

특집기사

소프트웨어 프로젝트 관리에서의 품질보증 시스템의 프로세스 기술방식

양 해 술* 이 용 근** 허 태 경***

❖ 목

차 ❖

1. 서 언

2. 소프트웨어의 프로세스

3. 품질보증시스템의 프로세스 기술방식

4. 결 언

1. 서 언

최근 소프트웨어 프로젝트관리에서의 소프트웨어 프로세스에 대한 연구가 활성화되고 있다. 소프트웨어의 개발에 따라 작업을 요구정의, 명세작성, 설계, 코딩, 테스트 등 요구되는 제품의 직접/간접적인 개발작업에 관한 것과 자원 분배, 스케줄 조정, 품질보증 등 관리작업의 제어에 관한 것으로 나눌 수 있다.

그러나 지금까지 개발 프로세스의 기술에 대해서는 많은 연구가 이루어졌지만 관리프로세스에 대해서는 연구가 진행되지 못하고 있는 실정이다. 그 원인으로는 소프트웨어 개발에 있어서 관리대상이 대부분 조직이나 프로젝트마다 다르고 일반적인 모델의 작성이 어렵다는 점, 개발작업과 관리작업 모두를 프로세스로 기술하게 되면 프로세스도가 복잡하게 될 뿐만 아니라 실용적이지 못하다는 점 등을 들 수 있다.

관리작업에 대해서 개발작업과 관리작업을 '생산대상'으로 하여 함수로 가정하는 고단계함수를 이용하여 개발 프로세스와 관리 프로세스를 동일

한 레벨에서 정확하게 기술하려는 시도가 있지만 의미 정의가 비교적 복잡하고 직관적으로 전체를 파악하는 것이 곤란하다.

그러나 실제적으로 관리 프로세스의 사고는 소프트웨어 품질보증 시스템으로서 폭넓게 사용하는 것으로 나타나고 있다. 소프트웨어 품질보증 시스템이란 소프트웨어의 품질을 보증하기 위해 행하는 작업을 기술한 것으로 현재 일반적으로 이용되고 있는 소프트웨어 품질보증 시스템으로는 CMM(Capability Maturity Model)과 ISO 9000-3의 두가지를 들 수 있다. 이 가운데 CMM은 주로 미국에서 사용하고 있고, ISO 9000-3은 유럽에서 사용되고 있다. 현재 우리나라에서는 ISO 9000-3에 대한 관심이 높으며 많은 기업이 ISO 9000시리즈의 인정을 받기 위해 노력중에 있다.

본 고에서는 이들 품질보증 시스템을 기술하기 위한 소프트웨어 프로세스 기술법에 대해 설명하고 그 프로세스 기술 방법을 CMM과 ISO 9000-3에 적용하여 살펴보기로 한다. 이들 기술방식은 실제로 요구사항의 기술을 나열한 시스템 전체의 파악이나 시스템간의 비교가 곤란하다는 단점을 극복할 수 있다고 본다.

* 종 신 회 원 : 강원대학교 전자계산학과 교수

** 종 신 회 원 : 한림전문대학 전산정보처리과 강사

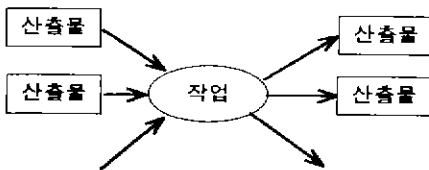
***정 회 원 : 한국통신 품질보증단 개발품질부 부장

2. 소프트웨어의 프로세스

2.1 소프트웨어 프로세스 모델

소프트웨어 프로세스를 (그림 1)과 같이 산출물을 기준으로 작업을 진행하여 새로운 산출물을 도출하는 형태로 구성하였다. 산출물은 작업에 의해 생겨나는 것으로 본다.

프로세스 기술방법은 관리작업을 고단계의 것으로 개발작업과 분리하는 것이 아니고 각 관리작업도 어떤 종류의 산출물을 생성하는 작업으로 동일 레벨로 취급하였다. 어떤 관리작업에 의해 제어되는 개발작업은 그 관리작업이 만든 산출물을 입력의 하나로 받아들임으로써 제어를 행할 수 있다는 생각과 모든 프로세스의 기술을 작업과 산출물의 흐름으로 나타낸다. 더욱이 기술의 용이성과 이해용이성을 위한 기술을 복수의 정밀도로 나누어 실시한다.



(그림 1) 소프트웨어 프로세스 모델

작업은 크게 개발작업과 관리작업의 2가지로 분류할 수 있으며 개발작업은 사용자의 요구명세를 만족하는 제품 작성의 작업이고, 관리작업은 제품 그 자체에 대한 것이 아닌 제품의 작성에 필요한 산출물을 작성하는 작업이다. 예를 들면, 사용자가 만족하는 문서의 작성은 개발작업이고 개발스케줄 등의 문서작성은 관리작업이다.

개발작업은 요구정의, 명세 작성, 설계, 코딩, 테스트 등을 포함하며 요구정의서, 명세서, 설계서, 소스 프로그램 등이 개발작업에 대한 입출력 산출물이 된다. 한편 관리작업에는 스케줄 관리나 품질보증 작업, 작업자에게 할당되어 있는 작업을 포함하며 스케줄표, 품질에 관한 매트릭스 데이터, 작업에 할당된 표 등이 관리작업의 입출

력 산출물이 된다.

