

소프트웨어공학 표준화 기술현황 분석

이상무* · 오행석*

*한국전자통신연구원

Analysis on the current status of standardization technology in Software Engineering

Sang-mu Lee* · Haeng-seok Oh*

*Korea Electronics and Telecommunications Research Institute

E-mail : sangmu@pec.etri.re.kr

요 약

소프트웨어 공학이란 적절한 자원을 사용하여 품질 높은 소프트웨어를 개발, 획득, 운용하기 위해 필요한 기법, 도구 및 방법론의 총칭이다.

소프트웨어공학 활동의 관리목표는 적절한 자원의 사용을 통한 높은 생산성 추구 및 양질의 소프트웨어 생산이라 할 수 있다. 이러한 관리목표를 달성하기 위해서는 공학적인 이론과 개념을 바탕으로 한 공학원리를 확립하고 개발방법, 방법론 및 소프트웨어공학 도구를 발전시켜 효율적인 공학기술 환경을 구축하여야 할 것이다. 또한 매년 급증하고 있는 공공 정보화사업의 효율성을 제고하고 상호호환성과 운용성을 보장하기 위해 소프트웨어 공학을 바탕으로 정보시스템 기획, 획득, 개발, 운영, 유지보수 및 평가 등과 관련된 각종 지침도 개발되어야 한다. 본 논문에서는 소프트웨어공학 관련 기술 기술개발현황 및 시장 표준개발현황에 관해 다룬다.

ABSTRACT

Software Engineering covers techniques, tools, and methodologies needed to develop, acquire, and manage software in high quality, using appropriate resources. Therefore it can be said that the administration target of software engineering activities is the pursuit of high productivity for production of software in good quality. To do this, they should establish the engineering principle based on engineering theory and concept, and form the effective environment of engineering technology through improving development methods/methodology and tools of software engineering. In addition, several guidances related to information system planning, acquisition, development, management, maintenance, and evaluation, etc. must be invented on the basis of software engineering to encourage efficiency of public information business yearly increasing fastand to make sure compatibility and interoperability.

Technology, its developing and marketing situation, and standardization status are treated in this paper.

1. 서 론

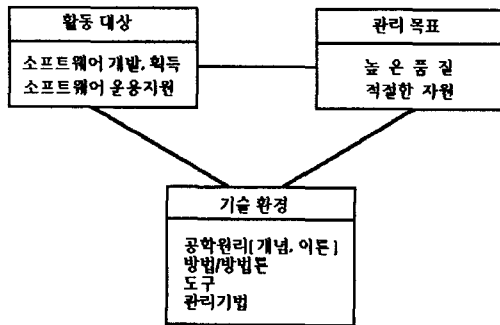
소프트웨어공학이란 적절한 자원을 사용하여 품질 높은 소프트웨어를 개발, 획득, 운용하기 위해 필요한 기법, 도구 및 방법론의 총칭이다.

소프트웨어공학 활동의 관리 목표는 적절한 자원의 사용을 통한 높은 생산성 추구 및 양질의

소프트웨어 생산이라 할 수 있다. 이러한 관리 목표를 달성하기 위해서는 공학적인 이론과 개념을 바탕으로 한 공학 원리를 확립하고 개발방법, 방법론 및 소프트웨어공학 도구를 발전시켜 효율적인 공학기술 환경을 구축하여야 할 것이다. 또한 매년 급증하고 있는 공공 정보화 사업의 효율성을 제고하고 상호호환성과 운용성을 보장하기 위

해 소프트웨어공학을 바탕으로 정보시스템 기획, 획득, 개발, 운영, 유지보수 및 평가 등과 관련된 각종 지침도 개발되어야 한다.

소프트웨어공학이 다루는 관심사항은 그림 3.9.1과 같이 소프트웨어공학 활동대상, 활동 수행의 관리 목표 그리고 효과적인 활동 수행을 위한 기술환경으로 생각할 수 있다.



