

정보시스템의 개발방법론에 관한 연구

- 시스템 개발 수명주기(System Development Life Cycle)의 단계별 통제를 중심으로 -

정 충 식*

제 1 장 서 론

제 1 절 연구의 목적

오늘날 기업 조직의 모든 부문에 걸쳐서 컴퓨터의 활용이 폭발적으로 증가하고 있다. 이제 컴퓨터의 활용은 작업계층의 일상업무 자동화를 위한 시스템의 구축에서부터 관리자를 위한 정보시스템의 구축을 거쳐 경영층을 위한 전략계획의 수립을 지원하는 전략정보시스템에 이르기까지 광범위한 영역에 사용되고 있다.

이러한 여러 적용부문의 정보시스템 구축을 위해서는 이제까지 프로그램 개발위주의 노동 집약적인 방법에 의존하여 왔으며 따라서 낮은 생산성으로 인하여 업무의 적체현상을 가져왔고 대부분의 정보시스템 개발이 기대에 미치지 못하고 실패하는 경우가 많았다. 또한 정보시스템 개발 프로젝트의 경우, 계획대로 프로젝트가 진행되는 경우가 드물었고 따라서 일정이 지연되거나 품질이 미흡하거나 예산이 초과되는 등 여러 문제점이 되풀이되어 발생하고 있다. 더구

나 제대로 시험되지 않은 정보시스템이 가동되어 시스템의 운영단계에서 심각한 오류가 발견되어 경영전반에 걸쳐 막대한 타격을 입기도 하였다^{1) 김 세현, 1989}. 따라서 정보시스템의 개발에 대한 통제의 필요성이 제기되고 있다^{2) 정 충 식, 1994}.

이제까지 사무자동화는 이러한 정보시스템의 구축에 수반되는 하위 개념으로, 그 일부분을 구성하는 것으로 인식되어져 왔다. 따라서 정보시스템의 개발은 사무자동화의 관점과는 먼 것으로 간주되어 왔다. 그러나 정보시스템은 Client / Server의 관점에서 분산처리를 더욱 지원하는 추세로 발전하고 있으며 사무정보시스템과 자동화기기는 PC 성능의 비약적인 발전 및 LAN, WAN 등의 NETWORK의 구축으로 인하여 그 이용 수준이 고도화되고 있다. 따라서 사무자동화의 관점에서 이젠 단순히 단능적인(stand-alone) 사무자동화 기기의 도입에서 벗어나 점차 정보시스템의 구축에 관심을 기울여야 하며 또한 적극적으로 참여하여야 한다. 이제 정보시스템의 개발도 이전의 운영 중심의 개발에서 사용자 중심의 개발환경으로 바뀌고 있다. 따라서 사무자동화가 사무실의 효율성을 높이기 위하

* 용인공업전문대학 사무자동화과

여 컴퓨터기기를 단순히 사무실에 설치하는 것을 의미하지 않고 통합된 시스템을 활용하여 효율성을 높이도록 하는 것이라면, 정보시스템 개발의 모든 과정이 사무자동화의 관점에서 즉 사용자의 입장에서 재조명되어야 한다.

따라서 본 논문의 목적은 정보시스템 개발의 전 과정을 사용자의 관점에서 파악해 보는 데 있다. 특히 이제까지 정보시스템 개발의 방법론 중에서 가장 널리 사용되어 온 시스템개발 수명주기의 방법론을 각 단계별로 통제의 측면에서 살펴보고자 한다.

제 2 절 연구의 범위 및 방법

본 연구는 정보시스템 개발방법론 가운데서 시스템개발 수명주기의 제 단계를 연구대상으로 한다. 이제까지 정보시스템의 개발과 관련한 연구는 자료처리 수준(EDPS), 경영정보수준(MIS), 의사결정지원(DSS) 및 전략정보시스템의 구축(SIS) 등 정보시스템의 성장 유형에 관한 연구에 치우쳐 왔다. 그러나 이러한 정보시스템 구축의 거의 모든 방법에서 시스템개발 수명주기의 방법이 사용되어져 왔으나 이에 대해서는 모두 시스템의 설계의 관점에서만 분석되어져 왔다.

정보시스템의 개발과 관련하여서는 이제까지 여러 컴퓨터 관련 기업 및 학자에 의해서 방법론이 제기가 되었으나 본 연구는 Nolan의 정보시스템 발전 단계론과 이제까지 국내의 여러 교재에 가장 많이 인용되고 있는 Gordon B. Davis의 이론 등을 정리할 것이다. 통제의 관점에서는 Vallabhaneni의 SDLC의 제 단계를 단계별로 정리해 보고자 한다. 특히 각각의 단계별로 국내에서 발생하였던 실제 사례를 인용하여 검토해 보고자 한다.

연구의 범위와 관련하여서는 대형 정보시스템을 자체개발하는 경우에 국한하여 살펴보고

자 한다. 일반적으로 정보시스템의 구축 시에는 소프트웨어의 자체개발, 위탁개발, 패키지의 구입 및 아웃 소싱 등 여러 방법을 선택할 수 있으며 이러한 각각의 경우에 따라서 개발 과정이 달라질 수 있다^(이 주현, 1991). 시스템개발 수명주기의 방법론은 위의 여러 경우에 모두 사용될 수 있으나 자체개발의 경우를 살펴보면 방법론을 가장 잘 이해할 수 있을 것이므로 본 연구는 대형 정보시스템의 자체개발에 초점을 두고 분석하고자 한다.

제 2 장 이론적 배경 및 제 이론

제 1 절 Nolan의 정보시스템 발전 단계

정보시스템의 개발방법론에 관한 고찰을 위해서는 우선 정보시스템의 발전단계에 대한 이해가 선행되어야 한다. 이제까지 정보시스템의 발전 단계 이론 가운데서 학계로부터 가장 일반적으로 수용되는 Nolan의 발전단계론을 간략히 살펴보고자 한다. Nolan은 70년대 중반 정보시스템의 발전단계를 기초단계(Initiation), 확산단계(Expansion), 공식단계(Formalization), 성숙단계(Maturity)의 4단계로 나누어 각 단계를 컴퓨터 적용업무의 발전, 정보시스템 인력의 전문화, 관리기법 등의 정도에 따라 구분하였다. 이러한 4단계의 발전론은 발표 당시에는 어느 정도 타당성을 갖고 있었으나 정보시스템이 고도화되고 기능이 전문화됨에 따라 타당성을 상실하게 되었다^(Nolan & Gibson, 1974). 이에 따라 Nolan은 이후 공식단계를 통제단계(Control), 종합단계(Integration), 관리단계(Administration) 등으로 세분화하여 정보시스템의 발전을 6단계로 구분하였다. 이러한 정보시스템의 발전 제단계는 적용업무, 정보시스템 조직, 정보시스템의 기획 및 통제 그리고 사

용자의 인지도 등에 따라서 다음과 같이 구분되고 있다^(Nolan, 1979).

1) 기초단계(Initiation)

컴퓨터를 도입하여 급여나 재고관리 등의 몇 가지 단순한 업무를 전산화하는 자료처리 수준의 단계이다.

2) 확산단계(Contagion)

기초단계에서 인식된 컴퓨터의 이점이 급속히 파급되어 다른 업무의 전산화가 급격히 이루어진다. 또한 여러 경영계층에서 시스템에 대한 요구가 증대하나, 정보시스템의 비용만 30 ~ 40% 증가할 뿐 각 시스템간의 통일성이 없어 여러 계층의 요구를 충족할 수 없게 된다.

3) 통제단계(Control)

확산단계에서의 무분별한 전산화 확대로 통제의 필요성이 절실해져서 여러 수단을 마련하나 사용자의 인지도가 낮기 때문에 실효를 거둘 수 없다. 따라서 정보시스템을 조직화하게 되며 종합단계로의 눈에 보이지 않는 전이가 이루어진다.

4) 종합단계(Integration)

정보시스템을 조직화하여 데이터베이스 및 데이터통신 시스템의 구축에 의해 양질의 서비스를 제공하나 확산단계에서와 같이 정보시스템 비용의 급격한 증가를 유발한다.

5) 관리단계(Administration)

5개의 새로운 컴퓨터 관련 기술 즉, 네트워크 자동화, 소프트웨어 개발 자동화, 데이터베이스 자동화, 공정 자동화, 공장 자동화 등이 조직 내에 수용되어 컴퓨터의 활용이 종래와는 다른 양상을 보이게 된다.

6) 성숙단계(Maturity)

조직의 요구와 컴퓨터의 특성이 충분히 조화되어 정보시스템에 조직 내에서의 정보의 흐름이 그대로 반영되고 적용업무에 대한 모든 시스템이 개발되어 있는 상태이다.

이러한 정보시스템 발전의 제 단계에서는 사용자의 인지도가 가장 중요한 역할을 하며 정보시스템의 구축은 통제단계에서 종합단계로의 전환 시기에 이루어진다. 그러나 대부분의 정보시스템 개발이 이러한 단계를 거쳐 성공하는 것이 아니므로 80년대에 들어와 정보시스템의 구축을 위한 개발방법론에 관한 연구가 시작되었다.

