

ソフトウェア開発委託契約紛争事例の研究(1)

内 布 光

はじめに

IT (Information Technology) 社会の到来とともにインターネットの世界的普及とあいまって、企業はもとより一般家庭においてもコンピュータ（主としてパソコン）は、日常生活に欠かせない必需品となっている。

本来、パソコン（パーソナル・コンピュータ）は、ソフトウェア¹⁾ なくして使用することができないものであるが、実際には、殆どの人がこれらのソフトウェアを意識することなくパソコンを使用しているのが通常である²⁾。それは、普通一般に使用される幾つかのソフトウェアが、あらかじめパソコンに搭載（プレインストール）されているからである。

また、パソコン用のソフトウェアは、多種多様の用途に応じて開発されたものが大量生産され、パッケージソフトとして安価で市場に提供されているので、それを購入してパソコンを使用すれば、ある程度の仕事（コンピュータ処理）をすることができる。

一方、企業や学校など（以下「企業等」という。）においては、コンピュータを使って大量の様々な情報データを共有しながら各種業務を処理しているが、これらの各種業務を処理するためのソフトウェアの全てを、パッケージソフト市場から調達することができない。なぜなら、企業等はそれぞれ独自の内容・方法で業務処理を行っているので、これら全ての企業等

の業務処理内容・方法に適合するようなパッケージソフトを開発することは事実上困難であり、結果的には、企業等の独自の業務処理に適合するパッケージソフトは、市場で流通していないからである。以上のことから、企業等は、どうしても自己の業務処理に適合するソフトウェアを開発（コンピュータ・プログラムを作成）しなければならないことになる。この場合、ソフトウェアハウスなどの専門事業者に委託して開発を行うのが通常であるが、従来から、この委託取引を巡り様々な法的紛争が生じている。

そこで、このようなソフトウェア開発委託取引において、どのような法的紛争が生じているのかについて研究することにした。この研究成果を2回に分けてとりまとめ、第1回の本稿では、ソフトウェア開発委託取引の実態を明らかにし、この取引における法的紛争について裁判事例を中心に調査したうえで紛争原因の究明までを行なうことにする。

1. ソフトウェア開発の概要

ソフトウェア開発委託取引において、なぜ法的紛争が生じるのだろうか。この問題を究明するためには、ソフトウェアとはどのようなもので、なぜソフトウェアを開発する必要があるかなどを知っておかなければならない。そこで、ソフトウェア開発の概要について言及しておく。

(1) ソフトウェアとは

ソフトウェアはコンピュータにとって不可欠のものであり、通常、これがないとコンピュータを動かすことができない。

「ソフトウェア」という言葉は、もともとハードウェア（コンピュータ、プリンターなどの機器）に対する言葉（対語）であるが、広義には、プログラム及びマニュアルその他の関連資料を含む総称として使われている。これに対して狭義には、コンピュータで文書作成など一定の仕事を行なう

ために使用されるコンピュータ・プログラムと同じ意味で使われる場合が多い。また、コンピュータ・プログラムは、通常、単に「プログラム」と呼ばれることが多い。そこで本稿でも、「ソフトウェア」の言葉を「プログラム」と同義で使用する。

なお、プログラムは、ソフトウェア技術者³⁾がCOBOLやC言語などのプログラム言語を用いて作成されているが、著作権法では「電子計算機を機能させて一定の結果を得ることができるようにこれに対する指令を組み合わせたものとして表現したものをいう」(第2条1項10の2号)と定義されており、著作権法により著作物としての保護を受ける⁴⁾。そして、このようにして人間(ソフトウェア技術者)によって作成されたプログラム(これを「ソース・プログラム」という)は、そのままの形ではコンピュータは読むことができないので、コンピュータが読める形に変換(コンパイル等)しなければならない。このようにコンピュータが読めるバイナリコードの形態に変換されたプログラムは、オブジェクト・プログラムと呼ばれているが、通常、パッケージソフトは、このオブジェクト・プログラムの形態で流通に付されている。

(2) 企業等におけるソフトウェア使用実態

ところで、一定規模以上の企業等は、通常、コンピュータを用いて経営・事業運営に係る生産・在庫・販売・経理・人事管理等の基幹業務を処理している。このような企業等で使用されているコンピュータは「情報システム」と呼ばれ、経営・事業運営に係る生産・在庫・販売・経理・人事管理等の基幹業務の処理を行なうソフトウェアは、「基幹システム」と呼ばれている。そして、情報システムは、多数のハードウェア(コンピュータ、プリンター、通信機器など)と多種多様のソフトウェア(プログラム群)を体系的に組合せて組織されており、現在、「クライアント・サーバー・システム(略称「CSS」)」と呼ばれるWeb対応(ネットワーク利用)

の分散処理型の情報システムが主流となっている。この Web 対応・分散処理型情報システム (CSS) を構成するコンピュータを区分すると、大きくは、ペータベース・サーバーや Web サーバーなどの数種類のサーバーと多数のクライアントとに分かれている。通常、サーバーには、その役割に応じて大容量・高速処理用などの高性能コンピュータが使われ、クライアントの殆どはパソコンが使われている。もちろん、これらのサーバーもクライアントも、その機能・性能等に差があるものの全てがコンピュータであり、これらのサーバーとクライアントが相互に有機的に通信回線網 (LAN 等のネットワーク) によって接続されている。

従って、サーバーもクライアントもコンピュータである以上、全てのサーバーには UNIX や Linux などの OS (オペレーティング・システム) が、また、全てのクライアントには Windows などの OS が必ず搭載されている。これに加えて、データベース・サーバーには DBMS⁵⁾ が、Web (www) サーバーには Apache やマイクロソフト社の IIS などの Web サーバソフトウェア (情報送信機能を持ったソフトウェア) が搭載されている。つまり、これらのソフトウェアがそれぞれのコンピュータで使われることによって、コンピュータは所定の機能を発揮できることになる。

なお、ここに例示した Linux や Apache は無償で提供されているため、いわゆる「フリーソフト」⁶⁾ に属するが、同時にソースコードも開示され誰もがこれを自由に改良することもできるため、特に、このようなフリーソフトは「オープンソース」⁷⁾ と呼ばれている。

また、DBMS には、ORACLE 社やマイクロソフト社などが有償提供しているライセンスソフト⁸⁾ のほかに、PostgreSQL などのフリーソフトもある。

本稿では、有償・無償の別を問わず、主として不特定多数のパソコン使用者向けに開発され市場で流通しているソフトウェアを総称してパッケージソフトという。

(3) ソフトウェア開発の必要性

今日ではパッケージソフトと呼ばれるソフトウェアが豊富に市場で流通しているのに、なぜ、企業等はソフトウェアを新規に開発しなければならないのであろうか。

企業等が基幹業務処理に適合する情報システムを構築しようとする場合、コンピュータなどのハードウェアの調達は比較的容易であり、ハードウェアとはほぼ一体となっている基本ソフトすなわち OS（オペレーティング・システム）の調達についても同様である。

