

특집기사

소프트웨어 공학과 품질관리 방법론의 동향

양 해 술[†] 정 호 원^{††}

❖ 목 차 ❖

- | | |
|--------------------|-------------------------------|
| 1. 서론 | 4. ISO9 9000-3의 소프트웨어 품질관리 체계 |
| 2. 소프트웨어 공학의 동향 | 5. 소프트웨어 품질특성 |
| 3. 소프트웨어 품질보증 연구체계 | 6. 결론 및 향후 연구과제 |

1. 서론

종래에 고품질의 소프트웨어라는 것은 신뢰성이 높은 소프트웨어라고 생각해 왔다. 이로 인해 프로그램의 테스트시 가능한 많은 오류를 발견하고, 제거함으로써 고품질의 소프트웨어를 얻을 수 있다고 생각하였다. 그러나 소프트웨어의 품질이 좋다고 나쁘다고 하는 것은 그 본질 자체가 명확하게 정의되어 있지 않으면 객관적으로 평가할 수 없다. 품질관리 용어에서 품질은 “제품 또는 서비스가 사용목적에 만족되고 있는지를 결정하기 위한 평가의 대상이 되는 고유의 성질 또는 성능의 전체”로 정의하고 있다.

소프트웨어 품질관리 방법론중 품질보증은 생산물이 설정된 명세와 일치하는가를 확인하는데 필요한 모든 계획된 체계적인 활동을 말한다. 이와 같은 소프트웨어 품질보증은 개발과 수정동안 사용된 절차, 도구 및 기법들이 원하는 수준만큼 생산물의 품질을 보장하는데 적합함을 증명하는 것이다. 그러나 문제는 각각의 사람들이 품질을 보는 견해가 다른데 있다. 즉, 프로젝트 관리자에게 있어서 품질

은 표준에 따라 제한된 비용과 기간내에 기능적 요구사항을 구현하는 것을 의미할 것이며, 사용자에게 있어서 품질이란 것은 사용하기 용이하고 빠른 응답시간을 갖는 것이 될 것이다. 또한, 프로그래머에 있어서 품질은 수행을 올바르게 하고 프로그램 표준에 맞아야 할 것이다. 그리고 시스템 유지보수에 있어서의 품질은 명확한 문서화와 이해가 용이한 코드를 의미할 것이다.

주어진 시스템의 품질에 대한 명백한 서술없이 개발자가 어떤 품질을 정착해야 할 지 알지 못할 것이고 품질보증자 또한 어떤 품질을 주안점으로 시스템을 평가할지 알 수 없으며 프로그램 관리자 또한 수행될 시스템의 품질의 중요성이 어떠한 지를 결정할 수가 없게 된다.

따라서 본 연구에서는 2장에서 소프트웨어 공학의 연구 동향을 살펴보고, 3장부터는 최근 국제적으로 많은 연구가 활발히 진행되고 있으나, 아직 국내에서 문제 인식이 미흡한 소프트웨어 품질관리 방법론에 대하여 논의하고자 한다. 이를 위해 소프트웨어 품질에 대한 국제적 연구활동과 연구내용을 토대로 소프트웨어 품질관련 연구체계, 품질관리 개념, 품질특성, 품질평가 및 인증방법에 대해 고찰해 본다.

[†] 종신회원, 강원대학교 전신과 교수

^{††} 정 회 원 : 한국전산원 책임연구원

2. 소프트웨어 공학의 동향

소프트웨어 공학은 신뢰성있는 소프트웨어를 경제적으로 개발하기 위해 공학적인 원리나 방법을 창출하고 사용하는 기술 활동의 총칭이라고 볼 수 있으나, 가시성과 물질적인 특성을 바탕으로 한 다른 분야의 공학적 방법과는 근본적으로 차이가 있기 때문에 기존의 공학적인 방법을 이용한 소프트웨어 개발은 매우 어렵다.

현재의 소프트웨어 공학의 주요 연구 동향을 살펴보면 정의, 개발, 유지보수, 사업관리, 품질보증, 개발 환경 등으로 분류할 수 있다. 정의, 개발 및 유지보수는 생명주기의 단계들에 대한 연구이며, 사업관리와 개발 환경은 제단계의 관리와 지원에 관한 사항들을 고려한 연구분야이다. 그리고 개발 및 유지보수는 소프트웨어 개발 단계의 전단계를 지원하는 것을 의미한다. 소프트웨어 공학에서는 소프트웨어의 처리 단계 뿐만 아니라 제단계의 관리의 물론 품질보증도 매우 중요한 연구분야이며, 또한 이들을 효율적으로 지원할 수 있는 개발 환경도 매우 중요한 의미를 갖는다.

이러한 소프트웨어 공학은 단독으로 존재하는 경우와 컴퓨터 시스템 공학의 한 부분으로 할당되는 경우도 있다. 단독으로 존재하는 경우에 소프트웨어 공학의 외부 환경은 주위 환경이 되며 소프트웨어를 시스템과 동일하게 취급한다. 컴퓨터 시스템 공학의 한 부분이 되는 경우에 소프트웨어 공학의 외부 환경은 컴퓨터 시스템이므로 소프트웨어 공학의 내용이 반영되어야 한다.

2.1 생명주기

생명주기는 소프트웨어 개발의 시작부터 종료까지 준수해야 할 단계와 그에 따른 작업의 형태 변화를 말한다. 방법론에 따라 여러가지 형태가 있으나 정의, 개발 및 유지보수의 범위 내에서 주요 작업과 그에 따른 생산물로 표현된다. 정의 단계에서는 생성할 소프트웨어 생산물이 무엇을 할 것인지에 중점을 두고, 개발 단계에서는 생산물을 어떻게 구현할 것인지에 중점을 두며, 유지보수 단계에서

는 그 생산물의 유지보수에 중점을 둔다.

2.1.1 정의

정의는 작업의 계획과 요구사항으로 구성되며, 작업계획은 작업의 범위, 개발 과정, 조직, 비용, 자원, 시간 계획, 기법, 도구 등 사업에 관련되는 모든 것을 포함한다. 그리고, 요구사항은 사용자가 요구하는 모든 지식을 개발자와 함께 정의하는 것이다.

