

# 소프트웨어의 監査技法에 대한 研究

— 費用을 중심으로 —

## A Study on Audit Techniques of Software

— Emphasis on Costs —

卞 容 煥\*

(Byun, Yong Hwan)

### 目 次

- I. 序 論
- II. 電算資料處理監査
  - 1. 具現以前の 電算監査
  - 2. 具現以後의 電算監査
- III. 電算監査技法
  - 1. 靜的離散技法
  - 2. 連續竝行技法
- IV. 監査技法에 관련된 費用分析
- V. 要約 및 結論

## I. 序 論

전체시스템을 통괄하여 생산활동을 하는 소프트웨어는 會計情報시스템(accounting information system)에서 기술적으로 가장 중요한 기능을 맡고 있는 요소들중의 하나로 볼 수 있다. 소프트웨어를 設置하거나 作動하는 과정에서 잘못된 점이 있으면 전체 회계정보시스템의 信賴度を 구조적으로 낮추는 결과가 발생하게 된다. 회계정보시스템이라는 회계학의 한 분야가 근자에 태동하게 된 중요한 원인중의 하나는 회계정보처리를 과거에는 수작업으로 처리할 수 밖에 없었으나 최근 급속한 소프트웨어의 발달로 많은 회계정보처리를 신속하게 전산처리할 수 있게 된 때문이다. 따라서 소프트웨어의 신뢰도는 전체 회계정보시스템의 효율과 효과를 좌우하는 가장 큰 변수중의 하나라고 볼 수 있다.

현대화된 회계정보시스템을 감사하는데 있어서는 소프트웨어의 감사가 거의 필수적이다. 會計情報의 電算처리가 점차 확대되어 가고 있는 추세를 감안해 보면 電算資料處理監査(EDP auditing: electronic data processing auditing)가 전체감사에서 차지하는 비중도 점차 확대되고 있음은 당연하다. 그런데 소프트웨어의 감사에는 과거의 전통적 감사기법이 직접 통용되기 힘들므로 전산자료처리분야에 특유한 기법이 활용되어야 한다.<sup>1)</sup>

소프트웨어의 감사에 대해서 학문적으로는 電子計算學과 監査의 두가지 분야가 관련된다. 일반적으로 전산학자 및 기술자들은 소프트웨어의 保證(software certification)과 信賴度(reliability)에 초점을 맞추어 연구 또는 작업을 실시한다. 그리고 이러한 일들은 사용자(user)의 요구에 얼마나 잘 부응될 수 있는가에 중점을 두어 대체로 다음의 두가지 기법을 사용하여 진행된다. 첫째 사용자가 요구한 明細(specification)를 수행할 수 있도록 명령문을 有效化(validation)시키는 기법이며 둘째 확률적 모형에 기초하여 소프트웨어의 신뢰도를 측정하는 기법이다. 전산관련종사자들은 근본적으로 새로운 模型의 開發과 이의 經濟的 壽命을 연장하는데 주된 관심을 가지고 있는 반면 정확성(correctness)에 대한 관심도는 상대적으로 낮은 경향을 보인다. 이를테면 최근까지도 전산관련 학술논문집에 소프트웨어의 정확성을 확실하게 보증하기 위한 檢定기법이 발표되는 경우는 별로 없었다. 그러나 構造化 設計(structured design) 및 구조화 프로그래밍(structured programming)과 관련된 개념 및 기법이 소프트웨어의 信賴度 연구에 도입되는 경우가 많아짐에 따라 궁극적으로는 소프트웨어의 정확성을 향상시키기 위한 시도가 확산될 것으로 추정된다.

1) Wilkinson, Joseph W. Accounting and Information System, John and Wiley M Son, Inc., 1991, pp.770-799.

감사인의 주된 관심사항은 소프트웨어의 監査可能性과 信賴度라고 할 수 있다. 소프트웨어에 대한 감사가 이루어질 수 없다면 電算處理기법을 사용하는 회계정보시스템 전체의 감사도 불가능할 것이다. 소프트웨어의 신뢰도가 낮다면 시스템의 内部統制구조 자체가 구조적 결함을 안고 있는 것으로 해석될 수 있다.<sup>2)</sup> 감사인들중에서 소프트웨어의 감사와 관련된 감사인들은 그 업무의 특성상 電算資料處理監査人(이하 『전산감사인』이라고 함)으로 따로 분리될 수 있다. 전산감사인은 전산관련종사자들보다 훨씬 實用的인 면에 업무의 초점을 맞추고 있다. 전산감사인은 소프트웨어의 업무수행과정 및 결과가 지정된 소프트웨어 業務明細에 준하고 있는가를 평가하는 책임을 가지고 있다.<sup>3)</sup> 이러한 작업은 가능한 한 소프트웨어의 開發단계에서부터 실시되어야 한다. 이는 소프트웨어가 실질적 업무를 수행하기 이전에 현실적인 업무와 구분된 獨立的인 상태에서 信賴度を 檢定하는 것을 의미한다. 이렇게 함으로써 정보시스템이 핵심적인 기능인 소프트웨어가 의도했던 論理(logic) 및 明細에 맞춰 具現(implementation)되었는지 여부에 대한 확신을 가질 수 있다. 소프트웨어의 구현이후(post-implementation)에 감사인들의 주된 관심사항은 소프트웨어 信賴度の 繼續성과 소프트웨어에 대한 人爲的인 영향의 존재여부를 점검하는데 있다. 일정시점에서 소프트웨어가 완벽했다고 하더라도 시스템상의 문제 또는 人爲的인 조작 등으로 인해 일정기간 후에는 얼마든지 원래의 정보처리목적에서 離脫된 업무를 수행하게 될 수도 있기 때문이다.

본 연구에서는 전산화된 회계정보시스템의 감사에서 중요한 부분을 차지하는 전산 감사와 관련된 사항들을 살펴보고자 한다. 이를 위해 우선 電算監査의 내용을 정리해 보도록 한다. 그리고 그동안 개발되어온 소프트웨어의 감사기법들을 살펴보고 이 기법들의 適用分野, 그리고 技術 및 費用측면에서의<sup>4)</sup> 장단점을 비교분석하는 것을 주목적으로 한다.

## II. 電算資料處理 監査

과거에는 소프트웨어가 生産(production)활동을 시작하기 이전에 그 正確性を 點檢

2) 회계정보처리담당자가 실수로 잘못 처리한 자료는 일반적으로 비반복적 오류의 성격을 가지므로 그 문제가 일회성에 그치지만 소프트웨어에 결함이 있으면 오류가 반복적으로 발생하므로 지속적으로 정보시스템 전체에 구조적인 문제를 안겨주게 된다.

3) Vasarhelyi, M. A. and T. W. Lin, *Advanced Auditing : Fundamentals of EDP and Statistical Audit Technology*, Addison-Wesley Co., 1988, pp.540-554.

4) 전산감사기법들의 기술적 측면, 사용빈도 등에 대한 연구는 있었으나 경비구조에 대한 연구는 거의 없었다.

