

CBSD 환경에서의 ISO/IEC 12207 소프트웨어 생명주기 공정 표준에 대한 적용성 연구

김은영^o, 백인섭
아주대학교 컴퓨터공학과

A Study on Adaptability of ISO/IEC 12207 Software Life Cycle Process Standard in CBSD Environment

Eun-Young Kim^o, In-Sup Paik
Dept. of Computer Engineering, Ajou University

요 약

소프트웨어 시스템에 대한 요구사항이 증가하고 그 규모가 점차 거대·복잡해짐에 따라 시스템을 개발하기 위한 절차, 방법 등이 무수히 생겨나게 되었다. ISO/IEC 12207 소프트웨어 생명주기 공정 표준은, 존재하고 있는 소프트웨어 개발 및 유지보수에 대한 절차 및 방법에 대한 일반적인 지침을 제시하기 위해 제정되었다. 이 표준은 표준 자체의 특성의 하나인 보편성을 유지하기 위해, 특정 프로세스 모델에 치우치지 않고 기반 표준(Base Standard)의 성격을 갖는다. 결국 표준은, 어떠한 프로젝트에서 어떠한 프로세스 모델을 사용하고 임의의 방법론을 사용하든 모든 경우에 적용될 수 있어야 한다. 그런데 최근 소프트웨어 산업의 새로운 패러다임으로 받아들여지고 있는 컴포넌트 기반 소프트웨어 개발(CBSD)환경에서는, 전통적인 개발환경과 많은 차이점을 지적하며 새로운 개발 프로세스들이 제안되고 있다[4]. 따라서 본 논문에서는 CBSD환경에서의 ISO/IEC 12207 표준의 적용성을 고찰해보고 표준의 발전, 개선 방향에 대해 모색해 보고자한다.

1. 서론

소프트웨어 시스템의 규모와 복잡도가 증가함에 따라 많은 문제점이 나타나게 되었으며, 이를 해결하기 위하여 다수의 절차, 방법, 도구 등이 출현하였다. 그러나 이것은 한편으로 소프트웨어 시스템 개발에 혼란을 가중시키게 되었다. 결국 통일된 지침을 제시하는 소프트웨어 공정표준의 필요성이 대두되었고, 1980년대 이후 여러 표준들의 통합과 발전을 거듭하여 1995년 ISO/IEC 12207 소프트웨어 생명주기 공정 표준(Software Life Cycle Process Standard)이 국제 표준으로 제정된 후 현재까지 사용되고 있다[1][2][3].

한편 컴포넌트 기반 소프트웨어 개발(CBSD : Component Based Software Development) 방식은 최근 소프트웨어 산업의 새로운 패러다임으로 각광받고 있다. CBSD는 소프트웨어를 하드웨어 개발환경과 유사하게 '플러그 앤 플레이(Plug-and-Play)' 방식으로 구축하는 방식이다. 그런데, 이는 처음부터 개발하는(develop from scratch) 전통적인 개발방식으로는 수용이 불가능하다. 기존에 개발된 컴포넌트를 획득(procurement)하고 통합(integration)하는 작업이 전체 공정의 핵심이 되고 있기 때문이다. 결국 CBSD 환경에 적합한 개발 프로세스 모델의 출현이 요구되었고 최근 이와 관련된 연구가 현재 활발하게 진행되고 있다[4].

따라서 본 논문에서는, 현재 사용되고 있는 국제 표준인 ISO/IEC 12207 소프트웨어 생명주기 공정 표준이 전통적인 개발환경과는 매우 다른 재사용 중심의 CBSD환경에서 얼마나 적용성이 있는지를

검증해 보고, 이를 바탕으로 표준의 발전, 개선방향에 대하여 모색해 보고자 한다.

2. ISO/IEC 12207 소프트웨어 생명주기 공정 표준 [5]

ISO/IEC 12207 소프트웨어 생명주기 공정 표준은 1995년 소프트웨어 개발을 위한 일관적이고 체계적인 구조/framework를 제공하기 위해 제정되었다. 이 표준은 소프트웨어 개발 시 고려해야 할 5개의 기본(primary) 프로세스, 8개의 지원(supporting) 프로세스, 4개의 조직(organizational) 프로세스에 대한 내용으로 구성된다. 표준을

Primary Life Cycle Processes	Supporting Life Cycle Processes
<ul style="list-style-type: none"> • Acquisition • Supply • Development • Operation • Maintenance 	<ul style="list-style-type: none"> • Documentation • Configuration Management • Quality Assurance • Verification • Validation • Joint Review • Audit • Problem Resolution
Organizational Life Cycle Processes	
<ul style="list-style-type: none"> • Management • Infrastructure • Improvement • Training 	

[그림 1] ISO/IEC 12207 표준의 구조

이루고 있는 각 프로세스는 다음과 같다 [그림 1]. 각 프로세스들은 하나 이상의 액티비티(activity)들로 이루어지며, 액티비티는 하나 이상의 태스크(task)들로 이루어진다.

