
碩士學位論文

소프트웨어開發費의 會計處理基準에 관한 研究

- 韓國·美國·日本을 中心으로 -

指導教授 李 相 奉



濟州大學校 經營大學院

會計學科 會計學專攻

梁 文 禎

1995

소프트웨어開發費의 會計處理基準에 관한 研究

- 韓國·美國·日本을 中心으로 -

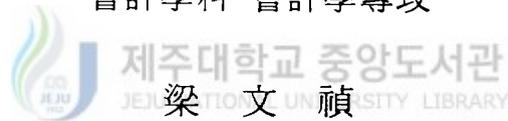
指導教授 李 相 奉

이 論文을 經營學 碩士學位 論文으로 提出함.

1996年 6月 日

濟州大學校 經營大學院

會計學科 會計學專攻



梁文禎

梁文禎의 經營學 碩士學位 論文을 認准함.

1996年 6月 日

委員長_____

委員_____

委員_____

목 차

I. 서론	1
1. 문제의 제기	1
2. 연구목적과 범위	2
II. 소프트웨어개발비의 개념 및 분류	4
1. 소프트웨어개발비의 정의	4
2. 소프트웨어개발비의 분류 및 체계	12
III. 소프트웨어개발비의 회계처리	19
1. 소프트웨어개발비의 자본화	22
2. 소프트웨어개발비의 비용화	33
3. 이용목적별 소프트웨어개발비의 회계처리	39
IV. 주요국의 소프트웨어개발비의 회계처리기준 비교	46
1. 한국의 회계처리기준	46
2. 미국의 회계처리기준	54
3. 일본의 회계처리기준	63
V. 요약 및 결론	71

表 目 次

표 2-1	소프트웨어매출원가명세서의 일례	10
표 2-2	원가계산대상의 변천	11
표 2-3	정보처리산업과 제조업과의 유사점	12
표 2-4	소프트웨어비의 취득방법별 분류	13
표 3-1	비용화 기업과 자본화 기업의 수	20
표 3-2	업종·영업형태·소프트웨어 유형별 비용화 기업과 자본화 기업	21
표 3-3	소프트웨어 개발목적별 기업의 수	21
표 3-4	소프트웨어자산성 판정의 기준	29
표 3-5	미래의 경제적편익과 소프트웨어의 자산성	30
표 3-6	소프트웨어개발 방법 및 이용목적별 회계처리	45
표 4-1	정보통신부 기준의 구성 및 체계	50
표 4-2	소프트웨어 개발공정단계별 업무범위	51
표 4-3	한·미·일 소프트웨어개발비의 회계처리기준 비교	68

圖 目 次

그림 2-1	소프트웨어의 범위상 구분	6
그림 2-2	소프트웨어개발비의 개발단계별 분류	14
그림 3-1	소프트웨어제작공정과 제품원가성	28
그림 4-1	세법에 의한 소프트웨어회계처리법	66

I. 서 론

1. 문제의 제기

최근 정보화 사회가 진전되면서 모든 기업뿐만 아니라 일반 사람에게도 컴퓨터 이용이 보편화되었으며, 이에 따라 하드웨어 뿐만 아니라 소프트웨어에 대한 중요성도 점차 크게 인식되고 있다. 우리나라의 소프트웨어산업을 볼때, 95년도말 매출은 2조 5천 5백억원 정도로 국민총생산(GNP)에서 차지하는 비율은 1.99%로 추산되며, 97년도에는 3조 6천 3백억원¹⁾으로 GNP의 2.84%로 예측되고 있지만 향후 대폭적인 증가가 예상된다. 또한 인터넷 관련 소프트웨어시장이 매년 100%씩 성장하여 오는 2000년까지 세계적으로 40억달러 규모²⁾가 될 것으로 예상하고 있다.

이러한 사회적 환경에도 불구하고 생산성 측면에서 하드웨어의 생산성은 소자의 집적화 기술의 급속한 발전에 의해 대폭 향상되고 있는데 비해, 소프트웨어의 생산성향상은 매우 낮은 속도로 성장하고 있는 상태라고 말할 수 있다. 소프트웨어 개발을 주도 하고 있는 소프트웨어산업의 경우 무형(intangible)의 재화를 생산한다는 점에서 산업분류상 제조업으로 분류되지 못하고 있는 실정임을 감안한다면, 소프트웨어 개발비의 적정산정을 통한 생산성의 향상은 현재 기업이 해결하지 않으면 안될 중요한 과제의 하나이다.

또한, 증가하는 소프트웨어의 개발비에 대한 회계처리 결과에 따라 당해 기업의 기간손익, 재무상태 및 기업의 경영합리화에 직결되기 때문에 소프트웨어 개발회사에서 개발비에 대한 회계처리 문제는 대단히 중요한 사항으로

1) 전자신문, 1996. 2. 8자

2) 上揭書, 1996. 2. 2자

인식되어 진다. 과거에는 원가계산의 대상이 제품을 중심으로 이루어져 왔으나, 현재 사회에 있어서는 정보나 무형의 지적생산물도 원가계산의 주요한 대상으로 받아들여져야 한다는 의견이 확산되고 있음에도, 아직도 우리나라에서는 이에 대한 회계처리기준이 정립되지 못하고 있으며, 단지 연구개발비에 준한 회계처리를 하고 있는 실정이다.

따라서, 소프트웨어개발비의 회계처리에 대한 회계기준이 정립되지 않은 우리나라에서 소프트웨어개발비의 회계처리에 관한 관련규정을 살펴보고, 이에 대한 선진국[미국, 일본 등]의 회계기준을 검토하여 우리 실정에 맞는 통일된 회계기준을 마련하는 노력이 필요하다고 하겠다.

2. 연구목적과 범위

기업에 있어서 영업형태가 복잡해짐에 따른 체계적인 이익관리, 제조원가 명세서 등 재무제표의 작성, 적절한 조세납부 등의 필요에 의해서, 그리고 자사용 소프트웨어를 개발 또는 구입하여 사용하는 기업에 있어서도 재무제표의 적정성을 위한 자산가치의 평가 등 이해관계자의 합리적인 의사결정에 유용한 정보제공을 위해서도 소프트웨어개발비에 대한 회계처리가 적절하게 이루어져야 할 것이다.

이러한 소프트웨어개발비의 회계 문제는 개발비를 당기 비용화 할 것인가, 또는 대차대조표에서 자산으로 계상할 것인가로 요약되어 진다. 그 이유는 첫째, 소프트웨어개발비는 일반적으로 거액이라는 점이다. 개발비가 소액인 경우에는 그 비용을 자산화하거나 비용화하더라도 기간손익에 큰 영향을 미치지 않으나, 거액인 경우에는 수익·비용의 대응이나 자산평가에 커다란 영

향을 미친다. 둘째, 소프트웨어개발비에 대한 회계관행이 확립되지 못하여 기업마다 각각 다른 회계처리를 하고 있다. 셋째, 소프트웨어의 가치는 이용방법과 이용자에 따라 크게 다르기 때문에 객관적인 평가가 어렵다는 소프트웨어 자체에 내재하는 성질 때문이다.

이러한 문제점들로 인하여 소프트웨어개발비에 대한 통일적이고 적절한 회계처리 방법이 모색되지 않는다면 투자자의 정보인식에 혼란이 우려되고, 또한 기업의 자산을 왜곡처리할 가능성의 내포 및 이익계상에 혼란을 줄 수 있다. 그리고 급속도로 변화하는 정보사회의 환경에서 회계정보의 유용성을 저하시킬 수 있다.

따라서, 본 논문의 목적은 소프트웨어개발비의 회계처리에 대하여 이론적으로 고찰하고, 한국·미국·일본을 중심으로 소프트웨어개발비에 대한 회계처리기준을 검토하는 것이다. 그리고 현재 고유의 소프트웨어개발비의 회계처리기준이 없는 우리나라의 통일적인 회계처리기준의 제정을 위한 방향설정에 일익을 담당코자 하는데 있다.

그리고 본 논문의 범위는 문헌에 의한 연구로서 소프트웨어회계가 비교적 체계화되어 있는 미국과 일본의 소프트웨어에 관한 회계처리기준과 이 분야에 대한 여러학자 및 유관 단체의 논문, 저서, 연구보고서 및 기타 관련자료 등을 비교·검토하는 방법으로 소프트웨어의 경제적 성질과 그에 따른 회계상의 제반 문제를 분석·검토하고자 한다.

본 논문의 구성은 제1장 서론에 이어 제2장에서는 소프트웨어개발비에 대한 개념과 특질, 분류 및 체계를 고찰하고, 제3장에서는 소프트웨어개발비의 자본화 및 비용화에 대한 논거를 살펴보고, 제4장에서는 한국·미국·일본을 중심으로 각국의 회계처리기준을 비교·검토하고, 마지막으로 제5장에서는 본 논문을 요약하고 결론을 맺는다.

II. 소프트웨어개발비의 개념 및 분류

1. 소프트웨어개발비의 정의

1) 소프트웨어의 정의와 특질

(1) 소프트웨어의 정의

소프트웨어는 문헌상 다양한 의미로 쓰이고 있으나 일반적으로 소프트웨어란 용어는 원래 컴퓨터 그 자체의 물리적 장치를 의미하는 하드웨어의 상대적인 의미로서 하드웨어를 기능시키기 위한 수단을 칭하고 있다. 그러나 최근에는 네트워크 시스템이나 데이터베이스 등으로 넓게 쓰이고 더구나 무형의 지적 창조물의 총칭으로서 컴퓨터 이외의 분야에도 사용되고 있다.³⁾

현대에 있어서는 인간의 지식이나 아이디어에 의해서 창작된 형체가 없는 재화를 총칭하고, 소프트웨어라고 부르는 일이 많아 지고 있다. 그러나 본 논문에서 소프트웨어는 약간 한정적으로 컴퓨터소프트웨어의 의미로 사용되었다.

컴퓨터 소프트웨어(computer software : 이하, 소프트웨어라고 약칭)는 컴퓨터를 움직이는 것만이 아니라 이용기술을 말한다. 아무리 훌륭한 컴퓨터가 있어도 소프트웨어가 없으면 단지 상자에 지나지 않는다고 말할 수 있다. 문헌상 소프트웨어는 여러가지 의미로 사용되고 있지만, 비교적 많이 사용되고 있는 것은 <그림2-1>과 같은 소프트웨어의 범위에 따라 각기 다양하다.

3) 위준복·김영수; "소프트웨어 회계처리에 관한 연구". 「경영논총」 제15집, 전남대학교 경영대학원, 1990, p.1.

① 프로그램

소프트웨어를 갖고 기계 또는 인간이 읽게 되는 프로그램의 작업이라고 한다. 그것은 컴퓨터소프트웨어 본래의 의미이고, 예전에는 컴퓨터소프트웨어는 이 의미로 사용되는 일이 많고, 현재에서도 이와 같은 해석을 하는 연구자도 많다.⁴⁾ 여기에서는 주로 컴퓨터프로그램과 같은 의미로 사용하고 있다.

② 프로그램, 흐름도, 사양서, 조작마이크로 등 관련자료

소프트웨어는 프로그램 뿐만 아니라, 흐름도, 사양서, 조작마이크로 등의 관련자료까지도 포함하는 것으로 한다. 미국표준국(National Bureau of Standards)은 소프트웨어를 데이터처리 시스템의 조작으로 관련한 컴퓨터프로그램, 절차, 규칙 및 관련문서로서 정의를 부여하고 있다. 동일한 관점에서 일본공업규격(Japanese Industrial Standard)도 소프트웨어를 이용하여 데이터처리시스템의 운용에 관계있는 계산기프로그램, 절차, 규칙 및 그것에 관련 있는 문서로서 규정하고 있다.⁵⁾

한편 회계학의 관점에서 미국 내국세입청(IRS)은 제2의 해석에 의해, 소프트웨어를 프로그램뿐만 아니라 문서류를 포함한 것으로 정의하고 있다.

③ 프로그램, 관련서류, 지원서비스

조언이나 기술원조 등의 지원서비스까지도 포함하는 광의의 해석으로서 구체적으로는 FM(Facility Management : 시설관리)의 작업도 또한 소프트웨어라고 한다. 소프트웨어를 이용하기에는 프로그램이나 user 마이크로 등의 관련서류뿐만 아니라, user 지원을 위한 각종 도구, FM의 체제, 및 교육, 훈련 등 다수의 요소가 필요하다. 그러나 회계학상으로 자산성이나 원가관리의 대상이 된 소프트웨어에는 일정한 틀이 필요하게 된다.

이상과 같은 소프트웨어의 개념은 그 범위 구분에 따라 대상의 폭도 상이

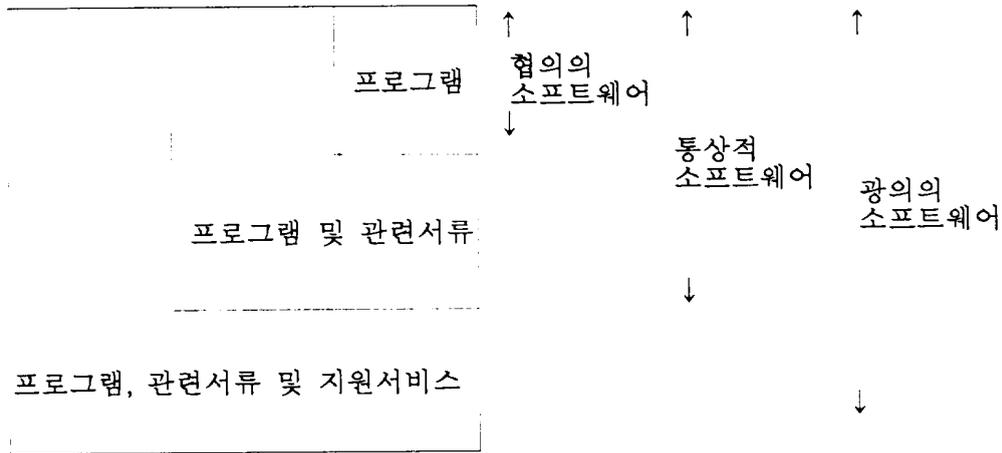
4) 櫻井通晴: 「ソフトウェア原價計算」, 白桃書房, 1987, p.18.

5) 櫻井通晴: 上掲書, p.19.

하지만, 본 논문에서는 광의의 개념을 사용하기로 한다.

그리고 소프트웨어의 유사용어로서는, 영화필름, 레코드, 노하우(know-how) 등도 있다.⁶⁾

<그림2-1> 소프트웨어의 범위상 구분



(2) 소프트웨어의 특징

소프트웨어의 특징을 경제적 측면에서 살펴보면 소프트웨어는 컴퓨터본체인 하드웨어에 대한 컴퓨터을 움직이는 이용기술로 인식되고 있다. 따라서 하드웨어가 고유의 제품인데에 대해 소프트웨어의 실용적 내용은 무형의 정보이다. 무형의 정보로서 소프트웨어가 갖고 있는 경제적 특징은 다음의 3가지를 들 수 있다.⁷⁾

① 무형의 이용기술

소프트웨어는 무형의 이용기술이라고 일반적으로 이해되고 있다. 그러나 무형이라는 점에 있어서 유형이다라고 하는 견해도 있다. 예를 들면, 콜로라도주 워싱턴 D.C 등 많은 주에서는 소프트웨어를 갖고 무형이라고 한다. 매

6) 櫻井通晴: 上掲書, pp.24-28.

7) 櫻井通晴: 上掲書, p.22.

출세(미국에서는 소매단계에서 과세된다. 일반적인 소비세의 형태이다)가 걸리지는 않지만 캘리포니아주나 아이호아주에서는 수주소프트웨어는 무형으로 면세되고 있어도 범용소프트웨어는 유형이라 하고 매출세가 과세되고 있다. 확실히 범용팩키지 소프트웨어를 갖고 소프트웨어제품이라고 하는 용어가 있는 그대로 유형이라고 하는 견해가 경제실태에 합치하는 일도 많다. 그러나 일반상식부터 말하면 적어도 현재에 있어서 무형이라 해석되어야 할 것이다.

이용기술이라는 점에 있어서도 다른의견이 있다. 예를 들면, 고교수준은 '계산기의 하드웨어와 동일한 형태같이 소프트웨어도 또 하나의 기계라고 보는 경우도 있다'라고 서술하고 있다. 회계상의 용어에서 이것을 표현하면 용역잠재력(service potential)이라는 면에서 하면 소프트웨어도 또 이 견해를 전면 부정하지는 않지만, 소프트웨어는 기계 그것과 구별된 이용기술이라는 면을 갖고 있는 일을 부정할 수 만은 없다라고 생각한다. 무형의 기술이라는 점에서 노하우(know-how)와 유사하다. 그러나 노하우는 소프트웨어와 다른 기술비밀이다.

②저작권

소프트웨어는 소설, 회화, 음악등의 저작물(현재, 저작권법의 대상이 되고 있다)과 유사하다. 저작권은 공업소유권과 함께 무형재산권에 속하고 저작물을 독점적으로 지배하고 이익을 받는 권리를 갖는 배타적인 권리이다. 미국에서는 저작권법 중에 명문 규정을 설치하고 소프트웨어에 보호를 부여하고 있다. 여기에서 법률상 소프트웨어에도 저작권을 인정해야 한다고 하는 이론이 성행하고 있다.

소프트웨어의 보호를 현재, 저작권의 일부개정으로 대응(1986년 1월 1일 시행)하고 있다. 소프트웨어를 저작권으로서 인정한다고 하면 회계상은 무형 고정자산으로 될 것이다. 소프트웨어가 경제적활동, 기업활동에 이용된다는 점에서 예술이나 학술작품 등 문화활동에 관련하는 저작물과는 그 성격을 다

르게 한다. 그리고 또 하나는 미국에서 저작권은 75년 (현행회계원칙을 전제로한 40년), 독일 70년, 일본에서 일반적인 저작권법에 의하면 저자의 사후 50년이면 길기 때문에 비용화가 실상 곤란하게 되는 어려움이 있다. 이로 부터 소프트웨어를 저작권으로서 처리하기 위해서는 회계학상 내외동시에 반대가 많다. 현실적으로는 경제활동에 이용하는 소프트웨어도 저작권법에 대해서 규정되고 회계상의 상각기간도 5년으로 단기적인 상각이 되고 있지만, 위에 논한 것처럼 소프트웨어는 기존의 개념 구조 속에서 해석하기 보다는 경제활동의 수행에 필요한 정보작성에 관련될 필요성이 있다.

③경제재

소프트웨어가 경제활동에 관련하는 이용기술이라고 하면, 다음으로는 소프트웨어가 경제재로서 인식여부에 대해 검토되어야 한다. 일반적으로 경제재란 경제활동수행을 위한 희소 수단이다. 특정기업에 있어서 경제재란 이용 또는 판매를 통해서 장래기업에 효익을 제공할 것으로 기대되는 희소수단이다. 이러한 의미에 있어서 소프트웨어는 경제재라고 말할 수 있다. 일본통산성 정보산업부회의 「소프트웨어 기반 정비를 하는 방법에 대해서」 라는 보고서에 의하면 소프트웨어는 컴퓨터의 이용기술로서 산업활동에는 빠뜨릴 수 없는 경제재라고 규정하고 있다.