(그림1) 소프트웨어공학 분야의 관심사항

정보화 사회를 위한 컴퓨터 기술기반 확립의 중요성과 세계적인 정보기술 활용부문의 시장규모가 확대되고 경쟁이 치열해짐에 따라 소프트웨어공학 분야의 기술은 급속도로 발전하였다. 그러나 소프트웨어공학 분야의 전문인력의 부족과 전문인력 양성을 위한 비용이 크게 증가되고 있는 현실을 타개하고 높은 품질의 소프트웨어 생산을 촉진하기 위해 표준화 작업이 필요하게 되었다. 정보처리시스템을 모델링하고 명세화하고 설계하는 일에도 이식성(portability)을 필요로 하게 되었고 소프트웨어공학 도구에 대한 수요가 크게 증대됨에 따라 자연히 상응하는 표준화의 요구도 커지고 있다. 정보시스템의 유지보수 비용이 차지하는 비중이 점차 확대되고 있는 현실도 표준화의 필요성을 더해주고 있으며 소프트웨어의 안전, 보안신뢰도를 다룰 때도 표준화의 필요성은 절실히 해지고 있다.

이것을 해결하기 위해 필요한 것이 효과적인 표준들을 제정하도록 추진하는 것이다. 효과적인 표준 제정이란 표준의 적용 효과를 높이는 일과 표준들 사이의 일관성을 유지하고, 중복성을 줄이며, 표준의 유지보수를 원활하게 하는 일이다.

소프트웨어공학 표준은 소프트웨어 개발과정에 적용되어 산출물의 품질과 성능을 목표 수준에도달하게 하고, 조직내에서 의사소통을 원활하게 하며, 소프트웨어사업의 위험도를 줄일 뿐만 아니라 조직 내에서 각자의 위치와 업무영역을 명확히 하는 데 도움을 준다. 표준화 사업 추진으로 정보기술분야의 기술능력의 향상과 아울러 정보산업 발전과 국제교역의 증대 등으로 얻는 효과 또한 표준화의 필요성을 더해주는 요인이다.

모든 정보기술분야의 사업추진과 소프트웨어 응용 및 유지 활동을 위한 소프트웨어공학 표준화의 목적을 요약하면 다음과 같다.

- 원활한 의사소통(모델, 명세, 문서의 정형화)
 - 소프트웨어 산출물의 이식성 및 재사용성 증대
 - 소프트웨어 산출물의 품질향상
 - 소프트웨어 개발규모 및 비용산정을 위한 지침제공
 - 품질과 생산성 향상을 위한 기술능력 제고
- 공공 부문의 입장에서 살펴보면, 정보기술의 급속한 발달로 인해 정보기술 표준은 민간의 사실 표준화기구와 글로벌 표준화 기구를 중심으로 개발되고 있어서 개발과 관련된 정부의 역할은 축소되는 대신 이러한 정보기술의 적절한 선택과 정보화 사업의 효율적인 관리의 중요성이 증대되고 있으며, 소프트웨어공학을 바탕으로 한 필요성과 함께 실제적인 관리를 위한 각종 지침 개발의 필요성이 증대되고 있다.

하지만 국내의 경우 1999년도에 공공부문의 정보화책임관(CIO) 제도를 도입하였으나 정부부처의 정보화 책임관이 활용할 수 있는 종합적인 정보기술 관련 지침이 부족하고 정보기술의 선택, 관리, 평가 등을 위한 기준과 지침이 부족한 실정이다.

따라서, 정보화 추진을 위한 모델이나 지침, 정보화 사업간 상호호환성 확보를 위한 지침, 조직의 전체 정보자원(지식, 인력, 정보기술, 업무, 시스템 등)을 관리하기 위한 지침과 함께 계획, 유지보수, 평가, 감리지침등 정보화사업 실무자가 참고할 지침을 시급히 개발하고 보급해야 한다.