제 2 절 Davis의 시스템개발 수명주기론

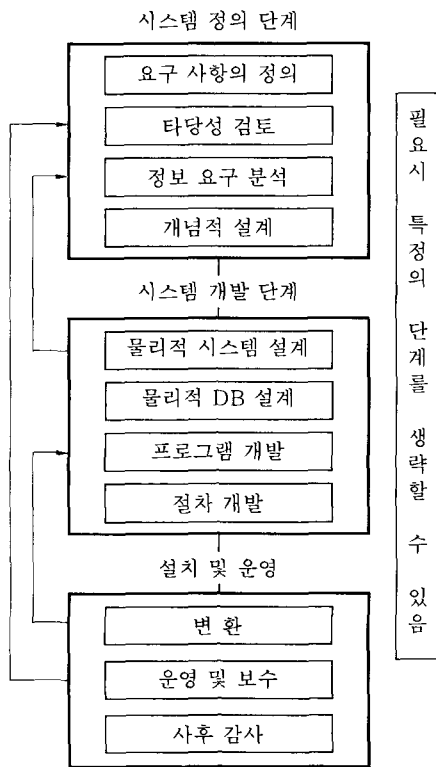
정보시스템의 개발방법은 프로젝트의 크기, 구조화의 정도, 개발요원의 능력, 사용자의 태도 및 조직의 문화 등에 따라서 그 방법을 달리할 수 있다. 전통적으로 정보시스템의 개발방법으로 가장 많이 사용되고 있는 것이 시스템개발 수명주기(System Development Life Cycle)일 것이다. 시스템개발 수명주기란 정보시스템을 개발하는데 필요한 잘 정의된 절차를 의미한다. 따라서 정보시스템의 개발을 성공적으로 추진하기 위해서는 시스템개발 수명주기의 제 단계에 대한 이해가 필수적이다. 이러한 시스템개발 수명주기의 단계에 대해서도 기업이나 학자들에 따라서 모두 다른 구분을 적용하고 있으나 일반적으로 잘 알려진 Gordon B. Davis의 단계론을 중심으로 살펴보면 다음과 같다^(Davis & Olson, 1985).

우선 Davis의 시스템개발 수명주기론은 아래와 같은 단계로 이루어져 있다.

1) 시스템 정의 단계

- ① 요구 사항의 정의
- ② 타당성 검토

- ③ 정보요구 분석
 - ④ 개념적 설계
- 2) 시스템 개발 단계
- ① 물리적 시스템 설계
 - ② 물리적 데이터베이스 설계
 - ③ 프로그램 개발
 - ④ 절차 개발
- 3) 설치 및 운영 단계
- ① 변환
 - ② 운영 및 보수
 - ③ 사후 감사



(그림1) Gordon B. Davis의 시스템개발 수명주기론

(출처) Davis, G.B & Olson, M.H. Management Information Systems, 1985, p. 571.

Davis의 시스템개발 수명주기론은 크게 시스템의 정의, 개발 및 설치·운영의 단계로 구성되어 있다. 정의 단계는 문제에 대한 정의로서 프로젝트가 제안된 후에 타당성 검토가 이루어지고 개념적 설계가 진행된다. 개발의 단계는 요구사항 및 개념적 설계에 기초하여 데이터베이스 및 프로그램에 대한 개발이 진행된다. 설치 및 운영의 단계는 새로운 응용시스템으로의 변환인데 변환은 모든 프로그램과 절차가 준비되고 인수시험 및 교육이 실시되고 이후 새로운 시스템을 가동한다. 이러한 시스템개발 수명주기의 제 단계는 [그림 1]에서 보여지는 것처럼 필요시 각 단계를 생략할 수 있고 또한 단계별로 feedback의 기능을 갖고 있다.

이러한 Davis의 시스템개발 수명주기론은 이전의 이론들과 비교하여 볼 때 feedback의 기능을 도입하였고 또한 사용자 참여의 수단으로 원형화(Prototyping) 접근 방식을 강조하나 기본적으로 시스템 설계의 관점에 입각한 수명주기론임을 부인할 수 없다^{Naumann & Jenkins, 1982}. 특히 시스템의 변환과 관련하여 어떠한 시험(Test)도 단계로 설정해 놓고 있지 않다. 따라서 사용자 중심이 아닌 개발자 즉 시스템 분석자의 관점에서 구성된 단계론이라 할 수 있다.

제 3 절 시스템개발 프로젝트 참여자의 역할

이제 정보시스템의 개발은 프로그래머 개인이나 프로젝트 책임자의 능력에 의해 성공여부가 결정되지 않는다. 왜냐하면 정보시스템의 구축을 위해서는 프로젝트의 팀이 구성되어 방법론에 따라 단계별로 개발업무를 수행하기 때문에, 시스템 개발 수명주기의 제 단계를 이해하기 위해서는 정보시스템 부문의 인력을 포함한 프로젝트 구성원의 역할에 대한 이해가 선행되어야 한다. 특히 우리나라의 경우처럼 정보시스템 관련 인력이 세분화되어 있지 않은 상황에서는 각 단계별로

프로젝트 참여자의 활동에 대하여 이해하기 곤란한 경우가 생길 수도 있을 것이다. 따라서 시스템개발의 각 단계에 참여하는 참여자들의 역할을 간략히 살펴보면 다음과 같다.^[정 중식, 1986]

1) 사용자

새롭게 개발되는 시스템이 여러 조직에서 사용될 경우, 사용자는 각각의 조직을 대표하여 기존의 부서업무를 숙지하고 있는 인물이어야 하며 또한 적절한 권한을 가지고 프로젝트의 전체 단계를 사용자의 관점에서 단계별로 파악하여야 한다.

2) 관리층

관리층은 사용자 부서의 장으로, 사용자가 시스템의 기능에 근거한 요구 사항 및 이의 실현에 점을 둘 때, 이와는 다르게 업무 협조의 관점에서 프로젝트에 참여하며 특히 사용자부문의 책임을 진다. 따라서 사용자의 경우 현재의 시스템에 대하여 이해를 가지고 있어야 하나 관리층의 경우에는 시스템에 대한 깊은 지식은 필요치 않다.

3) 자문위원회

타당성 검토의 단계에 참여하며 여러 가지 대체안 가운데서 전체 조직의 목표와 부합되는 안을 결정하여 프로젝트의 방향을 설정한다.

4) 프로젝트 책임자

프로젝트 진행의 전반을 지휘·감독하며 전체에 대해서 책임을 진다. 따라서 관리기능 및 기술적인 지식을 겸비하고 있어야 하는데 프로젝트의 성공을 위해서는 관리적인 기능이 더욱 요망된다.

5) 시스템 분석가

사용자의 요구 사항 및 업무의 흐름을 파악하여 시스템을 설계한다.

6) 시스템 프로그래머

시스템의 설계 단계에서 시스템 관련 처리의 방법 등을 결정한다.

7) 전산실장

정보시스템의 운영을 지휘·감독하여 컴퓨터 운영에 대한 책임을 진다. 따라서 개발의 거의 모든 과정에 참여하여 개발에 대한 시스템의 지원을 보장한다.

8) 데이터베이스 관리자

데이터베이스 설계 및 통제에 책임을 진다. 따라서 자료의 내용과 형식 및 구조 등을 결정한다.

9) 적용업무 프로그래머

프로그램 명세서에 의거하여 프로그램을 작성하며 단위 시험을 실시한다.

10) 품질보증 인력

프로그래머가 개발하여 단위시험을 실시한 프로그램을 인계받아 종합적인 인수시험을 실시한다. 특히 사용자의 관점에서의 시험을 실시한다.

11) 전산감사인

정보시스템의 개발 과정에서 발생할지도 모르는 오류 및 사고에 대처하기 위하여 사전에 표준통제 처리절차를 수립하고 이에 의거하여 프로젝트의 전과정을 감사하여 안전관리의 역할을 수행한다.

이상 살펴본 인력이 정보시스템 개발의 프로젝트에 참여하여 각 단계별로 업무를 수행하게 된다. 따라서 시스템개발 수명주기의 각 단계를 이해하기 위해서는 이들 참여자들의 역할에 대한 이해가 선행되어야 한다.

제 3 장 시스템개발 수명주기의 각 단계별 통제사례

제 1 절 Vallabhaneni의 시스템개발 수명주기

우선 Vallabhaneni의 시스템 개발 수명 주기론은 아래와 같이 14단계로 이루어져 있다.^[Vallabhaneni, 1988]

● 시스템개발 수명주기(System Development Life Cycle : SDLC)의 제 단계

- 1 단계 : 사용자 요구
(User Service Request)
- 2 단계 : 타당성 검토
(Feasibility Study)
- 3 단계 : 상세 요건 정의 (Detail Requirement Definition)
- 4 단계 : 일반 시스템 설계 (General Systems Design)
- 5 단계 : 상세 시스템 설계
(Detail Systems Design)
- 6 단계 : 프로그램 개발
(Programs Development)
- 7 단계 : 단위 시험 (Unit Testing)
- 8 단계 : 시스템 시험 (System Testing)
- 9 단계 : 매뉴얼 개발
(Manual Development)
- 10 단계 : 교육 (Training)
- 11 단계 : 인수 시험
(Acceptance Testing)
- 12 단계 : 변환 (Conversion)
- 13 단계 : 실제 운영 및 지원

(Production Operations and Support)

14 단계 : 실행후 검토
(Postimplementation Review)