2) 소프트웨어개발비의 정의

소프트웨어개발비는 소프트웨어의 개발에 소요된 모든 비용을 총칭하며, 이 비용은 후술하는 분류체계에 따라서 그 내용과 범위가 매우 상이하다.

소프트웨어개발비는 소프트웨어개발 목적에 따라 그 비용산정의 기준이나 범위 등이 각각 상이하지만, 개발단계에 따라서는 자체개발이나 위탁개발에 포함되는 모든 비용을 대상으로 하고 있다.

3) 소프트웨어개발비의 특징

소프트웨어는 일종의 지적인 공업제품이라고 간주되고 있다. 따라서 기본적으로는 하드웨어와 같은 원가계산제도를 소프트웨어에 적용할 수가 있다. 이와 같이 원가계산제도를 이용하여 소프트웨어의 개발 및 운용 등에 소요된 모든 비용을 총칭하여 소프트웨어개발비(software costs)라고 정의할 수 있다. 이러한 소프트웨어개발비를 소프트웨어의 대상이나 범위에 따라 상이할 수 있지만 주로 다음과 같은 3가지의 특징을 갖고 있다.⁸⁾

첫째는 외상매출품의 원가가 불투명하다는 점이다. 이로 인해 소프트웨어 개발비의 관리가 용이하지 않다는 점과 또한 원가관리를 위한 소프트웨어 개발에 소요된 시간 등을 정확하게 측정하는 것이 중요하다는 점이 발생한다.

둘째는 원가구조가 하드웨어와는 매우 상이하다는 점이다. 예를 들어, 일본의 소프트웨어하우스 422사의 평균 노무비를 보면 1985년의 경우 제조원가의 58%로서 동 년도의 일부 상장 제조업(건설업 제외)의 평균 11%에 비해 5배나 높게 나타나고 있다.

<표2-1>의 소프트웨어 제조원가보고서는 소프트웨어개발에서의 인건비(노무비 및 외주비)가 전형적인 제조업에서의 원가구조에 비하여 매우 높음을 제시해 주고 있다.

셋째는 소프트웨어개발에서 투입(in-put)과 산출(out put)관계가 불명확하다는 점이다. 소프트웨어개발이 실패할 가능성은 점차 대폭적으로 감소되었다고는 하지만 아직도 그 가능성이 완전히 없어진 것은 아니다. 더우기 소프트웨어엔지니어링의 기술차가 아주 크다. 따라서 일정한 노동을 투입했다고 해서 반드시 일정한 소프트웨어 산출을 보증하게 되는 것은 아니기 때문이

8) 한국세계연구회; "일본의 소프트웨어 관리실체", 「월간 경영과 세무」 VOL.8, No.4, 1989,p.144.

다.

<표2-1> 소프트웨어매출원가명세서의 일례

(단위 : 천엔)

과 목	제 28 기 19X1. 4. 1 19X2. 3.31		제 29 기 19X2. 4. 1 19X3. 3.31		비교증감 (△)
	금	액 구성비	금	액 구성비	
I.재료비	1,508,079	4.4	1,424,775	3.5	△ 83,303
II.노무비	8,564,177	24.5	9,902,973	25.3	1,388,795
III.외주비	17,857,214	51.6	20,276,599	51.7	2,419,384
경 비	6,667,705	19.3	7,609,107	19.4	941,402
당기경제조비용	34,597,177	100.0	39,213,455	100.0	4,616,278
기초외상품인 재고	2,096,344		2,375,700		
합 계	36,093,521		42,089,246		
타계정에의 대체금액	2,015,607		2,385,300		
기말외상품인 재고	2,875,790		3,108,726		
당기소프트웨어소원가	31,802,123		36,595,219		4,793,096

(출처) 한국세제연구회; “일본의 소프트웨어 원가관리실체”, 『월간 경영과 세무』 VOL.8, No.4, 1989. p.144.

4) 원가계산 대상으로서의 소프트웨어

정보화사회의 진전은 기업경영에 충격을 가져다 줌과 동시에 원가계산에도 현저한 영향을 미치고 있다. 예를 들면 원가계산의 대상은 종래 제품을 중심으로 하는 물적 재화가 주체로 인식되어 왔다. 그러나 현재 사회에 있어서는 이것에 더불어 정보나 무형의 지적생산물도 주요한 원가계산의 대상으로서 인식되고 있다.

櫻井通晴 및 Kaplan에 의하면 <표2-2>과 같이 원가계산의 대상이 종래

의 제품 중심에서 점진적으로 정보로 이동하는 경향을 보이고 있다. 이 배경에는 정보기술 등의 발달에 의해 정보의 중요성이 높아가고 있음을 나타내고 있는 것이며, 이에 따라 정보 원가계산의 필요성도 지적되고 있다.⁹⁾

<표2-2> 원가계산대상의 변천

원가계산연구의 중심과제	시 대 구 분	원가계산대상
(1)원가발견형 원가계산	15,6세기부터 1885년경	
(2)제품원가산정지향형 원가계산	1885년 ~ 1920년경	산출(제품)
(3)경영관리지향형 원가계산		
①경영통제지향형 원가계산	1920년 ~ 1945년경	산출, 활동
②경영의사결정지향형 원가계산	1945년 ~ 1980년경	산출, 활동 프로젝트, 활동
(4)정보지향형 원가계산	1980년 ~	산출, 활동 프로젝트, 정보

(출처) 櫻井通晴: 前掲書, p.36.의 <표 3-2>

그러나 제품과는 달리 추상적인 무형의 정보의 자산계상 및 비용처리 또는 원가계산은 적어도 현재로서는 사실상 곤란한 경우가 대부분이다.

소프트웨어의 개발에서는 사람의 솜씨와 기계설비를 사용해서 제품을 완성시킨다. 그 점에 있어서 소프트웨어의 개발은 제조업에 있어서의 제조과정과 본질적으로 다르지 않다. 소프트웨어개발의 반 이상을 실시하고 있는 소프트웨어산업은 산업분류상은 서비스업이지만 동시에 제조업적인 요소를 강하게 갖고 있다.

소프트웨어산업이 제조업과의 대비에 있어서 <표2-3>과 같은 유사한 점

9) 櫻井通晴: 前掲書, pp.33-35.

을 나타내고 있다.¹⁰⁾

상기에서 소프트웨어 개발업자(developer)가 원가계산을 경쟁력강화를 위한 수단으로써 인식하여 원가계산을 유효하게 활용하는 것에 대한 중요성을 시사하고 있다.

<표2-3> 정보처리산업과 제조업과의 유사점

소프트웨어산업	제 조 업
펀치처리센터 펀치·센터	데이터가공업
소프트웨어하우스	시스템제조업(건설업)
F M 업	시스템제조, 운전청부업
TSS서비스	시스템제조산업

(출처)櫻井通晴: 上掲書, p.39.의 <표 3-1>

2. 소프트웨어개발비의 분류 및 체계

1) 소프트웨어개발비의 분류



소프트웨어개발비는 취득방법별, 개발단계별, 개발목적별 등으로 구분할 수 있는데, 이중 취득방법별 및 개발단계별 분류는 다음과 같다.

(1) 취득방법별 분류

취득방법별 분류는 매입·위탁연구개발, 매입·위탁제조와 자체연구개발, 자체제조로 구분할 수 있다. 그리고, 조합으로는 ①매입·위탁연구개발-자체제조, ②자체연구개발-자체제조, ③자체연구개발-매입·위탁제조, ④매입·위

10) 櫻井通晴: 上掲書, p.39.

탁연구개발-매입·위탁제조를 들 수 있다.11)

매입·위탁연구개발, 매입·위탁제조에 관한 회계처리에 대해서는 정도의 차이는 있지만, 계약일 경우 성공률이 높기 때문에 비용으로 처리해야 하는가 또는 자산으로 계상하여야 하는가, 만약 자산으로 계상하는 경우에는 어느 정도의 기간에 걸쳐서 상각할 것인가에 대해서는 사용목적 및 사용상황에 따라 상당히 객관적인 판단을 세울 수 있을 것이다.

매입 및 자체연구개발, 자체제조에 관한 소프트웨어개발비의 분류를 회계 처리방법과 연관시켜서 나타내면 다음의 <표2-4>와 같다.

<표2-4> 소프트웨어개발비의 취득방법별 분류

소프트웨어 개발활동	매입 (위탁 개발)	장래 대체적 이용가능	자 산 계 상	
		장래 대체적 이용불가능	연 구 개 발 비	
	자체 개발	연구개발비		
소프트웨어 개발의의 활동	매입 (위탁 개발)	장래 가능한 경제적 효익	자 산 계 상	
		장래 불가능한 경제적 효익	판매비, 일반관리비 또는 제조경비	
	자체 개발	실행가능성, 개념설계 및 보수	판매비, 일반관리비 또는 제조경비	
		상세설계, 코딩 및 테스트	장래 가능한 경제적 효익	자 산 계 상
		장래 불가능한 경제적 효익	판매비, 일반관리비 또는 제조경비	

(2) 개발단계별 분류

개발단계별 분류는 소프트웨어의 개발프로세스를 중심으로 한 분류로서

11) 김준식·이호열: 前掲書, p.8.

<그림2-2>와 같이 연구단계 . . . 개발단계 . . . 유지·보수단계의 6가지의 주요 기능에 의한 분류이다. 각 단계별로 계획 및 실행가능성의 검토과정에는 각각 조사비 및 분석비로, 설계과정은 개념설계와 상세설계로 구분되어 설계비(개념설계비와 상세설계비로 구분됨)로 분류된다. 그리고 코딩과정에서는 코딩비로, 테스트과정은 테스트비로 각각 분류되며, 유지·보수과정의 유지·보수비는 소프트웨어개발비의 범위에 따라 귀속여부가 결정된다.

한편, 우리나라의 경우 정보통신부에서 고시한 「소프트웨어개발비 산정 기준」에 의하면 <그림2-2>와 같이 상기 절차와는 다수 상이한 내용을 규정하고 있으나 공정구분에 의한 소프트웨어개발비의 구분은 매우 유사하다.

<그림2-2> 소프트웨어개발비의 개발단계별 분류



또한, 기본소프트웨어와 응용소프트웨어의 개발 및 제조단계를 비교하면, 연구개발단계와 제조단계의 비중이 상당한 차이가 나는 것이 일반적이다.

기본소프트웨어는 통상 하드웨어와 더불어 연구개발을 하기 때문에 거액의 연구개발투자액이 소요되는 경향이 있다. 한편, 응용소프트웨어는 연구개발이라 하더라도 개발연구(연구개발단계를 기초연구-응용연구-개발연구로 구

분하는 경우)가 주축이 되며, 또한 제조단계가 그 중심이 된다는 점에 유의할 필요가 있다.

자체연구개발의 계획단계에서 소요된 소프트웨어비용은 사전에 성공을 확신하는 것이 불가능하기 때문에 성과가 장래에 보장되지 않는다면 소위 연구개발비라는 독립항목으로 하여 당기비용으로 처리될 수 있다. 자체제조, 즉 설계에서 보수까지의 단계, 전술한 매입·위탁연구개발 및 매입·위탁제조에 소요된 소프트웨어비용은 개발 목적을 고려하여 적절한 비용의 구분과 회계처리방법 등을 모색하여야 한다.

2) 소프트웨어개발비의 체계

소프트웨어개발비는 제조원가계산과 같은 원가계산제도의 이론적 틀에 의거하여 계산될 경우 다음과 같이 원가계산준칙에 의해 재료비, 노무비 및 경비로 구분된다.¹²⁾

(1) 개발 소프트웨어의 재료비

원가계산준칙에 의하면 원가요소는 제조원가와 판매비 및 일반관리비로 구성되어 있는데, 그 중 제조원가요소는 형태분류에 따라 재료비, 노무비 및 경비로 대별된다. 소프트웨어 산업은 현재까지도 전형적인 노동집약적 산업이기 때문에 원가중에 노무비의 비중이 매우 높다. 따라서 공업제품에 있어서 상당한 비중을 차지하고 있는 재료비 즉, 기억매체 또는 용지의 소비원가를 재료비로 간주하는 경우에도 소프트웨어 개발비에서 점하고 있는 비율은 금액적으로 매우 사소한 것이다.

재료비는 물품의 소비에 의해 발생한 원가로서 다음과 같이 분류되는데,

12) 원가계산준칙, 1994, 제2장.

이 비목들이 개발 소프트웨어의 재료비로서 적절한 것인가에 대해 검토하기로 한다.

①주요재료비

원가계산준칙에 있어서 재료비의 개념은 제품의 소재가 되는 물품의 소비에 의해 발생하는 원가라고 할 수 있다. 여기서 소프트웨어는 일반적인 공업제품과는 달리 가시적으로 확인할 수 없는 급부인데, 문제는 눈으로 확인할 수 없는 급부를 만들어 내기 위하여 소비된 물품의 측정이 매우 어렵다는 것이다.

②부분품비

소프트웨어개발을 위해 부품을 구입한 것이 아니라, 외부로부터 구입한 소프트웨어를 가공하여 새로운 소프트웨어를 개발하는 경우, 구입한 소프트웨어는 마치 개발 소프트웨어의 부분품에 해당한다고 할 수 있는데, 이러한 의미에서 구입한 소프트웨어는 개발 소프트웨어의 재료비로 간주된다. 그러나 원가계산기준에 있어서 재료비는 물품의 소비를 전제로 하기 때문에, 구입한 소프트웨어는 직접경비로 계산되는 것은 가능하나 재료비로서 계산되는 것은 곤란하다고 할 수 있다.

③공장소모품비

공업제품의 생산에서 소비되는 것과 같은 소모품으로서 편치카아드, 코딩 용지, 인쇄용지, 자기 테이프, 플로피 디스크, 서적, 잡지 등이 있다. 이러한 소모품은 주로 개발부문에서 소비되는데, 이러한 소모품의 소비에 의해 발생하는 원가를 개발부문 소모품비라고 할 수 있다.

④소모공구비품비

원가계산준칙에 있어서 소모공구비품비는 내용연수가 1년 미만 또는 취득 원가 10만원 이하이기 때문에 고정자산으로서 처리되지 않는 공구비품의 소비에 의해 발생하는 원가이다. 소프트웨어의 개발에 있어서 개발부문 등의

전자계산기를 위시하여 각종의 기기를 사용함으로써 수선 또는 보수를 위한 비용이 발생한다. 개발 소프트웨어의 재료비에는 개발부문소모품비, 소모공구비품비 등이 있다.

(2) 개발 소프트웨어의 노무비

노무비는 노동용역의 소비에 따라 발생하는 소프트웨어의 개발에 있어서는 공업제품의 생산의 경우와 거의 차이가 없다. 다만 개발 소프트웨어의 원가계산에 있어서는 다른 원가계산의 경우에 비해서 노무비의 점유율이 높다고 할 수 있으므로, 노무비의 계산에 있어서는 실제로 소프트웨어의 개발에 종사하는 인원의 인건비에 추가해서, 어느정도의 인건비를 제조원가에 산입할 것인가 하는 것이 중요한 요건이 된다.

일반적으로 개발부문의 인건비는 모두 인건비로 인식할 수 있다. 그 중 프로젝트에 배속되어 소프트웨어개발에 직접 종사하고 있는 인원의 인건비는 직접노무비로 인식할 수 있다. 한편 시스템 개발부문의 사무원 또는 시스템 개발책임자 등의 인건비는 간접노무비로 하는 것이 타당하다. 따라서 시스템 개발부문을 공업제품생산공장의 경우로 보아 프로젝트 단위를 각 제조부문으로 가정하면, 노무비의 인식범위 또는 직접비와 간접비의 분류방법도 명확하게 할 수 있을 것이다.

(3) 개발 소프트웨어의 경비

경비는 재료비, 노무비 이외의 제조원가요소를 말하며 기업의 실정에 따라 적당한 계정과목으로 분류한다. 또한 재료비 및 노무비에 속하는 항목 중 이것을 재료비 및 노무비로 분류하는 것이 곤란한 경우에는 경비에 속하게 할 수 있다.

개발 소프트웨어의 원가계산에 있어서는 비목 또는 내용에 있어서 특징이 있다. 즉, 외주가공비의 내용에는 프로그램 위탁비, 카드 천공위탁비, 계산위

탁비, 데이터 작성위탁비 등이 있으며, 임차료에 있어서는 부동산 임차료 이외에 각종 OA기기의 리스료도 포함되어 있다. 한편 새로운 비목을 설정할 필요도 있는데, 이미 언급한 구입 소프트웨어비, 전자계산기이용료, 통신회선 이용료, 프로그램 임차료, 테스트용 데이터비 등이 있으며, 이미 전자계산기 또는 각종 단말기를 구입한 경우에는 유지보수비가 발생할 것이다.



Ⅲ. 소프트웨어개발비의 회계처리

기업에 있어서 영업형태가 복잡해짐에 따른 체계적인 이익관리, 제조원가 명세서 등 재무제표의 작성, 적절한 조세납부 등의 필요에 의해서, 그리고 자 사용 소프트웨어를 개발 또는 구입하여 사용하는 기업에 있어서도 재무제표의 적정성을 위한 자산가치의 평가 등 이해관계자의 합리적인 의사결정에 유용한 정보제공을 위해서도 소프트웨어개발비에 대한 회계처리가 적절하게 이루어져야 할 것이다.

이러한 소프트웨어개발비의 회계 문제는 개발비를 당기 비용화 할 것인가, 또는 대차대조표에서 자산으로 계상할 것인가로 요약되어 진다. 그것은 첫째, 소프트웨어개발비는 일반적으로 거액이다. 개발비가 소액인 경우에는 그 비용을 자산화하거나 비용화하더라도 기간손익에 큰 영향을 미치지 않으나, 거액인 경우에는 수익·비용의 대응이나 자산평가에 커다란 영향을 미친다. 둘째, 소프트웨어개발비에 대한 회계관행이 확립되지 못하여 기업마다 각각 다른 회계처리를 하고 있다. 셋째, 소프트웨어의 가치는 이용방법과 이용자에 따라 크게 다르기 때문에 객관적인 평가가 어렵다는 소프트웨어 자체에 내재하는 성질 때문이다.

이러한 문제점들로 인하여 소프트웨어개발비에 대한 통일적이고 적절한 회계처리 방법이 모색되지 않는다면 투자자의 정보인식에 혼란이 우려되고, 또한 기업의 자산을 왜곡처리할 가능성의 내포 및 이익계상에 혼란을 줄 수 있다. 그리고 급속도로 변화하는 정보사회의 환경에서 회계정보의 유용성을 저하시킬 수 있다.

