

회계정보시스템 개발에 관한 연구*

A Study on the Development of Accounting Information Systems

이 찬 민**

Lee, Chan-Min

目

次

I. 서언

1. 문제의 제기

2. 연구 목적

II. 정보시스템의 개발방법

1. 개발전문가중심 개발법

2. 사용자중심 개발법

3. 통합개발법

III. 회계정보시스템의 개발

1. 자료처리시스템

2. 정보처리시스템

3. 자료처리시스템의 개발

4. 정보처리시스템의 개발

IV. 결론

I. 서 언

1. 문제의 제기

기업에는 기업활동을 지원하고 기업의 내부 경영자와 외부 투자자 등의 의사결정에 유용한 각종 기업정보를 처리하여 제공하는 여러 가지 형태의 정보시스템이 설치되어 운영되고 있다.

오늘날 기업간의 경쟁은 점차 심화되고, 기업의 주주들과 투자자들은 지속적인 기업성장을 요구하는 한편, 정부는 기업윤리의 실천과 자연환경의 보호를 요구하는 등 정부의

* 이 연구는 1998년도 단국대학교 대학연구비의 지원으로 연구되었음.

** 본 연구소 상임연구원, 단국대학교 상경학부 교수

규제가 날로 강화되고 있는 바, 이러한 기업내부와 외부환경의 변화와 요구를 수용하는데는 새롭고 다양한 정보를 신속하게 제공해줄 수 있는 기업내의 정보시스템의 개발이 그 어느 때 보다도 필요하다고 본다. 또한 컴퓨터와 통신기술 등 정보기술의 눈부신 발달은 기존 정보시스템의 하드웨어는 물론 소프트웨어를 급속하게 전부화시켜 보다 성능이 우수한 정보시스템을 개발하여 도입할 필요성이 더욱 높아지게 되었다.

기업의 정보시스템을 개발하고 설계하는 데 있어 이제까지 가장 널리 일반적으로 이용된 개발 방법이 개발전문가 중심의 시스템개발 수명주기법(Systems Development Life Cycle, SDLC)이다. 이 전통적인 개발방법(이하에서 전통적 SDLC 개발법)은 계획과 조사, 분석, 설계, 실행, 그리고 사후 관리 등 여러 단계가 연속적으로 진행되는데, 각 단계별로 작업이 완료되면 개발작업을 총괄하고 있는 개발운영위원회의 최종 심사와 승인을 받아 다음 단계로 진행이 이어지는 특성을 가지고 있다.

그런데 이 개발방법은 개발에 필요한 시간이 많이 요구되는 것과 정보시스템 사용자의 정보요구가 충분히 개발에 반영되지 못한다는 등, 여러 가지 문제점이 지적되었다.¹⁾ 이러한 전통적 SDLC 개발방법의 문제점을 해결하고자 새롭게 개발된 것이 사용자 중심의 원형개발법(prototyping approach)이다. 이 새로운 개발 방법은 시스템의 핵심적인 부분을 모형화한 시스템 원형(prototype)을 중심으로 시스템을 개발하는 것과 사용자가 이 시스템원형을 모의실험을 통해 수정할 수 있다는 점을 그 특징으로 하고 있다. 최초원형(initial prototype)의 구축으로부터 시작하여 시스템 사용자의 요구를 충족시키는 최종 원형(final prototype)이 구축될 때까지 계속 수정 작업이 반복적으로 이루어진다.

이와 같이 새롭게 제시된 원형개발법도 개발과정에서의 통제장치가 마련되어있지 않아 시스템원형에 대한 수정작업이 끝없이 계속 반복될 수 있다는 것과 개발할 정보시스템이 대규모일 경우 원형의 구축이 쉽게 이루어지지 못한다는 문제 등이 지적되고 있다.²⁾

원형개발법이 전통적 SDLC 개발법의 부족한 점을 보완하기 위해 개발된 것은 사실이나 전통적 SDLC 개발법을 대체하기에 충분한 개발법은 못되며 따라서 독자적으로 이용할 수 있는 개발법이 아니고 전통적인 SDLC 개발법과 함께 사용해야 그 개발효과를 최대화할 수 있는 것이다.

2. 연구 목적

1) T. R. Berrisford and J. C. Wetherbe, "Heuristic Development: A Redesign of Systems Design," MIS Quarterly, 1979, pp. 1-19.

2) B. Marshall Romney, J. Paul Steinart, and E. Cushing, Accounting Information Systems, Addison Wesley, 7th ed., 1997, pp. 407-413.

회계정보시스템 개발에 관한 연구

회계정보시스템은 기업의 여러 정보시스템 가운데 가장 중요한 정보시스템의 하나로 회계정보시스템은 크게 일상적인 회계상의 거래를 처리하는 자료처리시스템(data processing system)과 회계정보이용자의 의사결정을 지원하는 정보처리시스템(information system)으로 구분할 수 있다.

자료처리시스템은 정보처리시스템을 포함한 기업의 각종 정보시스템이 필요로 하는 자료원(data source)을 제공하는데, 정보처리시스템은 자료처리시스템을 통해 산출되는 정보와 그 밖의 정보를 처리하여 회계정보이용자의 합리적인 의사결정에 유용한 회계정보를 산출시키는 의사결정을 지원하는 성격을 가진 회계정보시스템이다.

본 논문의 목적은 첫째, 개발전문가 중심의 개발법 가운데 대표적인 전통적 SDLC 개발법과 사용자 중심의 개발법 가운데 대표적인 원형개발법의 문제점이라고 지적 받고 있는 점들을 재평가하여 전통적 SDLC 개발법과 원형개발법을 각각 수정하여 수정 SDLC 개발법과 수정 원형개발법을 제시한다. 둘째, 전통적 SDLC 개발법과 원형개발법이 서로 대체 가능한 개발법이 아니라 서로 보완적인 관계라는 것을 밝혀, 두 개발법 간의 새로운 관계를 정립하고자 한다. 셋째, 전통적 SDLC 개발법과 수정된 SDLC 개발법을 이용하여 자료처리시스템을 개발하는 과정을 살펴보며, 아울러 SDLC 개발법과 원형개발법을 통합한 통합개발법을 이용하여 정보처리시스템을 개발하고자 한다.

따라서 자료처리시스템의 개발을 위해서는 SDLC 개발법에 기초한 시스템 개발전문가 중심의 개발방법이 제시되는 한편, 정보처리시스템의 개발을 위해서는 SDLC 개발법에 원형개발법을 합친 통합개발법에 기초한 사용자중심의 개발법의 활용을 제안하고자 한다.

다음의 제2장에서는 전통적인 SDLC 개발법과 원형개발법 각각에 대한 개발과정, 특성, 문제점과 이에 대한 재해석, 그리고 각 개발법의 수정에 대해 논하고, 제3장에서는 회계정보시스템의 두 축인 자료처리시스템과 정보처리시스템의 특성과 각각의 하위처리시스템들을 논하고 이어서 자료처리시스템은 SDLC 개발법을 활용하여 개발하고 정보처리시스템은 통합개발법을 활용하여 개발하고자 한다.

II. 정보시스템의 개발방법

기업의 정보시스템을 개발할 때 사용하는 개발방법에는 여러 가지가 있는데, 이들은 크게 정보시스템 개발전문가의 주도하에 개발작업이 이루어지는 개발전문가중심 개발법과 정보시스템을 이용하는 사용자의 주도적인 참여 하에 개발작업이 이루어지는 사용자중심 개발법으로 나눌 수가 있다.

