

# ソフトウェア再利用に適合的な会計処理-ソフトウェアに係る会計基準の不備を補正するために-

メタデータ	言語: jpn 出版者: 明治大学経理研究所 公開日: 2014-03-06 キーワード (Ja): キーワード (En): 作成者: 長田, 芙悠子 メールアドレス: 所属:
URL	<a href="http://hdl.handle.net/10291/16454">http://hdl.handle.net/10291/16454</a>

# ソフトウェア再利用に適合的な会計処理 —ソフトウェアに係る会計基準の不備を補正するために— A Compatible Accounting for Software Reuse —To revise inadequacy of accounting standards for software—

長田 美悠子

キーワード：ソフトウェア会計，ソフトウェア再利用，情報有用性，ソフトウェア資産，  
振替処理

## 目 次

1. はじめに
2. ソフトウェア再利用の概要
3. ソフトウェア再利用に係る会計処理
  - (1) ソフトウェアに係る会計基準における再利用の取扱い
  - (2) ソフトウェア再利用に適合的な会計処理
4. おわりに

### 1. はじめに

ソフトウェアの開発並びに保守には，ソフトウェア再利用（Reuse）が密接に関係している。しかしながら，開発等におけるそれをも含めた複合的な会計上の取扱いは少々複雑となるため，筆者はそれの利用を行わない「プリミティブな」想定で考察を進めている。しかし，実態的には，おおよそ1970年代以降多少なりともソフトウェア再利用は行われており，それを含めた考察が欠かせない。それ故，本稿ではソフトウェア再利用を主題的に取上げ，それに係る会計処理を考察する。

具体的には，ソフトウェア開発における再利用の特性と，会計におけるソフトウェア再利用の識別の必要性という2つの視点から考察を行う。まず1つ目の視点については，本稿で取上げるソフトウェア再利用は，開発するソフトウェアへの既存のソフトウェア資源の利用であり，意図的に広範囲に利用する場合のことである。既存のソフトウェア資源の再利用の程度や方法，それによって軽減できた開発の工数並びに費用を精確<sup>1</sup>に捉えることによ

り、ソフトウェアの開発費用の内実が明らかとなる。これを踏まえて、2つ目の視点については、保有ソフトウェアの資産形成の効率性を厳密かつ精細に捉えることに見出せる。開発したソフトウェアの一部として組込まれたソフトウェアは、現行制度上は全く会計処理されず、会計的には捕捉不能となっている。つまり、ソフトウェアの資産価値が評価されないことを問題視しており、既存ソフトウェア資源の利用に際しての調査・検討に係る費用に加えて、利用するソフトウェアそのものも開発対象であるソフトウェアの一部分を構成するものとして、適切な価値を貨幣計数に反映させる必要があるのである。

それに対し、現行のソフトウェアに係る会計基準における再利用の取扱いには、不備があると言える。単に、費用あるいは資産としての処理の振分けをしているのみで、実態に即した再利用の会計処理を実質的には取扱っていないに等しい。換言すれば、再利用を識別するという着眼が欠落していると言える。しかし、実態に即して捉えるには、再利用の諸ケースを設定した上で会計上の取扱いを考える必要がある。このようにソフトウェア再利用に着目することは、ソフトウェア会計を精緻化する意義を有すると考える。なお、ソフトウェア再利用を主題的に取扱っているソフトウェア専門書は、ビジネスの成否や効率性は問題にしても、会計処理に言及しているものは筆者の知る限り皆無と言ってよい。又、会計基準に関する先行研究も同様に見当たらない。ただ1つ、内部管理的視点からの考察を後程取上げるが、それ以外に先行研究と言えるものはない。従って、会計処理に関しては筆者が独自に考察を進める他なく、本稿では再利用に係る会計処理の筆者案を提示することとしたい。

論文の構成は、2.でソフトウェア再利用を概観する。3.ソフトウェア再利用に係る会計処理では、まず(1)で現行会計基準における取扱いを取上げ、それに対し(2)でソフトウェア再利用に適合的な会計処理を提示する。具体的には、外部購入ソフトウェアと自社開発ソフトウェアの再利用に係る会計処理を、ケース毎に詳論する。4.おわりにでは、研究の構想の全貌における本稿の位置付けを確認する。

## 2. ソフトウェア再利用の概要

会計処理に先立って、その前提となるソフトウェア再利用を概観する。ソフトウェアの再利用とは、その名の通り、ソフトウェアの開発又は保守において既存のソフトウェアを(再)利用することである。開発実務では「流用」という用語が多く使われているが、主題的に取扱う文献では専ら「再利用」という用語が使われており、引用等との調和を勘案して本稿では主に再利用とする。まず、ソフトウェアの普及段階の比較的早い時期から、(a) 共通サブルーチン<sup>2</sup>が利用されていた。例えば、日数計算、西暦和暦変換、利息計算、利率計算等のサブルーチンのように、様々なところで利用するものを共通サブルーチンとするのである。これには、ソフトウェア開発において複数個所で同様の処理を行うために仕様を共通化し、1つのモジュール(サブルーチン)として開発する場合と、既に開発済みのサブ