また、サーバー用の DBMS やクライアント用のブラウザなどのソフトウェアも、一般的にパッケージソフトとして広く市場で流通しているものが多いので、これらの調達もあまり問題とならない。

しかし、企業等の基幹システム用のソフトウェア（アプリケーションソフト）については、ぴったりと適合するパッケージソフトが市場に流通していないことが多いので、当該ソフトウェアを開発せざるを得なくなる。ここに、ソフトウェア開発の必要性が生じることになる。そして、一般的にソフトウェア開発技術を有しない企業等（以下「ユーザー」という。）は、この開発業務を外部のソフトウェアハウス等の専門事業者（以下「ベンダー」という）に委託するのである。

(4) ソフトウェア開発ニーズの変遷

ソフトウェアを洋服に例えると、パッケージソフトはレディメイドすなわち既製服である。これに対し、ベンダーに委託して開発したソフトウェア（注文服のようにオーダーメイドのソフトウェア）は一般に「カスタムソフト」と呼ばれている。

一昔前の汎用コンピュータ全盛時代には、コンピュータ自体が極めて高額であったため、一台の汎用コンピュータをエアコンの効いた専用のコンピュータルームに設置して、専門の技術者がこのコンピュータを操作しな

が大量データを一括処理しているという光景を目にすることができた。そして当時は、この汎用コンピュータの OS は、当該コンピュータのメーカーが当該コンピュータ専用に独自に開発してユーザーに提供するのが通常であった。そこで、このような汎用コンピュータで使われるソフトウェアを見ると、DBMS などの一部のミドルウェア⁹⁾を除き、業務処理用ソフトウェア（アプリケーションソフト）は殆ど市場に流通していなかった。つまり、ミドルウェアやアプリケーションソフトは、基本ソフトである OS に適合するように開発されなければならないから、不特定多数のユーザーの業務処理に共通して使えるようなパッケージソフトを開発するのは極めて困難であり、この結果、ソフトウェア市場が形成できなかったものといえる。そこで、各ユーザーは、自らの業務処理用のアプリケーションソフト（カスタムソフト）だけは開発する必要があったのである。

これに対して現在では、企業等の基幹業務処理のため情報システムを見ても、CSS が主流となっており、各人が自席でパソコンを一人一台使用しながら仕事を行なうのが当たり前の時代となっている。つまり、大型の汎用コンピュータによる処理形態は影を潜め、クライアントやサーバーなどの小型コンピュータが利用され、かつ、クライアントである各人のパソコンでは、メール、インターネット検索・閲覧、文書作成、表計算など、いわばパソコンを通常使用する際に誰もが行なう画一的な処理も行っている。また、これらのパソコンで使用されるソフトウェアを見ると、基本ソフトである OS（Windows など）やメール、インターネット検索・閲覧、文書作成、表計算などのアプリケーションソフトは、いわば事実上の世界標準（デ・ファクト・スタンダード）となっているものが使用されている。このようにパソコン用のソフトウェアのうち、誰もが通常行なう共通業務の画一的な処理を実現するソフトウェアは、パッケージソフトとして開発しやすいので、現実も大量に市場で流通している（市販されている）のである。

しかし、これら市販のパッケージソフトを、各ユーザーの基幹業務の処理用にそのまま使用することができない。なぜなら、市販のパッケージソフトは、上記の通り、誰もが行う共通業務について画一的処理ができるように開発されたものであり、個々のユーザー固有の業務処理の特性（独自性）を一切考慮していないソフトウェアであるからである。つまり、各ユーザーの業務処理に完全に適合する市販のパッケージソフトは殆ど存在しないことになる。しかも、市販のパッケージソフトは、一般にライセンス契約で一切の変更を禁止し、かつ、オブジェクト・プログラム形式で提供されているので、各ユーザーが自己の業務処理に合わせて修正（プログラムの改変）することは不可能である。

例外的に、近年はERPパッケージ¹⁰⁾と呼ばれるパッケージソフトを、かなり多くのユーザーが基幹システムとして利用しているが、このERPを利用する場合でも、当該ユーザーは、自らの業務処理のやり方に合わせてこれを修正する作業（これもソフトウェア開発作業の一種）が不可欠となっている。

このようにパッケージソフトが豊富に出揃っている今日でも、ユーザーの業務処理用ソフトウェアについては、当該ユーザーの業務処理の特性に適合する形でソフトウェアを開発しなければならないのである。従って、現実にもソフトウェア開発のニーズが根強く残っているのである。

（5）開発されるソフトウェアの特徴

パソコン用パッケージソフトと同様、今日の開発対象となるソフトウェアも従前のものとはかなり変わってきている。すなわち、今日のソフトウェアには、画面を見ながらクリックだけで簡単に操作できるようなもの、すなわち誰もがIT専門技術なしに容易に使えるようなもの（操作性が良いもの）が要求される。このような操作性が良いソフトウェアのことを、「ユーザーフレンドリーである」とか「使い勝手がよい」などと表現してい

る。

そして、このような特徴を持つソフトウェアを開発するためには、従前のプログラム設計だけでなく、当該プログラムを使用する際に映し出されるディスプレイの画面設計も不可欠となる。そうすると、必然的に画面上に映し出すアイコン・イラスト写真等のコンテンツや情報データ（これらは「デジタルコンテンツ」と呼ばれている）などを取り込む必要が出てくる。

従来からソフトウェア開発においては、第三者ソフト（当事者以外の第三者が開発し、その著作権を有するソフトウェアで、パッケージソフトはこれに当たる）も盛んに利用されてきている。今日のソフトウェア開発においては、このような第三者ソフトのみならず、上記のような第三者が著作権を有する各種デジタルコンテンツも利用せざるを得なくなる場合が多いと思われる。

(6) ソフトウェア開発工程と開発手法の変化

Web 対応・分散処理型情報システムにおいて、そこで使われるソフトウェアの開発はどのような手法で行なわれているのであろうか。

パソコン用のパッケージソフトはもとより、各種デジタルコンテンツも大量に市場（主としてインターネット上の市場）で流通しているので、ソフトウェア開発においても、これらのソフトウェアやデジタルコンテンツを、利用できる場所には利用しようという傾向が強くなってきている。これは、第三者ソフトやデジタルコンテンツが豊富に出回っており、それらと同等なものを自ら新規に作るよりも安価に入手できるばかりか、目的とするソフトウェア開発作業のスピードアップが図れるという大きなメリットがあるからである。この環境の変化によって、ソフトウェアの開発手法も、従前の汎用コンピュータ用ソフトウェア開発当時のものに比べて激変している。