따라서, 개발작업, 관리작업의 양쪽을 작업과 산출물의 흐름으로 기술함으로써 기술이 용이하게 되고 필요한 기술을 간단하게 행할 수 있다.

2.2 관리작업의 분류

소프트웨어 관리작업에 대해서는 종래부터 다양하게 분류되어 오고 있다. 예를 들면, 개발관리는 관리대상을 공정관리, 제품관리, 원가관리, 품질관리 등의 4가지로 분류하고 있다. 또한 일반적으로 광범위하게 이용되고 있는 관리대상의 분류로서는 QCD(Quality, Cost, Delivery)를 들 수 있다. 즉, 관리대상을 소프트웨어 개발에서 중요한 관리항목인 품질, 비용, 스케줄의 3가지로 분류하는 방법이다. 이외에도 4M 즉, Man, Money, Machine, Material에 의한 분류 등이 있다.

본 고에서는 소프트웨어 개발에 있어서 관리항목을 작업과 산출물의 2가지로 분류한다. 이것은 소프트웨어의 개발이 산출물을 이용하여 작업을 처리하고, 산출물을 작성하는 것으로 인식하여 관리 대상을 작업, 산출물의 2가지로 분류할 수 있다고 생각하기 때문이다.

다음에 작업과 산출물 각각의 관리항목에 대해 보다 상세히 분류하여 설명하기로 한다.

(1) 작업

작업의 범위에 속하는 관리항목은 스케줄, 협약, 표준 프로세스 등이 있다. 스케줄에 대응하는 관리항목으로서는 계획의 작성, 진척의 감시·평가, 계획의 조정 등을 생각할 수 있다. 협약은 프로젝트 실행시에 정해진 작업자간의 약속으로 관리작업으로서는 협약의 결정, 변경 등을 들 수 있다. 표준 프로세스는 조직에서 표준적인 작업의 진행하는 방법으로 관리작업에서는 표준 프로세스의 작성, 변경 등을 생각할 수 있다.

(2) 산출물

산출물의 범위에 속하는 관리항목으로서는 사람, 기술, 소스 등이 있다. 사람에 대한 관리작업으로서는 관리자의 고용, 교육, 훈련 등을 들 수

있다. 또한 기술에 관해서는 신규 기술의 도입과 그것에 부수적으로 딸려 있는 관리작업 등을 생각할 수 있다. 소스에 대한 관리작업으로서는 소스의 평가나 구성관리, 재이용을 위한 소스 집합의 관리 등을 들 수 있다.

2.3 프로세스의 기술 방식

실제 관리프로세스의 기술은 다음과 같은 방식으로 처리한다. 즉, (그림 1)과 같이 작업은 타원으로 표시하고 산출물은 박스 형태로 표시한다. 작업과 산출물 간에는 유향 에지(edge)가 존재하는데 이 작업과 산출물의 집합 및 그 유향 에지로 구성되어 있는 유향 그래프를 프로세스라고 부른다. 다만 하나의 작업은 그 자체만으로도 프로세스이지만 하나의 산출물은 프로세스가 아니다. 산출물은 작업과 이어져 처음부터 프로세스가 된다. 작업에서 산출물로의 에지는 산출물이 작업에 의해 생성되고 있는 것을 표시하고, 산출물에서 작업으로의 에지는 산출물이 작업에 사용되고 있는 것을 나타낸다. 그러나 작업에서 작업으로의 에지는 존재하지 않고 산출물에서 산출물로의 에지도 존재하지 않는다.

2.4 작업의 정확도

종래의 대부분의 관리작업이 대부분 프로세스로 기술되지 않은 원인의 하나로서 개발작업과 관리작업 모두의 프로세스를 기술함으로써 처리도가 복잡하게 되고 기술과 이해가 곤란하게 되

는 점을 들 수 있다. 이 문제를 해결하기 위해 본 고에서는 프로세스의 기술을 단위작업 클래스, 단계 클래스, 프로젝트 클래스, 조직 클래스 4가지로 나누어 처리하고 있다.

4가지 클래스간의 관계는 (그림 2)에 나타내었다. 즉, 조직 클래스에서 기술되는 프로젝트를 나타내는 작업을 보다 정확하게 기술하는 레벨이 프로젝트 클래스이고, 프로젝트 클래스에 있어서 각 공정(단계) 마다 처리되는 작업을 기술한 레벨이 단계 클래스이고, 프로젝트 클래스에서 기술되는 임의의 작업과 산출물의 관계를 보다 정확하게 기술한 레벨이 단위 작업 클래스이다.

다음에 각 클래스에 대해서 설명하기로 한다.

(1) 조직 클래스

조직 클래스는 처음의 프로젝트가 종료한 후 다음의 프로젝트를 개시할 때까지의 사이에 처리되는 프로세스를 기술하기 위해 사용된다. 이 클래스에 관리작업을 산출물의 관점에서 말하면 재이용을 위한 소스 집합의 관리를 들 수 있다.