ルーチンを新たな開発で利用し、当該ソフトウェアのコンポーネント<sup>3</sup>とする場合があるが、本稿で取上げるのは後者である。前者は再利用ではなく、共通化と言う。後者が再利用の言わば初期段階であるが、再利用はそのような局限的な利用の範囲を超えた意図的で広範囲な利用をする場合がある<sup>4</sup>。現象形態としては、異なるシステム環境への(b)移植 (conversion)、パッケージ・ソフトウェア等の(c)カスタマイズした利用等、旧(現行)システムをベースにした (d)再構築 (rebuilding) 等がある。移植は、基本的に機能変更は行わず、ハードウェアやOS等の異なるシステム環境に適合的な改造をして使用することである。再利用を利用のレベル(範囲)の観点から捉えると、上原三八ら[(2000)P.115~116]は、①システムの再利用、②プログラムの再利用、③プログラム部品の再利用、④仕様の再利用、と4つのレベルで捉えている。①~③はソフトウェアを直接再利用するもので、利用の範囲をシステムの全体ないしより局部的・限定的な範囲として捉えている。④はそれらとは異質で、ソフトウェアの直接的再利用ではなく、仕様の利用に留まるため、開発の実装作業は一般的な新規開発と変わらない。上記の現象形態と対応付けると、(a)は③、(b)は①(プログラムの移植は②)、(c)は①(開発予定のシステムの一部とするパッケージならば②)、(d)は④(部分的に②又は③を含む場合もあるが、一般的に再構築は異なるアーキテクチャで行うので、プログラムは限定的な利用に留まる)<sup>5</sup>、に相当すると言える。

次に、再利用の目的を取上げる。目的は再利用を行わない新規開発に比べ、工数並びに費用を軽減することである。これに関してカーマ・マックルーア[(1990)P.249~250]は、「ソフトウェア構成要素とソフトウェア開発経験を最大限に再利用することによって、次のような著しい効果が得られる」として、「1.ソフトウェア開発の合理化と単純化/2.ソフトウェアに対する信頼性の向上/3.ソフトウェア費用の削減」を挙げている。しかし、1.と2.は再利用するソフトウェアが模範的・良質的なものであることを前提として効果が得られうるので、予め再利用を意図して開発した又は結果的に成功裡に開発したソフトウェアのみが該当する。従って、そうした限定的な条件下でしか成立しないことは一般化できず、3.のみが一般的に該当する効果と言うべきだろう。

### 3. ソフトウェア再利用に係る会計処理

#### (1) ソフトウェアに係る会計基準における再利用の取扱い

ソフトウェアに係る会計基準<sup>6</sup>は、再利用をいかに取扱っているか。この設問への解答は、至極簡単である。後述するように、会計基準では確かに取扱ってはいるが、実質的には再利用の特性を看過しているのである。会計基準に関する問題点を大きく3つ挙げる。第1に、再利用の会計処理の全ての規定に共通していることであるが、費用あるいは資産に振分けているだけである。全て新規に開発した場合と、再利用した場合がいかに異なるか、それを会計上はいかに扱うか、あるいはその取扱いは必要ないか、そういった考慮が

なされていない。会計基準はカスタマイズする場合を挙示しているが、それを単に修正作業とみなしている。つまり、再利用の特性を看過しているのであるが、実際には修正とは別に再利用の作業が発生するのである。又、既存資産の利用と引換えにその開発費用を当該開発で負担する場合は、その費用の二重計上を避けるために振替処理を行うのが適切であるが、その考慮もなされていない。これらは、会計基準が見落としている重大な欠陥と言える。このように、会計基準は実態に即した再利用を実質的には何も取扱っていないに等しい。それでは再利用した開発において、内訳的に再利用を識別することは不可能であり、換言すれば、認識する必要性を看過している。他方、開発実務では、多少なりとも再利用が行われている実態があるが、これでは現行基準に則っては適切に会計処理をなし得ないのである。

第2に、想定されている事態が著しく偏っていて、抜けがあることである。会計基準では、実務指針14項及び38項(2)における購入パッケージを小幅にカスタマイズ(仕様変更)し、それをそのまま自社で使用する広義の再利用と、同15項及び39項における自社開発ソフト又はパッケージを大幅にカスタマイズし、開発するソフトの部品として利用する狭義の再利用のケースを挙げている<sup>7</sup>。しかし、カスタマイズの程度の違いによりケース分けすることにはいかなる意味があるのか、大いに疑問である<sup>8</sup>。更に、何故、大幅なカスタマイズの場合に限り部品としての利用を想定しているか、疑問である。何故ならば、小幅にカスタマイズしたパッケージを部品として利用する、あるいは大幅にカスタマイズをしてそのまま自社で使用する場合も十分に考えられるからである。又、購入ないし既存のパッケージの自社仕様のカスタマイズを当然のように想定しているが、そもそもパッケージに対してそれは可能なのか。会計基準が実態を踏まえているようには思えないので、ソフトウェア業界の実態を取上げる。パッケージは、第一義的にはライセンス契約により、第二義的には提供形態により、カスタマイズを許容する場合と、しない場合とがある。許容する場合も、カスタマイズは専ら提供元が担当し、一元的に管理することが一般的である。そのため、提供形態はオブジェクト提供<sup>9</sup>とすることが多い。もう1つの提供形態であるソース提供でも、契約による縛りは可能であるが、購入先がプログラムの閲覧や変更を容易にできることから、紳士協定的とならざるを得ない。従って、安全を考えれば、オブジェクト提供とすることが多いのである<sup>10</sup>。それ故、会計基準が想定している事態とは、契約的にカスタマイズを許容し、かつ提供形態がソース提供<sup>11</sup>の場合に限られる。その場合も、購入先が自主的にカスタマイズするとみなす規定となっているが、そうとは限らない。仕様の習熟度合いからすると、提供元へのカスタマイズの依頼が、特に初期的には大いにあり得ることである。つまり、会計基準はいずれも相当稀ないし局部的な事態を一般化していると言わざるを得ないのである。なお、購入先の自由なカスタマイズ又は別様の利用を許容する場合でも、第三者への提供を許諾することは殆どなく、あくまで購入先の自社利用の範囲内となる<sup>12</sup>。他

