정보시스템 감리가 소프트웨어 품질에 미치는 영향

김 용 경* 김 필 중**

A Study on the Information Systems Audit to the Aspects of Effectiveness on the Software Quality

Yong Kyong Kim* Pil Joong Kim**

Abstract

The National Computerization Agency has been leading the Information Systems(IS) audit since 1987, as development of IS projects in governmental and public sectors have become daily issues. While many considerations on audit guidelines and techniques for quality management in IS have been given to respond to the fast increasing demand of audit process since mid 1990's, the study on the fact that how audit activities directly affect the quality of software process has been put aside.

In this paper, we went through 74 audit results, which were reported by H auditing organization, and performed on 59 IS projects carried on 1999 through 2001.

As a result of study, we found that more than one fourth of errors of Coding and Test phases are caused in the earlier phases, which naturally leads to the conclusions that auditing on earlier phases can save a lot of stitches in the later phases. Comparing two types of IS projects - the one that went through only one audit process, and the other that went through two audit processes-, we found that projects with two audits carry less errors on Planning, Analysis, Design phases as a whole, which revealed the facts that IS audit actually gives positive effects on the quality of software.

* 건양대학교 경영정보학과 교수
** 건양대학교 정보통신공학부 교수

1. 서론

1.1 연구 의의 및 목적

정보화의 급진전과 함께 빈번하게 발생하고 있는 각종 정보 관련 범죄와 정보화의 역기능은, 정보화에 대한 막대한 투자 필요성 못지않게 정보시스템의 부실 방지와 품질 확보에 대한 필요성을 점차 높여가고 있다. 이와 같은 이유로 정보시스템 감리는 정보시스템의 개발과 운영시 독립적인 제3자의 평가활동을 통해 부실 방지와 품질 향상에 기여할 수 있기 때문에 그 중요성과 필요성은 점차 증대되고 있다.

정보화 사업이 효율적으로 개발되고 관리되기 위해서 발주자측과 개발자측에서는 개발되고 있는 정보시스템이 본래의 사업 목적에 맞게 추진되고 있는지를 점검하고 확인할 필요가 있으며, 사용자의 요구사항이 적절하게 반영되고 그리고 구현되고 있는지를 점검하는 활동이 필요하다. 좀더 자세히 말하면 사용자 요구사항이 품질속성인 소프트웨어 프로세스 즉, 분석 및 설계에 적절히 반영되고 구현을 통해서 품질 목표가 달성되었는지 확인이 필요하다는 것이다.

크로스비는 품질을 "요구사항에 대한 적합성"으로 정의하고 품질의 목표는 '무결함(zero defect)'으로 하며, 수정보다는 예방위주의 시스템으로 접근해야 함을 주장하였다[Crosby 1980]. 기본적으로 품질관리의 원칙은 발생 가능한 결함을 예방함으로써 정의된 요구사항에 대하여 무결성을 보장할 수 있다는 것이다.

소프트웨어를 개발하여 공급하는 계약을 체결할 때 고객이 품질시스템을 계약사항으로 명시하는 경우에 공급자는 그 요구사항을 만족시켜 주어야 한다. 고객은 외부고객과 내부고객으로 나눌 수 있는데, 외부고객은 계약의 당사자처럼 전형적인 외부조직을 의미하며, 내부고객은 동일한 기업이나 조직내의 타 부서나 하부 부서가 될 수 있다.

대부분의 조직에서 정보시스템 개발에는 시스템 품질의 문제, 오랜 개발기간, 사용자 불만족 그리고 높은 개발비용 등과 같은 문제들이 나타난다. 그리고 이와 같은 문제들은 서로 복합적으로 작용하여 새로운 시스템의 요구를 강요하게 된다[Cusumano 1991].

정보시스템 개발에 대한 품질관리 활동은 소프트웨어 수명주기에서의 개발공정 및 생산물이 설정된 명세와 일치하는데 필요한 모든 계획된 체계적인 활동을 의미한다. 또한 정보시스템의 품질보증은 소프트웨어 제품을 개발하거나 공정을 수행하는 직접적인 책임이 있는 사람에게 조직적인 자유 및 권한의 부여를 필요로 한다.

미국 소프트웨어 개발회사의 경우 결함율이 15%인 제품이 소비자에게 그대로 제공되는가 하면, 전체 개발 프로젝트의 25%가 실패, 이미 제작된 소프트웨어에 대한 재작업을 위해 전체 작업시간과 비용의 30%-44%를 투자, 프로젝트 일정의 50%만이 계획된 시간에 맞추는 등, 개발 프로세스 상에 많은 문제를 가지고 있다. 또한 소프트웨어 개발 생산성 향상이 대부분의 소프트웨어 개발 조직의 주요 과제였음에도 불고하고, 컴퓨터 하드웨어의 성능은 3년마다 거의 2배로 향상되는 반면 소프트웨어의 생산성은 매년 4% 정도의 향상에 그치고 있는 실정이다[Curtis 1995].

선진국에 비해 소프트웨어 시장이 영세한 한국의 경우에는 개발 프로젝트의 수행능력이나 문화가 상대적으로 성숙되어 있지 않으므로 소프트웨어 품질관리에 대한 인식은 아직도 미흡한 상태라고 할 수 있다. 그러나 다행히 우리나라는 1987년부터 공공부문

의 정보시스템 개발 사업에 한하여 부분적으로나마 한국전산원과 그 위탁업체들을 통해 감리활동이 진행되어 오고 있다. 1990년대 중반 이후 급격히 증가된 감리 수요와 함께 감리지침의 개정 등 감리기술과 발전방안에 대하여는 많은 연구가 진행되어 왔으나, 감리가 소프트웨어 품질에 직접적으로 어떤 영향을 미치고 있는가에 대하여는 아직도 본격적인 연구가 되고있지 않은 것이 사실이다.

따라서 본 연구에서는 먼저 국내에서 수행된 공공정보시스템 개발 사업을 대상으로 실시된 감리보고서를 입수하여 면밀히 검토한 뒤, 국내의 공공정보시스템 소프트웨어는 개발과정에서 수명주기 단계별로 어떤 문제점들을 지적받고 있는가를 세부적으로 조사하였다. 이렇게 조사된 지적사항(문제점)들은 바로 우리나라 공공소프트웨어의 개발행태를 나타내게 되고, 그것을 통해 정보시스템 감리 활동이 결국 소프트웨어 품질에 어떤 영향을 미치고 있는가를 밝혀보는데 본 연구의 의의와 목적이 있다.

1.2 연구 범위 및 방법

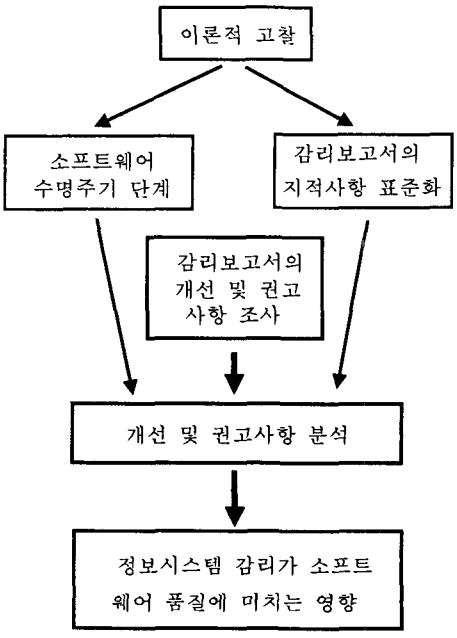
국내에서는 1987년부터 한국전산원의 주관하에 공공정보시스템에 한한 감리를 실시해 오고 있으며 1997년에 74건, 1998년에 127건, 1999년에 214건, 2000년에 226건, 2001년에 250여건의 감리가 실시되었다.

본 연구에서는 한국전산원의 위탁을 받아 H감리원이 1999년부터 2001년까지 3년동안에 실시한 정보시스템 감리 중 59개 프로젝트에 대한 74건의 감리보고서를 선정하여 자료를 조사하였다. 연구 목적상 조사 대상이 된 감리보고서는 주로 소프트웨어 개발 사업에 대한 것이며, 가능한 자료의 일관성을 위해 사업규모가 특별히 크거나 작지 않은 중간규모(2억 이상 30억 미만이 대부분임) 사업을 대상으로 하였다. 따라서, 사업별 감리 횟수도 대부분 1회(최종감리)로 끝났거나 아니면 1, 2차(중간감리, 최종감리)로 나누어 실시된 감리만을 대상으로 조사하였다.

전체 감리실적 74건 중 44건은 최종감리만 실시한 것이고, 30건은 중간감리와 최종감리를 각각 15회씩 실시한 것이다.

연구에 사용된 자료는 사업별로 감리를 실시한 후에 작성된 '감리보고서'이며, 보고서 내의 '개선 및 권고사항'에 명시된 개발단계별 세부지적사항을 조사하여 통계처리하는 방법을 사용했다.