(2) 프로젝트 클래스

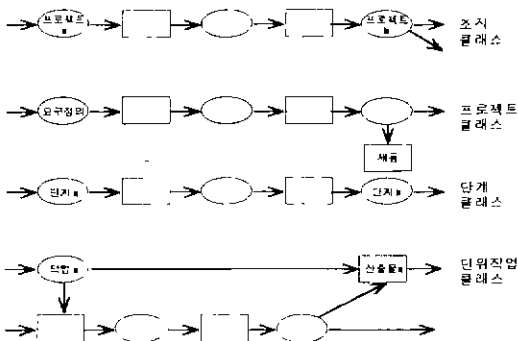
프로젝트 클래스는 처음 프로젝트에 관해서 처리되는 프로세스를 기술하기 위해 사용된다. 이 클래스에서 관리작업을 산출물의 관점에서 말하면 프로젝트에 있어서의 품질목표의 생성을 들 수 있다.

(3) 단계 클래스

단계 클래스는 설계와 보수라고 하는 프로젝트 클래스에 있어 각 공정마다 부여되어 있는 프로세스를 기술하기 위해 사용된다. 이 프로세스에서의 관리작업은 공정마다 처리되는 임출력의 확인을 들 수 있다.

(4) 단위작업 클래스

단위작업 클래스는 프로젝트에 있어 임의의 하나의 작업과 그 산출물의 부수적인 프로세스를 기술하기 위해 사용된다. 이 프로세스에 있어서 관리작업을 산출물의 관점에서 말하면 소스의 구성관리를 들 수 있다.



(그림 2) 각 클래스간의 관계

3. 품질보증 시스템의 프로세스 기술

3.1 CMM의 개요

CMM(Capability Maturity Model)은 소프트웨어를 개발한 회사와 조직의 프로세스가 어느 정도 성숙되었는지를 평가하는 지표이다. CMM에 의해서 조직의 현 상태를 평가하는 것이 가능하고 조직을 개선하는데에도 도움이 된다.

CMM은 미국방성의 요청에 의해 카네기 메론 대학의 SEI(Software Engineering Institute)의 Humphry 등이 중심이 되어 1991년에 Version 1.0이 작성되었다. 현재는 1993년에 작성된 Version 1.1이 사용되고 있다. CMM에서는 프로세스의 성숙도를 양제이트 등을 이용하여 Initial, Repeatable, Defined, Managed, Optimized의 5단계로 분류한다.

(1) 레벨 1 : Initial

공식적인 절차는 거의 없으며, 개발자는 편한 대로 여러 작업을 할 수 있다. 비록 규격이 있더라도 무시되고, 도구가 있어도 사용되는 것은 별로 없다. 생산효율은 낮으며, 견적을 추출하는 것도 어렵다. 외부에서는 개발자가 현재 무엇을 실행하고 있는지를 짐치럼 이해할 수 없다.

(2) 레벨 2 : Repeatable

조직에 있어서 과거의 경험을 기초로 의견이 같을 수 있다. 같은 의견에 따라 과거와 같은 방법으로 작업을 처리하는 것이 가능하다. 레벨 1에 비해서 생산성은 다소 상승하고, 견적도 통계적으로 허용된 범위내에서 추출이 가능하다. 그러나 각 작업은 아직 블랙박스 되어 있고 프로젝트 관리자의 능력에 많은 부분을 의존하고 있으며, 만약 관리자가 프로젝트 도중에 빠진다면 프로젝트는 붕괴된다. CMM 레벨 2의 목표는 <표 1>과 같다.

(3) 레벨 3 : Defined

작업과 작업 결과가 문서에 의해 정의되고 문서에 의해 관리가 가능하다. 작업결과는 외부로

<표 1> CMM 레벨 2(Repeatable)의 목표

CMM 레벨 2(Repeatable)의 목표	
(1) 요구관리	(목표 1) 소프트웨어에 배치된 시스템 요구가 소프트웨어 공학과 관리를 위한 기준은 평가하기 위해 제어된다 (목표 2) 소프트웨어의 시회, 제작, 작업이 소프트웨어에 배치된 시스템 요구와 일치한다
(2) 소프트웨어 프로젝트 계획	(목표 1) 소프트웨어 평가가 계획과 사후 조사에 사용되도록 문서화한다 (목표 2) 소프트웨어의 프로젝트의 작업과 협력이 계획되어 문서화한다 (목표 3) 관계하는 그룹과 개인이 프로젝트에 관한 협약을 동의한다
(3) 소프트웨어공학의 추적과 감시	(목표 1) 실제의 효과와 실행이 계획을 반영하고 있는지의 여부를 추적한다 (목표 2) 조사작업이 수행되고, 실제의 효과와 실행이 계획으로부터 벗어날 때 관리된다 (목표 3) 관계한 그룹과 개인이 의해 동기가 되지 않는 범위를 제한하고, 소프트웨어의 협약 변경을 처리하지 않는다
(4) 소프트웨어의 수주 관리	(목표 1) 설계자가 적합한 소프트웨어 수주자를 선택한다 (목표 2) 설계자와 수주자가 상호 협약을 동의한다 (목표 3) 설계자가 수주자가 진행중에 커뮤니케이션을 유지한다 (목표 4) 설계자가 수주자의 작업절차와 실행이 공약에 반영되어 있는지의 여부를 추적한다
(5) 소프트웨어의 품질보증	(목표 1) 소프트웨어의 품질보증 작업이 계획된다 (목표 2) 소프트웨어 프로젝트와 작업으로 표준, 절차, 요구가 객관적으로 변경된다 (목표 3) 영향받는 그룹과 개인이 소프트웨어 품질보증작업과 결과를 인식한다 (목표 4) 프로젝트에 있어서 해결되지 않은 복잡한 문제는 담당자에게 의해 처리된다
(6) 구성 관리	(목표 1) 소프트웨어 구성 관리의 작업이 계획된다 (목표 2) 선택된 소프트웨어의 사양 프로젝트가 식별되고, 제어되고, 이용된다 (목표 3) 식별된 소프트웨어의 사양의 프로젝트의 변경이 제어된다 (목표 4) 영향받는 그룹과 개인이 소프트웨어의 기준 상태와 내용을 인식한다