方、個別1品で所有権の移転を伴う購入の場合は、ソース提供であり、第三者への提供も任意であることは言うまでもない。

第3に、再利用のケース設定に重大な欠落がある。会計基準のソフトウェアの分類規準は制作目的であり、資産認識の対象のソフトウェアは販売用と自社利用用とに大別される。そのうち自社利用のみの再利用が想定されており、カスタマイズの外部委託は視野に入っているが、販売用ソフトの再利用は想定されていない。この会計処理を取入れる必要性は、主に2つの点から主張することができる。1つは、販売を目的としたパッケージ開発には特にコストダウンが強い要請としてあり、開発費用の軽減を可能とする再利用の必要性は自ずと高いのである。価格設定の基礎となる原価という意味では、余程の新奇性があるものでない限り、価格競争というビジネス要件からすれば、強い要請であると言える。2つには、技術的な要請として、シリーズ製品であるパッケージであれば、シリーズ内での下位互換性<sup>13</sup>が求められるのが一般的であり、それへの対応という点では旧製品の再利用が最も堅実な手段である。従って、販売用ソフトの再利用のケース設定が欠落していることは、重大な瑕疵と言わなければならない。このように、再利用に関する会計基準の規定には、大きな問題があると指摘することができる。会計基準で取扱われている再利用のケースと会計処理の一覧を示すと、以下の表1ようになる。

表1 ソフトウェアに係る会計基準における再利用のケースと会計処理一覧

制作目的	取得形態	再利用方法	開発形態	ケース設定の有無	再利用会計処理	資産計上の仕方		
市場販売目的				×	—	—		
自社利用	購入	独立	ノン・カスタマイズ	○ (社会的な再利用、個別企業主体ではソフトウェアの購入)	不要		購入ソフトウェアの資産計上	
			カスタマイズ	△ (小幅(付随的修正)のみ)	再利用作業費用 カスタマイズ費用	× ○	— 付随費用と同様に取得原価に含める	
		部品	ノン・カスタマイズ	×	—		—	
			カスタマイズ	△ (大幅のみ)	再利用作業費用 カスタマイズ費用	× ○	— 資産要件を充足すれば資産計上	
		自社開発	部品	ノン・カスタマイズ	×	—		—
				カスタマイズ	△ (大幅のみ)	再利用作業費用 カスタマイズ費用	× ○	— 資産要件を充足すれば資産計上
	再構築			△ (大幅なカスタマイズがこれを包含していれば)	再利用作業費用 カスタマイズ費用	× ○	— 資産要件を充足すれば資産計上	

(○:規定あり, △:一部規定あり, ×:規定なし。用語に関しては、表2と表3と同様であり、その説明は関連箇所を参照のこと。出典:筆者作成)

## (2) ソフトウェア再利用に適合的な会計処理

ここからは、ソフトウェア再利用に係る会計処理の筆者案を示すが、まず、会計基準との大きな相違に触れておく。第1に、再利用の対象であるソフトウェアの第一次的な分類規準は取得形態とすることである。会計基準は、制作目的別分類のうち市場販売目的ソフトウェアの再利用の規定が欠落している。こういうケースをも漏れなく捉えるためには、取得形態別分類が適していると考える。従って、外部購入ソフトウェアと自社開発ソフトウェアとに大別し、自社開発は更に狭義の再利用と広義の再利用の一種である再構築に分けて取上げる。第2に、第二次的な分類規準は再利用の方法とすることである。購入又は自社開発のいずれも、そのまま使用するか、あるいは更に開発で部品として利用するかは種々のケースが想定でき得る。その場合にも、そのまま使用する又は販売する場合はあくまで再利用に係る会計処理は不要となるが、更に再利用して開発する場合は自社開発のいずれかのケースに当てはめた会計処理をすることになる。

表2 外部購入ソフトウェア再利用会計処理一覧

No.	購入形態	再利用会計処理	資産計上の対象
a	ノン・カスタマイズ	不要	パッケージ・ソフト購入価格
b	カスタマイズ済み購入	不要	外部委託費 パッケージ・ソフト購入価格 (上記はセットでの購入価格の場合もある)
c	購入後、自社カスタマイズ	再利用作業費用 カスタマイズ費用	再利用作業・カスタマイズ費用 ソフト購入価格

(出典:筆者作成)

まず、外部購入ソフトウェアの再利用を取上げる。考えられるのは全3ケースである。(a)ノン・カスタマイズで購入する、(b)自社用にカスタマイズしてもらい購入する、(c)購入して自社でカスタマイズする、というケースである。具体的な会計処理と併せて、それらをまとめた表2の内容を説明する。会計処理の全体的な方針は、外部購入ソフトに対するカスタマイズ費用<sup>14</sup>は、カスタマイズを必要とする当該企業にとっての資産価値を増大させるものとして、ソフトウェア資産の取得原価に含めるものとする。しかも、会計基準のようにカスタマイズの程度により取扱いを区分することはせず、資産要件<sup>15</sup>を充足すれば資産認識し、資産の取得原価を構成することが妥当であるとする。

ケース(a)はソフトウェアを新規開発する替わりにほぼ同等の機能を有するパッケージを購入し、カスタマイズはしない場合である。これは購入先が購入価格をもってソフトウェアを資産計上するが、再利用の会計処理は必要ない。

ケース(b)は(a)と同じパッケージであるが、そのままでは機能要件等を充足しないので、カスタマイズを施してもらった上で購入する場合である。カスタマイズ及び再利用作