감리보고서의 세부지적사항들은 감리인마다 그 표현 방법이 다르고 내용이 지나치게 세부적이거나 추상적이기도 하다. 따라서, 표준화된 일정한 틀(항목)에 맞추어 다시 정리하고 지적된 횟수를 집계하기 위해 '지적사항 조사표'를 작성하였다.



(그림 1-1) 연구의 틀

'지적사항 조사표'에 옮겨진 세부지적사항들은 감리형태별 또는 사업규모별 등으로 집계하고 분석하여, 정보시스템감리가 소프트웨어 품질에 미치는 영향을 확인하였다.

연구의 틀은 (그림 1-1)과 같다.

2. 소프트웨어 프로세스와 품질

2.1 소프트웨어 프로세스

소프트웨어 프로세스란 고품질의 소프트웨어를 구축하는 데 요구되는 태스크(task)에 대한 프레임워크(frame work)로 정의된다. 따라서 소프트웨어 프로세스는 소프트웨어를 공학화 할 때 채택하는 접근법으로 설명된다[Pressman 1997].

기업의 실제 문제를 해결하기 위해 소프트웨어 개발팀은 소프트웨어 프로세스, 방법 그리고 도구에 일반적인 개발과정을 포함하는 개발 전략을 통합시켜야 한다. 이러한 전략을 소프트웨어 프로세스 모형(software process model) 또는 소프트웨어 공학 패러다임(software engineering paradigm)이라고 부른다.

소프트웨어 프로세스 모형은 시대적 흐름과 정보기술의 발전 그리고 대상 업무의 규모와 성격 등에 따라 매우 다양하게 제시되어 왔다. 대표적인 모형으로는 ① 고전적생명주기모형(classic life cycle model) 또는 폭포수모형(waterfall model)이라고 부르는 선형순차모형(linear sequential model), ② 프로토타이핑모형(prototyping model), ③ 신속응용개발모형(rapid application development model), ④ 점증적 모형(incremental model). ⑤ 나선형모형(spiral model) ⑥ 4세대기술모형(fourth generation techniques model) 등 다양하다.

이들 여러 가지 모형 중 선형순차모형은 가장 오래동안 적용되어 온 모형임은 물론 대형 프로젝트의 개발에 많이 적용되어 온 모형이다.

앞서 제시된 모형들은 각각의 특성과 장점 그리고 적절하게 적용되는 응용시스템이 있기 마련이지만, 소프트웨어 프로세스 모형은 응용시스템의 영역, 프로젝트의 크기, 또는 복잡도와 관계없이 일반적으로 3단계로 구분된다.

첫째는 정의단계(definition phase)로 무엇(what)에 초점을 맞춘다. 이 단계에서는 소프트웨어의 핵심 요구사항이 정의된다. 즉, 개발하고자 하는 소프트웨어의 기능적 범위는 어떻게 되며, 처리되는 정보는 무엇이며, 요구되는 성능은 무엇이고, 설계시 기술적 제한점은 무엇이며, 성공적인 시스템으로 평가하기 위해서는 어떤 검증기준이 요구되는지 등을 정의하게 된다. 따라서 이 단계에서 수행하게 되는 주요 업무는 계획(planning)과 대상업무의 분석(analysis)이라고 할 수 있다.

둘째는 개발단계(development phase)로 방법(how)에 초점을 맞춘다. 이 단계에서 개발자는 데이터는 어떻게 구조화하는지, 요구된 기능은 소프트웨어로 어떻게 구현되는지, 인터페이스는 어떻게 하는지, 세부 처리절차는 어떻게 구현하는지, 테스트는 어떤 방법으로 어느 수준까지 하는지 등을 정의하게 된다. 개발 단계에서 수행되는 업무는 아주 다양하지만 크게 소프트웨어의 설계(design), 코딩(coding), 시험(test) 3단계로 나누어진다.

셋째는 유지보수단계(maintenance phase)로 변경(change)에 초점이 맞추어진다. 개발된 소프트웨어는 운영단계에 들어가며,

환경의 변화에 따라 새로운 적용, 발생된 문제의 수정, 그리고 기능 및 성능의 향상 등이 주 대상업무가 된다.

2.2 소프트웨어 품질

소프트웨어 품질은 크게 소프트웨어 개발 및 관리 프로세스의 품질과 소프트웨어 제품의 품질로 나눌 수 있다. 소프트웨어 개발 및 관리 프로세스의 품질을 측정하는 기준은 여러 가지가 사용되고 있지만 그중 국제표준 ISO 9000 시리즈가 가장 체계적으로 잘 발전하고 있으며, 소프트웨어 제품의 품질을 평가하는 국제 표준으로는 ISO 9126이 있다.

1986년 IBM의 Humphrey는 소프트웨어 성숙도 프레임웍을 Carnegie Mellon 대학의 소프트웨어공학연구소(SEI : Software Engineering Institute)로 가져와, 조직이 소프트웨어 프로세스 성숙의 어느 단계에 도달했는지를 제시해 주는 성숙도 모델(CMM : Capability Maturity Model for software)을 개발하였다[Humphrey 1989]. CMM은 소프트웨어 조직에게 소프트웨어를 개발하고 유지보수하기 위하여 그들의 프로세스를 어떻게 관리해야 하는가와 소프트웨어 개발측면과 관리측면의 문화를 어떻게 개선해 나가야 하는가를 제시한다.

소프트웨어 프로세스의 개선은 혁신적인 방법 보다는 지속적인 노력을 통하여 얻을 수 있다. CMM은 지속적인 프로세스 개선을 위한 기초로 다음과 같이 다섯 단계의 프로세스 성숙도 수준을 제시하고 있다.

가) 초기(initial)단계 : 소프트웨어 프로세스는 주먹구구식이고 매우 혼란스러운 상태다. 일부 프로세스가 정의되어 있을 수 있으나 그것의 성공여부는 특정 개인의 노력에 의존한다.

나) 반복(repeatable)단계 : 비용, 일정, 기능을 추적하기 위해 기본적인 프로젝트 관리가 이루어지고 있는 단계다. 프로젝트에 대한 정형화된 교육은 없고 새로운 프로젝트를 시작할 때는 이전에 성공한 프로젝트에서 사례를 찾아서 활용한다.

다) 정의(defined)단계 : 관리와 공학활동에 필요한 소프트웨어 프로세스는 조직 중심으로 문서화하고, 표준화하고, 통합한다. 모든 프로젝트는 소프트웨어를 개발하고 유지보수하기 위해 조직에서 공식적으로 사용하는 표준 소프트웨어 프로세스로 정립되어 있다.

라) 관리(managed)단계 : 소프트웨어 프로세스와 제품 품질의 세부적인 측정값을 수집한다. 소프트웨어 프로세스와 제품은 세부적인 측정을 사용해서 양(계수)적으로 이해되고 관리된다.

마) 최적화(optimizing)단계 : 지속적인 프로세스 개선은 프로세스와 혁신적인 아이디어와 기술로 부터의 피드백을 통해서 이루어진다.

CMM은 제품 자체의 품질을 평가하기 보다는 제품을 생산하는 프로세스의 효과적인 관리에 더욱 초점을 맞추고 있다. 이는 CMM의 적용을 통해 지속적인 프로세스 개선이 가능하며 개선된 공정은 제품 품질에 반영된다는 CMM의 기본 가정에 근거한다. 즉, 성숙된 소프트웨어 관리체계를 가진 조직은 ① 소프트웨어의 크기, 비용, 품질 그리고 개발일정의 예측력 향상, ② 프로세스 관리를 통한 제품의 결과 향상, ③ 소프트웨어 개발과정의 가시성 증대와 같은 조직적 이득을 가져온다는 것이다[Gaffney 1995].

최초의 CMM은 미 국방성의 소프트웨어 조달을 돕기위해 개발되었다. 미 국방성 계약자들은 계약체결을 위해 CMM을 기본으

로 하여 조직의 프로세스 개선을 해야할 필요성을 인식하게 되었다. 점차적으로 일반 기업에서도 내부적 개선을 위한 프레임웍으로 CMM을 수용하기 시작하였으며, 1994 이후에는 Motorola, Schlumber, Siemens 등의 기업을 대상으로 CMM 사용의 영향에 대하여 본격적인 연구가 시작되었다 [Herbsleb 1994].

ISO 9000 시리즈 중 용어를 정의하고 있는 ISO 8402에서는 품질을, "제품이나 서비스가 가지고 있는 명시적 또는 묵시적 요구를 만족시키는 능력에 관한 특성 및 특징의 전체"라고 정의하고 있다. 여기서 제품이란 ISO에서 정의한 제품의 분류 즉, 하드웨어, 소프트웨어, 가공재료, 서비스가 될 수 있다. 이 품질에 대한 정의에서 "묵시적 요구"라는 말에 주목할 필요가 있다. 문서 형태로 비교적 명확하게 정의된 명시적 요구사항 뿐만 아니라 고객이 생각할 때 공급자측에서 당연히 해주리라 믿고 명시하지 않은 묵시적 요구사항까지 파악하여 만족시켜주어야 한다.

