

會計情報시스템 開發 接近方法에 관한 研究

A Study on the Accounting Information System Development Approaches

田 永 昇*

(Cheon Young Seung)

目 次

- | | |
|-------------------------------|--------------------------|
| I. 序 論 | 2. 原型開發接近法の 模型 |
| II. 會計情報시스템의 本質 | 3. 原型開發接近法에 要求되는 資
源 |
| 1. 會計情報시스템의 概念 | 4. 세 가지 構成要素의 役割 |
| 2. 會計情報시스템의 利用者 | V. 兩 接近法の 比較 |
| 3. 會計情報 시스템과 經營情報
시스템의 관계 | 1. 概 念 |
| III. 시스템 開發 壽命週期接近法 | 2. 特 性 |
| 1. 시스템 開發 壽命週期接近法の
意義와 模型 | 3. 模 型 |
| 2. 定義段階 | 4. 開發週期 各 段階에 미치는 效
果 |
| 3. 開發段階 | 5. 有用성과 限界 |
| 4. 設置와 運營段階 | VI. 結 論 |
| IV. 原型開發接近法 | 參考文獻 |
| 1. 原型開發接近法の 意義 | |

I. 序 論

會計는 環境의 산물이다. 會計는 社會, 經濟的 環境속에서 생성하고 성장하는 것이다. 오늘날 會計環境의 대표적인 변화는 컴퓨터 이용에 의한 情報處理能力의 증대와 情報通信技術의 발전에 따라 會計情報가 時, 空을 초월하여 전달되는 社會가 되었다는 것이다. 이른바 情報化社會를 맞이하고 있다. 따라서 현대 會計의 概念은 去來事實을 단순히 記錄, 分類, 要約하고 이를 解釋하는 技術로서가 아니라 會計情報利用者들에게 그들이 合理的 判斷과 經濟的 意思決定을 할 수 있도록 經濟實體에 관한 財務的 情報를 識別, 測定, 傳達하는 일련의 過程으로 확대 되었다. 즉, 會計를 하나의 情報시스템으로 파악하게 된 것이다.

성공적인 情報시스템의 구축은 시스템의 開發에 달려있다. 情報시스템의 開發이란 使用者 및 시스템 分析家들이 어떤 組織의 情報要求事項(information requirements)을 분석한 것을 토대로 하여 情報시스템을 設計하는 과정을 말한다.

情報시스템의 傳統的 開發接近法은 시스템 開發 壽命週期接近法(System Development Life cycle : SDLC)이다. 그러나 이 방법은 業務가 複雜해지고 開發過程의 不確實性이 증대되는 狀況에선 부적합하다. 더구나 使用者 滿足度を 이용하여 情報시스템의 成果를 측정하는 경우에는 使用者의 요구를 정확하고 완전하게 충족시켜 주는 것이 급선무인데 이 傳統的 방법은 使用者 要求를 충족시키기 어렵다. 이러한 問題點을 補完, 情報시스템 開發에 使用者를 參與시킬 수 있는 한 방법으로 등장한 것이 原型開發接近法(Prototyping)이다.

이러한 認識을 바탕으로 본 研究의 目的은 보다 效率的인 會計情報시스템 開發에 一助를 加하는 데 있다. 이와 같은 目的達成을 위하여 본 研究는 광범위한 文獻調査를 통한 理論的 研究方法에 의거하여 전개되었다. 研究內容은 序論에 이어 II장에서 會計情報시스템의 本質을 살펴보고 II장에서 시스템 開發壽命週期接近法, IV장에서 原型開發接近法의 내용을 고찰하고 V장에서 兩 接近法을 比較分析하였으며 VI장에서 結論을 맺는다.

II. 會計情報시스템의 本質

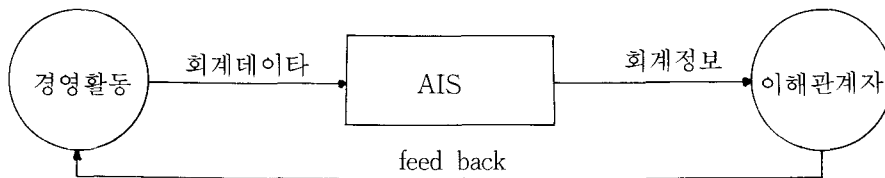
1. 會計情報시스템의 概念

會計情報시스템은 가장 오래 되었고 또 가장 광범하게 사용되는 企業情報시스템이다. 會計情報시스템(Accounting Information System : AIS)의 정의는 다양하다. 美

國會計學會의 견해에 의하면 “企業의 利益, 富 기타 經濟的 事상을 測定 또는 豫測하기 위한 公式情報시스템”이라 했다.¹⁾ 쿠싱(Cushing)은 “財務的 情報과 去來資料의 蒐集 또는 처리로부터 얻어지는 情報의 산출에 대하여 책임있는 組織內의 모든 人的 資本的 資源의 接合”이라고 정의하였다.²⁾ 모스코프(Moscove)와 심킨(Simkin)은 “企業內의 利害關係者에게 적절한 財務指向的인 意思決定 情報를 蒐集, 分類, 處理, 分析, 傳達하는 組織上의 구성요소다.”라고 하였고³⁾ 우(Wu)는 “情報利用者에게 成果記錄, 注意喚起 및 意思決定에 有用한 情報를 제공하기 위하여 財務的 去來를 처리하는 組織體의 한 構成要素 또는 實體”라고 정의 하고 있다.⁴⁾

위와 같은 정의 들을 종합해 볼 때 會計情報시스템은 목표를 달성하기 위해 유기적으로 결합된 결합체 이며 會計去來 資料를 蒐集(入力), 處理하여 會計情報를 生成(出力)하고 이를 利害關係者에게 그들이 意思決定 하는데 유용하도록 신속, 원활하게 전달하는 시스템이라 할 수 있다. (그림 1 참조) 또한 人間, 資料, 소프트웨어, 하드웨어로 구성된 開放시스템이다.

〈그림 1〉 會計情報시스템



2. 會計情報시스템 利用者⁵⁾

어떠한 종류의 會計實體든 간에 會計情報시스템의 역할이 매우 중요하며 내외적 環境에 큰 영향을 미친다. 또한 環境의 변화도 會計情報시스템의 성격에 영향을 미치게 된다. 내적환경이란 會計實體 내부의 利害關係者 즉, 經營者 내지 管理者를 구

註 1) AAA, "Report of the Committee on Accounting and Information Systems", The Accounting Review, Vol.46, 1971, p.288.

註 2) Barry E.Cushing, Accounting Information Systems and Business Orgization, 2nd ed., Addison Wesley Publishing Co., 1987, p.9.

註 3) Stephen A.Moscove, Mark G. Simkin, Accounting Information System : Concepts and Practice for Effective Decision Making, New York: John Wiley and Sons, Inc., 1982, p.6.

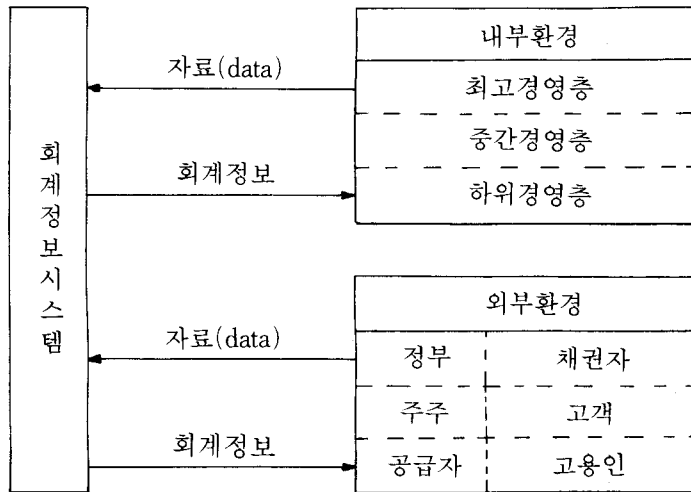
註 4) F.H. Wu, Accounting Information Systems: Theory and Practice, McGraw-Hill Inc., 1983, pp.6-8.

註 5) 金永孝, 會計情報시스템, 改訂增補 2版, 世明書館, 1990. pp.52-56.

성요소로 하는 會計實體의 組織을 의미하며, 외적환경이란 주로 政府, 債權者, 顧客, 株主, 供給者, 從業員 등을 말한다.