1. 개발전문가중심 개발법

개발전문가 중심의 개발법의 대표적인 것이 전통적 SDLC 개발법 또는 시스템개발 수명주기법인데 다음에서 개발과정, 특성, 평가, 그리고 전통적 SDLC 개발법의 수정에 대해 논의한다.

(1) 전통적 SDLC 개발법의 개발과정

전통적 SDLC 개발법은 시스템개발을 여러 단계로 나누어 수행한다. 첫 단계는 시스템 조사와 계획(systems investigation and planning)인데, 이 단계에서는 시스템 개발을 주도할 집단을 구성한다. 먼저 시스템 개발전문가집단(systems developers)을 구성하는데, 이들은 시스템개발의 모든 단계에 걸쳐 개발작업에 직접 참여하며 이들이 가지고 있는 전문지식을 제공한다. 구성원은 시스템 개발과 관련된 여러 분야(시스템분석, 전산자료처리, 회계, 산업공학 등)의 전문가로 이루어진다.

또 다른 개발을 주도하는 집단은 개발 운영위원회(steering committee)인데 이 위원회는 시스템개발과 관련된 최고의 의사결정기관이다. 구성원은 주로 내부의 경영자(최고경영자, 회계책임자, 재무담당책임자, 정보시스템관리 책임자 등) 각 분야별 책임자, 그리고 내부감사인 등)로 구성되며 이들은 각 개발 단계마다 개발 전문가집단이 단계별 개발작업을 끝낸 후 정리하여 작성하는 단계별 개발작업 보고서를 검토하여 승인 여부를 결정하는, 즉 개발작업을 총 지휘 감독하는 역할을 맡고 있다.

두번째 단계인 시스템분석(systems analysis)은 기존시스템이 시스템사용자의 정보 요구를 만족시키고 있는가 여부를 조사하고 분석하여 사용자의 불만족이 심각하다고 판단되면 새로운 시스템의 개발을 통한 해결방안을 모색하는 단계이다.

시스템분석은 다음의 네 가지 세부적인 과정을 거쳐 이루어진다. 첫째, 현행시스템의 효율성을 검토하기 위해 먼저 시스템목적이 명시된다. 중요한 시스템목적 가운데는 시스템운영에 소요되는 비용보다 시스템운영으로 기대되는 효익이 더 커야한다는 것과, 시스템을 통한 출력보고서는 신뢰성이 있고 유용해야한다는 것 등이 포함된다. 시스템분석을 위한 두 번째 작업은 현행시스템의 실태조사와 분석이다. 현행시스템의 강점과 약점을 파악하기 위해 각종 자료수집(예를 들면, 문서검증, 시스템운영 상황의 점검, 관련자 면담 등)과 자료분석(각종 통계처리, 시스템흐름도 작성 등)을 하게 된다. 세 번째 작업으로는 내부통제 시스템의 실태조사와 분석이 이루어진다. 마지막으로 시스템분석의 결과를 개발 전문가집단은 보고서로 작성하여 개발 운영위원회에 제출한다. 이 시스템 분석보고서(systems analysis report)에는 시스템분석 결과에 따른 현행시스템의 문제점과 이에

회계정보시스템 개발에 관한 연구

대한 해결 방안이 제시되며, 운영위원회의 승인을 거쳐 다음 개발 단계인 시스템설계로 이어진다. 그러나 운영위원회의 승인을 받지 못하면 시스템분석에 대한 재 작업을 해야 한다.

세 번째 단계인 시스템설계(systems design) 작업도 앞의 시스템 분석과 마찬가지로 다음의 네 가지 과정을 거쳐 완성된다. 먼저 시스템 실현가능성을 검토(feasibility evaluation)한다. 이는 시스템 분석보고서에서 해결방안으로 제시된 새로운 시스템이 현실적으로 실현가능한지를 검토한다. 이에는 구체적으로 기술상의 검토(technical feasibility: 시스템분석보고서에서 제시된 해결방안이 기술적인 면에서 현실성이 있는가 여부를 검토), 운영상의 검토(operational feasibility: 운영담당자의 자격과 경험상의 능력 여부를 검토), 일정상의 검토(schedule feasibility: 시스템의 설치 운영기간이 회사가 미리 정한 기간이내에 가능한가 여부를 검토), 그리고 경제상의 검토(economic feasibility: 시스템 도입에 따른 추가 효익이 시스템 도입에 따른 추가비용 보다 큰가 여부를 검토)가 포함된다.

시스템 실현가능성 검토과정에서 궁정적이라는 결론이 나면 시스템설계의 본 작업인 출력과정, 처리과정, 그리고 입력과정이 설계된다. 아울러 자료저장과 관리, 그리고 통제장치에 대한 구체적 설계가 이루어진다. 설계작업이 끝나면 '시스템설계 작업을 정리하여 시스템설계 명세서(systems specifications report)'를 작성한다. 이 보고서에는 회사의 운영실태를 설명하고 현행 시스템이 안고 있는 문제점, 상세한 시스템 설계내용, 시스템 공급자들이 그들의 납품 제안서에 포함할 내용(CPU처리 속도와 용량, 입·출력장치의 유형, A/S와 교육훈련 program, 그리고 설치운영에 소요되는 절차와 시간 등)이 포함된다. 이 시스템설계 명세서는 개발 운영위원회의 검토와 승인을 거쳐 시스템 공급자에게도 제출된다.

네번째 단계는 시스템의 실행과 사후관리(systems implementation and follow-up)이다. 시스템의 실행은 공급자의 선정(vendor selection)과 실행작업으로 나누어 지는데 실행작업은 시스템을 설치할 장소의 선정을 시작으로 새로운 시스템의 작동(initial operation)에 이르기까지 여러 작업이 단계적으로 이루어진다. 사후관리는 분석단계에서 설정한 시스템목적에 비추어 새로운 시스템의 효율성을 검토하여 개선이 가능한 시스템 부분을 보완하게 되며, 동시에 이 과정에서 획득한 각종 유용한 정보들이 앞으로 다시 시스템을 개발할 때 참고 자료로 사용하게 된다.

(2) 전통적 SDLC 개발법의 특성

전통적 SDLC 개발법은 몇 가지 특성을 가지고 있는데, 이는 다음과 같다. 첫째, 개발

작업이 계획과 분석, 설계와 실행, 그리고 사후관리 등 여러 단계로 나누어 있으며 또한 각 단계별로 작업이 아주 세밀하게 구성되어 있다.

두번째 특성은 하나의 개발 단계에서 다음 개발단계로 이행하기 전에 반드시 최고경영자로 구성된 개발 운영위원회의 감독과 통제가 이루어진다. 이 위원회의 승인을 얻지 못하면 다음 개발단계로 진행이 안되고 승인을 받을 때까지 재 작업을 해야 한다.

세번째 특성은 시스템 사용자의 시스템에 대한 여러 요구사항(system requirements)이 충분히 반영이 안 될 수가 있다. 그 이유는 시스템 사용자의 시스템 요구사항이 개발 전문가에게 전달되지만 이 요구사항의 수용은 개발 전문가의 전문가적 판단에 따라 결정되기 때문이다.

네번째 특성은 개발의 매 단계의 작업이 종료되면 개발작업을 요약한 보고서를 작성하게 함으로써 정보시스템의 통제를 구체적인 문서화(documentation) 작업을 통해 수행하고 있다.

