

실용적 S/W공학 효과의 재인식을 위한 Best Practices 활용 사례연구

장주관*, 박상엽*, 최정일*, 김기정*, 박재형**

*LG CNS 기술서비스부문 품질공학센터, S/W공학센터, **LG CNS 기술대학원

A Case Study on the Enterprise application of Software Engineering's Best Practices for a new understanding of Practical Software Engineering

Chang Joo-Kwan*, Park Sang-Yeob*, Choi Jeong-Il*, Kim Ki-Jung*, Park Jai-Hyoung**

*LG CNS Quality, S/W Engineering Center, **LG CNS University

E-mail : jkchang@lgcns.com, saypark@lgcns.com, jichoi@lgcns.com, gjungkim@lgcns.com,
hyoungpark@lgcns.com

요 약

현재 IT서비스업체에서 주요사업으로 수행중인 S/W의 개발분야에서 S/W공학은 마치 현실과 동떨어진 이론, 지식으로만 인식되고 실제 S/W공학의 효과를 인식하고 활용하는 사례가 극히 드문 것이 현실이다. LG CNS에서는 이러한 현실에서 실용적인 S/W공학을 재정의하고 이를 현장에 적용시키기 위하여 사업수행을 통해 얻은 S/W공학 측면의 사례를 발굴하여 이를 전사적으로 확산시키는 활동을 진행하고 있다. 본 논문에서는 IT서비스의 전사적 측면에서 실용적인 S/W공학의 효과성을 재인식시키기 위한 방법론 측면의 접근사례를 소개한다.

1. 서론

1-1. 연구배경 및 목적

소프트웨어 공학이란 소프트웨어의 품질과 생산성을 향상시키기 위하여 사용자의 요구사항을 체계적으로 분석하여 설계 및 구현, 구현된 시스템의 시험 그리고 유지보수 및 폐기 시까지 소프트웨어 전 생명주기에 걸쳐 이루어지는 체계적인 접근법을 말한다[1][2]. 따라서 소프트웨어 시스템을 효율적으로 개발하고 관리하기 위해서는 소프트웨

어 공학과 같은 체계적인 방법론 및 프로세스가 필수적이라 할 수 있다.

전체 산업에서의 소프트웨어가 차지하는 비중과 중요성이 증대되어가고 있으며 하드웨어의 기능을 소프트웨어가 대체하여 기능을 제공하는 추세로 발전하고 있다[3]. 이런 환경에서 IT서비스 역시 소프트웨어 개발을 위한 다양한 방법론과 지원도구를 적용하고 있다. 그러나 국내의 소프트웨어 공학기술은 선진국과 기술수준이 5~8년의 기술격

차가 존재하며 소프트웨어 공학 기술 전반에 걸친 기술에서 비교적 열세에 있다. 뿐만 아니라 소프트웨어 공학의 적용은 장기간에 걸쳐서 지속적인 노력이 필요한 형편이지만 국내 IT 서비스업체들은 특정 도구 등 시스템 개발에 단기적으로 투자하고 효과를 보려는 잘못된 인식으로 인하여, 소프트웨어 공학의 필요성을 알지 못한 것이 더욱 큰 문제이다[4].

이러한 문제해결을 위해, 본 연구의 목적은 IT 서비스 업체의 전사적 측면에서 실용적인 소프트웨어 공학의 효과성을 재인식하기 위해 Best Practice 사례 발굴, 콘텐츠 작성, 교육과정 개발 등 활용 단계별로 새로운 접근방안을 제시하는데 있다.

1-2. 연구범위 및 방법

본 연구에서는 IT서비스 업체에서 실용적인 소프트웨어 공학의 효과성 재인식을 위한 접근방안을 제시하고자 한다.

이를 위해 본론의 2-1에서는 소프트웨어 공학관련 업무성과의 문제를 다루고, 2-2에서는 문제해결을 위한 HPT측면의 접근방안을 알아본다. 마지막으로 2-3에서는 LG CNS에서 수행되었던 Best Practice 사례 발굴, 전사적 확산방안을 제시하고 3장에서 결론을 맺는다.