<표 2> CMM 레벨 3의 목표

CMM 레벨 3 : Defined에 있어서 Key Process Area의 목표	
(1) 조직의 프로세스 통합	(목표 1)인 소프트웨어 프로세스 개발과 개선의 작업이 조직을 통해서 조화를 이룬다 (목표 2) 이용되고 있는 소프트웨어 프로세스의 장점과 단점이 프로세스 표준에 적용하여 인식된다 (목표 3) 조직의 프로세스 개발과 개선작업이 계획된다
(2) 조직의 프로세스 정의	(목표 1) 조작성 위한 표준 소프트웨어 프로세스가 개발되고 보수된다 (목표 2) 조직의 표준적인 소프트웨어 프로세스의 사용에 관한 정보가 소프트웨어 프로젝트에 의해 수집되고 검토되어 이용되는 것이 가능하다
(3) 트레이닝 프로세스	(목표 1) 트레이닝 작업이 계획된다 (목표 2) 소프트웨어 관리와 기술적인 역학을 위해서 필요한 기술과 지식은 개발한 트레이닝이 있다 (목표 3) 소프트웨어 엔지니어링 그래프에 있어서 개인과 소프트웨어에 관계한 그룹이 자신의 역할을 실행하는데에 필요한 트레이닝을 받는다
(4) 통합 소프트웨어 관리	(목표 1) 프로젝트에 있어서 소프트웨어 프로세스가 조직의 표준 소프트웨어 프로세스에 적용하도록 버전으로서 정의된다 (목표 2) 프로젝트에 있어서 소프트웨어 프로세스가 정의되도록 프로젝트가 계획, 관리된다
(5) 소프트웨어 프로젝트 관리	(목표 1) 소프트웨어 엔지니어링의 임무가 정의, 총합되고, 항상 소프트웨어를 생성한다 (목표 2) 소프트웨어 임무의 프로덕트가 상호 통일성을 갖고 있다
(6) 그룹내에서의 공동작업	(목표 1) 고려의 요구가 영향을 받는 전체 그룹에 의해서 동의된다 (목표 2) 엔지니어링 그룹간의 협력이 영향을 받는 전체 그룹에 의해 동의된다 (목표 3) 엔지니어링 그룹이 그룹내에서의 분체를 인식하고, 추적, 해결한다
(7) 종결관리	(목표 1) 종결 검토 작업이 계획된다 (목표 2) 소프트웨어 작업에 의한 프로젝트에 있어서의 결정이 인식되고, 제어된다

부터 문서를 통하여 파악하는 것이 가능하다. 프로세스를 문서화하는 것으로 프로세스의 개선이 용이하게 처리될 수 있다. CMM 레벨 3의 목표는 <표 2>와 같다.

(4) 레벨 4 : Managed

각 프로세스에 대해서 매트릭스를 이용한 상세한 측정이 행해진다. 이것에 의해 외부로부터 조직의 현 상태를 정확하게 파악하는 것이 가능하고 품질의 향상에 커다란 도움이 된다. CMM 레벨 4의 목표는 <표 3>과 같다.

<표 3> CMM 레벨 4의 목표

CMM 레벨 4 : Managed에 있어서 Key Process Area의 목표
(1) 정적인 프로세스 관리 (목표 1) 정적인 프로세스 작업 관리가 계획된다. (목표 2) 소프트웨어 프로세스에 정의된 프로세스의 실행이 정적으로 제어된다. (목표 3) 조직의 표준 소프트웨어 프로세스 특성이 정적으로 인식된다
(2) 소프트웨어 품질관리 (목표 1) 프로젝트의 소프트웨어 품질 관리 작업이 계획된다 (목표 2) 소프트웨어 프로덕트 품질과 그 우선도에 관해서 측정 가능한 목표가 정의된다 (목표 3) 소프트웨어 프로덕트에 대한 품질목표의 달성에 있어서 실패의 진의 상황이 정량화하여 판리된다.

(5) 레벨 5 : Optimized

이 단계에까지 도달한 조직에서는 항상 프로세스의 개선이 행해지고, 조직은 최적의 프로세스에 접근할 수 있다. 생산성은 대단히 높고, 전적 정도도 정확하다. CMM 레벨 5의 목표는 <표 4>와 같다.