業は受託側が行うので、委託側の再利用の会計処理は不要となる。これらの費用に利益を上乗せした売価が外部委託費用であり、同じくパッケージの開発費用の売価がパッケージの購入価格となり、これらの合計額をソフトウェア資産に計上する。なお、カスタマイズ及び再利用の部分は、ケース(c)と区別するために外部委託費として捉えるが、初期段階では購入パッケージに対する追加購入費としても差し支えない。又、追加購入費ではなく、それらの費用を含めてセットでの購入となる場合もあり、その場合はパッケージ購入価格のみとなる。

ケース(c)は(a)(b)と類似のパッケージであるが、違いは購入先の当該企業が自由にカスタマイズできることである。この場合、費用総額はカスタマイズ及び再利用作業の費用に利益を含んだ売価としての上乗せがないので、(b)より安価になることが考えられるが、(c)を選択する主たるメリットはそこにはなく、パッケージ仕様に習熟し、自社で保守していけるようにすることである。これを重視する場合は、初期的には高価であっても自社でカスタマイズするであろう。費用重視であれば、売価としての上乗せと、仕様に習熟していく学習曲線との兼ね合いが判断規準となろう。再利用の会計処理は、カスタマイズ費用及び再利用作業費用となる。これらに、パッケージの購入価格を加えた金額を資産計上する。ここで、「カスタマイズ費用」とは購入ソフトの直接的な仕様変更部分に係る費用を意味し、実装作業は新規開発と同様である。「再利用作業費用」とは、購入ソフトの利用に係る作業の費用を意味する。それは、外部調達したコンポーネントの仕様に習熟していないため、詳細な仕様を理解するまでの調査・分析と、無修正部分とカスタマイズ部分との整合性や影響の有無に問題がないことの確認をする作業であり<sup>16</sup>、新規開発とは明らかに異なる再利用に特有のものである。

ところで、ケース毎には提供形態はオブジェクトあるいはソースでの提供の場合があり、更には汎用的で不特定ユーザを対象とするパッケージと汎用的ではなく特定ユーザを対象とする個別1品との場合分けが可能であるが、あくまで会計処理の内容及び取得費用並びに開発費用の測定は全く同様となるので細かくケース分けしていない。ケース(a)(b)は、広義には再利用であり、個別企業の枠を越えた社会的な再利用と言えるが、個別企業主体では単なるソフトウェアの購入である。

次に、自社開発ソフトウェアの再利用を取上げる。社内のソフトウェア資源を再利用する場合、その取扱いは幾通りもあり得る。ソフトウェア資源の再利用レベルやプロジェクトの編成状態、管理方式やカスタマイズの作業担当の違い等により取扱いが異なり、更はその費用の取扱いをケース分けし、整理して捉えることは可能である。しかし、会計上は当該プロジェクト(以下、「当該PJ」と略記)の会計処理として、再利用に係る費用負担に着目したケース設定で必要十分である。従って、ケース設定に関して、一定の合理性を有するのは全7ケースである。なお、機械的な組合せとしてはあり得るが、合理性を欠く



ものは割愛し、その理由は後述する。

表3 自社開発ソフトウェア再利用会計処理一覧

No.	開発形態	既存資産 開発費用負担	カスタマイズ費用 負担	再利用会計処理	資産計上の対象
d	ノン・ カスタマイズ	負担なし		再利用作業費用	再利用作業費用 新規開発費用
e		一部負担		再利用作業費用 既存資産開発費用 振替	再利用作業費用 新規開発費用 既存資産振替額
f	カスタマイズ	負担なし	負担なし	再利用作業費用	再利用作業費用 新規開発費用
g			一部負担	再利用作業費用 カスタマイズ費用	再利用作業・カスタマイズ費用 新規開発費用
h			全て負担	再利用作業費用 カスタマイズ費用	再利用作業・カスタマイズ費用 新規開発費用
i		一部負担	一部負担	再利用作業費用 カスタマイズ費用 既存資産開発費用 振替	再利用作業・カスタマイズ費用 新規開発費用 既存資産振替額
j			全て負担	再利用作業費用 カスタマイズ費用 既存資産開発費用 振替	再利用作業・カスタマイズ費用 新規開発費用 既存資産振替額
k	再構築	負担なし	全て負担	機能仕様の承継費用 再利用作業費用 旧システム除却 除却費用	機能仕様の承継費用 再利用作業費用 新規機能設計及び全工程に おける方式設計の開発費用

(出典:筆者作成)

個々のケースの説明に先立ち、ケース設定のポイントを挙げ、それをまとめた表3の内容を説明する。第1に、開発に利用するソフトウェア資源の開発形態を「ノン・カスタマイズ」と「カスタマイズ」に大分類したが、それらは再利用に係る性格及び費用の取扱いが異なるためである。又、購入ソフトと統一的な取扱いにする意味もある。再利用する際にそのまま再利用するか、仕様変更を行うかの区別である。

第2に、既存資産は開発に利用するために提供されるコンポーネントが資産要件を充足し、予め資産計上されていることを前提とする。既存資産の開発費用は、再利用する際に当該PJが負担することが考えられるが、その費用負担の有無を「負担なし」「一部負担」に区別している。なお、再利用に際し、この費用の全てを当該PJが負担することは考え難いので、割愛した。

第3に、開発形態がカスタマイズの場合には、既存資産の仕様変更の費用が発生するので、これを当該PJで負担するか否かにより、「負担なし」「一部負担」「全て負担」に区別している。但し、既存資産開発費用を一部負担しながら、まさに当該PJのためのカスタ

マイズ費用を負担しないケースは合理性を欠いているので割愛した。

第4に「再利用会計処理」は、再利用に係る会計処理の対象となる費用を挙げている。「再利用作業費用」は既存資源の利用に係る作業費用を意味し、「カスタマイズ費用」は既存資源の仕様変更に係る費用を意味する。又、「既存資産開発費用 振替」は、既存資産の開発費用を負担するケース(e)(i)(j)で、既存資産を開発及び管理するグループ又はプロジェクト(以下、「G又はPJ」と略記)から負担費用の振替処理を行うことを意味している。