2. 본론

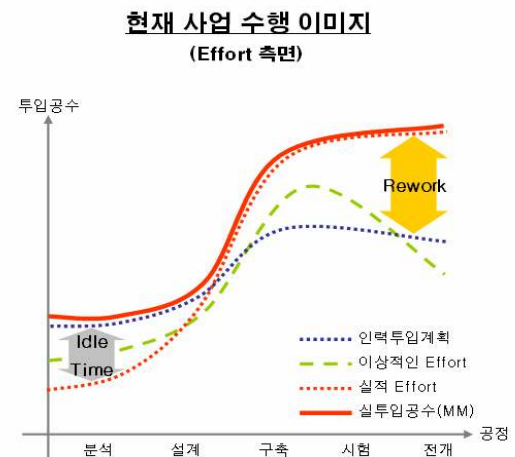
2-1. 소프트웨어 공학관련 업무성과 측면의 문제

그동안 소프트웨어 분석, 설계, 구축, 시험, 전개에 이르는 소프트웨어 공정에 있어서 기존에 검증된 소프트웨어 공학기법을 적용하려는 시도는 많은 IT서비스 업체에서 진행되어왔다. 그러나 그러한 노력에도 불구하고 현재까지 많은 프로젝트가 납기를 달성하지 못하고 주어진 예산범위를 벗어나는 경우가 자주 발생하고 있다. 특히 프로젝트 후반부에 있어서 과도한 재작업, 요구사항의 불명확성 등으로 인하여 소프트웨어 개발자들은 주말 및 야근 근무를 하고 있으며, 점차 소프트웨어 산업은 두뇌집약적이 아닌 노동집약적인 사업

으로 인식되고 있는 것이 현실이다.

소프트웨어 개발공정에 있어서 현재의 사업을 수행하는 모습은 <그림1>과 같이 초반에 낭비시간(Idle Time)이 발생하며 후반부에는 재작업(Rework)으로 인하여 인력투입계획보다 실제 투입되는 공수(Effort)는 늘어나고 있다.

<그림1. 소프트웨어 개발공정의 사업수행 이미지>

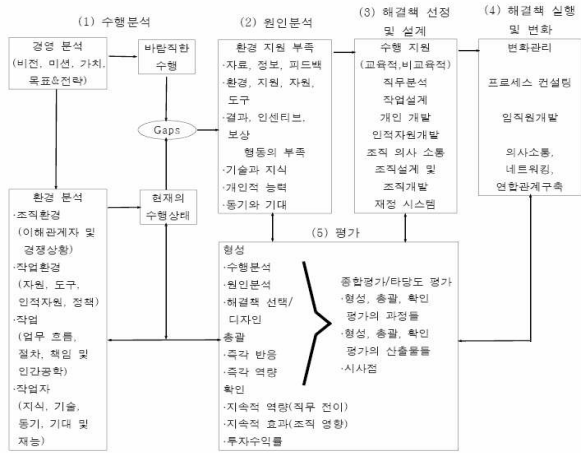


앞서 살펴본 소프트웨어 공학의 다양한 기법을 적용함에도 불구하고 사업수행상에서 발생하는 성과의 미흡은 어떤 요인에서 발생하는 것일까? 이러한 문제에 대한 답을 얻기 위해 본 사례연구에 적용하였던 HPT(Human Performance Technology)라는 개념의 이해가 필요하다.