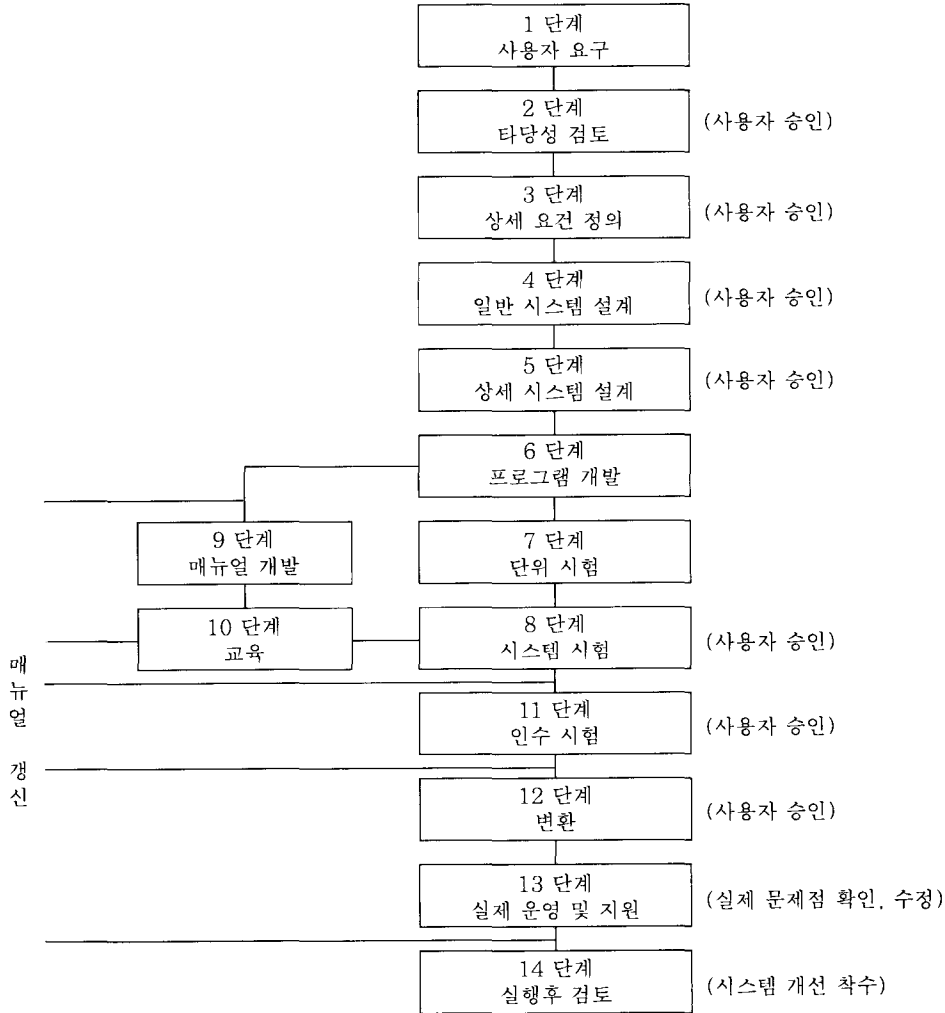
Vallabhaneni의 방법론이 앞에서 살펴본 여러 방법론과의 차이는 우선 시험단계에 대한 고려에서 찾아볼 수 있다. 1단계의 사용자 요구 단계부터 6단계의 프로그램의 개발까지는 다른 여러 방법론과 커다란 차이가 없다. 그러나 시험의 경우는 7단계의 단위시험, 8단계의 시스템시험 및 11단계의 인수시험 등 3차례의 시험 단계를 제시하고 있다. 특히 이러한 3차례의 시험의 단계는 각각의 단계별로 특별한 목적을 가지고 있으며 전체의 단계에서 중요한 역할을 수행하고 있다. Vallabhaneni는 이러한 SDLC의 여러 단계가 꼭 순차적으로 수행되어야 할 필요는 없다고 강조하며 [그림 2]에서 보여지는 것처럼 9단계의 매뉴얼 개발 및 10단계의 교육은 병행적으로 이루어질 수 있다고 설명한다.

제 2 절 각 단계별 통제사례

1. 1 단계 : 사용자 요구 단계

SDLC의 첫번째 단계인 사용자 요구 단계는 SDLC의 전 과정 가운데서 가장 중요한 부분이다. 이 단계에서 문제를 인식하고 개선점을 제시하며 사용자가 기대하는 시스템의 성능이 대략의 프로젝트 비용과 함께 제시되어 진다. 그러나 이제까지의 정보시스템의 개발이 개발자 위주의 기술적인 측면에 치우쳐왔기 때문에 이 부분의 중요성이 간과되어져 왔다. 여러 통계자료에 의하면 이 첫번째 단계가 정보시스템의 성패를 좌우하는 것을 알 수 있다. 예를 들어, 소프트웨어 생산품의 전형적인 오류를 대별해

(시스템개발 수명주기의 상호연관성)



[그림 2] Vallabhaneni의 시스템개발 수명주기론

출처) Vallabhaneni, S. Rao, Information Systems Audit Process, 1988, p.83.

보면, 크게 요구 명세와 설계(64%) 및 구현(36%)으로 나눌 수 있다.^[한국정보시스템관리학회, 1992] 또한 현재 정보시스템의 유지보수에 전체 인력의 70 ~ 80%가 투입되어 정보시스템 부문의 인력운용에 큰 부담으로 작용하고 있다. 그런데 이러한 유지보수의 발생원인으로는 잘못된 요구사항이 56%, 잘못된 설계가 27%, 잘못된

코딩이 7% 그리고 기타 부문이 10%를 차지하고 있다. 따라서 요구사항의 단계부분이 가장 중요함을 알 수 있다. 이 단계는 사용자에게 의해서 모든 활동이 이루어지며 감사인의 참여가 필수적이지는 않다.

사용자 요구의 단계에서의 목적, 활동 및 산출물을 표로 요약해 보면 다음과 같다.

[표 1] 1 단계 : 사용자 요구 (User Service Request)

| 목 적 | 활 동 | 산 출 물 |
|---|---|--|
| 정보시스템 부문에 아래의 정보를 제공 <ul style="list-style-type: none"> • 해결되어야 할 문제의 정의 및 영향 • 사용자가 기대하는 시스템의 성능 * 참여자 <ul style="list-style-type: none"> - 사용자 - 관리층 - 감사인(선택) | <ul style="list-style-type: none"> • 문제의 인식 또는 효율적이고 효과적인 개선의 필요성 • 사용자 부서의 업무 분석 수행 • 문제에 의해서 영향을 받는 업무 및 부서를 명확히 하고 기대되는 해결책을 분석 • 대략의 비용, 효과 및 프로젝트 완료일 제시 | <ul style="list-style-type: none"> • 사용자에게 의해 작성된 업무 관련 문제 보고서 • 사용자의 요구사항에 포함되어야 할 사항들: <ul style="list-style-type: none"> - 수행되어야 할 목적에 대한 정의 - 요청 이유에 대한 설명 - 본 요청에 의해서 영향을 받는 부서 및 시스템의 명세 - 예상되는 비용, 효과 프로젝트 완료일 |

2. 2 단계 : 타당성 검토 단계

이 단계에서는 프로젝트 책임자의 주관 하에 프로젝트의 타당성을 검토하기 위하여 경제적, 운영적 및 기술적인 관점에서의 검토가 이루어진다. 초대형의 프로젝트일 경우에는 자문위원회(Steering Committee)가 구성되어 종합적인 타당성 검토를 수행한다. 이 단계에서는 시스템에 대한 사용자의 요구 사항을 인식한 후에 현재의 시스템과 제안되어지는 시스템의 성능을 기술한다. 이후 여러 대안 및 최선의 해결책을 제약 사항의 관점에서 도출해 낸다. 물론 최고경영자의 지시에 의한 프로젝트의 경우 이러한 타당성 검토가 생략되는 경우도 일어날 수 있겠으나 그러한 경우에도 비용/편익 분석 등만이 생략될 뿐 기타 활동은 수행된다^[한국생산성본부, 1991]. 특히 이 단계에서 각각의 단계에서의 프로젝트의 범위 및 시스템 이행 계획의 초안이 작성된다. 또한 이 단계에서 가장 중요한 활동은 프로젝트의 팀을 구성하는 것과 프로젝트의 책임자를 선임하는 것이다. 새롭게 개발되어질 시스템이 여러 조직 단위나 또는 한 조직 단위의 여러 계층에서 사

용될 것이라면 프로젝트 팀은 각 조직 단위와 각 수준의 대표자를 필요로 한다. 이 경우에 기존의 사용자 부서에서는 현재의 업무를 숙지하고 있는 유능한 직원보다는 기존 부서의 업무에 지장을 초래하지 않는 범위 내에서 초심자를 위주로 하여 프로젝트 팀의 구성원으로 보내려고 하는 경향이 있으므로 이 경우에도 프로젝트 책임자의 역할이 중요하다. 대표자들은 필요한 업무 지식은 물론 조직 내에서 적절한 권한을 반드시 가지고 있어야 한다. 따라서 프로젝트의 팀을 구성하는데는 경영층의 지원이 절대적인 요소가 된다^[안 중호, 1994]. 또한 대부분의 프로젝트 책임자가 시스템 분석가 등 기술적인 부문을 관장하는 사람이 선임되는 경우가 많다. 그러나 프로젝트라는 것이 기술적인 측면 보다는 구성원 간에 의사소통이나 다른 사용자 부서와의 이해관계를 조정하는 측면이 더 많으므로 사용자 부문의 장이 선임되는 것이 훨씬 효율적이다^[Ives & Olson, 1982].