第5に「資産計上の対象」は、再利用の会計処理と併せて、当該開発対象のソフトウェアが資産要件を充足することを前提として、資産計上の対象となる費用を挙げている。

第6に費用負担に関して「一部負担」とし、負担割合の数値等を提示していないが、様々あり得るのでここではそれ以上には踏み込まない。基本的には企業における決め事であり、当為的規を一義的に定めるのは適切ではない。これに関しては、別稿において測定の実例を示すこととし、それが具体的な数値を代入した測定値ということになる。

第7に、ソフトウェア資源の種別には「共通コンポーネント」と「個別コンポーネント」とがあり得る。前者は再利用に供する目的での開発又は管理をしている資源であり、後者は再利用を目的とした開発及び管理は行わないが、今回の当該PJの開発に再利用する資源である。開発実務ないし開発後の管理上は重要となるが、いずれも再利用時点での会計上の取扱いには大きく影響しないので、ケース分けに反映していない。ケース設定のポイントは以上である。

さて、具体的な会計処理は個々のケース毎に詳しく論ずるが、紙幅の都合により重複する説明は割愛し、主にケース毎の違いのみを説明する。前提条件は、新規開発の一部として既存資源を再利用することであり、当該既存資源は資産要件を充足し既に資産計上されていることである。ケース(d)は、コンポーネントは無償供与でかつ無修正で再利用する場合であり、再利用の作業費用のみが発生する。コンポーネントの無修正の再利用には、プログラム製造等の作業は一切不要であるが、コンポーネントの仕様の調査・分析と利用に問題がないことの確認テストが必要となる。但し、パッケージ等の総合的な利用ではなく、ごく局限的な利用なので、新規開発作業に比して、ごく僅かな作業を遂行するに留まる。それが再利用作業費用である。この費用は、カスタマイズや既存資産の費用負担の有無に関わらず、再利用する限り必ず発生し、以降の全ケースに共通している。そして、これに新規開発費用を加えた金額をソフトウェア資産に計上する。全てを新規開発する場合に比して、コンポーネントの開発費用分が軽減できることになる。なお、既に資産計上されているコンポーネントに対する会計処理は、不要である。

ケース(e)は、コンポーネントは無修正の再利用であるが、再利用と引換えにコンポーネントの開発費を一部負担する有償供与の場合である。よって、前ケースとの違いは、再利用作業費用に既存資産開発費用負担額が加わり、多少費用負担が増すことである。既存

資産開発費用は、既存コンポーネントを部品として利用する開発であり、言わば製造原価に占める直接材料費に相当する。それに対して、直接労務費に相当するのが再利用作業費用である。ここで、既存資産開発費用に特異な処理が発生する。既存コンポーネントの開発費の一部を当該PJに振替えると同時に、当該振替額だけ既計上のコンポーネント資産から減額する。既存資産との二重計上を避けるため振替を行うのである。既存資産が既に減価償却されている場合は、未償却残高からの振替額の減額となる。未償却残高が振替額未満の場合は、減額可能額は未償却残高までとして同額を減額し、当該未償却残高を当該PJに振替える。償却済みであれば、当初の利用価値を超えたプレミアム価値と捉え、既存資産及び当該PJへの振替処理は不要となる。これらの振替額と全くの未償却の振替額とでは差が生ずるが、その差額は、再利用して開発したソフトウェアの潜在的な価値が高いものと捉えることができるだろう。但し、既存資産開発費用を負担する場合にも振替を行わないことが考えられるが、その場合は費用を負担しないとみなすこととなる。しかし、より実態を反映させるためには、振替を行うことが妥当であろう。この振替処理は、既存資産開発費用を負担する以降の該当ケースに共通している。そして、新規開発費用にこれらを含めた開発費用総額を資産計上する。

この既存資産開発費用の振替を、井手吉成佳[(2010)P.181]は次のように取扱っている。最初に、ここでは引用との兼合いで「流用」という用語を用いるが、再利用と同義である。受注ソフトウェアの価格決定の考察の中で、ソフトAをソフトBに部分的に流用する場合、例えば、Aの開発原価は固有部分a(40万円)とBへの流用部分(60万円)、Bの開発原価は固有部分b(70万円)とAからの流用部分(60万円)から構成されるとした上で、「無形固定資産として計上されるソフトウェア開発の支出額はソフトウェアAの100万円とソフトウェアBの固有部分bの70万円との合計である170万円である」としている。この考察の問題点を2つの観点から指摘しよう。1つは、開発実態を数値上、忠実に表現しているかという点である。資産計上されるBには、開発において流用した実態が反映されていないことに大きな問題がある。Bの開発では、全てを新規に開発することをせずに、Aの一部を流用したのであるから、全てを新規に開発する場合に比して流用部分の開発費用を軽減できたことを反映させなければならない。開発費用の軽減が、まさに流用する目的であるからである。この結果が数値上も反映されなければ、開発実態に忠実であるとは言えない。他方、Aに関しても、資源を利用されたということが反映されておらず不適切である。このように、A並びにBのソフトウェアに関わる流用という経済的実態を、適切に反映し得る情報とする必要がある。もう1つは、ソフトウェアの資産情報の質が良好であるかという点である。Bの資産額に流用部分の原価を全く反映しないならば、流用部分の資産価値が評価されないことになり、ソフトウェア資産の情報有用性は乏しいと言わざるを得ない。何故ならば、資産Bに流用部分に係る潜在的な価値が存在していることを、情報