2-2. 문제해결을 위한 HPT 접근방법의 이해

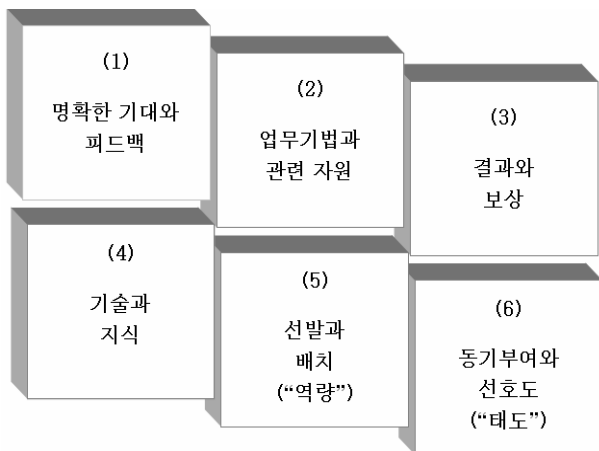
HPT(Human Performance Technology)란 사람들의 성과문제에 있어서 성과를 개선시키기 위한 바람직한 상태와 현상태와의 격차를 체계적으로 분석하여 교육을 포함한 성과개선에 효과적인 다양한 해결책을 제공함으로써 성과목표(Business Results)를 달성하도록 하는 체계적인 접근법이라고 할 수 있다[5]. HPT의 산실이라고 할 수 있는 ISPI(International Society for Performance Improvement)에서는 이러한 성과공학을 보급, 확산시키고 현실적용을 용이하게 하기 위해 모형[6]을 제시하였는데 이는 다음과 같다.

<그림3. HPT(성과공학)의 프로세스 모델>



위 모델에 의하면 바람직한 수행과 현재의 수행 상태 사이의 차이를 설명할 수 있는 원인분석을 하게 되는데, 이때 유용한 것이 바로 6 Box Model이다[7]. 이는 업무성과(Business Results)를 만들어내는 행동들에 영향을 미치는 요인을 6가지로 나누어 설명하는데 매우 유용한 도구이다. 여기서 (1)명확한 기대와 피드백, (2)업무기법과 관련자원, (3)결과와 보상은 환경적 요인이고 (4)기술과 지식, (5)선발과 배치, (6)동기부여와 선호도는 개인적 요인으로 분류해볼 수 있다. 기존에 우리가 교육을 통해 얻는 (4)기술과 지식이 아무리 많더라도 나머지가 부족하다면 업무성과를 만들기 어렵다는 것을 잘 설명해주고 있다. 즉, 통합적 측면의 문제인식에 매우 유용하게 활용할 수 있다.

<그림4. 6 Box Model의 구성>



앞서 살펴보았던 소프트웨어 공학의 업무성과측

면의 문제를 위 6 Box Model을 적용하여 본다면 업무성과에 있어서 Gap발생을 아래와 같은 장애요인/부정적인 행동요인으로 설명될 수 있다.

<그림5. 6 Box의 장애요인/부정적 행동요인>

1. 명확한 기대와 피드백 • 기대되는 업무 아웃풋에 설명 미비. 업무 목표 측정 안함 • 업무 수행방법에 대한 정보, 기대사항이 불분명하거나, 모순 되었거나, 제시되지 않음	2. 업무기법과 관련 자원 • 불충분한 인적자원, 자금, 시간, 지원사항, 장비, 또는 다른 자원	3. 결과와 보상 • 탁월한 성과에 대한 인센티브 없음 • 제대로 일을 수행하여도 관리거나 동료들로부터 얻게 되는 비공식적이며 부정적인 사회적 결과
4. 기술과 지식 • 업무성과 향상에 기여하지 못하는 교육, 학습방법 • 우수 성과자의 특성이 근거하지 않은 교육	5. 선발과 배치 • 직무상 요구되는 역량과 구성원 역량에 상이 • 역량과 무관한 승진	6. 동기부여와 선호도 • 일한만큼 인센티브나 보상이 주어지지 않음 • 구성원의 선호도를 고려하지 않은 업무 할당, 직무 배치

즉, 현장에서 품질과 생산성 향상을 위한 소프트웨어 공학의 적용이 바로 어떻게 실행되어야 하는지에 대한 1) 명확한 기대와 피드백이 없어서이고 2) 이와 관련된 업무기법과 관련자원도 불충분하다. 또한 3)과 같이 성과를 잘 내더라도 그에 따른 보상은 거의 없고 4)를 통하여 올바른 기술과 지식을 습득하였다 하더라도 5) 선발과 배치를 통해 요구되는 역량 대비 구성원의 역량이 떨어지므로 앞에서 보았던 성과의 Gap이 발생하게 된다.