図1 ソフトウェア開発手法の比較

名称	ウォーターフォールモデル	スパイラルモデル	プロトタイピングモデル
概念図			
対象	大規模／中規模システム	全体を区分可能なシステム	- 小規模システム - アプリケーションソフト
特徴	- システムの開発工程をいくつかに分け、その順に各工程での作業内容をチェックしたり、評価したりして開発を進める開発方法 - 段階ごとに仕様書に従い作業が行えるため、管理しやすい - トップダウンの形で開発が進むため、後戻りして前の段階の決定を変更しにくい	- ウォーターフォールモデルとプロトタイピングモデルの両方の手法を取り入れた開発方法 - 比較的密接な関係にある機能ごとに開発を区分し、区分ごとに設計→プログラミング→テストを行いながら全体を完成させていく開発方法	- システム開発工程において、早い段階で試作品（プロトタイプ）を作成し、利用者の要求を確認できるようにした開発方法 - 組込型プロトタイピング；要求仕様の評価に使用したプロトタイプを修正・変更して本番用のシステムを作成する - 破棄型プロトタイピング；プロトタイプをシステムの要求仕様の評価目的のみに使用し、評価後は廃棄され、システムはウォーターフォールモデルで改めて開発される

出典；JISA 法的問題委員会契約部会編「新しいソフトウェア開発委託取引の契約と実務」（商事法務、2003）15 頁

ここで、一般的なソフトウェア開発作業の流れ（いわゆる「開発工程」）を見てみると、概ね「要求分析（システム調査分析、外部仕様の決定など）」、「設計（基本設計、機能設計など）」、「製造（プログラム作成、テストなど）」の三段階に区分されることが多い¹¹⁾。

そして、ソフトウェア開発手法の種類（タイプ）を見ると、図1に示す通り、ウォーターフォールモデル、スパイラルモデル及びプロトタイプモデルの三つのタイプがあるとされている。

まず、「ウォーターフォールモデル」は、開発工程に従って要求分析、設計そして製造の各工程をウォーターフォールのように上流から下流への流れに沿って開発する方法であり、従前の汎用コンピュータ用ソフトウェア開発においては、この開発手法が主として採用されていた。そして、この手法は汎用コンピュータ用の大規模ソフトウェア開発、すなわち膨大なプログラミングなどの作業を多数のプログラマー（主としてプログラミング作業を行うソフトウェア技術者）に分担させて、いわば人海戦術的に開発する場合に適しているが、他方では、開発期間は長期化を要し、かつバックログ（後戻り作業）等の小回りが利かないという大きな欠点がある。

これに対し、今日のソフトウェア開発では、前述の通り、使い勝手のよいソフトウェアを短期間で開発することが多いので、明らかにウォーターフォールモデルの作業手法は適せず、「プロトタイプモデル」ないしは「スパイラルモデル」と呼ばれる開発手法が主として採られている¹²⁾。つまり、今日ではユーザーの基幹システム（CSS方式が主流）の開発やインターネットを利用した電子商取引システム¹³⁾の開発の殆どは、プロトタイプモデルないしはスパイラルモデルの開発手法が採られているのである¹⁴⁾。

(7) ソフトウェア開発の委託形態の変化

このように開発対象ソフトウェアの内容や開発手法などが従来とは大きく変化したことに伴い、開発に要求される技術内容や委託形態も従前のも

のとは大きく変わってきている。

まず、ソフトウェア開発における技術内容を見ると、今日のユーザーの情報システムは、インターネットやイントラネットなどのネットワークを利用した Web 対応・分散処理型の CSS となっている。このような情報システムを構築するためには、従前のようにプログラミング技術を中心としたソフトウェア開発技術だけでは実現（構築）できないことは明らかである。つまり、情報システムの構築においては、ハードウェア（コンピュータ）に関する技術からデータベース作成技術やネットワーク技術までの様々な IT 関連技術を要求される作業が数多く発生するのである。すなわち、ソフトウェア開発技術（システム分析・設計からプログラム作成・テストに至るまでに必要とされる技術）はもとより、各種デジタルコンテンツやデータベースなどの作成技術、ネットワーク（通信）技術など幅広い IT 関連技術（いわば IT 関連する総合力）が要求されることになる。

このことは、一昔前と今日の住宅建築請負の委託取引形態の変化に似ている。つまり、従前の木造の住宅の建築においては、棟梁を中心に大工や左官などの職人が在来工法によって建てる形態が主流であった。しかし、今日では建材の多様化や新建築工法の導入などによって、大工や左官などの職人だけの手には負えなくなっている。この典型が、マンションなどの大規模住宅建築請負の場合であって、総合建築事業者が一括して請け負い、様々な専門事業者などを下請けに使いながら建築するという形態で行なわれている。

このような住宅建築請負の委託の場合と同様に、今日の情報システムの構築（ソフトウェア開発）においては、プログラム作成はもとより、各種デジタルコンテンツ作成、データベース作成、ネットワーク（通信網）構築など、それぞれ異なる分野の IT 専門技術を要する作業が生じる。ユーザーがソフトウェア開発を委託しようとする際、これらの専門技術作業ごとに、当該専門事業者に個別に委託したのでは各専門技術作業を整合させ

ながら行なわなければならないので、目標通りの日程で開発を完了させることは難しい。そこでユーザーは、IT 関連総合技術を有する（又は、これらの技術の取りまとめ能力を有する）ベンダーを選定し、当該ベンダーに一括して委託したほうが安心であり、かつ、開発期間も短縮できるばかりか、価格や品質などの各面においてもメリットを期待できることになる。そうすると、委託形態も従前のものとは大きく変わることになる。

2. ソフトウェア開発委託取引の形態とその問題点

ソフトウェア開発委託取引は、ユーザーが使用するソフトウェア（カスタムソフトと呼ばれるアプリケーションソフト）を、ベンダーに委託して新規開発することを目的とした取引である。

以下、この取引はどのような形態で行われているか、また、この取引では何が問題となりやすいかなどについて検討する。

(1) ソフトウェア開発委託取引形態と契約類型

ユーザーがベンダーに対するソフトウェア開発委託取引の委託形態を見ると、通常、請負型、委任型及び派遣型の三つに分けられている¹⁵⁾。つまり、それらの取引契約の目的・内容等に応じて、請負型は請負契約、委任型は準委任契約、そして派遣型は労働者派遣法¹⁶⁾に基づく労働者派遣契約というように契約類型を当てはめているのである。例えば、大規模ソフトウェア開発の場合、通常、ウォーターフォールモデルの開発手法が採られるので、この手法により委託する場合は、ソフトウェアの開発工程における各工程の作業（要求分析、設計、製造）毎に委託業務を分けて、要求分析は準委任契約で、設計は準委任契約又は請負契約で、製造は請負契約又は派遣契約で行なうことも可能である¹⁷⁾。