이러한 내, 외적 環境과 會計情報시스템과의 관계를 그림으로 나타내면 다음과 같다. (그림 2참조)

〈그림 2〉 會計情報시스템과 環境



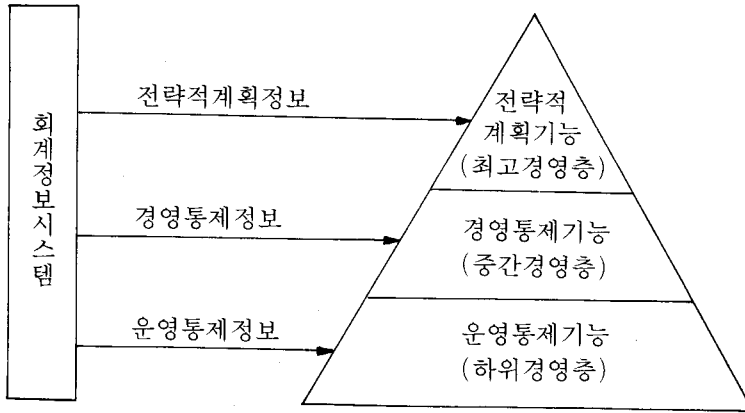
會計情報시스템의 내부이용자는 일반적으로 最高經營層, 中間經營層, 下位經營層으로 분류된다. 최고경영층은 戰略的 計劃機能을 담당하고, 중간경영층은 經營統制機能을 담당하며, 하위경영층은 주로 實務的 차원에서 運營統制機能을 담당한다.

따라서 최고경영층에 제공되는 정보는 戰略的 計劃情報로서 總括的이며 比較分析的 情報의 특성을 갖는다. 중간경영층에 대한 정보는 經營統制를 위한 短期的 情報로서 分析, 評價의 성격을 갖는다. 하위경영층에 대한 정보는 분기, 월간회계정보로서 현행 營業活動 結果의 집계정보가 주종을 이룬다.(그림 3참조)

3. 會計情報시스템과 經營情報시스템의 관계

經營情報시스템(Management Information System : MIS)에 대하여도 많은 정의가 내려져 있으나 포괄적으로 정의하면 組織의 計劃, 統制 및 運營을 위한 情報를 蒐集, 貯藏, 檢索, 處理하여 적절한 시기에 적절한 형태로 적절한 구성원에게 제공해

〈그림 3〉 會計情報시스템과 内部利用者



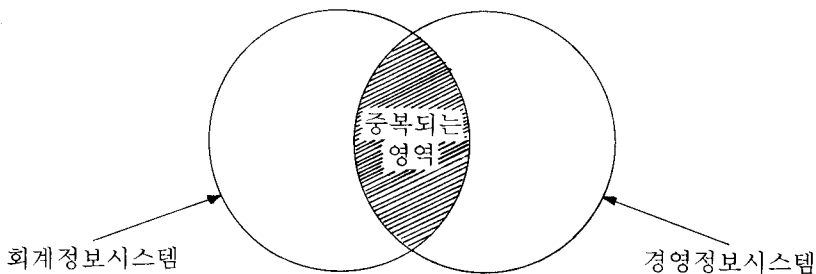
중으로써 組織의 目標을 보다 效率的 및 效果的으로 달성할 수 있도록 組織化된 統合的 人間-機械시스템이라 할 수 있다.⁶⁾

會計情報시스템과 經營情報시스템 사이에 그 範圍와 關係를 두고 상당한 논쟁이 벌어지고 있다. 이를 간단히 요약하면 다음과 같다.⁷⁾

- 1) 會計情報시스템은 經營情報시스템의 下位시스템이라는 견해
- 2) 經營情報시스템이 會計情報시스템의 下位시스템이라는 견해
- 3) 會計情報시스템과 經營情報시스템은 그 범위에 있어서 중복되는 부분이 많지만 각각 獨立된 시스템이라는 견해

本 研究는 위 세번째 견해에 입각하여 논의를 전개한다. 즉, 두 시스템 모두 경영진으로 하여금 計劃과 統制의 機能을 원활히 수행할 수 있도록 情報를 제공한다. 그러나 經營情報시스템은 단지 經營者들 만을 위한 情報시스템인 반면에 會計情報시스템은 단지 會計資料를 통해 情報를 제공하는 시스템이다. 그러므로 두 시스템이 독특한 영역도 있으나 많은 영역이 중복되는 것을 알 수 있다.⁸⁾(그림 4참조)

〈그림 4〉 會計情報시스템과 經營情報시스템의 領域



Ⅲ. 시스템 開發 壽命週期接近法

1. 시스템 開發 壽命週期接近法の 意義와 模型

情報시스템 開發壽命週期란 情報시스템을 開發하는데 필요한 過程 또는 節次를 의미한다. 그러므로 壽命週기에 있는 어떠한 段階라도 소홀히 하는 것은 시스템 開發을 실패로 이끄는 원인이 될 수 있다.

시스템 開發 壽命週期接近法이란 시스템 開發過程을 논리적으로 段階化하여 매 段階마다 산출되는 中間산물을 점검 확인하면서 시스템을 開發하는 기법이다. 이 방법의 기본적인 假定은 시스템이 構想되고 開發되며 實行되는 과정이 잘 정의되어 있다는 것이다. 壽命週期の 各 段階들은 管理와 統制를 위한 기초를 제공해 준다. 왜냐하면 各 단계들은 管理目的을 띤 作業흐름의 세부항목을 정의해주고 各 단계에서 산출되는 문서등을 상세히 해주기 때문이다.

〈표 1〉 시스템 開發 壽命週期の 分類比較

학 자	Davis/Olson (1985)	Zmud/Cox (1979)	Bally etal (1977)	Whitten etal (1986)	Lucas (1986)
시스템 분 석	문제의 정의 타당성평가 정보요구분석 개념적 설계	착수 전략적설계	착수 문제점조사 요구사항확인 상세조사분석	상황조사 현행시스템요 구, 요구정의, 해결대안평가	예비조사 타당성조사 시스템분석 요구분석
시스템 설 계	물리적시스템 설계, 물리적 DB설계, 프 로그램작성 지침서 작성	기술적 설계 개발	실예 검사 및 시험운용	최적안선택 신규시스템 설계	설계, 명세, 프로그래밍, 테스트, 사용 자 훈련
시스템 실 행 및 유 지	변경, 운용 및 유지, 사후감 사	변경평가	수용, 수용및 유지, 사후감 사, 수정	신규시스템 신규시스템인 도	전환 및 설치 운용

壽命週期 各 段階들은 학자에 따라 조금씩 다르게 주장한다. 이러한 차이는 주로

註 6) 李軫周外 4人, 經營情報시스템, 茶山出版社, 1989, p.19.

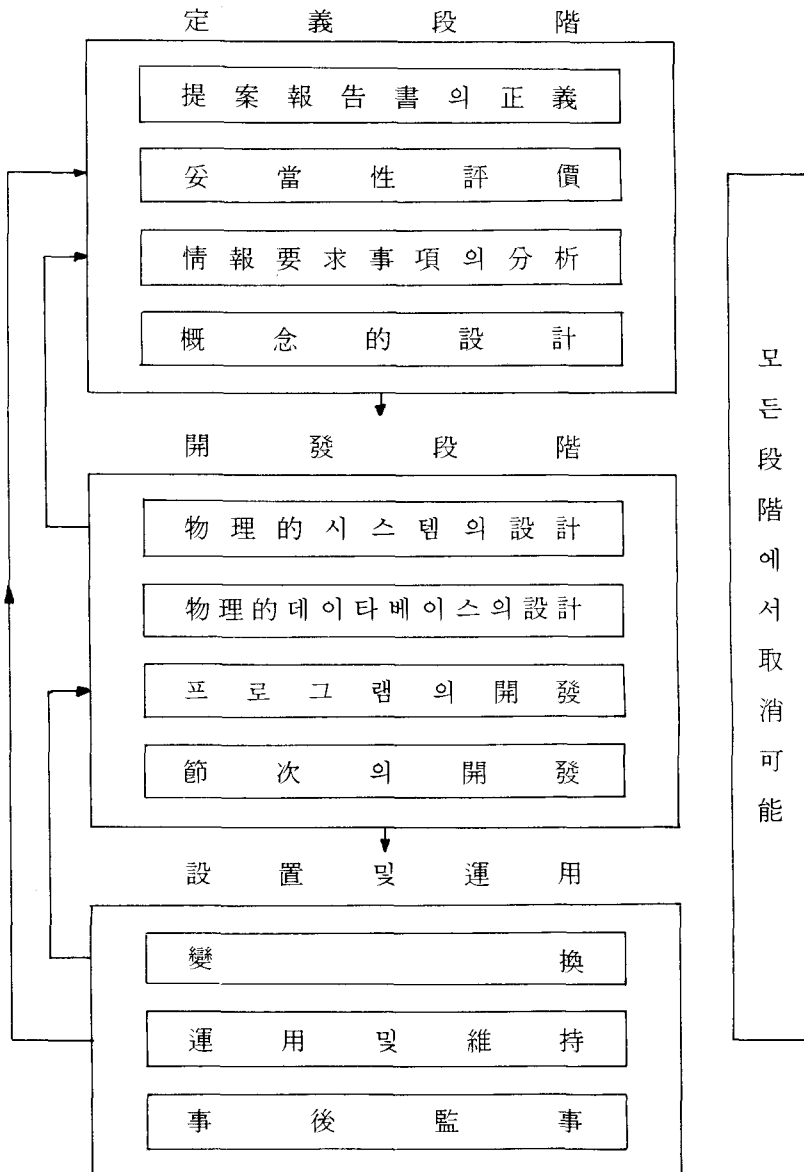
註 7) Mawdudur Rahman and Maurice Halladay, Accounting Information System, Prentice-Hall International, Inc., 1988. pp.7-8.