ISO 8402에서는 품질관리를 "품질에 대한 요구사항을 만족하는데 사용되는 운용기법과 활동"이라고 정의하고 있다. 소프트웨어 모듈과 시스템 시험이 전형적인 예로서, 일반적으로 회사는 제품을 고객에게 납품하기 전에 제품검사를 통해 품질목표를 달성하려고 했었다. 그러나 품질관리의 철학은 부적합 품, 공정, 장비, 서비스가 고객에게 영향을 미치기 전에 찾아내자는 것이다. 품질문제는 종종 검사한 공정이나 활동과는 거리가 먼 요인에서 발생한다. 예를들어 부적합사항의 근본 원인이 부적절한 교육, 문서관리의 부실, 업무분장의 불명확 등에서 기인할 수도 있다. 품질관리가 품질확보에 중요한 역할을 담당하고는 있지만 그것만으로 품질이 보증된다고 볼 수는 없다.

ISO 8402에서는 품질보증을 "제품이나 서비스가 주어진 요구사항을 만족함에 대한 적절한 신뢰감을 주는데 필요한 모든 계획적이고 체계적인 활동"이라고 정의하고 있다. 이 정의에서 중요한 용어는 "활동이 계획적이고 체계적이어야 한다"는 것과 이들 활동이 "품질을 확보한다는 데에 믿음을 주어야 한다"는 것이다. 품질보증 활동은 품질요구사항을 만족시키기 위해 사용하는 운용기법과 활동을 포함한다.

소프트웨어 품질보증은 고품질의 소프트웨어를 생산하기 위한 관리기법이다. 과거의 소프트웨어 개발 결과는 비용과 일정이 계획에 비해 지연되는 경우가 많이 발생했는데 개발 및 관리 측면에서의 품질보증활동을 수행함으로서 이러한 문제를 예방할 수 있는 것이다. 소프트웨어에 대한 감리나 감사는 소프트웨어 품질을 보증하기 위한 대표적인 활동이 된다.

3. 정보시스템 감리

3.1 정보시스템 감리의 정의

국내에서는 아직까지 공공기관을 제외하고는 정보시스템의 감리제도가 보편화 되어 있지 않은 것이 현실이며, 이에따라 정보시스템 감리에 대한 학문적인 개념 역시 명확하게 정립되어 있지 않은 것이 사실이다. 그러나 '정보화촉진기본법 제15조의 2항'과 이에 근거한 '정보시스템감리기준'에 따르면, 정보시스템 감리는 "정보시스템의 효율성, 효과성, 안전성(신뢰성) 달성 여부를 독립적으로 평가하여 문제점의 개선을 권고하는 활동"으로 정의하고 있다.

국내의 정보시스템 감리제도는 1987년 행

정전산망에 대한 한국전산원의 감리 도입과 시행 이후 국내의 실정에 맞게 발전되어 온 분야이므로, 해외에서 우리의 감리제도와 완벽하게 일치하는 제도나 체계를 발견하기는 어렵다.

미국의 ISACA는 정보시스템감리를 "자동화된 정보처리시스템의 모든 측면 또는 특정부문을 검토하고 평가하는 각종 활동"이라고 정의하면서 시스템적인 성격 즉, 신뢰성, 유용성 및 합법성 측면을 강조하고 있다.

일본의 감사기준(1985)에서는 정보시스템 감리를 "감리대상으로부터 독립된 객관적인 입장에서 컴퓨터를 중심으로 하는 정보시스템을 종합적으로 점검 및 평가하여 관계자에게 조언하고 권고하는 것으로 정보시스템의 유효한 이용의 촉진과 폐해 제거를 동시에 추구하며 건전한 정보화를 도모하는 것"이라고 정의하고 있다.

감리와 유사하게 사용되는 용어로 감사가 있다. 사전적 의미로만 보자면 감사(監査)는 '감독하고 조사하는 것'이며 감리(監理)는 '감독하고 관리하는 것'이다. 따라서 감사의 특징은 오류의 지적과 시정에 역점을 두고 있으며, 정당성과 합법성을 강조하고, 계약의 준수나 회계처리의 정확성 등을 강조한다. 그러나 감리의 특징은 관리 및 엔지니어링에 역점을 두고 있으며 따라서 기획, 계획, 설계, 엔지니어링을 강조하고 정당성, 합리성, 적정성을 강조하는데 있다. 감사가 사후평가방식으로 법에의한 처벌과 의사결정에 유용한 정보로 활용될 수 있다면, 감리는 사전, 진행, 사후 평가방식으로 엔지니어링 의사결정에 유용한 정보로 사용할 수 있는 점이 다르다. 그러나 감사나 감리 모두 업무상 부정행위를 적발할 수 있다는 공통점이 있다.

우리나라 정보시스템 감리에 대한 접근방법은 기본적으로 세 가지로 나누어 볼 수 있다. 첫째, 외국 특히 미국과 일본을 중심으로 한 정보시스템 감사제도에 바탕을 둔 견해와 둘째, 정보시스템 이외의 타 분야 즉, 건설감리 등과 정보시스템 감리를 동일하게 보는 시각이 있으며, 셋째는 품질보증과 관련된 개념이다.

우선 미국과 일본을 중심으로 정보시스템 감리제도는 감리체계가 회계감사 이론에서 출발하고 있으며 위험관리를 기반으로 하는

<표 3-1> 감사와 감리의 구분

구 분	감 사	감 리
사전적 의미	감독하고 조사하는 것	감독하고 관리하는 것
특징	- 오류의 지적과 시정에 역점 - 계약 및 회계처리 측면 강조 - 정당성, 합법성을 강조 - 부정행위 적발 - 사후평가 방식	- 관리 및 엔지니어링에 역점 - 기획, 계획, 설계, 엔지니어링을 강조 - 정당성, 합리성, 적정성을 강조 - 부정행위 적발 - 사전평가 방식
결과 활용	- 법에 의한 처벌 - 의사결정에 유용한 정보 (투자 의사 결정)	- 법에 의한 처벌 - 의사결정에 유용한 정보 (엔지니어링 의사 결정)

내부통제제도에 근거를 두고 있다. 그리고 근본적으로 정보시스템 감리의 주된 관심사는 부정의 적발 등 정보시스템으로 인한 피해의 최소화와 안전성 확보와 같은 감사기능에 있다.

따라서 본 연구에서도 사전적 의미에만 연연하여 정보시스템에 대한 감사와 감리를 별도의 개념과 행위로 보지 않고, 감리는 감사를 포함하는 광의의 개념으로 인식하고자 한다.

다음으로 타 부문의 감리와 정보시스템 감리를 동일시 하는 시각이다. 이는 감리라는 동일한 용어를 사용하는데서 오는 인식의 차이로 보인다. 기본적으로 건설감리나 통신설비감리 등은 공사가 설계된대로 기술기준에 입각하여 시공되었는지를 관리하고 감독하는 행위다. '전기통신공사업법' 상에는 "감리라 함은 공사에 대하여 발주자의 위탁을 받은 감리업체가 설계도서 기타 관계서류의 내용대로 시공되었는가의 여부를 감독하고 품질관리, 시공관리 등에 대한 지도를 하며 발주자의 권한을 대행하는 것을 말한다" 라고 정의하고 있다. 그러나 정보시스템 관리와는 달리 국내에서 실시하고 있는 여타부문의 감리는 주로 개발 및 구축에 국한된 것이며 계획의 수립이나 운영부문에는 관련이 없어 정보시스템의 감리와는 기본적으로 상황이 다르다는 것에 유의해야 한다. 예를 들어 건설감리는 준공된 건물의 운영 및 관리상태에 대하여 감리하는 경우는 없다.

마지막으로 다른 측면으로는 소프트웨어 품질보증과 관련된 경우이다. 소프트웨어 품질보증은 국제표준의 다양한 개발과 많은 전문가들의 활동에 의해 급속하게 발전하고 있는 분야이다. 그러나 품질보증의 목적이 '품질이 좋은 정보시스템의 제공'에 맞추어져 있기 때문에, 그 시스템을 도입하여 효과적이며 효율적이고 안정적으로 운영하는지에 대한 사항은 품질보증에서는 제외되어 있다. 정시스템과 관련된 품질보증 기법과 기술은 정보시스템 감리의 도구이지 감리 자체는 아니라고 할 수 있다.

3.2 정보시스템 감리의 목적

정보시스템에 관련된 문제점으로는 정보자원의 효율적인 이용 등 투자를 중심으로 하는 경영상의 문제, 그리고 컴퓨터 범죄와 오류에 관련된 안전 및 보안상의 문제, 사생활 보호 문제, 회계처리 시스템의 문제 등이 두드러지게 드러난다. 정보시스템 감리는 이러한 여러 문제들을 종합적이고 합리적으로 해결하는 수단으로서 낭비를 제거하고 정보시스템의 역기능에 의한 피해 발생을 방지할 수 있다. 또한 정보시스템 개발시 품질이 확보된 시스템 개발을 실현하여 궁극적으로 정보시스템의 효과를 달성할 수 있도록 해 준다. 정보시스템 감리의 목적을 세분화 하여 정리하면 다음과 같이 다섯 가지로 나눌 수 있다.