2.1.2 개발

개발은 소프트웨어 생명주기에서 가장 중요시되는 것으로 자원이 가장 많이 투입되는 부분이다. 요구사항이 정의되면 개발이 시작되는데 요구사항에서 정의된 내용을 설계하고 구현하는 것이 개발이다. 개발에는 개략설계, 상세설계, 코딩 및 시험을 포함한다. 개략설계는 소프트웨어의 구조를 만드는 단계인데 소프트웨어 생산물을 구성하는 중요한 부분들과 그들 간의 관계를 정의한다. 상세설계는 개략설계와 관련하여 구현 단위의 구체적인 처리 절차를 정의하는 것이다. 그리고, 코딩은 채택된 프로그래밍 언어를 이용하여 처리 절차에 상응하는 프로그램을 작성하는 것이며, 시험은 단위 시험, 통합 시험 및 시스템 시험을 준비하고 수행하는 것이다.

2.1.3 유지보수

유지보수는 개발단계에서 생성된 소프트웨어 생산물이 사용자에게 인도된 다음에 발생하는 소프트웨어 생산물에 대한 수정과 보안을 의미한다. 실질적으로 소프트웨어 생명주기에서 가장 많은 비용과 노력이 소요되는 부분이지만, 현실적으로 문제점을 해결하는데 소요되는 비용과 노력은 많이 투입되지 못하고 있다. 최근 연구에서 유지보수의 성패는 유지보수 단계의 활동에 있는 것이 아니라, 최초의 시스템 정의부터 개발완료되는 단계까지 유지보수에 대한 고려가 전제되어야 한다고 보고되고 있다.

2.2 사업관리

사업관리의 목표는 전체 생명주기를 통해 소프트

웨어의 가시성을 향상시켜 효율적인 소프트웨어 관리를 가능하게 하는데 있다. 사업관리는 소프트웨어 개발에 대한 사업 계획과 사업에서 채택한 방법론을 바탕으로 사업수행의 원칙에 따라 사업을 관리하고 평가하며 통제하는 모든 활동을 말한다.

2.3 품질보증

품질 보증은 생산물의 품질이 일정한 기준에 부합되는지를 확인하기 위해 수행되는 활동의 총칭을 말한다. 이러한 품질보증은 유지보수의 비용을 줄이고, 고품질의 소프트웨어를 개발하기 위해서 생명주기 전단계에서 개발되는 소프트웨어의 품질을 각 단계에서 평가함으로써 소프트웨어의 생산성을 높일 수 있을 뿐만 아니라 신뢰성 향상, 그리고 유지보수성에 대한 효율성을 최대한 보장하기 위한 활동을 의미한다.

2.4 개발환경

개발환경은 생명주기 각 단계를 자동화하기 위한 통합된 개발환경의 도구를 의미한다. 그러나 도구들은 일반적으로 개별적으로 사용되는 경우가 대부분이기 때문에 각각의 도구들을 통합하여 생명주기 모든 활동을 지원할 수 있는 통합된 개발환경을 제공해야 한다. 좋은 개발 환경은 간단하고 일관성이 있으며 사용자에게 친근한 인터페이스를 제공함으로써 사용자의 개발환경을 용이하도록 제공하는 것이다.

3. 소프트웨어 품질보증 연구체계

소프트웨어 공학의 진정한 개념을 실현하기 위해 중점적으로 연구해야 할 분야 중의 하나는 소프트웨어의 품질에 대한 것이다. 이는 공장에서 생산되는 제품에 대한 품질과 동일한 개념을 갖는다. 즉, 공장에서 하나의 제품이 정해진(규격화 된) 공정에 따라 필요한 부품들이 조립되어 생산된 제품의 품질을 시험(Test)한 후 구매자에게 전달되고, 또한 구매자가 품질에 대해 서비스를 받는 것과 마찬가지로, 소프트웨어도 개발 단계를 거쳐 개발완료된 완성품에 대해 품질을 시험해야 하고 사용자에

게 인수된 후에도 품질에 대해 서비스를 제공해야 한다.

기존의 많은 소프트웨어 방법론들은 개발측면 즉, 요구되어진 기능을 어떻게 개발할 것인가의 관점이었고, 사용자(구매자, 개발 위탁자)들도 원하는 기능을 제공하느냐의 관점에서만 소프트웨어를 평가하였다. 이와 같은 방법론들은 개발과정과 각 과정에서 필요한 지원도구, 지침, 방법, 절차의 적용에만 치중하였고, 소프트웨어 품질에 대한 시험평가 부분은 별도로 처리하여 제공되지 않았다.

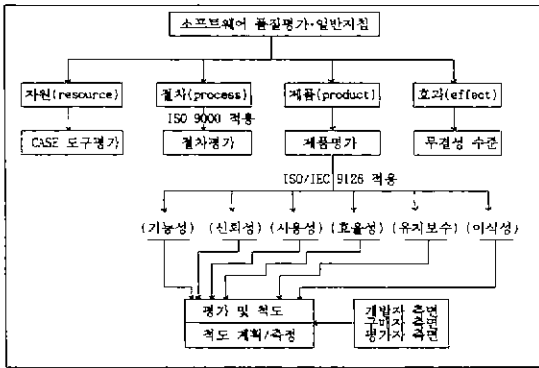
그러나, 1980년대 후반부터 개발자 뿐만 아니라 사용자측을 고려한 소프트웨어 품질에 관한 많은 연구가 수행되어져 오고 있다. 특히 일반 품질관리 및 보증체계-품질개념, 품질시스템, 품질주기 등에 관한 국제표준(ISO 9000)의 제정에 편승하여 소프트웨어 분야에서의 품질관리 및 보증을 위한 국제표준(ISO 9000-3)의 지침이 제정되었다. 이에 관련한 구체적인 연구는 ISO, IEC, IEEE 등의 국제 표준화 기구에서 활발히 수행되고 있다.