II. 소프트웨어공학의 핵심기술

미래의 소프트웨어 개발 기술은 다양한 관점에서 변혁과 발전을 지속할 것으로 전망된다. 먼저, 사용자 요구의 변화로 인해 새로운 어플리케이션이 지속적으로 개발될 것이며, 인터넷과 웹 환경의 등장으로 네트워크 지향 소프트웨어 아키텍처가 주축을 이루는 환경으로 변화해 갈 것으로 예상된다. 또한 소프트웨어 사용자가 전문가로부터 일반 사용자로 변화해 감으로써 엔드유저컴퓨팅(EUC)이나 엔드유저개발(EUD) 환경으로 이동될 전망이다. 객체지향, 컴포넌트웨어 등 소프트웨어 개발기술에 대한 패러다임에도 큰 변화가 예상된다.

1) 소프트웨어 아키텍처

변화에 대응할 수 있는 소프트웨어를 구현하기 위해서는 소프트웨어 전체의 포괄적인 구조를 고려하지 않을 수 없다. 이 문제는 소프트웨어 아키텍처로서 수년간 연구개발이 진전되어 왔다. 종래에도 소프트웨어 아키텍처의 개념이 있었지만 소프트웨어의 각 계층에서 아키텍처를 설계하거나

재사용하는 구조는 없었다. 이를 위해 소프트웨어 아키텍처를 기술하는 아키텍처 기술언어(Architecture Description Language), 아키텍처 설계 방법론, 아키텍처 설계 지원환경, 아키텍처 스타일(패턴), 아키텍처 지향 개발(Architecture-based Development) 등 광범위한 연구 개발이 진행되고 있다.

2) 컴포넌트웨어

컴포넌트웨어란 조합하고 컴파일하여 그대로 이용할 수 있는 Plug & Play형 소프트웨어 부품을 조합시켜 어플리케이션을 구축하는 기술이다. 초기의 컴포넌트웨어는 Windows를 주체로 하는 말하자면 독립형(Stand-alone) 컴포넌트웨어였다. 지금은 네트워크상에서 부품을 제휴하는 분산 컴포넌트웨어로 진화하고 있다. 대표적인 컴포넌트웨어 아키텍처로서 ActiveX/Windows DNA, CORBA, Java Beans가 있다. 이것들은 분산처리 환경에서 부품을 조합시키는 기반이 되는 분산객체 환경을 일체로 하여 제공된다. 국내에서도 업무 어플리케이션 개발로 컴포넌트웨어를 적용한 예가 나타나고 있다. 따라서 특정 업종이나 업무 분야의 컴포넌트 표준화도 진행되고 있다. 이와 같은 하드웨어나 소프트웨어의 범용부품을 COTS(Commercial Off-The-Shelf)라 부른다.

이와 같이 컴포넌트 기술의 발전은 소프트웨어 개발형태, 또는 소프트웨어 산업구조의 재구조화를 재촉하고 있다. 그 징조는 국내외에서 소프트웨어 개발이 부품개발과 부품을 조합시켜 어플리케이션을 개발하는 컴포넌트 인티그레이터(integrator)로 발전된다는 점이다. 국내에서도 소프트웨어 컴포넌트의 벤더(vendor)가 나타나고 있으며, 이를 기반으로 정부에서도 컴포넌트를 구매하기 위한 정책을 추진하고 있다.

3) 객체지향 재사용 기술의 진화

객체지향이 목표로 하는 것 중 하나는 재사용이다. 초기에는 계승(Inheritance)에 의한 코드의 재사용 또는 차분(差分) 프로그래밍이 이용되었다. 이것을 구현의 계승이라 한다. 그러나 구현의 계승은 밀접한 계승 관계이고 상위 클래스의 구현변경이 하위 클래스에 영향을 미치게 된다. 이것에 대응하여 인터페이스의 재사용이 착안되고 인터페이스의 형태를 계승하는 인터페이스 계승(Interface Inheritance)이 사용되고 있다.