첫째, 정보시스템의 효과성 증진이다. 즉, 소프트웨어 개발 과정 중 기획업무의 타당성 검토, 소프트웨어 개발 공정의 적합성 검토, 개발된 소프트웨어의 품질 보증 그리고 감리를 통한 정보시스템의 효과를 증진시킬 수 있다.

둘째, 정보시스템의 효율성 증진이다. 정보시스템 감리는 정보시스템을 개발하고 운영하는 조직중에 어디에 문제점이 있는지, 그리고 무엇을 우선적으로 개선해야 하는지를 제안하고 권고하여 정보시스템의 효율성을 증진시키는데 목적이 있다.

셋째, 정보시스템의 안전성을 확보하는데 있다. 정보시스템 안전성에는 데이터의 무결성, 기밀성 및 가용성이 포함된다. 특히

데이터의 무결성은 정보시스템의 효과를 보장하는 가장 중요한 요인인 동시에 정보시스템 감리가 보장하는 특성이기도 하다.

넷째, 정보시스템의 개발과 운영에 관련된 각종 법, 규정, 지침 또는 표준 등에 대한 준수 여부를 확인함으로서 준거성을 확보해 주는데 도움을 줄 수 있다는 것이다.

다섯째, 기타 목적으로 감리를 통해 사용자와 개발 및 운영자들간에 상호 이해를 증진시키고 정보시스템과 관련된 객관적인 정보를 제공함으로서 관련자들 간의 이해를 증진시켜 정보시스템의 전 생명주기 단계에서 정확한 개발이 이루어질 수 있게 한다.

각국의 정보시스템감리에 관련된 목적을 정리하면 <표 3-2>와 같다. 표에서 보는 것과 같이 각국 모두 거의 유사한 목적을 가지고 있으나, 한국은 개발과정에서의 준거성이나 효과성에 비중을 두고 있는 반면, 미국과 일본은 정보시스템의 운영상에 나타날 수 있는 안전성과 신뢰성 등을 강조하고 있으며, 유럽은 그 중간정도를 유지하고 있음을 알 수 있다.

<표 3-2> 한국과 외국의 정보시스템 감리 주요 목적

한국	미국	일본	유럽
준거성 효과성 효율성 실증성 가용성	효과성 안전성 무결성 기밀성 실증성 신뢰성	신뢰성 안전성 기밀성 유효성 준거성	효과성 효율성 안전성 실증성 준거성

3.3 선행 연구 사례

소프트웨어는 일반적으로 제조업과는 달리 기술이 인간중심으로 이루어져 있기 때문에 의학이나 물리학처럼 단지 실험만을 통해서 품질에 관한 모형을 세우고 검증하기는 대단히 어렵다. 다른 학문분야와 마찬가지로 소프트웨어에도 여러 가지 원인과 학습 그리고 대상 업무에 대한 이해 등 결과에 대한 수많은 변수들이 존재한다. 이에 따라 아직도 소프트웨어공학 분야에는 그 결과에 대한 이유를 명료하게 설명할 수 있는 모형이 거의 없으며, 특정 환경에서는 기술의 한계에 대한 인식이 부족하고, 분석이나 실험도 충분하지 못한 형편이다[Basili 1998].

더욱이 선진국에 비해 소프트웨어에 대한 인식이 미성숙되어 있는 우리나라에서는 소프트웨어 품질에 대한 연구는 물론, 감리활동이 소프트웨어 프로세스에 미치는 영향에 대하여는 그 연구사례를 찾아보기가 어렵다.

지금까지 소프트웨어공학 분야에서 대부분의 연구는 통계적 실증연구를 많이 이용했는데, Henry는 프로세스 개선효과를 평가하기 위해 선형회귀, 순위회귀, X^2 검증 등을 사용하였다[Heny 1995].

R. S. Pressman은 소프트웨어 개발 노력을 인시(person/months)로 표현하여, 정의(definition)와 개발(development) 단계에서 노력의 분배는 일반적으로 40-20-40의 법칙을 적용할 것을 제시하였다. 노력의 40% 이상은 분석과 설계에 배분되어야 하며, 40% 정도는 시험(testing), 그리고 나머지 20%는 구현(coding)에 분배되어야 함을 권유하고 있다. 이를 좀더 세분화 시키면 프로젝트 계획에 2-3%, 요구사항 분석에 10-25%, 그리고 설계에 20-25%를 분배하고 있다. 구현은 분석과 설계의 결과에 따라 실시되는 것이므로 약 15-20% 정도로 설계에 비해 상대적으로 노력이 적게 투여

된다. 소프트웨어의 시험과 그에 뒤따르는 디버깅(debugging)은 30-40%의 노력이 필요하며, 소프트웨어가 사람의 생명과 관련이 깊은 것일수록 그 노력의 비중은 더해져야 한다고 주장하였다[Pressman 1997].

정보시스템에 대한 종래의 연구에서는 소프트웨어 개발과정에서 사용자들의 심리적이고 행동과학적인 상태의 관리 중요성이 제기되어 왔다. 이와 같은 연구 흐름에서 중요한 발견은 사용자의 관여와 참여가 정보시스템의 제품과 서비스 측면에서 긍정적인 영향을 미친다는 것을 알게되었다는 점이다[Hartwick 1994].

T. Ravichandran과 Arun Rai는 정보시스템 개발에서 품질향상에 대한 중요한 발견을 하였는데 그것은 품질에 대한 산발적인 노력은 정보시스템의 품질 향상에 큰 영향을 미치지 못한다는 것이다[Ravichandran 1996]. 대신 그 산발적인 노력들의 상호작용을 결합하여 조직적인 시스템을 형성해야만 품질의 성능에 대하여 전체적으로 관찰할 수 있는 수준의 중추적인 역할을 하게된다는 것이다. 결론적으로 최고경영자의 리더쉽과 실무적인 관리 기반, 그리고 개발작업 수준의 활동들을 연결하는 것이 절대적이고 중요하다는 것이다.

한국전산원은 "감리결과 분석을 통한 주요 문제점 및 개선사례 연구"에서 자체적으로 수행했던 26건의 감리보고서를 조사했다. 조사대상 소프트웨어 프로젝트를 '프로젝트 관리', '응용시스템', '데이터베이스 및 자료 변환' 등 3분야로 나누고 각 분야별로 사업수행단계를 4단계 즉, '계획', '분석', '설계', '구현'으로 나누어 각 단계에서 지적된 주요 문제점과 개선권고사항을 분석하였다[한국전산원 2001].

'프로젝트 관리'에 있어 사업수행단계별로 지적된 문제점은 총 158건으로, 계획단계가 10건, 분석단계가 8건, 설계단계가 32건, 구현단계가 108건으로 나타났다.

'응용시스템'에 있어 사업수행단계별로 지적된 문제점은 총 96건으로, 계획단계에서는 지적사항이 없으며, 분석단계 2건, 설계단계 9건, 구현단계 85건으로 구현단계에서 많은 지적사항이 발생되었다.

'데이터베이스 및 자료 변환'에 있어 사업수행단계별로 지적된 문제점은 총 157건으로 계획단계 4건, 분석단계 7건, 설계단계 43건, 구현단계 103건으로 대부분 구현단계에서 지적사항이 발생되었다.

문대원은 1996년부터 2000년까지 시행된 124건의 감리보고서를 대상으로 감리분야를 대구분, 중구분으로 나눈 뒤 세부항목별로 분석하여 정보시스템 개발 프로젝트의 성공요인을 실증적으로 식별하였다. 프로젝트의 성공여부는 감리결과를 정량화 하여 결정하였으며, 그 결과 다음과 같이 15항목의 성공변수를 중요한 순서대로 판별하였다[문대원 2001].

- 프로젝트 관리자의 발주자와의 협조/신뢰 관계
- 주 개발업체 개발팀 구성의 적정성
- 발주처 최고경영자 의사결정의 적시성
- 사업담당자의 전체 업무 이해도
- 사용자의 사업 수용도
- 법, 제도, 정책이 정비 또는 지원된 상태에서의 사업추진 여부
- 프로젝트 관리자의 대내 리더쉽
- 사업 담당자의 프로젝트 관리의 적극성
- 사용자 요구사항의 정확성
- 사업기간
- 문서화의 충실도
- 첨단기술 적용에 따른 사업 위험도
- 컨소시엄 참여 협력업체 수
- 개발업체의 유사 프로젝트 수행 경험

• 사업 담당자의 정보기술 이해도

위 결과에서 나타나 있는 것 처럼 가장 영향력이 있는 상위 3개 항목은 모두 의사소통과 조직화 그리고 관리의 중요성을 나타내고 있다. 특히 상위 10개 항목 중에서 프로젝트의 특성의 나타내는 항목은 '사업기간' 하나뿐이어서 개발 프로젝트의 성공은 프로젝트 자체 특성보다는 이를 수행하는 인적자원과 이와 관련된 여러 관리통제 요인이 더 중요하다는 것을 보여주고 있다.