위와 같은 실정에서 우리나라 소프트웨어를 개발 또는 구입하는 기업에서

도 소프트웨어개발비에 대한 회계처리를 자본화하는 기업, 비용화하는 기업과 같이 다양하게 이루어지고 있는 실정이다. 손성호의 연구(1991)¹³⁾에 의하면 조사대상 기업 108개 업체중 비용화하는 기업이 90(83.33%)사, 자본화하는 기업이 18(16.66%)사로 대부분 기업들이 비용화를 선호하고 있으며, 업종별로 보면 서비스업체 75사중 63(84.00%)사가 비용화하고 있으며 12(16.00)사가 자본화하고 있다. 그리고 제조업체 33사중 27(81.82%)사가 비용화, 6(18.18%)사가 자본화하는 것으로 나타나고 있다. 영업형태별로 보면 전문업체 22사중 14(63.64%)사가 비용화하고 있으며, 8(36.36%)사가 자본화하고 있다. 겸영업체 86사중 76(88.37%)사가 비용화, 10(11.63%)사가 자본화하고 있다. 또한 소프트웨어유형별로 보면 응용업체 35사중 30(85.71%)사가 비용화, 5(14.29%)사가 자본화하고 있으며, 시스템업체 73사중 60(82.19%)사가 비용화, 13(17.81%)사가 자본화하고 있는 것으로 나타나고 있다. 위 사항을 요약하면 <표3-1>과 같다.

<표3-1> 비용화 기업과 자본화 기업의 수

구 분	업 체	비용화 기업	자본화 기업	합 계
조사대상기업(%)	전 체	90(83.33)	18(16.67)	108(100.00)
업 종(%)	서비스업체	63(84.00)	12(16.00)	75(100.00)
	제조업체	27(81.82)	6(18.18)	33(100.00)
영 업 형태(%)	전문업체	14(63.64)	8(36.36)	22(100.00)
	겸영업체	76(88.37)	10(11.63)	86(100.00)
소프트웨어유형(%)	응용업체	30(85.71)	5(14.29)	35(100.00)
	시스템업체	60(82.19)	13(17.81)	73(100.00)

또한 업종·영업형태·소프트웨어 유형별 비용화 기업과 자본화 기업의

13) 손성호 : 前掲書, pp.104-105

실태를 보면 <표3-2>와 같다.

<표3-2> 업종·영업형태·소프트웨어 유형별 비용화 기업과 자본화 기업

업종	영업형태	소프트웨어	비용화 기업	자본화 기업	합계
서비스업	전문업체	응용	1	1	2
		시스템	10	4	14
	겸영업체	응용	17	3	20
		시스템	35	4	39
제조업	전문업체	응용	0	0	0
		시스템	3	3	6
	겸영업체	응용	12	1	13
		시스템	12	2	14
합	계	90	18	108	

그리고 소프트웨어 개발목적별 기업의 수를 살펴보면, 조사대상기업 108사중 용역제공목적이 76(70.37%)사, 시장판매목적이 32(29.63%)사를 나타내고 있으며 업종 및 영업형태 그리고 소프트웨어유형에 따라 소프트웨어 개발 목적을 살펴보면 <표3-3>과 같다.

<표3-3> 소프트웨어 개발목적별 기업의 수

구분	업체	용역제공목적	시장판매목적	합계
조사대상기업(%)	전체	76(70.37)	32(29.63)	108(100.00)
업종(%)	서비스업체	57(76.00)	18(24.00)	75(100.00)
	제조업체	19(57.58)	14(42.42)	33(100.00)
영업형태(%)	전문업체	13(59.09)	9(40.91)	22(100.00)
	겸영업체	63(73.26)	23(26.74)	86(100.00)
소프트웨어유형(%)	응용업체	25(71.43)	10(28.57)	35(100.00)
	시스템업체	51(69.86)	22(30.14)	73(100.00)

이하에서 소프트웨어개발비의 자산화와 비용화에 대한 이론적 주장을 검토한다.

1. 소프트웨어개발비의 자본화

1) 소프트웨어의 자산성 인식

(1) 자산의 본질

FASB의 SFAC No.3에 의하면, 자산(asset)이란 과거의 거래나 사건의 결과로서 특정실체에 의해 획득되어 지배되며 예상할 수 있는 미래의 경제적 효익으로 정의되고 있다. SFAC No.3에 나타난 자산의 세가지 기본적인 특성은, ①예상되는 미래의 경제적 효익, ②실체에 의한 지배, ③과거 거래나 사건의 결과이다.¹⁴⁾

경제적 자원으로서 자산은 용역잠재력이나 기대되는 효익에 대한 권리를 나타낸다고 볼 수 있는데 이러한 입장은 포괄적인 자산의 정의를 낳게 하고 측정의 문제를 별도로 다룰 수 있게 하였다. 이와같은 기본적 정의와 함께 자산에는 다음과 같은 본질이 존재한다.

- ①미래효익이나 용역잠재력에는 어떤 특정의 권리가 존재 하여야 한다.
- ②권리는 특정의 개인 또는 기업에 대해서 발생한 것이어야 한다.
- ③자산이 되기 위해서는 권리 용역에 대해서 법적으로 강제 할 수 있는 청구권이 있어야 하며, 미래 효익의 수입이 예상되는 어떤 다른 증거가 있어야 한다.
- ④경제적효익은 과거의 거래나 사건의 효과로서 나타나야 한다. 따라서

14) 위준복·김영수: 前掲書, p.4.

자산의 분류가 각 항목들이 지닌 자산으로서의 특성을 변화 시키지는 않는다.

이러한 근본적인 본질 속에서 소프트웨어의 자산성의 본질을 파악해 본다면, 회계상 거래에 기인할 지출등을 자산으로서 인식 할 경우의 요건은 해당 지출이 장래의 수익획득 능력을 가지고 동시에 합리적인 금액측정이 가능해야 할 것이다. 이러한 두가지의 소프트웨어의 자산성에 대해서 살펴보면 3가지로 나누어서 살펴볼 수 있다.

첫째, 소프트웨어의 무형성에서 장래의 수익획득 능력을 중심으로 살펴보면 소프트웨어는 통상적으로 플로피디스크나 ROM 등에 그 기능이 기억되어 있으며 그 자체는 볼 수 없다. 또한 소프트웨어는 하드웨어와 일체가 되어 사용되므로 가치를 가진 것이 있고, 그 효용은 하드웨어와 일체화하여 표시된다. 소프트웨어는 이와 같이 무형이지만 그 효용으로서 장래의 수익획득 능력은 구체적으로 다음과 같이 존재한다.

- ① 생산수단으로서 이용하는 것에 의한 생산효율의 향상에 따른 제품원가의 절약효과
- ② 사무관리 수단으로서 이용하는 것에 따른 관리비용의 절약효과
- ③ 판매용으로서 제작하는 것에 의해 장래매출에 따라 얻을 수 있는 수익획득 능력 등이 있다.

기업이 소프트웨어를 구입 또는 제작할 경우에는 이러한 미래의 수익획득 능력에 대해서는 경영목적상 충분한 조사가 되고 그 능력여부가 확인되어서 취득의 의사결정이 된 상태이다.

둘째, 합리적인 금액측정의 가능성으로서 소프트웨어를 외부에서 구입할 경우에는 그 구입가액에 따라 합리적인 금액이 측정될 수 있다. 또한 내부제작의 경우에도 적정한 원가계산을 하는 것에 의해 금액측정이 가능하다.

이상과 같이 소프트웨어는 장래 수익획득 능력과 합리적인 금액측정 가능

성 2가지 요건을 만족시키는 것이 되며 이것을 자산으로 계산해야 한다는 것이다.

셋째, 소프트웨어의 유동성으로 현대의 기술혁신시대에서 소프트웨어는 그 가치를 장기적으로 유지시키기가 곤란하다. 보통 소프트웨어를 호전시키고 보다 효과적인 가능성을 향상시키고 우수한 것으로 유지보수 또는 교체를 하여야 한다. 그러나 이것이 소프트웨어의 자산성을 부정할 만한 근거는 되지 못한다. 즉 적절한 시기에 그 자산성을 조사하고 평가액이나 경제적 사용 가능 기간을 산정하는 것에 의해 적정한 자산계상은 가능하게 된다.

(2) 자산계상의 대상

개발소프트웨어의 원가계산을 실시함에 있어서 언제 발생한 비용을 제조원가로 할 것인가 하는 원가의 인식 시점이 중요한 문제가 된다. 다시 말해서, 개발 소프트웨어의 수주에서 부터 발주자에게 납품하는 과정 중 어느 시점까지의 비용이 영업활동의 비용이며, 또는 어느 시점이후의 비용이 제조원가인가 하는 문제가 제기된다. 소프트웨어 개발과정은 요구분석, 기본설계, 상세설계, 코딩, 테스트, 도입 및 유지보수등의 주기로 구성되어 있다.

소프트웨어를 개발할 때 개발과정을 모두 자사에게 실시하는 기업이 있는가 하면, 요구분석과 기본설계 등은 자사에서 실행하고 다음 과정은 모두 타사에 위탁하는 경우도 있다. 또한 전과정을 위탁하여 실증을 마친 신뢰성있는 소프트웨어를 구입하므로써 소요시간의 최소화 등 절차상 편리, 자체개발에 따른 위험의 회피, 가격의 저렴 등의 장점을 얻고자 하는 기업도 있다.

원가계산준칙을 기초로 한 소프트웨어 특유의 원가계산제도의 필요성이 명확하게 되어야 하고 다음에 소프트웨어비계산의 계산대상을 명확히 하지 않으면 안된다. 그것은 소프트웨어제작의 과정과의 관계에서 이것을 검토할 필요가 있다. 소프트웨어부문에서 행해지고 있는 업무 중에서 어떤 부분을

소프트웨어제작의 범위에 귀속시킬 것인가에 대한 議論은 다양하다. 예를 들면, 운용단계를 소프트웨어개발에서 제외하는 경우도 있고, 운용과 보수간에 구분하기 어려운 면이 존재하는 경우도 인정해야만 한다. 그리고 여기에서는 소프트웨어비계산 대상, ①소프트웨어의 개발단계, ②보수단계의 원가계산으로 구분하도록 한다. 이들 2가지 비용계산 중에서 보다 중요성이 있는 것은 소프트웨어의 개발단계의 비용계산이다. 그리고 다음으로 소프트웨어개발 비용계산에서 검토해 보자.

소프트웨어개발의 과정은 <그림3-1>에서 나타내는 것과 같이 문의부터 수주확정에 이르는 일련의 영업활동과, 시스템조사·분석 이후, 운용에 이르는 생산활동으로부터 이루어 진다. 이론적으로는 영업활동의 과정에서 발생하는 비용을 기간비용으로 취급하는 것에 대해 하드웨어의 생산과정의 제조비용에 상당하는 것이 생산활동의 과정에서 발생하는 비용이다. 다시 말하면, 소프트웨어제작에 있어서 제작품원가(제품원가)는 이론상 <그림3-1>에 있어서 생산활동의 과정으로 발생하는 비용이다.¹⁵⁾

판매가능성을 명확하게 하기 위한 시장조사 비용은 일반적으로 소프트웨어의 제품화 여부의 의사결정을 위해 지출되고 특정의 소프트웨어생산에 직접적으로 필요한 비용이라고는 말할 수 없는 경우가 많다. 따라서, 시장조사를 위한 비용은 제품원가성을 갖고 있지 않으므로 판매비·일반관리비로서 처리해야 할 것이다.

시스템조사·분석을 위한 비용은 제안서 제출전에 발생하는 경우도 있다. 이와 같이 제안서를 위한 매매거래가 있는 특정의 소프트웨어의 발표(presentation) 때문에 요구하는 시스템조사·분석의 비용에 대한 회계처리 방법은 다음과 같다.

제안서 제출까지는 많은 작업과 비용을 요구하기 때문에 특정의 프로젝트

15) 櫻井通晴: 前掲書, pp.53-54.

가 인식될 때부터 프로젝트별로 시스템조사·분석을 위해서 필요한 비용의 집계를 시작으로 성공하면 그것에 제품원가성을 부여하는(실패시는 기간비용 취급한다) 경우도 있다. 이와 같은 회계처리가 이론적으로 허용될 수 있는가에 대해 櫻井교수는 이것도 허용되는 것이 좋다고 지적하고 있다. 그 이유로서는 하청으로 수주한 소프트웨어개발 과정은 하청에 의한 건설공사에 유사한 성질을 크게 갖고 있지만, 장기도급공사의 원가계산에 있어서는 판매비 및 일반관리비도 또 제품원가에 포함해서 계산할 수 있기 때문이다.¹⁶⁾

생산활동의 범위를 시스템조사·분석에서 시작할 것인가 혹은 설계과정(기본설계 또는 상세설계)만을 인식할 것인가 또는 코딩이후로 보아야 할 것인가는 의견이 양분되고 있다. 실무상의 관행을 보면, 미국에서는 코딩과 테스트, 또는 상세설계부터 테스트까지를 갖고 생산활동으로 이해하는 견해가 유력하다. 한편 일본에서는 시스템조사·분석부터 테스트(또는 운용), 또는 기본설계부터 테스트까지를 제품원가의 범위라고 하는 기업이 많다.¹⁷⁾

소프트웨어개발비를 자산계상하기로 한 경우 자산계상의 대상으로 어느 공정원가를 제품원가로서 인식할 것인가의 문제(<그림3-1>참조)에 관해 다음의 4가지의 견해가 있다.¹⁸⁾

- ①코딩과 테스트 공정비용을 제품비용으로 한다.
- ②코딩, 테스트와 상세설계공정의 비용을 제품비용으로 한다.
- ③코딩, 테스트, 상세설계와 기본설계공정의 비용을 제품비용으로 한다.
- ④코딩, 테스트, 설계와 수주후의 시스템조사·분석비용을 제품비용으로 한다.

이 중 ①은 FASB¹⁹⁾에 따른 것이다. 그러나 건전한 원가계산 이론에서 본

16) 일본 「기업회계원칙」 제2의 3F, 「채무제표규칙」 제84조.

17) 櫻井暎晴: 前掲書, p.55.

18) 한국세계연구회: 前掲書, p.85.

19) FASB, SFAS No.86, "Accounting for the Costs of Computer Software to Be Sold,

다면 설계비를 제품원가에서 제외해서는 안된다. 또한 실무적으로도 설계비를 제품원가에 포함하는 공정한 회계관행에서 판단해 보더라도 이러한 견해는 바람직하지는 않다.

②는 McGee²⁰⁾ 등 미국의 연구자 간에서 지배적인 견해이다. 다만 설계공정의 원가중에서는 상세설계의 원가까지 자산계상의 범위를 한정할 것인가에 대해 가치창조적인 공정이 기본설계공정의 원가로 제품원가에서 제외하는 것은 타당성이 없다고 지적되고 있다.

실태조사의 결과에²¹⁾ 따르면, ③과 같이 적어도 기본설계까지의 소프트웨어개발비는 일본에서 90%이상의 기업이 자산계상의 범위로 하고 있다.

그리고 ④의 계획공정인 시스템조사·분석공정의 원가에 제품원가성을 부여할 것인가의 문제는 시스템공정의 비용을 수주전과 수주후로 나누어 수주후의 비용은 제조원가, 수주전의 비용은 원칙적으로 판매비로 처리해야만 한다. 수주후의 비용을 제조원가로 하는 이유는 수주후에 엔지니어가 소프트웨어의 시스템조사·분석에 있어서 이 시점에서 제조지침서가 개설되는 것이 일반적이기 때문이다.

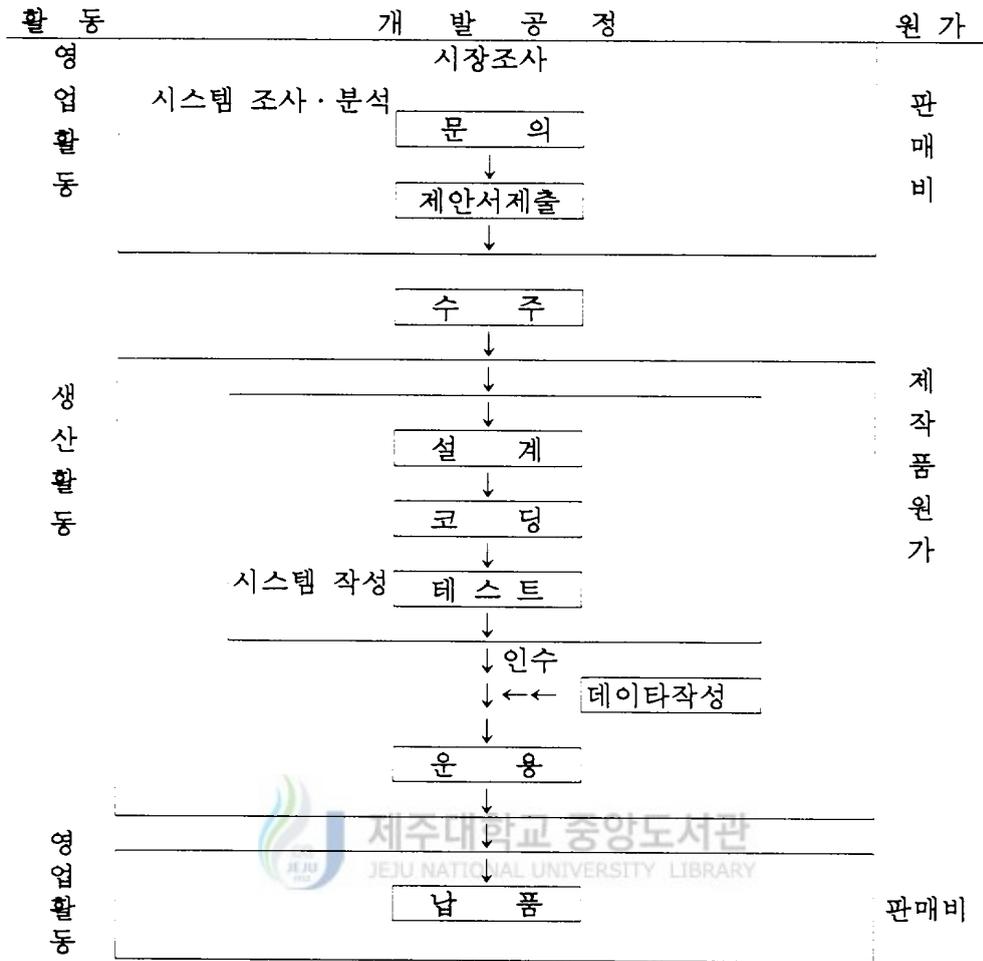
이러한 경우에는 수주가 확정되기까지 비망계정으로 기록해두어 수주가 확정되었을 때 제조비용을 대체할 수가 있게 되는 것이다. 이상 자산계상의 대상으로 할 공정비용으로는 시스템조사·분석에 의한 비용중 제품원가가 있다고 인정되는 수주후의 비용(패키지·소프트웨어의 경우에는 해당전공정비용) 및 기본설계, 상세설계, 코팅, 테스트에 필요한 모든 비용이 포함된다. 또한 시장조사, 객선도입의 비용은 판매비로서 처리하는 것이 타당하다.

