 검토 등을 통한 평가 활동 및 실제 컴포넌트 수행을 통한 평가활동이 동시에 이루어진다.

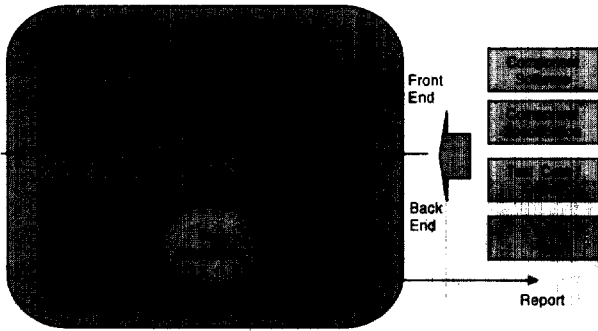
시험사업에서의 품질 평가는 각 컴포넌트 별로 수행되었다. 다음은 시험사업의 한 컴포넌트의 품질 평가 결과를 일부 요약한 것이다.

컴포넌트의 수행을 통한 확인을 위해 (그림 3)과 같은 EJB 컴포넌트 시험환경이 구축되었다. EJB 애플리케이션 서버 상에서 구현된 시험환경에서는 단일 컴포넌트로써 수행가능하도록 필요한 테스트 드라이버와 스텝을 사용하여 컴포넌트를 수행할 수 있도록 지원한다[17].

〈표 7〉 시험사업 컴포넌트 평가 결과의 예

메트릭(값 유형)	목표 대비 실제 측정값	
	목표	실제
Virus 부재(Yes/No/N.A)	목표	Yes
	실제	Yes
설치 및 제거 가능성(Yes/No/N.A)	목표	Yes
	실제	Yes
기능 완전성(%) (구현 기능수/전체 기능수)	목표	100%
	실제	100%(4/4)
기능 정확성(%) (정확하게 구현된 기능수/전체 기능수)	목표	100%
	실제	75%(3/4)
인터페이스 준수성	목표	N.A
	실제	N.A
입출력데이터 표준준수성(Yes/No/N.A)	목표	Yes
	실제	Yes

(N.A : 적용대상이 아님)



(그림 3) EJB 컴포넌트 시험 환경의 구조

프로세스의 검토 또한 품질평가 단계에서 이루어진다. 이를 통해 데이터와 경험 등을 수집하고 분석하여 적용된 프로세스가 적절한 것인지를 검토하고 프로세스를 보완하는데 활용한다.

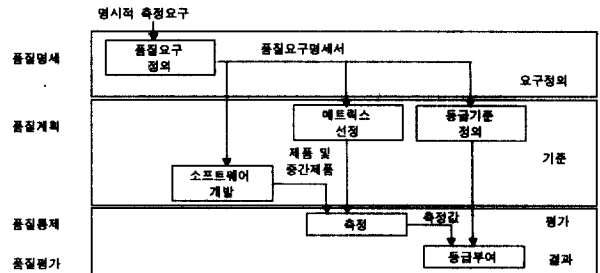
4.5 경험베이스(EB : Experience Base)의 필요성

컴포넌트 품질 개선을 위한 본 접근방법은 지속적인 경험, 데이터의 활용을 통해 보다 효율적으로 적용할 수 있다. 경험베이스에는 이러한 경험 및 데이터가 향후 활용을 위해 조직화되어 저장된다. 특정 CBD 프로세스를 고려하여 컴포넌트 품질 개선을 위한 본 접근방법을 지속적으로 적용할 경우, 컴포넌트 품질 속성 및 CBD 프로세스간의 관련성을 정의할 수 있고, 이를 경험베이스에 저장할 수 있다. 그 결과 특정 컴포넌트 품질 속성을 개선하기 위해서 관련된 프로세스를 찾아 노력을 집중할 수 있다. 이러한 경험베이스의 구축은 많은 시간이 요구되며 지속적으로 갱신될 필요가 있다. 컴포넌트 시험사업에는 특정 CBD 프로세스가 적용되지 않았으며, 평가는 1회에 걸쳐 이루어졌기 때문에 경험베이스를 구축하지 못하였다.

