

# ソフトウェア開発を巡る法的紛争の動向 — 「プロジェクトマネジメント責任」を中心に —

内 布 光

## 目 次

1. はじめに
2. ソフトウェア開発の背景
  - 1) ソフトウェア関連市場規模
  - 2) ソフトウェア開発の必要性
  - 3) ソフトウェア開発市場及び技術者数の日米比較
  - 4) ソフトウェア開発形態の日米比較
  - 5) 小括
3. ソフトウェア開発におけるプロジェクトマネジメントの問題
  - 1) プロジェクトマネジメントの必要性
  - 2) ソフトウェア開発手法の変化とプロジェクトマネジメント
  - 3) 分散処理型システム開発とプロジェクトマネジメント
  - 4) 多段階契約とプロジェクトマネジメント
  - 5) ソフトウェア委託開発におけるユーザーの役割
4. ソフトウェア委託開発におけるベンダーの義務
  - 1) ソフトウェア委託開発の契約類型
  - 2) ベンダーが負うべき契約上の義務
  - 3) 開発作業におけるプロジェクトマネジメントの意義
  - 4) ソフトウェア委託開発を巡る法的紛争の背景
5. プロジェクトマネジメント義務に関する判例動向
  - 1) 健保組合システム開発事件
    - (1) 本件の概要
    - (2) プロジェクトマネジメント義務の内容
    - (3) 本件判決の実務への影響
  - 2) スルガ銀行対日本 IBM 事件
    - (1) 本件の概要

- (2) 裁判の経緯
  - (3) 本件プロジェクト失敗の背景
  - (4) プロジェクトマネジメント義務の内容
  - (5) プロジェクトマネジメント義務の範囲
- 3) 小括
6. おわりに

## 1. はじめに

ソフトウェア開発<sup>1)</sup>は、いわば「無」から「有」を作り出す作業であり、その開発工程（プロセス）は、建築物（例えば、注文住宅）の建築プロセスに似ている。しかし一方では、開発成果物であるプログラム<sup>2)</sup>は、建築物と比較すると決定的に異なっている。つまり、プログラムは、プログラミング言語で記述された著作物であり、しかもコンピュータで実際に使用してみなければ、その良し悪しがわからないという特性を有している。このため従来からソフトウェア委託開発取引を巡って委託者と受託者との間でトラブルが生じやすく、中には法的紛争（裁判）にまでに発展することがあった<sup>3)</sup>。

そして今日では、ICT<sup>4)</sup>の進展に伴って、私たちは、パソコンで様々な種類のプログラム（ソフトウェア）を簡単に使用できるようになった。このようなコンピ

- 
- 1) ソフトウェア開発は、プログラム（コンピュータプログラムの略語）を作成するための一連の作業プロセスをいう。なお、ソフトウェアは、ハードウェア（コンピュータなどの機器類）に対する用語であるが、一般に「ソフトウェア」という場合、プログラムを指すことが多い。
  - 2) 著作権法（第2条第1項十の二）では、プログラムについて「電子計算機を機能させて一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をいう。」と定義している。
  - 3) 内布 光「ソフトウェア開発委託契約紛争事例の研究（1）」現代法学第10号（2005年11月）P157～186では、ソフトウェア開発委託取引を巡って生じた法的紛争のうち、昭和63年12月から平成16年1月までに判決された裁判事例26件についての主要争点等を分析している。
  - 4) 従来、一般に、IT（「Information Technology（情報技術）」の略語）の用語が使われていたが、近年はインターネットなどの通信（Communication）が欠かせないため、情報技術と通信技術を合わせた合成語の「Information Communication Technology（情報通信技術）」の略語として、この「ICT」が広く使われている。

ユータの使用環境の変化を受けて、企業等の情報システム（ソフトウェア）は、一部の限られた者だけが使用できるものでなく、インターネットやLAN（構内ネットワーク）を利用して多数の人が簡単に使用できるようなもの、すなわち操作性が重視されるようになった。この結果、企業等の情報システム（ソフトウェア）の開発形態、開発手順・手法、作業工程等も、従来とは大きく変容している。