가) 설계의 재사용과 프로그램의 재사용

좋은 설계를 재사용하는 구조로서 시스템의 아키텍처를 재사용하는 소프트웨어 아키텍처나 국소적인 객체군의 설계를 재사용하는 디자인 패턴이 제안되었다. 소프트웨어 아키텍처는 포석, 패턴은 정석에 대응된다. 아키텍처에 부합하여 객체를 구조화하고 말하자면, 어플리케이션의 반체

품을 어플리케이션 프레임 워크라 부른다. 패턴이나 프레임워크는 커스터마이징하여 사용한다. 커스터마이징을 용이하게 하기 위해 변경개소를 미리 명시해 두는 방법이 사용되고 있다. 이러한 개소를 Hot Spot 또는 Extension Point 라 부른다. 최초의 프레임워크라고 일컬어지는 MVC(Model View Controller)는 현재 GUI의 기본 프레임워크로서 널리 이용되고 있다.

나) 범용 부품과 도메인 고유부품

부품은 처음에는 도메인에 의존하지 않는 범용 부품이 주를 이루었다. 그러나 부품의 이용효과를 높이기 위해서는 특정 도메인 고유의 요구나 도메인 고유 소프트웨어 아키텍처에 적합한 부품이 필요해진다. 특히 비즈니스 어플리케이션 쪽의 부품은 비즈니스 객체라고 불리고 있다. 도메인 고유 부품의 업계 표준화도 제안되고 있다. 예를 들면, 유통에서는 POS(Point of Sales)용 부품의 표준화가 이루어지고, 이것을 이용한 POS가 제품화되고 있다.

다) 분산객체 환경

클라이언트/서버 어플리케이션을 개발하기 위해서는 클라이언트/서버간의 객체를 이용할 수 있는 구조가 필요하다. 이것을 지원하는 기반 환경으로서 CORBA, DCOM 등의 분산객체 환경이 제공되고 있다.

4) 소프트웨어 개발 프로세스

개발 프로세스는 개발의 절차를 누구든지 이해할 수 있도록 형식적으로 모델화하고 개발자 각각의 배후에 있는 개발방식의 구조적 문제를 해결하는 것이다. 1987년의 프로세스 프로그래밍의 제안에 의해 소프트웨어 개발 프로세스의 연구와 개발이 본격화되었다. 지금까지의 개발 프로세스는 단일 릴리즈(release)를 가진 폭포수형 모델이 중심이었다. 이 개발 프로세스는 대규모의 복잡한 소프트웨어 개발에 적용되고 있었다. 그러나 현재 새로운 소프트웨어 개발에 요구되고 있는 것은 개발 기간의 단축과 변화(성장, 진화)에 신속히 대응하는 것이다. 이로 인해 단계적으로 소프트웨어를 개발하고 제공하는 Incremental Delivery가 주목되고 있다. 특히, 객체지향 개발의 도입이나 RAD(Rapid Application Development)의 도입에서는 이 기술이 적용되고 있다.

개발 프로세스의 개선을 위한 우수한 기법으로서 ISO 9000 시리즈 인증이나 CMM(Capability Maturity Model) 또는 SPICE(Software Process Improvement & Capability dEtermination)를 이용한 심사와 개선이 주목되고 있다.

III. 소프트웨어공학의 시장 현황

1) 소프트웨어공학의 국외 시장

21세기에는 소프트웨어산업이 가장 경쟁력 있는 산업으로 성장하리라는 공통된 인식아래 각 국가들이 소프트웨어산업 육성에 전력을 다하고 있다.

1994년부터 1997년까지의 소프트웨어 년 평균 성장률은 11.1%였으며 이러한 성장세는 2001년까지 지속될 것으로 예상된다. 세계 소프트웨어 시장은 1998년 말 현재 약 4,46억 달러 규모인데, 이러한 성장세가 지속될 경우 2001년에는 6,063억 달러 규모에 이를 것이다. 구체적인 내용은 <표1>에 주어졌다.