註 8) 유관희, 회계정보시스템, 나남, 1990, p.21.

세부항목의 수와 분류방법의 차이에 기인한다. 그러나 크게보면 定意段階(시스템 分析段階), 開發段階(시스템 設計段階), 設置와 運營段階(시스템 實行 및 維持段階)로 나눌 수 있다.(表 1參照)⁹⁾

本 研究에서는 데이비스(Davis)의 분류에 의한 壽命週期 模型을 살펴보고(그림 5 참조)¹⁰⁾ 다음 절에서 각 단계별로 그 내용을 살펴보기로 한다.

〈그림 5〉 시스템 開發壽命週期 模型



資料 : Davis, G. B. and Olsan, H., op. cit., p. 571

2. 定義段階¹¹⁾

시스템 定義段階는 提案報告書의 定義, 妥當性 評價, 情報要求事項의 分析, 概念的 設計 등이 포함된다. 이들 활동은 情報시스템 부서와 使用者의 협조하에서 이루어져야 하며 現行시스템의 分析을 전제로 하여 技術的, 經濟的, 運營의 측면을 고려한 대안을 제시해야 한다.

使用者에 영향을 주는 대부분의 주요 意思決定들은 定義段階가 완료되는 순간에 이루어진다. 따라서 시스템이 使用者를 충족시켜 주기 위해서는 定義段階를 소홀히해서는 안된다.

(1) 提案報告書의 定義(proposal definition)

提案報告書는 복잡해서는 안되며, 1-2페이지 정도의 짧은 文書로써 妥當性分析을 포함시켜 意思決定을 충분히 뒷받침 할 수 있어야 한다.

이 段階에 포함될 수 있는 항목은 다음과 같다. 첫째, 프로젝트를 제안한 사람과 시스템에 관심을 가지고 있는 사람들의 확인 둘째, 시스템에 대한 組織의 要求및 시스템으로부터 얻는 利益 세째, 組織의 支援, 豫算, 支援者, 經營層의 支援 네째, 日程에 대한 고려사항 등이다.

이러한 提案報告書는 使用者側 經營層의 承認을 받은 후에 시스템 開發者側 經營層에 의해 檢討된다.

(2) 妥當性 評價(feasibility assessment)

새로운 情報시스템이 提案되면 開發 承認을 얻기전에 妥當性 評價를 실시해야 한다.

妥當性 評價는 다섯가지로 세분할 수 있는데 이 結果를 보면 提案된 시스템의 개발과 實行에 내재하는 利益과 危險을 알 수 있다.

- 1) 技術的 妥當性 - 提案된 시스템이 기존의 技術로 實行될 수 있을 것인가? 하는 문제
- 2) 經濟的 妥當性 - 所要되는 費用보다 더 많은 利益을 시스템이 제공해 줄 것인가? 하는 문제

註 9) 李軫周外 4인, 前掲書, p.372.

李永煥, 시스템分析과 設計, 法英社, 1989, p.106.

註10) G.B. Davis and M.H.Olson, "Management Information Systems: Conceptual Foundations, Structures and Development", McGraw-Hill Inc., 1985, p.571.

註11) Ibid., pp.573-577.

- 3) 動機的 妥當性 - 使用者 參與, 資源, 訓練時間 등으로 시스템의 開發과 實行을 지원하기 위해 組織이 충분히 動機賦與될 수 있는가 하는 것이다.
- 4) 日程的 妥當性 - 시스템 開發을 위하여 허용된 時間內에 組織이 開發過程을 完了할 수 있는가 하는 것이다.
- 5) 運營 妥當性 - 시스템 設置後 正常的으로 實行될 수 있을 것인가 하는 것으로 이분석에는 시스템이 실행되는 戰略的 管理的 環境에 대한 主觀的 評價가 포함된다.

妥當性 評價報告書에는 다음 항목들이 포함된다. 첫째, 전반적인 시스템의 技術 둘째, 예상되는 開發 日程 세째, 開發에 요구되는 資源 및 豫算 네째, 運營의 결과 豫想 되는 費用 및 利益 다섯째, 각 妥當性에 관한 評價의 요약.

(3) 情報要求事項의 分析(information requirements analysis)

情報시스템은 그것이 속하여 봉사하는 組織의 요구에 부응해야 하고 使用者들의 요구에도 부응해야 한다. 따라서 이들의 要求事項을 파악, 분석하여야 한다.

(4) 概念的 設計(conceptual design)

이 段階는 보다 완전한 使用者指向的인 시스템의 設計를 확립시켜 준다. 시스템 設計는 概念的 設計와 物理的 設計로 나눌 수 있으며 이러한 두 단계는 外部設計와 內部 設計 또는 概略設計와 細部設計로 특징지을 수 있다. 概念的 設計는 시스템의 出力을 사용하거나 運營할 사람들에 의해 보여지는 시스템을 강조하며, 物理的 設計는 要求事項을 시스템의 實行을 위한 지침서로 바꿔준다. 그리고 概念的 設計는 入力과 出力, 시스템에 의하여 수행 되어야 할 기능, 시스템의 監査와 統制機能 등을 확립시켜 준다.

3. 開發段階¹²⁾

(1) 物理的 시스템 設計(physical system design)

內部設計 또는 細部設計라고도 불리는 이 設計는 시스템의 細部的인 技術的 設計를 준비하기 위한 역할 들로 구성된다. 이는 情報要求事項의 分析和 概念的 設計에 바탕을 두고 있으며, 物理的 데이터베이스의 設計, 프로그램의 開發, 節次的 開發 등에 기초를 제공한다.

註12) Ibid., pp.577-588.

이 단계의 결과로 다음과 같은 指針書와 設計가 제출된다.

첫째, 작업, 프로그램, 使用者機能 등의 흐름을 보여주는 시스템 設計

둘째, 처리흐름의 여러지점에서 實行되어야 할 統制役割을 나타내는 統制設計.

셋째, 새로운 하드웨어가 필요한 경우 시스템의 하드웨어에 대한 지침서.

네째, 데이터통신에 대한 要求事項과 指針書.

다섯째, 응용에 필요한 프로그램의 全體的 構造와 각각에 修行될 機能에 대한 節次 明細

여섯째, 保安 및 백업(back up)

일곱째, 차후 開發段階에서 필요로 하는 시스템 檢査나 品質保證計劃.

이러한 設計作業은 시스템 分析家나 다른 기술자 즉, 통제전문가, 품질확인요원, 데이터통신 전문가 등에 의해 수행 된다.

(2) 物理的 데이터베이스 設計(physical database design)

物理的 데이터베이스 設計方法은 현존하는 데이터베이스에 의하며, 데이터베이스 요구사항 결정이 선행된다. 시스템 開發에 있어서 데이터 要求事項 충족을 위한 接近 方法은 다음과 같이 세 가지다.

- 1) 새로운 物理的 화일 및 데이터베이스 生成 — 이 방법은 현재 데이터베이스가 없거나 응용 시스템이 전용 데이터베이스를 갖고 있는 경우에 적합하다. 이 경우는 데이터베이스의 目的(공유성 및 데이터의 獨占性)이 상실 된다. 따라서 새로운 資料가 요구되고 화일이나 데이터베이스는 특정 응용시스템을 위해 蒐集된 資料들로부터 設計되고 創造되어야 한다.
- 2) 현존하는 데이터베이스의 利用 및 修正 — 이 방법은 데이터베이스의 開發에 있어서 점진적으로 戰略이 채택되는 경우에 적절하다.
- 3) 使用者 스키마(user schema)를 통한 기존 데이터베이스에의 접근 — 이 방법은 새로운 시스템의 필요를 예견하며 企業의 완전한 概念的 模型이 物理的 데이터베이스 設計의 基礎로 사용될 경우 가능하다. 새로운 資料가 데이터 베이스에 추가될 필요도 있지만, 시스템이 필요로 하는 자료에 대한 논리적기술과 물리적 데이터베이스간의 연결은 데이터베이스관리 소프트웨어를 통해 쉽게 이루어진다.

(3) 프로그램 開發(program development)

이 段階의 目的은 시스템에서 요구되는 프로그램을 코드화하고 檢査하는 것이다.

각 모듈에 대한 檢査는 使用者 環境의 入力資料面에서 완전한 다양성을 갖는 檢査資料에 대해 실시된다. 檢査는 시스템 개발에서 중요한 요소이다. 이것은 시스템의規模나 복잡성에 따라 전체 開發努力의 15-50%를 차지한다.¹³⁾ 프로그램의 檢査에는 段階가 있다.