(3) 전통적 SDLC 개발법에 대한 평가

전통적 SDLC 개발법은 통제되고 구체적이고 문서화된 개발법인 것이다. 이러한 통제로 인해 주어진 예산이나 시스템계획의 범주안에서의 개발을 가능하게 한다. 그리고 개발 과정에 필요한 모든 개발작업이 상세하게 포함되어 있어 어떤 형태의 정보시스템의 개발에도 적용이 가능하다. 또한 개발과정이 문서화되어 장차 유사한 정보시스템을 개발할 때 유용하게 활용할 수 있다는 장점이 있는 반면에 다음과 같은 실행상의 어려움이 있다.

첫째, 각 단계의 수행 후에 개발작업에 대한 승인이 이루어진 후 다음단계로의 작업이 가능하기 때문에 개발의 추진기간이 상당히 길어질 수 있어 길게는 수년, 짧아도 수개월 씩의 기간이 시스템 개발에 소요되어 시스템 개발에 요구되는 시간이 너무 길다.³⁾ 이러한 긴 개발시간은 필연적으로 개발에 대한 비용을 가중시키며 급변하는 정보기술의 효율적 활용을 불가능하게 한다.

둘째, 시스템분석 단계에서 사용자에게 새로운 시스템에 대한 요구목록을 제출할 것을 요구하지만 사용자들은 시스템에 대한 전문지식이 결여되어 새로운 시스템이 갖추어야 할 새로운 기능과 성능에 대한 표현과 요구를 제대로 하지 못할 수가 있다. 즉, 시스템 사용자들은 기존 시스템에 대한 불만은 이야기할 수 있지만 새로운 시스템에 대해 원하는 바를 체계화하여 이를 정확히 표현하는 것이 쉽지 않은 것이다.

3) W. Doll and G. Torkzadeh, "A Discrepancy Model of End-User Computing Involvement," Management Science, Oct. 1989, pp. 1151-1171. 이들의 연구에 의하면 151개의 개발 프로젝트를 조사한 결과 개발에 소요된 기간이 짧은 것이 4 개월이며 긴 것은 56 개월이 소요되었으며, 평균 개발기간은 18 개월이었다.

회계정보시스템 개발에 관한 연구

셋째, 한 번 시스템 사용자의 시스템 요구사항을 접수한 후에는 더 이상 요구사항을 추가하거나 처음의 요구사항을 수정할 수 있는 수단이 없다. 각 개발 단계별 작업이 완료됨에 따라 시스템 사용자의 요구사항과 일치하는지를 확인하는 절차가 있게 되는데, 이 과정에서 최초의 요구사항에 많은 변경이 있을 경우 이를 효과적으로 수용할 방법이 없다. 즉, 차수 단계부터 마지막 코딩단계에 이르는 과정에 이르기까지 사용자 피드백을 거의 허용하지 않는다. 따라서 새로운 시스템에 대한 사용자의 만족을 저하시켜 시스템 개발의 효과를 감소시키게 된다.

(4) 전통적 SDLC 개발법의 수정

전통적 SDLC 개발법은 신규로 정보시스템 전체를 개발하는 데는 효과적인 개발법으로 신규가 아닌 기존 정보시스템의 일부를 변경시키는 개발의 경우, 전통적 SDLC 개발법은 수정이 필요하다. 전통적 SDLC 개발법은 개발의 첫 작업단계인 조사와 계획단계에서 개발전문가 집단의 개발작업을 지원하고 감독하는 개발 운영위원회를 구성하는데 운영위원회의 구성원으로 최고경영자, 회계책임자, 재무담당책임자, 정보시스템관리 책임자 등 각 분야별 책임자를 비롯한 여러 분야의 다양한 사람들이 참여하는데, 일부 시스템의 변경의 경우, 이를 가운데 변경하는 일부 시스템과 직접적인 관계가 있는 부서의 책임자와 관련 사용자만이 참여하게 함으로써 운영위원회의 의사결정을 신속히 할 수 있는 것이다. 또한 전통적 SDLC 개발법의 경우, 매 단계의 작업이 끝날 때마다 운영위원회의 감독과 승인을 거쳐야 하나, 일부 시스템변경의 경우 운영위원회는 감독과 승인의 역할보다는 개발전문가의 개발작업을 지원하는 역할을 담당하게 함으로써 개발작업을 신속하게 진행할 수 있도록 해야 한다.

2. 사용자중심 개발법

전통적 SDLC 개발법의 문제점을 해결하기 위한 방안으로 제시된 것이 원형개발법(prototyping approach)이다.

(1) 원형개발법의 개발 과정

이 방법은 시스템 전체를 단계적으로 세밀하게 개발하는 것이 아니라, 시스템의 기본적인 부분만을 모형화한 형태, 즉 시스템 원형(prototype)의 구축을 통해 시스템을 개발하는 방법이다.

이 방법은 다음의 다섯 가지 과정을 거쳐 구축된다. 먼저, 최초의 시스템 원형이 개발

産業研究

된다. 시스템 사용자가 시스템이 갖추어야 할 시스템 요구사항을 제시하면 이에 기초하여 시스템 개발전문가가 시스템 원형을 최초로 개발한다. 이 시스템원형은 시스템이 갖추어야 할 기본요소를 모형화한 것으로서, 이를테면 시스템의 출력과정(또는 입력과 처리과정 까지를 포함할 수 있음)을 대상으로 하여 최초의 시스템원형이 구축된다.

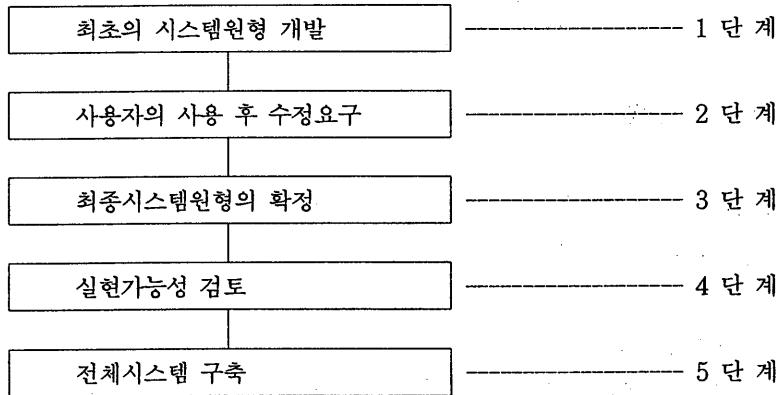
두번째 단계는 최초의 시스템원형을 시스템 사용자가 직접 테스트한다. 시스템 사용자가 최초로 개발된 시스템원형을 사용해본 후 불만족스런 점이 있으면, 이의 개선을 위해 시스템 개발전문가에 최초의 시스템원형에 대한 수정을 요구하면 이를 시스템 개발전문가는 수용하여 최초의 시스템원형을 수정한다. 수정된 시스템원형을 시스템 사용자가 다시 사용한 후, 보완할 점이 또 발견되면 다시 수정을 요구하여 시스템원형이 재 수정된다.

세번째 단계는 사용자가 만족하는 시스템원형이 최종적으로 구축된다. 즉, 시스템사용자의 테스트와 시스템원형의 수정이 시스템 사용자가 만족할 때까지 계속되어 최종적인 시스템 원형이 완성된다.