따라서 앞에서 살펴보았던 HPT측면의 6 Box Model을 이용한다면, 다음과 같은 해결안을 모색해볼 수 있다.

<그림6. 6 Box의 촉진요인/긍정적 행동요인>

1. 명확한 기대와 피드백 • 부여된 업무의 아웃풋에 대한 기대사항 설명, 업무 목표 측정 • 어떻게 업무수행을 해야 하는지에 대한 명확하고 적절한 정보 또는 기대사항	2. 업무기법과 관련 자원 • 최적의 작업 아웃풋을 내기 위해 고안된 기법, 자원, 시간, 재료, 물리적 환경 • 관리자, 리더, 코치, 전문가에 대한 접근 방법	3. 결과와 보상 • 성과에 근거한 충분한 금전적 지원 • 제대로 실행되는 보상계획 • 성공에 대한 비재무적 보상과 인정
4. 기술과 지식 • 집합교육, 자기 학습, 동료학습, 관리자/코치의 학습 등을 통해 가장 효율적이며, 효과적인 방법으로 배울 수 있는 기회 • 우수 성과자의 활동을 유지/권장하는 교육(Best Practice 선정 및 확산)	5. 선발과 배치 • 구성원과 직무간 역량과 특성을 매칭시키는 것 • 효과적인 선발 프로세스와 기법	6. 동기부여와 선호도 • 구성원들을 동기 부여시키는 인센티브, 보상 - 구성원들은 가능한 것을 원함 • 구성원들의 선호도를 고려한 직무 배치 및 업무 할당

(1)명확한 기대와 피드백 : 소프트웨어 개발규정을 통하여 품질과 생산성있는 프로젝트를 만들어내기 위한 가이드를 제정한다.

(2)업무기법과 관련자원 : Best Practice를 통하여 검증된 적용기법을 재사용이 가능하도록 한다.

(3)결과와 보상 : 성공적으로 프로젝트를 수행한 PM에게 부여하는 프로젝트 종료평가요인에 품질과 생산성 요인을 추가한다.

(4)지식과 기술 : Best Practice 선정 및 발굴을 통하여 우수한 성과활동을 전파하고 확산한다.

(5)선발과 배치 : 역량있는 PM 및 리더를 선정하는 기본요건에 Best Practice관련 사례를 습득했는지 여부를 평가하여 적절하게 배치한다.

(6)동기부여와 선호도 : 위 5가지 활동들이 병행해서 진행된다면 자연스럽게 동기부여 및 선호도 측면에서 긍정적인 결과를 얻을 수 있다.

이러한 6가지의 해결대안을 놓고 볼 때 (4)지식과 기술 분야의 Best Practice 선정 및 발굴에 대한 대안이 무엇보다 가장 시급하고 우선해야 할 과제로 볼 수 있다. 왜냐하면 프로젝트 현장에서 이미 입증된 Best Practice가 없다면 (1)명확한 기대와 피드백 그리고 (3)결과와 보상 측면에서도 현장에서 의미있는 변화를 이끌어내기가 부족하기 때문이다. 그래서 (4)기술과 지식 분야의 해결안을 우선적으로 실행에 옮기기 위해 LG CNS의 소프트웨어공학센터와 기술대학원이 협업하여 현장에서 성공했던 소프트웨어 공학 관련 사례를 발굴하는데 초점을 두게 되었다.

2-3. 소프트웨어 공학의 Best Practice 발굴, 확산

그동안 수행해왔던 많은 소프트웨어 개발사례 혹은 프로젝트에서 우수한 Best Practice를 선정하는 방법에는 2가지 측면의 고려사항을 염두에 두었다.