利用者は認識しようがないからである。仮に、Bの資産額に流用部分の原価を加えないならば、潜在的価値を知り得る他の方法により情報開示しなければ、Bの資産情報には不備があることになる。従って、これらの問題点を改善すべく、Aの流用部分の開発原価を何らかの負担割合に応じてBに負担させることが必要であり、振替処理が適切であることを主張したい。なお、流用特有の作業並びにカスタマイズのケースを考慮していないことも指摘しておく。

ケース(f)は、コンポーネントは無償供与であるが、仕様変更して再利用する場合である。但し、カスタマイズ費用は当該PJでは一切負担しない。従って、当該PJの再利用の会計処理は、ケース(d)と同様である。なお、発生したカスタマイズ費用は、既存資産を管理するG又はPJで既計上コンポーネントに加算処理を行う。ケース(g)は類似的であるが、前ケースとの違いは、カスタマイズ費用の一部を負担することである。従って、負担割合に応じたカスタマイズ費用の処理と、この費用の資産計上が加わる。他方、カスタマイズ費用から当該PJの負担額を除いた残額は、既存資産を管理するG又はPJで既計上コンポーネントに加算処理を行う。ケース(h)も類似的であるが、違いはカスタマイズ費用の全てを当該PJで負担することであり、金額が違ってくる。従って、既存資産を管理するG又はPJでの処理は不要となる。それ以外は同様である。

ケース(i)は、コンポーネントの仕様変更した再利用であるが、前ケースとの違いは、当該PJがカスタマイズ費用の一部を負担するだけでなく、既存資産の開発費用の一部も負担する有償供与の場合である。従って、負担割合に応じた既存資産開発費用振替額の振替処理と、この費用の資産計上を追加的に行う。他方、既存資産を管理するG又はPJでは既計上コンポーネントに対して、既存資産開発費用の振替額を減額し、併せてカスタマイズ費用の当該PJ負担残額を追加計上する。ケース(j)は類似的であるが、違いはカスタマイズ費用の全てを当該PJが負担することであり、金額が違ってくる。従って、既存資産を管理するG又はPJでのカスタマイズ費用の処理は、不要となる。それ以外は同様である。

ここで、既存資産開発費用を一部負担する場合の資産の取扱いを考えてみる。負担額を既存資産額の一定割合とする場合に、更に、減価償却が進んでいる場合には、当初資産額の一定割合とするか、未償却残高の一定割合とするかが問題となる。それにより実際の金額は異なる。しかし、いずれにしても決定的に優位とは言えない。当該PJで開発するソフトウェアに対する既存資産の再利用の貢献度合いは、容易ではないが、測定可能である。しかし、既存資産自体にとっての利用価値の測定は相当難しい。何故ならば、まず個別コンポーネントは再利用を予め企図してないので想定外の利用であり、ユーザーを含めた他PJで既に十分に利用価値を発揮しているとすれば、それを越えたプレミアムな利用価値と捉えるべきであり、その場合には無償供与でも差し支えない。次に、共通コンポーネントは再利用を予め企図しており、再利用頻度等を企画・開発計画等で明確にしていれば、そ

れに則った利用価値の測定が可能であるが、そうでない場合には、適当な決めをするしかないからである。故に、負担額の決定に際して規準となる金額は、いずれも便宜的であり、特段の優位性はない。従って、一貫して採用し簡便であれば、いずれであっても差し支えない。但し、減価償却との関連を考慮に入れれば、当該再利用時点での未償却残高を規準とした負担額の決定が、会計上は比較的優位であるように思う。既述の通り、減価償却手続を所与とした負担額の算出と振替処理を行うので、これらと連携させるという点では一貫性があるからである。又、関連して、仕様変更費用を一部負担する場合の取扱いにも、類似の問題が生じる。共通コンポーネントに関して、仕様変更が当該PJにとって不可欠であり、利用価値があるのは明白であるが、将来的な再利用価値やその頻度等を見積ることは中々難しい。従って、一部負担を具体的にいかにするかは、やはり便宜的たらざるを得ないであろう。

最後に、自社開発ソフトウェアの再構築を取上げる。これは、これまでの再利用とは性格が大きく異なる。第1に、再構築は広義の再利用の1種であるが、これ以外の再利用とは目的が異なる。再構築の主目的は、既存機能の継承と機能並びに技術の刷新という両面的なものである。よって、前述の工数並びに費用の軽減という狭義の再利用の主目的は、付随的ないし副次的なものに留まる。単なる継承ならば保守を継続するか、アーキテクチャに問題があるのみならばリエンジニアリング方式の再構築をすればよい。それを敢えて、多額の費用を投じ、スクラップ・アンド・ビルド方式で行うのは、相当数を継承し、然りながら刷新したい要件も少なからずあるからである。費用だけを付度すれば、現行システムに拘らず、全く新規に開発したほうが安上がりなことすらあり得る。それにも関わらず、長年蓄積してきた業務ノウハウ等を継承したいからこそ、再構築とするのである<sup>17</sup>。技術論に終始しがちな再利用の諸書が揃って看過している重要な事項である。第2に、再構築は再利用の範囲がどれ程であろうと、開発完了に伴い、旧システムを廃棄することである。これ以外の再利用には、再利用資源と再利用した新ソフトウェアの間にそのような連動的な関係はない。それ故、これに特有の会計上の取扱いが必要となる。それは、必ず発生する廃棄に伴う一連の定型的な処理であるので、それ以外の再利用のようなケース分けは不要となる。第3に、再構築で独立したPJを編成するため当該PJ自らが担当しないことはあり得ないので、複数のケースを設定する必要はない。