하기 위한 試驗과정은 중요한 의미가 없었다. 병렬처리(parallel processing)시스템하에서는 소프트웨어의 事前的인 正確도가 필수적으로 요구되지는 않았기 때문이다. 잘못 만들어진 소프트웨어가 실질 업무에 이용되는 경우에는 산발적으로 프로그램 缺陷(program bugs)이 나타난다. 그런데 병렬처리 시스템하에서의 프로그램 결함발생은 그렇게 큰 문제가 되지 않는다. 이는 이러한 정보처리 시스템이 대부분 하나의 입력화일(input file)과 둘이하의 생산(production)프로그램 및 하나의 출력화일(output file)을 가지고 있기 때문이다. 따라서 이런 구조하에서는 소프트웨어의 구현전뿐만 아니라 具現後에도 誤謬를 발견하고 그 내용을 파악하여 적절한 수정을 가하는 작업이 그다지 어려운 일이 아니었다.

최근에는 많은 정보처리시스템들이 온라인(online), 實時間(realtime) 및 自動連鎖형태(automated chaining fashion)<sup>5)</sup>로 개발되고 있다. 이러한 시스템하에서는 連續적으로 관련된 거래들을 처리하고 更新(update)시켜나가기 때문에 이미 실행중에 있는 소프트웨어에 誤謬가 있다면 이를 발견하고 수정하는 작업은 매우 어려울 뿐만 아니라 결함의 개선이 永久的으로 안될 가능성도 있다. 따라서 예전과는 달리 전산감사의 범위는 소프트웨어의 구현후 생산단계에서만 아니라 소프트웨어의 구현이전단계로 확대되었다고 볼 수 있다. 또한 경영활동에 필요한 基本 財務資料가 電算으로 처리되는 추세가 확산되고 있으므로 소프트웨어의 결함은 外部監査, 財務意思決定, 일상업무 등 조직과 관련된 경영환경전체에 영향을 줄 수 있다. 따라서 전산감사는 外部監査뿐만 아니라 内部監査를 위해서도 더욱 필요한 활동영역이 되어가고 있다. 실지로 소프트웨어에 결함이 있으면 이로 인한 조직의 손실은 상당한 규모에 이를 수 있다는 인식이 경영층 저변이 확산되고 있는 추세이다.<sup>6)</sup>

電算監査의 기능을 잘 이해하기 위해서는 우선 전산감사의 目的을 살펴볼 필요가 있다.

정의-전산감사의 목적은 정보시스템이 원래 조직이 의도했던 통제기준에 준하여 신뢰성있게 작동되고 있는지를 알아보는 데 있다.<sup>7)</sup>

이와 같이 電算監査의 目的은 근본적으로 전산자료처리활동과 관련된 統制構造를 評價하는 업무와 관련되어 있다. 그리고 이러한 업무는 궁극적으로 정보시스템의 신뢰도를 높이기 위한 것이다. 전산감사인의 업무는 일반적 분야와 특정 분야로 나누어 살펴볼 수 있다. 일반적인 분야는 조직내에 있는 전산기능들간의 葛藤的 要素를 파악하고

5) 자동연쇄형태의 예로서 판매와 관련해서는 신용도조사, 청구서, 재고처리, 외상매출금관리, 연수증발급 등의 업무가 자동적으로 연관됨을 들 수 있다.

6) Allen, B., "The Biggest Computer Frauds: Lessons for CPA's" Journal of Accountancy, Volume 143, Number 5, May 1977, pp.52-62.

7) Mair, W. C., Davis, K. W., and Wood, D. R., Computer Audit and Control, Q. E. D. Information Science, Institute of internal Auditors, Inc., Altamonte Springs, Florida, 1977, p.20.

이러한 문제점들이 잘 조정되고 있는지를 평가해 보는 것이다. 특정분야는 각 전산부문에 존재하는 특유한 기능을 점검해 보는 것으로서 이를테면 프로그램내에 적절한 편집점검(edit check)기능이 포함되어 있는지를 확인해 보는 일 등을 예로 들 수 있다. 그리고 이러한 전산감사인의 통제구조평가작업은 결국 정보시스템을 통해서 처리되는 자료의 信賴度를 향상시키기 위한 것으로 볼 수 있는 것이다.

전산감사인은 정보시스템이 조직의 활동을 잘 지지할 수 있도록 하기 위해서 소프트웨어가 원래의 명세에 맞춰 작동되고 있는가를 지속적으로 살펴보아야 한다. 이러한 목적을 달성하기 위한 감사기법은 그 동안 여러가지 형태로 개발되어 왔는데 감사인은 소프트웨어의 性格 및 壽命주기(life cycle)에 따라 각 상황에 적절한 감사기법을 선택해서 사용하여야 한다. 소프트웨어의 수명주기 및 각 단계별로 필요한 감사내용은 <그림 1>과 같이 요약되어질 수 있다.

감사인의 활동성격상 전산감사의 내용은 <그림 1>에서 보는 바와 같이 소프트웨어의 구현전과 구현후(pre- and post implementation)로 나누어 파악해 볼 수 있다. 소프트웨어의 具現前 電算監査의 주된 목적은 소프트웨어의 개발과정이 조직의 情報處理要求사항에 부응하고 있는지, 그리고 개발과정에 있어서 統制機能이 적절히 내장되는지를 평가하는데 있다. 소프트웨어의 具現後에 전산감사인은 소프트웨어의 管理維持와 관련된 주기적 점검에 업무의 중점을 두게 된다.

## 1. 구현이전의 전산감사

전술한 바와 같이 최근에는 소프트웨어의 구현이전부터 전산감사의 역할이 많이 요구되어지는 추세에 있다. 具現以前의 전산감사인의 활동은 準備단계, 統制활동단계 및 평가단계로 나누어 살펴볼 수 있다.