첫번째, 현장에서 소프트웨어 개발자들에게 충분한 가치(Value)를 줄 수 있었는가? 두번째, 소프트웨어 개발공정과 관련하여 반복적으로 적용가능한 사례인가? 이를 위해 사내의 소프트웨어 공학 전문가에게 의뢰하여 추천할만한 사례를 수집하였고 Focus Group Interview를 통하여 위 2가지 측면의 기준을 갖고, 총 3가지 Best Practice를

선정하게 되었다.

<그림7. 소프트웨어 개발사례 분류 및 명단>

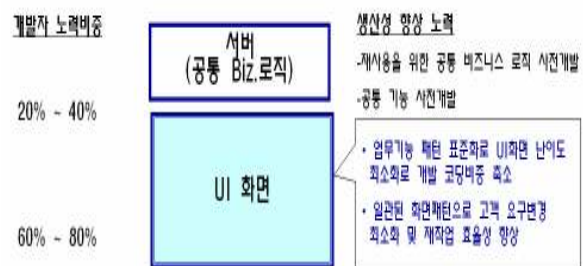
소프트웨어 공학 Best Practice 분류	대상 프로젝트명	SME 명단	역할
2.1 표준화/재사용 사례	00 컨소시엄 0000 시스템 구축	박상엽 차장 장주관 부장	아키텍트 사례개발
2.2 자동화/생산성 향상 사례	00 건설(주) 물류부문 통합시스템 구축	최정일 과장	모델러
2.3 재작업 최소화 사례	00 통합재정 정보 시스템	김기정 과장	시험평가

최종적으로 선정된 Best Practice의 내용은 아래와 같다.

첫번째, 표준화/재사용 사례는 패턴중심의 개발 (Pattern Driven Development;이하 PDD)을 적용한 우수개발법에 대한 내용이다. 사례에 대한 배경을 살펴보면, 해당 프로젝트에서는 단납기 사업에서 개발자의 반복적인 업무부하 영역이 비즈니스 로직을 공통 모듈화하여 해결하려고 하는 서버 계층(Layer)보다 사용자 인터페이스 계층(User Interface Layer)에서 고객 요구사항에 관련된 커뮤니케이션이 급증하게 되었던 상황이었다.

더구나 해당 프로젝트에서는 개발된 프로그램을 각각의 사이트(Site)에 맞게끔 커스터마이징(Customizing)을 해야하는 제약사항이 추가되어 일반적인 공공분야의 프로젝트보다 소프트웨어 개발단계 이후의 활동이 더욱 요구되는 상황이었다. PDD는 이런 환경에서 소프트웨어 아키텍처 측면에서 기본적인 업무기능 분석과 더불어 최종적으로 화면설계 결과물 분석을 통하여 사용자 인터페이스(User Interface) 패턴을 정의하였고, 여기서 한단계 더 나아가 재사용성을 높이기 위해서 표준화 작업을 진행하였다.

<그림8. 개발자의 노력비중 대비 생산성 향상노력>

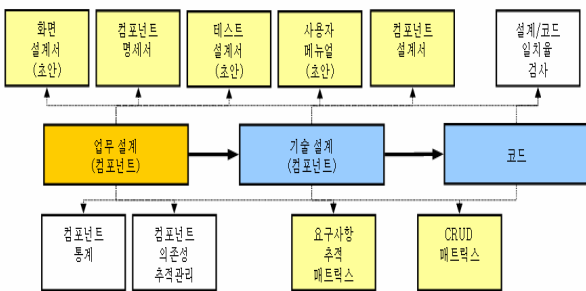


이렇게 PDD는 일관된 화면 패턴으로 개발자의 업무수행시간 중에서 고객과의 많은 커뮤니케이션을 줄여주고 User Interface 화면의 난이도를 최소화하고 소프트웨어 개발단계 코딩비중을 축소할 수 있다.

