2000 년 한국정보처리학회 추계 학술발표논문집 제 7 권 제 2 호

시스템 하우스에서의 소프트웨어 프로세스 개선 경험

조동환
LG 이노텍
e-mail : dhjo@lginnotek.com

Software Process Improvement Experience in the System House

Dong-Hwan Cho
LG Innotek R&D Lab.

요 약

제품시스템 전체에서 소프트웨어가 차지하는 비중이 압도적으로 커지고 있어 소프트웨어 품질이 제품 품질 전체를 좌우하게 되었다. 하지만 소프트웨어는 하드웨어 제품과는 다른 특성을 가지고 있어 제품의 품질을 쉽게 측정할 수 없다. 그러므로 제품 중심의 개선보다는 소프트웨어를 개발/생산하는 프로세스의 품질을 높임으로써 생산되는 소프트웨어의 품질을 높이는 프로세스 개선을 통하여 품질향상을 꾀하여야 한다. 당사에서는 1998 년 말부터 CMM 에 기반한 프로세스 개선을 추진하였으며 특히 시스템 하우스에서의 소프트웨어 개선 경험을 가지게 되었다. 본 논문에서는 이러한 개선 경험을 요약하여 기술하였다.

1. 서론

최근 상업용이나 군사용 시스템에 관계없이 시스템 전체에서 소프트웨어가 차지하는 비중이 과거에 비하여 압도적으로 커지고 있다. 이렇게 시스템의 기능을 구현하는데 있어서 하드웨어보다는 소프트웨어를 선호하는 이유로서는, (1)고객의 요구사항 변경에 대하여 보다 쉽게 대응할 수 있으며, (2)향후 기능추가 같은 유지보수가 용이하다는 점을 들 수 있다. 과거에는 이러한 좋은 점에도 불구하고 소프트웨어로 구현하는 경우 수행성능(Performance)이 문제가 되는 경우가 많이 발생하여(특히 군사용 실시간 처리의 경우) 높은 비용을 감수하더라도 부득이 하드웨어로 구현할 수 밖에 없었다. 하지만 기술발달의 결과로 인해 소프트웨어를 처리하는 컴퓨터의 성능이 점차 향상되어 대부분의 경우 소프트웨어로 처리하는 경우와 하드웨어로 처리하는 경우의 차이가 거의 나지 않게 되었다. 이런 기술발달 추세와 소프트웨어 구현의 장점으로 인하여 시스템에서의 소프트웨어의 비중이 점점 증가하고 있는 것이며 그에 따라 소프트웨어가 담당하는 기능이 많아지게 되어 소프트웨어 자체의 복잡도도 과거와는 비교할 수 없을 만큼 증가하였다. 이제는 소프트웨어의 품질이 전체 시스템의 품질을 좌우하는 시대를 맞이하게 된 것이다.

LG 이노텍(舊 LG 정밀)은 레이더, 유도무기, 무전기 등의 군사용 시스템을 전문적으로 개발/생산하는 시스템 하우스로서 국가방위산업에 중추적인 역할을 수행하고 있다. 1990 년대 중반까지 당사의 시스템 사업의 실질적인 중심은 하드웨어에 있었으며 소프트웨어에 대해서는 다소 전문성이 부족한 형편이었다. 하지만 변화하는 추세에 발맞추어 경쟁력을 확보하기 위해서는 하드웨어 중심에서 소프트웨어 중심으로 개발 역량을 재편하여야 하였으며 이를 위하여 많은 자구책을 강구하였으나 그다지 성공적으로 수행되지 못하였다. 여러 가지 시도가 실패한 이유로는 (1)장기적인 관점에서 지속적인 노력을 투입하기 보다는 일과성의 활동으로 그치거나 (2)주로 컨설팅에 의존하여 외부의 노력으로 내부의 문제를 해결하려고 하거나 또는 (3) 문제의 본질을 파악하는데 실패한 경우를 들 수 있다.

