

단기 개발 공공 프로젝트를 위한 효과적인 산출물 관리기법 연구

원달수

배화여자대학 컴퓨터정보과 교수

e-mail : dsweon@baewha.ac.kr

A Study on Effective Artifacts Management Methodology for Short-Term Public Sector Development Projects

Dal-soo-Weon

Dept. of Computer Science, Baewha Women's Colledge

요 약

최근 소프트웨어 개발 기술이 발전함에 따라 사용자들의 관심이 높아지면서 컴포넌트 기반의 개발방법이 빠르게 도입되고 있다. 사용자의 서비스에 대한 요구가 다양해지면서, 다양한 요구를 소프트웨어에 반영하여 단기간에 개발해야 하는 경우가 점차 늘어나고 있으며 산출물의 수도 점차 증가하고 있는 추세이다. 그러나 현실적으로 사업 유형별, 사업 주체별, 그리고 적용 방법론 별로 납품해야 하는 산출물이 서로 다르고, 산출물의 내용 역시 개발목적과 관리목적에 따라 다양하므로 이러한 내용들을 표준화하기가 쉽지 않은 것이 현실이다. 이런 점을 고려하여 본 논문에서는 각 방법론에 대한 산출물들을 비교 정리하여 향후 단기 개발 공공 프로젝트에서 산출물 선정 및 관리를 효율적으로 할 수 있는 가이드라인을 제시하고자 한다.

1. 서론

컴포넌트 기반 개발(CBD; Component Based Development) 방법중 현재 국내에서 범용적으로 사용되고 있는 RUP(Rational Unified Process)나 마르미III(MarMI III)와 같은 방법론은 장기 개발에 적합한 Heavy Weight 프로세스로서, 산출물과 정의된 작업이 방대하여 단기 개발 프로젝트에 적용 하기에는 어려움이 있다. 또한 개발과 별 개의 산출물 작업, 복잡한 작업 단계로 인한 작업 파악시간의 증가 등이 문제가 되고 있어서, 단기 개발 프로젝트에 적용하기 위해서는 적절한 커스터마이징이 요구되고 있다. 공공사업부문에 있어서 현실적으로는 사업 유형별, 사업 주체별 그리고 적용방법론 별로 납품해야 하는 산출물이 서로 다르고, 산출물의 내용 역시 개발 목적과 관리 목적에 따라 다양하므로 이러한 내용들을 표준화하기가 쉽지 않은 것이 현실이다.

따라서 본 논문에서는 공공사업에서 사용되고 있는 객체 지향 개발방법론의 절차와 산출물의 유형을 비교·정리하여 공공사업의 단기 개발 프로젝트 추진시에 필요한 산출물에 대한 기본적인 가이드라인을 제시하고자 한다. 본 논문의 구성은 다음과 같다.

2장에서는 관련 연구로써 ISO/IEC 12207의 개발공정(Proc- ess)에 대한 연구를 바탕으로 RUP(Rational Unified Proce- ss), 마르미III(MarMI III), Computer Associate사의 PPC(Pl atinum Process Continuum)에 대하여 연구하였다. 3장에서는 공공사업의 단기 개발 프로젝트를 위한 산출물 선정 및 관리 에 대한 연구로써 개발환경 및 프로젝트 적용 문제점 분석, 개발방법론의 비교, 방법론별 산출물 현황 분석, 개발 프로 세스 정의, 개발 산출물 정의, 개발 산출물간 연관성, 개발 산출물과 관련 기법, 공공사업의 단기 개발 프로젝트를 위한 산출물 목록에 대해서 각각 기술하였다. 4장에서는 3장에서 기술한 공공사업의 단기 개발 프로젝트를 위한 산출물 선정 및 관리에 대하여 간단한 사례연구를 통한 평가결과를 정리

하였다. 5장에서는 본 연구의 결론과 향후 연구 과제에 대해 기술하였다.

2. 관련연구

2.1. ISO/IEC 12207 의 개발 공정(Process)

개발 활동은 소프트웨어 개발이 진행되는 동안에 수행 되는 활동(activity)이나 작업(task)을 의미한다. 각 방 법론에서는 제각기 다른 형태로 개발활동들을 정의하고 있다. 그러나 대부분의 개발방법론들은 소프트웨어의 개발단계를 크게 분석, 설계, 구현, 테스트로 동일하게 분류하고 있으며, 개발 각 단계의 세부 활동이나 작업은 방법론의 특성에 따라 각기 다르기 때문에 프로젝트 수행기관에 따라 표준이 정해지기도 한다. ISO/IEC12207 이나 MIL-STD-498 이 대표적인 경우이다.