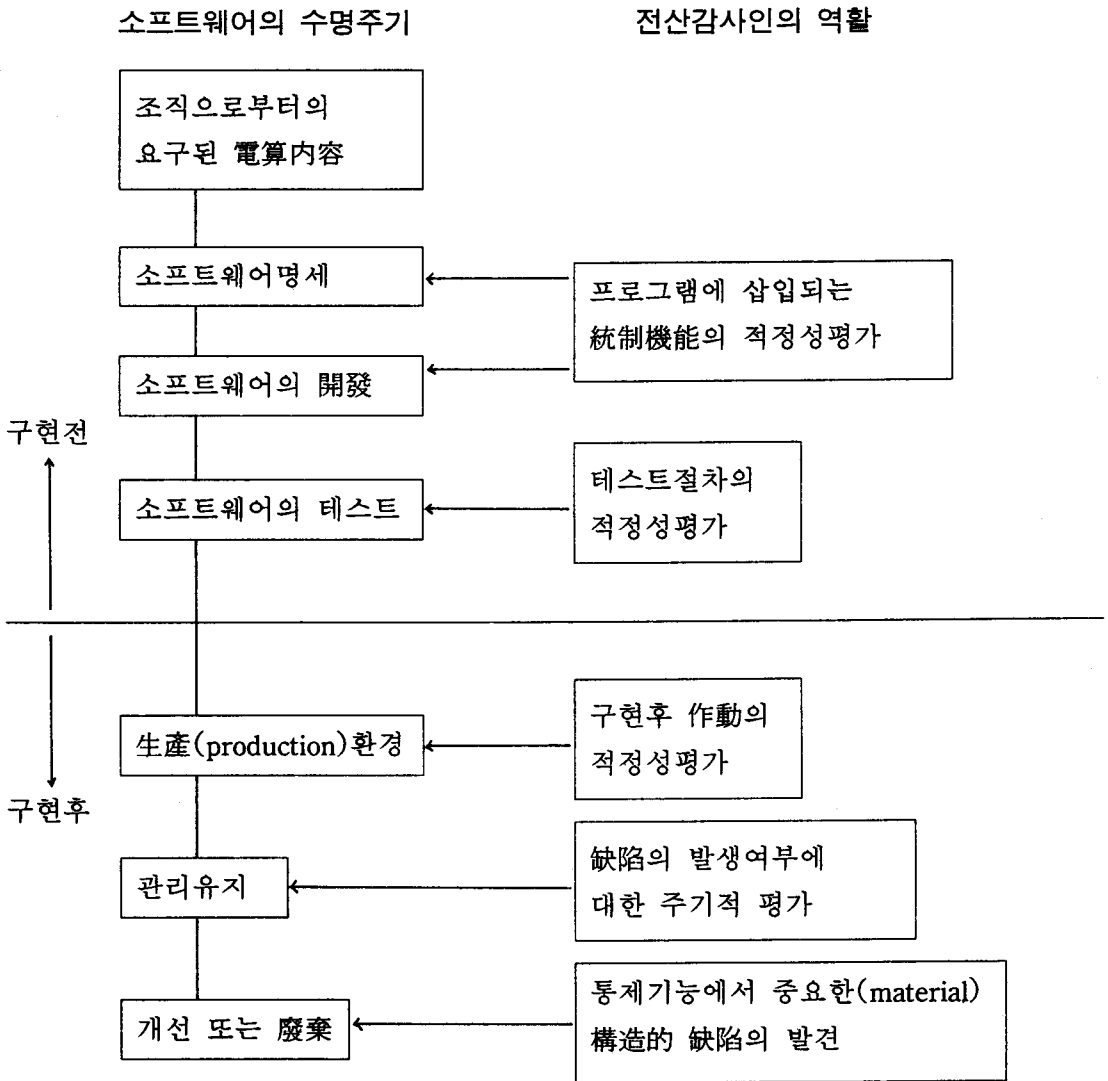
### 1.1 준비단계

준비단계에서 감사인은 우선 사용자(user)와 소프트웨어개발팀사이에 충분한 의사소통이 되어 관련당사자 모두가 프로젝트의 內容, 性格, 開發기간 등 모든 중요한 사항에 대해 이해가 제대로 되어 있는지를 확인해야 한다. 또한 소프트웨어개발의 수명주기에 따라 감사인은 각 단계별로 점검해야할 요소들을 선정하여야 한다. 준비단계에서 감사인의 역할은 다음과 같이 요약할 수 있다.

첫째, 모든 관련당사자가 소프트웨어개발 프로젝트의 目的을 제대로 理解하고 있는지를 확인해야 한다.

둘째, 사용자가 소프트웨어의 설계 전반에 걸친 중요한 요점들에 대해 이해가 되어 있는지를 확인해야 한다.

〈그림 1〉 소프트웨어의 수명주기에 따른 電算監査內容



셋째, 사용자가 바라는 업무를 수행할 수 있는 소프트웨어의 개발가능성과 이에 따른 수익비용분석을 점검해야 한다.

넷째, 개발계획 및 豫算이 현실적으로 달성가능하게 작성되어 있는지 살펴보아야 한다.

다섯째, 소프트웨어 수명주기의 段階別 評價方法을 설정해야 한다.

이상의 다섯가지 역할은 소프트웨어가 사용자의 목적에 잘 준거하여 개발되는지를 평가해 보는 기능과 관련되어있다. 감사인이 소프트웨어의 분석가(analyst)나 설계자의 기능을 가질 필요는 없다. 감사인의 주된 관심사는 개발단계에서의 결함이나 오해를 지적하기 위해 개발계획과 관련된 주요 서류를 점검해 보는데 있다. 소프트웨어 개발

계획은 조직통제기능의 중요한 골격을 만들어 나가는 일로 볼 수 있다. 이런 관점에서 볼 때 감사인의 임무는 소프트웨어개발계획 및 과정이 얼마나 조직의 통제구조에 잘 준거해서 이루어지고 있는가를 평가하는 것이라고 볼 수 있다. 이 임무를 수행하기 위한 감사방법으로는 質疑, 說問調查, 相談 등을 들 수 있다.

## 1.2 통제단계

통제활동과 관련하여 감사인이 맡는 역할은 설계와 통제의 두가지 단계로 구분하여 파악해 보는 것이 편리하다. 設計단계에서는 統制構造의 適合性を 평가하고 檢査단계에서는 통제방법의 效果를 평가한다.<sup>8)</sup>

감사인들은 소프트웨어의 설계단계에서부터 적극적으로 참여해야 되는 경우가 많다. 이 경우 소프트웨어시스템을 위해 고안된 통제방법이 적합하지 않다면 감사인은 통제 전문가로서 이를 개선하기 위해 소프트웨어의 明細에 영향을 줄 수 있는 능력 또는 권한을 가지고 있어야 한다. 이 시점에서 감사인의 주된 관심은 자료보관의 안전성, 자료사용 및 소프트웨어작동의 承認방법, 豫備(backup), 保有(retention) 등에 있다. 점검내용으로서 편집(edit), 論理(logic), 조정(reconciliation) 및 보고(reporting)기능의 검사를 들 수 있다. 통제기능이 늦게 점검될수록 검사와 관련된 업무량이 늘어나 관련비용이 높은 쪽으로 증대하므로 감사인은 소프트웨어의 구현전 시점에서 적극적으로 통제활동에 참여할 필요가 있다. 통제기능이 늦게 점검되는 경우 각 단계별로 엄청난 쪽의 비용상승이 예상되며 이는 Kodak Corporation의 경험에 의해 나타난 바 있다.<sup>9)</sup> 이를테면 시스템명세가 결정되기 이전에 통제기능을 점검하는 것이 가장 바람직한데 시스템명세가 결정된 후에는 그 비용이 네배가 소용되고, 프로그램이 완성된 후에는 여덟배, 시스템전체가 테스트된 후에는 열두배, 그리고 시스템이 구현되고 난 후에는 열 여섯배의 비용이 소요된다는 것이다.

검사단계에서는 각 통제수단들이 시스템내에서 제대로 작동될 수 있도록 내장되었는지를 확인하여야 한다. 이는 통제수단들이 원래 의도된 대로 작동될 수 있는지에 대한 확신을 가지기 위해서 시스템 전반에 걸쳐 관련된 통제수단들을 검사해 보는 것을 의미한다. 전체적인 검사단계에서 감사인은 사용자와 함께 점검을 할 수도 있다. 사용자가 전산시스템에 의해 처리되는 자료를 잘 이해할 수 있다면 檢査計劃의 現實性を 높이는데 상당한 도움이 될 수 있기 때문이다. 만약 소프트웨어 개발팀에 의해 작성된 통제수단의

8) 통제방법에 대한 완벽한 평가는 불가능하므로 주로 유사 내지는 가정에 의한 추론을 사용하여 평가한다. 확률적 추론은 보편화되지 않고 있다.