두번째, 자동화/생산성 향상 사례는 MDA(Model Driven Architecture)를 적용했던 우수개발법에 관한 내용이다. 사례를 적용했던 프로젝트에서는 단납기 내에 CBD(Component Based Development)를 적용하고 이를 통하여 고객의 업무혁신을 통한 가치까지를 보여주어야만 하는 곳이었다.

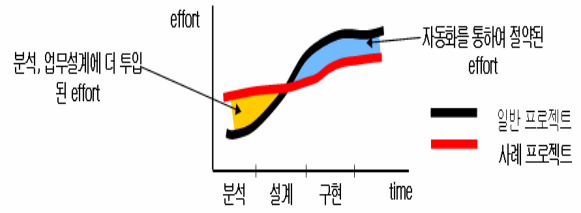
해당 프로젝트에서는 MDA를 통하여, 소프트웨어 분석/설계 결과를 텍스트 형식의 문서로 표현하기 보다는 소프트웨어 모델(UML: Unified Model Language)을 통해 표현함으로써 소프트웨어 분석/설계작업 자체의 생산성 향상은 물론 소프트웨어 모델을 활용하여 품질검증, 문서생성, 모델생성, 코드생성 등의 분야를 자동화함으로써 소프트웨어 개발공정의 자동화(코드생성률: 70%)를 극대화시킬 수 있었다. 또한 설계와 코드의 일치여부 검사, 컴포넌트 의존성에 대한 추적관리 등의 모델 통합 및 관리를 자동화함으로써 모델의 품질을 향상시켰다.

<그림9. MDA를 적용한 개발과정>



자동화를 통하여 절약된 Effort를 분석 및 업무설계 작업에 투입함으로써 정확하고 상세한 업무 분석을 토대로 선진 프로세스를 적용할 수 있었다.

<그림10. MDA를 적용시 공수(Effort)측면 효과>

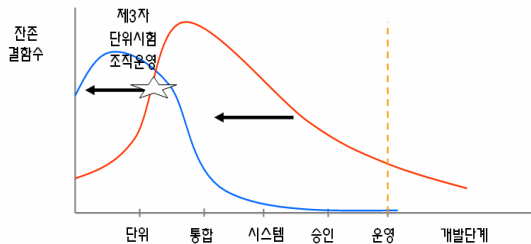


또한 컴포넌트 정의 및 명세화 작업을 충실히 함으로써 높은 품질의 컴포넌트 개발이 가능했다. 이로 인하여 IT를 통한 업무혁신 및 소프트웨어 품질 측면에서 모두 고객을 만족시키게 되었다.

세번째, 재작업 최소화는 제3자 테스트를 조직적으로 적용하였던 우수개발법에 대한 내용이다. 사례에 대한 배경을 살펴보면, 해당사업은 단계별 사업으로 기존에 개발된 프로그램의 품질상의 이슈로 인하여 고객의 기대치는 매우 높을 수 밖에 없는 상황이었고, 사용자 테스트 이전 단계에서 모든 결함을 제거하고 또한 발견된 결함은 모두 수정이 되어야만 하는 제약사항이 있었다.

해당사례에서는 소프트웨어 단위 시험 단계에서 제3자 단위시험 조직을 운영하여 구축초기부터 결함의 발견 및 수정을 용이하게 만들었다. 통상 개발자가 직접 수행하는 단위 테스트는 시험에 대한 커버리지를 낙관적으로 바라보는 깨끗한 테스트(Clean Test)가 되기 쉬운 것이 현실이다. 왜냐하면 개발자가 직접 수행하다 보니 본인 시스템의 신봉자가 되어 프로그램이 잘 작동되는 것만 시험하기 때문이다. 그렇기 때문에 희생되는 것은 역시나 품질이다.

<그림11. 제3자 단위시험적용을 통한 효과>



그러나, 해당사례에서는 제3자 단위시험 조직을 운영하여 프로젝트의 체계적인 단위시험 활동실시

를 유도하고, 초기에 결함을 발견, 재작업을 최소화시킬 수 있었다. 그 결과 최종적으로 99.8%의 사용자 시험적합률을 달성하였다.