具体的な会計処理はケース(k)で、スクラップ・アンド・ビルド方式により再構築する場合である。既存の機能要件の大半を承継するので、機能設計に対する追加・変更はそれ程多くないのに対し、技術要件(方式設計)はその殆どを新たに設計し直すので変更は広範囲に亘る。よって、再利用の会計処理は、機能仕様の承継費用及びその再利用作業費用である。前述のように、製造原価に占める直接材料費が機能仕様の承継費用であり、直接労務費が再利用作業費用に相当する。これらに加えて、新規機能設計及び全工程における方

式設計の開発費用がシステム全体の費用を構成し、併せて資産計上する。又、旧システムは、新システムの稼働と同時もしくは一定期間並行運用しても遠からず廃棄するので、未償却残高のある資産はそれを全額除却し、償却済みであれば不要となる。そして、廃棄に係る一時費用を除却費用として処理する。この処理は一般的にはシステム運用Gで行い、当該PJで行う処理ではないが、再構築に特有の処理として挙示しておく。これで、再構築に係る会計処理の道筋は示せたと思う。

#### 4. おわりに

ソフトウェア再利用に適合的な会計処理を、現行会計基準における不備に対する指摘を踏まえて、諸ケースを詳論することにより十分に説明し得たと考える。それが財務報告にとって有する意義をまとめておきたい。ソフトウェア資産の取得原価は、その金額の多寡だけでは、当該ソフトウェアがいかなるものか、実のところ明瞭ではない。ソフトウェア開発の測定に関する考察を進めているが、開示情報としてソフトウェア測定項目の開示の必要性を痛感している。それに対する本稿の意義は、再利用という開発方法の内容と併せて、その精細な会計上の取扱いを示したことにある。現行会計基準には不備があり、筆者の案が妥当だと思うが、ごく僅かに振替処理等により複雑になるとは言えよう。しかし、実態に即した会計処理を可能にするものであるから、それはごく軽微な負担に過ぎないと考える<sup>18</sup>。

今回は紙幅の都合により割愛した、ソフトウェア再利用に係る費用の測定については、本稿における諸ケースに即して測定方法を含めて示すことにより、再利用の包括的な会計上の取扱いを示し得ると考えており、続稿にて詳論する。又、再利用とも関連して、ソフトウェア開発に密接に関係している開発ツールの利用の会計上の取扱いに関しても考察するつもりである。更に、開発局面だけでなく、保守や運用、廃棄の取扱いも重要であり、それらの実態に即して一貫した捉え方を可能とするソフトウェア測定をその基礎に位置付け、ソフトウェア・ライフサイクルを包括したソフトウェア会計の考察を進めていきたいと考えている。

- 1 ソフトウェア工学において、「精度」は精細さの度合い、「確度」は確実さの度合いとされている。本稿ではこの両者の意味で「精確」を使用し、「正確」とは明確に区別している。
- 2 他に、共通マクロ、共通関数等の呼称があり、各々分野等の違いがあるが、本稿ではそれらの総称として共通サブルーチンを使う。
- 3 この文脈では狭義の部品を意味しているが、コンポーネントは広義にはプログラム(群)ないし機能(群)をも意味する。
- 4 ウィル・トレイツ(2001) P.134は、そのことを「再利用の問題を、単にサブルーチンやクラスの開発と取り違えて片付けてしまわないこと」と言っている。但し、再利用を包括的に捉えるには、ミニマムな再利用として、それも含めなければならない。
- 5 再構築を技術的に大別すると、リエンジニアリング方式とスクラップ・アンド・ビルド方式がある[上原三八(2000) P.9, P.114]。前者は、システムの再利用とも言え、機能は変更せずに、主に技術的な作り直しであるが、実施例はそれ程多くない。費用対効果が乏しいからである。本文では後者を念頭に置いている。既存機能の相当数を継承するが、機能追加や変更をアーキテクチャの変更と併せて実現するからこそ、費用対効果が生まれるのであり、大半はこちらを採用するとみて大過ないだろう。
- 6 企業会計審議会(1998)と日本公認会計士協会(1999)のソフトウェアに係る箇所の総称である。以下、「会計基準」と略記する。
- 7 日本公認会計士協会(1999)の14項では「外部から購入したソフトウェアについて、(中略)自社の仕様に合わせるために行う付随的な修正作業等」の費用は、当該ソフトウェアの取得価額に含める。同38項(2)の補足説明では、「完成したソフトウェアを購入する場合でも、例えば、(中略)自社の仕様に合わせて画面や帳票などを修正する場合などがある。これらの作業は、自社で行う場合と外部委託する場合がある。(中略)外部から購入したパッケージソフトウェアに対して(中略)自社の仕様に合わせるための付随的な修正作業等の費用は、購入ソフトウェアを使用するために不可欠な費用であり、有形固定資産の取得に要する付随費用と同様に」処理するとされる。同15項では「自社で過去に制作したソフトウェア又は市場で販売されているパッケージソフトウェアの仕様を大幅に変更して、自社のニーズに合わせた新しいソフトウェアを制作する」ための費用は、資産要件を満たせば資産計上し、それ以外は費用処理とする。これに対して同39項で、「既存のパッケージソフトウェアの仕様を変更して自社の要望に合わせた新しいソフトウェアを制作する場合は、完成品のソフトウェアを購入したとは考えられず、むしろパッケージソフトウェアを部品として利用していると考えの方が適切である」ことから、パッケージの取得費用は、仕様変更後の新ソフトウェアの利用により資産要件を満たすときを除き、費用処理が適当とされる。
- 8 この会計処理は、市場販売目的のソフトウェアにおける「著しくない改良」と「著しい改良」(実務指針9項及び33項)と、同列の論法である。それがソフトウェア開発の大半を研究開発とみなす根幹の規定からの派生系であることを筆者は承知しているが、これらが致命的な誤認であることは、ソフトウェア開発の大半が研究開発でないことを論証しなければ決着が付けられず、その主題的な論稿を準備しているので、ここでは立入らない。
- 9 この「オブジェクト」は、プログラミング言語で記述するソース・プログラムに対して、2進コードの機械語に変換された、ロード・モジュールを含めた慣用的な総称である。
- 10 高難度の技術であるが、機械語からプログラミング言語へ逆変換するリバース・エンジニアリングにより、オブジェクトからソースを復元することは可能である。しかし、第3世代以降の言語(高級言語)の使用が普及したため、古くからある技法であるがこれを駆使できる技術者は少なく、今日的には難しいものとなっている。なお、技術的に可能でも、オブジェクト提供の内部仕様は非公開の機密事項であるか