9) Hannye, L. G., "Auditors and Dp'ers Benefits from Association in the Systems Development Process," The Internal Auditor, Volume 6, December 1977, pp.67-70.

검사계획이 적절하지 않다고 판단되면 통제전문가인 감사인이 전체적인 검사계획을 독자적으로 작성하여 실시해 볼 수도 있다.

통제구조의 평가에 사용되는 감사기법은 감사인이 얼마나 적극적으로 참여하는가에 따라 달라질 수 있다. 전체적인 통제구조의 적정성을 평가하기 위해서는 일반적으로 감사인은 설문지를 사용하여 통제구조가 시스템내에 제대로 포함되었는지를 알아보는 경우가 많다. 응용통제구조(application control structure)내에서의 통제수단의 강점과 약점을 파악해 보기 위해서는 흐름도(flow chart)를 이용하는 것이 효과적이다. 만약 감사인이 검사계획의 중요한 구성원이라면 보다 적극적인 감사기법이 사용될 수도 있다.

### 1.3 評價단계

소프트웨어의 평가활동단계에서는 소프트웨어의 설계가 소프트웨어명세에 적합하도록 이루어졌는지를 알아보는데 중점을 두게 된다. 첫째, 거래를 입력시켰을 경우에 소프트웨어가 원래의 소프트웨어명세에 준해서 모든 내용을 일관되게 처리해 나가는가를 점검해야 한다.

둘째, 여러 종류의 거래를 처리해 나갈때 원래의 명세와는 다르게 처리가 되는 거래 내지는 부분이 있는가를 조사해야 한다.

평가단계에서 감사인의 참여수준은 전체적인 검사계획의 내용에 따라 달라진다. 만약 소프트웨어 개발팀에서 이상의 내용들을 적절히 점검할 수 있는 검사를 한다면 감사인은 검사내용과 결과가 요약된 서류들을 살펴보는 정도의 일만 하면 된다. 그런데 전술한 검사내용이 검사계획에 제대로 포함되어 있지 않다면 감사인은 평가단계에서의 감사 목적을 달성하기 위하여 적극적으로 검사계획의 수립에 참여해야 한다. 소프트웨어명세에 준한 작동을 하는지를 알아보는 一貫性검사와 통제수단의 적절한 내장여부를 알아보는 통제검사는 편리상 동시에 실시될 수도 있다. 이 경우 다음 장에서 소개될 자동화된 감사기법들이 유용하게 사용될 수도 있다.

소프트웨어 구현전 단계에서 감사인이 적극적으로 참여하게 되면 구현후에 감사인이 감사활동을 수행함에 있어 지표로 삼아야 하는 소프트웨어의 信賴度를 事前的으로 파악할 수 있다. 구현전 감사는 또한 정보처리시스템이 원래 조직에서 목적인 바에 준하여 작동될 수 있는지에 대한 獨立的인 評價를 가능하게 한다.<sup>10)</sup>

10) 온라인, 실시간시스템의 사용이 보편화됨에 따라 구현이후에 소프트웨어를 독립적으로 평가하려면 조직의 업무에 막대한 지장을 초래하게 된다.



## 2. 具現以後의 電算監査

소프트웨어의 구현이후에 감사인은 정보시스템 및 소프트웨어와 통제구조의 평가자로서 활동한다. 감사인은 구현된 시스템이 사용자와 프로젝트팀이 설계한 시스템명세에 맞추어 작동하는지를 평가하게 된다. 이를 위한 감사기법으로서는 질의 및 관찰이 일반적으로 사용된다. 또한 감사인은 구현이후의 시스템에서 작동하는 논리 및 통제수단이 구현이전에 의도한 대로 작동되는지를 살펴보아야 한다. 시스템의 생산(production)활동이 진행됨에 따라 원래의 表現(representation)이 변동될 수도 있기 때문이다. 감사인은 이러한 변동을 파악하고 그로 인한 영향을 반영할 수 있도록 검사절차를 更新(update)하여야 한다. 구현이후에 사용되는 감사기법들은 상당수가 구현이전에 사용되는 기법들과 동일하다.

감사인은 전산정보처리시스템의 신뢰도를 판별하는데 기본적인 책임이 있다. 그 중에서도 감사인은 생산프로그램의 신뢰도에 가장 비중을 많이 두어 일을 하게 된다. 소프트웨어의 구현이전에는 靜的(state-in-time)이고 獨立的(independent)인 확인작업이 이루어지며 구현이후에는 動的(dynamic)이고 連續的(continuous)인 감사업무가 이루어진다.

## III. 電算監査技法

전산감사인들이 활용할 수 있는 감사기법은 근자에 와서 여러 방향에서 다양하게 개발되어지고 있다.<sup>11)</sup> 본 논문에서는 소프트웨어의 응용에 있어 統制(controls)와 論理(logic)의 검정(verification)에 사용되는 감사기법들에 한정하여 설명을 하도록 한다.

감사기법들은 그 성격상 크게 정적이산기법(static/discrete)과 연속병렬기법(continuous/concurrent)의 두가지로 분류될 수 있다. 靜的離散기법을 사용하면 감사인은 특정시점에서만 검정작업을 할 수 있다. 따라서 감사를 자주 하지 않으면 소프트웨어 신뢰도의 일관성에 대한 추정을 할 수가 없다. 반면 연속병렬기법하에서는 감사인은 연속적으로 생산처리가 진행되는 상황에서도 소프트웨어의 신뢰도를 점검해 볼 수 있다. 한편 비용은 정적이산기법이 상대적으로 저렴하다. 따라서 소프트웨어의 수명주기와 생산활동의 계속여부에 따라 비용을 고려하여 각 상황에 맞는 기법을 선택해야 할 것이다.

11) Cash, J., Bailey, A., and Whinston, A., op. cit., pp.813-832

## 1. 정적이산기법

정적이산기법은 생산환경에 아무런 영향을 끼치지 않고 사용할 수 있는 감사기법이다. 그러나 이 기법은 生産活動을 일시 정지시켜야만 사용할 수 있는 한계를 가지고 있다. 이 기법은 구현이전뿐만 아니라 구현이후에도 생산활동이 일시적으로 정지될 수만 있다면 적용되어질 수 있다. 그리고 이 기법은 특정 시점에서의 소프트웨어의 상황을 점검해보는 것이므로 감사행위 사이의 기간에 관련된 신뢰도는 점정할 수가 없는 단점이 있다.

### 1.1 자료검사(test data)

자료검사기법은 인위적으로 만든 거래를 입력하여 소프트웨어가 원래의 명세에 준하여 제대로 처리하는지를 알아보는 기법이다. 유효한 자료와 유효하지 않은 자료가 함께 거래내용에 포함되며 소프트웨어에 문제가 없다면 유효한 자료는 정확하게 처리되는 반면 유효하지 않은 자료는 처리가 되지 않는다. 이 기법을 효과적으로 사용하기 위해서는 우선 입력되는 기록의 配置(layout), 처리하는 論理, 유효한 거래와 유효하지 않은 거래에 대한 숙지가 잘 되어 있어야만 한다. 그리고는 원래 의도한 대로 소프트웨어의 논리가 작동되는지를 알아보기 위해 가공적인 거래들을 준비해야 한다.