소프트웨어 역량을 강화하고 품질을 개선하기 위해

서는 기존의 제품 중심의 품질 개선 운동과는 다른 방식의 개선이 요구되었다. 왜냐하면, 소프트웨어는 "눈에 보이고 만질 수 있는" (곧 쉽게 물리적으로 품질을 측정할 수 있는) 하드웨어 제품과는 전혀 다른 특성을 가지고 있으므로 같은 방식으로는 소프트웨어 제품의 품질을 개선하는 데에는 많은 어려움이 따랐다. 예를 들어, 품질을 측정하는데 가장 기본이 되는 결함(Defect)을 정의하는 데에도 정의하는 사람에 따라, 보는 시각에 따라, 프로젝트의 성격에 따라 서로 달라졌으며, 어떤 소프트웨어 제품이 품질이 좋은 지도 정의하기 어려웠다. 이러한 어려움으로 인해 소프트웨어 품질 개선은 제품 중심의 개선보다는 제품을 개발하고 생산하는 프로세스(업무) 중심의 개선이 주된 접근방법이 되었다.

이러한 기본적인 이해를 기반으로 1998 년 말부터 당사 연구소 소프트웨어 전문개발그룹의 발의에 의하여 소프트웨어 프로세스 개선 활동이 시작되었으며 본 논문에서는 현재까지 근 2 년간의 프로세스 개선 활동의 경험을 요약하여 향후 소프트웨어 품질 개선을 수행할 다른 개발 조직에 도움을 주고자 한다.

2. Capability Maturity Model for Software (S/W CMM)

개선 활동은 적용할 조직에 알맞게 최적화되어야 그 효과를 극대화 할 수 있다. 하지만 개선경험이 많지 않은 조직에서 처음부터 최적화된 개선활동을 계획하여 수행하기는 거의 불가능하며 통상 문제분석→해결안 개발→시범적용→결과분석을 통한 문제점 도출 등의 사이클을 반복하면서 많은 시행착오를 겪게 된다. 이러한 시행착오를 줄이기 위해서 과거 여러 조직에서 성공적으로 수행한 개선 활동과 결과를 수집하여 공통적인 특징과 장점을 뽑아 모델화하여 사용하게 된다. 산업계에는 1980 년 대 이후 Capability Maturity Model for Software(이하 CMM)[1], Bell Trillium, Bootstrap, TicketIt, NASA SEL, ISO/IEC 15504 Software Process Improvement and Capability dEtermination(이하 SPICE)[2] 등의 여러 가지 소프트웨어 개선 모델이 제시되었다. 그 중에서 CMM 은 1987 년 미국방성에서 납품업체의 소프트웨어 개발 능력을 객관적으로 평가하고 소프트웨어 품질 개선의 가이드라인을 제시하기 위해서 미연방예산으로 설립된 Software Engineering Institute(이하 SEI)에서 개발되었다. CMM 은 사실상 소프트웨어 프로세스 개선에 있어서 De Facto Standard 라 할 수 있을 정도로 널리 사용되고 있으며 특히 군사용 시스템 개발 업체에서 사용하기에 적합하게 구성되어 있다. 당사에서는 여러 가지 가능성을 고려하여 소프트웨어 품질 개선의 모델로써 CMM 을 선정하게 되었다. 또한, 개선 활동의 모델로는 SEI 에서 같이 개발한 IDEAL[3]을 사용하게 되었다. IDEAL 은 Initiating→Diagnosing→Establishing→Acting→Leveraging 의 첫 글자를 딴 것으로 기존의 많은 개선 조직에서 사용하고 있는 PDCA(Plan→Do→Check→Act)와 그다지 다르지 않으나 소프트웨어 프로세스 개선 추진 조직이 수행해야 할 일을 상세하고 기술하고 있다. 앞으로 이 글에서는 IDEAL 의 개선 활동 모델에 맞추어 개선 경험을 기술하게 된다.

3. Initiating Phase: 1999 년 1 월 ~ 1999 년 9 월

개선의 필요성이 제기되었다고 해서 저절로 개선 활동이 시작되는 것은 아니다. 이런 종류의 활동에는 초기 선구자가 필요하며 그 선구자는 필요한 자원을 얻을 수 있도록 경영층을 설득할 수 있어야 비로소 개선 활동이 시작될 수 있는 것이다. 모든 기업활동에는 경영층의 직/간접의 동의나 결정이 필요하지만 특히 품질 개선 같은 종류의 활동에는 경영층의 직접적이면서도 강력한 의지가 반드시 선행되어야 하며 추진 도중에 계속적으로 의지가 변치 않고 유지됨을 공개적으로 천명하여야 한다.