이상엽은 "소프트웨어 프로세스 성숙도가 프로젝트 성과에 미치는 영향"에서, 프로세스 성숙도가 개선되면 프로젝트의 납기 준수율과 프로젝트 규모의 예측율에 정(+)의 영향을 미침을 발견하였다. 그리고 프로세스 성숙도는 고객 의사소통을 통하여 프로젝트 성과에 간접적인 영향을 미침을 알 수 있었으며, 프로세스 성숙도는 팀원 의사소통을 통하여 프로젝트 성과에 간접적인 영향을 미치고 있음을 알게되었다[이상엽 2000].

4. 정보시스템 감리결과 조사 및 분석

4.1 조사 대상 및 범위

정보시스템 감리는 감리 내용에 따라 감리 형태를 사업감리와 운영감리로 나누고, 감리 시점에 따라 감리 종류를 중간감리와 최종감리로 나눈다. 사업감리는 주로 프로젝트 방식으로 추진되는 정보시스템 개발사업이 대상이 되며, 운영감리는 개발된 정보시스템에 대한 운영과 유지보수가 주 감리 대상이 된다. 중간감리는 정보시스템의 설계단계가 끝난 후 실시되며, 최종감리는 개발 된 정보시스템이 사용자에 인도되기 직전에 실시된다. 감리 형태에 따른 감리 내용과 감리 목적은 <표 4-1>과 같다[문대원 1999].

본 연구에서는 국내의 전문감리업체인 H 감리원에서 1999년부터 2002년까지 실시한 공공기관의 정보시스템 개발사업 59건에 대한 감리실적(횟수) 총 74회을 대상으로 조사를 실시했다.

<표 4-1> 감리 형태에 따른 감리 내용 및 목적

감리 형태	감리 내용	감리 목적
사업감리	· 정보시스템 중장기 계획 · 응용시스템 분석 · 설계 감리 · 응용시스템 구현 감리 · 시스템 시험 및 통합사업 감리	개발사업의 성공적인 수행을 목적으로 하며, 개발사업의 진행단계에 따라 실시
운영감리	· 시스템 통제 준거성 감리 · 시스템 안전성 감리 · 시스템 효율성 감리 · 시스템 효과성 감리	정보시스템의 설비, 응용 소프트웨어 운영, 오류 및 보안 등을 점검하는 감리로서 주기적 또는 특별한 사안이 발생하면 실시

조사는 정보시스템 개발사업 즉, 사업감리를 대상으로 하였으며, 중간감리와 최종감리가 모두 포함되어 있다. 자료의 일관성을 위해 사업규모가 특별히 크거나 작지 않은 사업(주로 2억 이상 30억 미만)을 대상으로 하였으며, 따라서 사업별 감리 횟수도 1회(최종감리)로 끝났거나 아니면 1, 2차(중간감리, 최종감리)로 나누어 실시된 감리를 대상으로 조사하였다.

조사 대상의 정보시스템 개발 사업 59건 중 감리를 1회만 실시한 사업은 44개이며, 2회(중간감리와 최종감리)를 같이 실시한 사업은 15개다. 따라서 전체 감리횟수는 74회(44 + 15 * 2)가 된다.

년도별 감리횟수와 사업규모별 감리횟수는 <표 4-2>, <표 4-3>과 같다.

<표 4-2> 년도별 감리 횟수

구분 / 년도	전체	중간 감리	최종 감리
1999	12		12
2000	12	3	9
2001	50	12	38
계	74	15	59

<표 4-3> 사업규모별 감리 횟수

구분 / 금액(원)	전체	중간 감리	최종 감리
5억 미만	38	5	33
5억 - 10억	15	3	12
10억 - 15억	4	1	3
15억 - 20억	3	1	2
20억 이상	14	5	9
계	74	15	59

4.2 조사 방법

먼저 정보시스템 개발사업의 감리보고서 74건을 모두 입수하여, 보고서 내용 중 '개선권고사항'을 중점적으로 조사했다. '개선권고사항'은 크게 4분야(프로젝트관리 및 품질, 응용시스템, 데이터 및 데이터베이스, 아키텍처 및 보안)로 나누어 각 분야마다 다수의 세부항목으로 지적사항을 기술하고 있다.

감리보고서에 명시된 개선권고사항 세부항목들은 감리전문기관의 전문가들에 의해 지적된 사항이므로 그 내용의 타당성이나 신뢰성에 대해서는 문제가 없을 것으로 판단된다.

감리보고서의 개선권고사항에서 지적된 세부항목들은 감리인마다 표현방법이 다소 다를 수 있음은 물론 그 내용이 지나치게 세부적이거나 아니면 추상적일 수도 있다.

따라서 본 연구에서는 감리보고서에 명시된 개선권고사항의 세부항목들을 일정한 틀 즉, 표준화된 항목에 맞추어 다시 정리하고 지적된 횟수를 집계하기 위해서 '지적사항 조사표'를 작성하였다. '지적사항 조사표'를 구성하고 있는 세부 조사항목은 <표 4-4>와 같다.

'지적사항 조사표'의 개발단계는 소프트웨어 고전수명주기모형(Classic life Cycle Model)에 나타나는 5단계를 기초로 하였으며[Pressman 1997], 여기에 문대원, 장시영이 제시한 정보시스템 감리지침의 검토항목과[문대원 1999], 감리 실무자 그리고 대학의 소프트웨어공학 분야 교수들의 의견을 모아 세부항목들을 선정하고 정리 하였다.

<표 4-4> 지적사항 조사표의 세부 조사항목

· 계획단계	1. 범위계획
	2. 일정계획
	3. 비용계획
	4. 자원계획
	5. 위험 및 품질계획
	6. 기타
· 분석단계	1. 요구사항의 타당성 검토
	2. 요구사항 분석내용의 정확성
	3. 요구사항의 내부적 일관성
	4. 요구사항의 추적 가능성
	5. 요구사항의 설계, 구현, 유지보수 가능성
	6. 분석 결과의 문서화 및 관리
	7. 기타
· 설계단계	1. 요구사항 분석내용의 설계 반영
	2. 데이터 설계
	3. 데이터베이스 설계
	4. 시스템 구조(구성요소) 설계
	5. 내부 인터페이스 설계(모듈-모듈)
	6. 외부 인터페이스 설계(소프트웨어-사람)
	7. 처리절차 설계(상세설계)
	8. 시험 설계
	9. 보안 및 품질 설계
	10. 설계의 내부적 일관성
	11. 설계자료(명세서)의 문서화 및 관리
	12. 기타
· 구현단계	1. 소프트웨어 오류
	2. 기능 누락
	3. 사용자 편리성
	4. 사용 프로그램 언어
	5. Source Program의 관리
	6. 기타
· 시험단계	1. 소프트웨어 단위시험
	2. 소프트웨어 통합시험
	3. 요구사항 검증시험
	4. 시스템 통합시험
	5. 수락시험
	6. 신뢰도 검증시험
	7. 기타

4.3 지적사항 조사 및 분석 결과

4.3.1 총괄 현황

중간감리와 최종감리 모두를 합한 전체 74건의 감리 결과, 개선권고사항으로 지적된 건수 즉, 세부 지적항목은 총 1301건이다.

1301건의 세부 지적항목 중 긴급개선사항은 463건, 통상개선사항은 681건, 권고사항은 157건이다. 여기서 '긴급개선사항'은 지적된 사항이 중대하여 프로젝트 다음 단계를 위해 시급하게 수정 또는 보완을 해야 할 사항을 의미한다. '통상개선사항'은 긴급개선에 비해서는 덜 중대하고 시급하나 시간을 갖고 수정 또는 보완을 해야 할 사항이다. '권고사항'은 프로젝트 다음 단계를 위해 도움이 되는 사항으로 관련자간 협의에 의해 추진할 사항이다.

전체 1301건의 지적사항 중 계획단계에 해당하는 지적사항은 247건(19%), 분석단계의 지적사항은 95건(7%), 설계단계의 지적사항은 697건(54%), 구현단계의 지적사항은 167건(13%), 시험단계의 지적사항은 95건(7%)으로 설계단계에서 가장 많은 지적사항이 발생했다.

4.3.2 프로세스 단계별/감리 종류별 지적 현황

(1) 계획단계

계획단계의 전체 지적사항은 총 247건으로, '위험 및 품질계획'에 94건(38%), '일정계획'에 57건(23%)이 지적되었으며, '범위계획'에 34건(14%), '자원계획'에 20건(8%)이 지적되었다.