감사인은 가공적인 각 거래의 예상결과를 사전에 미리 정확하게 계산하여 이를 프로그램에 의해 처리된 결과와 비교할 수 있도록 준비를 해 두어야 한다. 만약 프로그램에 의해 처리된 결과가 감사인이 수작업으로 계산한 결과와 다르다면 이는 소프트웨어가 서류의 명세대로 움직이지 않는다는 것을 의미한다. 그런데 이러한 결과는 감사하고자 하는 소프트웨어의 논리가 잘못되어서 나타난 것으로 해석하는 것이 일반적이지만 감사활동에 사용되는 명세가 更新(update)되지 않아서 나타날 수도 있다. 따라서 이 기법에 의해서는 프로그램내에 있는 論理經路(logic path)와 통제를 완전히 테스트할 수 있다고는 볼 수 없다. 또한 프로그램의 결과가 수계산의 결과와 일치하더라도 프로그램이 원래의 명세대로 입력자료를 처리하였다고 단정할 수도 없다. 프로그램자체내의 잘못된 부분이 있더라도 의도한 바와 일치하는 결과치를 산출하는 경우도 있기 때문이다.<sup>12)</sup> 요약하면 자료검사기법에서는 테스트되는 소프트웨어의 결과치가 구현후에 실질적으로 생산활동을 하고 있는 프로그램의 산출치라고 단정하기는 힘든 문제점이 있다.

### 1.2 병렬 모의실험(parallel simulation)

병렬모의실험은 감사인이 감사프로그램을 작성하여 소프트웨어를 모의실험하는 기

12) 이를테면 잘못된 급여프로그램이 다른 프로그램들과 호환이 됨으로서 결과적으로 제대로 된 결과치를 만들어 낼 수도 있다.

법이다. 감사인은 거래화일을 모의프로그램을 통해서 출력된 결과(output)를 동일한 거래화일을 감사대상이 되는 소프트웨어에 의해 처리시켜 나오는 결과와 비교함으로써 소프트웨어의 신뢰도를 평가한다. 이 감사기법을 효과적으로 사용하기 위해서는 감사인은 감사대상이 되는 소프트웨어를 심도있게 숙지할 수 있어야 한다.

감사범위에는 소프트웨어의 전체적인 논리가 포함될 수도 있고 일부분만 선별되어 포함될 수도 있다. 모의실험에는 일반화된 감사소프트웨어언어(generalized audit software language)가 주로 사용되나 이 언어가 감사처리과정과 특별한 관계를 가지는것은 아니다. 작성된 모의프로그램에 거래파일을 처리하면 특정 결과가 출력된다. 감사인은 모의프로그램의 결과와 소프트웨어에서 동일한 과정을 거쳐 출력되는 결과를 비교하게 된다. 만약 이 두 결과사이에 상이점이 있다면 이는 모의프로그램이나 소프트웨어중에 잘못된 부분이 있다는 것을 의미한다. 그런데 동일한 결과를 나타낸다고 하더라도 소프트웨어가 원래의 명세에 맞게 되어 있다고 단정짓기는 힘들다. 이는 사용중인 소프트웨어가 원래의 明細에 포함되지 않았던 모듈(module)을 가지고 있을 수가 있는데 이런 요소는 모의화되지 않으므로 검사되어질 수가 없기 때문이다. 또한 생산프로그램을 모의화했을 때 모의화된 프로그램이 생산활동에 사용되는 프로그램과 늘 일치한다는 보장도 없다.

## 2. 連續並行(continuous and concurrent)技法

연속병행기법은 소프트웨어의 설계단계 또는 수정단계에서 감사가능성(auditability)을 내장(built)하는 원칙에 준하여 구현된다. 이 감사기법들은 소프트웨어가 실질적인 생산처리활동을 하고 있는 동안에 사용되어질 수 있는 장점을 가지고 있다.<sup>13)</sup> 이 기법들은 필요한 경우에는 특정 소프트웨어가 생산처리활동을 하고 있는 때 시점마다 사용되어질 수도 있다. 이 기법들을 사용하면 소프트웨어의 일상적인 생산처리활동이 저해되지 않는다. 또한 정보시스템에 근무하는 사람들이 감사를 받고 있다는 사실도 모른다. 따라서 감사인은 본인이 편리하다고 생각되는 때에 조직의 정보처리활동에 피해를 주지않고 얼마든지 임의적인 감사활동을 할 수 있다. 또한 소프트웨어의 구현이전에 필요한 檢査 내지는 誤謬수정의 목적에도 이 기법들이 효과적으로 사용될 수 있다.

연속병행기법들을 내장시켜 사용하기 위해서는 소프트웨어에 약간의 수정이 가해져야 한다. 이러한 수정은 소프트웨어의 설계단계에서 이루어지는 것이 가장 바람직하다. 소프트웨어의 설계초기부터 이러한 감사기법을 내장하면 감사목적에 뿐만 아니라 일반적인 검사에도 편리하게 사용할 수 있으며 소프트웨어의 수명주기 전반에 걸쳐 활용할

13) Holly, Charles L., and Reynolds, K., "Auditor Concerns in an On-line Distributed Computer Network," Journal of Systems Management, (June 1984), pp.32-36.

수 있다.

연속병행기법들의 단점으로는 우선 상대적으로 높은 費用이 소요된다는 점을 들 수 있다. 비용은 실제적인 감사지출비용뿐만 아니라 시스템사용비용까지도<sup>14)</sup> 포함된다. 둘째 이러한 감사기법을 사용하기 위해서는 감사인이 상당한 수준의 電算技術을 소유하고 있어야 한다.

연속병행기법으로 분류되는 감사기법중에 대표적인 것으로는 종합검사설비(integrated test facility), 스냅샷(snapshot), 추적(tracing) 및 사상(mapping)을 들 수 있다.

## 2.1 綜合檢査設備

종합검사설비는 감사과정 및 해석방법에 있어 전술한 자료검사기법과 유사한 점을 가지고 있다. 감사인은 우선 감사대상이 되는 소프트웨어에 처리시킬 가공적인 조직과 거래를 만들게 된다. 감사인은 반드시 가공적인 거래의 처리결과를 사전에 계산하여 이를 소프트웨어에서 동일한 거래를 실질적으로 처리한 결과와 비교할 수 있도록 하여야 한다. 이 기법을 제대로 사용하기 위해서는 정보처리시스템을 주의깊게 설계하는 일이 사전적으로 요구된다. 이를테면 소프트웨어가 수표나 현금지급을 처리하는 생산활동을 하고 있는데 가공적인 거래가 실질적인 거래와 구별이 안되어 관련없는 거래처에 현금지급을 하라는 명령을 유효화시키는 일이 발생해서는 안되기 때문이다.

이 기법의 가장 중요한 장점은 현실적으로 생산활동을 하고 있는 소프트웨어내에서 실질적인 거래들과 함께 가공적인 거래가 처리된다는 점이다. 따라서 현실적으로 생산 프로그램의 신뢰도를 평가하는데 있어 자료검사기법에서 가지고 있던 不確實性은 완전히 제거된다. 다른 감사증거 수집 및 평가방법은 자료검사기법과 동일하다. 요약하면 종합검사설비기법은 자료검사기법을 현실적으로 생산활동을 하고 있는 소프트웨어에 적용시킬 수 있게 한 기법이라고 볼 수 있다.