ISO/IEC 12207 에서의 개발 공정은 <표 1>과 같다.

<표 1> ISO/IEC 12207 의 개발활동 및 작업

활 동	작 업
시스템 요구사항 분석	-시스템 요구사항 기술 및 분석 -시스템 요구사항 평가
시스템 구조설계	-시스템 개략 설계 -시스템 구조 및 항목 평가
소프트웨어 요구사항분석	-소프트웨어 요구사항 기술 및 분석 -소프트웨어 외부 인터페이스 분석 -데이터 정의 및 데이터베이스 요구사항 분석 -소프트웨어 요구사항 평가 -합동 검토 수행
소프트웨어 구조 설계	-소프트웨어 컴포넌트 정의 -인터페이스 설계 -데이터베이스 설계 -사용자 문서 개발 -소프트웨어 통합 계획 및 요구사항 정의

	-소프트웨어 구조 설계 평가
소프트웨어 상세 설계	-소프트웨어 컴포넌트 상세 설계 -인터페이스 상세 설계 -데이터베이스 상세 설계 -사용자 문서 갱신 -단위 테스트 요구사항 및 일정 수립 -소프트웨어 통합계획 및 요구사항 재정의 -소프트웨어 상세설계 및 테스트 요구사항 평가 -합동검토 수행 -단위 소프트웨어 및 데이터베이스 개발 -단위 소프트웨어 및 데이터베이스 테스트 -사용자 문서 갱신 -소프트웨어 코드 및 테스트 결과 평가
소프트웨어 통합	-통합계획 수립 -단위 소프트웨어 및 컴포넌트 통합 -사용자 문서 갱신 -테스트 케이스 및 테스트 절차 수립 -소프트웨어 통합 평가
소프트웨어 적합성 테스트	-적합성 테스트 수행 -사용자 문서 갱신 -설계, 코드, 테스트, 테스트결과, 사용자 문서평가 -검사 -베이스 라인 설정
시스템 통합	-소프트웨어 형상 항목 통합 -테스트 케이스 및 테스트 절차 수립 -시스템 통합 평가
시스템 적합성 테스트	-시스템 적합성 테스트 수행 -시스템 평가 -검사 -베이스 라인 설정

2.2. RUP(Rational Unified Process)

RUP은 4개의 단계(Phase)와 각 단계에서 반복적으로 수행하는 워크플로우(Workflow)로 구성된다. 프로젝트는 도입(In ception) 단계에서 시작되어 전이(Transition) 단계에서 종료되며, 각 단계에서는 단계별로 할당된 유스케이스(Use Case)들에 대한 개발 워크플로우를 진행하게 된다. 이를 통해서 RUP는 점증적이면서도 반복적인 개발 주기를 제공 한다. 또한 RUP의 가장 중요한 특징은 유스케이스 중심(Use-Case Driven) 프로세스란 점과 아키텍처 중심적(Architec- ture Centric)인 4+1 View란 점이다.

2.3. 마르미 III(MarMI III)

마르미는 국내의 개발여건을 반영하여 개발된 한국형 정보시스템 구축방법론이다. 이 방법론은 국내방법론의 요 구사항을 적극적으로 수용하고자 하였고, 국내 기술진에 의해 개발되었다. 국내방법론에 대한 실태조사나 면담분석, 방법론의 논리적 분석 및 기존방법론의 상세분석 등으로 도출되어 방법론 사용자의 강력한 요구사항을 적극적으로 수용하였다. 마르미는 관리공정과 개발공정으로 이루어져 있다. 관리공정은 프로젝트 시작 단계, 프로젝트 진행 단계, 프로젝트 종료 단계로 이루어져 있으며, 개발공정은 요구획득 단계, 아키텍처 단계, 점진 적 개발 단계, 인도 단계로 이루어져 있다. 마르미는 컴포넌트 개발과 컴포넌트 기반 개발 애플리케이션 개발을 동시에 진행하도록 프로세스를 구성하였다.

3. 단기개발 공공 프로젝트를 위한 산출물 선정 및 관리

3.1. 개발환경 및 프로젝트 적용 문제점 분석

본 논문에서 제시하는 공공사업의 단기 개발 프로젝트를 위한 산출물 선정 및 관리를 적용함에 있어서 프로젝트의 투입인원 및 수행기간에 따른 프로젝트의 규모를 산정한다. 또한 공공사업에서 단기 개발방법을 적용할 수 밖에 없는 프로젝트의 경우 현실적 프로세스 적용의 문제점은 <표 2>에서 처럼 경영적 요소, 개발환경 요소, 개발인력 요소, 프로젝트 요소로 구분 분석하였다.