품질관련 용어 개념정립 및 정의...	→ ISO 8402
표준선택 및 사용을 위한 지침	→ ISO 9000
○ 품질체계-설계/개발, 생산, 설치, 서비스에서 품질보증 모델	→ ISO 9001
○ 품질체계-생산 및 설치에서 품질보증 모델	→ ISO 9002
○ 품질체계-최종검사 및 시험에서 품질보증	→ ISO 9003
○ 품질관리 및 품질체계요소-지침	→ ISO 9004

(그림 1) 일반 품질관리 및 보증에 관한 연구

소프트웨어 개발/공급/유지보수에 ISO 9001을 적용하기 위한 지침	→ ISO 9000-3
--	--------------

(그림 2) 소프트웨어 품질관리 및 보증에 관한 연구



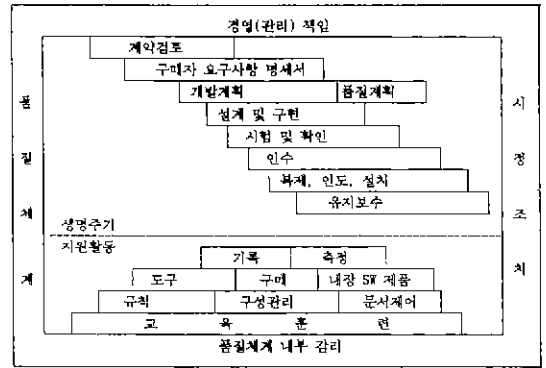
(그림 3) 소프트웨어 품질평가에 관한 연구체계

소프트웨어 품질에 관한 이러한 연구들은 크게 두 분야로 구분된다. 첫째 분야에는 ISO 9000과 ISO 9000-3과 같이 일반적인 품질 및 품질관리에 대한 기본 개념, 절차 등에 관한 것과 이를 소프트웨어 품질관리에 적용시키려는 연구로 품질을 효과적으로 관리하는데 필요한 전반적인 활동지침을 제공하려는 연구이다. 둘째 분야는 소프트웨어 품질관리를 위한 전반적인 지침을 실제로 어떻게 적용할 것인가에 관한 연구로서 이 분야에선 지금까지 소프트웨어 제품에 대한 평가와 보증 방법에 대하여 중점적으로 연구하고 있다. 소프트웨어 품질에 관한 연구체계와 분야를 정리하면 (그림 1), (그림 2), (그림 3)과 같이 구성할 수 있다.

4. ISO 9000-3의 소프트웨어 품질 관리 체계

소프트웨어에 대한 일반적 품질관리 및 보증체계를 수립하는 방법중 하나로 소프트웨어 품질관리 및 보증에 관한 지침을 제공하는 것이다. 본 연구에서는 1991년에 ISO 9001(품질체계-설계/개발, 생산, 설치 및 서비스에 있어서 품질보증을 위한 모델을 제시하는 지침)을 소프트웨어에 적용하기 위해 표준으로 제정된 ISO 9000-3에 대하여 간략히 살펴보도록 한다.

ISO 9000-3은 소프트웨어를 개발하고 공급하며, 유지보수하는 조직이 ISO 9001의 적용이 용이하도록 지침을 제공하는 것이다. 이것은 소프트웨어 생명주기 전단계에서 품질을 보증하기 위한 일



(그림 4) ISO 9000-3 구조

련의 활동들을 수행하도록 지침을 제공함으로써 생산된 소프트웨어가 공급자의 능력을 실증하고, 구매자의 요구를 만족시킬 수 있도록 유도하기 위한 것이다. ISO 9000-3은 크게 품질체계, 생명주기, 지원활동으로 구성된다. ISO 9000-3의 각각의 구성요소들에 대한 구조는 (그림 4)와 같다.

4.1 품질체계

품질체계의 틀은 경영(관리)책임, 품질체계, 내부 품질체계의 감사, 시정조치로 구성된다. 경영책임은 품질관리 및 보증을 위해 공급자가 갖추어야 할 품질방침(정책), 조직, 검토활동과 구매자 입장에서 수행해야 할 업무 및 권한, 공급자와 구매자가 수행할 합동 검토에 대해 지침을 기술한다.

품질체계에서는 품질체계에 대한 개념, 품질관리 및 보증 활동시 필요한 모든 요소에 대한 문서화, 공급자가 수행해야 할 품질관리 및 보증업무 수행 계획서의 필요성에 대해서도 지침을 기술하고 있다. 또한, 내부 품질체계 감사에서는 품질활동이 계획된 합의안에 따르는지를 검증하고, 품질체계의 효과를 점검하기 위해 계획되고 문서화된 총체적 내부 품질감사의 수행을 요구하고 있다. 그리고, 시정조치에서는 공급자가 시정에 관련된 다음과 같은 활동에 대하여 절차를 수립하고 문서화하여 유지할 것을 요구한다.

- 부적합한 제품의 원인조사 및 재발 방지를 위해 필요한 시정조치

- 부적합한 제품의 잠재원인을 탐지하여 제거하기 위한 모든 절차, 작업, 양해사항, 품질기록, 서비스 보고서 및 고객불만을 분석
- 당면한 위험에 상응하는 수준으로 문제를 처리하기 위한 예방조치
- 시정조치를 취하고 그 조치가 효과적임을 보장하기 위한 제어
- 시정조치 결과를 나타내는 절차의 변경을 구현하고 기록

4.2 품질체계(생명주기 활동)

소프트웨어 개발 프로젝트는 생명주기 모델에 따라 조직된다. 그리고, 품질과 관련된 활동들은 이용되는 생명주기 모델의 특성에 관련해 계획되고 구현된다. ISO 9001 및 ISO 9000-3은 계약 상황에 중점을 두고 있으며, ISO 9000-3에서 정의하고 있는 소프트웨어 생명주기 활동과 내용들을 살펴보면 다음과 같다.

4.2.1 계약 검토

공급자는 계약검토 및 계약의 조정을 위한 절차를 수립하고 유지해야 하며, 다음과 같은 사항들을 보장하여야 한다.

- 가) 계약과 요구사항의 범위가 정의되고 문서화되었는지 확인
- 나) 우발사건이나 위험 가능성을 식별
- 다) 특허 정보가 보호되었는지 확인
- 라) 입찰시의 요구사항과 차이가 있는 요구사항에 대해 확인
- 마) 공급자는 계약 요구사항을 만족시킬 능력이 있는지 확인
- 바) 용어의 통일에 대한 확인
- 사) 구매자가 계약상의 의무사항을 만족시킬 능력을 갖고 있는지를 확인

그리고 품질에 관련된 계약상에 명시되어야 할 요소로는 다음과 같은 것들이 있다.