ところが、今日の多くのソフトウェア開発は、前述の通り、プロトタイ

プモデルないしはスパイラルモデルの開発手法で行われているが、これらの開発手法での作業はウォーターフォールモデルのように各開発工程の作業を切り離せず一連の作業として行う必要がある。そこで、ユーザーがソフトウェア開発業務を委託しようとする際は、これらの開発手法の作業の性質から見て、どうしても特定のベンダーに委託先を絞って請負型で委託せざるを得ない。

一方、ユーザーがソフトウェア開発を請負型で委託することは、委任型や派遣型で委託する場合に比べて、より大きなリスクを負うことにもなる。なぜなら、請負型の開発委託においては、どうしてもユーザーの業務情報をベンダーに提供せざるを得ないからである。特に、当該情報の中にユーザーの高度の機密情報（たとえば、企業秘密）が含まれている場合であれば、それが万一漏えいすると、ユーザーの経営や事業自体に大きな影響を及ぼすからである。

また、ユーザーにとっては大規模なソフトウェア開発になればなるほど、多額の費用もかかるので、そう簡単に初めに選んだベンダーを他のベンダーに変えることは容易ではなくなる。つまり、ユーザーは特定のベンダーを一旦選んで委託した以上、そのベンダーを信頼し、後戻りができないのである。

このようにユーザーが請負型でソフトウェア開発委託を行なうことは、ユーザーにとって、委任型や派遣型に比べて、大きなリスクを覚悟せざるを得ないことになる。

(2) 請負型開発委託取引の背景（なぜ、請負型が主流となっているのか）

一方、請負契約における請負人であるベンダーの法的義務・責任は、委任契約や派遣契約における受託者や派遣元事業者の法的義務・責任に比べてかなり重くなっている。このことから、本来ならば、ベンダーとしても請負型を避けたい筈である。しかしながら、今日のソフトウェア開発委託

取引実態を見ると、請負型が主流となっている。ユーザー・ベンダー双方にとって請負型はリスクや法的責任が大きいのに、なぜ、このように請負型が主流となっているのであろうか。この背景をベンダーの立場で検討してみる。

請負契約は、当事者の一方がある仕事の完成を約束し、相手方がその仕事の結果に対してその当事者に報酬を与えることを約束することによって効力が生じる有償・双務・諾成契約である（民法 632 条）。

今日の開発対象ソフトウェアは Web 対応・分散処理型情報システムが主流となっていることから、この情報システムの構築を請け負おうとするベンダーには、IT 関連総合技術や開発作業の取りまとめ力が要求される。しかし、ソフトウェア開発技術とコンテンツ作成技術やネットワーク技術はそれぞれ異なる専門分野の技術であり、ベンダーが、これら異分野の技術を全て保有することは事実上かなり困難であろうし、かつ、経営効率から見ても好ましくない。

そこで、前述した大規模住宅建築を行う総合建築事業者のように、ベンダーは、ネットワーク技術やデータベース技術など不足する技術は、当該技術を有する事業者と協力（業務提携や再委託など）することにより IT 関連総合技術力を発揮できるようになる。つまり、ベンダーは各種の IT 関連技術を有する事業者をとりまとめる能力を有すれば足りることになる。このようなソフトウェア開発委託取引におけるベンダーの受託業務形態は、情報システムの立案からハードウェアの調達、ソフトウェア開発、システム運用・保守まで一括してサービスを提供するシステムインテグレーション¹⁸⁾の事業形態に似ている。このシステムインテグレーション（以下「SI」という）は、かなり古くから行われてきている事業であるが、この事業者（システムインテグレータ）は、コンピュータ・メーカー、ソフトウェアハウス、情報サービス事業者などであり、これらの事業者は、SI サービス提供に必要となる全ての専門技術を保有しているわけではないので、

不足する技術については当該専門事業者の協力（業務提携・再委託など）により補完しながら事業遂行している。

今日の Web 対応・分散処理型情報システムの構築においても、ハードウェア選定からソフトウェア開発、ネットワーク接続など IT 関連総合技術が必要となるので、これらの業務をユーザーから一括して請負う場合は、ベンダーは、まさに SI そのものを行うことになる。このように情報システムの構築の受託業務形態を SI として捉えた場合であっても、この中心を成す業務はソフトウェア開発であり、他の業務（ハードウェア選定・調達、ネットワーク敷設・接続等）は、それぞれが専門技術を要する業務であるが、いわば付带的業務として捉えられることが多い。

以上の背景から、ベンダーがソフトウェア開発を受託するためには、どうしても一部の専門技術を要する業務を再委託せざるを得ない場合が多いので、委任型や派遣型を選択することができず、これが可能な契約類型として請負型を採用せざるを得ないのである。

（3）ソフトウェア開発請負契約の問題点

これまで見てきたようにソフトウェア開発委託取引では、請負型の契約（ソフトウェア開発請負契約）が主流となっている。しかし、実際の取引契約においては「業務委託」などの言葉を用いて、「請負」で行う旨明記している契約は多くない。このような実態になっているのは、ベンダー（請負人）にとって請負契約が準委任契約に比べて重い法的義務・責任を負わされるので避けたいという意識が強く働いていることも一つの理由と考えられる。

しかし、このような理由など実際の取引においては大した問題とはいええず、むしろ、請負契約はソフトウェア開発委託取引の契約に馴染むのだろうかという問題が大きいのではないだろうか。

請負契約は売買契約や賃貸借契約などと同様に民法が規定する典型契約

であり、私たちの日常生活においても運送契約、住宅建築契約、洋服オーダーメイド契約などの身近な契約として馴染んでいる。そして、請負契約は、仕事の目的物がある場合とない場合とに分けて請負人の法的義務・責任が一部異なっている。そこで、ソフトウェア開発委託を請負契約に当てはめて考えてみよう。

まず、ソフトウェア開発委託の請負契約における仕事の目的物は、もちろんソフトウェアであろう¹⁹⁾。

しかし、ソフトウェアは、住宅建築契約における注文住宅や洋服オーダーメイド契約における注文服などの身近な目的物とは、性質が大きく異なっている。住宅や洋服などは有体物であるから、注文者は、注文通りに仕事が完成したかどうかを実際に見て確認できるが、ソフトウェアについては、その外観を見ただけで完成しているかどうかの確認をすることは極めて困難である。そして、ソフトウェアは、プログラム言語で記述された著作物でもあるが、これ以外の典型的な著作物、例えば小説や絵画などように、その表現の良し悪しだけでその本質を推し量ることもできない。つまり、ソフトウェアは、あくまでもコンピュータで使用するによって、その機能や性能を発揮するという特有の性質を持っている。このことから、実際に、そのソフトウェアをコンピュータで使ってみないことには目的に合ったソフトウェアであるかどうかの確認はできない。このようにソフトウェアは、それが発揮する機能や性能を重視した著作物であり、その良し悪しは外観等だけでは見定めることができないのである²⁰⁾。