5. 결 론

본 논문에서는 컴포넌트 품질 관리를 위한 프로세스로서 개발 대상 컴포넌트의 품질 요구사항을 품질 목표로 설정하

고, 이를 달성하기 위해 컴포넌트 품질 명세, 품질 계획, 품질 통제, 품질 평가의 4단계 접근방법을 제시하였다. 그리고 이 프로세스를 컴포넌트 시험개발 사업에 활용한 예를 보여주고 있다. 또한 국제표준을 기반으로 하여 본 프로세스가 적용될 수 있는 근거가 되는 컴포넌트 품질 모형을 제시하였다.



(그림 4) 품질관리 프로세스와 관리 활동

이 컴포넌트 품질모형과 품질관리 프로세스는 시험사업을 통해 개발된 컴포넌트에 적용되었으며 특히 품질평가에 초점을 맞추어 품질관리 활동이 이루어졌다. 본 접근방법은 컴포넌트가 특정 CBD 프로세스를 통해 개발될 경우 품질 목표를 달성하기 위해 특정 측면에 초점을 맞추어 노력을 기울일 수 있도록 한다. 이 때 경험베이스는 성공적인 컴포넌트 품질 개선을 위한 핵심적인 부분을 담당할 수 있기 때문에 향후 실제 사례의 수집 및 축적을 통한 경험베이스의 구축 및 활용이 요구된다.

본 연구는 제안된 품질관리 프로세스를 통해 컴포넌트의 품질 명세에서부터 각 품질 목표를 이루기 위한 하부 특성의 메트릭스 값을 구하는데까지 진행중이며, 이들의 등급별 구분을 통하여 컴포넌트의 품질을 보증하고 계량적인 점수를 통한 인증에 대한 연구는 많은 실험데이터를 수집하고 분석하는 과정에 있다(그림 4).

[부록 1] 평가모듈로 작성된 기능구현완전성 메트릭의 예

1. 개요

이 평가모듈은 컴포넌트 설명서에 기술되어 있는 기능이 구현되어 있는지 평가하기 위한 것이다. 컴포넌트 설명서에 언급된 모든 기능은 관련 설비, 속성 및 데이터와 함께, 그리고 주어진 범위 값 내에서, 컴포넌트 설명서에서 주어진 형태로 실행될 수 있어야 한다. 평가해야 하는 기능은 문서에 기술되어 있는 것을 평가해야 한다.

2. 적용 범위

2.1 특성

기능성 - 완전성 - 기능구현 완전성

2.2 기법

해당사항 없음

3. 참조 문서

- ISO/IEC12119 3.3.1
- ISO/IEC9126-2 8.1.1

4. 입력물과 매트릭

4.1 입력물

컴포넌트 설명서, 사용자문서

4.2 데이터항목

데이터항목	평가 대상
컴포넌트 설명서에 명시된 기능의 수(A)	문서
평가 단계에서 발견한 빠진 기능의 수(B)	프로그램

4.3 매트릭

매트릭 이름	수식 및 측정유형
기능구현 완전성	유형 III (X = A / B)

5. 결과 해석

5.1 측정치의 매핑

$0 \leq$  기능구현의 완전성  $\leq 1$

5.2 보고사항

- 1) 기능구현 완전성 :
- 2) 누락된 기능 항목

6. 적용 절차

6.1 필요 자원

기능 점검표, 기능 테스트 도구

6.2 상세 절차

- 1) 문서에 기술된 기능항목을 식별한다. 미리 작성된 기능 점검표가 존재하면 이를 활용할 수 있다.
- 2) 각 기능 항목이 프로그램에 구현되어 있는지 확인한다.
- 3) 4.3의 매트릭을 이용하여 측정한다.
- 4) 누락된 기능항목을 기록하여 보고한다.