<표1> 국외 소프트웨어 시장현황 및 전망(단위:백만불)

구분	1997	1998	1999	2000	2001	1999-2001 평균성장률
패시시스템	119411	137209	157579	179477	208211	136%
컴퓨터관련서비스	281749	319103	337318	359677	408182	93%
합계	401160	456312	494897	539154	606333	107%

(출처 : 소프트웨어 진흥원, 소프트웨어산업의 국내외 시장 및 기술 동향, 1997)

2) 소프트웨어공학의 국내 시장

국내 소프트웨어산업은 정부의 집중적인 육성 정책으로 인하여 매년 40% 내외의 꾸준한 성장을 하고 있으며, 1998년 말 현재 국내 생산은 약 30.5억 달러, 내수 시장은 32.3억 달러의 규모를 나타내고 있다. 하지만 국내 소프트웨어 산업이 아직까지는 미성숙 단계에 있기 때문에 급속한 성장에도 불구하고 소프트웨어의 수출이 수입을 따라잡지 못하는 실정이다.

국내 소프트웨어 산업 현황 및 전망은 <표2>에 주어졌다.

IV. 소프트웨어공학 기술개발 현황

1) 소프트웨어공학의 국외 기술개발

1960년대부터 소프트웨어 개발예산의 초과, 개발일정 지연 문제의 심각성이 제기되면서 소프트웨어공학이란 단어가 탄생하였다. 1970년대에 접어들면서 프로그래밍 자체보다 분석, 설계 단계의 결합이 심각하다는 품질과 생산성 향상에 관련된 논문이 발표되면서 소프트웨어 생명주기에 대한 제안이 등장하고 1980년대에는 방법론과 소프트웨어 재사용, CASE, 품질보증 등의 기술적인 진보가 있었다.

1990년대에는 객체지향, 정보공학, CASE 등이

<표2> 국내 소프트웨어산업의 현황 및 전망 (단위: 백만달러)

구분	1997	1998	1999	2000	2001	1999-2001 평균성장률
생산	4024	3055	3885	5551	8254	457%
내수	4382	3231	4041	5555	7984	403%
수출	51	80	173	366	788	1134%
수입	49	27	33	40	58	257%
무역지	-33	-177	-155	-24	20	-

(출처: 정보통신연구원, 정보기술 2000년전망)

본격적으로 활용되고 소프트웨어 성능공학과 재사용 등이 강조되기 시작하였으며 S/W Life Cycle Process(ISO/IEC 12207-1995) 등이 국제표준으로 제정되었다. 또한 CMM이나 SPICE 등의 표준화 작업도 활발하게 추진되었다.

한편 ISO/IEC JTC1 SC7의 최근 동향을 살펴보면 ISO12207을 기본으로 해서 CASE 선정과 활용 지침, 프로젝트의 관리, 품질 측정과 보증, 그리고 프로세스 심사를 통한 조직의 능력평가와 개선 전략을 위해서 14개의 Working Group에서 작업을 계속하고 있고 이들 작업을 지원하기 위하여 두 개의 Special Working Group에서 SC7의 사업 계획과 각 표준에 적용할 용어 표준을 만들어가고 있다.

최근에 인터넷 활용이 많아지고 개방 체제하에서 멀티미디어 정보처리를 위한 표준을 새롭게 추가해야 되는 과제도 생겨나고 있다.

세계적으로 소프트웨어 기술의 급속한 발전 및 통합으로 소프트웨어 기술표준화가 기술적 차원을 넘어 산업의 국제 경쟁력 강화를 위한 전략적 도구로 부상하고 있는 상황이다. 특히 국제화와 지역적 불균형 현상으로 표준화 활동이 국제 협력의 당면과제로 대두되고 있다. 최근에는 정보통신, 정보처리, 가전산업의 융합이 전개되고 있다.