しかし、ソフトウェア自体の特性は、従来と本質的に変わらない。そしてソフトウェアを開発するには、所要の専門的知識・技術が欠かせないなどのソフトウェア開発作業の特殊性なども従来と大きな差異がない。このため今日においても、ソフトウェア委託開発の取引を巡って当事者（委託者と受託者）間でトラブルが生じ、裁判（ソフトウェア開発関連訴訟）まで発展することも後を絶たないようである。

そして、ソフトウェア開発関連訴訟の争点については、従来は、所定の納期までにソフトウェアが完成しないなどの債務不履行の問題やソフトウェアの不具合（プログラム・バグ）などの瑕疵担保責任の問題がほとんどであったが<sup>5)</sup>、近年では開発体制・推進における「プロジェクトマネジメント義務・責任」<sup>6)</sup>が新たな争点としてクローズアップされている。

通常、訴訟（裁判事件）を担当する裁判官は、当該事件に関連する豊富な法律知識と経験は有している。しかし、民事訴訟に限定しても訴訟原因である法領域は広く、特に、専門技術等が絡む事実関係の究明・把握においては、事実関係を構成する当該専門的技術等についての知見が必要となる場合が多い。その典型の一つが、いわゆるソフトウェア開発関連訴訟であろう。そこで、この種の事件（訴訟）を担当する裁判官にとっては、ソフトウェアの内容や仕組み等についての一般的な知見を持つとともに、ソフトウェア開発の特殊性（特有の背景・事情等）についての十分な理解が欠かせない。このためソフトウェア開発関連訴訟を“複雑困難な訴訟類型の一つ”と位置付け、東京地方裁判所<sup>7)</sup>や大阪地方裁判所<sup>8)</sup>で

5) 内布 光「ソフトウェア開発委託契約紛争事例の研究 (2)」現代法学第11号(2006年3月)P93～142では、前掲3)の拙著「ソフトウェア開発委託契約紛争事例の研究(1)」で取り上げた裁判事例26件について紛争原因等を分析している。

6) 「プロジェクトマネジメント義務」の言葉が判例上はじめて使われたのは、東京地裁民事第24部平成16年3月10日判決(判例タイムズNo.1211、P129～187)においてである。

は、この種の事件に係る裁判官や書記官が研究会等を作り、研究成果を幾つかの論考として発表している。

また、このようなソフトウェア開発関連訴訟は、東京や大阪に限らず全国の裁判所でも提起されるであろうし、それを担当する裁判官にとっては大きな負担となることが容易に想定されるので、司法研修所<sup>9)</sup>は、「民事実務研究会 (IT)」を毎年1回開催するなどして、この種の事件 (ソフトウェア開発関連訴訟) に対する対応力も高める努力を行っているようである。

そして筆者は、「平成 26 年度民事実務研究会 (IT)」<sup>10)</sup>に招かれて「ソフトウェア開発の背景事情」と題して講演し、その後、事前に参加者 (裁判官) から寄せられた「話題事項」を中心に意見交換を行う機会を得た。今日では、パッケージ・ソフト利用が主流となっているので、ソフトウェア開発関連訴訟は、ほとんどなくなっているのではないかと思っていた。しかし、このような裁判官との意見交換などを通して、ソフトウェア開発関連訴訟が、従来と変わらず散発しており、しかも最近の訴訟では、従来はあまり問題とならなかった「プロジェクトマネジメント義務・責任」が大きな論点となっていることも知った。

本稿は、以上の講演や意見交換の内容を整理し直して、ソフトウェア開発技術者 (プロジェクトマネージャ) 等へのヒアリングなどの調査結果等も踏まえて、今日の大きな論点となっている「プロジェクトマネジメント義務・責任」を中心に考察する。

---

7) 東京地方裁判所プラクティス委員会第二小委員会「ソフトウェア関係訴訟の手引き」(判例タイムズ No. 1349、P4～27)

8) 大阪地方裁判所裁判官田中俊次ほか 8 名 (専門委員)「ソフトウェア開発関連訴訟の審理」(判例タイムズ No. 1340、P4～15)