이렇게 우수한 성과로 검증된 3가지의 Best Practice는 직접 해당 프로젝트에 관여하였던 소프트웨어 공학센터 전문가에 의해 초안이 작성되었고, 해당분야의 전문가 검토를 통하여 수정 및 보완되었다. 뿐만 아니라 실제 프로젝트에 효과적으로 적용되기 위해 소프트웨어 개발관리에 주요한 의사결정자인 프로젝트 관리자를 대상으로 하는 필수 교육에 콘텐츠로 활용되고 있다.

이러한 활동을 앞서 살펴본 6 Box Model에 근거하여 해석해본다면 (1)명확한 기대와 피드백 측면에서는 품질과 생산성 향상을 위한 소프트웨어 개발규정을 준수하도록 가이드하고 있다. 또한 (2)업무기법과 관련자원 측면에서 PDD와 MDA는 전사적으로 재사용 가능하도록 패키지화되어 있으며 (3)결과와 보상 측면에서도 프로젝트 종료평가 기준에 품질과 생산성 향상에 대한 내용을 추가하여 운영 중에 있다. 마지막으로 (5)선발과 배치 측면에서도 일정 규모 이상의 프로젝트에 대해선 위 교육을 반드시 이수해야만 사내 PM으로 인증 및 선발될 수 있도록 규정화되었기에 전사적으로 프로젝트 관리자들에게 Best Practice가 유용하게 활용될 수 있는 기반이 마련되었다고 볼 수 있다.

3. 결론

지금까지 소프트웨어 공학관련 업무성과의 문제를 다루고, 문제해결을 위한 HPT측면의 접근방안, 그리고 실행측면에서 LG CNS에서 수행되었던 Best Practice의 사례발굴, 전사적 확산방안을 알아보았다. 무엇보다 전사적 측면의 품질과 생산성 향상을 위해선 Best Practice를 만들어 나가는 프로젝트 현장, 품질 및 소프트웨어공학센터와 같은 전문조직, 그리고 기술대학원과 같은 인사 및 교육훈련 관점의 삼위일체된 전사적 선순환 메커니즘이 필요하다.

국내 IT서비스 선두업체 중의 하나인 LG CNS

에서는 이렇게 실용적인 소프트웨어 공학을 현장에 적용시키기 위해서 사업수행을 통하여 얻은 표준화/재사용, 자동화/생산성 향상, 재작업 최소화의 대표적인 Best Practice를 발굴하여 이를 확산시키는 활동을 지속적으로 전개할 예정이다.

이는 향후 글로벌 IT기업과 대등한 수준에서 경쟁하기 위해, IT서비스 업체가 소프트웨어 공학분야에서 전문성을 강화하고자 하는 목적을 달성하는데 하나의 방향성을 제시할 수 있을 것이다.

[참고문헌]

- [1] I. Sommerville, "Software Engineering – Seventh Edition", Addison Wesley, Boston, 2004.
- [2] IEEE Computer Society, "SWEBOK: Software Engineering Body of Knowledge", IEEE Computer Society, 2005.
- [3] 한국정보산업연합회, "해외기업의 임베디드 소프트웨어 개발역량 강화사례", 한국정보산업연합회, 임베디드소프트웨어산업협의회, 2006.
- [4] ETRI, "소프트웨어공학 기술 및 시장동향", 한국전자통신연구원, 소프트웨어 연구소, 2001, 2002.
- [5] 정재삼, "수행공학의 이해(기업교육 엔지니어링의 틀)",교육과학사,2006
- [6] D.M Van Tiem, J.L. Mosley, J.C. Dessinger, "Fundamentals of Performance Technology",ISPI,2000
- [7] Carl Binder, "Six Box Model", "http://www.SixBoxes.com/sixboxesgraphic.html", Binder Riha Associates,2005