2) 소프트웨어공학의 국내 기술개발

국내 현황을 살펴보면, 국제 위원회(JTC1/SC7 Software Engineering)에 대응해서 산업자원부 산하 기술 표준원에 소프트웨어공학 전문위원회를 구성하고 WG(Working Group)별로 활동하고 있다. 소프트웨어 상품화에 관련된 표준화를 국제 수준에 맞추어서 국제 시장에서 국내 상품 경쟁력을 강화시키고, 소프트웨어 공학에 관련된 기술을 표준화함으로써 기업에서 공통으로 필요한 기술을 공동 활용하고, 개발 계획 소프트웨어의 개발, 품질보증, 구성 관리 및 프로젝트

트 관리를 수행함으로써 품질을 보증하고 생산성을 향상시킬 수 있을 것이다.

특히, Integral Life Cycle Process와 같은 새로운 프로젝트 관리 개념이 도입되고있고, 공정 심사를 위한 골격을 표준화하여 조직의 소프트웨어 개발능력을 개선하는데 노력하고 있다. CASE 툴의 평가와 선택에 관한 연구는 국내 기업들이 생산성과 품질향상을 위해서 CASE 툴을 도입하고자 하는 시점에서 매우 절실한 표준화 과제라고 판단된다.

V. 소프트웨어공학의 표준화 현황

1) 소프트웨어공학의 국외 표준화

소프트웨어공학에 대한 국외 표준화는 주로 ISO/JTC1 SC7, IEEE 등에서 주로 수행되고 있다.

소프트웨어공학과 관련된 소프트웨어 생명주기, 소프트웨어 품질 및 생산성 향상 등에 대한 표준화 작업이 국제 표준화기구와 사실표준 연구 그룹에서 활발하게 진행되고 있다.

국제표준의 경우 ISO/IEC JTC1이 ITU-T와 상호 협력하여 정보기술 및 정보처리 분야의 표준화 활동을 전개하고 있다.

미국의 경우 소프트웨어 개발 환경 및 품질 보증 등 소프트웨어공학 분야 ANSI 및 IEEE의 표준 활동이 활발하다. IEEE는 ANSI에 의해 미국 국가표준을 개발하도록 위임받은 기구로 주로 소프트웨어공학 분야 표준화를 추진하고 있다.

유럽의 경우 ICTSB(Information & Communication Technology Standard Board)가 소프트웨어 기술분야의 표준화를 담당하고 있으며, 일본은 JSA 산하의 정보기술 표준화 전문기관인 INSTAC을 중심으로 소프트웨어 기술분야의 표준화를 추진하고 있다. 그 밖에 사실표준화 기구인OMG(Object Management Group, 1989년 결성된 객체관련 표준화 기구)에서는 OMA(Object Management Architecture), CORBA(Common Object Request Broker Architecture), UML(Unified Modeling Language) 등을 발표하였다.

공공부문의 경우 공공정보화 추진기관에서 국가 정보화에 필요한 기술구조나 정보화 지침을 민간기관과 협력하여 개발하고 이를 활용한 컨설팅서비스를 제공하고 있다. 미국도 96년 정보기술 관리개혁법(ITMRA)을 통하여 정부조직의 종합적인 정보자원관리의 기준을 마련한 바 있으며, 정보기술자문위원회(ITRB)가 행정기관의 정보시스템의 개발, 조달, 관리 향상, 정보기술의 활용 및 투자에 대한 조사, 분석 및 위험평가를 수행할 수 있도록 자문 역할을 수행하고, 프로젝트관리(project management), 정보시스템 관리를 위한 실제적인 전략, 정보시스템관리의 실용적인 자체 평가 툴, 응용 상용제품(COTS) 리스크 평가지침

등을 제시하고 있다.

영국에서도 정보화 집행 기관인 CCTA가 정보화를 통한 공공서비스 개선과 행정효율화를 위해 프로젝트 관리방법론, 시스템분석·설계방법론, 정보기술서비스관리방법론, 정보기술 획득 방법론과 지침을 주도적으로 개발하여 보급하고 있다.