- 1) 모듈檢査(module testing) - 각 개별 프로그램의 모듈은 다른 모듈들에 대한 요구를 나타내는 데미모듈을 사용하여 檢査한다.
- 2) 統合檢査(integration testing) - 프로그램 모듈집단은 그것이 적절히 접속되어 있는지를 보기위해 檢査된다. 이것은 프로그램 전체시스템의 檢査될 때까지 開發정도에 따라 점진적으로 행해진다.
- 3) 시스템檢査(system testing) - 이것은 프로그램 전체를 檢査하는 것이다.

(4) 節次의 開發(procedure development)

節次開發(매뉴얼, 指示書式, 入力양식 등)은 프로그램 開發과 동시에 발생할 수 있다. 질차는 그 시스템과 접촉하는 모든 사람을 위해서 마련되어야 하며, 다음과 같은 것이 포함되어야 한다.

- 1) 1차적 使用者에 관한 사항 - 報告書를 어떻게 解釋하며, 報告書의 여러 다른 옵션을 어떻게 선정하는가에 대한 지시사항들을 포함한다.
- 2) 2차적 使用者에 관한 사항 - 각 종류의 入力物을 어떻게 入力시키는가에 대한 지시가 포함되는데, 무엇을 入力시킬 것인가 보다는 어떻게 入力시킬 것인가 하는 문제가 강조된다.
- 3) 컴퓨터 運營要員에 관한 사항 - 일반적으로 維持節次는 컴퓨터 運營者나 統制要員에 의해 수행되어야 할 작업을 상세하게 해주는데, 이에는 品質保證, 백업을 위한 시스템화일, 維持프로그램의 文書化 등이 포함된다.
- 4) 訓練節次 - 경우에 따라 분리된 訓練用 매뉴얼, 혹은 일련의 訓練書面이 實行段階와 그 후의 訓練을 위해 開發된다.

節次의 開發時는 시스템 分析家와 使用者가 협력하여 技術的으로 正確하면서도 理解하기 쉽도록 작성하여야 한다.

4. 設置와 運營段階¹⁴⁾

註13) Ibid., p.573.

註14) Ibid., pp.589-591.

이 段階는 새로운 시스템으로서 轉換과 함께 시작된다. 시스템이 運營되기 시작한 후 시스템의 運營과 維持段階에서 變化와 補強이 이루어진다. 또 週期的으로 시스템에 대한 事後 檢査가 실시된다. 이러한 활동을 기반으로 시스템은 修正, 補完되거나 對替된다.

(1) 轉換(conversion)

새로운 시스템으로의 轉換은 모든 프로그램과 節次가 준비되고 이러한 것들이 個別的으로 檢査된 후에 시작된다. 실제 轉換을 위한 주요활동은 다음과 같다.

- 1) 受容檢査 - 이는 완성된 시스템을 檢査하며 그것을 指針書와 比較한다. 그리고 使用者에게 시스템이 成果基準과 運營的 要求事項을 충족시켜주고 있다는 것을 증명해 준다. 이러한 檢査에는 使用者의 出力, 運營 및 統制의 節次, 出力 등이 포함된다.
- 2) 화일구축 - 이것은 시스템에 의해 요구되는 모든 새로운 자료를 수집하여 컴퓨터가 관독할 수 있는 형식으로 전환시키는 것을 말한다.
- 3) 使用者 訓練 - 적절한 使用者 訓練은 새로운 시스템에 대한 使用者의 抵抗을 극복하는데 중요한 요소다.

轉換은 위 세 가지 활동이 완료된 후에 이루어진다. 轉換은 여러가지 방법으로 이루어 질수 있다. 가장 신중한 방법은 기존 시스템과 새로운 시스템을 병행하여 運營하는 방법이다. 즉, 새로운 시스템은 실제 상황하에서 운영되며 그 결과는 신뢰성과 정확성의 評價를 위해 기존시스템과 비교된다. 새로운 시스템은 적정기간 동안 일관된 결과를 보여준 후에야 運營되며 기존시스템을 폐기된다.

(2) 運營과 維持(operation and maintenance)

시스템이 별다른 어려움 없이 運營되고 있다고 생각되면 情報處理 生産機能으로 넘겨진다. 이것은 시스템이 미리 규정된 承認基準을 충족시켜주고 있다는 使用者의 承認뿐만 아니라 시스템의 運營과 維持, 補修를 담당하는 부서의 承認까지 받아야 가능하다.

시스템의 維持作業은 補修나 向上(enhancements)으로 분류할 수 있다. 補修는 不完全하거나 不正確한 코딩으로 인하여 시스템에 결함이 생겼을 때 필요하며, 向上은 시스템을 개선하는 것이다.

(3) 事後監査 (post audit)

시스템이 일정기간 運營된 후에는 그 시스템에 대한 檢討가 이루어져야 한다. 使用, 開發, 維持, 運營, 內部監査 등을 담당하는 부서의 대표들로 구성된 監査팀이 제 반사항을 檢討한다.

이상에서 情報시스템 開發 壽命週期の 各 段階別 주요내용을 살펴보았다. 이들 段階는 相互排他的으로 적용하는 것이 아니라 feed back에 의해서 相互補完的, 反復的으로 적용되어야 바람직한 情報시스템을 構築할 수 있다.

IV. 原型開發接近法

1. 原型開發接近法の 意義

시스템 開發에는 不確實性이 수반된다. 人力과 資金을 투입하여 開發한 시스템이 使用者의 요구를 충족시킨다는 보장이 없다. 이러한 不確實性은 使用者의 要求를 정의하기가 어려운 意思決定내지 戰略企劃支援 시스템일수록 더욱 커진다. 왜냐하면 非構造的인 문제를 해결하려면 어떠한 情報가 필요한지 사전에 정의하기가 힘들기 때문이다. 앞에서 고찰한 시스템 開發 壽命週期接近法은 보다 複雜한 시스템 開發엔 적합하지 않다. 따라서 이에 적합한 방법으로 原型開發 接近法이 등장하게 되었다.

原型開發接近法(prototyping)이란 시스템 開發初期에 시스템 原型(prototype)을 만들어 使用者에게 보여주고 使用者로 하여금 이를 실제 작동시켜 기능의 追加, 變更내지 削除를 요구토록하여 시스템을 改善시켜 나가는 방식을 뜻한다.¹⁵⁾ 프로토타이프란 初期模型 또는 原型을 의미한다.

2. 原型開發接近法の 模型

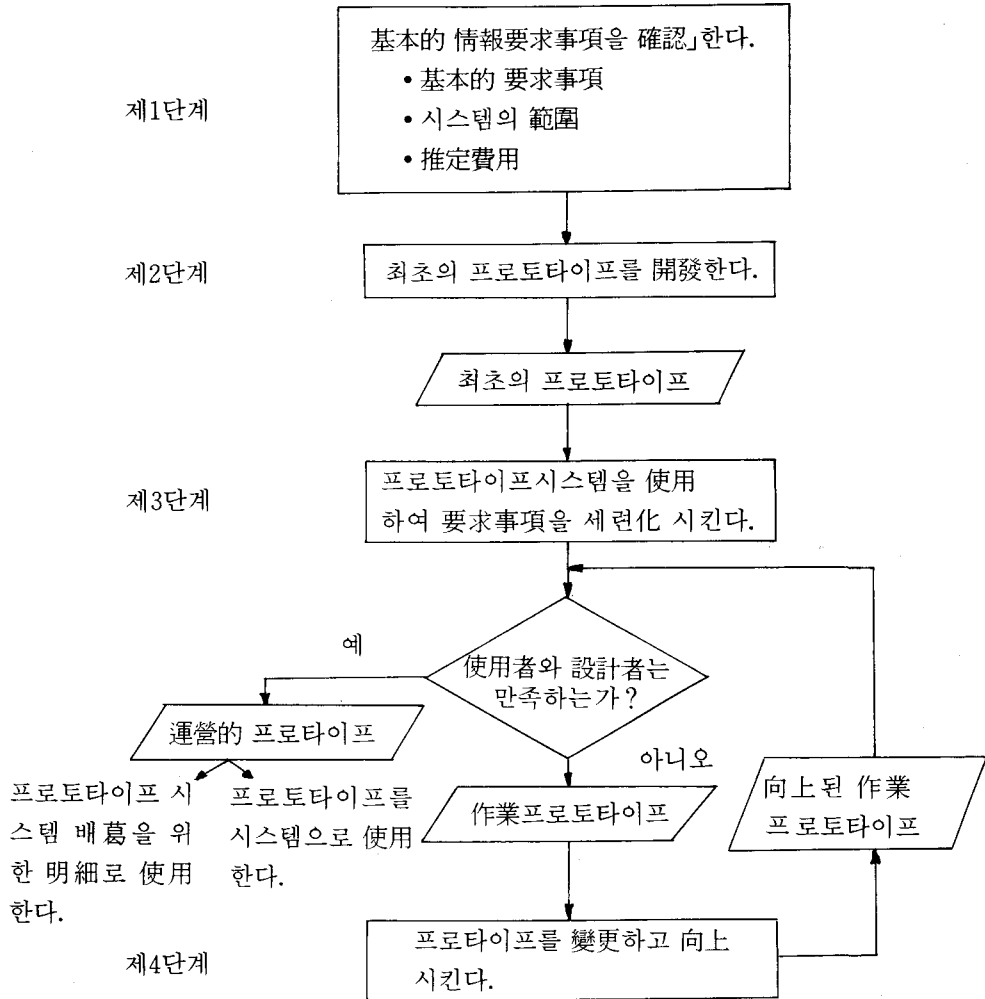
原型開發接近法の 模型은 네 段階의 過程으로 구성된다. (그림 6참조)

(1) 使用者의 基本的 情報要求事項 確認

이 段階에서는 使用者가 基本的으로 필요로 하는 出力情報가 무엇인가를 명확히 파

註15) 이영환, 전계서, p.112.