- 가) 인수기준
- 나) 개발중 구매자 요구사항 변경에 대한 처리

- 다) 인수후에 탐지된 문제에 대한 처리
- 라) 구매자의 활동중 요구사항 명세서, 설치 및 인수에서 구매자의 역할
- 마) 구매자가 제공하는 설비, 도구, 소프트웨어 항목
- 바) 사용되는 표준과 절차
- 사) 복제 요구사항

4.2.2 구매자 요구사항 명세서

정확한 소프트웨어 개발을 위해서 공급자는 완전하고 명확한 기능 요구사항을 제공해야 한다. 이러한 요구사항들은 구매자의 요구를 만족시키기 위해 성능, 안전성, 신뢰성 등 필요한 모든 요소를 포함하며 향후 인수시에 이용된다. 그리고, 공급자는 개발에 착수하기 전에 구매자의 승인을 얻어야 하며, 개발 문서화의 일부로 문서 제어와 구성관리를 받는다. 또한, 소프트웨어 제품 및 다른 소프트웨어나 하드웨어 제품간의 인터페이스도 명세서에 완전하게 기술한다.

4.2.3 개발계획

가. 개발 계획서

개발 계획서에는 착수하기 이전에 검토후 승인을 받아야 하며 다음과 같은 사항들을 처리한다.

- 프로젝트 정의 및 목적
- 팀구조, 책임, 외주 계약자가 사용하는 자원 등을 포함한 프로젝트 자원과 조직
- 개발단계
- 수행할 작업과 각 작업에서 요구되는 자원, 시간, 작업들간의 상호관계
- 관련 계획서의 식별 및 구분: 품질계획서, 구성관리 계획서, 통합계획서, 시험 계획서

① 단계

개발 계획서에서는 구매자 요구사항 명세서를 소프트웨어 제품으로 변화시키기 위한 절차나 방법론을 정의한다. 정의되어야 할 주요 요소들은 다음과 같다.

- 개발단계

- 각 단계에서의 요구 입출력
- 각 단계에서의 검증절차

② 프로젝트 관리

개발 계획서에는 프로젝트를 어떻게 관리할 것인가를 정의하는 것으로 다음과 같은 사항들을 기술한다.

- 개발, 구현 및 관련된 산출물 인도 일정
- 진도 제어
- 조직의 책임, 자원, 작업 배정
- 다른 조직간의 인터페이스

③ 개발방법 및 도구

개발 계획서에서는 모든 활동이 정확하게 수행됨을 보장하기 위한 방법으로 다음과 같은 사항들이 포함된다.

- 개발에 관한 규칙, 실행, 관례
- 개발도구 및 기술
- 구성관리

가. 진도제어

주요 자원들에 대한 문제해결 및 개발계획의 효과적인 수행을 보장하기 위한 진척 정도를 제어하고 문서화한다.

나. 개발단계 입력

각 개발단계에서 요구되는 입력들을 정의하고 문서화한다. 그리고 각 요구입력에 대해 검증할 수 있도록 정의한다.

다. 개발단계 출력

각 개발단계에서 요구되는 출력을 정의하고 문서화한다. 이때 각 개발단계의 출력은 검증되어야 한다.

라. 각 단계의 검증

공급자는 각 단계의 모든 개발 출력을 검증하기 위한 계획을 수립한다. 이 검증은 다음과 같은 측정수단에 의해 해당 입력 요구사항을 만족하는지를 입증한다.

- 개발단계의 적절한 시점에서의 개발검토
- 시험 및 실증 실시

4.2.4 품질계획

품질 계획서는 구현과 관련된 모든 조직들이 공식적으로 검토되어 합의된 것으로 다음과 같은 사항들이 정의된다.

- 품질목적
- 각 개발단계의 입출력 기준
- 수행되는 시험/검증/확인 활동 유형
- 검토 및 시험에 대한 책임 소재
- 구성관리 및 변경제어에 대한 책임소재
- 시정조치 및 결점제어에 대한 책임소재

4.2.5 설계 및 구현

설계 및 구현활동은 구매자의 요구 명세서를 소프트웨어 제품으로 변환시키는 활동이다.

가. 설계

여기서는 설계시 고려사항들을 식별하고 설계방법론을 채택한다. 설계는 시험, 유지보수 및 사용을 용이하게 할 수 있도록 해야 한다.

나. 구현

프로그래밍 규칙, 프로그래밍 언어, 명명 규칙, 코딩, 주석삽입 규칙 등과 같은 규칙들이 구성되며, 공급자는 구매자 요구사항을 만족하기 위해 적절한 구현 방법 및 도구를 이용하여 개발한다.

다. 검토

공급자는 요구사항과 위의 방법들이 올바르게 수행되었다는 것을 보장하기 위해 생성된 결과물을 검토한다.

4.2.6 시험 및 확인

가. 시험계획

공급자는 시험전에 시험 계획서, 시험 명세서 및 시험절차를 수립하고 검토하며, 검토시 다음과 같은 사항들을 고려한다.

- 소프트웨어 항목, 통합, 시스템 시험, 인수시험을 위한 계획
- 시험항목, 시험 데이터 시험시 기대값
- 수행될 시험형태(기능 시험, 경계 시험, 성능 시험, 사용성 시험)

- 시험환경, 도구, 시험 소프트웨어
- 시험완료 판단 기준
- 사용자 문서
- 필수 요원과 요원 훈련 요구사항

나. 시험

시험시 다음 사항을 주의한다.

시험결과는 관련 명세서의 정의대로 기록한다.

- 발생된 문제에 대해 파생되는 영향까지 분석하여 정의한다.
- 수정할 경우, 그 영향부분을 식별하여 재시험한다.
- 시험이 충분한지, 적절인지 평가한다.
- 하드웨어와 소프트웨어 구성을 문서화한다.

다. 확인

공급자는 제품을 제공하기 이전에 가능한한 계약에서 정의한 환경을 제공하는지를 확인한다.

라. 현장시험

현장시험시 다음과 같은 사항을 주의한다.

- 현장환경에서 시험할 때의 특이사항
- 시험평가할 공급자, 구매자의 책임사항
- 시험후 사용환경으로의 복원

4.2.7 인수

공급자가 시험 확인된 제품을 인도할 준비가 되었을때, 구매자는 사전에 합의한 기준 및 계약에 따라 제품의 인수여부를 판단한다. 인수과정에서 발견되는 문제의 취급 처리방법은 구매자와 공급자가 합의하여 문서화한다.

인수하기 전에 공급자는 설치일정, 평가절차, 소프트웨어 및 하드웨어 환경, 인수기준 등 구매자가 식별하고자 하는 사항들을 지원한다.

4.2.8 복제, 인도 및 설치

복제는 인도전에 행하는 활동으로서 복제 제공시 다음 사항들을 고려한다.