소프트웨어 전문개발그룹의 개선 추진 담당자는 정식으로 개선팀이 구성되기까지 9 개월(1999 년 1 월부터 9 월까지) 간에 걸쳐 경영층과 조직 구성원 전체에 대해 개선의 필요성을 설득하고 개선 의지를 확인하며 기본적인 자원을 획득하여 다음 단계로 나갈 수 있도록 준비하는 Initiating Phase 를 수행하였다. 이 단계에서는 우선 최고 경영층이 확고한 개선 방향을 결정하고 의지를 가질 수 있도록 많은 객관적인 자료와 대안을 정리하여 제시할 필요가 있었다. 경영층은 개략적으로는 당사의 문제점에 대해 파악하고 있었으나 객관적인 자료를 바탕으로 한 세부적인 분석자료를 제공함으로써 문제의 심각성을 보다 구체적으로 인식하고 자원 투입을 통한 추진 결정을 내리는 과정을 도와줄 수 있었던 것이다.

경영층에 대한 설득과 추진 의지 확보가 얻어지면 그 다음 단계로는 각 개발 조직의 관리자들을 설득하여야 한다. 이 과정 또한 매우 중요하다. 왜냐하면 실제로 개선 활동의 적용의 대상은 개발 조직일 수 밖에 없으므로 그 개발 조직의 긴밀한 협조 없이는 개선 작업을 원활히 수행해 갈 수 없기 때문이다. 서로 다른 이해관계를 가진 개발 조직은 이러한 개선안에 대하여 전반적인 수준에서는 동의하지만 실제로 자신들의 업무가 가중된다고 느끼게 되면 거부하는 경우가 많다. 이런 경우에는 선진 업체나 경쟁 업체의 개선 사례를 소개함으로써 설득할 수도 있으며 또는 현재 개발 조직의 문제점을 조목조목 지적함으로써 관리나 감독의 목적이 아니라 도움을 주기 위한 목적임을 주지시켜 설득할 수 있다. 당사에서는 여러 차례의 전체 관리자 회의, 워크샵 및 인터뷰 등을 사용하여 경영층과 관리층의 동의와 추진 의지를 확보하게 되었다. 그 결과 개선 과제를 전사적인 최우선 혁신과제로 등록하여 필요한 자원을 지원 받을 수 있게 되었다.

이렇게 획득한 추진 의지를 바탕으로 각 개발 조직은 개선을 추진할 인력을 차출하여 개선팀을 구성하였으며, 보다 효율적인 활동 수행을 위하여 CMM 에 대하여 전문적인 경험과 지식을 가진 외부 컨설팅 그룹의 도움을 받기로 결정하였다.

4. Diagnosing Phase: 1999 년 10 월 ~ 1999 년 11 월

Initiating Phase 를 종료하는 시점에서 외부 컨설팅 그룹에서는 다음 단계인 Diagnosing Phase 를 위하여 개선팀원을 대상으로 CMM 과 심사 스킬에 대하여 약 3 주간의 심사원 교육을 실시하게 되었다. 이 과정에서 팀 빌딩을 같이 실시하여 새로 구성된 팀원들간의 결속을 같이 도모하였다. 특히 각 팀원들이 CMM Level 2 와 Level 3 에 언급된 핵심 프로세스 영역을 자신의 전문적인 분야로 선정하여 심사 시 해당 영역의 전문가로 참여하도록 훈련하였으며 향후 개선팀원으로 활동 시에도 해당 분야의 전문가가 되도록 하였다.

심사를 수행할 팀원들에 대한 준비가 완료된 뒤에는 심사를 실시할 대상 프로젝트를 선정하고 심사계획을 수립하였다. 대상 프로젝트는 시스템 프로젝트 중 일정 규모 이상의 프로젝트이면서 소프트웨어 비중이 높은 프로젝트를 각 개발 조직별로 고르게 선정하였다. 이 과정에서 심사 대상으로 선정된 프로젝트의 반발이 있을 수도 있으나 심사의 목적이 업무 평가가 아니라 샘플링을 통한 조직 전체의 능력 평가임을 주지시켜야 한다. 심사 계획을 수립할 때에는 심사에 참여하는 심사원들의 역할과 책임, 일정, 산출물 등을 세심하게 수립하여야 하며 특히 심사 장소와 물품 조달 등 간과하기 쉬운 부분에 신경을 써야 한다. 심사는 심사 대상 프로젝트 팀원에게 심사목적과 심사과정, CMM 에 대한 간략한 소개, 개인에 관련된 사항의 비공개 원칙 등을 설명하는 것으로 시작된다. 당사의 경우에는 심사 전 워크샵 을 개최하여 전반부는 모든 소프트웨어 엔지니어를 대상으로 CMM 과 심사에 대해 소개하고 후반부에서는 심사 대상 프로젝트 팀원들을 대상으로 좀더 상세한 내용과 일정을 소개하였다.