<표 2> 프로젝트 적용의 문제점 분석

비교내용	기존 개발 방법	CBD 방법론	비 고	
경 영 적 요 소	사업적 요소	-수행 경험 풍부	-신기술 제공 기술 경쟁 우위 -수행경험 미숙	-초기개발 Risk 큼 -단기간 개발 어려움
	계약 요소	-금액 경쟁력 우수 -개발경험 풍부 -기간단축	-개발기간 산출 및 컴포넌트 개발비용 증가	-신기술 적용경험 부족 -금액, 기간 예측 어려움
	고객 요구 사항	-요구사항 습득용이 -고객신뢰 개발기간 조정	-새로운 기술 적용 선호 -재사용 가능 컴포넌트 개발 요구	-기술력부족 Risk -단기간 개발/개발 금액 최소
환 경 요 소	적용 기술	-C/S, Web 기반기술 -JAVA, JSP, C, VB -구조적/객체지향 방법론	-J2EE, 닷넷 기술 -EJB, ASP/ADO.NET C#	-CBD 방법론 단기술특불가 -신기술 적용 추가비용 발생
	PM	-구조적/객체지향 경험 풍부 -도메인지식 풍부 -단기간 구현 가능	-CBD 방법론 개발경험 부족 -프로젝트 관리 능력 부족 -사전 철저한 업무분장 미흡	-PM 이 새로운 방법론 및 컴포넌트기술 부족
인 력 요 소	개발자	-실무 경험 풍부 -개발 후 문서화	-CBD 방법론 기본사항 이해 부족 -컴포넌트 구현 경험 없음 -방법론 가이드 적용 무시 -문서화 작업 회피	-단기 신기술 교육 한계 -신기술에 대한 불안감 -개발자 의식 개혁, 적극성 유도
	컨설턴트	-자체 경험 반영	-전문가 초빙 단기 교육효과 없음 -컴포넌트 구현기술 전수 교육-자체 컴포넌트 구현경험 부족-교육 실효성 없음	-단지 가이드만 제시 -핵심적인 기술구현 한계 -책임 전가
프로젝트 관리 요 소	프로젝트 통제	-PM 혼자 다량 업무 소화 -PM 과 개발자간 의사소통 장애 : 문서화 없이 즉흥적 -신기술 적용 독자적 실행-시행착오발생	-PM 혼자 다량 업무 소화 -PM 과 개발자간 의사소통 장애 : 문서화 없이 즉흥적 -신기술 적용 독자적 실행-시행착오발생	-비효과적 수행전략 -문서작성 후 내용 활용 안함 -요구사항 찾은 변경중심 흔들림
	S/W 품질	-개발 후 문서작성 시행 -유지보수의 문제	-컴포넌트 구현 한계 (DLL) -요구사항 개발 후 수시 발생	-재사용 가능 컴포넌트 구현 실패 -찾은 유지보수 비용발생
납품/ 운영	-적기 납품 가능 -전산전문가 운영	-장기간 소요 납기일 넘김 -인터넷 발달로 비전문가 운영	-운영자 비전문가 운영 -프로그램 품질 향상 절대 필요	

프로젝트 규모가 작을수록 시스템 구축에 필요한 산출물의 내용과 통합조정할 사항이 대규모 프로젝트보다 적을 수 있고, 용이하게 접근이 가능하다. 이러한 점을 고려한다면, 소규모 프로젝트인 경우에는 동일 산출물의 순차적이고 반복적인 작업을 최소화하여 작성하여도 전체 시스템 구축에 문제가 발생할 가능성이 낮다.

3.2. 방법론별 산출물 현황 분석

각 산출물은 프로젝트 수행목적에 따라 후속하는 산출물 작성을 위해 필요한 중간 산출물이 작성될 수 있다. 또한 시스템 Owner에게 제출하여 검토 또는 진척도를 평가받기 위한 제출 산출물이 정의될 수 있다. 이러한 산출물들은 품질관리 수행 대상과 형상관리 수행 대상으로 정의될 수 있다.