ソフトウェアは以上のような性質を本来的に有しているので、請負契約に馴染んでいる注文住宅や注文服などのように、あらかじめ具体的な完成図を描くことができない。このことから、一般的に、注文者であるユーザーは、ソフトウェア開発の委託に際して具体的な形で完成のイメージをベンダーに提示することが難しく、自ずと「このような処理ができる（機能を持つ）ソフトウェアを開発してほしい」という形でしか注文できない。

このユーザーの要求内容はあまりにも抽象的といえるが、請負人であるベンダーは、競合ベンダーとの競争上からも、自らの経験や技術等に照らし、この抽象的な要求内容をソフトウェアとして実現できるギリギリの範囲内で開発業務を受けざるを得ないのが通常といえよう。

そこで、ベンダーは、このユーザーの要求内容を実現できるようにするため、ソフトウェアを技術的な観点から基本設計書や機能仕様書として具体的に展開する必要がある。これらの設計・仕様書は、一方では、より具体的内容となっているが、他方では、技術的表現となっているため、ソフトウェア技術を有しないユーザーにとっては、これを正確に理解し、自らが提示した要求内容に合致したものであるかどうかを見極めることは困難といえよう。

以上のことを請負という契約の本質に照らして考えてみると、以下の問題が生じることになる。

請負契約における請負人の債務の本旨は、「ある仕事を完成すること（民法632条）」であるが、上述のソフトウェアの性質から開発対象ソフトウェアの内容を開発作業着手前に特定することが難しいことがわかる。この結果、ユーザーとベンダーとの間で、「完成すべき仕事の内容」についての認識にズレが生じ、ソフトウェアの開発完了に関する捉え方において乖離が生じやすい。そして、このような認識のズレ等が修正されないままで、完成を急ぎたいというユーザーの要望を受けて、ベンダーは開発作業に着手することが多いのではないだろうか。

このような状況の下でソフトウェア開発作業を進められ、ベンダーとしては目的とするソフトウェアを完成させたと認識し、ユーザーに請負契約の仕事の目的物である当該ソフトウェアが引き渡されたとする。そうすると、当該ソフトウェアの完成あるいは未完成を巡って、ベンダーは「既に完成している」と主張し、ユーザーは「未だ完成していない」と主張することになり、当事者間で法的紛争が生じやすいのである²¹⁾。

このソフトウェアの完成あるいは未完成を巡る問題は、今日の開発対象ソフトウェアが再委託やパッケージソフト等第三者の著作物を利用するものが多くなっているという実態を勘案すると、従来に比べて、技術的見地からも複雑・難題化することが予想される。更に、ソフトウェアについては知的財産権（主として著作権と特許権）が認められているので、この権利帰属や処理の問題も絡んで、ますますソフトウェア開発請負契約における法的問題は複雑化し、難解化してきているといえる。

この点、請負契約と同じ役務型の契約類型である準委任契約は、法律行為でない事務（ソフトウェア開発作業）の委託を目的（民法 656 条）としているから、受任者は善良な管理者の注意をもって当該事務処理を行い、その状況報告などを行なえばよい（民法 644～646 条など）。従って、請負契約のように仕事の完成あるいは未完成を巡る問題や瑕疵担保責任などの法的問題が生じる余地はほとんどないといえる。

以上の通り、請負契約によるソフトウェア開発委託取引においては、仕事の完成あるいは未完成を巡る問題を起因として当事者間で法的紛争へと発展しやすいと考えられるのである。

3. ソフトウェア開発委託取引における法的紛争処理の特徴

ソフトウェア開発委託取引において法的紛争が生じた場合、現実にはどのように処理・解決されるのであろうか。

わが国の企業等においては、日常的に各種取引を巡って数多くの法的紛争が生じていると思われるが、このうち紛争解決のため裁判まで持ち込まれるケースは、あくまでも氷山の一角と見られる。特に、ソフトウェア開発委託取引を巡る法的紛争について、最終的に裁判で解決されたというケースは非常に少ない。

なぜ、このように裁判での紛争解決が少ないのであろうか、特に、請負型のソフトウェア開発委託取引を巡る法的紛争解決における背景等について考察する。

(1) 法的紛争発生背景

ソフトウェア開発委託取引の契約は、ソフトウェアという技術的成果物を創り出すことを目的としているから、必然的に長期の継続的契約とならざるを得ない。また、ユーザーからの抽象的な種々の要求を受けて、ベンダーがこれを仕様書として具体化せざるを得ない請負型の取引においては、委任型や派遣型の場合に比べて取引当事者間で利害が対立する場面が多く、法的紛争を惹起しやすい。

そして、請負型では相手方の重要な秘密情報を保有するなど互いにリスクを負担せざるを得ないことが多いので、当事者双方（ユーザー及びベンダー）は、強固な相互信頼関係の樹立に努めなければならない。

この結果、仮に、当事者間で法的紛争を含む取引上のトラブルが生じたとしても、先ずはできるだけ当事者間の話し合いで解決することを優先する。なぜなら、わが国では、紛争当事者は「法的紛争を表沙汰にたくない」「他人を介入させず、内々で処理解決したい」という意識が強く働き、特に、企業等は法的紛争が生じていること自体が外部に漏れると、企業イメージが低下すると考えるのが一般的であるからである。

また、ソフトウェア開発委託取引を見ると、それぞれの取引ごとに契約内容が多種多様となっており、また未だに取引慣行が定着していないので、契約内容を定型化しにくく、運送、建築などの業界にあるような約款や標準契約書などは存在しない。このことから、この取引を巡る法的紛争の内容も個々に異なり、かつ単純ではない。つまり、近年は、紛争内容（事実関係）が複雑化しており、しかも、そこにはIT専門技術が絡む場合が多いのである。このことから、紛争解決をするためには、どうしても複雑に絡

んだ事実関係を理解する能力と IT 関連技術の知識が必要となる場合が多い。従って、紛争当事者が裁判や調停・仲裁等 ADR²²⁾ で当該紛争を解決しようとする場合、争点に関する有利な証拠提出が不十分となりがちであり、また、判断を下すべき立場の裁判官や仲裁人などが当該事実関係を十分に理解できないままに判断してしまうのではないだろうかという不安を抱くのである。

以上のことから、ソフトウェア開発委託取引に係わる法的紛争は、当事者間の話し合いで解決される場合が多く、外部の紛争解決手段（裁判や ADR）に付される場合は極めて少ないと推測できる。しかし、この場合でも当該ソフトウェアに関して特許権・著作権等知的財産権の帰属や侵害等を巡る法的紛争については、当事者以外の第三者（知的財産権者や侵害者など）が関与する場合もあり得るが、このように第三者が絡む法的紛争については、当事者間の話し合いで解決することができないので、裁判や ADR に付されることになるだろう。このような知的財産権を争点とした紛争解決には、IT 関連技術のほか知的財産権に関する専門知識が不可欠となるが、現在では、知的財産権関係専門の裁判所や仲裁機関が開設されていることもあり、また、ソフトウェア開発委託取引の契約内容を争点とした場合に比べて比較的に事実関係の立証が容易である場合が多いと思われることから、外部の紛争解決手段（裁判や ADR）に付されることが多いであろう。