Leased or Otherwise Marketed," : FASB, 1985.

20) McGee, Robert; "Accounting for software," Dow Jones-Irwin, 1985.

21) 소프트웨어 정보센터(SOFTIC)에서 1990년에 실시한 실태조사 결과로서, 한국세계연구회; "일본의 소프트웨어 회계실무지침의 기본적인 개요", 「월간 경영과 세무」 VOL.5, No.48, 1992, p.82.

<그림3-1> 소프트웨어 제작공정과 제품원가성



(3) 자산성 인식기준의 연혁

자산성에 대한 인식기준을 보면, 해당소프트웨어가 미래의 경제적 편익 (future economic benefits)이 있는가의 여부에 따라 판정되어야만 한다. 미래의 경제적 편익이란 기업이 장래의 기업활동에 대한 강제적인 편익을 의미한다.

기업은 궁극적으로 이익의 획득을 목적으로 하는 조직체이므로 경제적 편익이란 이익의 원천을 이루는 수익획득 능력 또는 기업에게 있어서의 잠재적인 용역을 말하는 것이다. 여기에서 잠재적인 용역이라고 하는 것은 일반적으로 용역잠재력(service potential)이라고 불리우며, 그것은 기업에 있어서의 유용성이라고도 바꾸어 말할 수 있다.

또한 자산이기 위해서는 측정가능성(measurability)이 없으면 안된다고 하는 것은 말할 것도 없는 일이다. 그렇다면 소프트웨어의 자산성은 무엇에 의해 판정되어질 것인가를 <표3-4>에 제시한 바와 같은 몇가지의 견해를 토대로 검토할 수 있다.²²⁾

<표3-4> 소프트웨어자산성판정의 기준

발 표 년 차	소프트웨어의 자산성판정의 기준
FASB(1974)	측정가능성 (Measurability)
FASB(1984)	회수가능성 (Recoverability)
FASB(1985)	기술적 실행가능성(Technological Feasibility)
	기술적 실행가능성 (Technological Feasibility)
櫻井/폴/벨(1987&1990)	판매가능성 (Marketability)
	이용가능성 (Usefulness)

(출처)한국세제연구회: 前掲書, p.84.

소프트웨어의 자산성은 미래경제적 편익의 여하에 따라 판정된다고 되어 있으나 보다 구체적으로는 기술적 실행가능성 및 판매가능성 또는 이용가능성에 따라 판정된다.

여기에서 기술적 실행가능성 (Technological Feasibility)이 있는 소프트웨

22) 한국세제연구회 : 前掲書, p.84.

어에서는 해당소프트웨어의 연구개발이 끝나고 제작이 가능한 상태에 있는 소프트웨어를 말한다. 기술적 실행가능성은 시작품이 완성된 시점 또는 제품 제작이 완성된 시점에서 확립된다. 한편 판매가능성(Marketability)이란 판매용소프트웨어에 대해서도 판매되며 수익획득이 예상되는 상태에 있음을 말한다. 또한 이용가능성(Usefulness)이란 소프트웨어가 사내에서 항상 이용가능하며 용역의 잠재성이 확실히 예상되는 상태에 있음을 말한다.

소프트웨어를 판매용 소프트웨어와 사내이용 소프트웨어로 구분할 경우, <표3-5>에 제시한 바와 같이, 판매용 소프트웨어개발비는 기술적 실행가능성과 판매가능성에 의해 자산성이 판정된다. 그리고 사내이용 소프트웨어개발비는 기술적 실행가능성과 이용가능성에 의해 판정된다.

<표3-5> 미래의 경제적 편익과 소프트웨어의 자산성

자산성 인식요건	자산성 인식기준	소프트웨어개발비의 구분
미래의 경제적 편익	기술적 실행가능성	판매 소프트웨어개발비
	판매가능성	
	기술적 실행가능성	사내이용 소프트웨어개발비
	이용가능성	

기술적 실행가능성은 시작품의 완성 및 제품제작의 예상이 가능하기까지의 공정완료 등 소프트웨어개발의 확립이라고 하는 사실에 따라 비교적 객관적으로 확인된다. 어려운 것은 판매가능성 및 이용가능성의 구체적인 판정기준이다. 판매가능성은 테스트에 대해 그의 구체적인 조건으로서 일본의 소프트웨어의 자산성에 관한 검토위원회(이하, 위원회라 약칭)는 개발계획서프로젝트지침서, 시장조사의 자료, 판매계획서 시가 및 원가에 관한 자료담합·수

주·판매에 관한 자료 및 중빙의 하나 또는 그 이상의 자료를 갖춘 것이 필요하다고 했다. 다행히도 현실정에서는 수주소프트웨어의 개발이 약 90%를 점하는 일본에서는 수주서의 확인에 따라 수많은 수주소프트웨어의 판매가능성을 확인하는 것이 비교적 용이하다.²³⁾

한편 범용패키지 소프트웨어의 판매가능성의 판정에 관해서는 다음의 자료를 혼합하여 범용소프트웨어의 판매가능성을 확인하는 수 밖에 없다. 현재로서는 객관성있는 판매가능성의 테스트를 요구하는 것은 곤란하다. 그러나 위원회는 가까운 장래에 소프트웨어의 판매에도 저서와의 제작·판매에 유사한 회계처리가 가능하게 되는 것이 아닌가 하는 인식을 강화하고 있다. 즉 미리 판매수량을 예측하여 초판(최소의 소프트웨어의 제작)에서 제판비(개발비)를 회수하여 재판(복제품)이후에는 제본비(복제원가)를 회수하게 된다.

다음으로 소프트웨어를 사내이용할 때의 이용가능성의 테스트에 대해 소프트웨어의 사내이용에 따라 기업에 직접적으로 수익이 유입하게 되는 것이 아니라 불확실성도 높으므로 용역잠재력의 객관적인 판정은 반드시 용이한 것은 아니다. 따라서 판매용 소프트웨어개발비와는 다르게 이용가능성의 변화가 자산성의 결정적인 요인이 되는 사내이용 소프트웨어개발비의 자산성은 일반적으로 낮은 것으로 된다.

2) 소프트웨어개발비의 자산성 논쟁

경재제라고 하는 것은 회계학상 소프트웨어개발비를 자산계상하지 않으면 안되는 것을 의미한다. 이 문제에 대한 찬성과 반대 양론은 다음과 같은 논거에 의한다.²⁴⁾

23) 한국세제연구회: 上揭書, p.84.

24) 櫻井通晴: 前掲書 pp.28-32.

(1) 자산계상에 대한 반대론

우선 소프트웨어개발비의 자산계상 반대론의 논점은 다음과 같다.

① 연구개발비와의 유사성

소프트웨어개발비는 일회성의 혁신적인 프로젝트인 연구개발비와 비슷하다. 개발에 성공하면 경제재가 될 수 있지만, 실패하면 기업에 이익을 주지 않을 뿐만 아니라 불량자산이 발생하기 쉽기 때문에 당기의 비용으로 처리하는 것이 건전한 회계처리법이다.

② 자산가치 측정이 곤란성

소프트웨어개발비는 원자재의 투입량과 소프트웨어 제품의 산출량간의 상관관계가 공업 생산물만큼 명확하지 않기 때문에, 소프트웨어의 원가계산에 있어서는 완성품의 가치나 개발과정에 대한 객관적인 측정이 곤란하다. 복제품의 원가는 원형(master)에 비해서 매우 싸지만 기술적 성능과 제품으로서의 그 가치 성질은 원형과 완전히 같기 때문에, 원가에 따라 그 평가액을 결정하면 회계상의 평가액이 소프트웨어가치에 적합하지 않게 되 버린다.

③ 원가계산기준의 미비

소프트웨어산업은 전형적인 노동집약산업으로서 원가의 대부분은 인건비이므로 하드웨어 원가계산에서 일반적으로 인정되는 재료비, 노무비, 경비 등의 원가분류는 소프트웨어 원가계산에는 적용될 수 없는 등 현실적으로 많은 제한이 있다.

(2) 자산계상에 대한 찬성론

다음으로 소프트웨어개발비의 자산계상 찬성론의 논점은 아래와 같다.

① 소프트웨어 개발기법의 향상

소프트웨어의 개발과정은 연구개발 보다는 건설공사에 유사하며, 기존의

모듈화된 코딩방법을 이용하여 제작되는 경우 실패의 위험성도 적기 때문에 자산계상해도 기업의 건전성을 얻을 수 있다.

②수익과 비용의 대응

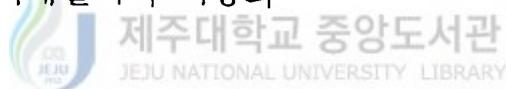
현재 소프트웨어개발비를 비용화한다면 수익과 비용을 대응시킬 수 없고, 따라서 기업의 보고이익과 미래영업활동 기간에 이연된다면 막 개업한 소프트웨어 회사가 거액의 자본투자를 추가적으로 할 수 없다는 것을 의미한다.

또한, 매출수익이 장래 리스 등을 통해서 계속적으로 얻어지는 경우 수익·비용을 적절하게 대응시키기 때문에, 자산에 계상하고 얻어진 수익의 액수에 대응해서 비용화 해야 할 것이다.

③구입비용

구입한 소프트웨어는 경제재이다. 따라서 그 비용은 유상취득의 대가를 무형고정자산에 계상하는 경영권에 준해서 자산에 계상해야 할 것이다.

2. 소프트웨어개발비의 비용화



소프트웨어개발비의 비용화를 지지하는 사람들은 '소프트웨어의 내용년수를 예측하기가 어렵고, 어떠한 경우든 단기이기 때문에 현행 산업실무와 일치시키기 위해 비용화해야 한다'라고 주장하고 있다. 또 다른 주장자들은 '해당 원가가 별로 중요하지 않거나 측정하기가 어렵고 비용화하는 것이 편리하므로 비용화해야 한다'는 것이다.

이러한 비용화의 주장을 살펴보면 다음과 같다.²⁵⁾

25) 손성호; "컴퓨터 소프트웨어 개발원가의 회계선택에 관한 실증적 연구", 박사학위논문, 경북대학교 대학원, 1991, p.14.

1) 재무보고목적

가장 기본적으로 이익정보를 제공하는데 초점을 맞추고 있다. 이익정보는 투자자와 기타 이용자(잠재적인 투자자와 채권자)가 투자의사결정을 내리는데 유용하여야 한다. 투자자와 채권자가 여러가지 실체에 대하여 이와 유사한 의사결정을 내리는데 도와 줄 수 있는 수단으로서 재무보고기준은 동종 산업내 실체간의 비교가능성을 제공해 주어야 한다.

소프트웨어개발비를 선택적으로 자본화하게 되면 동종 산업내의 모든 실체를 객관적으로 상호 비교할 수 있는 기준을 제공할 수 없다. 일관성 없는 기준에 의해 적용되는 선택적 자본화(selective capitalization)는 분기별 순이익의 비교를 왜곡시킬 것이다. 왜냐하면 선택적 자본화는 객관성과 비교가능성이 없기 때문이다. 따라서 보고성과(reported results)에 대한 투자자의 신뢰도는 감소할 것이다.

미국의 관리회계위원회는 소프트웨어산업 전문가의 대부분이 내부적인 소프트웨어 개발비가 자본화되어서는 안되는 것으로 믿고 있다고 지적하였다. 그 이유로서, 첫째 소프트웨어 개발원가는 연구개발비와 유사하므로 비용화되어야 한다는 점, 둘째 변화의 속도도 빠르고 산업내의 경쟁도 극심하므로 소프트웨어 개발원가를 자본화하는 것이 의문스럽다는 점, 셋째 소프트웨어 개발원가의 자본화에는 주관이 개입되기 때문에 남용의 가능성이 많다는 점을 제시하고 있다.

2) 비용과 수익의 대응

소프트웨어개발비의 자본화를 주장하는 사람들은 제품의 기술적 실행가능성이 확보된 이후에 발생한 소프트웨어개발비만을 자본화하고 비용과 수익을

적절히 대응시키기 위하여 상각해야 한다고 주장하고 있다. 만약 제품의 기술적 실행가능성이 확보된다면, 여러가지 고도의 판단기준을 조사하고 다음 사항을 계속적으로 검토해야 한다.

- ① 개선된 제품을 계속적으로 시판하려는 경영자의 의도와 능력
- ② 제품의 미래 효용이나 시장성
- ③ 원가에 포함시킬 요소의 식별

급속한 기술 변화로 특징지을 수 있는 극도의 경쟁 산업에서 계속적으로 이 기준을 적용하거나, 기간별 이익의 왜곡을 방지하는 것이 어려울 것이다. 뿐만 아니라 자본화된 소프트웨어개발비가 새롭거나 개선된 제품의 총 개발비를 나타낼 수 없으며 따라서 자본화 주장자들이 강조하는 ‘대응개념’이 소프트웨어개발비의 자본화 기준으로서 완벽하게 적용될 수도 없을 것이다.

3) 현행기준과 산업실무간의 불일치

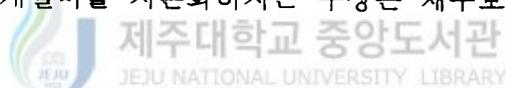
영화산업의 경우 영화라는 소프트웨어의 개발과 배급노력은 소프트웨어 개발에 기울인 노력과는 전혀 다르다. 또한 소프트웨어는 개발 이후에도 새로운 기술변화를 수용하여 계속적으로 보완되어야하고 다른 제품과는 통합되어야 한다. 영화산업은 제작노력과 그 결과로 나타난 제품 사이에 직접적인 관련성이 강하다고 할 수 있지만, 소프트웨어산업은 지속적인 수정 노력을 기울여야 하는 독특한 측면이 있으므로 기술적 실행가능성의 면에서 많은 차이가 있는 것이 사실이다. 다시 말하면, 영화필름을 제작하는 데 든 노력은 실패할 가능성이 적고, 이들 개발비를 계량화하는데 엄격한 판단기준을 마련할 필요가 없으므로 자본화가 적절하게 허용되고 있다.²⁶⁾

26) FASB, SFAS No.53, "Financial Reporting by Producers and Distributors of Motion Picture Films," FASB, 1981.

소프트웨어 구입은 전혀 주관이 개입되지 않고도 그 금액을 측정하거나 계량화하는 문제를 해결할 수 있고, 그 가치는 재무보고의 기본요소의 하나인 원가이므로 구입한 소프트웨어비의 회계처리와 내부적으로 개발된 소프트웨어개발비의 회계처리가 다름으로써 거래의 실질이 반영된다. 더우기 소프트웨어 구입은 제품에 대한 미래활동이나 시장성에 대한 불확실성이 거의 없다는 것을 의미하므로, 대응개념을 적절하게 반영하기 위해 자본화하고 상각시켜야 한다. 이것은 내부적인 자원의 배분과 전문가의 의견을 신중히 고려한 후의 의사결정이라는 의미가 된다.

4) 미기록자산

재무보고에서는 기업실체의 시장 가치나 부를 나타내지는 않는다. 많은 기업에서는 영업권, 탁월한 종업원관계 그리고 고객명부 같은 미기록 무형자산을 보유하고 있다. 이들 자산은 결코 자본화되지 않고 있다. 그래서 미기록자산인 소프트웨어 개발비를 자본화하자는 주장은 재무보고목적에 부합되지 못한다.



FASB가 1974년에 공표한 SFAS No.2 「연구개발비의 회계처리」에서 소프트웨어회계를 처음으로 언급하였을 때, 소프트웨어개발비는 전액 연구개발비, 즉 당기비용으로 처리하도록 하였다.

그 이유로서 FASB는

- ①장래 효익이 불확실하다.
- ②지출과 장래 효익사이에 인과관계가 불명확하다.
- ③경제적 자원으로 인식되었다 하더라도 회계상 측정가능성을 충족시키지 못한다.
- ④수익·비용의 대응이 성립되지 않는다.

⑤자산으로 계상하여도 기업의 잠재적 수익력을 평가하는데 도움이 되지 못한다.

와 같은 다섯 가지 요인을 들고 있다.²⁷⁾ 그 배경으로서는 당시의 소프트웨어 개발기술로 보아 실패할 위험성이 대단히 높았기 때문이었다고 판단된다. 그리고 1975년에 발표한 SFAS No.6 Draft 「SFAS No.2의 컴퓨터 소프트웨어에 대한 적용」²⁸⁾에서도 컴퓨터 소프트웨어개발에 소요된 비용은 연구개발비, 즉 당기비용으로 처리하도록 하였다.

그리고, 재무분석가연합회(Financial Analysts Federation : FAF)의 재무회계방침위원회(Financial Accounting Policy Committee)도

- ①연구개발비와 유사하다.
- ②변화가 심하기 때문에 자산화에는 문제가 있다.
- ③자산화에는 주관이 개입될 우려가 있다.

와 같은 이유로 모든 소프트웨어개발비를 비용화하여야 한다고 주장하고 있다.²⁹⁾

또한, 根岸 및 櫻井通晴도 다음과 같은 이유로 모든 소프트웨어개발비를 비용화하여야 한다고 주장하고 있다.³⁰⁾

- ①연구개발과의 유사성

소프트웨어의 개발은 한번뿐인 혁신적인 프로젝트이어서 그 개발에 성공하면 경제적 자원이 될 수 있지만 실패하면 기업에 효익을 전혀 가져다 주지 않기 때문에 당기비용으로 처리하는 것이 건전한 회계처리방법이다.

- ②불량자산계상의 가능성

27) FASB, SAFS No.2, "Accounting for Research and Development Costs", FASB, 1974.

28) FASB, Interpretation of Financial Accounting Standards No.6, "Applicabilty of FASB Statement No.2 to Computer Software", FASB, 1975.

29) 김준식·이호열: 前揭書, p.15.

30) 김준식·이호열: 上揭書, p.15.

만약 소프트웨어를 자산으로 계상하는 것을 인정한다면 그 중에는 시대에 뒤떨어진 소프트웨어 등이 자산으로 계상될 위험이 있다. 더우기 그러한 소프트웨어의 질에 대해서는 타인이 판단하기에는 대단히 어려운 것이 소프트웨어의 특징이기 때문에 재무제표의 신뢰성을 손상시킬 위험이 크다고 할 수 있다.

③ 자산가치측정의 곤란성

소프트웨어는 영업권과도 유사하지만 영업권 등의 무형고정자산도 외부 매입을 제외하고는 자산으로 계상하지 않는다. 또한 소프트웨어의 특성상 복제품(copy)은 원품(master)과 완전히 같은 가치, 같은 성질을 가지면서도 그 복사본가가 극히 저렴하기 때문에 자산평가액이 비정상적으로 낮게 된다.