네번째 단계는 최종적으로 구축된 시스템원형에 대해 실현가능성 검토(기술상, 운영상, 일정상, 그리고 경제적 타당성 검토)를 하여 이를 통과한 시스템원형은 실제 사용 가능하게끔 세부사항들이 첨가되며, 아울러 시스템원형의 개발에서 제외시킨 입력과 처리과정, 자료의 저장과 관리, 통제 장치 등이 추가된다. 실현가능성 검토가 끝나면 최종적으로 전체시스템의 구축이 완성된다.

개발과정을 도표로 표시하면 <도표 1>과 같다.

<도표 1> 원형개발법의 개발과정



(2) 특성

개발초기에 시스템 모형을 간단히 만들어 사용자에게 보여주고 사용자로 하여금 이를

회계정보시스템 개발에 관한 연구

실제로 작동시켜 시스템의 기능 및 제공하는 정보의 추가와 변경 또는 삭제를 요구하도록 하여 시스템을 점차 개선시켜 나가는 방식이다. 중요 구현기법으로는 기본적인 요구사항들을 명확하게 파악하기 위해 일시적으로 구축되었다가 사용자의 승인 후 폐기하는 폐기용 방식과, 최초의 원형을 점차 개선시키고 기능을 덧붙임으로써 결국 최종적인 시스템으로 구현하는 구축용 방식이 있다.⁴⁾

원형개발법은 다음과 같은 특성이 있다. 먼저, 능동적인 시스템 사용자의 개발 작업에의 참여와 새로 개발한 시스템에 대한 높은 만족도이다. 즉, 시스템 사용자가 시스템원형을 여러 번 반복하여 테스트하고 그 결과 필요한 부분을 보완함으로써 시스템에 대한 높은 만족도를 기대할 수 있다.

둘째는 시스템의 계획과 분석, 그리고 설계가 동시에 이루어질 수 있어 개발에 필요한 기간이 비교적 짧아, 새로운 정보기술을 시스템개발에 효과적으로 접목시킬 수 있다. 아울러 개발비용을 절감할 수 있다.

셋째, 시스템개발의 신축기능성(flexibility)이다. 시스템원형에 포함할 시스템의 기본 요소를 시스템개발 당시의 형편에 따라 그 범위를 좁혀 시스템 출력과정만을 다룰 수도 있고, 범위를 넓혀 입력과 처리과정까지도 포함시킬 수도 있는 것이다.

따라서 시스템에 대한 요구사항을 미리 구체화하기 어려울 때와 개발도중 상당한 요구 사항의 변화가 예상될 때 특히 유용한 개발법인 것이다.

(3) 원형개발법의 평가

원형개발법의 장점으로는 사용자의 정보요구사항과 시스템요구사항에 대하여 명확한 인식이 이루어진다는 것이다. 즉, 시스템 사용자와 시스템 개발전문가간에 의사소통이 원활하다는 것이다. 전통적 SDLC 개발법에서는 시스템분석 단계에서 사용자에게 새로운 시스템에 대한 요구목록을 제출할 것을 요구하지만 사용자들은 시스템에 대한 전문지식이 결여되어 새로운 시스템이 갖추어야할 새로운 기능과 성능에 대한 표현과 요구를 체계적으로 하지 못할 수가 있다.

원형개발법은 새로운 시스템에 대한 요구사항을 보다 명확히 인식할 뿐만 아니라 사용자들을 시스템개발과 관련하여 교육과 훈련을 시키는 데 큰 도움을 준다. 즉, 시스템의 설계와 개발과정에 사용자들이 참여함으로써 이들은 시스템에 대해 더 많은 관심과 이해를 가지며 이를 토대로 시스템을 사용하는데 있어 보다 효율적으로 활용할 수가 있는 것이다. 또한 사용자들이 개발에 직접 참여함으로써 나중에 시스템을 최종적으로 개발하여

4) M. A. Mahmood, System Development Method-A Comparative Investigation,"MIS Quarterly, 1987, pp.293-311.

産業研究

설치 운영할 때는 이미 시스템의 새로운 내용에 대해 익숙해져 있음으로 인하여 거부감 없이 쉽게 수용하여 사용할 수가 있는 것이다.

이와 같이 원형개발법은 최종시스템에 대한 친밀감과 만족, 개발자와 사용자간의 원활한 의사소통, 개발과정중의 사용자의 새로운 요구의 반영, 그리고 개발기간의 단축이 가능하다⁵⁾는 장점이 있는 반면에, 원형개발법은 다음과 같은 실행상의 어려움이 있다.

먼저, 사용자가 시스템설계에 직접 참여한다는 이유로 개발전문가는 문서화작업을 중요하게 생각하고 있지 않다는 점이다. 즉, SDLC 개발법처럼 단계별 과제와 작업보고서가 구체적으로 제시되지 않기 때문에 관리상의 어려움이 있는 것이다.⁶⁾

둘째, 시스템 사용자의 요구가 최대한 반영되기 위해 마련된 장치인 사용자 테스트와 시스템원형의 수정이 끝없이 반복될 수 있는 것이다. 즉, 시스템원형의 수정을 위한 객관적 기준이 마련되지 않은 상태에서 사용자의 정보요구가 만족될 때까지 시스템원형에 대한 수정작업이 계속 끝없이 이루어질 수 있는 것이다.

셋째는 원형개발법은 시스템의 모든 기능을 다 갖춘 시스템이 아니라는 점이다. 시스템 원형에 대한 시스템 사용자의 요구사항은 최대한 반영되지만, 시스템원형이 확정된 이후에 추가하는 부분에 대해서는 사용자의 요구사항은 반영이 안되고 있다.

넷째는 이 방법은 개발 작업에 사용자의 많은 시간을 요구하고 적극적이고 지속적인 참여와 의견 표명이 필수적으로 전제가 되어야 한다. 사용자의 적극적 협조가 없이는 효과를 기대하기 어렵다.⁷⁾ 따라서 개발 프로젝트의 규모가 대형일 경우 사용자들도 다양하고 많아 이들의 의견을 종합하기가 쉽지 않아 원형개발법은 특히 중소형 시스템개발에 유용하다고 하겠다.

따라서 원형개발법은 전체정보시스템이 아닌 일부분의 변경을 요구하는 경우, 사용자의 요구가 불분명하여 개발과정에서 그 요구가 변할 수 있는 경우, 그리고 시스템환경의 빠른 변화로 인해 자주 변경을 해야하는 경우에 보다 효과적인 정보시스템 개발방법인 것이다.

(4) 원형개발법의 수정

원형개발법을 이용할 경우 시스템원형을 사용자가 테스트해 본 후 수정을 여러 번 반복적으로 하게 되는데 이 수정작업을 언제까지 할 것인가에 대한 지침을 마련할 필요가 있다. 이를테면, 최대로 허용되는 수정회수를 사전에 지정한다든지, 또한 최소한의 사용

5) E. C. Harel & E. R. Mclean, "The Effect of Using a Nonprocedural Language on Programmer Productivity," MIS Quarterly, Vol. 9, No. 2, 1985, pp. 109-120.

6) M. Alavi, "An Assessment of the Prototyping Approach to Information Systems Development," Communications of the ACM, Vol. 27, No. 6, 1984, pp. 556-563.

7) L. Gavurin Stuart, Journal of Systems management, 1991, pp. 13-17.