타당성 검토 단계에서의 목적, 활동 및 산출물을 표로 요약해 보면 다음과 같다.

[표 2] 2 단계 : 타당성 검토 (Feasibility Study)

| 목적 | 활동 | 산출물 |
|---|---|---|
| 프로젝트의 타당성을 검토하기 위하여 경제적이고, 운영적이고, 또한 기술적인 관점에서의 정보를 제공한다 * 참여자 - 사용자 - 관리자 - 시스템분석가 - 프로젝트 책임자 - 전산실장 - 자문위원회 - 전산감사인 | <ul style="list-style-type: none"> • 사용자의 근본적인 시스템에 대한 요구사항을 인식 • 현재의 시스템에 대한 이해 • 사용자의 요구 사항과 문제를 개략적으로 시스템 설계 • 제안되어지는 시스템 및 그 성능을 기술 • 대안적인 수단으로의 또 다른 시스템의 해결방안 모색 • 각각의 대안에 대한 비용/편익 분석 실시 • 각각의 대안에 대한 운영 및 기술적인 제약 고려 • 최선의 해결책을 결정 • 제안되어지는 시스템에 대한 근거 작성 • 각각의 단계에서의 산출물 결정 • 시스템 이행계획의 초안 작성 • 프로젝트의 작업계획 및 일정 작성 | <ul style="list-style-type: none"> • 타당성 검토 보고서 및 사용자의 요구사항에 포함되어야 할 사항들: <ul style="list-style-type: none"> - 사용자 요청사항의 명확화 - 문제의 특성에 대한 상세한 설명 - 대안에 대한 토의 및 검토 (수작업 & 자동화) - 각각의 대안에 대한 운영 및 기술적인 제약에 대한 토의 - 각각의 단계에서의 프로젝트의 범위, 목적, 활동 및 산출물 그리고 일정을 산출 - 시스템 이행계획 초안 - 사용자의 확인 |

3. 3 단계 : 상세 요건 정의 단계

이 단계는 정보시스템 부문의 주도하에 이루어지나 필요한 모든 자료는 사용자에게 의해 제공되어야 한다. 일반적으로 사용자들은 시스템에 대한 무지 또는 거부반응으로 인하여 사용자 요구 단계의 이후에는 프로젝트에 잘 관여하지 않으려고 하는 경향이 있다. 그러나 이 단계는 기술적인 사항보다는 사용자의 관점에서 시스템을 정의하는 단계라고 볼 수 있다. 따라서 기존 시스템 및 새로운 시스템의 입·출력에 대한 사용자의 설명 및 필요사항의 정의가 필수적이다. 소프트웨어는 그 특성상 작은 요구라고 여겨지는 것이라도 막대한 시간을 투자하여 개발 또는 변경을 수행해야 할 경우가 생길 수 있다. 그러

나 사용자들은 소프트웨어란 글자 그대로 소프트웨어하기 때문에 개발 도중에 사용자의 요구사항에 따라서 쉽게 수정될 수 있다고 생각하는 경향이 있다. 그러므로 이 단계에서 상세하게 요건이 정의가 되어야 한다. 특히 통제에 관해서는 새로운 시스템의 도입에 따르는 다양한 통제의 방법(보안 특성, 감사 증적, 파일 백업 및 오류 수정 방법 등)이 이 단계에서 정의가 되어야 한다.^[감사원, 1990]

상세 요건 정의 단계에서의 목적, 활동 및 산출물을 표로 요약해 보면 다음과 같다.

(표 3) 3 단계 : 상세 요건 정의 (Detail Requirement Definition)

| 목 적 | 활 동 | 산 출 물 |
|---|--|--|
| <p>시스템에 대한 요구 사항과 운영여건을 정확하게 분석하기 위하여, 사용자와 정보 시스템부문의 사이에 제안되는 시스템에 대한 상세한 요구사항이 필요하다.</p> <p>* 참여자</p> <ul style="list-style-type: none"> - 사용자 - 관리층 - 시스템분석가 - 프로젝트 책임자 - 전산실장 - 데이터베이스 관리자 - 전산감사인 | <ul style="list-style-type: none"> • 기존 시스템에 대한 장점 및 약점의 검토 • 사용자 및 운영부문의 직원을 면담하여 기존 시스템에 대한 자료를 수집 • 사용자에게 의해서 요청되어지는 시스템으로 수행할 수 있는 업무 및 시스템 기능 확인 • 개개의 업무를 지원하는 입력 및 출력 활동 정의 • 자료의 요구 범위 및 그 출처 확인 • 새로운 시스템의 도입에 따르는 기기, 소프트웨어, 보안 및 통제의 요구사항 확인 • 시스템 이행계획의 초안 갱신 • 프로젝트의 작업계획 및 일정 갱신 | <ul style="list-style-type: none"> • 정보시스템 부문에서 작성되는 보고서에 포함되어야 할 사항들: <ul style="list-style-type: none"> - 새로이 제안되는 시스템에 의해서 수행되어질 업무정의 - 기존 시스템의 입력, 출력에 대한 설명 - 새로운 시스템의 기대 효과 요약 - 데이터의 요소 및 구조에 대한 설명 - 새로운 시스템의 도입에 수반되는 장비 및 소프트웨어 - 새로운 시스템의 보안 및 통제 요소 - 시스템 이행계획 초안의 갱신 - 프로젝트 작업계획, 일정의 갱신 - 사용자의 확인 |

4. 4 단계 : 일반 시스템 설계 단계

이 단계에서는 업무의 수행을 위해 시스템의 기능을 모듈 또는 하위시스템의 활동으로 파악하여 다른 기존의 시스템과의 연결을 나타내는 흐름도를 작성한다. 특히 개략적이거나 입력 양식, 화면, 보고서 및 파일의 배열에 대한 초안

이 작성되어지며 이러한 산출물에 대해서는 사용자의 확인 절차를 거친다. 통제의 관점에서는 백업, 장애복구 및 접근통제에 대한 절차가 수립이 되어야 한다. 또한 데이터베이스의 특성도 고려가 되어야 한다^(Menkus & Ruthberg, 1990).

일반 시스템 설계 단계에서의 목적, 활동 및 산출물을 표로 요약해 보면 다음과 같다.

(표 4) 4 단계 : 일반 시스템 설계 (General Systems Design)

| 목 적 | 활 동 | 산 출 물 |
|--|--|--|
| <p>정보시스템의 개발 및 설계의 표준에 의거한 일반적인 설계 명세서의 작성</p> <p>* 참여자</p> <ul style="list-style-type: none"> - 사용자 - 관리층 | <ul style="list-style-type: none"> • 사용자에게 의해서 요청되어진, 새로운 시스템에서 수행되어질 시스템의 기능 확정 • 개개의 업무를 지원하는 입·출력 활동의 확정 • 업무의 수행을 위해 시스템의 기능을 모듈 또는 하위시스템의 활동으로 파악 • 처리의 방법(일괄처리, 즉시처리, 분산처리 등)에 대한 검토 | <ul style="list-style-type: none"> • 보고서 및 문서에 포함되어야 할 사항들: <ul style="list-style-type: none"> - 다른 기존의 시스템과의 연결을 나타내는 일반 시스템 흐름도 - 입력양식 및 출력 보고서의 초안 - 컴퓨터의 연산기능 및 계산 논리 |

| | | |
|---|--|---|
| <ul style="list-style-type: none"> - 시스템분석가 - 프로젝트 책임자 - 전산실장 - 데이터베이스 관리자 & 직원 - 시스템 프로그래머 - 전산감사인 | <ul style="list-style-type: none"> • 처리주기 및 평균 처리량의 확정 • 대략의 시스템 흐름도 작성 • 입력 양식, 화면, 보고서 및 파일의 배열에 대한 초안 작성 • 일반설계의 변경에 대한 통제절차 수립 • 백업, 장애복구 및 접근통제에 대한 절차 수립 • 시스템 이행계획 갱신 • 프로젝트의 작업계획 및 일정 갱신 | <ul style="list-style-type: none"> - 자료의 편집 및 승인 기준 - 오류코드, 메시지 및 수정 활동 - 데이터파일의 특성 - DB의 특성 - 백업 및 장애복구 절차 - 시스템 시험 및 인수시험의 기준 - 데이터파일 변환의 기준 - 시스템 이행계획 갱신 - 작업계획 및 일정 갱신 - 사용자의 확인 |
|---|--|---|

5. 5 단계 : 상세 시스템 설계 단계

이 단계에서는 이전 단계에서 확정된 개개의 모듈이나 하위시스템을 위해 프로그램의 흐름도 및 설명서가 작성되어 진다. 또한 이에 기초하여 상세한 프로그램의 사양서가 작성되어야 한다. 그런데 여기에서 중요한 점은 작성되는 여러 문서들이 설계표준에 의거하여야 한다는 것이다.

시스템의 이행 후에 유지보수가 행해지거나 개발자의 이직으로 인해 다른 사람이 프로그램을 수정할 경우에는 이러한 문서에 의존해야 하는데, 설계표준에 기초하지 않을 경우에는 유지보수가 어려움에 처하게 된다. 이 단계부터 단위 시험을 위한 준비 및 파일의 변환을 위한 준비가 시작된다. 상세 시스템 설계 단계에서의 목적, 활동 및 산출물을 표로 요약해 보면 다음과 같다.