그러나 최근에는 소프트웨어의 개발기술이 비약적으로 발전함으로써 소프트웨어의 개발에 실패할 위험성이 급속하게 감소되고 있다. 더구나 제작된 소프트웨어는 공업제품의 성격이 강한 경제적 자원이다. 즉, 무형이라는 이유만으로 자산성을 인정하지 않는 것은 이론적으로 맞지 않는다는 논의가 제기되기 시작하였다. 또 매입소프트웨어를 자산으로 계상하면서 같은 용역잠재력을 갖는 소프트웨어를 자체개발이라는 이유만으로 비용으로 처리한다는 것은 논리의 모순이다. 소프트웨어 중에는 무형이라는 점을 제외하고는 기계나 설비와 본질적으로 아무런 차이가 없는 것에 유념하여야 한다.

이러한 경제적 실태를 반영하여 미국에서는 1980년대에 들어서면서 소프트웨어를 자산으로 계상하는 기업들이 속출하고 있으며 그 중에는 자산성에 의문이 있는 소프트웨어까지 자산으로 계상하는 기업이 나타났다.³¹⁾ 따라서 증권거래위원회(SEC)는 투자자의 보호관점에서 1983년 4월에 통첩 33-6476 (moratorium)을 발표하여 FASB가 자산성을 인정하기까지는 판매·리스 등을 목적으로 하는 자체 개발소프트웨어비를 자산으로 계상하지 못하도록 하

31) FASB Technical Bulletin No.79-2, "Computer Software Costs", FASB, 1979.

였다.(다만, 동일 이전에 소프트웨어개발비용을 자산으로 계상하고 있고, 동시에 그 내용을 회계방침으로 공시하고 있는 기업은 계속해서 자산으로 계상하는 것을 인정하였다)

3. 이용목적별 소프트웨어개발비의 회계처리³²⁾

소프트웨어개발비의 자본화 또는 비용화에 대한 연구는 현재 반드시 충분하다고는 볼 수가 없다. 왜냐하면 소프트웨어개발비의 회계처리에 관한 방향은 소프트웨어의 종류에 따라 자산성이 다르게 되는 것이 분명해지고 있기 때문이다. 소프트웨어는 이용목적 여하에 따라 판매용인가 자사이용 목적인가가 자산성 판정을 위해 중요한 기준이 될 수가 있다.

소프트웨어 자산성 판정을 위한 중요한 기준으로는 소프트웨어의 종류 즉, 판매용과 자사이용으로 구분하여 볼 필요가 있다. 판매용은 기업에 직접적으로 수익을 가져올 뿐만아니라 불확실성도 낮기 때문에 자산성이 높고, 당기의 원가요소는 집계된 뒤 제품, 제품 또는 상품등의 재고자산으로 계상이 되며 판매시 매출원가로서 수익의 인식과 함께 비용화하거나 원가성이 없는 제조간접비는 당기의 기간비용으로 처리한다.

자사이용은 직접적으로 수익을 가져오지 못하나 기업에 대한 용역잠재력의 기간이 비교적 길기 때문에, 임시거액의 지출중 자사개발의 경우는 이연자산으로 위탁개발의 경우는 자산가치의 공정성을 유지할 수 있기 때문에 무형고정자산으로 계상하며 경상적 또는 반복적 지출은 비용화한다.

32) 한국세제연구회: 前掲書, pp.85-89. 및 박승원·박성규; “소프트웨어원가 및 수익인식에 관한 연구”, 『성심논문집』 제25집, 1993. pp.8-10.

이하에서 판매용과 자사이용 소프트웨어의 개발목적에 따른 회계처리에 관한 내용을 상세히 검토한다.

1) 자사이용 소프트웨어개발비의 회계처리

자사이용 소프트웨어는 자사개발, 위탁개발, 구입 및 기타로 구분할 수가 있다. 그러므로 이러한 분류기준에 따라 사내이용소프트웨어의 자산성을 살펴보면 다음과 같다.

(1) 자사개발 소프트웨어비

자사개발 소프트웨어의 개발에는 기술적인 노하우가 확립되어 있지 않은 것도 적지가 않다. 또한 사내에서 소프트웨어의 이용에 따라 얻어지는 이익은 직접적인 메뉴·플로로서의 수익이 아닌 그의 대부분은 자산이 지닌 용역의 잠재적인 유용성 및 편익이다. 다른 것으로 전용이 곤란한 것도 많고 담보가치도 현저하게 낮은 것이 일반적이다. 이와같이 자사개발 소프트웨어는 기술적실행가능성이 희박하다고 말할 수 있다. 또한 원가계산을 계속적으로 실시하고 있는 기업은 위원회의 조사에 의하면 16%로서 낮게 나타나고 있으며 측정가능성에서 문제가 제기되고 있다.

실태조사의 결과³³⁾에서도 90%의 기업은 기간비용(또는 시험연구비)으로 처리하고 있다. 즉 기업 관행에서는 비용처리가 대부분이라고 말할 수가 있다. 일본의 경우 세법에서는 자사개발 소프트웨어개발비의 비용처리를 허용하고 있으나 이것은 당연한 것이라고 할 수가 있다. 미국에서의 기업관행도 일본과 거의 유사한 경향을 보이고 있으며 비용처리를 원칙적으로 하여 현실적으로는 거의 90%의 기업이 비용처리하고 있다.

33) 소프트웨어 정보센터(SOFTIC): 前掲書, p.88.

그러나 비정상적인 거액의 비용이 소요되고 장기간에 걸쳐 이용이 예상되는 항공사의 좌석예약시스템, 은행의 온라인 등 대규모의 네트워크 시스템의 경우에는 기술적 실행가능성과 이용가능성을 검토하여 미래의 경제적효익을 이연자산화 하여야 한다.

(2) 위탁개발 소프트웨어비

위탁개발 소프트웨어는 판매용 소프트웨어에 관해 다룬 것과 같은 이유에서 위험부담이 자회사에게 있는 한 기술적 실행가능성이 높다. 그러나 현실적으로는 위험부담이 모회사에 있는 일이 많으므로 그러한 경우에는 기술적 실행가능성이 낮다. 이용가능성은 결코 낮다고는 말할수 없으나 사내이용 소프트웨어이므로 판매용 소프트웨어에 비교한다면 잠재적인 용역에 불확실성이 없다고는 말할 수는 없다.

이와 같이 위탁개발 소프트웨어의 자산성은 모든 기업에 자산계상을 강조할 수 있을 만큼 높다고는 말할 수는 없다. 일본의 경우 법인세법에서는 파견과 코딩과정만의 위탁개발에 대해서는 비용처리를 허용하고는 있지만 기타 위탁개발 소프트웨어개발비는 이연자산에 계상을 강제하고 있다. 분명히 위탁개발을 법률적인 관점에서 본다면 구입에 유사함으로 이연자산에 계상을 강제하는 것은 타당성이 있다. 그러나 경제적인 관점에서 본다면 기술적 실행가능성의 측면에서 자사개발 소프트웨어와는 완전히 다른 부분이 없다.

따라서 기술적 실행가능성과 이용가능성을 검토하여 미래경제적인 편익이 있는 소프트웨어에 대해서만 자산계상을 인정하는 것이 유용하다고 사료된다.

(3) 구입소프트웨어비

구입소프트웨어는 개발이 종료되어 있어 구입의사가 있는 것이므로 이용

가능성, 즉 용역잠재력도 높다고 간주된다. 판매를 목적으로 구입한 소프트웨어는 상품과 같기 때문에 원칙적으로 재고자산으로 처리한다. 그러나 복사전매권과 개작권을 가지고 장기에 걸쳐 기업에 수익이 원천이 남아있는 소프트웨어는 무형고정자산으로 3년간에 걸쳐 상각한다.

(4) 기타 소프트웨어비

소프트웨어복제에 소요되는 비용은 재고자산으로 계상하고 판매시에 매출원가로 계상한다. 이때 재고자산의 계상방법은 FASB(1985)에 의하며 소요부품을 각각 재고자산으로 계상한다고 하는 견해도 있으나, 금액적으로 경미한 소프트웨어 복제비는 중요성의 원칙에 따라 일괄적으로 재고자산으로 계상하고 매출시 매출원가로 처리하여야 한다.

그리고 리스사업에 있어서 리스임대된 소프트웨어의 구입 제작비 또는 복제비는 하드웨어와 분리하여 무형고정자산에 계상하고 리스기간에 걸쳐 상각한다. 그러나 하드웨어와 분리가 불가능할 경우는 하드웨어에 포함하여 하드웨어의 내용년수에 걸쳐 상각한다. 또한 하드웨어에 포함된 소프트웨어는 원칙적으로 하드웨어의 재고자산금액에 포함한다.

2) 판매용 소프트웨어개발비의 회계처리

판매용 소프트웨어는 자사개발, 수탁개발, 위탁개발, 구입, 기타로 분류할 수가 있으며 이러한 분류기준에 따라서 판매용소프트웨어의 자산성에 대해 살펴본다.

(1) 자사개발 소프트웨어비

자사개발의 소프트웨어에 자산성이 정식으로 인정된 것은 1984년에 발표

된 공개초안 및 1985년에 발표된 FASB, SFAS No.86³⁴⁾ 「판매, 리스, 기타, 시판목적의 컴퓨터·소프트웨어원가의 회계」이다.

동 기준에 의하면 판매용의 자사개발 소프트웨어개발비에 대해 '기술적 실행가능성 수립후에 발생된 제품마스터 제작의 원가는 자산계상해야만 한다.'라고 규정하고 있다.

판매용 자사개발 소프트웨어의 제작비는 공업생산물의 제조원가에 해당하기 때문에 원칙적으로 재고자산으로 계상한 뒤 매출시 매출원가로 비용화한다. 상각기간에 있어서는 판매용 소프트웨어의 급속한 기술혁신에 의한 진부화를 고려한다면, 현행 상각기간인 5년은 제품의 수명주기와 시장상황을 고려하여 임의 조정할 수 있다. 더우기 소프트웨어에 의해서는 장기간에 걸쳐 그 효과를 파악할 수가 있으므로 장기성자산으로 계상하는 것이 타당하다. 그러나 장기성자산에 계상하는 것에도 문제점이 있다.

즉, 소프트웨어개발을 연구개발이라고 규정한다면 회계상의 이연자산으로 처리하여야 하는데 기술적인 개발능력이 진보된 현재로서는 연구개발활동이라고 규정하기는 곤란하며, 소프트웨어를 무형의 공업제품이라고 인정하여 더우기 법률상의 권리로서의 측면을 강조할 때 무형고정자산에 계상하는 것이 타당하다. 그러나 기업실태는 아직까지 그러한 단계까지는 이르고 있지 못한 상황이다. 즉, 기술력 있는 거대기업에서 개발한 소프트웨어는 무형고정자산이라고 생각할 수도 있으나 기술력이 미약한 중소기업에서 개발한 소프트웨어의 모두가 무형고정자산이라고 해석할 수는 없는 것이기 때문이다.

(2) 위탁개발 소프트웨어비

개발을 위탁한 소프트웨어는 건설공사의 도급과 본질적인 차이는 없다. 따라서 위탁개발 소프트웨어비는 일반에게 외상품에 계상하여 매출시에 매출

34) FASB, SFAS No.86: 前掲書, 1985.

원가로 처리하는 것이 타당하다.

(3) 위탁개발 소프트웨어비

소프트웨어를 개발함에 있어서는 개발에 따른 위험부담이 수탁자측에 있는 한, 위탁자측의 기술적 실행가능성에 의문의 여지가 없다. 판매가능성에 있어서는 구입의 경우와 같기 때문에 구입 소프트웨어의 경우와 같이 재고자산으로 처리하는 것이 타당하다. 그러나, 자사내에 자회사를 만든 뒤 소프트웨어의 개발을 위탁하는 경우 개발에 따른 위험부담이 위탁자에게 있게 되며 기술적 실행가능성은 확립되지 않기 때문에 이 경우 회계처리하는 자체개발의 경우와 동일하게 하면된다.

(4) 구입 소프트웨어비

전매를 목적으로 구입한 소프트웨어는 상품과 마찬가지로 원칙적으로 재고자산으로 처리한다. 그러나 복제전매권만이 아니라 개작권을 갖고 장기간에 걸쳐 기업에 수익의 원천이 남는 소프트웨어는 무형고정자산으로 처리하여 3년으로 균등상각하는 것이 타당하다

(5) 기타 소프트웨어비

소프트웨어복제에 소요되는 원가는 재고자산으로 계상하고 판매시에 매출원가로 계상한다. 이때 재고자산의 계상방법은 SFAS No.86에 의해 소요부품을 각각 재고자산으로 계상한다고 하는 견해도 있으나, 금액적으로 경미한 소프트웨어 복제비는 중요성의 원칙에 따라 일괄적으로 재고자산으로 계상하고 매출시 매출원가로 처리하여야 한다.

그리고 소프트웨어의 리스산업에 있어서 리스임대된 소프트웨어의 구입 제작 또는 복제비는 하드웨어와 분리하여 무형고정자산에 계상하고 리스기간

에 걸쳐 상각한다. 그러나 하드웨어와 분리가 불가능할 때는 하드웨어에 포함하여 하드웨어의 내용년수에 걸쳐 상각한다. 또한 하드웨어에 포함된 소프트웨어개발비는 원칙적으로 하드웨어의 내용년수에 걸쳐 상각한다.

이상과 같은 이용목적별 소프트웨어개발비의 회계처리를 요약하면 다음 <표3-6>와 같다.

<표3-6> 소프트웨어 개발방법 및 이용목적별 회계처리

종 류	개발 방법 및 이용목적		계 정 과 목	회계처리
응용 소프트웨어	자 사 개 발	자 사 이 용	이 연 자 산	감가상각
			당 기 비 용	비용처리
	위 탁 개 발 타 사 구 입	자 사 이 용	무형고정자산	감가상각
			자 사 개 발 위 탁 개 발 타 사 구 입	판 매 용 리 스 등
시스템 소프트웨어		자 사 용	유형고정자산	감가상각
던키 베이스 등		판 매 용	재 고 자 산	매출원가

(출처)박성규·박성원: 前掲書, p.8.

IV. 주요국의 소프트웨어개발비의 회계처리기준 비교

1. 한국의 회계처리기준

1) 기업회계기준상 관련규정

소프트웨어개발비의 회계처리는 소프트웨어개발비를 연구개발비로 볼 수 있는가가 이 문제의 회계처리와 관련하여 핵심적인 질문이라고 할 수 있다. 이에 관하여서는 우리나라 「기업회계기준」과 연구개발비에 관한 규칙인 「연구개발비에 관한 회계처리준칙(이하, 연구개발비준칙으로 약칭)」에 의해 검토해 보면 다음과 같다.

우리나라의 경우는 소프트웨어 개발비에 관한 회계기준이 아직까지 확립되어 있지 않은 실정에 있으나, 기업회계기준 제23조 5호 연구개발비 항목에서 “신제품 또는 신기술의 연구 또는 개발활동과 관련하여 비경상적으로 발생한 비용으로서 미래의 경제적 효익을 기대할 수 있는 것을 연구개발비로 한다”라고 규정하고 있다. 예를 들어 소프트웨어의 개발이 경상적인 활동의 일환이며 그것이 가져다 줄 미래의 효익에 관한 불확실성이 높을 때는 이를 당기의 비용으로 처리하여야 할 것이다.

그리고 기업회계기준 제95조의 규정에 의거 연구개발 활동 및 이와 유사한 활동에 관한 회계처리와 재무보고에 관하여 필요한 세부사항을 정하고 있는 연구개발비준칙[1987]에 의하면, “연구개발비는 연구개발지출의 성격을 경상적인 것과 비경상적인 것으로 구분하여 경상적인 것은 발생시에 비용화하고 비경상적인 것은 자본화하여 일정기간 동안 상각해야 한다”는 회계처리규

정을 명시하고 있다.

연구개발비에 대하여 신제품 또는 신기술의 연구 또는 개발활동과 관련하여 비경상적으로 발생한 비용으로서 미래의 경제적 효익을 기대할 수 있는 것으로 규정하고 있어서 제품화를 위한 개발이나 자사목적의 경우를 막론하고 모두 자본화할 수 있게 하였다.

매입·위탁연구개발에서 기술적 지식이 장래 다른 소프트웨어의 연구개발, 제조에 이용될 수 있는 경우에는 연구개발목적이 제조, 복제하여 판매 및 내부이용 모두 매입·위탁연구개발에 소요된 비용은 일단 자본화, 즉 이연자산으로 계상하고 만약 지적 소유권까지 양수한 경우에는 무형고정자산으로 계상하여 지적 소유권의 효용지속기간에 걸쳐 상각되어야 할 것이다.

또한 신제품 또는 신기술과 관련한 계약내용이 제품생산 또는 기술이용에 대한 법률적 권리 또는 경제적 권리가 보장되어 있는 경우에는 이와 관련하여 지출한 비용은 무형고정자산으로 계상할 수 있고, 계상한 무형고정자산은 제품생산연도 또는 기술이용연도부터 계약만료연도까지의 기간에 정액법으로 상각하도록 규정하고 있다.³⁵⁾

판매를 목적으로 하는 것이라면 기술적 지식이 장래 어느 기간까지 다른 소프트웨어의 제조에 도움이 되는가, 또 판매가 가능한가를 예상하여 상각기간이 결정되어야 할 것이고, 각각의 상각비는 소프트웨어의 제조경비(직접경비)로 하여 복제된 판매용소프트웨어의 제조원가에 포함되어야 할 것이다. 즉, 재고자산의 일부를 구성하게 된다. 내부이용을 목적으로 한다면, 기술적 지식이 장래 어느 기간까지 기업 내의 관리시스템의 구축에 도움이 되는가를 예상하여 상각기간이 결정되어야 하고, 각기의 상각비는 대체로 그 소프트웨어가 이용되고 있는 관리시스템, 예를 들면 사무관리시스템에 있다면 일반관리비(간접경비), 그리고 판매관리시스템에 있다면 판매비(간접경비)와 같은

35) 증권관리위원회, “연구개발에 관한 회계처리준칙”(제정:1987.7.10), 제10조.

당기비용으로 처리되어야 하고, 생산공정관리시스템에 있다면 앞과 마찬가지로 제조경비로 처리되어야 할 것이다. 이와 같이 기술적 지식이 여러 부문의 관리시스템에 이용되고 있는 경우에는 해당 상각비는 일정한 기준에 따라 배분되어야 할 것이다.

만약 기술적 지식이 장래 다른 소프트웨어의 제조에 이용될 수 없는 경우의 비용에 대한 회계처리는 연구개발비로 처리되어야 할 것이다. 다만 이러한 연구개발비의 회계처리를 연구개발비와 같게 취급할 것인가에 대해서는 검토할 필요가 있다. 왜냐 하면, 당해 연구개발이 장래 이용될 수 없는 경우라도 현재 특정 소프트웨어의 개발에 이용되는 소프트웨어의 연구개발비는 판매를 목적으로 하는 것이라면 판매가능 기간에, 내부이용을 목적으로 하는 것이라면 이용가능 기간에 걸쳐 상각되어야 할 것이다.