> 기본 프로세스(Primary Process)

소프트웨어 개발 프로세스의 주 골격을 이루는 프로세스들이다. 소프트웨어의 획득, 공급, 개발, 운영, 유지보수에 대한 활동을 정의한다.

> 지원 프로세스 (Supporting Process)

기본 프로세스들이 원활히 진행될 수 있도록 보조해주는 역할을 하는 프로세스이다. 각 기본 프로세스로부터 산출되는 문서, 품질 보증, 감사, 문제해결 등에 대한 활동을 정의한다.

> 조직 프로세스 (Organizational Process)

개발 전주기에 거쳐 전체 프로젝트를 관리하는 역할을 하는 프로세스이다. 프로젝트의 기반구조, 개선, 인력훈련 등에 대한 활동을 정의한다.

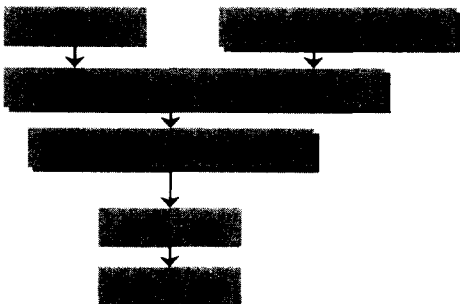
본 표준은 보편성을 유지하기 위하여 특정 프로세스 모델에 치우치지 않고 기본 표준을 지향하고 있기 때문에, 소프트웨어 개발에 필요한 요소적 프로세스들을 전체적으로 보여주고 있지만 실제 프로세스들이 어떻게 상호 연관되는지, 프로세스내의 액티비티 및 태스크들은 어떻게 상호 연관되는 지에 대해 명시하고 있지 않다. 결국 이 표준을 이용해 소프트웨어 개발을 수행하는 조직들은 개발하려는 프로젝트의 성격에 맞도록 방법론 및 프로세스 모델을 선정 혹은 정립하고, 프로세스 모델의 세부 절차들은 본 표준에서 제시된 항목들을 취사선택, 변형(tailoring)하여야 한다.

3. CBSD 환경에서 제안된 프로세스 모델

ISO/IEC 12207 표준의 적용성을 고찰하기 위해 본 절에서는 두 개의 컴포넌트 기반 프로세스 모델을 선택하였다.

3.1 M. Aoyama의 제안 프로세스 모델[6][7]

[그림2]는 M. Aoyama가 제안한 CBSD(Component Based Software Development) 프로세스 모델이다. 이 모델에서 소프트웨어 시스템 개발에 필요한 컴포넌트들은 외부 조직으로부터 구입하거나 주문 생산할 수 있고, 내부에서 자체적으로 개발할 수도 있다. 컴포넌트 획득, 컴포넌트 지향 설계, 컴포넌트 조립 절차를 추가시킨 것이 기존의 프로세스 모델과 크게 달라진 점이다.



[그림 2] M. Aoyama가 제안한 프로세스 모델

> 컴포넌트 획득(Component Acquisition)

컴포넌트 시장으로부터 필요한 컴포넌트를 선택하는 단계이다. 여기서의 컴포넌트는 내부에서 직접 개발한 것일 수도 있고, 다른 조직으로부터 개발된 것일 수도 있다.

> 컴포넌트 지향 설계(Component Oriented Design)

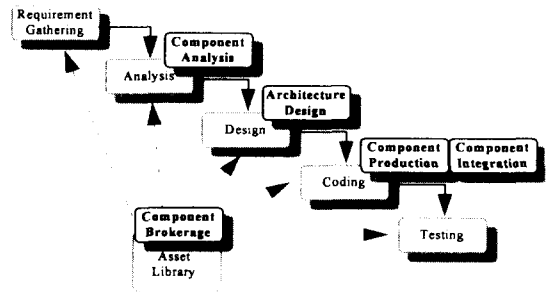
도입한 컴포넌트를 어떻게 조합해야 할지 설계하는 단계이다. 필요하다면, 앞 단계에서 도입한 컴포넌트를 시스템에 맞도록 수정할 수 있다.

> 컴포넌트 조립(Component Composition)

실제 환경에서 컴포넌트를 조합함으로써 시스템을 구축하는 단계이다.