<표 4> CMM 레벨 5의 목표

CMM 레벨 5 : Optimized에 있어서 Key Process Area의 목표
(1) 결정 예방 (목표 1) 결정의 예방작업이 계획된다 (목표 2) 결정의 일반적 발생이 발견되고, 식별된다 (목표 3) 결정의 일반적 발생에 우선순위를 부여하고, 체계적으로 평가한다
(2) 기술변경 관리 (목표 1) 기술변경의 원인이 계획된다 (목표 2) 새로운 기술이 품질과 생산성면에서 평가된다 (목표 3) 적절한 새로운 기술이 조직에서 보통의 습관으로 변환된다
(6) 프로세스 변경관리 (목표 1) 연속적인 프로세스 개선이 계획된다 (목표 2) 조직의 소프트웨어 프로세스 개선 작업의 참가가 확장, 처리된다. (목표 3) 조직의 표준 소프트웨어 프로세스와 프로젝트에서 정의된 소프트웨어 프로세스가 항상 개선된다.

이와같이 조직이 성숙함에 따라 레벨을 단계적으로 끌어 올릴 수 있으며, 단계를 건너 뛰는 것은 불가능하다. 또한 새로운 조직이 높은 레벨에

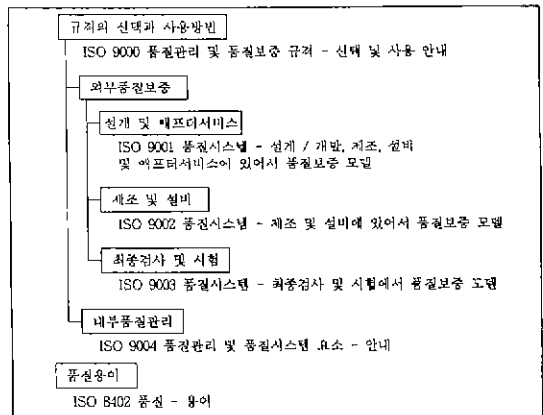
서부터 출발하는 것도 있을 수 없다. 레벨을 올리는 것으로 조직의 생산성이 향상되고, 전적 정도도 보다 정확하게 될 수 있다.

과거의 조사결과에 따르면 조직의 대부분은 레벨 1이고, 레벨 4 이상의 조직은 거의 없는 것으로 나타나고 있다(1991년까지의 조사에 의하면 미국의 조직의 81%가 레벨 1이고, 12%가 레벨 2, 7%가 레벨 3, 레벨 4이상은 0%였다). 그러나 특히 미국의 기업을 중심으로 조직의 개선 작업을 진행하고 있으며 장래에는 보다 높은 레벨의 결과가 나오리라고 예상된다.

3.2 ISO 9000-3의 개요

ISO 9000 시리즈는 모든 산업에 적용되는 품질 시스템의 국제 표준으로 일반적이며 세계적으로 보편 타당성이 있다고 여겨지던 품질 시스템과 관련된 내용들을 ISO가 체계화하여 도출해 낸 표준이다. 이 ISO 9000 시리즈의 배경으로는 품질의 중요성이 상업이나 산업분야에서 높이 인식됨에 따라 다양한 국내외적 표준이 각 분야의 요구를 충족시키기 위해 개발되어 왔다. 이처럼 점점 증가하는 품질 요구에 부응하며 국제간에 공통으로 사용될 수 있는 품질 시스템을 만들고자 하는 노력으로 ISO 9000 시리즈가 제정되었다.

ISO 9000 시리즈의 구성은 (그림 3)과 같다.



(그림 3) ISO 9000 시리즈의 구성

ISO 9000 시리즈 중 규정 범위가 가장 광범위한 ISO 9001의 본체 부분인 제 4장의 목차를 살펴보면 <표 5>와 같다.

<표 5> ISO 9001의 제 4장의 목차

4. 품질시스템 요구사항
4.1 경영의 책임
4.2 품질 시스템
4.3 계약 내용의 확인
4.4 설계 관리
4.5 문서 관리
4.6 구매
4.7 구입자에 의한 지급품
4.8 제품의 식별 및 추적성
4.9 공정 관리
4.10 검사 및 시험
4.11 검사, 계측 및 시험 장치
4.12 검사 및 시험의 상태
4.13 부적합품의 관리
4.14 시정 조치
4.15 취급, 보관, 포장 및 인도
4.16 품질기록
4.17 내부품질 감사
4.18 교육, 훈련
4.19 애프터 서비스
4.20 통계적 방법

ISO 9000-3은 소프트웨어 제품의 품질에 관한 국제규약이다. ISO 9000-3은 1987년에 완성한 ISO 9001을 기초로 1991년에 ISO(International Organization for Standardization : 국제표준화기구)에 의해 작성되었다. ISO 9000-3은 이 ISO 9001을 소프트웨어에 적용할 경우에 가이드라인의 성격을 가지고 있다.