[표 5] 5 단계 : 상세 시스템 설계 (Detail Systems Design)

| 목 적 | 활 동 | 산 출 물 |
|---|---|--|
| <p>조직 및 정보 시스템의 설계표준, 명세서에 의거한 상세한 시스템 설계와 프로그램 명세의 작성</p> <p>* 참여자</p> <ul style="list-style-type: none"> - 사용자 - 관리층 - 시스템분석가 - 프로젝트 책임자 - 전산실장 - 데이터베이스 관리자 & 직원 - 적용업무 프로그래머 - 시스템 프로그래머 - 전산감사인 | <ul style="list-style-type: none"> • 일반 시스템의 설계 단계에서 확정된 개개의 모듈이나 하위시스템을 위한 프로그램의 구상 (사용자와 관리층의 검토후에 변경 가능) • 프로그램 흐름도 작성 • 입·출력 활동을 나타내는 프로그램 설명서 작성 • 프로그램 흐름도와 설명서를 고려한 상세한 프로그램의 사양서 작성 • 입력양식, 화면, 보고서 및 파일의 배열 확정 • 프로그램의 단위 시험에 필요한 데이터 준비 • 입력양식, 화면, 보고서 및 파일의 배열 확정 • 프로그램의 단위 시험에 필요한 데이터 준비 • 상세설계의 변경에 대한 통제절차 수립 • 일반 시스템 설계 보고서 갱신 • 프로젝트의 작업계획 및 일정 갱신 | <ul style="list-style-type: none"> • 이 단계의 산출물은 일반 시스템의 설계단계와 거의 동일하나 좀더 상세하게 아래의 사항에 포함되어야 함: - 프로그램 흐름도 - 프로그램 설명서 - 프로그램 사양서 - 단위 시험 기준 - 교육계획 초안 - 시스템시험 및 인수시험 기준 갱신 - 데이터 파일 변환의 기준 갱신 - 시스템 이행계획 갱신 - 작업계획 및 일정 갱신 - 사용자의 확인 |

6. 6 단계 : 프로그램 개발 단계

이 단계는 SDLC의 14단계 중에서 사용자의 확인을 거치지 않는 2단계 가운데 하나이다. 즉 프로그램 개발의 단계는 프로그래머의 고유 권한에 속한다. 하지만 이전의 단계에서 정의된 상세 프로그램 명세서 및 정보시스템의 개발 표준에 근거해서 프로그램의 개발이 이루어지게 된다. 특히 요즘은 프로그램 사양서의 누락과 프로그래밍 코딩의 오류를 조기에 발견하기 위하여 구조화된 프로그래밍의 방법이 널리 쓰이고 있다. 또한 이 단계에서는 다음 단계인 단위 시험의 준비를 위한 자료들이 산출되어야 한다. 전산감사인의 입장에서는 프로그램의 오류를 방지하기 위한 대책이 수립되어 있는지의 여부 및 매뉴얼의 준비 상태를 확인하여야 한다.^[정 홍석, 1988]

프로그램 개발 단계에서의 목적, 활동 및 산출물을 표로 요약해 보면 다음과 같다.

7. 7 단계 : 단위 시험 단계

이 단계는 앞서의 프로그램 개발 단계와 마찬가지로 사용자의 확인이 필요하지 않은 단계이다. 따라서 전적으로 개발자인 프로그래머에 의해서 수행될 수 있는 단계이다. 단위 시험의 단계에서는 개별 프로그램 또는 모듈이 오류를 포함하고 있는지의 여부를 개발자 자신이 시험해 보는 것이다. 따라서 개발자는 사용자에 비하여 업무에 대한 지식이 미흡한 관계로 사용자 중심이 아니라 개발자의 관점에서 기술적인 측면이 시험의 대상이 된다. 우리나라의 경우 아직까지 사용자의 정보시스템 개발에 대한 참여도가 높

(표 6) 6 단계 : 프로그램 개발 (Programs Development)

| 목 적 | 활 동 | 산 출 물 |
|---|--|--|
| 상세 프로그램 명세서 및 정보 시스템의 개발 표준에 의거한 프로그램의 개발 * 참여자 - 사용자 - 관리층 - 적용업무 프로그래머 - 시스템 프로그래머 - 프로젝트 책임자 - 전산실장 - 데이터베이스 관리자 & 직원 - 전산감사인 | <ul style="list-style-type: none"> • 프로그램 흐름도, 설명서, 사양서, 입력 양식, 화면, 보고서 및 자료의 배열 확인 • 프로그램의 개발 • 프로그램 에러의 방지를 위한 고려 • 프로그램 사양서의 누락과 프로그램 코딩의 오류를 조기에 발견하기 위한 구조화 프로그래밍 도입 • 단위시험 계획의 준비(시험 사례, 시험 자료, 기대되는 시험 결과) • 사용자 및 운영 매뉴얼 작성 준비 • 상세 시스템 설계보고서 및 문서 갱신 • 시스템 이행계획 갱신 • 시스템 시험 및 인수 시험의 기준 갱신 • 데이터 파일의 변환 기준 갱신 • 교육계획의 갱신 • 프로젝트의 작업계획 및 일정 갱신 | <ul style="list-style-type: none"> • 명확하게 문서화된 프로그램 • 단위시험계획, 시험사례, 시험자료, 기대되는 시험의 결과 • 갱신된 상세 시스템 설계 보고서 • 갱신된 시스템 이행 계획서 • 갱신된 교육 계획서 • 갱신된 작업계획 및 일정 • 사용자의 확인(선택) |

지 않은 관계로 이러한 단위 시험만으로 시스템의 전체 테스트를 종료하는 경향이 있다. 그러나 이러한 경우에는 시스템의 운영시 항상 사고의 위험성을 내재하고 있다고 보아야 한다. 단위 시험만을 실시하고 시스템을 가동하여 발생한 사고의 사례를 살펴보면 다음과 같다.^[한국금융연구원, 1990]

모 시중은행의 경우 1986년 7월경에 자유저축예금의 프로그램을 새로이 개발하여 자체적으로 테스트를 한 후에 이상을 발견하지 못하고 86년 11월 10일부터 신규 적용하였다. 그러나 이 프로그램은 해약이자 계산용 루틴(Routine)에 예입일로 부터 해약 전일까지의 이자를 계산한 후 동 금액에서 이미 원금에 가산된 매분기말의 결산이자를 빼고 지급되도록 되어야 함에도 불구하고 이미 계산된 매분기말의 이자를 차감하지 않고 그대로 지급하도록 되어 있었다. 이 잘

못된 프로그램은 그대로 쓰여지다가 수유동지점에서 해약이자 지급과정 중에 지급이자가 과다하게 발생한 것을 이상하게 여긴 직원이 전산 부서에 문의하는 과정에서 발견되었다. 따라서 1986년 11월 10일부터 26일 사이에 124개 점포에서 해약된 자유저축예금에 대해서 4천5백여만원이 과다 지급되었다.

이 사례의 경우는 정보시스템 부문의 개발자인 프로그래머가 단위 시험만을 거친 후에 바로 프로그램을 적용한 것으로 확인되었다. 따라서 단위 시험은 전체의 시험 가운데 일부분을 차지할 뿐 시스템의 안정성을 보장해 줄 수 있는 단계는 아니라고 할 수 있다.

단위 시험 단계에서의 목적, 활동 및 산출물을 표로 요약해 보면 다음과 같다.

[표 7] 7 단계 : 단위 시험 (Unit Testing)

| 목 적 | 활 동 | 산 출 물 |
|--|---|--|
| 개별 프로그램 또는 모듈이 오류를 포함하고 있는지의 여부를 정보 시스템의 시험 표준에 따라 시험한다. * 참여자 - 사용자 - 관리층 - 적용업무 프로그래머 - 시스템 프로그래머 - 프로젝트 책임자 - 전산실장 - 데이터베이스 직원 - 전산감사인 | <ul style="list-style-type: none"> • 단위 시험계획, 시험 사례, 시험 자료 및 예상되는 시험 결과를 확정 • 실제의 단위시험을 프로그램 또는 모듈별로 실시 • 예상결과와 실제결과의 비교 • 예상결과와 실제결과 사이의 불일치에 대해서 확인 및 문서화 • 시험 자료, 프로그램 논리의 변경 등을 통하여 시험 불일치의 수정 • 새로운 시스템 구축을 위한 기기의 발주 • 사용자 및 운영 매뉴얼의 작성 계속 • 시스템 이행계획 갱신 • 시스템 시험 및 인수 시험의 기준 갱신 • 데이터 파일의 변환 기준 갱신 • 교육계획의 갱신 • 프로젝트의 작업계획 및 일정 갱신 | <ul style="list-style-type: none"> • 최종 단위시험 계획서 시험 사례, 시험 자료, 예상되는 결과 • 예상되었던 시험 결과와 실제 결과 사이의 불일치에 대한 문서화 • 갱신된 시스템 실행 계획 • 갱신된 시스템 시험 및 인수 시험 기준 • 갱신된 데이터 화일 변환의 기준 • 갱신된 교육훈련 계획 • 갱신된 작업계획 • 사용자의 확인(선택) |

8. 8 단계 : 시스템 시험 단계

이 단계는 시험의 환경 측면에서 이전의 단위 시험 단계와 구분된다. 즉 단위 시험이 개발자인 프로그래머의 독립된 라이브러리에서 개별 프로그램을 시험했던 것에 비해 시스템의 시험은 시험의 환경이 실제 가동되고 있는 시스템의 상황과 동일한 조건하에서 실시된다. 따라서 단위 프로그램이 아닌 전체 프로그램을 그 시험 대상으로 한다. 그러므로 단위 시험의 단계에서 실행하기 힘든 통합된 시험, 즉 월말이나 연말의 데이터를 가동해 보는 시험 또는 다른 적용 업무 프로그램과의 연관성 시험 등을 행할 수

있다. 이 시험 역시도 사용자의 관점이 아닌 개발자의 관점에서 기술적인 측면과 시스템적인 측면에 초점이 두어 진다. 전산 감사인의 경우는 프로그램의 논리 변경을 통한 시험 결과의 불일치를 해소하는 절차가 표준의 범위 내에서 진행되는지 확인하여야 한다. 특히 시스템의 시험과정에서 프로그래머가 테스트 카드를 변조하여 이후에 그것을 사용한 사례도 보고되어 있으므로 시험 기간에도 시스템의 안전관리에 관심을 가지고 있어야 한다^[제 문규, 1990].