〈그림 6〉 原型開發接近法の 模型



자료 : Jenkins, A. M., "Prototyping : A Methodology for the Design and Development of Application Systems", Working Paper, School of Business, Indiana University, 1983.

약한다. 設計者は 實際 使用者의 기대를 확인하고 運營的 原型의 開發費用을 推定 한다.

이 段階에서 제시되는 두 가지 강조점은 "자료추상" 접근법 "과정시물레이팅" 접근 법이다. 前者는 데이터베이스에 運營되는 파일럿 시스템과의 實驗形態를 띤 infologic-al 시물레이션이 적절하다고 제안한다. 後者는 原型開發의 첫 段階는 過程을 模型化 하는 일이라는 것이다. 이러한 두 접근법은 모두 使用者 要求事項의 基本의 特性을

확인하려는 것이다. “자료추상” 접근법은 기본특성이 자료와 자료간의 관계라고 가정하며, 과정은 일반화된 데이터베이스 管理시스템의 보조적 특성을 통해 제공되기를 기대하는 것이다. 이와 반대로 “과정시물레이팅” 접근법은 資料와 過程이 모두 첫 단계에서 확인되어야 한다고 가정한다.

(2) 原型의 開發

이 段階의 目的은 使用者의 基本的 情報要求事項을 충족시켜주는 기능적인 原型을 신속히 開發하는 것이다. 設計者는 高度水準의 開發言語와 開發道具를 사용하여 시스템을 構築한다. 作動의 迅速性이나 完全性은 문제가 되지 않고 迅速한 開發이 關鍵이다. 즉, 能率보다는 速度가 강조되며 開發된 原型은 使用者에게 즉시 제공된다.

도노반(Donovan)과 매드닉(Madnick)은 原型開發이 몇 달 내지 몇 년 보다는 몇 주 만에 이루어질 때 效果의이라고 주장하였다.¹⁶⁾

(3) 原型의 實行

이 段階는 使用者가 情報要求事項을 理解하고 이러한 要求事項들을 충족시켜주기 위해 시스템이 할 수 있는 일과 할 수 없는 일을 구분할 수 있도록 使用者로 하여금 시스템을 직접 사용토록 한다. 使用者는 초기에 시스템의 問題를 발견할 수 있고 設計者 보다는 使用者가 언제 變更이 필요한지를 결정하며, 전체적인 開發 시스템을 統制한다.

시스템의 直接的인 使用은 經驗, 理解 및 評價를 할 수 있게한다. 시스템 開發에 傳統的 模型이 使用되든 原型開發模型이 使用되든 評價는 명백히 이루어져야 한다. 새로운 시스템을 경험할 때마다 使用者는 부적합을 야기시키는 不一致와 刺戟을 발견할 것이다. 原型開發接近法은 이러한 過程을 통해 使用者 要求事項을 수렴하게 된다.

(4) 變更 및 向上

使用者에 의해 確認된 바람직하지 못하거나 누락된 特性들은 즉시 修正되어야 한다. 使用者가 要求하는 기능의 追加, 變更, 削除要請을 迅速히 수용하는 것이다. 使用者는 한차례의 評價만으로 모든 問題를 確認할 수 없다. 따라서 위 (3), (4) 段階는 시스템이 적합하다고 承認될 때까지 反復되어야 한다.

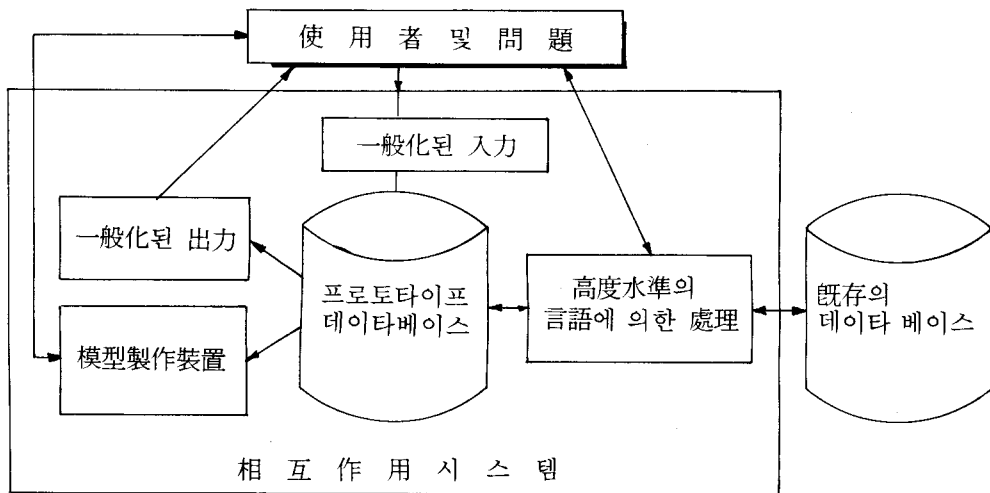
註16) J.J.Donovan and S.E.Madnick, "Institutional and Ad Hoc DSS and Their Effective Use", Database, Winter, 1977, pp.78-88.

이렇게 해서 使用者를 滿足시키는 시스템 模型이 완성되면 여기에 에러검증 기능을 추가시켜 生産시스템화 하든가 原型과 동일한 기능을 제공하는 節次型 言語로 開發한다. 그 이유는 처리속도 向上을 위해서다.¹⁷⁾

3. 原型開發接近法에 要求되는 資源¹⁸⁾

使用者와 시스템과의 相互作用을 강조하는 原型開發接近法을 支援하는 資源은 온라인 相互作用시스템, 데이터베이스관리시스템(DBMS), 高度水準의 言語, 一般화된 入.出力 소프트웨어(S/W), 接近이 용이한 模型製作裝置등이 있다. 이들 관계는(그림 7참조).

〈그림 7〉 原型開發接近法에 要求되는 資源



資料 : N aumann, J. D. and Jenkins, A. M., "Prototyping : The New Paradigm for Systems Development", MIS Quarterly, Sep. 1982, p. 35.

(1) 相互作用시스템

原型開發接近法에서는 開發者와 시스템은 使用者의 要求事項에 대해 迅速히 응답해야 한다. 몇몇 일괄(batch)시스템들은 신속하게 응답할 수 있는 능력을 具有하고 있지만, 이 시스템으로 使用者와 步調를 맞추어 대화하고 變更할 수는 없다. 相互作用

註17) 李永煥, 前掲書, p.113.

註18) J.D.Naumann and A.M.Jenkins, op.cit., pp.34-36.

시스템의 능력은 계산자원을 확대시켜 준다. 이러한 裝置는 遲延現狀을 減少시키고 資源에 대한 統制를 使用者에게 확장시킴으로써 情報處理資源의 힘을 증대시켜 준다.

(2) 데이터베이스 管理시스템

相互作用的 處理資源과 연결될 경우 DBMS의 質疑語에 바탕을 둔 自然言語는 資料를 관리하는 대부분의 보조기능을 제공한다. 傳統的인 시스템 設計의 상당부분은 物理的 貯藏構造를 設計하며 백업, 保安, 統合성을 제공하고 특별한 資料轉換 프로그램을 定義 創造하는 作業들과 意思決定에 중점을 두고 있다. 그러나 이러한 節次指向的 作業은 使用者에게 직접적인 도움이 되지 않는다. DBMS는 신속하고 쉬운 창조와 변경을 제공해야 하며 입력, 경신(update)시키는 절차 및 광범위한 보고장치를 포함해야 한다. 따라서 DBMS는 원형개발의 핵심부분이 된다. 데이터 베이스관리 S/W를 상용하지 않고서는 원하는 시간내에 자료처리 장치를 설계하고 원형을 개발할 수 없다.