심사는 3 단계로 진행되었는데 1 단계에서는 대상 프로젝트의 개발 산출물(문서 등)로 문서심사를 먼저 실시하여 해당 프로젝트의 상황에 대한 기본적인 정보를 수집함으로써 다음 단계의 기초적인 자료로써 활용할 수 있도록 하였다. 2 단계에서는 대상 프로젝트의 프로젝트 리더에 대해 개별 인터뷰를 실시하여 1 단계 문서심사 시 도출된 의문사항을 확인한 후, 모든 개발자를 각 기능조직으로 나누어 해당 기능조직의 개발자가 모두 참여하는 기능그룹 인터뷰를 실시하였다. 예를 들어, 형상관리 기능그룹 인터뷰에서는 심사 대상 프로젝트의 형상관리 담당자 모두가 참석하도록 하는 것이다. 특히, 2 단계의 프로젝트 리더를 대상으로 실시한 개별 인터뷰에서는 프로젝트 리더는 방어적인 입장을 고수하므로 사실이 약간 왜곡되는 경향이 있으나 여러 프로젝트 담당자가 모인 기능그룹 인터뷰에서 그러한 사항은 프로젝트 간에 교차확인이 되어 정확한 실상을 파악하는데 많은 도움이 되었다. 예를 들어, A 라는 프로젝트에 대해 문서심사나 프로젝트 인터뷰에서는 품질보증을 규정대로 수행하는 것으로 확인하였으나 기능그룹 인터뷰에서 해당 프로젝트 품질보증 담당자나 일선 개발자, 혹은 다른 프로젝트 팀원들에 의하여 실제로는 품질보증을 제대로 수행하지 않음이 밝혀지는 경우를 들 수 있다. 즉, 한가지 사실은 문서심사→프로젝트 리더 인터뷰→기능그룹 인터뷰를 통해 확인되므로 심사결과를 신뢰할 수 있었다.

마지막 3 단계에서는 심사팀원끼리 자신의 심사결과를 가지고 서로 공유하고 토론을 통해 일관된 하나의 결과로 취합하였다. 이 과정은 심사 전체에서 가장 시간이 많이 들고 노력이 많이 투입된 과정으로 서로 다른 심사원 각각의 생각을 하나로 일치시켜 심사원 모두가 인정하는 공통된 심사결과를 끌어내기 위함이다. 이렇게 3 단계로 진행된 약 3 주간의 심사를 거쳐 심사 결과 보고를 작성하여 경영층을 포함한 모든 소프트웨어 개발자에게 그 결과를 발표하는 것으로 Diagnosing Phase 를 마무리 하였다.

경영층, 관리자, 개발자는 발표된 심사 결과에 대해 처음에는 충격으로 받아들였다. 어느 정도 부족한 점은 인정하지만 수치화되어 발표된 만큼 심각하지는 않다고 항변하는 경우도 있었다. 하지만 바로 이 부분에서 심사의 필요성이 대두되는 것이다. 예를 들어, 우리의 몸 상태가 좋지 않다고 느끼기만 하는 것과 병원에 가서 종합진단을 받아 이상한 점을 발견하는 것 사이의 차이라고 할 수 있다. 결국 구체적이고 논리적인 의사의 진단에 대해서 심각하지 않다고 환자가 항변할 수는 없는 것이다. 마찬가지로 당사도 구체적인 지적사항에 대해서 결국 수긍을 하게 되어 Diagnosing Phase 의 목적인 (1)조직의 현재 상태를 명확히 파악하고 (2)조직 구성원에게 개선의 필요성을 스스로 느끼는 Buy-in 효과의 창출도 달성하였다고 평가되었다.