2) 법인세법상 관련규정

「기업회계기준」 및 「연구개발비준칙」에서는 기술적 지식에 초점을 맞추어 자산화 또는 비용화로 분류하는 것에 비해, 법인세법에서는 <국세청 예규통첩 (법인 22601-686, 1985.3.5)>의 규정에서는 전산시스템개발을 위한 비용의 회계처리는 법인의 업무를 자동화하고자 전산처리시스템을

- ① 자체개발하는 경우
- ② 외부용역으로 개발하는 경우
- ③ 기개발품을 구입하는 경우에

당해 비용의 회계처리는 새로운 전산시스템(소프트웨어)을 개발 또는 위탁개발하기 위하여 특별히 지출한 비용은 법인세법에 의하면 시험연구비에 해당한다. 다만, 하드웨어의 작동을 위한 시스템소프트웨어는 하드웨어의 취득가액에 포함시키고 있다. 즉, 기술적 지식이나 개발형태(자체개발, 위탁개발

또는 매입)에 초점을 맞추지 않고, 단순히 특별한 지출에 초점을 맞추어 자산화의 여부를 판단하고 있다.³⁶⁾

1990년 3월에 개정된 「기업회계기준」에서는 여기에 미래의 경제적 효익을 추가하고 시험연구비라는 과목을 연구개발비(이연자산)라는 과목으로 변경하였다.³⁷⁾

그리고 리스전산설비의 프로그램 사용료가 새로운 전산시스템의 개발을 위하여 지출하는 비용에 해당하는 경우에는 법인세법에 따르면 시험연구비로 처리하여야 하며, 비용 중 신기술의 개발 및 도입 등을 위하여 특별히 지출한 비용에 한하여 시험연구비에 해당하도록 규정하고 있다. 또한 시험연구비로 계상한 사업년도부터 5년이내 매 사업년도에 균등액 이상을 상각하도록 규정하고 있다.³⁸⁾

3) 정보통신부의 회계처리기준

정보통신부기준은 소프트웨어개발촉진법 제8조 및 동법시행령 제13조 및 제14조의 규정에 의하여 소프트웨어개발비 산정기준을 개정한 것으로서 1996년 3월에 공표(정보통신부 고시 1996-21호)된 「소프트웨어개발비의 산정기준」(이하, 「정보통신부기준」으로 약칭)를 말한다. 동 기준의 목적을 보면, 정부 등이 소프트웨어 개발용역사업을 소프트웨어 개발업자에게 위탁할 때 예정가격 산정과 소프트웨어 개발업자가 국내에서 수주하는 소프트웨어개발용역에 대하여 적정한 대가를 산정할 수 있는 기준을 정함을 목적으로 하고 있다. 동 기준의 구성 및 체계는 <표4-1>과 같다.

36) 법인세법 시행령 제38조 제1항 제4호, 1995.

37) 증권관리위원회, “기업회계기준(제정:1981.12.23, 최종개정:1990.3.29); 제38조 제5호.

38) 법인세법 시행령 제38조 제1항 제4호, 1995.

<표4-1> 정보통신부 기준의 구성 및 체계

조 항	내 용
제 1 장	총칙
제 1 조	목적
제 2 조	적용범위
제 3 조	용어의 정의
제 4 조	기준의 일반적인 원칙
제 2 장	소프트웨어 개발비 산정
제 5 조	소프트웨어개발공정
제 6 조	개발규모의 산출
제 7 조	일괄계약방식의 직접인건비 산정
제 8 조	인적지원방식의 직접인건비 산정
제 9 조	제경비 및 기술료의 산정
제 10조	직접경비의 범위
제 3 장	유지보수 및 재개발시의 대가 산정
제 11조	유지보수 수행형태 구분
제 12조	용역유지보수 대가 산정
제 13조	재개발 소프트웨어의 개발비 산정
제 14조	개발규모 증감조정 및 정산

(1) 소프트웨어개발비 산정기준의 일반적 원칙

정보통신부기준에 의한 소프트웨어개발비 산정시 준수해야 할 일반적원칙은 다음의 5가지가 있다.³⁹⁾

- ① 소프트웨어개발용역의 수행형태는 일괄계약방식과 인적지원방식으로 구분한다.
- ② 소프트웨어개발비는 직접인건비, 직접경비, 제경비 및 기술료의 합계액으로 산정함을 원칙으로 한다.
- ③ 소프트웨어 개발에 있어 공정별 직접인건비에 공정별 직접경비, 제경비 및 기술료를 합하여 산정한다.
- ④ 스텝수를 계산함에 있어 실행명령 스텝수만을 계산한다.

39) 정보통신부, "소프트웨어개발비 산정기준", 고시 제1996-21호.

⑤이 기준의 일반적인 사항과 소프트웨어개발비의 구성항목에 대한 기준 금액은 엔지니어링기술진흥법 제10조의 규정에 의하여 과학기술처장관이 인가하는 엔지니어링사업 대가의 기준을 적용한다.

(2) 소프트웨어개발비의 산정

①소프트웨어개발공정

소프트웨어의 개발공정은 <표4-2>와 같이, 기초요구분석, 상세요구분석, 설계, 프로그램 작성, 통합시험 및 설치, 하자보수로 구분함을 원칙한다.

<표4-2> 소프트웨어 개발공정단계별 업무범위

순번	공정구분	업무범위
1	기초요구 분석	<ul style="list-style-type: none"> · 시스템화 필요성 분석 · 시스템화 목표, 범위, 기능개요 설명 · 기초시스템정의를 위한 문서작성
2	상세요구 분석	<ul style="list-style-type: none"> · 시스템화 요구분석 · 시스템의 기능, 성능, 품질 결정 · 정보처리형태별 예상분수 산출
3	설계	<ul style="list-style-type: none"> · 시스템 환경 설계 · 시스템 구조 설계 · 시스템 세부 설계
4	프로그램 작성	<ul style="list-style-type: none"> · 프로그램 사양 작성 · 코딩· 컴파일 · 개별 시험
5	통합시험 및 설치	<ul style="list-style-type: none"> · 시스템 시험환경 조성 · 시스템통합시험 · 시스템 설치 초기작업 · 시스템 시험운용 · 사용자 지침서 작성 · 교육· 시스템 검수 및 인계
6	하자보수	<ul style="list-style-type: none"> · 하자보수기간중에 발견된 소프트웨어의 결함 수정

②개발규모의 산출

본수에 의한 스텝수를 산정할 경우 개발할 소프트웨어의 본수를 산출한 후 별표2(생략)의 정보처리형태별 평균 스텝수를 곱하여 계산하고 본수를 산출하지 않고 직접 스텝수를 계산하는 것이 적정한 경우에는 직접 스텝수를 계산한다.

③직접인건비 산정방식은 일괄계약방식과 인적지원방식에 의하여 수행되며 전자의 방식은 스텝수가 산출되면 공정별 스텝당 인건비단가를 곱하여 기초인건비를 산출한다. 그리고 기초인건비가 산출되면 규모별 보정계수, 프로젝트 형태별 보정계수, 개발언어별 보정계수, 적용대상 기종별 보정계수를 곱하여 직접인건비를 산출한다. 또한, 개발언어별 보정계수는 소프트웨어 개발 공정중 프로그램작성 공정 이후의 공정에만 적용하고 직접인건비 산출시 동일 프로젝트에 정보처리형태, 프로젝트형태, 개발언어, 적용대상 기종이 혼합되어 있는 경우에는 각각 구분계산하여 합산한다.

후자의 방식에 따르면 직접인건비는 계약에 의해 지원되는 전문인력의 등급별로 지원기간에 따라 엔지니어링사업 노임단가를 적용하여 산정함을 원칙으로 한다.



④제경비 및 기술료의 산정에 있어서 제경비 산정은 직접인건비에 별표 10(생략)의 기준에 따라 제경비율을 곱하여 산정한다. 그리고 기술료는 직접인건비와 제경비의 합계액에 별표11(생략)의 기준에 따라 기술료율을 곱하여 산정한다.

⑤직접경비의 범위는 엔지니어링사업 대가의 기준에서 정하고 있는 것 이외에 당해 프로젝트의 수행을 위하여 특별히 소요되는 다음 각호의 비용을 포함하여 산정한다. 즉, 시스템사용료, 자료입력비, 데이터베이스 구축비, 소프트웨어 Tool 사용료, 선투자 후정산 사업으로 추진되는 사업의 경우 지급이자, 수주자 및 발주자의 합의에 의한 특정외국기술 도입과 관련된 외국인

기술자 비용, 기타 일반 용역업무와는 달리 소프트웨어 개발에 특별하게 소요되는 직접경비 등이 있다.

(3) 유지보수 및 재개발시의 대가 산정

유지보수 수행형태는 자체유지보수와 용역유지보수로 구분하며 자체유지보수란 개발이 완료된 소프트웨어를 발주자가 관리하는 것을 말하고, 용역유지보수란 용역업체에 용역을 주어 소프트웨어를 관리하는 것을 말한다.

재개발 소프트웨어의 개발비 산정시에는 직접인건비에 재개발 정도에 따라 용역 수·발주자가 상호협의하여 재개발을 곱하고 제경비, 기술료 및 직접경비를 합하여 산정할 수 있다.

소프트웨어개발시 상기의 정보통신부기준을 적용함에 있어서 검토되어야 할 문제점으로는⁴⁰⁾

첫째, 현재 국내의 현실은 본 개발단계 중심으로만 개발비가 견적되고 있으며, 본개발단계의 선행단계(제안 및 심사)와 후적단계(하자보수)에 대해서는 비용견적 및 정산이 안되고 있어 개발업체의 적자가중의 요인이 되고 있다. 따라서 단계별 합리적인 소프트웨어 개발비 견적기준 및 제도도입이 요구된다.

둘째, 현행 소프트웨어 개발시 투입공수 산정은 프로그램 본수의 추정, 프로그램 유형별 평균 스텝수로 환산, 총 프로그램스텝수에 프로그램 언어별 보정계수의 적용과 공정별 생산성기준표에 의해 기초소요공수가 산정되고 여기에 프로그램 규모별, 형태별 보정계수를 적용하여 투입공수를 산정한다. 그러나 실제적으로 프로젝트 초기 단계에는 프로그램 본수의 추정 및 프로그램 유형별 스텝수의 정확한 산정은 어려운 실정이므로, 프로젝트의 규모, 특성,

40) 김길조·박승원; "소프트웨어개발원가의 특성 및 회계처리에 관한 연구", 『회계학연구』 제10호, 1990, p.129.

환경 등을 고려한 한국형 모델에 의한 투입공수 및 개발기간 산정모델의 도입이 필요하다 하겠다.

셋째, 현재 통용되고 있는 인력의 노임단가는 과학기술처의 기술용역 대가 기준의 전문기술업 노임단가를 적용한 것으로 실제보다 매우 낮게 책정되어 있으며, 원가구성항목도 직접원가(직접인건비, 직접경비), 간접원가(판매비/일반관리비 등), 기술료 등으로 일관성있게 조정되어야 한다.

네째, 소프트웨어개발 입찰제도는 가격중심평가제도에서 가격, 기술력, 실적 등을 고려한 종합평가제도로 개선되어야 한다.

다섯째, 소프트웨어 개발원가는 소프트웨어 품질의 질 및 양에 따라 커다란 차이가 발생되므로, 품질보증기준에 의한 개발원가의 산정이 이루어져야 한다.

2. 미국의 회계처리기준

1) 소프트웨어개발에 따른 회계처리규정의 변화

소프트웨어개발의 회계처리에 대한 미국 최초의 공식적인 규정은 IBM이 소프트웨어의 가격분리정책(unbundling)을 발표한 1969년에 발효된 미국 내국 세입청의 세입절차⁴¹⁾를 들 수 있다. 동 규정에 의하면 제작 소프트웨어는 하드웨어와 동시에 자산계상하고 구입 소프트웨어는 자산계상하는 반면 자사개발 소프트웨어는 판매·사내이용 어떤 것에 개의치않고 연구개발비로 비용처

41) IRS, "Guidelines in connection with examination of Federal income tax returns involving costs incurred to purchase, or lease computer software", IRS, 1969, REV. proc.69-21.

리하게 하도록 하고 있다. FASB 기준 제2호 「연구개발비회계」 [1974]에서도 판매용 자사개발 소프트웨어비는 연구개발비로 인식하여 비용이라는 관점에서 이 문제가 다루어 졌다.

그리고 연구개발비회계기준[FASB,1974]에 의하면, 소프트웨어회계를 처음 취급할 때 소프트웨어개발비는 전액을 연구개발비로하여 당기비용으로 처리할 수 있도록 하였다. 그 이유로서 동 기준에서 다음의 5가지 요인을 들고 있다.

- ①장래의 효과가 불확실하다.
- ②지출과 장래 효용 사이에 인과관계가 불명확하다.
- ③경제재로서 인식된다. 해도 회계상 측정가능성을 만족시키지 않는다.
- ④비용 수익 대응이 되지 않는다.
- ⑤자산에 계상한다해도 기업의 잠재적 수익력 평가에는 도움이 없다.

그 배경으로서는 당시 소프트웨어 개발기술은 실패의 위험성이 대단히 높았다는 것을 들 수 있다.

최근의 소프트웨어개발 기술은 비약적인 발전에 따라 실패할 위험성이 아주 적어지고 있으며 제작된 소프트웨어는 공업생산물의 성격이 강한 경제재로서 인식되고 있다. 따라서 무형의 이유만으로 자산성을 인정하지 않는다는 것은 이론적으로 타당하지 않다는 논쟁이 시작되고 있다. 일반적으로 구입 소프트웨어를 자산계상한다고 하면 같은 용역잠재성을 가진 소프트웨어를 자사개발이라는 이유만으로 비용화 한다는 것은 명백한 논리 모순이라고 사료된다. 소프트웨어 중에는 무형으로 존재한다는 것을 제외하면 기계설비와 본질적으로 크게 다르지 않다는 사실에는 놀라지 않을 것이다. 또한, FASB 해설 제6호 「FASB 기준 제2호의 컴퓨터소프트웨어의 적용가능성」 [1975]와 FASB 기술공보(technical bulletin) 「컴퓨터소프트웨어 회계」 [1979]에서도 이와 같은 입장에서 연구개발비로서 인식하여 비용처리 하도록 규정하고 있

다.

이와 같은 경제실태의 변화를 반영하여 미국에서는 1980년대 이르자 소프트웨어의 자산계상을 하고자 하는 기업이 속출하고 있으며, 그 중에서 자산성에 의문이 있는 것까지 자산계상하는 기업이 나타났다. 그리하고 SEC는 1983년에 통첩(33-6476)을 발표하고 FASB가 자산성을 인정하기까지는 새로운 회계방침으로서 자산계상으로 변경할 결정을 해서는 안된다고 규정하고 있다.

FASB 공개초안 「판매·리스 등 시판용 컴퓨터소프트웨어비의 회계」 [1984]에 의하면⁴²⁾, 당시까지 연구개발비로서 처리해온 판매용 자사개발 소프트웨어비(코딩과 테스트)를 자산계상할 수 있도록 하는 규정이 공표되었다. FASB에서는 회계처리대상을 주로 판매용 소프트웨어비에 한정하였으나, 한편, 미국회계인협회(NAA)의 관리회계실무위원회(MAP)의 검토 보고서 「사내이용 목적을 위한 소프트웨어 회계」 [1985]에서는 사내이용목적을 위한 소프트웨어비의 자산성에 논의를 전개한 후 동 소프트웨어비를 소프트웨어에 따라 자산계상을 해야 한다고 규정하고 있다.

FASB에 의하면, 경제재는 자산으로서 인식해야 할 기준을 측정가능성에 있다고 하고 미래의 경제적 편익이 객관적으로 측정된 경우뿐만 아니라 자산으로서 인식되어야 할 것이라고 규정하고 있다. 따라서 경제재가 새롭게 창조된다고 하는 확실한 증거가 없는 한 자산으로서 계상해서는 안 되었기 때문이다.

N.E Paulsen은 소프트웨어회사가 노무비와 제조간접비를 소비하고 소프트웨어를 구입해서 수익이 생겨나는 제품을 생산하는 것은 제조회사가 제품을 생산하는 과정과 본질적으로 다르지 않으며, 그것이 무형인 소프트웨어제

42) FASB, Exposure Draft, "Accounting for the Costs of Computer Software to be Sold, Leased, or Otherwise Marketed.", FASB, 1984.

품에 지나지 않는다는 견해를 표명하고 있다.⁴³⁾ 이처럼 생각할 때 소프트웨어는 공업생산물과 구별된 소프트웨어제품으로서 받아 드릴 수 있다. 여기서 소프트웨어제품과는 Paulsen에 의해 다음과 같이 정의를 부여할 수 있다. 소프트웨어 제품과는 가격을 가진 시장에서 판매할 수 있는 식별가능한 최초의 판매가능 단위를 구성하는 기존 지식의 기술을 사용하고 그 일련의 프로그램은 설계, 기술, 테스트 및 디버깅한 결과이다.⁴⁴⁾

이와 같이 소프트웨어도 제품으로써 생각될 수 있게 된 배경에는 범용팩키지 소프트웨어가 증대되고 있다는 사실에 있다. 향후 이러한 인식이 점점 확산될 것으로 예상되며, 사실 FASB에서도 1984년 8월 공개초안에 있어서 자산성판정을 기준으로서 측정가능성에 변화해서 투자자산 회수가능성을 들고서 소프트웨어의 자본화를 인정할 수 있도록 전진했다. 그리고 더욱이 1985년 8월에는 최종보고서로서 기술적 실행가능성을 조건으로 소프트웨어의 자산계상을 허용하는 것을 확정한 것이다.

FASB는 소프트웨어비의 지출에 대한 회계처리기준으로서 1985년 8월에 재무회계기준 보고서 제86호를 공표하였으며, 이 보고서는 FASB 기술공보 제79-2⁴⁵⁾호에 우선하며, 보고서 제2호의 일부규정⁴⁶⁾과 해설서 제6호의 일부 규정⁴⁷⁾을 제거시키고도 있다.

Ericson등은 “동 보고서의 기준이 적용된다면, 많은 기업 특히 소프트웨어비를 비용화한 기업의 보고이익은 증가하고, 소프트웨어 투자액이 많아질 수록 기준 없이 처리하던 때 보다 보고이익은 계속 증가할 것이다. 그리고

43) Neil E.Paulsen, "Software Development Costs Should be Capitalized," Management Accounting, 1983, p.41.

44) Neil E.Paulsen: 上掲書, p.41.

45) FASB, Technical Bulletin No.79-2, "Computer Software Costs," Stamford, Connecticut, FASB, 1979.

46) FASB, SAFS No.2, "Accounting for Research and Development Costs," : FASB, 1974.

47) FASB, Interpretation of Financial Accounting Standards No.6, 1975.