참 고 문 헌

[1] Standish Group, Chaos Report, 1995.  
 [2] B.Council, "Definition of a Software Component and Its Elements," in Component based software engineering edited by B. Council, 2001.  
 [3] Jeffrey Voas, "Maintaining Component-based Systems," IEEE Software, July/August, 1998.  
 [4] F. Bachman et al., "Volume II : Technical Concepts of Component Based Software Engineering," SEI, CMU/SEI-2000TR-008, <http://www.sei.cmu.edu/pub/documents/00-reports/pdf/00tr008.pdf>, May, 2000.  
 [5] 웹페이지 <http://www.oospice.com>.  
 [6] Keith Short, "Component based development and object modeling," Sterling Software, 1997.  
 [7] Princeton Softech, "Component based development-a roadmap to eBusiness success," Princeton Softech, 2000.  
 [8] D. D'Souza, A. C. Wills, "Objects, Components, and Frameworks with UML," Addison-Wesley, 1998.  
 [9] 한국전자통신연구원, "마르미 III-컴포넌트 기반 시스템 개발 방법론", 기술문서, 한국전자통신연구원, 2001.  
 [10] X. Cai, M. R. Lyu, K. Wong, "Component-based Software Engineering : Technologies, Development frameworks, and Quality Assurance Schemes," in Proceedings APSEC 2000, Seventh Asia-Pacific Software Engineering Conference, Singapore, December, 2000.  
 [11] ISO/IEC 9126-1. Information technology Software quality characteristics and metrics Part 1 : Quality characteristics

and sub-characteristics. ISO/IEC, 1998.

[12] M. Woodman et al., "Issues of CBD Product Quality and Process Quality," in the proceeding of 4th ICSE Workshop on Component-Based S/W Engineering, 2001.  
 [13] ISO/IEC 12119, Information technology Software packages Quality requirements and testing. Geneva, ISO/IEC, 1994.  
 [14] ISO/IEC 15504, Information technology Software process assessment Part 5 : An assessment model and indicator guidance, ISO/IEC, 1998.  
 [15] ISO/IEC FDIS 14598-6. Information technology Software Product Evaluation Part 6 : Documentation and Evaluation Module, ISO/IEC, 1999.  
 [16] 한국전자통신연구원, "소프트웨어 컴포넌트 시험을 위한 평가 모듈", 기술문서, 한국전자통신연구원, 2000.  
 [17] Gil-Jo Kim, In-Geol Chun, Ja-Kyong Koo, Su-Jung Ha, Jin-Ho Jang, Jin-Sam Kim, Development of an Automated Tool for Software Component Testing, ICACT 2001, Muju, Korea 2001.  
 [18] "EJB 컴포넌트 시험 평가 체계 및 시험 환경 구축에 관한 연구", 정보처리학회지 제19권 제2호, 통권 제141호, 2001.  
 [19] J. M. Voas, "Certifying Off-The-Shelf Software Components," IEEE Computer, June, 1998.



김 갈 조

e-mail : kgj@etri.re.kr

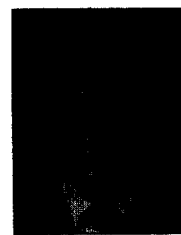
1987년 서울대학교 산업공학과 (공학사)  
 1989년 한국과학기술원 경영과학과(공학사)  
 1989년~현재 한국전자통신연구원, 컴퓨터.  
 소프트웨어기술연구소 선임연구원  
 관심분야 : S/W개발방법론, S/W품질보증,  
 S/W개발환경 등



장 진 호

e-mail : jhjang@etri.re.kr

1976년 서울대학교 화학공학과 학사  
 1978년 KAIST 화학공학과 석사  
 1978년~1982년 국방과학연구소  
 1983년~1985년 (주)한국화학  
 1985년~1998년 시스템공학연구소  
 1998년~현재 ETRI 책임연구원  
 관심분야 : 소프트웨어공학, S/W 컴포넌트 표준, e-Business,  
 영역 기반 Architecture Model



황 선 명

e-mail : sunhwang@diu.ac.kr

1982년 중앙대학교 전자계산학과  
 졸업(학사)  
 1984년 중앙대학교 대학원 전자  
 계산학과 졸업(이학석사)  
 1987년 중앙대학교 대학원 전자  
 계산학과 졸업(이학박사)  
 1988년 독일 Bonn대학 Informatik III post doctor  
 현재 대전대학교 컴퓨터공학과 부교수  
 관심분야 : 소프트웨어 테스팅, 소프트웨어 재공학, 역공학, 품질  
 보증, 표준화