회계정보시스템 개발에 관한 연구

자의 시스템수용 수준을 사전에 정할 필요가 있다. 원형개발법은 시스템전체가 아닌 시스템의 일부를 변경할 때 사용할 수 있는 개발법으로 시스템전체를 개발하고자 할 때는 추가적으로 개발작업을 보완하든지, 아니면 다른 개발법, 즉 전통적 SDLC 개발법과 함께 통합하여 사용해야 한다.

3. 통합 개발법

전통적 SDLC 법은 기업의 전체에 영향을 미치는 대규모의 시스템을 신규로 개발하는 데 유효한 개발법이며, 동시에 개발에 필요한 예산과 시간에 여유가 있으면서 사용자의 시스템에 대한 요구와 시스템정보에 대한 요구에 불확실성이 없으며, 또한 시스템 개발의 분석 단계에서 사용자의 시스템요구가 확정적이어서 개발기간 또는 그 이후 상당기간동안 변할 가능성이 적은 경우에 활용할 수 있는 정보시스템 개발법인 것이다. 따라서 이러한 조건들을 만족시키지 못하는 경우에 전통적 SDLC 개발법을 사용하는 것은 바람직하지 않다.

원형개발법은 정보시스템의 개발과정 가운데 두번째 단계인 분석단계에서 개발할 새로운 시스템에 대한 시스템 사용자의 요구사항과 새로운 시스템에서 산출할 정보에 대한 사용자의 요구사항이 확정적이 아니고 가변적이어서 개발과정에서 직접 사용자를 참여시켜 그들이 진정으로 필요로 하는 시스템과 시스템 산출정보가 무엇인가를 확실히 해주기 위한 방법으로 고안해낸 개발법이기 때문에 원형개발법만을 독자적으로 이용하여 시스템을 개발하기보다는 전통적 SDLC 개발법과 함께 사용할 때 그 효과를 극대화할 수 있는 것이다.

따라서 사용자의 시스템에 대한 요구와 시스템정보에 대한 요구가 불확실할 경우에는 전통적 SDLC 개발법의 분석단계에서 시스템 사용자가 원하는 시스템 요구사항(system requirements)과 시스템 정보요구사항(information requirements)을 확실히 하기 위해 시스템원형을 설계하여 이를 모의 실행시켜 사용자의 요구를 충족시키지 못하면 이를 계속 수정시켜 시스템 개발자로 하여금 개발할 시스템에 대한 확실한 청사진을 가지게 함으로써, 나중에 시스템개발을 완성하여 실행할 때 사용자의 새로운 시스템에 대한 만족을 충분히 기대할 수가 있는 것이다.

III. 회계정보시스템의 개발

회계정보시스템은 처리 목적, 그리고 시스템에 입력되는 자료와 산출되는 정보의 성격에

따라 자료처리시스템(data processing system) 또는 거래처리시스템(transaction processing system)과 정보처리시스템(information processing system)으로 나누어진다.

1. 자료처리시스템

(1) 의의

회계정보시스템의 여러 목적중의 가장 기본적인 것이 기업의 매일 매일의 일상 업무가 원활히 이루어지도록 지원하는 일이다. 이는 기업에서 매일 발생하는 거래를 기록하고 처리하고 그 결과인 각종 문서와 재무제표 등 회계보고서를 작성하는 일이다. 이러한 일련의 연속된 작업을 자료처리 또는 거래처리라고 하며 거래의 처리 과정은 기업회계기준과 각 기업의 회계처리 방침에 근거하여 표준화된 과정을 거쳐 이루어진다.

자료처리시스템은 복식부기상의 거래자료를 자료원(data sources)으로 활용하여 거래를 신속 정확하게 효율적으로 처리하는 것을 지향하고 있으며 거래는 기업활동 가운데 자산, 부채, 자본의 변동, 그리고 수익과 비용을 발생시키는 활동으로서 그 활동의 성격에 따라 거래는 영업활동거래, 재무활동거래, 그리고 투자활동거래로 구분할 수 있다. 이러한 성격을 갖춘 자료처리시스템은 상품 등의 재화와 용역의 판매, 설비자산의 취득과 용역의 구입, 현금의 관리, 임금의 지급, 신용의 결정 등과 같은 기본적인 기업의 활동과 거래를 추적, 처리함으로써 기업의 활동을 지원할 목적으로 운영된다.

(2) 자료처리 하위시스템

자료처리시스템은 먼저 자료처리과정에 컴퓨터를 활용하느냐 여부에 따라 전통적인 수작업 자료처리시스템과 전산 자료처리시스템으로 나누어진다. 또한 거래의 성격에 따라 수익거래, 지출거래, 자원거래, 그리고 결산거래로 세분하여 각 거래별 자료처리 시스템을 운영하고 있다.

수작업 처리시스템에서의 거래처리과정은 거래의 식별에서부터 분개, 총계정원장에의 전기, 수정전시산표의 작성, 수정분개, 수정후시산표의 작성, 재무제표와 기타 회계보고서의 작성의 순서로 이어지는데, 이 모든 작업이 각종 장부작성을 통해 수작업으로 이루어진다.

전산 자료처리시스템에서는 거래자료만 정확히 입력해주면 모든 거래처리과정이 자동적으로 처리되므로 어떠한 처리과정이 수행되었는가를 정확하게 알지 못해도 자료처리가 가능하다. 즉, 일상적인 거래내용과 연말의 수정분개 사항을 정확하게 이해하여 전산자료

회계정보시스템 개발에 관한 연구

시스템의 입력화면에 입력하는 방법과, 출력물인 재무제표나 기타 각종 장부와 보고서를 조회하거나 해석하는 방법만 학습하면 사용할 수 있는 것이다.

전산 자료처리시스템의 자료처리과정은 크게 다음의 세 가지 과정을 거쳐 이루어진다. 첫 단계는 원시문서의 준비와 거래데이터의 입력인데, 발생된 경제적 사건이 회계상의 거래인지를 식별하는 것으로 시작한다. 회계상의 거래로 판정되면 전표나 기타 원시문서를 작성한 후, 이를 기초로 적절한 분개과정을 거쳐 전산 자료처리시스템의 입력양식에 입력하여야 한다. 이때 입력과정이 자동화되어 있으면 데이터의 관찰, 입력 및 전환이 모두 동시에 이루어진다. 두번째는 거래데이터의 처리이다. 이는 입력된 거래데이터의 처리와 마스터 자료파일을 갱신하는 과정으로 전산 자료처리시스템에 입력된 데이터에 의해 거래자료파일을 생성하며, 각 거래파일에 새로운 자료가 추가되거나 변경이 이루어지면, 즉 시 또는 일괄적으로 자료처리 프로그램에 의해 자동적으로 기존의 마스터 자료파일을 새로운 마스터 자료파일로 갱신한다. 세번째 단계는 회계정보의 출력과 조회이다. 이용자들은 필요에 따라 수시로 회계정보를 조회하거나 인쇄매체나 기타 출력매체에 출력할 수 있다. 이 단계의 주요 출력물에는 각종 전표 등 주요장부와 문서, 재무제표와 보조장부, 조회응답정보, 거래일지 등이 있다.

수익거래 자료처리시스템(revenue transaction data processing system)은 재화나 서비스의 판매로부터 현금수취에 이르기까지의 전과정에서 발생한 모든 수익거래를 처리해주는 시스템을 말한다. 이 시스템의 목적은 신속한 현금수취를 위하여 제품이나 용역을 고객과 교환하는 것을 촉진하는 것이다.