- 인도되는 각 소프트웨어 항목의 복제 수량
- 각 소프트웨어 항목별 매체 유형
- 매뉴얼과 사용자 지침서와 같은 필요한 문서화 명기

- 마스터 및 백업복사 보관
- 복사 공급자의 의무 공급기간

4.2.9 유지보수

설치후 구매자가 소프트웨어 제품의 유지보수를 요구할 때 이를 계약에 명시한다. 공급자는 규정된 유지보수 활동을 위해 필요한 절차를 수립하고 유지한다.

소프트웨어 제품에 대한 유지보수 활동은 문제해결, 인터페이스 수정, 기능확장 또는 성능개선 유형으로 구분된다. 또한, 유지보수 대상항목 및 기간은 계약서에 규정하며, 대상 항목은 프로그램, 데이터 명세서, 구매자/사용자를 위한 문서, 공급자가 사용하는 문서 등이 있다.

모든 유지보수 활동은 공급자와 구매자가 사전에 합의한 유지보수 계획서에 따라 수행되고 관리되며, 유지보수 계획서에는 유지보수 범위, 제품의 초기상태, 지원조직, 유지보수 활동, 유지보수 기록 및 보고 등에 대해 기술한다.

4.3 품질체계 (지원 활동)

ISO 9000-3에서 지원활동은 생명주기 단계중 어떤 단계에도 해당되지 않는다. 품질체계를 지원하는 활동으로는 구성관리, 문서제어, 품질기록, 측정, 규칙/실행/관례, 도구 및 기술, 구매, 내장 소프트웨어 제품, 교육훈련 등이 있다.

4.3.1 구성관리

각 소프트웨어 항목의 버전을 식별하고 제어하며 추적하는 기법을 제공한다. 공급자는 구성관리 계획서를 개발하고 구현한다. 계획서에는 구성관리 조직, 조직의 책임, 구성관리 활동, 구성관리 도구 및 방법론, 구성제어 항목이 입력되는 단계 등을 기술한다. 구성관리 활동은 구성요소 식별 및 추적, 변경제어, 구성상태 보고로 구분된다.

4.3.2 문서 제어

공급자는 관련된 모든 문서를 제어하기 위한 절차를 수립하고 유지한다. 이를 위해서 문서 제어

절차에 따라야 하는 문서들을 결정하고 제어절차를 승인한다. 문서 제어절차에 해당하는 문서로는 다음과 같은 것이 있다.

- 소프트웨어 생명주기에 적용되는 품질체계를 기술하는 절차 문서
- 공급자의 모든 활동 및 구매자의 상호작용을 기술하는 문서
- 특정 소프트웨어 제품을 기술하는 제품 문서

4.3.3 품질 기록

공급자는 품질 기록(수집), 색인, 파일링, 보관, 유지, 폐기에 대한 절차를 수립하고 유지한다. 이 품질기록은 요구품질의 달성 및 품질체계의 효율성을 실증할 수 있도록 유지되며, 품질 기록은 적절한 환경에 검색이 용이하도록 유지되며 보존 기간 동안 보관된다.

4.3.4 측정

측정 제품측정과 공정측정으로 구분된다. 제품측정에는 고객의 관점에서 제품결함을 나타내는 최소한의 척도가 있어야 하며, 측정 결과들을 비교할 수 있는 해당 척도도 식별되어 있어야 한다. 공정 측정에서는 개발공정과 인도과정에서의 품질에 대한 정량적인 측정을 한다. 여기에서는 척도들을 일정으로 보아 개발공정이 얼마나 잘 수행되었는가 그리고 개발공정이 내재될 결함을 얼마나 감소시키는가를 반영한다. 척도는 가능하면 공정에 적합하고, 인도되는 소프트웨어 품질에 직접적인 영향을 줄 수 있는 척도를 선정해야 한다.

4.3.5 규칙, 실행 및 관례

ISO 9000-3에서 규정된 품질체계의 효율화를 위해 공급자는 규칙과 관례를 제공해야 한다. 또한, 공급자는 이러한 규칙과 관례를 검토해야 하며 필요시 이를 개정한다.

4.3.6 도구 및 기술

ISO 9000-3에서 규정된 품질체계의 효율화를 위해 공급자는 소프트웨어 도구, 설비 및 기술을

제공하여야 하며, 이것은 제품개발, 관리면에서도 효과적이다.

4.3.7 구매

공급자는 구매된 제품, 서비스가 규정된 요구사항에 적합함을 보장하여야 한다. 여기서 구매제품이란 요구되는 최종 제품에 포함하고자 하는 소프트웨어나 하드웨어, 제품 개발에서의 지원용 도구일 수도 있다.

4.3.8 내장 소프트웨어 제품 포함

공급자는 구매자나 제 3자가 제공하는 소프트웨어 제품을 포함시키거나 사용할 것을 요청받을 수 있다. 따라서, 공급자는 이러한 제품의 확인, 저장, 보호, 유지보수를 위한 절차를 수립하고 유지해야 하며, 인도되는 제품의 유지보수 합의시 소프트웨어 제품의 지원에 대하여 고려해야 한다.

4.3.9 교육훈련

공급자는 훈련 소요를 파악하여 절차를 유지해야 하며, 품질에 영향을 미치는 활동을 수행하는 모든 요원들에게 훈련을 실시한다. 그리고, 특정 업무를 수행하는 요원들은 교육훈련 경험에 따라 자격을 부여한다. 훈련의 주제는 소프트웨어 제품의 개발, 관리에서 사용되는 도구, 기술, 방법론, 컴퓨터 자원을 고려해 결정한다.