중간감리의 지적사항은 '위험 및 품질계획'이 26(43%)건, '일정계획'이 14(23%)건, '범위계획'이 9(15%)건의 순이다.

최종감리의 지적사항은 '위험 및 품질계획'이 68(37%)건, '일정계획'이 43(23%)건, '범위계획'이 25(13%)건으로 나타났다.

중간감리나 최종감리 모두 위험 및 '품질계획'에 제일 많은 지적사항이 발생한 것은, 감사자는 위험과 품질문제에 관심을 가지고 있으나 정보시스템 발주자나 개발자는 아직도 위험이나 품질에 대한 인식이 결여되어 있는 결과라고 볼 수 있다. 또한 '일정계획'에 많은 지적(전체 57건)이 발생하고 있음은 발주자가 무리하게 수주를 한 결과 그 일정이 지켜지지 못한 결과라고 분석된다.

<표 4-5> 총괄 현황

구분 / 단계	전 체				중간감리				최종감리			
	감리회수 : 74				감리회수 : 15				감리회수 : 59			
	계 (%)	긴급	통상	권고	계 (%)	긴급	통상	권고	계 (%)	긴급	통상	권고
합계	1301 (100)	463	681	157	260 (100)	82	148	30	1041 (100)	381	533	127
계획단계	247 (19)	65	142	40	61 (23)	11	40	10	186 (18)	54	102	30
분석단계	95 (7)	25	60	10	28 (11)	6	20	2	67 (6)	19	40	8
설계단계	697 (54)	39	373	85	152 (58)	55	80	17	545 (52)	184	293	68
구현단계	167 (13)	89	61	17	15 (6)	9	5	1	152 (15)	80	56	16
시험단계	95 (7)	45	45	5	4 (2)	1	3		91 (9)	44	42	5

<표 4-6> 계획단계의 세부항목별 지적사항

단계	세부 사항	전체				중간감리				최종감리			
		계 (%)	긴급	통상	권고	계 (%)	긴급	통상	권고	계 (%)	긴급	통상	권고
계획 단계	계	247 (100)	65	142	40	61 (100)	11	40	10	186 (100)	54	102	30
	범위계획	34 (14)	14	16	4	9 (15)	3	5	1	25 (13)	11	11	3
	일정계획	57 (23)	30	20	7	14 (23)	3	10	1	43 (23)	27	10	6
	비용계획	1 (0.4)		1						1 (0.5)		1	
	자원계획	20 (8)	3	14	3	3 (5)		3		17 (9)	3	11	3
	위험 및 품질계획	94 (38)	13	65	16	26 (43)	3	16	7	68 (37)	10	49	9
	기타	41 (17)	5	26	10	9 (15)	2	6	1	32 (17)	3	20	9

(2) 분석단계

분석단계의 지적사항은 전체 95건으로 긴급개선 25건, 통상개선 60건, 권고사항 10건이다. 총 95건의 지적사항 중 '분석결과의 문서화 및 관리'에 35건(37%), '요구사항의 추적 가능성' 27건(28%), '요구사항 분석내용의 정확성' 19건(20%) 순으로 지적되었다.

중간감리에서는 '분석결과의 문서화 및 관리' 항목이 10(36%)건, 요구사항의 추적 가능성이 7(25%)건으로 나타났으며, 최종감리에서도 '분석결과의 문서화 및 관리' 항목이 25(37%)건으로 가장 많은 지적을 받았

<표 4-7> 분석단계의 세부항목별 지적사항

단계	세부사항	전체				중간감리				최종감리			
		계 (%)	긴급	통상	권고	계 (%)	긴급	통상	권고	계 (%)	긴급	통상	권고
분석 단계	계	95 (100)	25	60	10	28 (100)	6	20	2	67 (100)	19	40	8
	요구사항의 타당성	7 (0.7)	3	3	1	3 (11)		2	1	4 (6)	3	1	
	요구사항 분석 내용의 정확성	19 (20)	7	10	2	8 (29)	4	4		11 (16)	3	6	2
	요구사항의 내부적 일관성	2 (2)		2						2 (3)		2	
	요구사항의 추적 가능성	27 (28)	7	19	1	7 (25)		7		20 (30)	7	12	1
	요구사항의 설계, 구현 및 유지보수 가능성	3 (3)		2	1					3 (4)		2	1
	분석결과의 문서화 및 관리	35 (37)	8	23	4	10 (36)	2	7	1	25 (37)	6	16	3
	기타	2 (2)		1	1					2 (3)		1	1

으며, 다음으로 많은 지적을 받은 항목은 '요구사항의 추적 가능성'으로 20(30%)건이다.

'분석결과의 문서화 및 관리' 항목에서 가장 많은 지적이 발생하고 있는 것은, 이를 통해 소프트웨어 개발 과정에서 담당자들이 문서화를 중요시 않고 있음은 물론, 소프트웨어 개발 과정에 정형화된 프로세스 모형을 사용하고 있지 않은 결과라고 분석된다.

(3) 설계단계

설계단계의 지적사항은 감리 전체 지적사항 1301건 중 697(54%)으로 가장 높은 비중을 차지하고 있다.

설계단계의 지적사항 전체 697건 중 '데이터베이스 설계' 항목에서 171(25%)건으로 가장 많이 나타났으며, '보안 및 품질 설계' 항목에서 124(18%)건, '설계자료(명세서)의 문서화 및 관리' 항목에서 109(16%)건, '데이터 설계' 항목에서 77건(11%), '처리절차 설계(상세설계)' 항목에서 49건(7%) 순으로 나타났다.

중간감리나 최종감리 모두 '데이터베이스 설계 항목'에서 가장 많은 지적사항이 발생했으며, 다음으로 '보안 및 품질 설계', '설계자료(명세서)의 문서화 및 관리' 순으로 지적사항이 발생했다.

'데이터베이스 설계'에 가장 많은 지적사항이 발생한 것은 데이터베이스 자체가 소

<표 4-8> 설계단계의 세부항목별 지적사항

단계	세부사항	전체				중간감리				최종감리			
		계 (%)	긴급	통상	권고	계	긴급	통상	권고	계	긴급	통상	권고
설계 단계	계	697 (100)	239	373	85	152 (100)	55	80	17	545 (100)	184	293	68
	분석내용의 설계 반영	30 (4)	16	12	2	12 (8)	5	6	1	18 (3)	11	6	1
	데이터 설계	77 (11)	30	37	10	18 (12)	9	8	1	59 (11)	21	9	9
	데이터베이스 설계	171 (25)	71	85	15	27 (18)	13	11	3	144 (26)	58	74	12
	시스템 구조 (구성요소) 설계	27 (4)	7	14	6	11 (7)	2	5	4	16 (3)	5	9	2
	내부인터페이스 설계(모듈-모듈)	15 (2)	6	7	2	2 (1)	1	1		13 (2)	5	6	2
	외부인터페이스 설계(소프트웨어-사람)	28 (4)	7	17	4	7 (5)		6	1	21 (4)	7	11	3
	처리절차 설계 (상세설계)	49 (7)	24	19	6	14 (9)	8	5	1	35 (6)	16	4	5
	시험설계	34 (5)	11	17	6	9 (6)	3	5	1	25 (5)	8	2	5
	보안 및 품질 설계	124 (18)	33	71	20	24 (16)	5	16	3	100 (18)	28	55	17
	설계의 내부적 일관성	18 (3)	6	12		7 (5)	2	5		11 (2)	4	7	
	설계자료 (명세서)의 문서화. 및 관리	109 (16)	25	73	11	20 (13)	7	11	2	89 (16)	18	62	9
	기타	5 (0.7)	3	9	3	1 (0.6)		1	14	3 (0.5)	8	3	

프트웨어 개발에서 차지하는 비중이 크기 때문임은 물론, 설계과정에 여러 가지 기술적인 문제들이 내포되어 있기 때문인 것으로 분석된다.

'보안 및 품질 설계' 항목이 두 번째로 많은 지적을 받고 있음은, 계획단계에서 '위험 및 품질관리'에 가장 많은 지적을 받은 것과 무관하지 않으며, 정보시스템을 개발하는데 있어 품질과 보안에 아직 많은 관심을 기우리지 않고 있음을 다시 증명하는 것으로 해석된다.

'설계자료(명세서)의 문서화 및 관리' 항목이 세 번째로 많은 지적을 받고 있는 것은, 분석단계에서 '분석결과의 문서화 및 관리' 항목이 가장 많은 지적을 받은 것과 유사한 이유로 해석되며, 이를 통해 국내 소프트웨어 개발에는 프로세스 모형이 잘 적용되고 있지 않음은 물론 프로세스 성숙도가 낮다는 것을 알 수 있다.

(4) 구현단계

구현단계의 지적사항은 전체 167건으로 긴급개선 89건, 통상개선 61건, 권고사항 17건으로 나타났다. 구현단계의 지적사항은 중감감리 보다 대부분 최종감리에서 발생하고 있는데, 그것은 중간감리가 설계를 마친 후(구현 전)에 실시되기 때문이다.