지출거래 자료처리시스템(expenditure transaction data processing system)은 기업이 재화와 서비스의 공급을 얻기 위해 발생한 모든 지출거래를 처리해주는 시스템을 말한다. 이 시스템의 목적은 필요한 재화와 용역을 취득하기 위해서 공급자와의 현금교환을 촉진하는 것이다.

기업의 자원은 인적자원과 설비와 원재료자원, 그리고 자금으로 구분된다. 자원거래 자료처리시스템(resources transaction data processing system)은 기업이 보유하고 있는 각종 자원의 취득과 유지 및 처분과 관련된 거래자료를 정보화기술을 활용하여 자동적으로 처리하는 시스템이다. 이 시스템에는 종업원용역관리처리, 설비관리처리, 원재료관리처리, 그리고 현금관리처리가 포함된다. 종업원용역관리 처리시스템(employee services management processing system)은 종업원들이 제공하는 용역의 대가로 받게 될 급여지급액을 산정하고 급여거래와 관련된 회계기록을 유지하기 위한 시스템이다. 설비관리 처리시스템(facility management processing system)은 설비자산의 취득, 재평가, 처분 및 상각 등과 관련된 설비자산거래의 변동사항을 처리하고 관련된 회

계기록을 유지하기 위한 시스템이다. 원재료관리처리시스템은 효율적인 생산과 재고통제 활동을 수행하고, 제품생산에 소요된 제 원가를 추적 집계하여 정보이용자들이 필요로 하는 원가정보를 산출해주는 시스템이다. 현금관리 처리시스템은 효율적인 현금관리를 위해 운영되는 시스템이다.

결산거래 자료처리시스템은 수익거래 자료처리시스템, 지출거래 자료처리시스템, 및 자원거래 자료처리시스템을 통합한 시스템으로 각 거래 자료처리시스템의 처리결과를 기초로 하여 총계정원장 처리시스템을 생성하고, 이로부터 최종적으로 각종 재무보고서를 출력한다.

2. 정보처리시스템

(1) 의의

회계정보시스템 개발의 초기단계에서는 앞서 설명한 자료처리시스템의 구축에 초점이 모아졌다. 자료처리시스템이 어느 정도 정착되고 정보화가 고도화됨에 따라 회계정보시스템은 자료처리의 역할을 넘어서 회계정보를 이용하는 내부 경영자와 외부 투자자 등의 의사결정을 지원하기 위하여 자료처리시스템에서 산출되는 회계자료와 여기에 다른 자료를 추가하여 재처리하는 정보처리의 역할을 수행하게 되었다. 정보처리시스템은 최고경영자의 전략적 의사결정, 중간관리자의 관리적 의사결정, 하급관리자의 일상적 의사결정 등 다양한 수준의 의사결정을 지원하게 된다.

자료처리시스템이 회계정보시스템의 거래처리의 측면을 강조한 것이라면 정보처리시스템은 회계정보시스템의 정보이용자들의 의사결정에 유용한 회계정보의 처리와 생산을 중점적으로 다루게 된다.

자료처리시스템의 절차 및 구조가 표준화되어 있는 반면, 정보처리시스템은 이용자의 직급, 요구사항, 조직구조의 유형 등에 따라 다양한 형태를 가지게 된다. 자료처리시스템이 기업회계기준과 각 기업의 회계처리지침에 따라 설치하여 운영되기 때문에 자료처리시스템에 대한 시스템요구나 시스템정보요구가 어느 정도 예측이 가능하고 확정적이기 때문에 시스템사용자와 시스템개발자간의 의사소통이 비교적 간단히 이루어질 수 있다. 반면에 정보처리시스템은 각 기업의 규모와 경영자의 직급에 따라 그리고 의사결정의 성격과 시기에 따라 다양하고 자주 변할 수 있는 비확정적인 확률적 정보를 제공하게 된다. 따라서 시스템 사용자와 시스템 개발전문가간에 시스템요구와 시스템정보 요구에 대한 지속적인 대화와 의사소통이 개발기간 동안에 필요한 것이다.

(2) 정보처리 하위시스템

정보처리시스템은 경영자들의 계획과 통제활동을 지원하는 역할을 맡고 있는 회계정보 시스템인 것이다. 경영자의 계획활동을 지원하기 위한 대표적인 회계정보가 단기적 성격을 가진 종합예산(master budget)정보이며 장기적 성격을 띤 것이 자본예산(capital budget) 정보인 것이다. 한편, 통제활동을 효과적으로 수행하는데 도움을 주는 회계정보가 업적평가(performance evaluation)에 관련된 정보인 것이다. 정보처리의 하위 시스템에는 종합예산정보를 처리하는 종합예산 정보처리시스템, 자본예산정보를 처리하는 자본예산 정보처리시스템, 그리고 업적평가정보를 처리하는 업적평가 정보처리시스템이 포함된다.

종합예산 정보처리시스템은 일반적으로 1년을 단위로 작성되는 단기성 예산으로 기업 전체를 대상으로 하는 정보처리시스템의 하나로, 다음 회계기간 동안에 예상되는 기업활동을 계량화한 경영계획보고서인 것이다. 종합예산처리시스템은 크게 영업예산처리시스템과 재무예산처리시스템의 두 부분으로 구성되어 있는데 영업예산처리시스템을 통해 최종적으로 산출되는 예산이 손익계산서예산이며, 이를 위해 제조기업의 경우, 판매예산, 생산량예산, 제조원가예산, 매출원가예산, 그리고 판매비와 관리비예산의 작성이 선행되어야 한다.

판매예산(sales budget)은 영업예산작성의 출발점으로 판매예측에 따라 예상판매수량과 예상판매가격에 의해 작성된다. 생산량예산은 예상판매수량과 예상기말재고량에 근거하여 작성된다. 제조원가예산은 생산량예산과 직접재료와 직접노무예산 및 간접제조원가예산에 근거하여 작성된다. 매출원가예산은 예산기간 중 생산한 제품 가운데 판매될 것으로 예상되는 제품의 원가로 기말재고예산과 제조원가예산에 근거하여 작성한다. 판매비와 관리비예산은 제조활동이외의 판매활동과 관리활동에서 발생할 것으로 예상되는 비용이다. 손익계산서예산은 예산기간 동안의 경영성과를 나타내는 예산으로 수익과 비용 및 이익 등에 관한 예측정보이며 판매예산과 매출원가예산, 그리고 판매비와 일반관리비 예산을 기초로 하여 작성된다.

한편 재무예산처리시스템의 최종 출력보고서는 대차대조표예산과 현금흐름표예산이다. 이를 위해 손익계산서예산과 현금예산 그리고 다음에 설명할 자본예산이 먼저 작성되어야 하며 아울러 기초 대차대조표를 기초로 하여 대차대조표예산이 최종적으로 작성된다. 현금예산은 현금의 수입액과 지출액에 관한 예산으로 불필요한 유휴현금과 현금부족을 사전에 방지하도록 하여준다. 대차대조표예산(balance sheet budget)은 예산기간 말의 기업의 재무상태를 나타내며 기초의 대차대조표를 출발점으로 그 기간동안의 영업활동과 투자와 재무활동의 결과를 반영시킨다.

고정자산에 대한 자본투자를 효율적으로 수행하기 위해 투자대상을 탐색하고 평가하는 일련의 체계적인 과정을 자본예산(capital budgeting) 정보처리시스템이라 한다. 즉, 자본예산 정보처리시스템은 자금이 제한되어 있고, 여러 투자대상이 존재하는 상황에서 기업의 수익성을 극대화시켜 주는 투자계획안을 심사하여 선택하는 일련의 체계적인 정보처리시스템을 말한다.