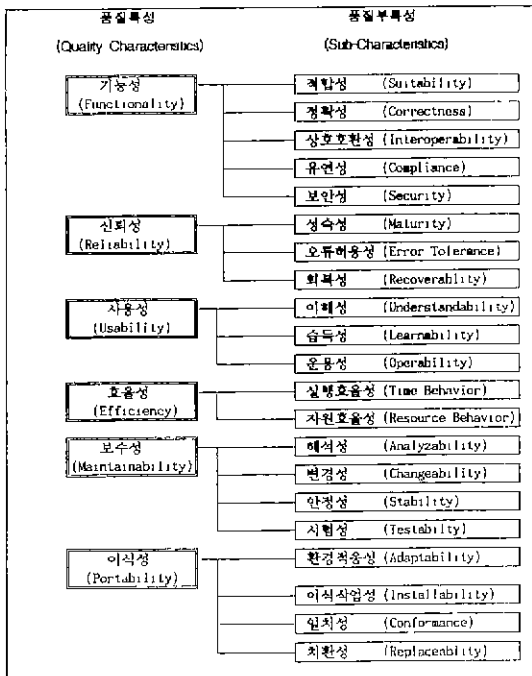
지금까지 소프트웨어 개발에 관한 국제표준 품질관리 및 보증체계라 할 수 있는 ISO 9000-3에 대해 살펴보았다. 이 지침은 원칙론적인 차원에서 기술된 상위수준의 규정이기 때문에 실제로 소프트웨어 개발업체가 공급자 입장에서 이 지침에 따라 소프트웨어 제품을 개발하고, 시험/확인/검증(VVT)하여 구매자의 요구사항을 충족시킬 수 있는 제품을 개발하는 데에는 많은 어려움이 따른다. 이것은 ISO 9000-3 지침이 각 단계에 대한 구체적인 수행절차와 방법까지는 기술하지 않고 다만, 특정 단계에서 어떤 활동이 필요하고 그 활동을 할 때는 어떤 점을 유의하는 것이 좋다는 정도만을 정의하고 있기 때문이다.

따라서, 궁극적으로 품질을 보장할 수 있는 소프트웨어 개발을 위해서는 본 지침을 기본틀로 한 좀 더 구체적인 지침 또는 방법이 필요하다. 현재, 이에 관련한 연구가 많이 수행되고 있으며 특히, 핵심적이고 연구의 가치가 높은 소프트웨어 품질 시험, 평가 분야에 대한 연구가 활발히 진행되고 있다.

5. 소프트웨어 품질특성

4절에서 살펴보았듯이 소프트웨어 품질관리 및 보증은 개발과정과 소프트웨어 제품을 대상으로 하고 있다. 따라서, 소프트웨어를 객관적이고도 정량적으로 특성화하고 평가하기 위한 기술은 매우 중요하기 때문에 소프트웨어 품질평가를 위한 기본적인 틀(framework)이 제공될 필요가 있다.

1991년에 제정된 ISO/IEC 9126은 소프트웨어 품질평가를 위한 기본틀로서 국제적으로 인정받고 있는 유용한 지침이다. 이 ISO/IEC 9126은 소프트웨어 제품 평가를 위한 품질특성을 식별, 정의한 것으로 간략히 살펴보면 (그림 5)와 같다.



(그림 5) 품질특성과 품질부특성과의 관계

5.1 ISO/IEC 9126 범위

이 지침은 소프트웨어 제품의 품질을 사용자, 구매자 관점에서 품질특성을 정의하였다. 여기서 정의한 품질특성은 모든 소프트웨어 제품에 적용될 수 있으며, 이들 품질특성들은 소프트웨어 품질을 좀 더 세분화하고 기술하는 기본틀을 제공하며, 소프트웨어 품질평가를 위한 품질특성으로 이용될 수 있다.

5.2 소프트웨어 품질특성과 품질부특성

5.2.1 기능성 (Functionability)

기능 집합과 사양화된 내용을 실현하는 특성의 집합으로써 명시적 또는 암시적 요구를 만족하는 속성 집합을 의미한다.

- 가) 적합성: 사양화된 TASK에 대한 일련된 기능 존재나 적절한 기능을 지닌 소프트웨어 속성
- 나) 정확성: 올바른 규격, 일치된 결과, 그리고 정확한 효과를 나타내는 소프트웨어 속성
- 다) 상호호환성: 사양화된 소프트웨어가 다른 시스템과 상호 운용 능력을 지닌 소프트웨어 속성
- 라) 유연성: 법률이나 동일한 형태의 규격, 규격, 협정, 규격에 대한 유연성을 지닌 소프트웨어 속성
- 마) 보안성: 우발적이거나 고장에 의한 것이 아닌 부당한 액세스로부터 프로그램이나 데이터를 보호하는 소프트웨어 속성이 아닌 부당한 액세스를 방지하는 속성

5.2.2 신뢰성 (Reliability)

명시된 조건하에서 명시된 기간, 소프트웨어의 실행 레벨을 유지하기 위한 능력을 만족하는 속성의 집합을 의미한다.

- 가) 성숙성: 소프트웨어에 존재하는 장애에 의한 고장을 대처하는 기능의 소프트웨어 속성
- 나) 오류허용성: 소프트웨어 장애 또는 사양화된 인터페이스를 유지하지 않는 경우에 사양화된 실행 수준으로 회복하기 위한 능력과 시

간을 가지는 소프트웨어 속성

- 다) 회복성: 고장시에 소프트웨어의 실행 수준을 재확인하여 직접 영향을 받는 데이터를 회복하기 위한 능력과 시간을 가지는 소프트웨어 속성

5.2.3 사용성 (Usability)

명시적 또는 암시적으로 사용자가 이용하기 위해 필요한 노력으로 각각의 사용 결과에 의한 평가를 나타내는 속성의 집합을 의미한다.

- 가) 이해성: 소프트웨어의 논리적인 개념을 사용자의 이해 정도를 나타내는 소프트웨어 속성
- 나) 습득성: 소프트웨어의 운용관리, 입출력 방법 등을 습득하기 위한 사용자의 노력을 나타내는 소프트웨어 속성
- 다) 운용성: 소프트웨어의 운용과 운용관리를 하기 위한 사용자 노력 정도를 나타내는 소프트웨어 속성

5.2.4 효율성 (Efficiency)

명시적인 조건하에서 소프트웨어의 실행 레벨과 사용되는 자원 양자간의 관계를 나타내는 소프트웨어 속성의 집합을 의미한다.

- 가) 실행 효율성: 소프트웨어의 기능을 수행할 때 응답시간과 처리시간을 나타내는 속성
- 나) 자원 효율성: 소프트웨어의 기능을 실행할 때 사용되는 자원의 양과 사용 시간을 나타내는 속성

5.2.5 보수성 (Maintainability)

사양화된 개정을 처리하기 위해 필요로 하는 노력을 나타내는 속성의 집합을 의미한다.

- 가) 해석성: 고장의 원인 분석이나 결함분석 및 개정 가능한 부분의 정도와 사용 시간을 나타내는 소프트웨어 속성
- 나) 변경성: 개정을 위한 장애 제거나 환경 변경에 필요한 노력을 나타내는 소프트웨어 속성
- 다) 안전성: 개정에 의해 예기되는 효과의 포함성을 나타내는 속성

- 라) 시험성: 소프트웨어의 타당성 검증에 필요한 노력을 나타내는 속성

5.2.6 이식성 (Portability)

임의의 환경에서 다른 환경으로 소프트웨어를 이식하기 위한 속성의 집합을 의미한다.