그러나 소프트웨어를 보다 효과적이면서 체계적으로 관리하기 위해서는 사업의 특성에 맞게 소프트웨어 개발 프로세스가 정의되어야 한다. MIL-STD-498은 범용성이 가지는 문제점을 테일러링(tailoring)이라는 개념으로 해결할 것을 제안하고 있는데, 본 공공사업 단기 개발 프로젝트에 개발표준을 적용함에 있어 보다 효과적인 소프트웨어 개발관리가 이루어지기 위해서는 소프트웨어 개발 계획 단계에서 사업의 특성에 맞게 개발 프로세스 및 문서화에 대한 범위 및 내용을 정의하여 이를 소프트웨어 개발계획서에 반영하여야 한다.

3.3. 개발 프로세스 정의

공공사업의 단기 개발 프로젝트에 사용되는 개발 프로세스는 ISO/IEC 12207과 RUP, MIL-STD-498 그리고 마르미III를 참조하여 다음과 같이 정의하였다.

□ **요구과약(Requirement)** : 개념모델을 통해 업무 도메인을 파악하고 사용자의 요구사항을 이해한다. 시스템의 범위와 기능을 정의하는데 목적이 있다.

□ **분석 및 설계(Analysis & Design)** : 요구과약 단계의 결과 산출물들을 분석하여 소프트웨어를 구축하기 위한 기본구조를 정의하고 설계한다. 이 단계의 목적은 소프트웨어 아키텍처와 컴포넌트 아키텍처를 정의하는 것이고, 정의된 아키텍처에 따라 컴포넌트, 사용자 인터페이스, 데이터베이스를 구체적으로 설계한다. 설계와 분석 작업은 서로 상호 보완적이고 밀접한 관계를 가지므로 동일한 단계에 두었다.

□ **구현(Implementation)** : 분석 및 설계단계의 산출물들을 기반으로 개발표준을 정의하고, 개발자들은 각종 설계서에 따라 프로그램 코드를 작성한다.

□ **테스트(Test)** : 테스트 종류 및 대상, 절차 등을 계획하고, 테스트를 수행한다. 테스트는 컴포넌트 테스트, 통합 테스트, 시스템 테스트, 인수 테스트 등이 있다.

또한 일반적인 CBD방법론 프로세스에서 사용하는 소프트웨어 생명주기(life cycle)를 기본으로 하여 요구과약, 분석 및 설계, 구현, 테스트라는 개발 단계와 세부 활동을 정의하였다. 개발 단계와 활동이 분리되고 독립된 것 처럼 보이지만 각 작업은 상호 연관성을 가지며 완성 된다. 예를 들어 아키텍처 정의 작업은 소프트웨어 생명 주기 전반에 걸쳐 영향을 주며 또 다른 작업의 영향을 받기도 한다.

3.4. 개발 산출물 정의

개발 산출물이란 개발 프로세스 동안 각 활동에서의 작업결과를 의미한다. 개발 산출물의 이름, 형식 등은

프로젝트의 규모나 성격에 따라 다를 수 있고 산출물의 종류도 유연하게 조절될 수 있다. 그러나 프로젝트 초기에 개발 프로세스와 단계별 활동과 산출물을 정의하는 것은 어떤 프로젝트이건 매우 중요하다. 산출물은 의사 소통의 근간이 되며 소프트웨어의 변경관리를 위한 지표가 된다. **요구과약단계**에서 이루어지는 활동으로는 요구 사항 이해, 요구사항 정의가 있다. 요구사항 이해 활동에서는 일반적으로 소프트웨어의 요구사항을 작성하고, 프로젝트에서 사용되는 전반적인 용어에 대하여 정의하며 업무의 도메인을 분석하여 개념모델을 작성하게 된다. 요구사항 정의 활동에서는 유스케이스 다이어그램을 사용하여 유스케이스 모델을 작성하게 된다. **분석 및 설계 단계**에서 이루어지는 활동으로는 요구사항 분석, 아키텍처 정의, 컴포넌트 설계, 데이터베이스 설계 등이 있다.

요구사항분석 활동에서는 유스케이스와 개념모델로부터 추출한 클래스들을 기반으로 해서 클래스 다이어그램을 작성한다. 산출물명은 마르미III에서는 객체모델이라고 정의되어있지만 일반적으로 단기 개발 프로젝트에서는 클래스 설계서라는 산출물로 작성하게 된다.