3.2 J. Ning의 제안 프로세스 모델[8]

[그림 3]은 J. Ning이 제안한 CBSE 프로세스 모델이다. 이 프로세스 모델 역시 앞에서 살펴본 M. Aoyama의 프로세스 모델과 마찬가지로 초기 요구사항 수집과정을 통해 내부적으로 직접 개발해야 할 컴포넌트와 외부에서 도입해야 할 컴포넌트를 결정한다. 그리고 이 두 종류의 컴포넌트는 각기 다른 과정을 거쳐 개발되거나 수집된다.



[그림 3] J. Ning의 제안 프로세스 모델

기존(conventional) 프로세스 모델과 달라진 부분은 그림에서 진한 테두리로 표시했다. 필요한 컴포넌트가 무엇인지, 무엇을 외부에서 도입할지를 결정하는 컴포넌트 분석 단계, 컴포넌트 사이의 상호관계를 정의하는 아키텍처 설계 단계, 시스템에 맞도록 컴포넌트를 수정하는 컴포넌트 생산(Production) 단계, 각각 만들어진 컴포넌트들을 하나의 시스템으로 조합하는 통합 단계, 서로 다른 환경에서 만들어진 컴포넌트들을 중개시켜주는 중개 단계가 이 프로세스 모델에서 새롭게 제안된 요소들이다.

4. ISO/IEC 12207 표준의 적용 및 적용성 고찰

본 절에서는 3절에서 적용시킨 표준의 요소들을 바탕으로 현재 국제 표준으로 사용되고 있는 ISO/IEC 12207 소프트웨어 생명주기 공정 표준의 적용성에 대해 고찰해 본다. 앞서 살펴본 두 개의 컴포넌트 기반 프로세스 모델은 컴포넌트 자체 개발과 컴포넌트 외부 도입에 대한 두 개의 프로세스로 나뉘어져 진행된다는 공통점을 찾아볼 수 있다.

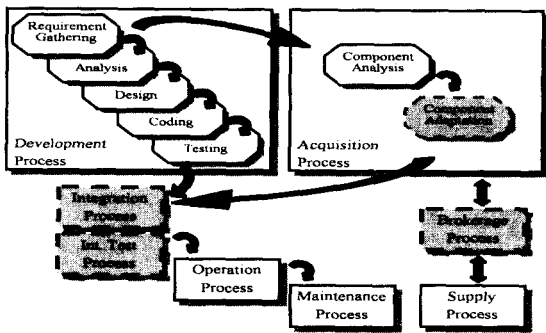
개발 프로세스

내부적으로 개발해야 할 컴포넌트를 위한 프로세스이다. 기존에 사용되던 전통적(Conventional) 모델과 동일하게 진행된다.

획득 프로세스

자체 개발을 하지않고 외부에서 필요한 컴포넌트를 선택, 도입하기 위한 프로세스이다. 필요한 컴포넌트가 무엇인지를 식별하는 단계에서부터, 아키텍처 설계, 컴포넌트 도입, 도입 후 시스템에 적절하도록 수정(Adaptation), 통합하는 단계로 진행된다.

[그림4]는 위에서 살펴본 내용을 바탕으로 한 일반적인 CBSD 프로세스 모델에 ISO/IEC 12207 표준의 요소들을 적용시킨 것이다. 프로세스는 사각형으로, 프로세스 내의 액티비티는 팔각형으로 표시하였다. 또한, ISO/IEC 12207 표준에 이미 존재하고 있는 요소들은 실선으로, 프로세스 모델에는 필요하지만 표준에 갖추어지지 않은 요소들은 점선으로 표기하였다.



[그림 4] CBSD 모델에 대한 ISO/IEC 12207 표준의 적용

일단 개발 프로세스의 요구사항 수집 과정에서 자체적으로 개발할 컴포넌트와 외부에서 도입해야 할 컴포넌트가 결정되면 프로세스는 두 갈래로 나누어져 진행된다. 자체 개발 프로세스는 ISO/IEC 12207 표준의 개발 프로세스(Development Process)의 요소들로써 구성되며, 표준의 모든 요소를 만족한다. 그러나 외부 도입 프로세스로 넘어가면 표준의 적용에 문제가 발생한다. 문제점은 액티비티와 프로세스(process) 차원에서 각각 살펴볼 수 있다.

먼저 액티비티 차원에서 살펴보면, 획득 프로세스(Acquisition Process) 내에서 문제가 발생함을 알 수 있다. 컴포넌트의 도입 시, 개발하려는 시스템의 부분으로써 완전히 만족되는 컴포넌트를 구하기는 쉽지 않다. 따라서 개발 프로세스에서 작성된 코드를 수정하거나 도입한 컴포넌트의 기능을 추가, 삭제해야 하는데, 이 부분에 대한 가이드라인이 부족하였다.