증가의 범위는 제품의 특성에 크게 좌우될 것이지만, 최첨단의 제품을 개발하는 기업은 그렇게 큰 변화는 없을 것이다. 그러나 설게 보다 상대적으로 코딩과 테스트를 적게 함으로써 혁신적인 제품개발이 부진한 기업은 그 변화가 아주 클 것이다. 어떤 기업에는 이 기준이 적용되면 비용의 5%내지 10%가 자본화될 것으로 추정되며, 심지어 어떤 기업은 50%이상이 자본화될 것으로 추정할 수 있다. 또한 자본화로 인하여 그 금액이 자본화되고 그 후 순실현가능원가까지 상각되어야 한다면, 보고이익이 매우 심하게 변동할 것이라는 점을 지적하지 않을 수 없다“라고 언급하고 있다.

2) FASB의 SFAS No.86 규정

SFAS No.86 「판매·리스 등 시판용 소프트웨어개발비의 회계처리」 보고서에서는 범위, 재무회계 및 보고기준(소프트웨어 연구개발비, 소프트웨어 제조원가, 취득 소프트웨어비, 재고자산비용, 자본화된 소프트웨어비의 평가) 기타 공표물의 수정, 부록으로 구성되고 있다. 여기서는 재무회계 및 제조기준만을 요약한다.⁴⁸⁾



(1) 소프트웨어개발비의 회계처리기준

SFAS No.86[1985]는 내부적으로 개발, 생산되거나 구입한 것에 관계없이 분리가능한 제품으로서 혹은 제품이나 공정의 일부분으로서 판매, 리스하거나 혹은 시장화하고자 하는 소프트웨어비에 대한 재무회계와 보고기준으로 사용될 수 있는데, 그 중요한 내용은 다음과 같다.

①소프트웨어의 연구개발비 즉, 판매나 리스, 시장화하기 위한 컴퓨터 소프트웨어 제품의 기술적 실행가능성을 위해 지출한 비용은 FASB규정에 의거하여 발생시의 비용으로 처리하도록 규정하고 있다.

48) FASB, SFAS No.86: 前掲書, 1985.

②소프트웨어의 생산비용 즉, 기술적 실행가능성을 위해 발생한 제품의 생산비용은 자본화 되어질 수 있도록 하고 있으나, 제품이나 공정의 내부 부분으로 사용되어지는 소프트웨어를 위한 소프트웨어 제조원가는 기술적 실행가능성이 소프트웨어를 위해 확정지어졌고, 제품이나 공정의 다른 구성품을 위해 모든 연구개발활동이 완성되었을 때까지는 자본화할 수 없도록 하고 있다. 또한 소프트웨어비의 자본화는 제품이 소비자에게 출하가 가능할 때 끝날 것이며, 유지비와 소비자에 대한 지원비는 관련 수익이 인식되었을 때나 그런 비용들이 발생되었을 때 비용으로 된다고 규정하고 있다.

③대체적인 미래의 사용을 가지지 않고 팔거나 리스 또는 시장화하기 위해 구입된 소프트웨어비는 내부적으로 그런 소프트웨어를 개발하기 위해 발생한 원가와 같이 비용으로 회계처리 해야 하나, 대체적인 미래 사용을 가지고 구입된 소프트웨어비는 소프트웨어가 구입되었고, 그것이 사용되어질 때 자본화되어야 하도록 하고 있다.

④자본화된 소프트웨어는 생산량 기준으로 상각할 수 있으며, 상각은 제품이 소비자들에게 출하가 가능할 때 시작된다.

⑤자본화가된 소프트웨어비의 평가에 대한 규정으로서, 대차대조표에 상각되지 않고 자본화된 소프트웨어비용 즉, 순실현가능가치(NRV)를 초과하는 소프트웨어비는 상각되어질 수 있도록 하고 있다.

⑥대차대조표에는 미상각된 소프트웨어비, 그리고 손익계산서에 있어서는 비용으로 될 총액 즉, 자본화된 소프트웨어의 상각액과 순실현 가치로 감소시킨 금액 등을 각각 표시하도록 하고 있다.

그리고 소프트웨어 제품은 일반 제품과는 달리 개발비의 대부분이 연구개발비에 해당되고, 생산비는 극히 일부분에 지나지 않는다는 점을 고려할 때, 미래효익을 기대할 수 있는 임시 거액의 연구개발비까지도 당기의 비용으로 처리함은 매우 불합리하며, 또한 소프트웨어를 제품화하기 위한 목적이 아니

고, 자사용으로 개발하는 경우에는 지출 전액이 당기 비용으로 처리되므로서 자산가치의 평가가 왜곡되는 문제점을 안고 있다. 또한 순실현가치는 시장판 매가격에서 처분비용을 차감한 것으로서, 소프트웨어의 시장가격산정은 연구 개발비 뿐만 아니라 개발에 실패한 비용까지도 고려되어야 하므로, 소프트웨어 개발비의 자본화 문제는 매우 복잡한 문제라 할 것이다.

(2) 소프트웨어관련 비용의 구분과 회계처리규정

① 소프트웨어 연구개발비

매각, 리스 및 시판용 소프트웨어의 기술적 실행가능성을 확립하기 위해 소요된 모든 비용은 연구개발비다. FASB의 SFAS No.2에 규정된 바와 같이 이들 모든 비용은 발생하였을때 비용화해야 한다. 이 보고서의 목적상 소프트웨어의 기술적 실행가능성은 기업이 제품의 기능, 특징, 기술적 성과 요구 사항을 충족시켜 제품생산에 필요한 모든 계획, 설계, 코딩, 테스트 활동을 완료했을 때 확립된다.

② 소프트웨어 제조비용

기술적 실행가능성을 확립한 이후에 발생한 마스터 제품의 생산비용은 자본화되어야 한다. 이 원가에는 기술적 실행가능성을 확립한 이후에 수행된 코딩과 테스트 비용이 포함되어 있다. 소프트웨어 제조원가가 제품 또는 공정 전체의 일부분으로서 사용되는 경우, 소프트웨어의 기술적 실행가능성이 확립되고, 제품 또는 공정의 기타 구성요소에 대한 연구개발활동이 완료된 후에 자본화되어야 한다. 소프트웨어를 고객이 널리 사용할 수 있는 경우 그 소프트웨어비의 자본화는 중지될 것이다. 수익과 비용 어느 것이 먼저 발생 하던 관련수익이 인식되거나 관련비용이 발생하였을때 유지원가나 고객지원 원가는 비용화되어야 한다.

③ 취득 소프트웨어비

미래의 대체적인 용도를 갖고 있지 않으면서 매각, 리스 및 시판하기 위해 취득한 소프트웨어비는 소프트웨어를 내부적으로 개발하는데 소요된 비용과 동일하게 회계처리하여야 한다. 만약 취득한 소프트웨어가 미래의 대체적인 용도를 갖는다면, 소프트웨어를 취득하였을때 그 비용은 자본화하고, 사용하는데 따라 회계처리하여야 한다.

자본화된 소프트웨어비는 제품별 기준에 의해 상각되어야 한다. 연간 상각액은 현행 제품 총수익과 장래 제품 총수익의 합계액에 대한 비율 또는 보고하려는 기간을 포함한 잔존 추정 내용연수에 걸쳐 직선법을 사용하여 계산된 금액 이상이어야 한다.

④재고자산비용

마스터 제품으로부터 소프트웨어를 복제하고 문서화하며 훈련자료를 만드는 데 소요된 비용과 유통시킬 제품을 실제로 패키징(packaging)하는데 소요된 비용은 개별법에 의해 재로자산으로 자본화시켰다가 매출로 인한 수익이 인식되었을 때 매출원가로 비용화한다.

⑤자본화된 소프트웨어비의 평가

대차대조표 작성일에 자본화된 소프트웨어 제품의 비용 중 미상각분은 제품의 순실현가능성 가치와 비교해야 한다. 자본화된 소프트웨어 제품의 비용 중 미상각분이 그 자산의 순실현가능 가치를 초과하는 금액은 상각해야 한다.

2) 내국세 관련규정

IRS(미국 내국세입청)은 소프트웨어를 무형이라고 규정하고 있으며, 우선 내국세법에서는 소프트웨어개발비는(특정 소프트웨어에 특허권 및 저작권의 유무에 관계없이) 많은 측면에서 내국세법 제174조에 해당하는 연구개발비에

유사하다라고 규정하고 있다. IRS[1969]에서는 비용화되어야 한다고 되어 있다. 한편, 미국의 연방소득세법의 관련규정<Rev, proc.69-21>에 의하면, 어떤 특정한 요소에 초점을 맞추고 있지 않고 납세자(기업)의 자유로운 선택에 맡기고 있다. 즉 자사개발이나 위탁개발에 상관없이 모두 당기비용으로 처리할 수 있도록 하고 있다. 더우기 위탁개발 소프트웨어비를 비용처리한 경우에는 연구개발비세법공제가 25%(다만, 65%에 한해서)받아들여지고, 역으로 무형고정자산에 계상하면 10%의 투자세액공제가 받아들여진다고 하는 특전이 있다. 경제재로서 자산성이 확실한 소프트웨어는 비용·수익을 적시에 대응시키기 때문에 자산계상하는 것이 이론적으로도 타당성을 갖는다.

<Rev. proc.69-21>

(1)소프트웨어의 개발비는 계속적용을 조건으로 하여 다음의 ①과 ②의 방법 중에 납세자가 선택하여 적용할 수 있다.

①전액을 발생시에 비용으로 하는 방법

②전액을 자본적 지출로 하여 개발완료시부터 5년간 균등상각하는 방법
(단, 납세자의 입증에 따라 상각기간을 단축할 수 있다)

(2)소프트웨어의 매입비는 계속적용을 조건으로 하여 다음의 ①과 ②의 방법 중에 납세자가 선택하여 적용할 수 있다.

①소프트웨어의 비용을 구별하지 않고, 하드웨어의 원가에 포함하여 감가상각을 하는 방법

②소프트웨어의 비용을 무형고정자산으로 하여 5년간 균등상각하는 방법
(단, 납세자의 입증에 따라 상각기간을 단축할 수 있다)

(출처) IRS, Rev. Proc. 69-21, "Guidelines in connection with examination of federal income tax returns involving costs incurred to purchase, or lease computer software", 1969.

상기 규정은 소프트웨어개발비를 개발형태(자체개발, 위탁개발)와 관계없이 시험연구지출(research or experimental expenditure)로 보고 있기 때문에, 개발에 의해 특허권 또는 저작권을 취득하는가에 관계없이 적용된다. 또한, 상기 규정은 개발목적(내부이용, 판매·리스)과 관계없이 적용되는데 FASB SFAS No.2와 No.86에서와 같이 개발목적에 따른 발상은 존재하지 않는다.

3. 일본의 회계처리기준

1) 기업회계원칙의 관련규정

일본에서는 소프트웨어에 관한 회계기준은 없기 때문에 세법의 규정에 따라 소프트웨어개발비에 대한 회계처리를 실시해야 한다.

2) 법인세법 관련규정

일본에서도 상법[제286조의 3]과 기업회계원칙[제3의 4-1]에 의해 신제품, 신기술의 연구를 위해 특별하게 지출된 비용은 이연자산(시험연구비)으로 계상하여 5년 이내에 상각하여야 하고, 특별하게 지출되지 않은 비용은 당기비용으로 처리하든가의 선택이 인정되어 있다.

일본의 법인세법에서는 다음의 관련규정 「기본통달 8-1-7」에서 보는 바와 같이, 개발형태에 초점을 맞추고 있으며, 위탁개발·매입, 자체개발 및 기술자의 파견유입에 따라 자본화의 여부를 판단하도록 하고 있다.

<법인세법 기본통달 8-1-7>

타인으로부터 소프트웨어를 매입하거나 또는 타인에게 위탁하여 소프트웨어를 개발한 경우에 그 제공을 받기 위해 또는 위탁하는 데 소요된 비용은 법인세법 시행령 제14조 제1항 제9호(용역의 제공을 받기 위한 권리금 등)에서 규정한 이연자산에 해당한다. 그러나, 법인이 소프트웨어의 개발에 관련하여 타인에게 지급한 비용 중에 다음에 열거한 것은 본문에 기재되는 비용에 해당되지 않는다.

- ① 자체소프트웨어를 개발하기 위하여 타인으로부터 기술자를 파견받은 경우(실질적으로 당해 타인에게 개발을 위탁하였다고 인정되는 경우는 제외)의 그 파견을 받기 위해 소요된 비용
- ② 자체개발한 소프트웨어에 대해 타인에게 코딩만을 맡긴 경우의 그 코딩에 소요된 비용

상기 규정은 공평과세의 관점에서 외부자료에 의해 객관적으로 금액을 확정할 수 있는 소프트웨어는 이연자산으로 계상하고, 그렇지 않은 소프트웨어는 당기비용으로 처리한다는 것으로 해석할 수 있다. 그리고 소프트웨어개발의 원가계산이 완전하게 행해지지 않고, 자체개발한 소프트웨어비를 정확하게 계산할 수 없다는 과세기술상의 문제도 반영되어 있다고 추측할 수 있다.

그러나, 상기 규정은 개발형태에 따라 획일적으로 처리하는 것이 큰 문제이다. 즉, 같은 응용소프트웨어라도 위탁개발한 경우에는 이연자산으로 계상하고 자체개발한 경우에는 비용으로 처리하는 차이가 생긴다. 이것은 수익·비용대응의 관점에서 보면 불합리한 것이다. 또, 판매를 목적으로 개발한 소프트웨어의 회계처리에 대해서는 전혀 언급이 없기 때문에 많은 혼란이 발생할 우려가 있다.⁴⁹⁾

49) 김준식·이호열: 前掲書, p.50.

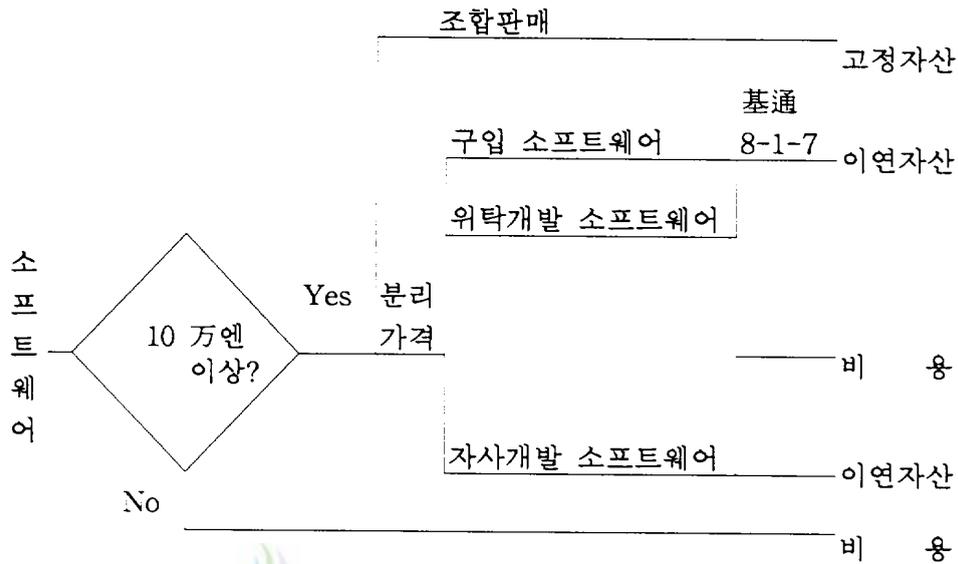
또한, 소프트웨어개발비의 회계처리에 있어서 가장 어려운 점은 공업제품과는 달리 개발상의 기술적인 문제가 모두 해결된다고 하는 것이 반드시 해당 소프트웨어가 미래 수익획득 능력이 있다고 말할 수 없는 것이다. 몇년 걸려서 연구개발을 하고 개발에 성공하여도 판매되거나 이용되지 않는다면 무형으로 처분가치가 없는 소프트웨어는 한푼의 가치도 없는 것이 된다. 따라서 기술적 실행가능성 확립 후의 판매용 소프트웨어의 자산계상을 강제하는 것은 불량자산을 계상하게 하는 위험성을 증대시키는 것이 된다.

일본조세연구협회의 세무회계상임위원회가 1985년 11월에 발표한 「소프트웨어의 회계기준」 [일본조세연구협회, 1985] 및 郵政省이 1986년 9월에 발표한 「텔레콤 세제기본문제 연구회중간보고서」 [郵政省, 1986]의 양자가 기업이 자유의사에 선택을 맡긴다고 하는 것은 이러한 이유에서이다. 그렇지만 소프트웨어 개발비의 자산계상을 기업의 자유의사에 맡기는 것만으로 문제가 해결된다는 것은 아니며 기술적 실행가능성의 테스트 이외에 소프트웨어 자산성 판정기준을 매매계약이나 주문서에 따라 판매가능성이 객관적으로 인정할 소프트웨어만을 자산계상 하도록 하면 어떤가하고 주장하는 학자도 있다.

법인세법 기본통달(8-1-7)에 의하면 자사개발 소프트웨어는 비용처리가 허용되어 있는 반면 위탁개발 소프트웨어 전체의 비용은 미국과는 달리 자산계상이 강제되어 있다. 이러한 위탁개발 소프트웨어의 자산계상은 일본기업에서 반대가 강하다. 특히 미국에서는 비용화와 자산계상의 선택적용이 인정되는 것에 대해 불공평감을 갖는 실무자도 많다. 일본조세연구협회는 「소프트웨어 세무회계기준(안)」 [1985]을 공표하고 위탁개발 소프트웨어비의 비용·이연자산의 선택적용을 제안하고 있다. 이것은 일본 주요 기업이 위탁개발 소프트웨어의 자산계상 강제화에 반대의 입장을 표명하고 있는 명백한 증거이다.

위탁개발 소프트웨어는 구입과 같은 양상으로 간주되고 파견과 코딩을 빼고 위탁소프트웨어의 비용과 일체가 되어 판매(조직판매)된 소프트웨어는 고정자산으로 취급된다. 이 내용을 <그림4-1>과 같이 도표화 할 수 있다.

<그림4-1> 세법에 의한 소프트웨어 회계처리법



(출처) 櫻井, 1987. p.31<그림2-1>인용



현행 세법에 있어서 가장 문제가 되는 것은 위탁개발 소프트웨어의 자산 계상에 대한 것이다. 위탁개발 소프트웨어의 개발비를 자산계상시키고 있는 이유는

- ①계약에 따른 금액이 확정되기 때문에 객관성으로 축적하는 것.
- ②실패의 위험성이 비교적 적다고 하는 점.

그렇지만 기업의 실태는 이와 같이 입법의 취지와는 전혀 다르다. 위탁 개발 소프트웨어라고 하는것은 본래에 자사에서 개발해야 할 소프트웨어를 정책상 등의 이유로 인해 子회사 등을 통해서 제작시키는 것이 많다. 이 때

문에 같은 성질의 소프트웨어에도 母회사에서 제작하면 비용처리하는 데 자 회사에서 만들어 지도록 하는 것이 왜 계상하지 않으면 안될까라는 의문이 생긴다.