ISO 9000-3은 소프트웨어를 개발하고 공급하며, 유지보수하는 조직이 ISO 9001을 쉽게 이용할 수 있도록 하기 위한 지침들을 제공한다. 이 지침은 양쪽간의 계약에서 소프트웨어 제품의 개발과 공급, 유지보수를 행하는 공급자 능력의 실 소프트웨어의 특수성 때문에 ISO 9001(품질체계-설계/개발, 생산, 설치 및 서비스에 있어서의 품질보증을 위한 모델을 제시하는 지침)에 소프트웨어를 적용하기 위한 추가 지침인 ISO 9000-3(소프트웨어 개발, 공급 및 유지보수에 ISO 9001적용)을 1991년에 제정하였다. ISO 9001과 ISO 9000-3과의 관계를 상호 비교해 보면 <표 6>과 같다.

<표 6> ISO 9000-3과 ISO 9001과의 상호 대조표

ISO 9000-3의 내용	ISO9000-3항번호	ISO 9001항 번호
경영자의 책임	4.1	4.1
품질 시스템	4.2	4.2
내부 품질 시스템 감사	4.3	4.17
시정 조치	4.4	4.14
계약 검토	5.2	4.3
구매자의 요구사항 명세서	5.3	4.3, 4.4
개발 계획	5.4	4.4
품질 계획	5.5	4.2, 4.4
설계 및 구현	5.6	4.4, 4.9, 4.13
시험 및 확인	5.7	4.4, 4.10, 4.11, 4.13
인수	5.8	4.10, 4.15
복제, 인도 및 설치	5.9	4.10, 4.13, 4.15
유지 보수	5.10	4.13, 4.19
구성 관리	6.1	4.4, 4.5, 4.8, 4.12, 4.13
문서 관리	6.2	4.5
품질 기록	6.3	4.16
측정	6.4	4.20
규칙, 실행 및 관례	6.5	4.9, 4.11
도구 및 기술	6.6	4.9, 4.11
구매	6.7	4.6
다른 제품의 포함	6.8	4.7
교육/훈련	6.9	4.18

증을 요구하는 경우에 제공되며, 개발부터 유지보수에 이르는 생명주기 전단계에서 부적합성을 사전에 방지하는 것을 목적으로 한다.

ISO 9000-3은 0~6장으로 구성되어 있는데 이 중 0~3장은 보통 국제규격의 구성과 마찬가지로 서문, 적용범위, 인용규격, 용어의 정의 등이 기술되어 있고, 4~6장은 소프트웨어에 대한 규격의 본체에 대한 내용을 담고 있다. ISO 9000-3은 <표 6>에서 보는 바와 같이 ISO 9001의 4장에 대응한다. 또한 ISO 9000-3은 '프레임 워크', '생명주기 활동', '지원 활동'으로 구성되어 있으며 그 세부 사항을 살펴보면 <표 7>과 같다.

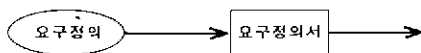
CMM이 주로 미국에서 사용되고 있는 것에 비해 ISO 9000-3은 EC 통합의 움직임에 호응하여 작성되었다고 하는 경위로 특히 유럽에 널리 사용되고 있다.

3.3 CMM과 ISO 9000-3의 프로세스 기술

상기 프로세스 기술 방법에 기초하여 CMM과 ISO 9000-3의 양자에 대해 프로세스를 기술하였

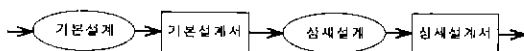
〈표 7〉 ISO 9000-3의 4.6장의 내용

4	품질 시스템 - 프레임워크
4.1	경영자의 책임
4.2	품질 시스템
4.3	내부품질 시스템의 감사
4.4	시정 조치
5	품질 시스템 - 생명주기 활동
5.1	일반
5.2	계약 검토
5.3	구매자의 요구 사항
5.4	개발 계획
5.5	품질 계획
5.6	설계 및 구현
5.7	시험 및 확인
5.8	인수
5.9	복제, 인도 및 설치
5.10	유지보수
6	품질 시스템 - 지원 활동
6.1	구성 관리
6.2	문서 관리
6.3	품질 기록
6.4	측정
6.5	규칙 및 실행
6.5	도구 및 기법
6.7	구매
6.8	다른 제품이 포함
6.9	교육/훈련



(단계 1 : 요구정의)

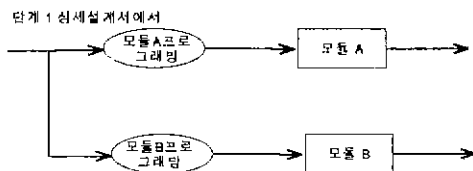
(그림 4) CMM 레벨 1에 의한 프로젝트 클래스에서의 프로세스



단계 1 요구정의서에서

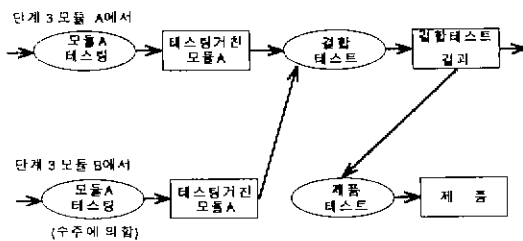
(단계 2 : 설계)

(그림 5) CMM 레벨 1에 의한 프로젝트 클래스에서의 프로세스



(단계 3 : 프로그래밍)

(그림 6) CMM 레벨 1에 의한 프로젝트 클래스에서의 프로세스



단계 3 모듈 B에서
(우주에 의함)

(단계 4 : 테스트)

(그림 7) CMM 레벨 1에 의한 프로젝트 클래스에서의 프로세스

다. 우선, 소프트웨어 개발의 간단한 프로세스 기술예를 작성하고, 이것을 CMM 레벨 1의 프로세스 기술이라 한다. 기술한 프로세스 기술에 대하여 CMM의 각 레벨이 요구하고 있는 관리작업을 (그림 4, 5, 6, 7)과 같이 차례로 부가한 예를 보이고 있으며 하위레벨의 진행 프로세스의 상세함은 생략하기로 한다.