- 가) 환경 적응성: 이식목적에 적합한 작업이나 수단 이외의 것은 이용하지 않고, 사양화된 다른 환경으로 소프트웨어를 이식할 수 있는 소프트웨어 속성
- 나) 이식 작업성: 소프트웨어를 사양화한 환경에 도입하기 위해 필요한 노력을 나타내는 속성
- 다) 일치성: 규격이나 규정에 소프트웨어를 적합시킬수 있는 속성
- 라) 치환성: 사양화된 다른 소프트웨어와 치환되어 그 환경에서 사용하기 위한 기회나 노력을 나타내는 속성

5.3 품질특성들의 활용

위에서 살펴본 소프트웨어 품질특성들은 다음과 같은 여러 상황에서 다양한 목적으로 활용될 수 있다.

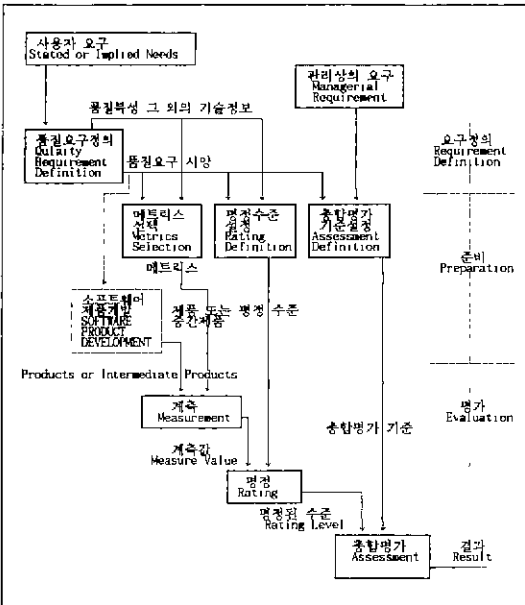
- 소프트웨어 품질의 요구사항 정의
- 소프트웨어의 제품 평가(측정, 등급판정, 심사)
- 소프트웨어 제품의 품질 요구사항 정의
- 소프트웨어 명세서가 소프트웨어 개발동안 품질 요구사항을 만족하는지 파악
- 구현된 소프트웨어의 기능과 속성 기술
- 인도(delivery) 전에 개발된 소프트웨어 평가
- 인사(acceptance) 전에 소프트웨어 평가

이 지침에서 제시하고 있는 품질특성을 위와 같은 상황에서 적용하기 위해서는 해당 조직 혹은 소프트웨어의 특성에 적합한 평가절차와 척도(metrics)를 설정하여야 한다.

5.4 품질평가 절차

평가절차에 대해 ISO/IEC 9126에서는 (그림

6)과 같이 제시하고 있으며, 각 단계에 대하여 살펴보면 다음과 같다.



(그림 6) ISO/IEC 9126에서의 소프트웨어 평가 절차 모델

5.4.1 요구 단계

가) 품질요구 정의(Quality Requirement Definition)

소프트웨어를 개발하는 경우와 동일하게 품질요구 정의 단계에서도 첫단계로 요구자의 요구 분석하고 정의해야 한다. 이 단계에서는 요구자의 요구를 만족하기 위해서 소프트웨어에 요구되는 조건을 분석하고 사양화하는 과정이다.

품질평가에서는 일반적으로 기능요구가 먼저 정의되고, 소프트웨어 품질요구 정의는 기능요구에 기초하여 품질요구를 품질특성에 적용시킨다. 일반적으로 소프트웨어는 신뢰성, 사용성, 성능 등 여러 특성에 의해서 품질이 평가되기 때문에, 어떠한 품질특성이 얼마나 중요한지는 대상으로 하는 소프트웨어에 따라서 다르다.

따라서, 품질요구 정의 단계에서는 사용자의 요구에 적합한 소프트웨어를 개발하기 위해 단순히 기능요구뿐만 아니라, 어떤 형태의 품질특성이 얼

마나 중요한가를 명확히 정의함으로써 효율적으로 준비단계를 처리할 수 있도록 사용자의 요구를 명확히 정의한다.

5.4.2 준비단계

가) 메트릭스 선택(Metrics Selection)

소프트웨어 품질을 계측하는데 이용되는 메트릭스는 계측 대상의 소프트웨어와 공정에 따라서 다르다. 보편적으로 메트릭스의 성능, 신뢰성 계측을 위한 메트릭스 등 국한된 메트릭스를 사용하고 있으나, 보편적인 메트릭스뿐만 아니라 특수한 메트릭스를 개발함으로써 특수한 목적을 지니는 소프트웨어도 평가할 수 있도록 메트릭스를 개발하면 대상 목적에 따라서 메트릭스를 선택적으로 사용하는 것이 가능하다고 볼 수 있다.

나) 평정 수준 설정(Rating Level Definition)

메트릭스를 이용하여 계측한 결과는 단지 수치를 나타내는 것이므로 그것이 적절한 값인지 또는 부적절한 값인지를 알 수 없다. 응답시간이 2초라고 해도 어떤 종류의 시스템에는 충분한 값인 반면에 다른 종류의 시스템에서는 충분하지 못한 값이라고 생각할 수도 있다. 따라서 대상으로 하는 시스템을 특정한 응답시간이 몇 초 동안이 가장 적절한지를 우수/적당/부적당 등과 같이 몇개의 단계로 분류하여야 하는데, 이와 같은 것을 소프트웨어 품질 평정 수준(평정 레벨)이라 한다. 평정수준을 5로 설정한 예를 살펴보면 (그림 7)과 같다.

다) 총합평가 기준 설정(Assessment Criteria Definition)

각각의 품질특성, 품질부특성마다 평정수준을 결정하고, 이 결과를 기초로 하여 전체적으로 소프트

평정 수준이 5인 경우

95	매우 우수 Excellent
90	우수 Good
85	적당 Fair
80	부적당 Poor
!	매우 부적당 Very Poor

(그림 7) 평정수준 설정 예

웨어의 품질이 어느 레벨에 해당되는지를 결정할 수 있도록 종합평가 기준을 설정하는 것이다. 종합평가 기준은 소프트웨어 품질이 우수/적당/부적당한지를 평가하기 위해 기준을 설정하는 단계이다. 여기서 우수/적당은 소프트웨어를 신뢰할 수 있지만, 부적당한 경우에는 평가 대상 소프트웨어에 대하여 신뢰할 수 없기 때문에 각각의 기준에 대하여 정확하게 설정하여야 한다.