구현단계의 지적사항 전체 167건 중 '소프트웨어 오류'가 54건(32%), '기능 누락' 49건(29%), '사용자 편리성' 38건(23%) 순으로 나타났다.

'소프트웨어 오류' 항목이 가장 많이 지적되고 있는 것은, 설계단계의 '처리절차 설계(상세설계)'의 오류가 '소프트웨어 오류'로 이어지는 경우가 있음은 물론 개발 담당자의 경력과도 관련이 있는 것으로 판단된다.

그러나 '기능 누락' 항목이 두 번째로 많이 지적되고 있는 것은, 설계단계에서 많이 지적되고 있는 '처리절차 설계(상세설계)'상의 문제는 물론, '설계자료의 문서화 및 관리'에서 누락되었거나 잘못 표기된 문제점들이 구현단계의 '기능 누락'으로 이어진 결과라고 판단된다.

<표 4-9> 구현단계의 세부항목별 지적사항

단계	세부사항	전체				중간감리				최종감리			
		계 (%)	긴급	통상	권고	계 (%)	긴급	통상	권고	계 (%)	긴급	통상	권고
구현 단계	계	167 (100)	89	61	17	15 (100)	9	5	1	152 (100)	80	56	16
	소프트웨어오류	54 (32)	42	8	4	5 (33)	5			49 (32)	37	8	4
	기능 누락	49 (29)	33	13	3	5 (33)	3	2		44 (29)	30	11	3
	사용자 편리성	38 (23)	8	23	7	1 (7)		1		37 (24)	8	22	7
	사용 프로그램 언어	2 (1)		1	1	1 (7)		1		1 (0.6)			1
	Source Program 관리	22 (13)	6	14	2	3 (20)	1	1	1	19 (13)	5	13	1
	기타	2 (1)		2						2 (1.3)		2	

(5) 시험단계

시험단계의 지적사항은 전체 95건으로 긴급개선 45건, 통상개선 45건, 권고사항 5건으로 나타났다. 시험단계의 지적사항이 대부분 최종감리에서만 나타나고 있는 것은 구현단계와 마찬가지로 중간감리는 설계가 완료된 후에 실시되기 때문이다.

전체 95건의 지적사항 중 '시스템 통합시험'에 34(36%)건, '소프트웨어 통합시험'에 20(21%)건, '소프트웨어 단위시험'에 14(15%)건의 순서로 지적사항이 발생했다.

'시스템 통합시험'과 '소프트웨어 통합시험'에 많은 지적사항이 발생하고 있는 것은, 그것 자체가 가지고 있는 기술적인 어려움도 있겠지만 다른 시스템 개발 담당자들과의 업무적 대화는 물론 동일한 시스템을 개발하는 담당자들과도 업무적 대화가 잘 이루어지지 않고 있음을 보여주는 것으로 해석된다.

4.3.3 지적사항 발생 원인 조사

고전적수명주기의 특징은 앞 단계에서 작성된 산출물을 기반으로 다음 단계의 작업이 진행된다는 점이다. 따라서 앞 단계의 작업에 문제(오류)가 있을 경우 그 문제는 다음 단계의 문제로 이어지게 된다. 설계단계에서 지적된 문제는 설계과정 자체에서 발생된 문제 일 수도 있지만 분석단계에서 발생한 문제가 설계단계에서 표출된 것 일 수 있다. 이와 같이 구현단계에서 발견된 문제들은 구현단계 자체에서 발생한 문제 일 수도 있지만 대부분은 설계단계의 문제가 구현단계에서 나타나게되는 것이다.

본 연구에서는 각 단계에서 발생한 지적사항들을 면밀하게 검토하여 그 지적시항을 유발하게 된 원인이 어느 단계의 어느 항목에 있나를 조사했다. 그 결과는 <표4-11>과 같다.

설계단계에서 발생한 전체 지적사항 697건 중 11건(1.6%)이 앞 단계인 계획단계와

<표 4-10> 시험단계의 세부항목별 지적사항

단계	세부사항	전체				중간감리				최종감리			
		계 (%)	긴급	통상	권고	계 (%)	긴급	통상	권고	계 (%)	긴급	통상	권고
시험 단계	계	95 (100)	45	45	5	4 (100)	1	3		91 (100)	44	42	5
	소프트웨어 단위시험	14 (15)	11	3		1 (25)	1			13 (14)	10	3	
	소프트웨어 통합시험	20 (21)	7	13	7	1 (25)		1		19 (21)	7	12	
	요구사항 검증시험	8 (8)	4	4						8 (9)	4	4	
	시스템 통합시험	34 (36)	17	16	1	1 (25)		1		33 (36)	17	15	1
	수락시험	6 (6)	2	4		1 (25)		1		5 (5)	2	3	
	신뢰도 검증시험	11 (12)	3	4	4					11 (12)	3	4	4
	기타	2 (2)	1	1						2 (2)	1	1	

<표 4-11> 지적사항 발생 원인 조사표

지적사항(발생) 항목			원인 항목		
단계	세부항목	지적 횟수	단계	세부항목	횟수
설계	분석내용의 설계 반영	7	분석	분석내용의 정확성	3
				요구사항 추적가능성	2
				기타	2
	보안 및 품질 설계	4	계획	위험 및 품질계획	4
구현	소프트웨어 오류	8	설계	처리절차 설계	5
				데이터 설계	1
				외부인터페이스 설계	1
				보안 및 품질 설계	1
	기능 누락	15	분석	분석내용의 정확성	3
			설계	분석내용의 설계반영	4
				처리절차 설계	4
				보안 및 품질 설계	4
	사용자 편리성	22	설계	외부인터페이스 설계	17
				처리절차 설계	3
				보안 및 품질 설계	2
시험	소프트웨어 단위시험	5	계획	위험 및 품질계획	1
			설계	데이터베이스 설계	2
			구현	소프트웨어 오류	2
	소프트웨어 통합시험	10	설계	시험설계	8
			구현	소프트웨어 오류	2
	시스템 통합시험	12	설계	시험설계	8
			계획	위험 및 품질계획 외	2
			구현	기타	2
	요구사항 검증시험	3	설계	데이터베이스 설계	3

분석단계에서 각각 4건과 7건씩 기인된 것으로 분석되었다.

구현단계에서 발생한 전체 지적사항 167건 중 45건(27%)이 앞 단계에서 기인된 것으로 분석되었는데, '소프트웨어 오류' 8건은 설계단계에서 기인되었으며, '기능 누락' 15건은 분석단계에서 3건, 설계단계에서 12건이 각각 기인되었다. 그리고 '사용자 편리성' 22건은 모두 설계단계에서 기인되었다.

시험단계에서 발생한 전체 지적사항 95건 중 30건(32%)이 앞 단계에서 기인된 것으로 분석되었는데, '시스템통합시험' 항목에 발생한 12건은 설계단계의 '시험설계' 항목과 '위험 및 품질계획' 항목에서 기인되었다. 그리고 '소프트웨어 통합시험' 항목 10건 중 대부분은 설계단계의 '설계시험'에서 기인된 것으로 나타났다.

이와 같은 조사결과, 소프트웨어 프로세스 각 단계에서 발생하는 문제점들은 그 원인이 많은 부분 앞 단계에서 발생하고 있음이 확인되었다. 따라서 현재처럼 개발사업당 1회나 2회정도 실시하는 감리를 좀더 세분화 하여 소프트웨어 프로세스 단계별로 감리를 실시한다면, 다음 단계인 설계와 구현 그리고 시험단계에서 발생하는 많은 오류가 사전에 방지될 것으로 판단된다.

4.3.4 감리종류별 비교 및 분석 결과

정보시스템 감리가 소프트웨어 품질에 어떤 영향을 미치는가를 알아보기 위해 감리종류별로 지적사항 횟수를 비교해 보았다 <표 4-12>. 감리를 2회(중간감리와 최종감리) 실시한 사업은 15개이며, 1회(최종감리)만 실시한 사업은 44개다.

<표 4-12> 감리종류별 지적사항

(*) 사업당 평균 지적횟수

구분 / 단계	전체	2회 실시 (사업 수 : 15)		1회실시 (사업 수 : 44)
		중간감리(*)	최종감리(*)	최종감리(*)
계획단계	247	61(4.0)	50(3.3)	136(3.3)
분석단계	95	28(1.9)	19(1.3)	48(1.1)
설계단계	697	152(10.1)	120(8.0)	425(9.6)
구현단계	167	15(1.0)	64(4.3)	88(2.0)
시험단계	95	4(0.3)	33(2.2)	58(1.3)
계	1301	260(17.3)	286(19.0)	755(17.1)

<표 4-12>에서 보는 것과 같이 감리를 2회 실시한 15개 사업을 대상으로 조사한 결과 계획단계, 분석단계, 설계단계에서 모두 최종감리의 지적횟수가 중간감리의 지적횟수에 비해 현저히 감소하였음을 알 수 있다. 이것은 중간감리를 함으로서 각 단계마다 사전에 문제점이 지적되어 개선됨으로 그 효과가 최종감리에 미친 것으로 분석된다. 즉, 정보시스템 감리가 소프트웨어 품질에 긍정적인 영향을 미치고 있음을 알 수 있다.