프로세스 차원에서 살펴보았을 때, 세 개의 프로세스가 ISO/IEC 12207 표준의 요소들로 만족되지 않았다. 각각은 다음과 같다.

통합 프로세스(Integration Process)

개발(Development) 프로세스 내에 통합 액티비티가 존재하지만 이것은 내부에서 개발된 컴포넌트에 대한 통합 행위만을 정의하는 것이지 서로 이질 환경에서 생성된 컴포넌트간의 통합을 의미하는 것은 아니다. 따라서 이 프로세스에서는 개발 프로세스와 획득 프로세스에서 각기 생성된 이질 컴포넌트들을 하나로 통합하는 부분에 대한 가이드라인을 제시하는 프로세스가 부족하다.

통합 테스트 프로세스(Integration Test Process)

통합된 컴포넌트가 구축하고자 하는 시스템에 맞도록 제대로 기능

을 수행하는지 컴포넌트와 전체 시스템을 테스트하는 부분에 대한 가이드라인을 제시하는 프로세스가 부족하다.

중개 프로세스(Brokerage Process)

컴포넌트 시장이 활성화 될수록 획득자(Acquirer)와 공급자(Supplier) 사이에 존재하는 중개자(Broker)의 역할이 증대된다. 따라서 획득 프로세스와 공급 프로세스 사이를 중개해주는 부분에 대한 프로세스의 출현 및 보장이 필수적이다.

위에서 언급한 프로세스들은 컴포넌트 기반 프로세스 모델에서 중요한 비중을 차지하고 있는 부분이다. 하지만 현재 국제 표준으로 사용되고 있는 ISO/IEC 12207에 이 프로세스들에 대한 표준의 요소들은 존재하지 않았다. 이것은 앞으로 소프트웨어 개발의 주 패러다임이 될 컴포넌트 기반 소프트웨어 개발방식에 이 표준을 적용하기에는 부족한 면이 많다는 것을 의미한다. 따라서 국제 표준으로써 여러 개발 패러다임을 수용하기 위해서는 위에서 언급한 표준의 요소로서 부족한 프로세스 및 액티비티를 보완, 확장할 필요가 요구된다.

5. 결론 및 향후 연구 과제

본 논문에서는 현재 소프트웨어 생명주기 공정 국제 표준으로 사용되고 있는 ISO/IEC 12207 표준이 컴포넌트 기반 소프트웨어 개발 환경에서 얼마나 적용성이 있는지를 고찰해 보았다. 연구 결과, 외부에서 개발된 컴포넌트를 도입하는 프로세스를 갖추고 있는 CBSD의 특수성 때문에 컴포넌트 기반 소프트웨어 개발 환경에서 ISO/IEC 12207 표준을 적용하는 데에는 약간의 문제가 발생함을 알 수 있었다. 또한 이러한 문제점을 보완하기 위해 표준의 개선, 확장의 필요성을 언급하였다. 향후 연구 과제로는 어떠한 개발 환경에서라도 적용 가능하도록 프로세스 및 액티비티의 차원에서 표준의 확장 방안을 찾아보는 연구가 필요하다.

참고문헌

- [1] Lewis Gray, "ISO/IEC 12207 Software Life Cycle Processes", <http://www.stsc.hill.af.mil/crosstalk/1996/08/isoiec.asp>
- [2] Jim Moore, "ISO 12207 and Related Software Life Cycle Standards", <http://www.acm.org/ts/lifecycle.html>
- [3] "Heritage of Systems Engineering Standards" <http://www.incose.org/stc/standards-evolution.htm>
- [4] 심우권, 백연섭, "컴포넌트 재사용성 제고를 위한 포괄적 프로세스 모델", 제 2회 한국 소프트웨어공학 학술대회 논문집, 제 2권 1호, pp. 211-220, 2000
- [5] ISO, "ISO/IEC 12207 International Standard, Information Technology - Software Life Cycle Processes", Aug, 1995
- [6] Mikio Aoyama, "New Age of Software Engineering Development : How Component-Based Software Engineering Changes the Way of Software Development?", in Proc. of the First International Workshop on Component-Based Software Engineering - In Cooperation with the 20th ICSE, Kyoto, Apr. 1998.
- [7] Mikio Aoyama, "Process and Economic Model of Component-Based Software Development : A Study from Software CALS Next Generation Software Engineering Program", in Proc. 5th IEEE SAST(Symp. on Assessment f Software Tools), Jun, 1997, pp.100-103.
- [8] Jim Q. Ning, "A Component-Based Software Development Model", in Proc. COMPSAC'96, Aug, 1996, pp.389-394