5. Establishing Phase: 1999 년 12 월 ~ 2000 년 1 월

Establishing Phase 는 심사 결과를 바탕으로 앞으로 무엇을 어떻게 개선할 것인가에 대한 개선 전략을 수립하고 계획하는 단계이다. 보통 심사결과가 도출되면 경영층은 도출된 문제점들에 대해서 신속히 해결책을 강구하여 해결하라고 요구한다. 하지만 개별 문제점에 대해 급히 미봉책을 사용하는 것은 문제의 본질을 해결하지 못하고 겉으로 드러난 현상만을 잠시 해결할 뿐이어서 그때가 지나고 나면 또다시 같은 종류의 문제가 발생할 수 있다. 그러므로, 개선 전략을 수립할 때에는 심사 시에 도출된 문제를 해결할 수 있는 근본적인 해결책을 개발하고 그 해결책이 기업의 중장기전략과 일관된 흐름을 유지할 수 있는지를 판단하여야 한다. 이론적인 모델로써 아무리 급한 과제라 할지라도 해당 기업의 전략이나 비전과 맞물려 돌아가지 못한다면 채택하는 것을 재고해야 한다. 당사에서는 개선팀은 기업의 중장기 전략과 비전을 고려하여 가장 효율적으로 자원을 투입하면서 개선 효과를 얻을 수 있는 개선 전략을 수립하여 경영층의 승인을 획득하였으며, 경영층을 공식적으로 Steering Committee 로 참여시켜 계속적인 전략의 검토와 상황에 따른 수정을 지시할 수 있도록 하였다.

개선 전략이 수립된 후에는 전략에서 도출된 각 세부 과제에 대한 세부 실행 계획을 수립하여야 한다. 이 세부 실행 계획은 경영층의 공식적인 승인을 받은 개선 전략과는 달리 내부적으로 수시로 변경될 수 있

으므로 각 핵심 프로세스 영역의 담당자가 스스로 수립하여 실행하도록 하였으며, 주기적으로 성과를 공유하며 팀원간의 토론을 통해 재수립 할 수 있도록 하였다. 이 때 실행의 진척도를 객관적으로 확인할 수 있도록 일정을 포함한 주요 측정 항목을 선정하여 주기적으로 진척상황을 취합하여 분석하는 작업이 반드시 병행되어야 한다. 이렇게 분석된 측정 항목은 내부적인 조율 목적 이외에도 경영층이나 다른 개발자들에게 개선 프로젝트의 추진에 대한 가시성을 제공하여 더욱 분발할 수 있는 계기를 제공한다.

6. Acting Phase: 2000 년 2 월 ~2000 년 11 월

도출된 개선안을 처음부터 모든 프로젝트에 적용하는 것은 상당한 위험을 안고 있다. 그러므로, 처음에는 일정기간 시범적용을 통해 개선안의 장/단점을 파악하는 것이 바람직하다고 생각된다. 이때 시범적으로 적용할 대상 프로젝트를 선정하는 것도 중요하다. 대상 프로젝트는 연구소의 다른 프로젝트를 대표할 수 있어야 하며 프로젝트 팀원들의 협조도 많이 필요로 하므로 반대 급부도 제공하여야 한다. 당사에서는 프로젝트 성격, 규모, 일정 등의 여러 가지 프로젝트 선정 기준을 개발하여 전체 프로젝트를 대상으로 필터링을 수행한 결과 전체 프로젝트의 10%정도의 프로젝트를 선정하였다. 선정된 프로젝트에 대해서는 개선안을 적용하는 대신에 프로젝트 업무에 다소간 개선팀의 담당자가 함께 참여하는 방식으로 추가적인 개선안 실행에 소요되는 추가적인 업무로드를 보상하여 주었다. 시범적용 기간은 2 월부터 6 월까지 5 개월 정도로 잡고 7 월에는 시범적용 결과를 참고하여 개선안을 정제하고 8 월부터 모든 프로젝트에 확대적용 하는 것이 개선 전략이었으며, 모든 시범적용 프로젝트에 대해서는 CMM 만족도와 그에 따른 업무성과 측정을 매주 실시하였다.