9) 司法研修所は、裁判所法 14 条に基づいて設置された研修機関で、裁判官の研究及び修養並びに司法修習生 (司法試験合格者) の修習などをつかさどっている。

10) 「平成 26 年度民事実務研究会」には、全国の 8 高裁管内の裁判所から現職の裁判官 35 名の参加があり、2014 年 7 月 7 日 (月)～8 日 (火) の 2 日間にわたり実施された。なお、筆者は、これに先立って 2014 年 3 月 11 日開催の研究会 (東京高裁管内の裁判官 10 数名参加) に招かれ、「ソフトウェア開発委託取引を巡るトラブル・紛争について」と題して講演した。

## 2. ソフトウェア開発の背景

### 1) ソフトウェア関連市場規模

ソフトウェア開発委託取引を巡って、なぜ法的紛争が生じやすいのかを理解するために、ソフトウェア開発の背景や事情などについて把握しておく必要がある。そこで、現在、わが国では、どの程度のソフトウェア開発ニーズがあるかを知るために、公表されているソフトウェア関連市場に関する統計・調査データを拾い上げてみる。

まず、総務省の「平成 25 年版 情報通信白書<sup>11)</sup>」によると、平成 23 年情報通信産業市場規模（名目国内生産額）は約 82.7 兆円であり、これは全産業の市場規模の 9.0% を占め、情報通信産業<sup>12)</sup>が国内の最大市場となっている。この情報通信産業のうち、情報サービスの市場規模は約 17 兆円であり、これは情報通信産業全体の 20.7% に当たる。更に、情報サービスのうち、ソフトウェア<sup>13)</sup>の市場規模は約 9.6 兆円であり、情報通信産業全体の 11.6% を占めている。

次に、民間大手の調査会社である IDC Japan が公表しているレポート「平成 26 年国内 IT 市場規模（予測）」によると、IT 産業<sup>14)</sup>の全体の市場規模は約 14 兆円であり、このうち IT サービスの市場規模は、約 5.1 兆円で、パッケージソ

---

11) 総務省は、毎年、日本の情報通信産業の現状等を調査して「情報通信白書」として公表している。なお、本稿で利用している全ての統計データは、平成 26 年 1 月現在で公表されていた最新データであるが、「情報通信白書」の最新版（平成 27 年版は、<http://www.soumu.go.jp/johotsusintokei/whitepaper/>）で公表されている。

12) 「情報通信産業」とは、前掲 4) の「ICT」に関連した産業の総称である。前掲 11) の情報通信白書では、①通信業、②放送業、③情報サービス業、④インターネット附随サービス、⑤映像・音声・文字情報制作作業、⑥情報通信関連製造業、⑦情報通信関連サービス業、⑧情報通信関連建設業、⑨研究の 9 業種をいう。

13) 「ソフトウェア」は、不特定多数のユーザー向けに開発された「パッケージソフト」と特定ユーザー用に新規開発する「カスタムソフト」に大別できる。前掲 11) の「情報通信白書」では、情報サービス業を、更に「ソフトウェア」と「情報処理・提供サービス」の 2 業種に区分している。

14) 「IT 産業」とは、ハードウェア製造販売やパッケージ・ソフト販売から IT サービスまでを含んだ IT に関連した全産業を総称している。そして、「IT サービス」は、①ハードウェア保守、②ソフトウェア保守、③コンサルティング、④開発・インテグレーション、⑤ IT マネジメント、⑥業務マネジメントの 6 分野に区分される。

フト<sup>15)</sup>の市場規模は約 2.7 兆円である。また、IT サービスとパッケージソフトの市場規模を合わせると、約 7.8 兆円となっている。

以上に掲げたとおり、情報通信白書（総務省）の公表データ数値と IDC Japan の公表データ数値とは、調査時点や市場の捉え方・区分方法に相違があるため、かなりの差異が生じている。

いずれにせよ以上のことから、わが国の最大産業である情報通信産業の中でもソフトウェア市場が重要な基幹産業となっていることが明らかである。

## 2) ソフトウェア開発の必要性

ここで、なぜ、わが国ではソフトウェア開発に対して根強いニーズがあるのかについて一考してみる。

私たち（個人）は、家庭でも日常的にインターネット上で検索したり、メールやフェイスブックなどの SNS（Social Networking Service）などで通信したり、文書作成や表計算などの情報処理をしたり、このほか様々な用途でパソコンやスマートフォンなどの情報機器を使用している。このようにパソコン等の情報機器は、今や、日常生活に欠かせないものとなっている。そしてパソコン等で使用している検索ソフト（ブラウザ）、メールソフト、文書作成や表計算ソフトなどのソフトウェアの殆どは、通常、パソコン購入時に、既に登載（プレ・インストール）されていたり、その後必要に応じてダウンロードしたり、市販のパッケージソフトを購入したりしたものである。