자본투자의 규모와 내용은 낡은 기계의 대체와 같은 일상적인 것으로부터 새로운 공장의 건설이나 신제품의 개발에 이르기까지 다양하다. 이러한 자본투자를 효율적으로 수행하기 위해서는 자본예산의 편성이 필요하다.

자본예산 정보처리시스템은 일반적으로 다음의 네 단계를 거쳐 완성된다. 먼저, 자본예산 계획의 수립이다. 기업의 각 부서로부터 필요한 자본투자 계획안이 마련된다. 둘째는 각 투자계획안의 평가방법의 설정이다. 이를 위해서는 먼저 자본투자계획안 별로 현금흐름의 예측이 이루어져야하며 아울러 여러 투자평가방법간의 비교와 분석이 이루어진다. 셋째는 자본예산 심의위원회의 각 투자 계획안의 평가와 승인이다. 평가결과 작성되는 평가보고서에는 제안된 각 투자계획안의 지출 소요액과 효과, 각 평가방법에 따른 각종 평가 측정치와 우선 순위, 그 밖의 각종 관련 정보가 포함된다. 넷째는 승인된 투자계획안의 실행과 사후관리이다.

경영자의 통제활동을 지원하기 위해 개발한 정보처리시스템이 업적평가 정보처리시스템이다. 각 부서의 관리자는 일정기간(1개월, 1분기, 6개월 또는 1년) 말에 부서별 영업 활동에 대한 결과를 업적평가 보고서에 담아 제출하게 된다. 업적평가 보고서에는 책정된 예산과 실제의 차이를 나타내도록 하여 업적평가가 쉽게 이루어지도록 해야한다. 각 책임부서에서 작성된 업적평가보고서는 상위 경영계층으로 종합되어 간다. 기업조직의 목표를 효율적으로 달성하기 위하여 기업조직안에 다양한 책임중심점을 설정하고 특정한 수익과 비용에 책임이 있는 각 책임중심점별로 특정기간내의 실제로 발생한 실적을 측정하여 계획의 계량화된 값인 예산과 비교하여 그 차이를 분석하는 정보처리시스템이다.

업적평가 정보처리시스템의 출력보고서는 기업의 조직상의 계층을 따라 각 책임부서별로 작성 보고되는데 책임부서에서 책임을 질 수 있는 부분 즉, 비용의 경우 통제가 가능한 비용에 대해서만 평가를 하고 통제불가능한 비용에 대해서는 평가대상에서 제외시켜야公正한 평가가 이루어진다. 통제불가능한 비용에는 본사에서 각 부서에 할당한 본사경비 등의 간접비와 재산세나 감가상각비와 같은 고정자산에 대한 비용 및 각 부서 책임자의 급여 등이 포함된다. 그러나 하위부서에서는 통제가 불가능한 비용일지라도 상위부서로 올라갈수록 통제가 가능한 비용이 될 수 있으므로 발생한 모든 비용을 통제가능성 여부에 따라 구분해 둘 필요가 있다.

회계정보시스템 개발에 관한 연구

일반적으로 예산과 실적의 차이의 크기에 상관없이 모두 포함시켜 업적평가 정보처리 시스템의 출력보고서를 작성할 경우, 기업의 규모가 클수록 경영책임자가 검토할 보고의 내용의 방대하게 된다. 따라서 예산과 실적의 차이가 사전에 정해놓은 기준값보다 큰 경우에 한하여 그 내용과 그 차이가 발생한 이유 등을 나타내는 출력보고서를 작성케 함으로써 책임경영자로 하여금 불필요한 정보의 흥수로부터 해방시킬 수가 있는데 이를 예외에의한관리 정보처리시스템이라고 한다.

3. 자료처리시스템의 개발

자료처리시스템은 정보처리시스템과는 달리 처리과정이 표준화되어있고 자료처리시스템의 사용자들이 시스템정보에 대한 요구가 비교적 확정적이기 때문에 개발과정에서의 사용자요구의 변경과 시스템설계의 수정은 거의 발생하지 않기 때문에 시스템분석 단계에서의 시스템원형을 개발할 필요성이 적다고 하겠다. 그러나 자료처리시스템을 신규로 개발하는 것이 아니라 기존 시스템의 일부를 변경시키는 경우 전통적인 SDLC 개발법을 그대로 적용하는 것은 비효율적이라고 본다. 따라서 신규시스템을 개발하는 경우와 기존 시스템의 일부 변경을 위한 시스템개발을 할 경우를 각각 구분하여 각기 다른 개발법을 적용해야 개발의 효과를 극대화할 수 있는 것이다.

(1) 신규로 기업 전체의 자료처리시스템을 개발하는 경우

예를 들면, 수작업으로 기업의 거래자료를 처리하던 기업이 컴퓨터를 도입하여 자료처리를 전산화할 의도로 자료처리시스템을 새로이 개발하는 경우라든가, 또는 이미 전산자료처리 시스템을 운용하고 있지만 현재는 전산일괄처리시스템(computerized batch processing system)으로 인한 불편함으로 인해 새롭게 전산온라인처리시스템(computerized on-line processing system)으로 시스템전체를 변경하려고 하는 경우가 이에 해당된다. 이 경우 전통적인 SDLC 개발법을 활용하여 개발하는 것이 효과적이라고 본다.

이 같이 전체시스템을 개발하는 경우, 한번 개발하여 설치 운영하면 핵심적인 시스템의 주요부분은 오랜 기간 계속 사용하면서 필요에 따라 시스템변경을 부분적으로 하는 경우로서 최초로 개발할 때 시간이 걸리고 개발비용이 많이 소요되더라도 모든 개발에 필요한 사항을 빠짐없이 고려하는 전통적 SDLC 개발법을 활용하는 것이 바람직하다. 또한 이 경우 사용자의 시스템에 대한 요구나 시스템을 통해 산출되는 정보에 대한 요구가 비교적 확실하고 개발이 진행되는 기간동안은 물론, 개발 후 오랜 기간 변동이 발생하지 않을 것으로 예상되므로 시스템원형을 설계하고 수정할 필요가 없는 전통적 SDLC

개발법을 사용하는 것이 효율적이라고 본다.

(2) 기존 시스템의 부분적인 자료처리시스템을 개발하는 경우

지금 현재 운영하고 있는 자료처리시스템의 일부를 변경할 필요성이 발생하여 기존 시스템의 대부분은 그대로 유지하면서 일부 시스템을 변경시키는 경우를 말한다. 예를 들면, 기업회계기준의 개정으로 인하여 회계처리 프로그램의 변경을 해야되는 경우라든가, 또는 기업내의 여러 부서간의 원활한 정보교환을 위한 데이터베이스의 구축과 정보통신망을 추가로 구축하는 경우 등이 이에 해당된다. 이 경우 전통적 SDLC 개발법 보다는 수정된 SDLC 개발법을 활용하여 시스템을 개발하는 것이 보다 효과적이다.