3) 평가 단계

가) 품질 계측(Quality Measurement)

평가 대상 소프트웨어의 사양서와 같은 중간 생산물과 최종 생산물들을 실제로 계측하여 척도로 나타내는 단계를 품질계측 단계라 한다. 즉, 소프트웨어의 품질특성이 요구되는 품질을 갖추고 있는지 없는지를 평가하기 위해요소 데이터(메트릭스) 값을 실제로 계측함으로써 소프트웨어 품질특성을 정량적으로 평가하는 단계이다.

나) 평정(Rating)

계측값을 평정 수준과 비교하여 소프트웨어의 품질특성이 어느 수준에 속하는지를 결정하는 단계이다. 소프트웨어 품질특성의 계측값이 정량적인 동시에 객관적으로 처리되고 평정 수준이 표준화되어 있으면 평정은 매우 정확하게 처리할 수 있다. 그러나, 계측값이 주관적인 입장을 완전히 배제할 수 없으므로 평정 가이드 라인을 제공할 필요가 있다.

다) 종합평가(Assessment)

평가 대상이 되는 소프트웨어 제품의 품질을 종합적으로 평가하기 위해 계측된 모든 품질특성의 결과값과 가중치를 고려한 계산식에 의해서 산출된 결과값을 종합평가 기준에 적용하여 판정하는 단계를 종합평가 단계라 한다.

6. 결론 및 향후 연구과제

본 연구에서는 향후 중요한 과제로 등장할 것으로 예측되는 소프트웨어 품질관리 방법론에 대해서 소프트웨어 전 개발과정에서의 품질관리 및 보증, 제품의 품질평가 방법으로 구분하여 살펴보았다. 이를 위해 ISO/IEC 등에서 제정된 국제표준과 수

행된 연구결과 등을 참조하였다.

지금까지 소프트웨어 품질관리 및 보증에 대해 살펴보았지만 눈에 보이지 않는 대상을 평가하는 것은 어렵다는 것을 알 수 있었다. 오히려 우수한 제품은 수요가 많게 되므로 사용자에게 의해 자연히 품질의 우수성이 평가되기 때문에 별도의 노력 없이도 품질을 평가할 수 있다고 생각할 수도 있다. 그러나, 소프트웨어 품질관리 및 보증 활동은 여러 상황에서 필요하다. 예를 들면, 동일한 기능을 갖는 여러 소프트웨어 제품의 평가시, 소프트웨어를 외주용역으로 개발한 후 인도되는 소프트웨어에 대하여 평가하고자 할 때, 자체 개발시 개발단계의 완료시 확인하고자 할 때 등이다.

본 연구를 통해 알 수 있듯이 소프트웨어 품질관리 및 보증분야는 ISO 9001, ISO 9000-3, ISO/IEC 9126에서 연구가 진행되면서 점차 그 틀이 잡혀가고 있기 때문에 이상에서 제기되는 문제점들은 점차 해소되리라 예상된다. 그리고, 이러한 연구결과는 소프트웨어 품질 시험 및 인증제도 등과 같은 정책적 제도로 발전되어야 한다.

이미 오래전부터 국외에서는 소프트웨어 품질관리를 위해 인증제도를 도입하기 위해 많은 노력을 기울이고 있다. 향후 국내에서도 이러한 변화에 적응하여 유용한 효과를 얻기 위해 소프트웨어 품질관리 및 보증에 관한 국제적 연구조류에 편승하여 많은 연구와 참여가 기대된다.

참 고 문 헌

1. IEEE Standard for Software Verification and Validation Plans, 1986.
2. ISO 8402 Quality-Vocabulary, 1986.
3. ISO 9000, Quality management and quality assurance standards-guideline for selection and use, 1987.
4. ISO 9001, Quality system-Model for quality assurance in design/development for selection, installation and servicing, 1987.
5. ISO 9002, Quality system-Model for quality assurance in production and installation,

- 1987.
6. ISO 9003, Quality system-Model for quality assurance in production and installation, 1987.
 7. ISO 9004, Quality system-Model for quality assurance in final inspection and tes, 1987.
 8. 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.
 9. ISO/IEC 9126,Information technology Software product evaluation Quality characteristics and gudelines for their use.
 10. ISO/DIS 9126, Software Product Evaluation, 1991.
 11. ISO/IEC 9126 Information technology software product evaluation quality characteristics and gudeliners for their use, 1991.
 12. ISO/IEC Guide 2, General terms and their definitions concerning standardization and related activites, 1986.
 13. SCOPE : A Method for Softwar Assessment and Certification, Feb. 1992.
 14. ISO/DIS 9000-2 : 1992, Quality management and quality assurance standardieadaction-Part 2:Generic guidelines for the application of ISO 9001, ISO 9002 and ISO 9003
 15. 정호원, ISO 9000 시리즈의 국내 적용과제, 정보산업, 한국정보산업연합회, 1982.
 16. 양해술, 임춘봉, 정호원, "소프트웨어의 품질 보증과 평가방법", 제2회 전산망 기술 및 표준화 심포지움, 1992.

17. 정기원, "소프트웨어 공학 표준화 동향", 전산망 기술 및 표준화 심포지움, 1991.
18. 정호원, 양해술, "ISO 9000 시리즈와 소프트웨어 품질시스템", 하이테크정보, 1993.
19. 양해술, 윤창섭, 임춘봉 "소프트웨어 인증제도 연구", 연구보고서, 한국전산원, 1992.
20. 양해술, "소프트웨어 품질보증과 평가 자동화 도구의 개발", 장기 기초 연구보고서, 한국통신, 1993.



양 해 술

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

1984년~94년 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), 전문가 시스템.



정 호 원

1979년 고려대학교 산업공학과 졸업(학사)
 1981년 한국과학기술원 산업공학과(공학석사)
 1990년 University of Arizona 경영정보시스템(박사)
 1990년~91년 DACOM(주), Clemson University 에서 근무

1991년~현재 한국전산원 책임연구원
 관심분야: 소프트웨어 공학, 시뮬레이션, 알고리즘 설계, 컴퓨터 네트워크