마찬가지로 ISO 9000-3에 대해서도 CMM 레벨 1의 간단한 프로세스 기술예에 ISO 9000-3에서 요구되는 여러가지 관리작업에서 (그림 8, 9)와 같이 조직 클래스와 단위작업 클래스를 대응시키는 예를 작성하였다.

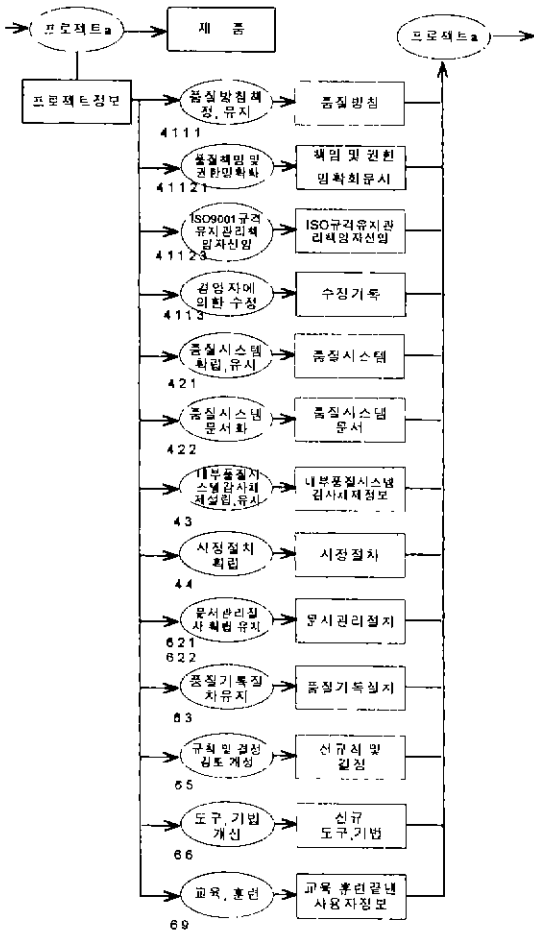
이 때 CMM의 프로세스 기술에 있어서 각 관리작업은 각각 CMM의 각 목표에 대응하는 것처럼 또 ISO 9000-3의 프로세스 기술에 대해서는 각 관리작업은 각각 ISO 9000-3의 각장에 대응하는 것으로 각각 기술하였다.

그러나 아직 예외적인 기술방식도 존재한다.

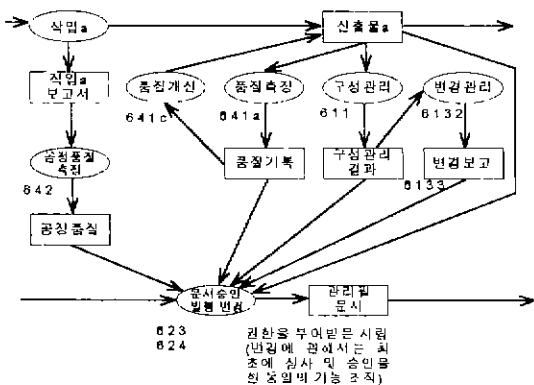
(1) 예외 1: 복수의 목표나 장이 1개의 작업에 대응하는 경우.

여기에서는 프로세스 기술상 복수의 목표를 1개로 정하는 쪽이 보다 이해하기 쉽다고 생각하는 경우나 CMM의 레벨이 오르는 단계에서 동일한 명칭의 작업이 질적으로 변화하는 경우 원래의 기술에 거의 같은 내용의 기술이 복수개 있는 경우 등이 있다.

(2) 예외 2: 목표 또는 장이 산출물에 대응하



(그림 8) 조직 클래스



(그림 9) 단위작업 클래스

는 경우.

이것은 목표나 장의 기술이 작업보다 오히려 산출물에 대한 요구가 있는 경우이다.

(3) 예외 3:1개의 목표 또는 장이 복수의 작업에 대응하는 경우.

여기에서는 본래의 기술이 복수의 요소를 포함하고 있는 경우나 본래의 기술의 추상도가 높고, 기술상 복수의 작업으로 할 필요가 있는 경우 등이 있다.

(4) 예외 4: 해당하는 목표 또는 장의 프로세스 기술이 없는 경우.

여기에서는 본래의 기술이 애초 프로세스 기술에 친숙해지지 않는 기술인 경우가 있다.