(3) 一般화된 入, 出力 S/W

다양한 能力을 가진 報告書作成裝置, 質疑語 등은 수년동안 사용되어 왔으며 종종 DBMS의 특성이 됨과 동시에 이것과 연결된다.

一括處理시스템에 있어서는 編輯, 確認, 誤謬의 修正, 統制등이 일반적인 방식으로 정의하기 어렵기 때문에 일반적인 S/W는 널리 사용되지 못했는데, 온라인 技術과 原始資料의 채택으로 資料入力時 단독처리기준으로 編輯하고 確認하며 誤謬를 修正하는 것이 가능하게 되었다.

일반화된 입력 S/W는 데이터베이스를 만들어 주며 복잡한 編輯프로그램의 構築이 없어도 更新할 수 있다. 일반화된 出力 S/W중에서 두 가지를 살펴보면 다음과 같다.

- 1) 質疑語¹⁹⁾ - 이 言語는 보통 DMBS와 연결되며 使用者로 하여금 簡單하거나 複雜한 선택기준을 사용하여 데이터베이스나 파일을 檢索할 수 있게 해 준다. 檢索結果는 상세하게 또는 요약형식으로 나타낼 수 있다. 이러한 유의 S/W는 資料檢索뿐만 아니라 更新도 가능하게 한다.
- 2) 報告書 作成裝置-이것은 보다 精密한 資料處理作業을 修正하고 거의 어떤 형식으로든 報告書를 제시할 수 있다는 점만 제외하고 質疑語와 類似하다. 일반적으로

註19) J.O.Hicks, Jr., Management Information Systems: A User Perspective, West Publishing Co., 1984, p.387.

質疑語는 資料處理專門家の 도움없이 使用者에 의해 使用될 수 있도록 設計된다. 그러나 報告書 作成裝置의 사용은 시스템 分析家와 같은 資料處理專門家の 도움을 필요로 한다.

(4) 高度水準의 言語

傳統的인 接近法은 情報시스템의 開發에 대한 한가지 언어(예컨대 COBOL이나 PL/1)의 관점에서 진행시키는데 原型開發法 하에서 設計者의 言語資源은 融通性을 지닌다.

言語의 選擇基準은 多様한데 運營上의 능율이 프로그램 開發의 主目的이 아니기 때문에 自動文書化, 變更의 容易性, 코딩 및 檢査의 速度등이 훨씬 더 중요하게 된다. 言語는 적절한 DBMS자원과 연결되거나 연결가능해야 하며 신속한 報告書 作成指針과 入力處理指針을 제공해야 한다.

(5) 模型製作裝置

直接 意思決定을 하거나 意思決定을 支援하는 模型은 많은 시스템에 있어서 필수적이다. 잠재적으로 사용가능한 模型들의 집합인 모델뱅크는 原型開發接近法의 중요한 資源이다.

高度水準의 言語도 模型의 開發을 容易하게 해주지만 原型開發接近法에서는 모델뱅크가 信賴性 있는 모형의 신속한 開發에 더 잘 사용된다.

4. 세 가지 構成要素의 役割²⁰⁾

原型開發接近法에서는 使用者, 開發者, 시스템사이의 相互作用이 강조된다. 이들의 役割은 다음과 같다.(그림 8참조)

(1) 使用者

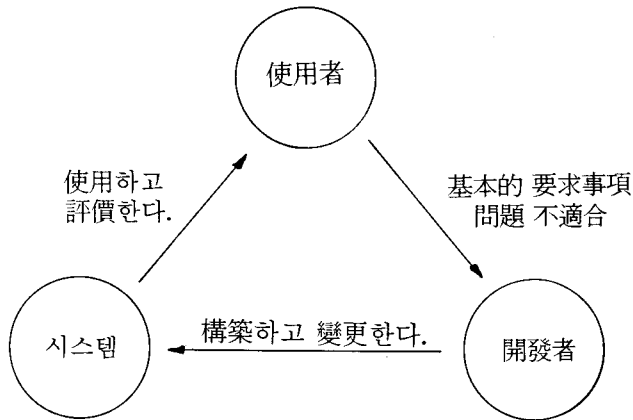
使用者들은 傳統的 開發接近法에서 보다 原型開發接近法에서 더 能動的인 役割을 한다. 사실상 그들은 시스템을 使用하고 評價하는 시스템 設計者이며, 過程에서는 問題點을 發見하고 그 解決策을 제시한다. 使用者들은 原型을 使用하고 評價하는데 소요되는 시간으로 개발속도를 결정하며 評價와 變更의 週期가 언제 完了될 것인가를

註20) J.D.Naumann and A.M.Jenkins, op.cit., pp.36-38.

결정한다.

使用者는 原型의 開發過程에서 시스템 開發者와의 관계속에서 그들의 特定技術에 대한 評價와 理解를 발전시켜나가야 할 것이다.

〈그림 8〉 原型開發接近法에 있어서 構成要素의 役割



資料 : Keen, P. G. W., "Adaptive Design for Decision Support Systems",
Database, 12 : 1-2, Fall 1980, pp. 15-25.

시스템 開發에는 다수의 組織單位와 여러 個人이 參與하는데 使用者의 役割에는 이들간의 調整과 커뮤니케이션이 포함된다.

시스템 開發에 使用者가 關여하게 되면 다음과 같은 利点이 있다.²¹⁾

첫째, 새로운 시스템은 그 시스템 使用者의 能力과 실제적인 情報要求事項을 더 잘 반영한다.

둘째, 새로운 시스템은 공동노력의 결과이기 때문에 외부에서 開發한 시스템보다도 더 잘 받아들일 수 있다.

셋째, 使用者가 關여하게 되면 새로운 시스템을 設置하여 運營할 때 전형적으로 일어날 수 있는 문제들을 解決하는데 使用者들은 보다 잘 協助하게 된다.

(2) 開發者

開發者의 役割은 技術과 經濟性에 의해 제약되는 컨텍스트(使用者의 要求事項)와 形式간의 葛藤을 調整, 解決해 나가면서 연속적으로 새로운 스템을 구축하는 것이다.

註21) 蘇令一, 李鍾敏, 經營管理를 위한 컴퓨터原論, 法英社, 1987, pp.537-538.

이러한 역할은 시스템 設計者나 프로그램 分析家의 役割과 유사하다고 할 수 있다. 開發者는 기능적 知識 이외에도 프로토타입 시스템의 開發을 支援하는데 이용가능한 技術을 熟知해야만 한다.

原型開發接近法 模型의 제 1段階에서 開發者는 프로토타입 시스템의 기본적 특성을 定義하기 위해 使用者와 協力한다. “자료추상법근법”하에서 開發者는 관련있는 實體, 이들의 屬性, 關係등을 확인해야 하는데, 데이터베이스의 정의를 創造해 내기 위해서는 충분한 細部事項이 포함되어야 한다. “과정시물레이팅접근법”에서 開發者는 使用者가 要求하는 過程을 확인해야만 한다. 또한 開發者는 資料와 過程을 利用 가능한 데이터베이스와 화일, 모형과 연결시켜야만 하며 어떻게 초기의 原型을 迅速하게 實行할 것인가를 결정해야 한다.

제 2段階에서 開發者는 原型을 구축하기 위해 시스템의 資源을 사용한다.

제 3, 4段階에서 開發者는 使用者와 密接하게 협력하여 급속한 變更에 수반되는 문제와 부적합한 인식에 대응한다. 따라서 開發者는 使用者의 可變의인 要求에 응답하며 프로토타입 시스템이 이러한 요구에 계속적으로 부응하도록 해준다.

(3) 시스템

情報시스템에 대한 고리(Gorry)와 머튼(morton)의 模型²²⁾을 準據의 틀로 사용한다면 原型開發法은 어떠한 기능 분야에서도 사용될 수 있다. 특히 시스템의 戰略的 내지 戰術的 水準에서 유용하게 사용될 것이다. 이들 수준에서의 管理活動은 비교적 非構造化되고 意思決定은 開放的이며 프로그래밍의 可能性은 比較的 작다. 그리고 使用者 環境과 設計의 대안들에는 더 많은 不確實性이 內在한다.

따라서 原型開發法을 適用하기에 가장 좋은 시스템은 管理機能과 관련 있다. 管理機能은 전형적으로 計劃, 指揮, 統制, 問題解決, 意思決定 등과 연관된다. 運營的 統制시스템과 처리시스템도 原型開發法을 적용할 수 있다. 下位水準의 시스템은 意思決定支援 시스템 만큼 定義하기가 힘들 수도 있지만 高度水準의 시스템과 마찬가지로 變更과 向上을 필요로 한다.

V. 兩 接近法의 比較

앞에서 고찰한 情報시스템 開發의 두 接近法을 概念, 特性, 模型, 開發週期 各段階에 미치는 效果, 有用성과 限界면에서 比較分析하면 다음과 같다.²³⁾

註22) J.D.Naumann and A.M.Jenkins, op.cit., pp.37-38.