그러나 감리를 2회 실시한 사업과 1회만 실시한 사업의 사업당 평균 최종감리 지적횟수를 비교해 본 결과, 설계단계를 제외하고 오히려 분석단계와 구현단계 그리고 시험단계에서 최종감리만 실시한 지적횟수가 더 적게 나타났다.

이런 현상은 최종감리만 실시하는 사업에서 더 많은 지적사항이 발생할 것이라는 상식적인 기대를 벗어나는 것으로, 그 이유는

<표 4-13> 5억미만 사업의 감리종류별 지적사항

구분 / 단계	2회 실시 (사업수 : 5)		1회실시 (사업수 : 33)
	중간감리(*)	최종감리(*)	최종감리(*)
계획단계	13(2.6)	12(2.4)	85(2.6)
분석단계	7(1.4)	3(0.6)	30(0.9)
설계단계	45(9.0)	30(6.0)	257(7.8)
구현단계	4(0.8)	17(3.4)	54(1.6)
시험단계	-	7(1.4)	39(1.2)

감리를 2회 실시하는 사업이 1회만 실시하는 사업보다 대부분 시스템의 규모가 크기 때문에 절대적인 지적횟수도 많이 발생하게 된 것으로 분석된다.

위와 같은 조사결과 상의 문제점을 해소하기 위해 사업규모가 비슷한 5억 미만의 사업 38건을 대상으로 조사를 실시했다. 조사 결과는 <표 4-13>과 같다.

조사 결과 감리를 2회 실시한 사업 5건의 계획단계, 분석단계, 설계단계 모두에서 중간감리보다 최종감리의 지적횟수가 적게 나타났다. 또한, 감리를 2회 실시한 사업의 최종감리와 1회만 실시한 사업의 최종감리의 사업당 평균 지적횟수를 비교한 결과 계획단계, 분석단계, 설계단계 모두에서 1회만 실시한 최종감리의 평균 지적횟수가 많이 나타나고 있음을 알 수 있다.

이와 같은 조사 결과를 통해, 감리 횟수를 증가시킨다면 즉, 소프트웨어 프로세스 단계마다 감리를 한다면 다음 단계에서 발생할 문제점을 사전에 감소시킬 수 있어, 감리가 소프트웨어 프로세스의 계획단계와 분석단계 그리고 설계단계에서는 소프트웨어 품질에 긍정적인 영향을 미치고 있다는 것을 확인 할 수 있다.

그러나 구현단계와 시험단계에서는 오히려 감리를 1회만 실시한 최종감리에서 사업당 평균 지적횟수가 적게 나타나고 있는데, 이와 같은 현상은 역시 최종감리만 실시하는 사업보다 감리를 2회 실시하는 사업규모가 일반적으로 크기 때문에 절대적인 지적횟수가 많은 것으로 분석되나, 이는 향후에 더 많은 연구가 되어야 할 것으로 판단된다.

5. 결론

공공분야의 정보화가 활발하게 진행되면서 우리나라는 1987년부터 한국전산원이 주관하여 정보시스템에 대한 감리를 실시해 오고 있다. 1990년 중반 이후 급격히 증가된 감리 수요와 함께 감리지침의 개정 등 감리 기술과 발전방안에 대하여는 많은 연구가 진행되어 왔으나, 감리활동이 소프트웨어 품질에 직접적으로 어떤 영향을 미치고있는가에 대하여는 연구가 소홀히 되어왔다.

본 연구에서는 1999년부터 2001년까지 3년동안 H감리원이 실시한 59개 정보시스템 사업에 대한 74건의 감리보고서를 조사했다.

소프트웨어 프로세스 단계별로 지적사항을 살펴보면, 전체 지적사항 1301건 중 설계단계에서 697(54%)건이 발생해 가장 큰 비중을 차지하였고, 다음으로 계획단계에서 247(19%)건, 구현단계에서 167(13%)건 순으로 발생했다.

계획단계에서는 세부항목으로 '위험 및 품질계획'에 가장 많은 지적사항이 발생해서, 우리나라의 공공소프트웨어 개발사업에는 위험과 품질 문제가 소홀히 되고 있음을 알 수 있다.

분석단계에서는 세부항목으로 '분석결과의 문서화 및 관리'에 가장 많은 지적사항이 발생해서, 우리나라의 공공소프트웨어 개발사업은 문서화가 소홀이 되고 있음은 물론 프로세스 단계별로 개발이 진행되고 있지 않음을 알 수 있다.

설계단계에서는 '데이터베이스 설계' 항목에 가장 많은 지적사항이 발생했으며, 다음으로 '보안 및 품질설계', '설계자료의 문서화 및 관리' 순으로 지적사항이 발생했다. 이를 통해 국내의 공공소프트웨어 개발사업은 품질관리가 소홀이 되고 있음을 다시 한

번 알 수 있다.

지적사항 발생 원인조사 결과, 구현단계의 지적사항 전체 167건 중 44건(26%)이 앞 단계에서 기인됐으며, 시험단계의 전체 지적사항 95건 중 26건(27%)이 앞 단계에서 기인된 것으로 확인됐다. 이를 통해 정보시스템 감리가 프로세스 단계별로 실시된다면 많은 문제점이 사전에 예방되어 다음 단계에서는 지적사항(문제점)이 감소할 수 있음을 알 수 있다.

감리를 1회(최종감리)만 실시한 사업과 2회(중간감리와 최종감리)를 실시한 사업의 각 단계별 지적사항 발생 건수를 비교해 본 결과, 감리를 2회 실시한 사업의 계획단계, 분석단계, 설계단계에서 현저하게 감소했음을 알 수 있었다. 이것은 정보시스템 감리를 프로세스 단계별로 실시할 필요성을 증명해 주는 것이며, 이를 통해 정보시스템 감리가 소프트웨어 품질에 긍정적인 영향을 미치고 있음을 알 수 있다.

참고문헌

[1] 문대원, "공공부문 정보시스템 개발 프로젝트의 성공요인 도출을 위한 탐색 연구", 국민대학교 대학원, 박사학위논문, 2001

[2] 문대원, 장시영, 「정보시스템 감리」, 명경사, 1999

[3] 이상엽, "소프트웨어 프로세스 성숙도가 프로젝트 성과에 미치는 영향에 관한 연구", 한국외국어대학교 대학원 박사학위논문, 2000

[4] 정기원, 윤창섭외, 「소프트웨어 프로세스와 품질」, 홍능과학풀판사, 1997

[5] 한국전산원, "감리결과 분석을 통한 주요 문제점 및 개선사례 연구", 2001

[6] Basili V., "Empirical Software Engineering", *Software Process Newsletter*, No.12, Spring 1998

[7] Crosby, Philips B, *Quality is free, The art of making quality certain*, NALP-ENGUIN, 1980

[8] Cusumano, M. A. *Japan's Software Factories*, Oxford University Press, Oxford, UK, 1991

[9] Curtis, W., "Building a Cost-Benefit Case for Software Process Improvement", *Notes from Tutorial given at the Seventh Software Engineering Process Group Conference*, Boston, MA, May, 1995

[10] Gaffney J., R. Cruickshank, R. Werling, *Software Measurement Guide Book*, International Thomson Computer Press, 1995

[11] Hartwick, J., and Barki, H. "Explaining the Role of User Participation Information Systems Use," *Management Science*, 1994

[12] Herbsleb J., D. Zubrow, J. Siegal, "Software Process Improvement: State of the payoff", *American Programmer*, Vol. 7 no. 9, September,1944

[13] Heny, J., A. Rosmann and J. Snyder, "Quantitative Evaluation of Software Process Improvement", *J. Systems Software*, No. 28, 1995

[14] Humphery, Watts S., *Managing the Software Process*, Addison Wesley, 1989

[15] R. S. Pressman, *Software Engineering*, Mcgraw-Hill, 1997

[16] T. Ravichandran & Arun Rai, *Quality Management in System Development*, 1996

■ 저자소개

김 용 경

고려대학교를 졸업하고, 숭실대학교에서 이학석사 그리고 명지대학교에서 경영학 박사학위를 취득했다. 현재는 건양대학교 경영정보학과 교수이며, 한국정보기술응용학회 부회장으로 활동 중이다. 관심분야는 소프트웨어공학, 소프트웨어 품질관리, 정보시스템 감리 및 감사 등이다.

김 필 중

고려대학교를 졸업하고, 미 죠지아 스테이트대학에서 전산학 석사학위를 받았으며, 충남대학교에서 이학 박사학위를 취득했다. 현재는 건양대학교 IT학부 교수로 재직 중이며, 관심분야는 데이터통신, 컴퓨터그래픽스, DSP, 소프트웨어공학 등이다.