시범적용이 4 개월정도 경과한 2000 년 5 월에 시범적용 성과에 이상한 점이 발견되었다. CMM 을 기준으로는 만족도는 높아지는데 실제 업무 성과가 좋아지는 비율이 현저히 낮아지게 된 것이다. 개선팀은 그 이유에 대해 분석하고 Steering Committee 와 토의한 결과 소프트웨어 부분에 국한한 개선은 한계가 있다는 결론을 얻게 되었다. 처음에는 개선노력을 집중시키기 위해서 소프트웨어 부분에 한정하여 개선 활동을 하였으나 시스템을 연구개발/생산하는 시스템 하우스에서는 소프트웨어 부분만의 개선은 한계를 가질 수 밖에 없었던 것이다. 이를 해결하기 위해서는 개선 활동의 범위를 소프트웨어에 국한할 것이 아니라 연구개발 활동 전반으로 확대시켜야 하며 CMM 도 그에 맞게 수정하여 적용할 필요성이 대두되었다. 이러한 내용을 골자로 수정된 개선 전략을 수립하여 2000 년 6 월 경영층의 승인을 받기에 이르렀다. 수정된 전략을 기반으로 연구개발 활동 전반에 걸친 개선안을 2 개월간 작성하여 2000 년 9 월부터 소프트웨어 전문그룹 전체에 시범적용하고 그와 병행하여 2000 년 11 월부터는 모든 연구 개발 조직에 확대적용 하도록 하였다.

여기에서 "적용"이라 함은 CMM 에 맞추어 세심하게 조율된 승인된 업무 프로세스와 해당 업무 프로세스에 대한 교육/훈련, 실제 업무 적용 시 발생하는 문제점에 대한 해결책 제시 등의 여러 업무를 포함한다.

7. Leveraging Phase: 2000 년 12 월

2000 년 12 월에는 개선안을 적용/실시하고 있는 프로젝트에 대해 심사를 실시하게 된다. 이 심사의 목적은 (1)개선된 업무 프로세스가 얼마나 잘 지켜지고 있는가와 (2)잘 안 지켜지고 있을 경우 그 원인이 무엇인가, (3)Diagnosing Phase 와 비교하여 얼마나 개선되었는가 등을 파악하는데 있다. 여기에서 도출된 심사결과로 다음 사이클의 개선 활동을 시작할 수 있게 된다. 즉, IDEAL 사이클의 Initiating Phase 부터 다시 시작하는 것이다.

8. 결론

시스템 하우스에서는 보통 연구개발 프로세스와 생산 프로세스가 분리되어 있다. 즉, 대량 생산을 위한 시스템 제품은 연구소에서 연구개발 프로세스에 따라 시제품이 제작되며 그 이후에 공장에서 생산 프로세스에 따라 대량 생산되어 완제품이 나오게 된다. 하지만 소프트웨어는 연구개발이 끝나면 그것으로 생산이 끝나게 되는 특성을 가지고 있으므로 연구개발 프로세스에서 시제품이 아니라 완제품이 나오게 된다. 이러한 이유로 소프트웨어의 경우에는 연구개발 프로세스에서 하드웨어보다 철저한 품질 확인 기능이 필요하게 된다. 바로 이 부분이 시스템 하우스에서 소프트웨어 품질 개선이 어려운 부분이다. 하드웨어 관련 연구개발 프로세스의 혼란이 없도록 하면서도 소프트웨어의 품질을 최대한 높일 수 있도록 통합된 연구개발 프로세스를 개발하고 실행하며 유지하는 것은 상당히 어려운 작업이 될 수 밖에 없다. 이는 조직원이나 기업의 문화에도 많은 부분을 의지할 수 밖에 없으며 사실상 기업 활동 전반과 밀접한 관계를 맺게 된다.

CMM 을 개발한 SEI 에서 소프트웨어 영역 뿐만 아니라 시스템 공학과 제품 공학 등에서 사용하는 모델을 통합한 CMMI[4]를 개발한 의도도 같은 맥락에서 이해할 수 있으며, 기업내의 BPR, 6 시그마, CRM, 지식경영, 소프트웨어 프로세스 개선 등의 여러 개선 활동은 전체적인 기업 비전과 전략에 맞추어 잘 조율되어야 한다.

참고문헌

[1] Mark C. Paulk, Bill Curtis, Mary Beth Chrissis, Charles V. Weber "Capability Maturity Model for Software, Version 1.1", CMU/SEI-93-TR-24, Software Engineering Institute, 1993

[2] ISO/IEC JTC 1/SC 7 "Information technology – Software Process Assessment", ISO/IEC TR 15504, 1998

[3] Robert McFeeley, "IDEAL: A User Guide for Software Process Improvement", CMU/SEI-96-HB-001, Software Engineering Institute, 1996

[4] "CMMI-SE/SW Version 1.0", Software Engineering Institute, 2000