구현단계에서 이루어지는 활동으로는 개발표준 정의, 코드구현 등이 있다. 개발표준 정의에서는 명명규칙(Naming Rule), 코딩 규칙같이 개발시에 유의해야 할 개발 지침들을 정의한 문서를 작성한다. 코드구현은 각 플랫폼 폼별로 작성된 구현 소스코드이다. **테스트단계**에서 이루어지는 활동으로는 테스트계획, 테스트수행 및 보고 등이 있다. 테스트 계획에서는 테스트종류 및 방법 등을 기록한 문서를 작성하고, 테스트수행 및 보고에서는 컴포넌트의 테스트 수행 및 절차를 기록한 문서, 통합 테스트와 인수테스트의 수행절차 및 결과를 기록한 문서를 작성한다.

3.5. 개발 산출물간 연관성

개발 단계별 활동은 상호 보완적이며 각 산출물들은 서로 입출력 관계에 있다. 즉 요구사항 명세서는 아키텍처 명세서의 입력산출물이 되며, 아키텍처 명세서는 컴포넌트 명세서의 입력으로 사용된다. 용어사전은 시스템에서 사용하는 표준 용어를 정의한 것으로서 모든 단계에 걸쳐 입력산출물이 된다. 아키텍처 명세서는 시스템 개발 전반에 걸쳐 다른 작업에 영향을 주고, 영향을 받는 산출물이다.

3.6. 개발 산출물과 관련 기법

개발 산출물은 정의된 개발 프로세스의 활동 결과이다. 각 산출물을 중심으로 산출물 작성과 관련된 기법들을 살펴보면 <표3>과 같다.

<표 3> 개발 산출물과 관련기법

산출물	관련 기법
요구사항 명세서	-업무매뉴얼, 기존시스템 산출물 -요구사항의 우선순위 결정
용어 사전	-업무 용어 사전 여부 조사
개념 모델	-요구사항과 용어사전을 사용하여 도메인 모델 분석 -후보 개념 클래스 다이어그램 -사용자 인터뷰를 통한 클래스 모델 정제

유스케이스 모델	-기존 시스템 사용자 매뉴얼과 설명서 참조 -비즈니스 유스케이스 모델 참조 -요구사항명세서, 용어 사전 참조 -액티 정의 -유스케이스 도출 -유스케이스 이벤트 기술 -관련 업무 규칙 파악 -유스케이스 다이어그램 -필요에 따라 유스케이스 패키지 정의 -유스케이스의 우선순위
객체 모델	-개념모델, 유스케이스 모델 분석 -분석 클래스 도출(boundary, control, entity) -분석 클래스 다이어그램
아키텍처 명세서	-요구사항명세서, 유스케이스 모델, 개념모델 참조 -아키텍처 기술방법 정의 -품질속성 정의 -아키텍처 패턴을 사용하여 후보 아키텍처 정의 -시스템 이해 관계자들과의 인터뷰 필요 -최초 아키텍처 정의 -하드웨어 구성도 -프레임워크 -구현 기술에 대한 이해
UI 설계서	-유스케이스 모델, 객체 모델 참조 -UI 설계 전략 필요 -디자이너 및 사용자와의 협의 필요
컴포넌트 명세서	-도출된 인터페이스의 상호작용 분석 -컴포넌트 아키텍처 필요
컴포넌트 설계서	-구현 언어 및 기술에 대한 지식 필요 -디자인패턴 또는 프레임워크 지식 필요
인터페이스 명세서	-인터페이스 도출 방법 필요 -유스케이스 모델과 객체모델 참조
데이터베이스 설계서	-OR Mapping -ISP 결과물 -객체모델, 아키텍처명세서 참조 -기존 시스템의 데이터베이스 스키마 참조
테스트계획서	-테스트 종류에 대한 관련자 협의
테스트보고서	-테스트 결과 검토 및 평가

4. 사례 연구

4.1. 프로젝트 수행 사례분석

본 논문에서 제시한 공공사업 단기 개발 프로젝트를 위한 산출물을 비교 검증하기 위한 사례연구는 다음과 같다.

□ 개발개요 : Data를 분석하여 결과에 따라 업무를 처리하는 시스템을 개발하는 프로젝트로서, 분석시스템과 관리시스템으로 구성되며, 분석시스템은 DW(Data Warehouse)로 관련 분석도구와 DW모델을 개발하였으며, 관리시스템은 웹기반의 워크플로우(Workflow)를 개발하여 분석결과를 처리하는 업무시스템이다. 본 논문에서는 관리시스템을 대상으로 분석하였다.

□ 개발기간 : 5개월

□ 개발규모 : 약 650본

□ 적용방법론 : PPC 방법론중에서 웹 개발 경로를 적용하였으며, 일부 패키지 경로를 적용하였다.