4. 결 언

최근 소프트웨어 품질관리 사고방식의 인식의 전환에 힘입어 소프트웨어 프로젝트 관리에서의 소프트웨어 프로세스에 대한 연구가 중점적으로 진행되고 있다. 특히, 소프트웨어 프로세스 기술 중 개발 프로세스에 비해 관리프로세스에 대한 연구가 많이 진전되지 못하고 있다. 그러나 실질적으로는 소프트웨어 품질보증 시스템으로서 관리 프로세스의 사고가 넓게 이용되고 있으므로 본 고에서는 개발 프로세스와 관리 프로세스를 분리하지 않고 동일 레벨로 취급하여 작업을 진행하는 방안을 모색하였다.

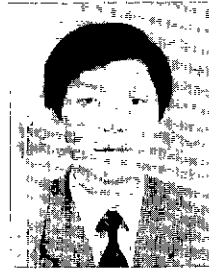
따라서 본 고에서는 소프트웨어 개발에서 관리 프로세스를 기술하기 위해 소프트웨어 개발관리 모델과 기술 방식을 살펴보았으며, 현재 프로세스 기술 방식 중 미국에서 널리 이용되고 있는 품질보증 시스템 CMM(Capability Maturity Model)과 유럽에서 널리 이용되고 있는 ISO 9000-3에 적용해 본 결과 양자의 품질보증 시스템이 상호보완하여 고품질 소프트웨어 개발이 가능하다는 것을 알 수 있었다.

향후 연구과제로는 앞서 기술한 품질보증 시스템의 프로세스 기술을 이용한 CMM과 ISO 9000-

3의 상호 비교하여 좀 더 상세한 프로세스 기술 방식이 필요하다고 생각되며, 실제의 프로젝트를 기초로 한 프로세스의 기술이나 CMM과 ISO 9000-3을 기초로 한 새로운 품질보증 시스템의 제안 등이 이루어져야 할 것으로 본다.

참 고 문 헌

1. ISO 9000-3 Quality management and quality assurance standards Part 3: Guidelines for the application of ISO 9001 to the development, supply maintenance of software, 1991.
2. ISO/IEC 9126, Information technology Software product evaluation Quality characteristics and guidelines for their use.
3. 양해술, 소프트웨어 인증제도 도입을 위한 제언, 정보산업, 한국정보산업연합회, 1992.
4. 정호원, 양해술, "ISO 9000 시리즈와 소프트웨어 품질시스템", 하이테크정보, 1993.
5. 양해술, 윤창섭, 임춘봉, 정호원, "소프트웨어 인증제도 연구", 연구보고서, 한국전산원, 1992.
6. 양해술, 정호원, "소프트웨어 공학과 품질관리 방법론의 동향", 정보처리, 한국정보처리 응용학회, 1994.3.
7. 水野幸男: ソフトウェアの總合的品質管理, 日科技連出版, 1990.
8. 飯塚悦功: ソフトウェアの品質保證, 日本規格協會, 1992.
9. 飯塚悦功: ソフトウェアの品質保證-ISO/DIS 9000-3 對譯と解説, 日本規格協會, 1990.
10. 吉澤 正, 東基衛, 片山禎昭: ソフトウェアの品質管理と生産技術, 日本規格協會, 1988.



양 해 술

1975년 홍익대학교 공과대학 전기공학과 졸업(학사)
 1978년 성균관대학교 정보처리학과 정보처리 전공(석사)
 1991년 日本 오사카대학 기초공학부 정보공학과 소프트웨어공학 전공(공학박사)
 1975년~79년 육군중앙경리단 전자계산실 근무

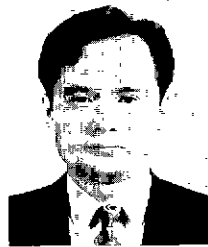
1984년~92년 성균관대학교 경영대학원 강사
 1986년~87년 日本 오사카대학 객원연구원
 1994년~현재 한국산업표준원 이사
 1994년~현재 한국정보과학회 학회지 편집부위원장
 1994년~현재 한국정보처리응용학회 논문편집위원장
 1980년~현재 강원대학교 전자계산학과 교수
 관심분야: 소프트웨어 공학(특히, S/W 품질보증과 평가, SA/SD, OOA/OOD/OOP, CASE), 소프트웨어 프로젝트관리.



이 용 근

1988년 강원대학교 자연과학대학 전자계산학과 졸업(이학사)
 1994년 강원대학교 대학원 전자계산학과 소프트웨어공학 전공(이학석사)
 1989년~1992년 강원대학교 전자계산학과 조교

1994년 현재 한림전문대학 전산정보처리과 강사
 관심분야: 소프트웨어공학(특히, 소프트웨어 품질보증과 품질평가, 객체지향 프로그래밍, 객체지향 분석과 설계)



허 태 경

1983년 경북대학교 전자공학과 졸업(학사)
 1985년 경북대학교 대학원 전자공학과 졸업(공학석사)
 1985년 경북대학교 공과대학 조교

1986년~1994년 한국통신 품질보증단 S/W품질연구부장
 1994년~현재 한국통신 품질보증단 개발품질부장
 관심분야: 소프트웨어공학(특히, 소프트웨어 품질보증과 품질평가의 체계, 인증, 표준화)