註23) 박연, “시스템 開發接近方法의 比較研究”, 朝鮮大學校 大學院 博士 學位論文, 1990.2, pp.62-64. 蘇令一, 李鍾敏, 前揭書, p.534.

1. 概 念

(1) 시스템 개발 壽命週期接近法

시스템이 生成, 成長, 成熟, 消滅 등의 壽命週期段階를 거치는 것처럼 정해진 節次(妥當性 調査, 시스템 分析, 設計, 運營 등)에 따라 시스템을 開發하는 방법.

(2) 原型開發法接近法

시스템 開發에 있어서 정해진 節次중 한 段階를 完了한 後 이의 結果를 使用者에 게 feed back시켜 使用者의 要求事項이 滿足되면 다음 段階로 계속 진행되며 만족되지 않으면 修正을 가하는 방식으로 시스템을 開發하는 방법.

2. 特 性

(1) 시스템 開發 壽命週期接近法

시스템 開發이 정해진 標準節次에 따라 수행되므로 시스템 開發 日程을 效率的으로 統制할 수 있다. 즉, 시스템 開發의 全體的인 管理가 용이하며 각 節次마다 指針이 뚜렷하여 效果的인 開發이 진행될 수 있다. 따라서 시스템의 質的 수준을 유지할 수 있다.

(2) 原型開發接近法

시스템 開發過程에 使用者를 直接 參與시킴으로써 使用者의 要求事項을 보다 명확히 파악할 수 있으며 開發期間의 초기단계에서 使用者의 變更要求를 발견하여 반영함으로써 시스템 開發期間 및 費用을 減少시킬 수 있다. 또한 시스템을 使用者 指向的으로 構築하며 使用者로 하여금 開發된 시스템에 만족할 수 있도록 한다.

3. 模 型

(1) 시스템 開發 壽命週期接近法

제Ⅲ장(그림5) 참조

(2) 原型開發接近法

제Ⅳ장(그림6)참조

4. 開發週期 各 段階에 대한 效果

시스템 開發에 있어서 두 接近法 開發週期 各 段階에 미치는 效果는 다음 (表2)와 같이 要約될수 있다.

5. 有用性和 限界

(1) 시스템 開發 壽命週期接近法

1) 有用性(長点)

가. 시스템 開發을 위한 標準이 제공된다.

시스템 開發은 많은 時間과 豫算이 소요되며 매우 복잡하다. 이와 같은 제약을 극복하고 전 시스템 開發에 대한 일반화된 방법을 제공한다는 점에서 시스템 開發 壽命週期接近法の 役割은 크다. 따라서 이 標準에 따라 시스템이 開發되며 文書化가 되므로 理論的 檢證이 쉽다.

〈표 2〉 開發週期 各 段階에 대한 兩 接近法の 效果比較

시스템 개발 활동	전통적인 시스템 개발	프로토타입을 도와주는 기구 (Prototyping Aid)를 응용발생기로써 사용하고 나머지를 프로그래밍하는 경우	전문적인 프로그래머 없이 컴퓨터를 이용하여 시스템을 개발할 경우
요구사항 분석	<ul style="list-style-type: none"> 시간이 많이 걸리는, 공식적인 작업이 긴 응용 백로그(long application backlog)에 의하여 때때로 지연됨 	<ul style="list-style-type: none"> 사용자의 상상력이 자극된다. 사용자는 요구사항들을 개발하기 위하여 분석가와 함께 스크린에서 일할 수 있다. 	<ul style="list-style-type: none"> 사용자의 상상력이 자극된다. 사용자는 스스로 요구사항을 개발하거나 분석가와 함께 개발할 수도 있다.
시스템 명세서	<ul style="list-style-type: none"> 긴 문서임. 읽기 지루함. 때때로 부적합함. 	<ul style="list-style-type: none"> 프로토타입을 도와주는 기구에 의하여 만들어진다. 정확하고 시험된 것이다. 	없어졌음.

사용자 종료 신호(signoff)	<ul style="list-style-type: none"> · 사용자는 때때로 그가 무엇에 대하여 종료신호를 하였는지를 확신하지 못함. · 사용자는 모든 미묘함(surtleties)을 인식할 수 없음. 	<p>사용자는 결과를 보고 종료신호를 하기 전에 여러 번 결과를 수정할 수도 있다.</p>	<ul style="list-style-type: none"> · 공식적인 종료신호가 없다. · 조정과 수정이 계속되는 과정이다.
코우딩 실험	<ul style="list-style-type: none"> · 늦음. · 비용이 많이 듦. · 백로그 때문에 때때로 지연됨 	<ul style="list-style-type: none"> · 프로토타입은 더욱 효율적인 코우드로 변환된다. · 상대적으로 빠르고 오류가 없다. 	<ul style="list-style-type: none"> · 빠르다. · 비용이 저렴하다. · 대부분 없어졌음.
문서화	<ul style="list-style-type: none"> · 번거로움 · 시간이 많이 걸림. 	<ul style="list-style-type: none"> · 부분적으로 자동화될 수 있다. 대화식 훈련(interactive training)과 헬프 반응(HELP response)을 온라인으로 만들 수도 있다. 	<ul style="list-style-type: none"> · 대개 자동화되어 있다. 대화식 훈련과 헬프 반응을 온라인으로 만든다.
유지보수	<ul style="list-style-type: none"> · 느림 · 비용이 많이 듦. · 때때로 지연됨. 	<ul style="list-style-type: none"> · 때때로 느림. · 때때로 비용이 많이 듦. · 때때로 지연됨. 	<ul style="list-style-type: none"> · 사용자와 분석가가 하는 조정을 하는 지속적인 과정이다. 이들 조정의 대부분을 아주 빠르게 할 수 있다(몇 개월이 아니라 몇 시간 내에 할 수 있다).

나. 프로젝트를 管理하고 統制하기가 용이하다.

이 방법은 프로젝트가 시작되기 전에 特定段階, 基準, 節次 등을 갖추고 있다. 따라서 프로젝트를 어떻게 計劃하고 豫算을 樹立하며 管理統制할 것인가에 대한 문제를 쉽게 해결할 수 있다.

2) 限界(短点)

가. 人件費가 많이 든다.

이 방법은 勞動集約的이며 시간이 많이 소요되는 과정을 갖는다. 기존의 시스템과

새로운 시스템을 위해 物理的, 論理的인 資料흐름의 다이어그램을 만들고 構造圖를 만들며 프로그램작성 手作業에 의한다. 따라서 人件費가 過多하게 支出된다.

나. 시스템 開發에 많은 時間이 要求된다.

다. 變化의 受容이 느리다.

시스템 開發에 要求되는 시간이 길기 때문에 발생하는 문제점이다. 企業의 動的인 屬性으로 인하여 妥當性 分析의 段階에서 처음으로 定義된 S/W가 시스템이 運營될 때는 그 시스템에 대한 要求事項이 變化될 수 있다. 이러한 使用者의 要求事項 變化를 민첩하게 수용할 수 없다.

라. 作業이 過重하다.

하드웨어(H/W)費用의 減少로 인하여 많은 使用者들이 컴퓨터를 購入하게 되었다. 만일 이들을 위한 S/W개발을 프로그래머에게 의존하게 된다면 이들의 작업은 過重하게 될것이고 入力은 부족하게 될 것이다.

(2) 原型開發接近法

1) 有用性

가. 使用者指向의인 시스템 開發方法이다.

이 방법은 시스템 開發過程 특히 分析 및 設計段階에서 使用者를 能動的으로 參與시킴으로써 그들의 要求事項을 開發過程에 구현시켜 使用者 指向의 시스템을 開發할 수 있도록 한다.

나. 使用者에게 開發對象 시스템에 대한 現實的 手段提供

이 방법은 使用者에게 開發對象 시스템을 理解하고 評價할 수 있는 현실적인 수단을 제공하여 그들의 要求事項에 效果的으로 feed back할 수 있게 해준다.

다. 開發過程段階에서 要求事項 明確化

시스템 開發의 초기단계에서 현존하는 問題點을 발견하고 반영할 기회를 갖도록 使用者와 開發者 모두에게 參照할 수 있는 基準을 제공한다.

라. 使用者의 積極的인 參與姿勢 誘導

原型開發接近法은 開發對象 시스템의 使用者 參與姿勢를 積極的인 방향으로 誘導하는 실질적인 방법이다. 使用者에게 原型시스템을 實際 運營해 볼 기회를 제공하여 자료처리가 자기들의 要求事項에 반응하는 것을 피부로 느끼게 해준다. 이에 따라 使用者의 시스템 開發에 대한 姿勢는 消極的인 姿勢로부터 積極的인 姿勢로 바뀐다.

마. 開發關聯 危險을 減少시킨다.