4.2. 평가 - 사례연구 vs 제안 산출물 비교검증

본 논문에서 제안하는 산출물의 명칭은 RUP와 MIL-STD-498의 표준을 따랐으며, RUP에서 정의되어 있지 않은 산출물의 경우에는 마르미III를 참조하여 산출물을 정의하였다.

현행업무를 분석하는 단계에서는 현행업무 분석서를

작성하게 된다.

요구사항을 정의하는 단계에서는 소프트웨어 요구사항 명세서를 작성하게 된다.

분석 단계에서는 유스케이스 목록, 유스케이스 모델, 유스케이스 명세서를 별도로 작성하지 않고 유스케이스 기술서라는 산출물로 작성하게 된다. 일반적으로 산출물을 작성하다 보면 유스케이스 기술서 산출물에서 액티 목록, 유스케이스 목록, 유스케이스 다이어그램, 시나리오를 모두 작성하게 되기 때문에 굳이 산출물을 여러 개로 나누어서 작성할 필요가 없다.

S/W아키텍처 분석 및 설계 단계에서는 시스템의 하드웨어 및 소프트웨어에 대한 아키텍처명세를 소프트웨어 아키텍처명세서 산출물로 작성하게 된다. 용어사전의 경우에는 반드시 필요한 산출물은 아니기 때문에 단기 개발 프로젝트의 경우에도 프로젝트의 성격에 따라 작성 하던 된다. 설계단계에서 작성하게 되는 산출물 중에서 사이트맵 기술서는 웹(Web)기반의 프로젝트일 경우에는 작성해야 하지만 애플리케이션이나 기타 프로젝트의 경우에는 작성할 필요가 없다. 유스케이스 설계서는 시퀀스(sequence)다이어그램을 작성하는 문서이고, 클래스 설계서는 클래스 다이어그램을 작성하는 문서이다.

시험단계에서는 테스트 계획서, 단위테스트 계획서, 통합테스트 계획서를 각각의 산출물로 작성하지 않고, 통합테스트 계획서, 통합테스트 보고서 산출물 안에서 옵션을 두어 단위테스트, 통합테스트, 인수테스트, 시스템테스트 등을 식별할 수 있도록 하여 하나의 산출물로 작성한다. 운영단계에서는 운영자매뉴얼과, 사용자 매뉴얼을 산출물로 작성한다.

5. 결론 및 향후 연구

지금까지 공공사업에서 필요한 산출물 관리를 위해 각 사업에서 적용할 수 있는 유형별 산출물에 대한 내용을 살펴보았다. 기존의 소프트웨어 개발프로세스는 장기 개발 프로젝트에 적합한 프로세스로서 단기 개발 프로젝트에 적용하기에는 많은 어려움이 있었다. 따라서 본 논문에서 제안한 공공사업의 단기 개발 프로젝트의 산출물 선정 및 관리 기법을 통하여 선정된 산출물은 각 방법론별 세부 산출물의 비교와 사례연구를 통하여 선정 하였으며, 본 논문에서는 공공사업의 단기 개발에 필요한 산출물을 선정하고, 관리하는 가이드라인을 제공 하였다. 향후에는 프로젝트 기간이나 예산을 한정적으로 하여 공공사업을 추진할 경우에 산출물 가이드라인을 어떻게 적용해야 하는지 등을 보완하여야 한다. 본 논문의 내용을 적극 활용하고 그 결과에 대해 피드백을 제공함으로써 보다 나은 산출물 관리를 할 수 있는 연구가 진행되어야 한다.

[참고문헌]

- [1] 공공부문 정보화사업 산출물 관리 방법론, 한국전산원, 2004. 5.
- [2] 소프트웨어 프로세스와 품질, 홍릉과학출판사, 정기간, 1997.
- [3] 중소기업 프로젝트의 컴포넌트개발을 위한 단기 개발 프로세스, 숭실대박사학위논문, 노재우, 2004. 6.
- [4] ISO 12207, ISO/IEC Standard for Information Technology-Software Life Cycle Processes, ISO/IEC JTC /SC7, March, 1998.

- [5] Rational Unified Process Version 2003, IBM Corporation, <http://www.rational.com/product/rup>
- [6] Magic and Robust Methodology Integrated(MaRMI) III, Electronics and Telecommunications Research Institute, 2001.
- [7] Magic and Robust Methodology Integrated-Reengineering Ver 0.9, Electronics and Telecommunications Research Institute, 2003.
- [8] MIL-STD-498 Data Item Description, <http://www2.umassd.edu/SWPI/DOD/MIL-STD-498/toc.html>