ところが、ソフトウェア開発委託取引に係わる紛争においては、第一次的紛争解決手段である当事者間の話し合いによっても解決できなければ、第二次的紛争解決手段である裁判や ADR などに付さざるを得ない。この場合、前述した通り、当事者は相手方の技術や営業に関する各種の秘密情報を互いに保有し合っている場合が多いので、公開が原則である裁判で争うと、当該秘密情報が開示あるいは流出するというリスクを負うことになる。そこで当事者は、裁判に付すよりも非公開で行なわれる調停・仲裁等の

ADRに付そうとするものと思われる。しかし、ADRは原則として非公開で行なわれているため、現実には、どのような内容を争点として、どのくらいの数の紛争が処理されているのかなどについて公表されていないので、残念ながら、これらの実態の詳細を窺い知ることができない。

(2) 裁判による知的財産権関係紛争の解決

前項で論述したとおり、ソフトウェア開発委託取引において法的紛争となりやすい事項の一つに知的財産権（特に、著作権）の帰属や侵害を巡る法的紛争がある。これはソフトウェアが著作権法を主とした知的財産権法で保護されているからであり、また、当事者にとって当該ソフトウェアの知的財産権を保有するかどうかによって、以後生じる経済的な利益が大きく異なってくるからである。

このような知的財産権関係を争点とする場合、紛争解決機関には知的財産権についての高度な専門知識が要求される場合が多い。そこで、このような紛争解決するためことを専門とする機関や組織の設置が望まれていた。

先ず、知的財産権関係事件を専門に取扱う裁判所としては、従来から、東京と大阪の地方裁判所及び高等裁判所に専門部が置かれて、それぞれの裁判所で集中的に知的財産権関係事件を取扱っていた。そして、2005年（平成17年）4月1日に、知的財産権関係を専門的に取扱う高等裁判所として「知的財産高等裁判所」が設置された。これに伴って、従来の東京高等裁判所の知的財産第1部～第4部と第6特別部（知的財産大合議部）が、それぞれ知的財産高等裁判所の通常部（第1部～第4部）と特別部（大合議部）に移行した²³⁾。

このように知的財産高等裁判所は、知的財産権関係事件を専門に取扱う裁判所であるため、ソフトウェア開発委託取引にかかわる法的紛争のうち知的財産権関係以外の契約関係を争点とした事件は取扱わないことになる。

従って、知的財産権関係以外のソフトウェア開発委託取引に係わる法的

紛争の裁判事例は、全国の裁判所の民事事件の中から関係ありそうな事例を一つ一つ調べあげるほかに方法はない。

なお、裁判所でも民事調停事件を取扱っているので、ソフトウェア開発委託取引契約にかかわる法的紛争が裁判所の調停（ADR）に付されている可能性は高いが、この調停は非公開で行なわれているため、残念ながら、その数や紛争内容等の実態を正確に把握することはできない²⁴⁾。

4. ソフトウェア開発委託取引の法的紛争事例

(1) ADR における紛争事例

わが国における民間の ADR 機関としては、従来から（社）日本商事仲裁協会（2003 年 1 月に「（社）国際商事仲裁協会」から名称変更）や（社）日本海運集会所などが活動しているが、中でも、知的財産権関係事件の仲裁や調停などを行う ADR 機関としては、日本弁護士連合会と日本弁理士会が共同で設立した「日本知的財産仲裁センター」が有名である。

ここで、日本知的財産仲裁センターがどのような事件を取扱っているのかについて同センターの公式ホームページを調査した結果は、以下の通りとなっている²⁵⁾。

日本知的財産仲裁センターが取扱った事件の総数を見ると、2003 年度は 23 件、2004 年度は 14 件の事件を取扱っている。14 件の事件のうち、調停が 95%、仲裁が 5% の割合となっている。また、調停・仲裁事件の対象権利の種類を見ると、特許権 61%、商標権 18%、意匠 10%、著作権 5% の順となっており、契約は僅か 3% となっている。そして、これらの事件の内容については、同ホームページに「当センターで取り扱った事件は、それが当センターに係属したという事実を含めて、一切非公開になっており内容の紹介はできませんので、事件が特定できないように、また当事者が特定できないように内容を改変し、かつ当事者の了解が得られた数例を紹介

します。」と記載されており、この紹介されている数例を見ても、この中にはソフトウェア開発委託取引に関する紛争事例は1件もない。

このことから推測できるように、強固な信頼関係に基づくソフトウェア開発委託契約においては、非公開である調停・仲裁の事例を見つけ出すのは極めて困難と言わざるを得ない。

この他の民間ADR機関である（社）日本商事仲裁協会（主として商事紛争に関する仲裁・調停等を行なっている）や（社）日本海運集会所（主として海事に関する紛争を取扱っている）は、それぞれ取扱った紛争事件について、その処理数や紛争内容を公表していない。しかし、これらのADR機関の性格から見てソフトウェア開発委託取引に関する紛争は取扱っていないと考えられる。

（2）裁判事例における主要争点

ソフトウェア開発委託取引を巡って、どのような争点を中心に裁判上争われているのであろうか。まずは、ソフトウェア開発委託取引を巡る裁判事例（判例）を調べる必要がある。このため、「ソフトウェア開発／請負／製作」をキーワードに、判例検索システム（判例マスタ、LexisNexis、下級裁主要判決情報²⁶⁾等）を利用して、どのような判例があるか検索してみた。この結果、昭和63年12月から平成16年1月までに判決されたソフトウェアの開発取引を巡る裁判事例として26件抽出した。

ここで抽出した26件の裁判事例を主要争点別に適宜分類した結果、以下の通りとなった。

- ① 債務不履行を争点として事例；9件
- ② 瑕疵担保責任（プログラムの不具合、欠陥等）を争点とした事例；6件
- ③ 債務不履行と瑕疵担保責任を争点とした事例；3件
- ④ 著作権の帰属又は侵害を争点とした事例；3件

⑤ その他を争点とした事例；5件

この調査結果からも分かるように、債務不履行ないしは瑕疵担保責任を主要争点とした事例（①～③の合計）が18件と多く、全体の69%を占めている。

更に、これらの26件の事例において、紛争の原因ともいえる当事者間で取り交わされた取引契約の類型について調べてみると、当該取引が請負契約で行なわれたことが明確である事例は7件²⁷⁾であり、これ以外の事例では、ソフトウェア購入の契約²⁸⁾、ソフトバグ等の除去をタイムチャージで下請けする契約²⁹⁾、コンピュータシステムの代理店契約・ユーザー契約³⁰⁾、プログラムの製作を目的とした業務委託³¹⁾、機器販売契約³²⁾、コンピュータ注文についての売買契約³³⁾、総合行政情報システムの開発・導入を目指してされた合意³⁴⁾、などの様々な取引契約の名称をもって行なわれている。