종합검사설비기법은 소프트웨어 감사 이외의 분야도 부수적으로 평가하는 장점을 가지고 있다. 감사에 사용되는 가공적인 거래가 실질적인 거래들과 동시에 시스템에 입력되어 처리되므로 감사인은 시스템전체가 원래의 명세에 준해서 작동되는지를 관찰해 볼 수 있기 때문이다.<sup>15)</sup> 따라서 이 기법을 사용하면 소프트웨어뿐만 아니라 시스템전체에 대한 평가를 할 수 있다.

14) 시스템사용비용은 일종의 기회비용개념이다. 즉 감사업무에 정보처리시스템의 일정부분을 할애함으로써 다른 일상적인 업무의 기계이용시간이 줄어들 수도 있기 때문이다.

15) 종합검사설비기법을 사용하면 거래의 승인, 전사(transcription), 오류보고, 오류수정 등 소프트웨어 뿐만 아니라 정보 시스템 전반에 걸친 업무내용이 관찰된다.

## 2.2 스냅샷

스냅샷기법은 특정시점에서 프로그램에 있는 중요한 意思決定變數들의 影響力을 파악하는데 중점을 두고 있다. 논리적 모듈(logical module)의 결과가 정확한지를 알아보기 위해서는 감사인은 하나의 모듈이 다른 모듈들과 연결될 때 실행되는 알고리즘(algorithm)과 관련된 모든 변수들과 자료값들을 관찰할 필요가 있다. 스냅샷기법은 특정간격사이에 있는 기억들의 이미지를 스냅사진을 찍는 것과 같은 방법으로 감사인에게 전달해 준다. 이 기법의 초점은 의사결정에 영향을 끼쳐서는 안되는 변수가 영향력을 행사하는 경우가 있는지를 알아보는 데 있다. 이 기법을 사용하면 특정의사결정과정에 포함된 모든 자료를 알아볼 수 있다. 물론 이 기법은 소프트웨어의 有效性을 확인하는 기능은 가지고 있지 않다. 단지 여러 종류의 자료들이 의사결정에 어떤 영향을 끼쳤는지 또는 처리되기 위한 자료로서 입력되었는지를 검정해 보는 기능을 가지고 있다.

이 기법을 효율적으로 사용하기 위해서는 프로그램내에 있는 중요한 의사결정시점에 대해 주의깊은 분석이 필요하다. 왜냐하면 프로그램 전체가 아니고 의사결정에 중요한 영향을 미치는 제한된 자료에 대해서만 포착이 가능하기 때문이다. 이 기법은 소프트웨어가 현실적으로 생산활동을 하고 있는 상황에서 의문시되는 정보처리결과가 나타난 경우에 그 원인을 파악하는데 매우 효과적이다.

## 2.3 추적

이 기법은 일반적으로 標準化된 프로그램언어로 작성되어 있다. 이 기법을 사용하면 거래를 처리하는데 있어 연속적으로 사용되는 명령(instruction)들의 흐름을 알아볼 수가 있다. 追跡技法은 프로그램의 검정보다도 臨界 經路(critical path)를 감사하기 위한 기법이라고 볼 수 있다. 이 기법을 효과적으로 사용하기 위해서는 우선 소프트웨어를 통해서 이루어지는 거래에 대해 탁상 추적(desk trace)을 실시해야 한다. 그리고는 이 탁상추적의 내용을 컴퓨터추적의 내용과 비교함으로써 임계경로에 대한 평가를 하게 된다. 두가지 방법에서 결과된 추적내용들이 동일하다면 이는 특정거래가 그 거래에 적합한 명령으로 구성된 경로를 따라 처리되었음을 의미한다. 그러나 개별적인 명령의 정확성에 대해 확신을 가질 수는 없는 한계가 있다. 만약 추적내용들이 동일하지 않다면 이는 감사인의 桌上追跡에 誤謬가 있었든지, 아니면 거래가 원래의 명세대로 시스템에서 처리되지 않았음을 의미한다.

일반적으로 이 기법하에서는 원시코드(source code)를 통해 거래를 추적하게 된다. 그러므로 감사인은 원시코드를 읽고 해석할 수 있어야 한다. 따라서 이 기법을 효율적으로 사용하기 위해서 감사인은 높은 수준의 電算技術能力을 소지하고 있어야 한다.

## 2.4 寫像

사상기법을 사용하면 거래화일을 처리하는데 있어 소프트웨어의 각 부분이 어떻게 작동되고 있는가를 요약해 볼 수 있다. 이를테면 실행되고 있지 않는 프로그램의 각부분들 (segments), 중앙처리(central processing unit)시간을 비정상적으로 많이 소비하는 처리단계들, 각 단계가 몇번이나 실행되었는가를 보여주는 원시코드 등을 알아볼 수 있다.

감사인원은 이 기법을 사용하여 출력된 결과를 처리되는 거래화일에 관련시켜 분석할 수 있어야 한다. 또한 이 기법에 의해 출력된 결과로부터 예외적인 현상을 발견할 수 있어야 한다. 이를테면 비정상적으로 중앙처리장치시간을 소모하는 단계, 특정 명령의 비정상적인 반복실행 등을 발견하여 현상을 평가하고 원인을 파악할 수 있어야 한다.<sup>16)</sup>

## IV. 監査技法에 관련된 費用分析

전술한 바와 같이 각 감사기법들은 適用範圍뿐만 아니라 효율과 효과면에서 그 내용을 달리하고 있다. 그런데 효율과 효과를 計數的인 방법으로 가장 잘 측정할 수 있는 기준은 각 기법에 소요되는 비용이라고 볼 수 있다. 따라서 각 감사기법을 현실적으로 사용하기 위해서는 각 기법에 소요되는 비용을 파악해 보아야 할 것이다. 그리고 특정 소프트웨어를 감사할 때 어느 기법을 적용할 것인가를 결정하는데 있어서는 적용가능한 각 기법들의 비용수익분석을 철저히 할 필요가 있다. 본 장에서는 각 전산감사기법들의 비용을 固定費와 變動費의 구조하에서 파악해 보도록 한다.

### 1. 자료검사

이 기법에 소용되는 固定費로는 소프트웨어 응용서류를 전반적으로 검토하는데 소요되는 비용만을 들 수 있다. 나머지 비용들은 變動費적 성격을 가지고 있다. 즉 검사 계획작성, 검사할 자료의 작성, 검사할 자료의 결과를 사전에 계산하는 작업, 검사자료를 수행시키는 일, 검사결과를 분석하는 일 등에 관련된 비용은 모두 변동비적 성격을 가지고 있다. 그리고 이러한 일들 대부분이 노동집약적인 구조를 가지고 있다. 그러므로 자료 검사기법에서는 소프트웨어의 복잡성 및 감사의 범위에 비례해서 비용이 발생된다고 볼 수 있다. 費用構造면에서 볼 때 자료검사는 고정비가 적게 소요되는 반면에 변동비가 많이 요구되는 감사기법이라고 할 수 있다.

16) Adams, D. L. "COMBI as an Audit Tool," EDPACS, Volume 2, Number 10, April 1975, pp.7-8.