2) 소프트웨어공학의 국내 표준화

소프트웨어 분야의 국내 표준화 현황을 살펴보면, 국가표준 건수 10,000여건 가운데 소프트웨어 분야는 3%인 300여건에 불과하다. 또한 소프트웨어 개발 기술과 관련해서는 문서화 공정, 품질보증공정, 일부 관리공정, CASE, 용어 등에 대한 표준을 개발하였으나 대부분 국제표준을 단순히 수용하는 수준이어서 국내 표준 사용자들이 실제적으로 이를 이용하기에는 무리가 있고 종합적인 관점의 체계적인표준이 개발되지 않고 있다.

매년 정보화투자가 급증하고 있고 표준화의 중요성이 강조되고 있으나 소프트웨어공학과 관련된 표준개발과 이를 바탕으로 한 체계적인 SW개발 관리는 미흡한 실정이며, 전체적인 체계에 따른 표준화 작업 또한 미흡한 상황이다. 그리고, 표준사용자에게 실질적으로 도움을 줄 수 있는 표준개발, 국제 표준화 활동 참여, 국내표준화 추진체계의 정비, 표준정보 보급 및 활용체계 구축은 미흡하다.

국내의 경우 소프트웨어공학 표준화 추진체계가 정보통신부와 산업자원부로 이원화되어 있어서 표준화 활동의 혼선이 야기되고 있으며, 소프트웨어 표준화를 국제표준의 단순한 수용 및 문서표준화 정도로 인식하고 있으며 민간기업의 표준화 참여가 미미한 실정이다.

소프트웨어 품질인증의 경우 시험인증의 기반이 되는 소프트웨어 표준화 활동이 미흡하며, 소프트웨어 기능표준, 시험규격, 개발 및 관리체계, 소프트웨어 시험평가 규격 등의 표준이 질적, 양적인 측면에서 낙후되어 있다.

국산 소프트웨어의 품질향상과 산업체의 품질개선 능력을 제고 시키고 해외 수출에 대비하기 위한 소프트웨어 품질평가 및 인증기술의 개발이 필요한 실정이다.

국내 표준화 기구로는 한국정보통신기술협회(TTA) : 정보통신소프트웨어 기술위원회 산하의 소프트웨어 개발기술 연구반과 소프트웨어 응용기술 연구반에서 소프트웨어 프로젝트관리, 소프트웨어 생명주기 및 개발방법, 객체지향개발, 개발도구, 소프트웨어 품질보증 및 품질평가 등에 표준화를 추진하고 있다. TTA 단체표준 중 소프트웨어 분야는 20여건, 한국정보통신표준(KICS)는 30여건 제정(1999년 4월 기준)되어 있다.

산업자원부 기술 표준원의 소프트웨어공학 전문위원회는 ISO/IEC JTC1/SC7의 대응활동을 수행하고 있으며 소프트웨어 산업 국가표준은 169건이 제정(1999년 4월 기준)되었다

표준개발 기관으로는 한국전산원과 한국전자통신연구원 등이 있다. 한국전산원은 공공부문 정보시스템 방법론 보급 및 정보화 표준측면에서 각종 지침을 개발하여 제공하고 있으며, 정보시스템 운영관리 지침(안), 정보화사업 감리 및 평가 관련 지침 등을 개발하고 있다.

ETRI는 ISO/IEC JTC1 중심의 국제표준화 활동과 OMG 등의 사실표준 연구그룹활동과 함께 문서화, 품질보증 등의 소프트웨어 개발환경 기준 표준연구, 데이터 표현, 검색, 저장기술 표준연구를 위한 CDIF(Case Data Interchange Format)연구, 객체기술 표준연구 등을 수행하고 있다.