수정 SDLC 개발법을 이용하여 회계처리시스템 변경을 위한 개발과정을 살펴보면 다음과 같다. 첫 단계는 프로그램 개발자와 회계담당 부서의 책임자는 개정된 기업회계기준과 기존의 회계처리프로그램에서 이제까지 사용한 폐기할 기업회계기준과를 비교하여 개정된 부분에 대하여 사용자로부터의 처리프로그램에 대한 시스템변경에 대한 요구를 수용하여 개발자는 새로운 처리프로그램을 설계하여 사용자로 하여금 개정된 처리기준과 관련된 테스트용 거래데이터를 모의실험하게 하여 예상된 결과를 얻게 되면 시스템 개발이 종료된다. 이와 같이 시스템의 일부만을 변경하여 개발하는 경우, 전통적 SDLC 개발법에서 운영했던 개발운영위원회의 역할을 사용자가 대신 맡아 예상된 결과가 나올 때 까지 프로그램설계를 재 작업하도록 한다.

따라서 이 경우 전통적 SDLC 개발법과는 달리 첫 개발단계인 조사와 계획은 생략할 수가 있으며, 분석과 설계작업을 거의 동시에 하여 개발시간을 단축할 수 있으며, 동시에 사용자가 만족하는 정보를 제공할 수 있는 것이다.

4. 정보처리시스템의 개발

정보처리시스템은 자료처리시스템과는 달리 사용자의 정보요구가 기업마다 다양하고 경영자의 직급에 따라 요구하는 정보의 내용이 다르고 시간이 경과함에 따라 변하는 가변적인 특성을 가지고 있기 때문에 시스템을 개발하는 과정에서 시스템 개발자가 사용자의 요구를 시스템 개발에 정확히 반영시키기가 매우 어렵다고 본다. 정보처리시스템도 자료처리시스템과 마찬가지로 전체시스템을 재발하느냐 아니면 일부 시스템을 변경하느냐의 두 가지 경우로 구분하여 논의하고자 한다.

(1) 신규로 기업 전체 정보처리시스템의 개발

회계정보시스템 개발에 관한 연구

예를 들면, 전산업적평가 정보처리시스템을 신규로 개발하여 도입한다든지, 또는 기업의 예산정보처리시스템을 전산화한다든지 하는 경우가 이에 해당된다.

이 경우 전통적인 SDLC 법과 원형개발법을 통합한 통합개발법을 사용하는 것이 효과적이다. 개발의 분석단계에서 사용자의 새로운 시스템요구와 새로운 정보요구를 확정하기 위해 시스템원형을 설계하여 이를 사용자가 모의 실험을 통해 시스템원형을 수정시켜 최종의 시스템원형을 확정시킴으로 해서 사용자의 요구사항을 확실히 한 후, 다음 개발단계인 설계단계로 이어진다.

이 개발과정을 도표로 표시하면 <도표 2>와 같다.

<도표 2> 신규 정보처리시스템의 개발과정

```
정보처리시스템의 계획 --> 정보처리시스템의 분석(최초 시스템원형의 설계) --> 사용자의 모의 실험과 수정 --> 최종 시스템원형의 확정 --> 정보처리시스템의 설계  
--> 정보처리시스템의 실행
```

(2) 기존 정보처리시스템의 일부를 수정하기 위한 개발

예를 들면, 기존의 업적평가 정보처리시스템 자체 내에 예외에 의한 관리정보처리시스템을 설치하여 운영한다든지, 또는 예산정보처리시스템 내에 판매예측 정보처리시스템을 추가 설치하는 경우가 이에 해당된다 하겠다. 이 경우 자료처리시스템과는 달리 사용자의 시스템에 대한 요구와 정보요구가 변동적이고 불확실하여 시스템을 개발하는 기간동안에 도 변할 가능성이 있고 또한 비교적 빠른 시간 내에 변경된 시스템의 사용이 이루어져야 하기 때문에 전통적 SDLC 법 보다는 수정 SDLC 개발법과 원형개발법을 통합한 수정 통합개발법을 사용하는 것이 효과적이다.

예외에 의한 관리 정보처리시스템을 기존의 업적평가 정보처리시스템에 추가하여 변경시키는 경우를 살펴보면, 먼저 이 정보처리시스템의 사용자인 중간관리경영자 등으로부터 예외에 해당되는 기준값을 제공받아 시스템원형을 설계한 후 이를 시스템 사용자가 테스트한 후 예상된 결과가 나오지 않으면 새로운 기준값을 적용하여 사용자가 만족할 때까지 새로운 시스템원형을 수정하여 이를 설계하여 실행시킨다.

이 개발과정을 일반화하여 도표로 표시하면 <도표 3>과 같다.

<도표 3> 기존 일부 정보처리시스템의 개발과정

```
일부 시스템변경의 계획 --> 분석 및 설계(최초시스템원형의 설계) --> 사용자의 모의 실험과 수정 --> 최종시스템원형의 확정 --> 일부 시스템변경의 실행
```

IV. 결 론

개발할 정보시스템의 규모, 정보시스템의 변경의 정도, 그리고 사용자요구의 확실성의 정도 등에 따라 서로 다른 개발법이 회계정보시스템의 개발에 이용되어야 한다.

이 논문에서 제시하고 있는 통합개발법은 전통적 SDLC 개발법의 두번째 단계인 분석 단계에서 시스템 사용자의 시스템요구와 시스템정보요구를 확정지기 위해 시스템원형과 이의 모의실험과 시스템원형의 수정의 과정을 추가한 개발법이다. 이 통합개발법은 특히 정보처리시스템과 같은 시스템 사용자의 정보요구에 대한 불확실성이 존재할 때 사용하면 그 개발효과가 크다고 하겠다.

참 고 문 헌

1. Alavi M, "An Assessment of the Prototyping Approach to Information Systems Development," Communications of the ACM, Vol. 27, No.6, 1984, pp. 556-563.
2. Baskerville Richard, "Controlling Prototype Development through Risk Analysis," MIS Quarterly, Dec. 1996, pp. 481-501.
3. Berrisford T. R. and Wetherbe J. C, "Heuristic Development: A Redesign of Systems Design," MIS Quarterly, 1979, pp. 1-19.
4. Bodnar George & Hopwood William, Accounting Information Systems, 7th ed., 1998, Ch. 9.
5. Cerullo Michael, "Designing Accounting Information Systems," Management Accounting, June 1985, pp. 37-42.
6. Dean Robert & Johnson Dale, "Designing an Information System for a Health Care Organization," Journal of Systems Management, Jan. 1991, pp. 23-31.
7. Doll W. and Torkzadeh G, "A Discrepancy Model of End-User Computing Involvement," Management Science, Oct. 1989, pp. 1151 -1171.
8. Gavurin Stuart, "Where Does Prototyping Fit in IS Development?"

회계정보시스템 개발에 관한 연구

- Journal of Systems Management, Feb. 1991, pp. 13-17.
- 9. Harel E. C. & Mclean E. R, "The Effect of Using a Nonprocedural Language on Programmer Productivity," MIS Quarterly, Vol. 9, No. 2, 1985, pp. 109-120.
 - 10. Lin C. Y, "Systems Development with Application Generators: an End User Perspective," Journal of Systems Management, April 1990, pp. 32-36.
 - 11. Mahmood M. A, "System Development Method-A Comparative Investigation," MIS Quarterly, 1987, pp.293-311.
 - 12. Moscove Stephen & Others, Accounting Information Systems, 4th ed., 1990, Chs. 13-15.
 - 13. Romney Marshall b, Steinart Paul J, and E. Cushing, Accounting Information Systems, Addison Wesley, 7th ed., 1997, pp. 407-413.
 - 14. Wilkinson Joseph & Cerullo Michael, Accounting Information Systems, 3rd ed., 1997, Ch. 18.