## 2. 병렬모의실험

병렬모의실험에서는 고정비적인 성격을 가진 비용의 지출이 많다. 서류의 총체적인 檢討, 設計作業, 코딩, 檢査節次에 소요되는 비용이 모두 고정비이다. 그리고 가공된 거래의 처리시간 및 결과의 분석작업에 소요되는 비용은 변동비에 속한다. 작업의 내용에서 보듯이 병렬모의실험은 노동집약적인 감사기법이다. 자료검사기법도 노동집약적이라는 점에서는 비용의 성격이 유사하지만 병렬 모의실험에서는 상대적으로 컴퓨터처리시간이 많이 소모된다. 병렬모의실험에 소모되는 비용의 규모는 모의화되는 시스템의 복잡성에 비례한다. 병렬모의 실험기법의 비용은 자료검사기법에 비해 고정비용이 훨씬 높은 반면 變動費가 상대적으로 낮은 구조를 가지고 있다.

## 3. 綜合設備檢査

종합설비검사에서서는 시스템전체에 대한 전반적인 검토, 설계시간, 통제문제; 검사절차, 가공적인 조직의 작성 등에 관련된 비용들이 고정비적인 성격을 가진다. 한편 가공적인 거래의 작성, 거래량의 증가에 따른 처리시간의 증가, 사전적인 계산절차, 결과의 분석에 따른 비용들은 변동비적인 성격을 가진다. 이상의 작업내용에서 보듯이 종합설비검사도 노동집약적인 감사기법이다. 단 기계를 이용하는 작업이 상당부분 포함되어있다. 이 감사기법을 구현하기 위한 初期費用(initial cost)은 소프트웨어의 수명주기초기에 이 기법을 내장할수록 적게 소요된다. 이를테면 소프트웨어의 수정단계에서 이 기법을 내장하면 설계단계에서 하는 경우보다 훨씬 많은 초기비용, 즉 고정비용이 소요된다. 변동비는 생산활동시간에 처리함으로써 소요되는 비용만 제외하고는 자료검사기법의 경우와 동일하다.

綜合設備檢査는 이 감사기법이 소프트웨어 수명주기의 어느 시점에 내장되는가에 따라 변화는 있지만 대체적으로 매우 높은 고정비가 소요된다. 그리고 변동비도 다른 기법의 경우에 비해 상대적으로 높은 편이다. 이 기법에서 고정비가 특별히 높은 이유는 실질적인 생산활동 도중에 가공적인 거래를 처리해야 하므로 그 내장에 고도의 주의가 요구되기 때문이다. 특히 통제설계 및 구현의 경우 매우 주의깊은 감독과 검사가 필요하다.

## 4. 스냅샷

이 기법에는 固定費 성격을 지닌 업무들이 많다. 서류의 총체적인 검토, 중요한 의사결정시점의 분석, 스냅샷의 이용가능성 검토, 트리거링의<sup>17)</sup> 설계, 스냅샷 화일의 설계,

17) 그 자신의 제어에 의하여 미리 정해진 시간동안만 동작하는 회로동작의 시작을 말한다.

스냅샷기법을 내장하기위한 프로그램의 수정, 스냅샷 보고방법의 설계, 검사 등이 이 기법의 적용내용의 대부분을 차지함과 동시에 고정비적인 성격을 가지고 있다. 변동비로는 컴퓨터의 증분(incremental)처리시간 및 결과의 분석에 소요되는 비용을 들 수 있다.

이상의 비용형태를 볼 때 스냅샷은 상당히 勞動集約的인 기법임을 알 수 있다. 그리고 자동화된 트리거링 및 보고장치의 설계와 내장이 업무의 대부분을 차지하므로 이 기법은 높은 고정비와 낮은 변동비의 비용구조를 가지고 있다고 볼 수 있다. 스냅샷이 엄청난 횟수로 이루어지지 않는 한 컴퓨터의 증분처리시간에 소요되는 비용은 아주 미미하기 때문이다. 스냅샷기법의 높은 고정비와 낮은 변동비로 이루어진 비용행태는 병렬 모의 실험의 경우와 매우 유사하다.

## 5. 追 跡

추적기법에서는 추적대상이 되는 임계프로그램의 분석 및 프로그램의 수정에 소요되는 비용이 고정비적인 성격을 가지고 있다. 변동비적인 성격을 가진것으로는 증분 컴퓨터 처리시간 및 결과분석에 소요되는 비용을 들 수 있다. 이 기법에서는 고정비용이 매우 적게 소요되는 것으로 알려져 있다. 臨界프로그램의 분석단계에서 감사인의 참여가 필수적이지만 응용상태가 매우 높지 않고서는 감사인의 노력이 크게 소요되지 않기 때문이다. 이 기법은 고정비용이 매우 낮고, 반면에 매우 높은 變動費 構造를 잠재적으로 가지고 있다. 실질적으로 많은 연구에서 추적기법을 사용하면 이를 처리하는 시간(throughput time)이 타 기법에 비해 두배 이상 증가하는 것으로 지적되어 왔다.<sup>18)</sup> 따라서 추적기법을 사용하면 현실적인 생산처리활동이 많이 위축될 가능성이 있는 것으로 해석되어질 수 있다. 일반적으로 추적기법은 고정비는 가장 낮고, 변동비는 가장 높은 기법으로 알려져 있다.<sup>19)</sup>

## 6. 寫 像

이 기법에서 고정비적인 성격을 가진 것은 소프트웨어 모니터의 구입 내지는 임대, 혐로프로그램의 분석, 프로그램사용을 위한 설치시간 등에 소요되는 비용들을 들 수 있다. 변동비로는 증분 컴퓨터 처리시간 및 결과의 분석에 필요한 비용을 들 수 있다. 초기 단계에서 감사인의 적극적인 참여가 요구되지는 않지만 사상(mapping)을 위한 팩키지의

18) Institute of Internal Auditors, System Auditability and Control AuditPractices, Part 4, Institute of Internal Auditors, Inc., Altamonte Springs, Florida, 1977, pp.109-164.

19) Lientz, B. P. and Weiss, I. R., "Trade-Offs of Secure Processing in Centralized vs Distributed Networks," Computer Networks, Volume 2, Number 1, February 1978, pp.35-43.



구입 내지는 임대비용이 매우 높으므로 전체적인 固定費는 높은 편이라고 할 수 있다. 한편 사상작업에 소용되는 시스템소요시간은 실질적인 생산활동의 상당한 위축을 의미하나 그 정도가 추적기법의 경우보다는 상대적으로 낮은 것으로 알려져 있다. 요약하면 사상기법의 비용구조는 상대적으로 높은 고정비 및 변동비로 이루어져 있다고 볼 수 있다.

## V. 要約 및 結論

본 연구에서는 소프트웨어를 감사하는데 사용될 수 있는 여섯가지 감사기법에 대한 내용을 알아보았다. 각 전산감사기법에 대한 평가는 目的, 業務內容, 適用範圍, 총비용, 비용의 구조 등 다차원적인 방법으로 이루어져 총체적인 파악이 가능하도록 노력하였다. 각 전산감사기법들을 분석한 결과 전반적으로 뛰어난 특정기법은 없으며 모든 기법이 각각 상대적 장단점을 가지고 있는 것으로 판명되었다. 감사인들은 정보시스템의 환경, 기술적 요구사항, 감사의 목적, 감사경비 등 감사와 관련된 복합적인 환경요인들을 모두 고려하여 특정기법을 선택하여야 할 것이다.