ら、その利用は契約に抵触する。

- 11 購入先の自由なカスタマイズを許容するパッケージ販売を何故行うかという、売切りとし、手離れをよくしたいからである。パッケージのカスタマイズを許容する場合で専ら提供元が対応・管理することは事業的には中々難しい。例えば10顧客(企業)に販売し、各々異なったカスタマイズを行った場合、マスターは1つでも、それから派生した単なる複製ではないクローン的なサブ・マスターが10存在することになるので、制度改訂等があり、且つカスタマイズ部分に関係するときは、マスターだけでなく、各サブ・マスターに変更を施さなければならない。こうした保守を継続的に採算ベースで行なっていくことは非常に難しい。その事情を知りながら、敢えて本来的なパッケージ事業を行なうか、さもなければカスタマイズを許容する売切り制を選択するのである。
- 12 ここまでの議論は営利目的のパッケージに関してであり、オープン・ソース(フリー・ソフトウェア)に関しては有償・無償に限らず、少し異なった議論が必要であるが、会計基準の制定年次からしてオープン・ソースを視野に入れていたとは思えないので、今回は立ち入らない。
- 13 ソフトウェアが相互に利用可能である互換性のうち、旧製品の仕様をそのまま継続的に利用できることである。
- 14 本稿で用いる「費用」という用語は全て、発生現象を意味し、costと同義である。
- 15 会計基準におけるソフトウェア開発を研究開発とみなす規定との関連で、資産要件に関しても問題視しているが(注8参照)、本稿では現行会計基準の資産要件の規定を踏襲した議論を行っている。
- 16 保守と類似しているが、それとの違いは、他のコンポーネントとの親和性並びに将来性(保守性、拡張性)の良否の確認に重点が置かれることである。
- 17 ここには、「社内政治」の問題がしばしば介入する。例えば、守旧派と刷新派の抗争といった、ソフトウェアの仕様や方式に託けて代理戦争を行う様相であるが、主題からは外れるので立ち入らない。
- 18 なお、この考察は細則主義に立脚したものとみなされるかもしれないが、ソフトウェア再利用の特性を適切に捉えるための会計処理の考察であり、原則主義・細則主義のいずれの立場に立脚しようとも、こうした考察を踏まえる必要があると考える。

#### <参考文献一覧>

- Gamma, Erich, Helm, Richard, Johnson, Ralph, Vlissides, John(1995), *Design Patterns Elements of Reusable Object-Oriented Software*, Boston, Addison Wesley Longman(エリック・ガンマ, リチャード・ヘルム, ラルフ・ジョンソン, ジョン・ブリシディース(本位田真一, 吉田和樹監訳)(1999), 『オブジェクト指向における再利用のためのデザインパターン 改訂版』ソフトバンク クリエイティブ).
- Greenfield, Jack, Short, Keith(2004), *Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools*, Indiana, Wiley Publishing (ジャック・グリーンフィールド, キース・ショート (野村一行監訳) (2005), 『ソフトウェアファクトリー パターン, モデル, フレームワーク, ツールによるアプリケーションの組み立て』日経BPソフトプレス).
- 井手吉成佳(2010), 「ソフトウェア原価計算における機能的規模測定法の適用可能性」広島大学大学院博士論文
- Jacobson, Ivar, Griss, Martin, Jonsson, Patric(1997), *Software Reuse: Architecture, Process and Organization for Business Success*, Boston, Addison-Wesley Pub(イヴァー・ヤコブソン, マーティン・グリッス, パトリック・ジョンソン(杉本宣男, 吉田幸彦, 落合修, 田中正仁監訳) (1999), 『ソフトウェア再利用ガイドブック アーキテクチャー, プロセス, 組織の変革による再利用ビジネス成功への道』トッパン).
- 企業会計審議会(1998), 「研究開発費等に係る会計基準」



- McClure, Carma(1989), *CASE is Software Automation*, New Jersey, Prentice-Hall(カーマ・マックルーア(三井銀総合研究所訳)(1990), 『CASE』日経BP社).
- 日本公認会計士協会(1999), 「会計制度委員会報告第12号 研究開発費及びソフトウェアの会計処理に関する実務指針」
- 櫻井通晴 (2006(2001初版)), 『ソフトウェア管理会計—IT戦略マネジメントの構築— 第2版』白桃書房
- Tracz, Will(1995), *Confessions of a Used Program Salesman: Institutionalizing Software Reuse*, Boston, Addison-Wesley Pub(ウイル・トレイツ(畑崎隆雄・林雅弘・鈴木博之訳)(2001), 『ソフトウェア再利用の神話』ピアソン・エデュケーション).
- 上原三八, Wei-Tek Tsai, 佐野隆, 馬場一弥(2000), 『保守とリエンジニアリング』共立出版