고마(Gomaa)와 스코트(Scott)에 따르면 原型에 소요되는 費用은 시스템 開發 全

體費用의 10%미만이다.²⁴⁾ 따라서 시스템 開發에 시간 및 費用이 많이 소요되는 큰 프로젝트에서의 危險을 原型을 통해 발견하고 豫防하며 극단적인 경우 시스템 開發을 取消한다.

바. 開發者와 使用者의 관계개선

使用者와 시스템 開發者는 공동으로 시스템을 開發하는 프로토타입에 의해 意思 疏通을 원활히 하고 친밀한 關係를 유지하며 상대방의 업무에 호감을 갖게된다.

사. 시스템 維持費用의 節減

시스템 維持費用은 開發完了後에 使用者의 要求事項 變更에 의해 크게 영향을 받는다. 開發完了後의 要求事項變更은 시스템 開發시 使用者의 要求事項이 명확히 반영되지 못하였기 때문에 발생하는 경우가 많다. 그러나 原型開發接近法에 의하면 이러한 문제들이 해결되어 시스템 維持費用을 節減할 수 있다.

2) 限界

가. 프로젝트를 管理하고 統制하는 일이 어렵다.

原型의 屬性과 生소함으로 인하여 어떻게 計劃하고 豫算을 수립하며 管理하고 統制할 것인가 하는 문제를 해결하기가 힘들다. 傳統的 方法은 프로젝트가 시작되기 전에 計劃과 統制에 대한 指針으로 各 段階別 基準, 節次 등이 마련되어 있다. 그러나 原型開發接近法에서는 프로토타입 시스템이 進化的 형태를 갖고 繼續的인 수정이 이루어지며 使用者 要求事項의 상당부분은 명확히 알기 어렵기 때문에 개발프로젝트에 대한 적절한 管理 統制가 어렵다.

나. 原型開發接近法은 濫用될 수 있다.

原型은 제한된 能力을 갖고 있으며 단지 運用 시스템의 基本的 特性만 갖추고 있다. 그런데 原型을 過大評價함으로써 使用者는 종종 非現實的인 要求를 하는 경우가 있다. 이로 인하여 기대가 충족되지 않을 경우 使用者가 失望하게 된다.

다. 開發者와 使用者가 齟齬를 낼 수 있다.

原型은 完了時点 判斷基準에 도달할 때까지 반복적으로 수행된다. 不確實한 시스템을 開發할 경우는 수 십번 반복될 수 있다. 이런 경우 使用者나 開發者는 처음 시작할 때와 같은 參與의 熱情이 식을 수가 있다.²⁵⁾

라. 費用의 많이 들 수 있다.

註24) H.Gomaa and D.S.h.Scott, "Prototyping as a Tool in the Specification of User Requirements" The Proceedings of the 5th International Conference on Software Engineering, 1981. pp.333-342.

註25) A.Prizant, "Is Prototyping counter Productive?", Data Processing, Vol.28, No.7, Sep. 1986, p.379.

原型開發接近法은 高度의 H/W 및 S/W를 필요로 하며 熟練된 使用者와 開發者를 필요로 한다. 제4세대 言語와 關係型 DBMS등의 開發道具 내지 環境이 갖추어져야 하는바 이러한 道具 購入에는 經費가 많이 소요된다.

마. 模擬實驗이 기대한 만큼 철저하지 않을 수 있으며 變更이 쉬운 만큼 文書化의 철저를 기하기가 어렵다.

V. 結 論

本 研究의 目的은 情報시스템 開發接近法을 考察하여 成功的인 會計情報시스템 開發에 一助를 加하는 것이다. 이러한 目的達成을 위해 지금까지 會計情報시스템의 本質을 살펴보고 情報시스템 開發의 傳統的 技法인 시스템 開發 壽命週期接近法과 이것의 缺點을 보완하기 위해 등장한 原型開發接近法의 內容을 考察하고 兩 接近法을 比較, 分析하였다.

시스템 開發 壽命週期接近法은 情報시스템 開發이 정해진 標準節次에 따라 수행되므로 시스템 開發日程을 效率的으로 統制할 수 있고 시스템 開發에 대한 管理가 용이하다. 하지만 보다 複雜한 시스템이나 開發에 대한 不確實性이 큰 시스템엔 이 방법이 適合하지 않다. 原型開發接近法은 다양한 資源을 使用하고 頻繁한 變更이 이루어지는 複雜한 시스템과 要求事項을 상세히 하기 어려운 시스템 開發에 適切한 방법이다. 특히 시스템 開發에 使用者 參與를 積極的으로 誘導할 수 있는 技法이다.

그러나 이 두가지 接近法은 하나가 다른 하나를 대치하거나 原型開發接近法이 모든 경우에 絶對적으로 優秀한 방법은 아니다. 오히려 相互補完關係에 있다고 할 수 있다. 가장 適切한 接近法의 選擇은 시스템의 規模, 使用者의 知識, 開發者의 技術, 시스템의 安定性 등을 포함하는 많은 요소들의 狀況에 의존한다. 대체로 시스템 開發 壽命週期接近法은 시스템의 大規模이고 支援하는 意思決定이 高度로 構造化 되어 있고, 作業에 대한 使用者의 理解와 開發者의 熟練도가 높은 경우에 적용된다. 原型開發接近法은 시스템이 小規模이고 支援하는 意思決定이 非構造化 되고 作業에 대한 使用者의 理解와 開發者의 熟練도가 낮은 경우에 적용된다.

따라서 會計情報시스템의 開發技法選定은 組織의 特性, 會計情報시스템관련자의 기능, 會計情報시스템자체의 特性 등 여러 狀況要因을 고려해서 이루어져야 할 것이다. 이를 테면 最高經營層의 支援정도, 電算部署의 歷史, 組織의 規模, 業務의 構造化 정도, 組織의 分權化 정도, 시스템의 革新性, 部署間 關聯性, 프로젝트의 規模, 組織의 資源能力, 使用者部署의 能力 등이 狀況要因으로 고려될 수 있다.

이에 대한 實證的 研究은 追後 遂行할 計劃이다.

參 考 文 獻

1. 國內文獻

- (1) 김강호, 배창기, “MIS개발계획과 접근방법에 관한 연구”, 산경연구 제1권 제1호 경북산업대학 지역산업경영연구소, 1990.9.
- (2) 김영효, 회계정보시스템, 개정증보 2판, 세명서관, 1990.
- (3) 박 연, “시스템 개발접근방법의 비교분석 연구”, 조선대학교 대학원, 박사학위 논문, 1990.2.
- (4) 소영일, 이종민, 경영관리를 위한 컴퓨터원론, 법영사, 1987.
- (5) 소영일, 이종민, MIS원론, 법영사, 1989.
- (6) 유관희, 회계정보시스템, 나남, 1990.
- (7) 이영환, 시스템분석과 설계, 법영사, 1989.
- (8) 이진주의 4인, 경영정보시스템, 단산출판사, 1989.

2. 外國文獻

- (9) AAA, “Report of the Committee on Accounting and Information Systems”, The Accounting Review, Vol.46, 1971.
- (10) Cushing, Barry E., Accounting Information Systems and Business Organization, 2nd ed., Addison Wesley Publishing Co., 1987.
- (11) Davis, G.B. and Olson, M.H., Management Information Systems: Conceptual Foundations, Structures, and Development, McGraw-Hill, 1985.
- (12) Donovan, J.J. and Madnick, S.E., “Institutional and Ad Hoc DSS and Their Effective Use”, Data base, Winter, 1977.
- (13) Gomaa H. and D.B.H. Scott, “Prototyping as a Tool in the Specification of User Requirements”, The Proceedings of the 5th International Conference on Software Engineering, 1981.
- (14) Hicks, Jr. J.O., Management Information Systems: A User Perspective, West Publishing Co., 1984.
- (15) Jenkines, A.M., “Prototyping: A Methology for the Design and Development of Application Systems”, Working Paper, School of Business, Indi-

- ana University, 1983.
- (16) Keen, P.G.W., "Adaptive Design for Decision Support Systems", Database, Fall, 1980.
 - (17) Mariam Alavl, "An Assessment of the Prototyping Approach to Information Systems Development", Communications of the ACM., 27:6 June 1984.
 - (18) Moscové, Stephen A and Mark G. Simkin, Accounting Information Systems: Concepts and Practice for Effective Decision Making, New York: John Wiley and Sons Inc., 1981.
 - (19) Naumann, J.D. and Jenkins, A.M., "Prototyping: The New Paradigm for Systems Development", MIS Quarterly, Sep. 1982.
 - (20) Prizant, A., "Is Prototyping Counter Productive?", Data Processing, Vol. 28, No.7, Sep. 1986.
 - (21) Rahman, M. and Maurice H., Accounting Information Systems, Prentice-Hall International, Inc., 1988.
 - (22) Wu G.H., Accounting Information Systems: Theory and Practice, McGraw-Hill, 1983.