この場合、「タイムチャージで下請け」や「プログラム製作を目的とした業務委託」といった取引は、一般的に、請負契約で行なわれたものとみなすことができる。また、購入・販売・売買などの契約名称で行なわれた事例は、一見、コンピュータ（ハードウェア）の購入等（売買契約）を目的としているように見えるが、主要争点の内容から考えると、何らかのソフトウェア開発業務（パッケージソフトのカスタマイズ等修正・改良を含む）を伴った取引であり、この業務は請負契約で行なわれたと見るべきであろう。

この結果、紛争処理が裁判に付されている事例を見ると、請負契約で行なわれたソフトウェア開発委託取引が多いことがわかる。更に、このことは裁判における争点が債務不履行ないしは瑕疵担保責任に関する事例が多数を占めていることも関連していると思われる。つまり、請負契約で行なわれたソフトウェア開発委託取引においては、債務不履行ないしは瑕疵担保責任に関して法的紛争が生じやすいのである。

これらのことは、「2. (2) 請負型開発委託取引の背景」及び「(3) ソフトウェア開発請負契約の問題点」で提起した取引実態上の問題点と符合しているように思われる。すなわち、ソフトウェア開発委託取引は請負型が主流となっていること、ソフトウェア開発請負契約では仕事の完成をどのように捉えるかが難しいなどに起因する問題点（債務不履行、瑕疵担保責任）が法的紛争の争点になりやすいということを裏付けているとも見ることができる。

おわりに

本稿では、ソフトウェア開発委託取引の実態を明らかにし、この取引を巡る法的紛争の裁判事例の概要とその主要争点が何かまで把握することができた。この結果、実態的にも主流である請負型のソフトウェア開発委託取引が多くの法的問題を内包しており、このことは裁判事例と照査しても「仕事の完成」に起因する法的紛争（債務不履行屋瑕疵担保責任を争点）が多いことが判明した。

本研究は、最終的にはソフトウェア開発委託取引における法的紛争原因を分析し、紛争解決のあり方を考察することを目的としている。従って、本稿は本件研究の中間成果報告であり、引き続き次稿では、本稿に掲げた裁判事例のうち債務不履行ないしは瑕疵担保責任に関する事例における法的紛争原因を更に分析したうえで、紛争解決のあり方を検討していく予定である。

註

- 1) 本稿でいうソフトウェアは、プログラム（正確にはコンピュータ・プロ

グラム) のことである。

- 2) ソフトウェアは、コンピュータ全体の機能(演算、記憶、入出力など)を制御する基本ソフト、インターネットなどの通信やデータベースの管理などを行なうミドルウェア(ミドルソフトともいう)、そして表計算や文書作成など一定の仕事を行なう(業務処理する)ためのアプリケーションソフトに大きく分類できる。これをパソコンの場合を例にとって考えると、基本ソフトは、マイクロソフト社の Windows に代表されているようにパソコンを使用するためには不可欠のソフトウェアであり、通常、OS(オペレーティングシステム)と呼ばれている。そして、店頭等でパソコンを購入すると、そのパソコンには、ほとんどの場合 Windows や Mac OS などの基本ソフトがあらかじめ搭載(プレ・インストール)されている。更に、MS-Word(文書作成)、Excel(表計算)などの幾つかのアプリケーションソフトもあらかじめ搭載されていることが多い。現在普及しているこれらのソフトウェアは、事実上の標準(de facto standard)となっており、世界各国のユーザーは自国の言語に合わせて、これらのソフトウェア(自国バージョン版)を使えるようになっていることから、一般的パソコン・ユーザーは、これらのソフトウェアを意識することなく、パソコンの一部の機能として使用しているのが実状であろう。
- 3) 従前はプログラムを専門的に作成する技術者を「プログラマー」と呼んでいたが、最近はソフトウェア開発に従事する技術者を総称して「SE=システムエンジニア」と呼ぶことが多い
- 4) 著作権法 10 条 1 項 9 号に規定されているプログラムの著作物は、伝統的に著作権法の保護を受けてきた小説や絵画などの著作物と比べて色々な面で異なった特性を有している。このため、著作権法はプログラムの著作物について多くの特例規定(同法 10 条 3 項ほか)をおいている。
- 5) DataBase Management System の略。共有データの集合体としてのデータベースを管理し、データに対するアクセス要求に応えるソフトウェアをいう。
- 6) 「free(自由)」に取扱えるソフトウェアの総称。この「free」は、無償で利用できるという意味と、ソースコードが入手でき、改変・再配布が制限なく行なえるという二つの意味で使われることが多いが、特に、後者の意味で

使われるフリーソフトをオープンソースと呼んでいる。

- 7) ソースコードを、インターネットなどを通じて無償で公開し、誰でもそのソフトウェアの改良、再配布が行なえるようなソフトウェア。The Open Source Initiative (OSI) という団体の定義「The Open Source Definition (OSD)」によれば、自由な再頒布の許可、派生ソフトウェアの頒布の許可、個人や集団の差別の禁止、適用分野の制限の禁止など10項目の要件を満たす必要があり、これに準拠しているソフトウェアライセンスには「OSI 認定マーク」が付与される。なお、オープンソースの現状と今後の課題の詳細は、ソフトウェア情報センター (SOFTIC) 研究会報告書 (<http://www.ipa.go.jp/NBP/14nendo/14cho1/030815opensoft.pdf>) を参照
- 8) ソフトウェア (プログラム) には著作権が与えられるので、その著作権者は利用条件を付してソフトウェアを提供することになる。このように利用条件を付してソフトウェアを提供することをライセンスといい、ライセンスされるソフトウェアをライセンスソフトという。なお、ソフトウェアの開発には多大なコストを要するので、フリーソフトとして公に無償で提供される場合を除き、有償でライセンスされることが多い。
- 9) OS とアプリケーションソフトの中間的性格を有し、データベースなどの特定の分野でアプリケーションソフトに対して OS よりも高度で具体的な機能を提供するソフトウェア。
- 10) Enterprise Resource Planning の略。企業全体を経営資源の有効活用の観点から統合的に管理し、経営の効率化を図るための手法・概念のことであり、これを実現する統合型 (業務横断型) ソフトウェアは、ERP パッケージとも呼ばれ、SAP 社の R/3 などが有名。
- 11) ソフトウェア開発取引に関する標準規格として日本工業規格 (JIS X 0160) や国際標準規格 (ISO/IEC12207) があるが、これに適合させてシステム開発作業のプロセス (工程) についても規格 (定義) したものに「共通フレーム 98—SLCP—JCF98」がある。この共通フレームの開発プロセスでは、この三段階の工程を13のアクティビティ (いわば詳細工程の作業名) に分けている。
- 12) ソフトウェア開発手法の変化については、(社) 情報サービス産業協会法的问题委員会契約部会 (部会長; 内布 光) 編「新しいソフトウェア開発取