追跡 및 寫像은 소프트웨어가 원래의 명세에 준해서 작동되는가를 알아보는 전산감사의 일반적인 목적을 명시적으로 만족시키기 위한 기법은 아니다. 따라서 전산감사의 전반적인 목표들을 충족시키기 위해서는 일반적 감사목적에 부합되는 기법이 함께 사용되어야 할 것이고 이 경우에 그 비용은 매우 높을 것이다. 사상은 패키지구입 또는 임대 때 높은 고정비용이 매우 높기 때문에 감사가 자주 있지 않는 경우에는 비효율적인 방법이다. 한편 추적기법은 변동비는 많이 소요되나 고정비가 매우 낮으므로 감사가 자주 있지 않는 경우에 매우 효율적인 방법이다.

소프트웨어 감사의 일반적인 목적, 즉 소프트웨어가 명세에 준하는지를 명시적으로 평가하는 기법은 자료검사, 병렬 모의실험 및 종합검사설비에 국한된다.

綜合檢査設備은 開發하는데 가장 많은 비용이 소모되는 것으로 알려져 있다. 이 기법을 개발하고 내장시키기 위해서는 상당수의 기술인력이 몇달동안 작업을 해야하는 경우가 많다. 이 기법은 固定費가 워낙 높으므로 臨界시스템(critical system)에만 사용하는 것이 일반적이다. 또한 기법의 개발 및 내장도 소프트웨어의 설계단계에서 행해져야만 한다. 소프트웨어의 구현후에는 이 기법의 개발과 내장에 엄청난 시간과 비용이 소요되기 때문이다. 설계단계에서도 많은 개발비용이 소요되므로 종합검사설비기법은 연속병행 통제가 필요한 임계시스템에만 국한시켜 적용하는 것이 좋다.

병렬 모의실험은 전산감사의 일반적인 목적인 프로그램 검정을 실시하는 기법이기는 하나 현실성이 약한 단점이 있다. 이 기법하에서는 서류화된 내용에 대해서만 모의실험이

적용될 수 있다. 따라서 서류가 持續적으로 완벽하게 更新되지 않은 한 감사를 실시하더라도 생산현장에서 실지로 작동되고 있는 소프트웨어를 검정했다고 단정하기는 힘들다. 그리고 생산프로그램의 모든 수정사항에 따라 감사프로그램도 수정되어야 하며 이 경우 고정비용의 계속적인 상승이 초래된다.

資料檢査技法은 잘만 운용하면 경비측면에서 상당히 유리한 기법이 될 수 있다. 이 기법에서 비용이 많이 소요되는 부분은 검사에 사용될 자료를 생성시키는 일이다. 그런데 최근에 효율적인 자료생성기(data generator)가 개발되어 이 작업에 소요되는 비용을 대폭 절감할 수 있게 되었다. 이 기법의 결정적인 단점은 靜的分析만 가능하다는 것이다. 그러나 감사간격을 줄이면 이러한 단점은 어느 정도 완화될 수 있으므로 경비면에서의 유리한 점을 고려하면 계속 實用的인 전산감사기법으로 사용될 수 있을 것이다.

이상 전산감사기법들은 技術的 또는 經濟的인 측면에서 각각 장단점을 달리하고 있으므로 각 기법의 우월성에 절대적인 우선순위를 부여하기는 어렵다. 따라서 감사적용 대상이 되는 소프트웨어의 수명주기 및 성격에 따라 그에 알맞는 기법을 적용할 수 밖에 없다. 대체적으로 綜合檢査設備나 寫像같이 固定費가 많이 소요되는 기법은 소프트웨어의 구현전과 구현후에 모두 사용되면서 감사빈도가 높은 시스템환경에서 경제성이 있다고 볼 수 있다. 한편 資料檢査나 追跡같이 變動費가 많이 소요되는 기법은 구현전 또는 구현후에 관계없이 감사빈도가 낮은 시스템환경에서 사용하기에 적합한 기법이라고 볼 수 있다.

어떤 감사기법을 사용하는가는 전적으로 監査目的 및 시스템環境에 달려있다. 단 각 기법마다 기술적 요구수준과 경비의 구조가 다르므로 감사인은 이를 모두 숙지하여 주어진 상황에 적절한 기법을 사용할 수 있도록 해야 한다. 이를테면 수익비용의 측면에서만 보면 자료검사기법이 선호될 수도 있으나 정적분석의 결함으로 인해 온라인 시스템에 사용되기 힘든 단점이 있다. 필요한 경우에는 여러 종류의 기법들을 함께 사용할 수도 있을 것이다. 효과적인 전산감사계획을 작성하기 위해서는 각 전산감사기법들에 소요되는 경비 및 기술적 내용을 정확하게 이해하고 계속하여 개발되어 나오는 감사 기법들을 숙지해 나가는 노력이 필요하다.

## 參 考 文 獻

- Adams, D. L. "COMBI as an Audit Tool," *EDPACS*, Volume 2, Number 10, April 1975, pp.7-8.
- Allen, B., "The Biggest Computer Frauds : Lessons for CPA's" *Journal of Accountancy*, Volume 143, Number 5, May 1977, pp.52-62.

- Biggs, S., W. Messier, Jr., and J. Hansen, "A Descriptive Analysis of Computer Audit Specialists' Decision-Making Behavior in Advanced Computer Environment", *Auditing : A Journal of Practice and Theory*, Spring 1987, pp.1-21.
- Cash, J., Bailey, A., and Whinston, A., "A Survey of Techniques for Auditing EDP-Based Accounting System," *Accounting Review*, Volume 111, Number 4, October 1977, pp.813-832.
- Dowell, C. and J. Hall, "EDP Controls with Audit Cost Implications", *Journal of Accounting, Auditing and Finance*, 1982, pp.30-40.
- Hannye, L. G., "Auditors and Dp'ers Benefits from Association in the Systems Development Process," *The Internal Auditor*, Volume 6, December 1977, pp.67-70.
- Holly, Charles L., and Reynolds, K., "Aiditor Concerns in an On-line Distributed Computer Network," *Journal of Systems Management*, (June 1984), pp.32-36.
- Institute of Internal Auditors, *System Auditability and Control Audit Practices*, Part 4, Institute of Internal Auditors, Inc., Altamonte Springs, Florida, 1977, pp.109-164.
- Mair, W. C., Davis, K. W., and Wood, D. R., Computer Audit and Control, Q.E.D. *Information Science*, Institute of internal Auditors, Inc., Altamonte Springs, Florida, 1977, p.20.
- Vasarhelyi, M. A. and T. W. Lin, *Advanced Auditing : Fundamentals of EDP and Statistical Audit Technology*, Addison-Wesley Co., 1988, pp.540-554.
- Wilkinson, Joseph W. *Accounting and Information System*, John and Wiley · Son, Inc., 1991, pp.770-799.
- Wilkinson, Joseph W. *Accounting and Information System*, John and Wiley · Son, Inc., 1991, pp.770-799.