시스템 시험 단계에서의 목적, 활동 및 산출물을 표로 요약해 보면 다음과 같다.

[표 8] 8 단계 : 시스템 시험 (System Testing)

| 목 적 | 활 동 | 산 출 물 |
|--|--|---|
| 일반적으로 인정되는 자료처리의 표준에 따라서 시스템의 성능이 계획대로 수행되는지를 확인하기 위하여 전체 프로그램을 시험함. * 참여자 - 사용자 - 관리층 - 시스템분석가 - 적용업무 프로그래머 - 시스템 프로그래머 - 프로젝트 책임자 - 전산실장 - 데이터베이스 직원 - 전산감사인 | <ul style="list-style-type: none"> • 시스템 시험 계획, 시험 사례, 시험 자료 및 예상되는 시험 결과를 확정 • 전체 시스템을 연결하여 실제로 모든 프로그램의 시험을 실시함. • 예상결과와 실제결과와의 비교 • 예상결과와 실제결과 사이의 불일치에 대해서 확인 및 문서화 • 시험 자료, 프로그램 논리의 변경 등을 통하여 시험 불일치의 수정 • 사용자 및 운영 매뉴얼의 갱신 • 시스템 이행계획 갱신 • 인수 시험의 기준 갱신 • 데이터 파일의 변환 기준 갱신 • 교육계획의 갱신 • 프로젝트의 작업계획 및 일정 갱신 | <ul style="list-style-type: none"> • 시스템 시험 계획, 시험 사례, 시험 자료, 예상되는 결과의 확정 • 예상되었던 시험 결과와 실제 결과 사이의 불일치에 대한 문서화 • 갱신된 시스템 실행 계획 • 갱신된 인수시험 기준 • 갱신된 데이터파일 변환의 기준 • 갱신된 교육훈련 계획 • 갱신된 사용자 및 운영 매뉴얼 • 갱신된 프로젝트 작업계획 • 사용자의 확인 |

9. 9 단계 : 매뉴얼 개발 단계

이 단계는 앞에서 언급했던 바와 같이 순차적으로 진행되어질 필요는 없는 단계이다. 즉 프로그램의 개발, 단위 시험 및 시스템 시험 단계의 모든 과정에 걸쳐서 준비되어야 하는 단계이다. 흔히 실제 업무 현장의 경험에 의하면 이러한 매뉴얼 개발의 단계는 가장 마지막의 단계로 잘못 인식되어 있다. 즉 개발자가 임의대로 시스템을 개발하여 사용자에게 인도한 후에 사용자가 실제로 사용하는데 불편을 겪을 경우에 개발자가 그때 가서 사용자 매뉴얼을 작성하여 건네주는 경우가 흔하다. 사용자가 매뉴얼을 건네 받고 시스템의 문제점을 지적하면 또 다시

시스템을 변경하는 경우까지 발생하기도 한다. 따라서 이러한 상황에서는 사용자의 인수 시험이 제대로 수행될 수 없다. 그러므로 매뉴얼의 개발은 어느 단계에서 이루어지든 간에 사용자의 인수 시험의 단계 이전에 준비되어야 하며 각각의 시험 단계의 결과가 반영되어야 하고 인수 시험의 결과에 따라 이후의 단계에서도 수정되어야 한다.

전산감사인은 이 단계에서 시스템의 시험 기간 동안에 요청된 변경 사항들이 매뉴얼에 제대로 반영이 되었는지의 여부 및 위기관리의 절차가 포함되었는지를 확인하여야 한다.

매뉴얼 개발 단계에서의 목적, 활동 및 산출물을 표로 요약해 보면 다음과 같다.

(표 9) 9 단계 : 매뉴얼 개발 (Manual Development)

| 목 적 | 활 동 | 산 출 물 |
|---|---|---|
| <p>완벽하고 정확하게 사용자 및 운영 매뉴얼이 개발되었는지 확인한다. 매뉴얼은 관리층, 사용자, 정보시스템의 운영 인력, 품질 보증 인력 및 전산 감사인의 확인을 받아야 한다.</p> <p>* 참여자</p> <ul style="list-style-type: none"> - 사용자 - 관리층 - 적용업무 프로그래머 - 시스템 프로그래머 - 프로젝트 책임자 - 전산부 직원 - 전산실장 - 전산감사인 | <ul style="list-style-type: none"> • 사용자 매뉴얼이 시스템의 시험 기간 동안에 요청된 변경사항을 반영하고 있는지 확인하고 갱신함 • 운영 매뉴얼이 시스템의 시험 기간 동안에 요청된 변경사항을 반영하고 있는지 확인하고 갱신함 • 교육훈련의 자료 및 일정의 개발 • 인수시험 일자, 기기 및 소프트웨어의 도입 일자, 데이터파일 변환 일자, 새로운 시스템의 가동일자 등을 포함하는 시스템 이행 계획의 갱신 • 인수시험 계획, 시험 사례, 시험 자료 및 기대되는 시험 결과의 개발 • 데이터파일 변환의 기준 갱신 • 프로젝트의 작업계획 및 일정 갱신 | <ul style="list-style-type: none"> • 사용자의 매뉴얼에 포함되어야 할 사항: <ul style="list-style-type: none"> - 시스템의 목적 기술 - 사용자 및 정보시스템 부문의 책임 - 입력 양식의 준비절차 출력 보고서 - 입·출력 대사 절차 • 운영 매뉴얼에 포함되어야 할 사항: <ul style="list-style-type: none"> - 입력형태, 파일, 출력보고서, 작업의 빈도 및 순서를 나타내는 컴퓨터 운영 흐름도 - 작업 통제 언어 - 백업 및 복구절차 - 단말기 조회 및 자료 입력 절차 • 갱신된 이행계획서 • 인수시험 계획, 시험 사례 및 자료 등 • 교육 자료 및 일정 |

10. 10 단계 : 교육 단계

이 단계 역시도 앞의 매뉴얼의 개발과 마찬가지로 병행되어 이루어질 수 있으며 역시 사용자의 인수 시험의 이전 단계에 준비가 완료되어야 한다. 사용자는 이 단계에서 시스템의 인수 시험 이전에 시스템의 사용 방법에 대하여 철저하게 교육을 받아야 한다. 따라서 이 교육의 결과로 인수 시험에서 수행할 시험의 데이터 및 사례를 작성할 수 있는 방법을 도출해낼 수 있으며 시스템의 사용자 사전 시험의 기능을 수행할 수도 있다. 대규모 프로젝트의 경우, 이러한 교육은 사용자의 대표를 중심으로 하여 이루어지며 이후 교육을 받은 사용자가 전달 교육을 실시할 때 지원을 하여야 한다. 예를 들어 지점

이 50개인 증권회사의 시스템 전환의 경우, 각 지점으로부터 1명씩 50명의 사용자 대표를 대상으로 하여 대표의 교육을 실시한 후에 이들에 의한 각 지점의 전달 교육 시에 정보시스템 부문에서 지원을 하여야 한다. 특히 대규모 프로젝트의 경우 사용자들의 휴가 시기 또는 다른 업무와의 중복 등을 피하는 것도 실제적으로는 중요한 문제가 된다. 따라서 이 단계의 경우도 경영층의 관심 및 지원이 중요한 역할을 하게 된다. 전산감사인의 경우는 사용자 대표의 교육에서 얻어진 피드백이 교육 매뉴얼에 반영되는지를 확인하여야 한다.

교육 단계에서의 목적, 활동 및 산출물을 표로 요약해 보면 다음과 같다.