현재 일본 세법상의 자사개발 소프트웨어비의 회계처리기준은 비용처리·자산계상의 선택적용이 허용되어 있다. 그러나 기술적 실행가능성과 판매가능성이 확인되는 한 자사개발 소프트웨어라고 하더라도 자산성이 인정되어질 뿐인 것이다. 판매용의 자사개발 소프트웨어의 제작원가는 공업생산물에서 말한다면 제품의 제조원가에 상당함으로 원칙적으로는 재고자산에 계상하여 매출시에 매출원가로 처리해야만 한다.

실태조사에 따르면 일본의 약 4%의 기업이 재고자산으로 처리하고 있으나 그 이유는 이러한 경제실태를 반영하고 있기 때문인 것으로 생각된다. 더우기 소프트웨어에 의해서는 장기간에 걸쳐 그 효과를 볼 수가 있는 것이다.

소프트웨어는 장기성 자산에 계상하는 것이 타당하다. 소프트웨어를 자산 계상하게 된다면 장기성 자산의 종류로서는 회계상의 이연자산(시험연구비), 세법상의 이연자산, 장기선불비용 및 무형고정자산 그 어느 것으로 계상할 것인가는 소프트웨어개발을 연구개발이라고 규정한다면 회계상의 이연자산으로 처리하는 것이 타당하다. 그러나 기술적인 개발능력이 진보된 현재로서는 모든 소프트웨어개발을 연구개발활동이라고 규정하기란 곤란한 것이다.

또한 일본에서는 소프트웨어를 일종의 노하우라고 정의하는 일이 있다. 분명히 개변권, 사용허락권을 지닌 구입 소프트웨어 등에는 노하우의 측면이 강하다. 따라서 법인세 기본통칙(8-1-7)이 소프트웨어를 세법상의 이연자산으로서 규정한 것은 무리라고 할 수가 없는 일이다. 노하우로서의 처리를 범용패키지 소프트웨어의 출현에 따라 소프트웨어가 유형이라고 하는 유력한

50) 한국세제연구회: 前掲書, p.86.

견해가 나타난다고 하는 점 및 세법상의 이연자산이 상법상에서는 이연자산으로서의 계상이 인정되어 있지 않다고 하는 점의 두가지의 문제점이 있다. 소프트웨어개발을 일종의 투자성질을 지닌 것으로 취급할 경우에는 장기선불비용에 계상하는 것이 타당하다.

일본의 법인세법에 따르면 소프트웨어가 하드웨어의 원가에 포함되어 있을 때 그것은 하드웨어의 원가의 일부로서 취급되며 자산에 계상되어 하드웨어의 상각기간에 따라 상각된다. 한편 소프트웨어가 하드웨어와는 별도로 판매될 경우에는 그의 원가는 위탁개발에 국한하여 자산계상되며 5년으로 상각된다.

상기의 내용을 요약 정리하여 보면 <표4-3>과 같다.

<표4-3> 한·미·일 소프트웨어개발비의 회계처리기준 비교

구분	적용기준	회계처리	회계처리의 주요내용	비고
한 국	기업회계기준 제23조 5항	자 본 화	· 비경상적으로 발생한 비용 · 미래의 경제적 효익 기대	연구개발비
		비 용 화	· 경상적으로 발생한 비용 · 미래의 경제적 효익 불확실	당기비용
	연구개발에 관한 회계처리 준칙 제10조	자 본 화	· 법률적 권리, 경제적 권리가 무형고정 보장된 경우에 이와 관련하여 지출한 비용	자산
	법인세법 시행령 제38조 제1항 제4호	자 본 화	· 위탁개발을 위하여 특별히 지시됨 · 하드웨어의 작동을 위한 시스템소프트웨어는 하드웨어의 취득가액에 포함	연구비

구분	적용 기준	회계처리	회계처리의 주요내용	비고
미국비회계기준	FASB No.2, 1974“연구개발	비용화	· 판매용 자사개발 소프트웨어비	연구개발비
	FASB 해설 No.6(1975) 및 FASB 기술공보(1979)	비용화	· FASB No.2와 동일하게 처리	연구개발비
	FASB 공개초안 (1984)	자본화	· 자사개발 판매용 소프트웨어비에 한정	
	NAA(1984)의 MAP “검토보고서의 사내이용목적에 위한 소프트웨어 회계”	자본화	· 사내이용목적에 위한 소프트웨어는 소프트웨어에 따라 자산계상	
	FASB No.86, 1985 “판매·리스 등 시판용 소프트웨어 개발비의 회계처리”	자본화	· 기술적 실행가능성을 위해 발생한 제품의 생산비용 · 미래 사용을 가지고 구입된 소프트웨어비	
		비용화	· 판매나 리스, 시장화하기 위한 컴퓨터소프트웨어 제품의 기술적 실행가능성을 위해 지출한 비용 · 판매나 리스, 시장화하기 위한 구입된 소프트웨어비	
	내국세 제174조	비용화	· 소프트웨어개발비는 연구개발비와 유사하다고 규정	연구개발비
연방소득세법 Rev.proc69-21	선택적용	· 소프트웨어개발비는 계속 적용을 조건으로 전액발생시 비용 또는 자본적지출 · 소프트웨어매입비는 계속 적용을 조건으로 하드웨어의 원가에 포함 또는 무형고정자산		

구분	적용 기준	회계처리	회계처리의 주요내용	비고
	기업회계원칙	-	· 일반적으로 세법의 규정에 따라 실시	
일본	상법 제286조의 3	선택적용	· 신제품, 신기술의 연구를 위해 시험연구비 · 특별하게 지출된 비용 · 특별하게 지출되지 않은 비용 당기비용	
	법인세 기본통자본화 달 8-1-7	자본화	· 소프트웨어가 하드웨어의 원가에 포함되어 있는 경우는 하드웨어 원가의 일부로 취급 · 소프트웨어가 하드웨어와 별도로 판매된 경우 그의 원가는 위탁개발에 국한하여 자산계상 · 매입 및 위탁개발 소프트웨어비는 이연자산	
		비용화	· 자사개발 소프트웨어비는 비용처리	

그리고 기타국의 소프트웨어개발비에 대한 회계처리를 살펴보면 다음과 같다. 영국에서는 소프트웨어개발비에 대해 특별히 규정된 기준은 없으나 몇 가지의 기준을 갖춘 자사개발 판매용 및 사내이용 소프트웨어개발비는 자본화된다.(기준을 갖추지 못하고 있는 것은 비용처리)

또한, 캐나다에서도 특별한 규정은 없으나, 자사개발 판매용 소프트웨어개발비는 FABS에 준해 처리하고 사내이용 소프트웨어개발비는 자본화 또는 비용처리 하고 있다.

독일에서는 원칙적으로 판매용 소프트웨어개발비는 자산계상되고 있으며, 사내이용 소프트웨어개발비는 통상적으로 비용처리 한다. 그리고 오스트레일리아에서는 판매용 및 사내이용 소프트웨어개발비는 비용처리하고 있다.

V. 요약 및 결론

소프트웨어개발비에 대한 회계처리는 소프트웨어개발비를 먼저 자본화할 것인가 또는 당기비용으로 할 것인가라는 두가지의 문제점으로 나누어 진다고 할 수 있다.

먼저 우리나라의 기업회계기준의 측면에서 살펴보면 「기업회계기준」 제 95조의 규정에 의거 연구개발활동 및 이와유사한 활동에 관한 회계처리와 재무보고에 관하여 필요한 세부사항을 정하고 있는 「연구개발비준칙」에 의하면, 연구개발비에 대해서 신제품, 신기술의 연구 또는 개발활동과 관련하여 비경상적으로 발생한 비용으로서 미래의 경제적효익을 기대할 수 있는 것으로 규정하여 제품화를 위한 개발 또는 자사목적은 개의치 않고 자본화할 수 있게 하였다. 또한 연구개발비는 계상년도 부터 5년기간에 걸쳐 매년 균등액을 상각하도록 규정하고 있다.

법인세법에 의하면 조세자료 87-45에서 컴퓨터 프로그램 개발용역이 시험연구비인지에 관한 질의 응답에서 국세청 법인 22601-2427[87.9.8]호에 의한 해석에서 새로운 전산시스템(소프트웨어)을 개발 또는 위탁개발하기 위하여 특별히 지출한 비용은 법인세법 시행령 제38조 제1항 제4호의 규정에 의한 시험연구비에 해당한다고 하고 있다. 다만 하드웨어 작동을 위한 시스템소프트웨어는 하드웨어 가액에 포함한다고 하고 있다.

또한 법인세법 시행령 제38조에 의하여 시험연구비를 이연자산으로 계상하지 아니하고 직접 당해년도에 손금으로 처리할 경우에도 소득금액 계산상 손금허용되도록 되어 있다. 즉 5년간 균등상각과 당해년도 손금처리 모두를 인정하여 주는 셈이 되는데 이것은 기준이 애매한 것으로 정비를 시급히 해

야할 실정이다.

현재까지 구입소프트웨어나 위탁개발 소프트웨어에 대해서는 대다수 비슷한 결론에 도달하고 있으나 가장 문제가 많은 자사개발 판매용 소프트웨어에 대해서는 전액 비용설, 특정 공정원가 자본화설 및 기업임의 선택론에서 일본은 주로 기업선택론에 긍정적인 반응을 보이는 반면 특정 공정원가 자본화설에 대해서는 포괄적인 입장을 취하고 있다. 또한 미국에서는 당초 비용화에서 자본화로 흐르는 뚜렷한 경향을 보이고 있으며 특정공정의 자본화론에 많은 동조를 하고있는 것으로 나타났다.

위 사항을 요약정리하면 소프트웨어개발비에 대하여 자본화 한다면 유동자산(일본·미국의 경우 : 재고자산 또는 재고자산의 일부), 고정자산(우리나라·일본 및 미국의 경우 : 유형고정자산의 일부, 우리나라·미국의 경우 : 무형고정자산), 이연자산(우리나라의 경우 : 연구개발비, 일본의 경우 : 시험연구비), 또 당기비용으로 한다면 연구개발비(미국의 경우), 경상연구개발비(우리나라의 경우), 일반관리비 또는 판매비로 처리할 수 있다.

그러나 아직 우리나라에서는 걸음마 단계인 시험연구비 단계를 벗어나지 못하고 있으며 그것도 당해년도 손비처리를 인정하고 있다. 이러한 것들을 조속히 정비하기 위한 방안으로서,

첫째, 「기업회계기준」의 측면에서 본다면 시급히 학계 및 실무계 등에서 실태조사를 정확히하여 우리나라 실정에 알맞는 소프트웨어개발비의 회계처리 표준화 방안이 정착되어야 한다.

둘째, 정부에서도 정보통신부 및 과학기술처에서도 이미 고시된 소프트웨어개발비 산정기준을 바탕으로 소프트웨어를 개발하는데 소요된 비용에 관하여 현실에 맞는 통일적인 회계처리기준의 제정이 시급하다.

셋째, 소프트웨어개발을 산업으로 보는 인식의 전환과 정부의 기반조성노력도 필요하지만, 무엇보다도 필요한 것은 소프트웨어 개발기업들의 건전한

경영이 전제되어야 하고, 또 그 필수조건으로 소프트웨어개발비에 대한 기업들의 고유한 원가계산제도를 갖춤으로서 적절한 회계처리가 선행되어야 할 것이다.

마지막으로 우리나라 소프트웨어산업의 매출액이 95년도말 GNP의 1.99%로 해마다 급성장 할 것으로 예상하고 있는 바, 소프트웨어산업은 고급기술인력을 대량으로 필요로 하는 자원절약형, 무공해형의 두뇌집단산업이라는 점을 고려할 때 부존자원이 부족한 반면 교육받은 인력자원은 풍부한 우리나라의 여건에 가장 적합한 미래의 성장산업이라는 측면에서 소프트웨어개발비에 대한 회계기준의 마련은 시급한 과제라 하겠다.



참 고 문 헌

1. 국내문헌

- 김길조·박승원; “소프트웨어개발원가의 특성 및 회계처리에 관한 연구”. 『회계학연구』 제10호, 1990.
- 김준식·이호열; “소프트웨어개발비용의 회계에 관한 연구”. 『산업경영』 1989.
- 박성규·박승원; “소프트웨어원가 및 수익인식에 관한 연구”. 『성심논문집』 제25집, 1993.
- 손성호; “컴퓨터 소프트웨어 개발원가의 회계선택에 관한 실증적 연구”. 박사학위논문, 경북대학교 대학원, 1991.
- 위준복·김영수; “소프트웨어 회계처리에 관한 연구”. 『경영논총』 제15집, 1990.
- 유문찬; “소프트웨어의 품질보중에 관한 연구”. 『경상논집』, 1990.
- 이호열; “소프트웨어의 원가계산제도에 관한 연구”. 『산업경영』 제15집, 1992.
- 정홍진; “한국 소프트웨어산업의 원가회계에 관한 연구”. 『중소기업연구』 제15권 제2호(通卷27號), 1993.
- 한국세제연구회; “제외국의 소프트웨어개발비의 회계처리”, 『월간 경영과 세무』, VOL.5, No.49, 1982.
- 한국세제연구회; “일본의 소프트웨어원가관리실체”. 『월간경영과 세무』 VOL.8, No.4, 1989.

- 한국세계연구회; “소프트웨어업에서의 원가계산과 원가관리”. 「월간 경영과 세무」 7월호, 1990.
- 한국세계연구회; “일본의 소프트웨어 회계실무지침의 기본적인 개요”. 「월간 경영과 세무」 VOL.5, No.48, 1992.
- 한국정보과학회; “소프트웨어 시스템의 원가 및 개발기간 평가에 관한 연구”. 「정보과학회논문집」 VOL.11, No.3, 1984.
- 한국정보산업연합회; “소프트웨어원가계산의 원리”. 「정보산업」 8월호, 1988.
- 한국정보산업연합회; “소프트웨어원가계산의 전제와 기준”. 「정보산업」 8월호, 1988.
- 한국정보산업연합회; “소프트웨어원가통제와 표준원가계산제도”. 「정보산업」 8월호, 1988.
- 한국공인회계사회; 「원가계산준칙」, 1994.12.31.
- 과학기술처; 「소프트웨어 개발비 산정기준」, 고시 94-1호.
- 정보통신부; 「소프트웨어 개발비 산정기준」, 고시 제1996-21호.
- 증권관리위원회; 「연구개발에 대한 회계처리준칙」, 1987. 7. 10.
- 증권관리위원회; 「기업회계기준」, 1996. 3.30.

2.외국문헌

- 櫻井通晴; “ソフトウェア原價計算”, 白桃書房, 1987.
- _____ ; “原價計算對象の 變遷”, 「會計」, 1978.
- 太田昭和監査法人; “ソフトウェア開發の 原價管理”. 中央經濟社, 1987.
- 西澤脩; “研究開發費の 會計と 管理”, (新訂版)白挑書房, 1993.

- _____; “情報處理費の會計と管理”, 白挑書房, 1993.
- 日本; “法人稅法” 基本通達 8-1-7.
- FASB, SAFS No.2, “Accounting for Research and Development Costs,” : FASB, 1974.
- _____, Interpretation of Financial Accounting Standards No.6, “Applicabilty of FASB Statement No.2 to Computer Software”, FASB, 1975.
- _____, Technical Bulletin No.79-2, “Computer Software Costs,” : FASB, 1979.
- _____, SFAS No.53, “Financial Reporting by Producers and Distributors of Motion Picture Films,” Stamford Connecticut,” : FASB, 1981.
- _____, Exposure Draft, “Accounting for the Costs of Computer Software to Be Sold, Leased or Otherwise Marketed,” : FASB, 1984.
- _____, SFAS No.86, “Accounting for the Costs of Computer Software to Be Sold, Leased or Otherwise Marketed,” : FASB, 1985.
- IRS, Rev. Proc. 69-21, [1969], “Guidelines in connection with examination of Federal income tax returns involving costs incurred to purchase, or lease Computer Software”.
- Kent T.Fields, Gary L.Waters and James H.Thompson, ‘Accounting for Costs of Certain Computer Software,” the CAP Journal(January 1986).
- McGee, Robert, Accounting for software, Dow Jones-Irwin, 1985.

ABSTRACT

A study on accounting standard for
software development cost

- Focusing on Korea, the U.S.A and Japan -

Mun-Jeong Yang

Department of Accounting
Graduate School of Business Administration
Cheju National University
Supervised by Professor Sang-Bong Lee

As our information society progresses, software development cost and the accounting for these costs has become an important issue. This is because software development costs directly influence period income(loss), financial position and management rationalization of the enterprise.

Software development cost is generally enormous, and yet there has been no standard on the accounting for these costs, leading to discrepancies in accounting practices across different firms. Therefore, it is a significant task to determine how to assess a software development cost properly.

The problem of accounting treatment of software development cost can

be summarized in the following way : ① the development cost is treated as a current period cost and ②the development cost is appropriated as an asset.

The purpose of this thesis is to examine accounting treatment of software development cost in korea, and compare with that used in the U.S.A and Janpan.

Another purpose is to take a direction for establishing the uniform accounting standard on the software development cost. Accounting treatments of software development cost are largely divided into two categories : an asset, or a cost.

Arguments for opposing asset appropriation include ①software development costs are similar to R&D costs, ②it is difficult to assess its asset value, and ③it does not have a fixed accounting standard.

However, there are also number of arguments supporting asset appropriation. ①it is possible to match a revenue cost, and ②purchasing costs are considered to be an asset.

The korean business accounting standards (38-5) say that if one can expect that non-recurring expenditure incurred for software development, the economic benefits in the future are appropriated as an asset. If recurring expenditures create the economic benefits in the future, they are treated as a cost.

On the othertrand, according to accounting treatment of development cost of software for sale and lease [FASB No. 86(1985)] , production cost incurred for a technological feasibility of a product is treated as an asset. Expenditure incurred for a technological feasibility of computer software

products which is to be introduced into the market for sale and lease is treated as a cost.

According to corporation tax standard 8-1-7 in Japan, a cost for software purchasing on consigned development of software is treated as deferred charges and a cost for an inhouse development of software is allowed to be treated as a cost.

If a cost hardware includes a portion of software, software cost is allowed to be appropriated as an asset, and to be depreciated. If software is sold separately from hardware the software cost is treated as an asset only in the case of consigned software development and it is depreciated over 5 year period.

In conclusion our country has to establish a standardized system of accounting treatment of software development costs which is suitable for our situation through the survey of actual condition by both academics and business world.

Also, the government has to develop uniform accounting standards on the expenditures that are incurred for software development. Above all it is necessary for software enterprises to maintain efficient management.

As an essential condition to do so is that all the enterprises has to make proper accounting treatment for software development costs.