VI. 결론

소프트웨어공학은 최근의 다변화되고 복합 구조화된 사회 활동 체제 속에서 기술 저변 확대에 필요한 정보기술 시스템의 효율성을 제고하고 생산성을 향상시킬 수 있는 공학적 방법론과 개발 도구를 만들어내기 위한 분야이다. 여러 하드웨어 자원의 유용한 이용을 위하여 응용성이 강한 소프트웨어 시스템들이 또한 구축되고 있다. 이러한 시스템들의 투자 효과를 증진시키기 위해서는 소프트웨어공학적 방법론이 개발되어야 한다.

소프트웨어 개발 방법론은 과거의 시간적 흐름에 따른 선형 변화 과정으로부터 실물 세계에 적합하고 휴먼 인터페이스 성향이 강한 다중 변화와 접속성, 모듈의 독립성, 재사용성을 감안한 부품화 등의 개념으로 발전되어 오고 있다.

최근 소프트웨어공학의 중요한 개념으로서 객체지향으로부터 전향된 컴포넌트웨어가 새로운 차세대 소프트웨어로서 각광을 받고 있고 마이크로소프트를 비롯한 이 분야 유수의 업체들이 이미 기존의 시장 세력을 밀바침으로 자사의 상품이 시장표준화되도록 경쟁을 가열시키고 있는 추세이다.

우리나라에서도 이러한 추세를 반영하여 소프트웨어공학의 핵심기술 분야 투자에 주력하고 있다. 앞으로 소프트웨어공학 표준화 기술개발에 있어서 유의하여야 할 것은 소프트웨어 객체간의 안정적인 통신성 확보와 용이한 플랫폼 적용이 이루어질 수 있는 기존 운영 환경과의 호환성 및 확장성이 갖추어져야 한다는 것을 들 수 있다.

또한 소프트웨어공학의 이론과 실체가 현실적 생산성과 대중성을 증진시켜 나가기 위해서는 사용자간의 기술적 이용 방식의 벽이 완화되고 기존 시장 환경을 기반으로하여 주요 시스템 및 객체지향 도구들을 공통적으로 수용할 수 있는 차원에서 적절한 시장 표준화가 꾸준히 진행되 나가야 할 것이다.

참고문헌

- [1] 백인섭 "소프트웨어공학 표준 기본 체제 연구", 한국전산원, pp.5-6 1991
- [2] 양해술, 이하용, 황석형, "소프트웨어 기술의 변혁과 발전", 정보처리 제7권, 제1호, pp.63-70, 2000.1.
- [3] 한국정보통신기술협회, "표준화 정보분류체계", pp.97-88, 1999.12.
- [4] 채병수, "소프트웨어 산업의 해외시장 개척 활성화 정도의 결정 요인에 관한 실증연구", 정보통신연구진흥, 제4호, pp.6-10, 2000
- [5] 선우종성, 이병만, 이주현, 김성근, "정보화 사업 관리 표준화 방안 연구", 한국전산원, pp. iii, 1999
- [6] <http://www.tta.or.kr>
- [7] <http://standards.ksa.or.kr>
- [8] Clemens Szyperski, Component Software -Beyond Object-Oriented Programming, ADDISON-WESLEY, 1999
- [9] 전자신문, "기획:애플리케이션 개발 컴포넌트 조립형 통합개발방식 각광", 컴퓨터월드 176, 1998.6., pp.234-237
- [10] 윤현식, "특집:컴포넌트에서 분산객체까지, 칩술에 배부르기", 마이크로소프트웨어 177, 1998.7., pp.248-255
- [11] 권오천외, "컴포넌트 기반 개발기술 검토 및 동향, 주간기술동향, 한국전자통신연구원
- [12] 홍기형, 서동수, "차세대 웹에서의 컴포넌트 소프트웨어", 정보처리 Vol.6, No.3, 1999.5., pp.45-51
- [13] Francois Coallier, et al., SOFTWARE ENGINEERING, MANAGEMENT REPORT AND BUSINESS PLAN FOR ISO/IEC JTC1/SC7, N1772, 1997-08-28