[표 10] 10 단계 : 교육 (Training)

| 목 적 | 활 동 | 산 출 물 |
|---|---|---|
| 사용자가 시스템의 사용방법에 대해서 철저하게 교육 받았으며 또한 교육이 표준의 절차에 따라 적절하게 이루어졌는지 확인 * 참여자 - 사용자 - 관리층 - 시스템분석자 - 프로젝트 책임자 - 전산실장 - 전산감사인 | <ul style="list-style-type: none"> • 교육 매뉴얼이 시스템의 시험이나 매뉴얼 개발의 단계에서 요청된 변화를 반영하고 있는지 검토 • 사용자 대표 교육의 실시 • 사용자 대표 교육중에 얻어진 피드백을 교육 매뉴얼에 반영 • 사용자 대표의 전달 교육을 지원 • 시스템 이행 계획의 확정 • 인수 시험 계획, 시험 사례, 시험 자료 및 기대되는 결과의 확정 • 변환되어질 파일의 이름 및 유형을 확인하고 변환을 검증할 방법을 확인함 • 신·구 시스템의 병행처리의 기준을 작성 • 프로젝트의 작업 계획 및 일정 갱신 | <ul style="list-style-type: none"> • 교육 매뉴얼에 포함되어야 할 사항: <ul style="list-style-type: none"> - 교육 계획 - 교육 일시 및 장소 - 교육참여자의 부서 및 이름 - 교육 보조 자료 • 확정된 시스템 이행계획 • 확정된 인수 시험 계획, 시험 사례, 시험 자료, 기대되는 결과 • 변환 검증 방법을 포함하는 변환 계획 • 신·구 시스템의 병행처리 기준 • 갱신된 작업계획 |

11. 11 단계 : 인수 시험 단계

이 단계는 처음의 사용자 요구 단계와 더불어 사용자의 관점에서 가장 중요한 단계라고 할 수 있다. 이 단계에서 사용자는 시스템이 사용자의 요구 사항에 맞게 제대로 개발이 되었는지를 확인, 시험하여 시스템을 인수하게 된다. 따라서 이 단계는 사용자의 참여에 의해서 이루어지며, 사용자는 시험 계획 및 기대되는 결과를 가지고 시스템을 시험하여 실제 결과가 이에 부합되는지를 확인하여야 한다. 이를 위해서는 실제 시험 기간보다 시험데이터를 작성하는데 더 많은 시간이 소요될 수 있다. 시험을 위한 데이터를 작성하는데 있어서 가장 중요한 사항은 시스템 시험의 목적을 분명히 인식하는 것이다. 즉 무엇 때문에 시스템의 사용자 인수 시험을 실시하는지 명확하게 정의하여야 한다. 대부분의 사용자들은 시험 데이터를 작성할 때 유효한

데이터만을 가지고 준비하는 경향이 있다. 그러나 사용자의 인수시험의 단계에서 중요한 시험 데이터는 무의미하다고 여겨지거나 또는 일어날 가능성이 없다고 생각되는 데이터까지도 포함시켜야 하는 것이다. 다시 말해서 시스템이 제대로 개발되었는지를 확인하는 것이 아니라 시스템이 잘못 개발되어 여러 오류를 내재하고 있다는 가정하에 이를 증명해 보이기 위한 인수 시험이 되어야 하는 것이다. 따라서 정상적인 입력절차에 따라 정상적인 출력물이 산출되는 것을 시험하는 것이 아니라, 숫자 필드에 문자의 입력이 허용되는 지부터 시작하여 금액의 필드에 마이너스가 입력될 수 있는지 등 비정상적인 입력 방법이 가능한지의 여부를 중점적으로 시험하여야 한다. 전산감사인은 예상 시험의 결과와 실제 결과 사이의 불일치에 대하여 확인 및 문서화가 제대로 수행되었는지를 파악하여야 한다.

인수 시험 단계에서의 목적, 활동 및 산출물을 표로 요약해 보면 다음과 같다.

[표 11] 11 단계 : 인수 시험 (Acceptance Testing)

| 목 적 | 활 동 | 산 출 물 |
|--|---|---|
| 시스템이 사용자의 요구 사항에 맞게, 또한 개발인력의 설계에 따라 제대로 이루어졌는지를 사용자의 관점에서 인수 시험을 실시함. * 참여자 - 사용자 - 관리층 - 시스템분석가 - 독립된 시험 요원 - 전산 품질 보증 인력 - 프로젝트 책임자 - 전산실장 - 전산감사인 | <ul style="list-style-type: none"> • 이전 단계에서 요청된 변화의 사항들이 인수 시험 계획에 포함되었는지 확인 • 다른 시스템과의 연결하에 전체 시스템의 인수 시험을 실시함 • 예상 결과와 실제 결과의 비교 • 예상 결과와 실제 결과 사이의 불일치에 대해서 확인 및 문서화 • 시험자료, 프로그램 논리의 변경 등을 통하여 시험 불일치의 수정 • 자료 파일 변환 계획의 확정 • 기기 및 소프트웨어 설치일 확정 • 병행 처리의 기준 확정 • 프로젝트의 작업계획 및 일정 갱신 | <ul style="list-style-type: none"> • 확정된 인수 시험 계획, 시험 사례, 시험 자료, 예상되는 결과 • 예상되었던 시험 결과와 실제 결과 사이의 불일치에 대한 문서화 • 확정된 자료 파일 변환 계획 • 확정된 기기 설치일 • 확정된 병행 처리 기준 • 갱신된 작업계획 • 사용자의 확인 |

12. 12 단계 : 변환 단계

이 단계는 여러 가지 방법에 의해 진행될 수 있다. 일반적으로 변환의 방법은 병행 변환 (Parallel Conversion), 직접 변환(Direct Conversion), 단계 변환(Phased Conversion) 및 시험 변환(Pilot Conversion) 등으로 분류될 수 있다.^[Hicks & Leininger, 1986]

이 가운데서 대부분의 대형 프로젝트의 경우는 변환 데이터의 일관성을 유지, 확인하기 위하여 병행 변환의 방법을 채택하게 된다. 따라서 이 단계에서는 기존의 자료가 새로운 자료로 제대로 변환되었는지 또한 새로운 기기 및 소프트웨어가 적절하게 도입되었는지 확인하는 것이 중요하다. 전산 감사인의 입장에서는 일정 기간동안 신·구

시스템을 병행처리 한 후에 이들 병행처리의 결과가 서로 동일한지 확인하는 과정이 또한 중요하다. 예를 들어 1개월 동안 신·구시스템의 병행 처리가 수행될 경우에 매일매일 산출되는 신·구시스템의 출력물을 대조하여 불일치의 유무 및 그 이유를 확인하여야 한다. 이전 단계의 사용자 인수 시험을 통과하였다 하더라도 이 단계에서 심각한 오류가 발견된다면 언제든지 프로젝트를 종결시킬 수 있다. 더구나 변환에 실패할 경우, 기존의 모든 자료 또한 훼손될 가능성이 있으므로 자료의 복구 절차에 관심을 기울여야 한다. 이 변환의 단계가 완료되면 정보시스템의 개발 프로젝트가 실질적으로 종결된 것으로 간주할 수 있다.

변환 단계에서의 목적, 활동 및 산출물을 표로 요약해 보면 다음과 같다.

[표 12] 12 단계 : 변환 (Conversion)

| 목 적 | 활 동 | 산 출 물 |
|--|--|---|
| 기존의 자료가 새로운 자료로 제대로 변환되었는지, 또한 새로운 기기 및 소프트웨어가 적절하게 도입되었는지를 확인함. 따라서 병행처리의 결과가 정확함을 확인함. * 참여자 - 사용자 - 관리층 - 시스템분석가 - 독립된 시험 요원 - 전산 품질 보증 인력 - 프로젝트 책임자 - 전산실장 - 전산부 직원 - 전산감사인 | <ul style="list-style-type: none"> • 인수 시험의 단계에서 요청된 변화 사항이 변환 계획에 제대로 반영되었는지 확인 • 자료 파일의 변환 및 검증 실시 • 새로운 기기 및 소프트웨어의 설치 • 병행처리를 수행하고 결과를 확인 • 시스템 매뉴얼, 프로그램 매뉴얼, 운영 매뉴얼 및 사용자 매뉴얼의 갱신 • 새로운 시스템을 실제 시스템으로 옮겨 가동 준비를 완료함 • 시스템 개발 프로젝트의 종결 | <ul style="list-style-type: none"> • 확정된 자료 파일 변환 계획에 포함되어야 할 사항들: <ul style="list-style-type: none"> - 변환되어야 할 자료 파일의 이름 및 유형 - 만족스런 변환을 위한 기준 - 변환을 검증할 방법 - 실제 변환 결과 - 신·구 시스템의 실제 병행 처리 결과 • 갱신된 매뉴얼들 • 새로운 시스템의 운영에 대한 확인 |

13. 13 단계 : 실제 운영 및 지원 단계

이 단계에서의 중요한 통제 요소는 다음의 두 가지이다. 첫째는 시스템이 시험 상태에서 실제 가동의 상태로 적절한 통제하에 이동되었는지를 확인하는 것이며 둘째는 실제 가동 시에 발생하는 문제점을 확인하고 이를 문서화하는 것이다. 우리나라의 경우 아직 정보시스템 부문의 직무 분리가 제대로 이루어져 있지 않기 때문에 개발된 프로그램이 실제 운영 시스템으로 넘어가서 등록되어 프로그램의 무결성(Integrity)을 유지하는지를 확인하는 전담요원들이 없는 실정이다(정 추익, 1991). 참고로 이러한 업무를 수행하는 범용 패키지가 국내에서 영업하는 미국계 은행에서 쓰여지고 있는데 그 기능을 업무순서에 의거하여 간략하게 살펴보면 다음과 같다(Citicorp, 1988).