引の契約と実務」(商事法務、2003) 12 頁参照

- 13) 電子商取引 (Electronic Commerce) は、e コマース、EC とも呼ばれ、インターネットを利用した契約や決済などを行なう取引形態である。取引相手により、企業間取引は「B to B」、企業・消費者間取引は「B to C」、消費者同士の取引は「C to C」と呼ばれている。なお、「B」は Business を、「C」は Consumer を指す。
- 14) 三つの開発手法の概要は、前掲注 12)「新しいソフトウェア開発取引の契約と実務」15 頁参照
- 15) 委託形態の詳細については、内布 光「IT 社会における企業取引法」(商事法務、2003) 66 頁参照
- 16) 正式名称は「労働者派遣事業の適正な運営の確保及び派遣労働者の就業条件の整備等に関する法律」で、職業安定法 44 条 (労働者供給事業の禁止) を適用除外するために昭和 60 年に制定された法律であり、近年盛んになっているアウトソーシング (業務の外部委託) などは、この法律に基づく派遣契約により行なわれている。
- 17) 吉田正夫「ソフトウェア取引の契約ハンドブック」(共立出版、1989) 40 頁では、「要求定義 (システム調査分析、外部仕様の決定)、システム基本設計は準委任契約であり、それ以後の工程 (システム詳細設計、プログラム設計、プログラミング、テスト) は請負ととらえられる」と述べている。
- 18) 1988 年に経済産業省がシステムインテグレータ認定・登録制度を創設しており、この制度により認定・登録された企業 (SI 認定企業) は、2003 年 7 月現在、444 社となっている。
- 19) 請負契約の報酬の支払時期について、民法は仕事の目的物の引渡と同時に定め、物の引渡を要しない場合は雇用の規定 (624 条 1 項) を準用して後払いとしている (633 条)。ソフトウェアそのものは、情報の集合体であるため無体物といえるが、通常はフロッピーディスク等の媒体 (有体物) に収納され一体となって流通しており、ソフトウェア開発請負契約においても、通常、媒体に収納した形で納品 (引渡) するのが実態となっている。そこで、ここではソフトウェアを引渡を要する仕事の目的物とみなすことにする。
- 20) 経済産業省が公表している「電子商取引等に関する準則 第 2. 情報取引 1. ライセンス契約」では、プログラム (本稿ではソフトウェア) やデジ

タルコンテンツを「情報財」と呼び、この情報財を「音、映像（画像）その他の情報であって、コンピュータを機能させることによって利用可能となる形式（いわゆるデジタル形式）によって記録可能な情報を指すものとする。」と定義している。詳細は、<http://www.meti.go.jp/policy/itpolicy/ec/e30613aj.html> 参照。

- 21) 前掲注12)「新しいソフトウェア開発取引の契約と実務」では、Web対応・分散処理型ソフトウェア開発取引における契約問題として、「仕様の確定と検収」などの11項目の論点を掲げているが、この「仕様の確定と検収」の問題は、まさに本稿で指摘している「ある仕事＝開発対象ソフトウェアの内容の特定」と「当該ソフトウェアの完成／未完成」の問題とほぼ同じ問題である。
- 22) Alternative Dispute Resolution の略で、わが国では代替的紛争解決手段（又は制度）と呼ばれており、裁判以外の行政機関や民間機関による和解、あっせん、仲裁及び民事調停・家事調停、訴訟上の和解などをいう。
- 23) 知的財産高等裁判所のほか、2005年（平成17年）4月1日現在、知的財産権関係事件を取り扱う専門部は、東京地方裁判所に4部、大阪地方裁判所に2部ある。また、大阪高等裁判所にも集中部がある。
- 24) 東京地方裁判所の調停委員などで構成されていると思われる「コンピュータ訴訟研究会」編の2冊の書籍（①「コンピュータ紛争事件のケース研究」尚文社、1999、②「コンピュータ紛争Ⅱ」尚文社、2000）が発刊されている。①では13件のシステム開発等を巡る紛争事件が、②では10件のソフトウェアに関する紛争事件が紹介されている。同書には「架空の事件についてケース研究をした」と断り書きがあるので、同書に掲載されている事例は実際の紛争事例でないとしても、実際の調停事件をヒントに想定したものと思われる。
- 25) 日本知的財産仲裁センターの活動概要等については、次の同センターのホームページ（URL；<http://www.ip-adr.gr.jp/>）を参照
- 26) 最高裁判所がホームページにて提供している「裁判例情報」の一つで、このほかに最高裁判例集、高裁判例集、行政事件裁判例集などがある。なお、「下級裁判主要判決情報」の検索ページ URL は次の通り。
；[http://courtdomino2.courts.go.jp/kshanrei.nsf/WebView/\\$SearchFor-](http://courtdomino2.courts.go.jp/kshanrei.nsf/WebView/$SearchFor-)

m?SearchView

- 27) 7件の判例は次の通り；①東京地判平 2. 3. 30 (昭 61 (ワ) 6549、昭 61 (ワ) 16768) 判時 1372 号 101 頁、②東京地判平 3. 2. 22 (昭 62 (ワ) 473、昭 62 (ワ) 4869) 判タ 770 号 218 頁、③東京地判平 5. 1. 28 (昭 62 (ワ) 6331) 判時 1473 号 80 頁、④東京地判平 12. 11. 24 (平 7 (ワ) 2630) 判例マスタ、⑤東京地判平 12. 12. 26 (平 11 (ワ) 6128) SOFTIC13 年度「ソフトウェア契約関連判例に関する調査研究報告書」43 頁、⑥東京地判平 13. 10. 17 (平 9 (ワ) 16792) 判例マスタ、⑦東京地 (八王子) 判平 15. 11. 5 (平 11 (ワ) 2327) 判時 1857 号 73 頁
- 28) 東京地判昭 63. 12. 23 (昭 59 (ワ) 5185) 判タ 705 号 179 頁
- 29) 東京高判平 5. 1. 25 (平 2 (ワ) 9098) 判タ 857 号 190 頁
- 30) 東京地判平 6. 1. 28 (昭 63 (ワ) 11776) 判時 1515 号 101 頁
- 31) 東京地判平 7. 6. 12 (昭 63 (ワ) 10976) 判時 1546 号 29 頁、判タ 895 号 239 頁
- 32) 東京地判平 8. 7. 11 (平 7 (ワ) 760) 判時 1599 号 99 頁
- 33) 東京地判平 9. 9. 24 (平 6 (ワ) 8866) 判タ 967 号 168 頁
- 34) 名古屋地判平 16. 1. 28 (平 11 (ワ) 3685、平 12 (ワ) 335) 下級裁主要判決情報