- ① 사용자 변경관리 신청
- ② 변경관리 위원회 승인
- ③ 전담 프로그래머 지정
- ④ 변경관리 패키지에 관련 내용 및 번호 등록
- ⑤ Librarian이 Production Library에 있는 Source를 복사하여 개발용 기기의 전담 프로그래머의 Library로 옮김
- ⑥ 전담 프로그래머가 프로그램 변경

- 자체 테스트(Unit Testing) 후에 Staging Library로 옮김
- ⑦ 품질보증 인력(QA Staff)이 해당 Source를 QA Library로 옮김
 - ⑧ Source Compile
 - ⑨ QA Staff에 의한 1차 Test
User와 함께 2차 Test(User Acceptance Test)
3차 종합 Test (Integration Test)
 - ⑩ Test 결과가 불만족스러우면 다시 Staging Library로 Bounce Back시킴
 - ⑪ Before and After Source Comparison
 - ⑫ Librarian이 QA Library의 Source를 Production으로 옮김
 - ⑬ Source Compile
 - ⑭ All documents Sign-off
 - ⑮ 사용자에게 변경 완료 통보

이 모든 기능들은 패키지를 통하여 이루어지며 자동으로 기록된다. 여기에 참여하는 모든 인원은 각자의 제한된 권한만을 가지고 업무를 수행하여 시스템의 변환 시에 어느 정도의 통제 기능을 수행할 수 있을 것이다.

실제 운영 및 지원 단계에서의 목적, 활동 및 산출물을 표로 요약해 보면 다음과 같다.

[표 13] 13 단계 : 실제 운영 및 지원 (Production Operations and Support)

| 목 적 | 활 동 | 산 출 물 |
|---|---|--|
| 시스템이 시험상태에서 실제 가동의 상태로 적절한 통제하에서 이동되었는지를 확인하고 실제 가동을 지원함 * 참여자 - 시스템분석가 - 적용업무 프로그래머 | <ul style="list-style-type: none"> • 시험 라이브러리에서 실제 가동 라이브러리로 프로그램과 데이터를 옮김 • 실제 가동에 대한 지원 • 실제 가동시의 문제점을 확인하고 문서화 • 즉시 또는 미래의 개선을 위해 실제 문제를 분류 • 즉각적인 수정 요청에 의한 문제의 해결 | <ul style="list-style-type: none"> • 실제 가동 문제 보고서에 포함되어야할 사항: <ul style="list-style-type: none"> - 문제의 특성 - 문제의 조사 및 분석 - 문제의 해결 • 시스템 운영시 요청되는 개선점 및 변경 요망사항에 대한 기록 |

| | | |
|--|--|--|
| <ul style="list-style-type: none"> - 시스템 프로그래머 - 전산실 직원 - 프로젝트 책임자 - 전산실장 - 전산품질 보증 인력 - 전산감사인 | | |
|--|--|--|

14. 14 단계 : 실행 후 검토 단계

이 단계는 SDLC의 최종 단계로, 사용자와 정보시스템의 부문에 종사하는 인력을 면담하여 시스템의 성공여부를 평가하는 것이다. 시기적으로는 시스템의 이행후 6개월 내지 1년후의 시점이 적당할 것이다. 이 단계에서의 평가는

사용자의 만족도가 중요시되며 또한 기존의 예산과 대비하여 새로운 시스템의 효과 분석이 행하여져야 한다. 이러한 모든 자료는 향후 새로운 시스템의 개발에 대한 제안 자료로 사용되어질 것이다.

실행 후 검토 단계에서의 목적, 활동 및 산출물을 표로 요약해 보면 다음과 같다.

[표 14] 14 단계 : 실행 후 검토 (Postimplementation Review)

| 목 적 | 활 동 | 산 출 물 |
|--|---|---|
| <p>기존의 예산과 비교하여 신시스템의 실제 비용, 효과의 관점에서 성공 여부를 평가함</p> <ul style="list-style-type: none"> * 참여자 <ul style="list-style-type: none"> - 사용자 - 관리층 - 전산 품질 보증 인력 - 전산실장 - 전산감사인 | <ul style="list-style-type: none"> • 사용자와 전산 인력을 면담하여 실행 후 검토를 수행함 • 기대되었던 비용 편익 분석의 관점에서 실행 후 검토 보고서를 작성하여 보고함 • 최초의 시스템 계획 및 일정이 지켜졌는지 확인함 | <ul style="list-style-type: none"> • 이행후 검토 보고서에 포함되어야 할 내용: <ul style="list-style-type: none"> - 예산대비 실제 비용 - 예산효과 대 실제효과 - 예산이익 대 실제이익 - 계획상 완료일 대 실제 완료일 - 원래 목적 대 실제 성취된 목적 - 문서화 및 통제의 질에 대한 평가 - 예상 가동률 대 실제 시스템가동률 - 전반적인 사용자 만족도 및 시스템에 대한 이해 수준 - 향후 시스템 개발 계획에 대한 제안 |

제 4 장 결론 및 요약

본 논문은 정보시스템의 개발방법론 가운데서 가장 널리 쓰여지고 있는 시스템개발 수명주기론을 Vallabhaneni의 모델을 중심으로 하여 살펴보았다. 앞에서 살펴본 바와 같이 이제까지의 대부분의 시스템개발 수명주기론은 개발자 위주로 설계의 측면에서 연구되어 왔으나 Vallabhaneni의 모델은 사용자의 관점에서 즉 개발통제의 관점에서 제시된 이론이다.

따라서 Vallabhaneni 모델의 특징은 정보시스템의 개발과정에서 단위시험, 시스템 시험 및 인수 시험 등 3가지의 시험을 각각 단계별로 실시하는데 있다.

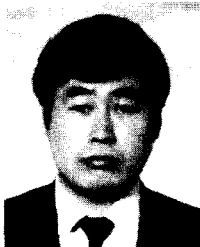
오늘날 정보시스템의 구축은 급증하는 사용자의 요구를 다 수용하지 못하여 업무개발에 있어서 정체현상이 나타나고 있으며 기존 시스템의 노후화에 따른 유지보수 비용의 증가추세라는 문제에 직면하고 있다. 더 나아가서 개발된 시스템의 안전성과 신뢰성에 대해서도 많은 의문이 제기되고 있는 실정이다. 그러므로 이러한 문제점들을 사전에 방지하고 제대로 된 정보시스템을 개발하기 위해서는 사용자의 참여가 절대적으로 요망된다. 정보시스템의 개발에 참여하는 사용자는 단순히 사용자의 요구사항만을 기술하는 것에서 벗어나 각 단계에 모두 참여하여야 하며 특히 인수시험시에 개발된 시스템이 사용자의 요구사항에 맞게 정확하게 개발이 되었는지를 확인하여야 한다. 결국 정보시스템의 개발은 정보시스템 부문에 의해서 이루어지는 것이 아니라 시스템을 사용할 사용자의 주도하에 수행되어야 성공할 수 있다는 것을 보여주고 있다.

참 고 문 헌

- 감사원, 『전산업부와 감사』 1990년 12월
- 김 세현, 『컴퓨터범죄와 프라이버시 침해』, 회성출판사, 1989년
- 안 중호, 『경영과 정보통신기술』, 학현사, 1994년
- 이 주현, “프로젝트 외주관리 기법”, 과학기술정보연구소, 1991년 1월
- 정 충식, “정보시스템 기능의 조직화에 관한 연구” 고려대학교 경영대학원 석사학위 논문, 1986년 11월 “정보시스템 감사에 관한 보고서 -은행의 컴퓨터 범죄를 중심으로-” 안권회계법인 경영자문부, 1988년 1월 “우리나라 은행의 정보시스템감사 실태분석” 성균관대학교 행정대학원 석사학위 논문, 1991년 11월 “은행정보시스템의 접근통제에 관한 사례연구 - 적용업무 프로그램의 불법변경 방지책을 중심으로-” 한국경영정보학회 추계학술발표논문집, 1994년 11월
- 채 문규, “컴퓨터범죄 선진국수준에 이르다”, 업데이트, 1990년
- 한국금융연수원, 『영업점 EDPS 안전관리와 사고예방』 1990년
- 한국생산성본부, 『정보시스템 감사 및 평가전문가』, 1991년 10월
- 한국정보시스템감사인협회, 『정보시스템 감사 기준에 관한 WORKSHOP』, 1992년 6월
- Citicorp, Change Management Control Manual, 1988
- Davis, G.B & Olson, M.H. Management

- Information Systems -Conceptual Foundations, Structure, and Development, Second Edition, McGRAW-HILL BOOK COMPANY, 1985
- Hicks, J.O.& Leininger, W.E. Accounting Information Systems. Second Edition, West publishing Company, 1986
- Ives, B. and Olson, M.H. "Manager or Technician? The Nature of the Information System Manager's Job" MIS Quarterly, Vol. 6, No.4(December 1982), pp.49-63.
- Menkus, B. & Ruthberg, Z.G. CONTROL OBJECTIVE, The EDP Auditors Foundation, April, 1990
- Naumann, J.D. and Jenkins, A.M. "Prototyping : The New Paradigm for Systems Development" MIS Quarterly, Vol. 6(September 1982)
- Nolan. R.L. and Gibson .C.F. "Managing the four stage of EDP growth" Harvard Business Review, Jan-Feb. 1974
- Nolan. R.L. "Managing the Crises in data Processing" Harvard Business Review, March-April, 1979
- Vallabhaneni, S. Rao, Information Systems Audit Process, The EDP Auditors Foundations, 1988

□ 著者紹介



정 충 식

1984년 고려대학교 사회학과 학사 취득
1987년 고려대학교 경영대학원 경영학 석사 취득
1992년 성균관대학교 행정대학원 행정학 석사 취득
1995년 현재 성균관대학교 대학원 행정학과 박사과정
용인공업전문대학 사무자동화과

※ 관심분야 : 정보시스템의 보안, 감사 및 정보화사회론