またパソコンは、企業等の職場でも、仕事を行う上で欠かせないものとなっている。企業の従業員等がパソコンを使って検索・通信や情報処理などの仕事を行う場合、そこで使われるメールソフトや文書作成、表計算ソフトなどは、日常生活におけるパソコンの使用の場合と同様に、プレインストールされたソフトウェアをそのまま利用することが多い。しかし、企業における仕事上の使用が日常生

---

15) 「パッケージソフト（英語では；packaged software）」は、単に「パッケージ」とも呼ばれる。狭義には、特定の業務あるいは業種で汎用的に利用することのできる既製の市販ソフトウェアである。「Windows」や「Internet Explore」などのソフトウェアは、マイクロソフト社がパソコンメーカーに対して包括的（再販権付）ライセンスしているので、私たちがパソコンを購入すると、これらのソフトウェアは既搭載（プレインストール）されている。

活での使用と大きく異なる点は、そのパソコンを当該企業所定の様々な業務処理システムの端末機としても使用していることである。

企業が、このような業務処理システムなどの情報システムを構築しようとする場合、その企業独自の情報システムを新たに開発するのに比べて、出来合い（市販）のハードウェア（パソコン等）やソフトウェア（パッケージソフト）などを利用して構築する方が格段に安価で効率的である。つまり、従業員等が使用するハードウェアは、専用端末機でなく市販のパソコンで十分であり、また、本支店間を結ぶ通信回線は、専用回線でなくインターネット等公衆回線の利用でよく、日常的に使用する文書作成、メール等の各種ソフトウェアも、そのパソコンにプレインストールされたソフトウェアや市販パッケージソフトでも何ら支障がなく使用できる。

一方、どのような企業でも、給与計算業務や会計処理業務などの全ての企業に共通する業務が存在する。このような業務は、企業によって処理データ項目等の一部が異なっていることが多いが、大筋において処理方法等の基本的部分は、業種に関係なく同じである。従って、これらの業務処理用のソフトウェア（プログラム）は、ソフトウェア会社等があらかじめ汎用的な仕様（できるだけ多くの企業がそのまま使えるような内容）で開発し、それをパッケージソフトとして市販している場合が多い。特に、様々な業務処理を一つのパッケージとして統合したソフトウェアの代表が「ERP」<sup>16)</sup>と呼ばれるパッケージソフトである。そこで各企業は、これらのパッケージソフトの中から自社に適したものを選んで使ってもよい。

しかし、各企業の定款所定の目的（事業）に関連する業務処理の方法等は、各社各様である。例えば、電気製品の販売業を営む企業を見ても、その販売業態には、特約店や代理店、量販店など様々であり、また、それぞれの販売業態によって、取り扱っている製品の種類・量なども大きく異なっている。すると、同業種

---

16) ERP (Enterprise Resource Planning の略) は、もともと「企業の持つ人材、資金などの様々な資源を統合的に管理・配分し、業務の効率化や経営の全体最適を目指す手法」(IT用語辞典 (e-Words)) という意味であるが、通常、このために導入・利用される統合型 (業務横断型) 業務ソフトウェアのことをいう。企業向け ERP ソフトウェアで有名なものが、ドイツ SAP 社製の「R/3」である。

の企業間でも、それぞれの企業における販売管理（業務処理）の方法や手順等も必然的に大きく異なることになる。

そして、わが国では、企業の規模が大きくなればなるほど、「他社のまねをしない」、「独自性を重んじる」という独自の企業文化が醸成されていることが多い。

このように各企業における共通の同種の業務処理においても、企業ごとに、その方法や手順等（インプット項目やアウトプットデータ項目、様式等も含む。）が異なる場合が多い。そこで、各企業は、当該業務処理用情報システムのソフトウェアとして適合するパッケージソフトが市販等されているとしても、そのパッケージソフトをそのまま利用することは殆どない。仮に、当該パッケージソフトを利用する場合においても、何ら手を加えずそのまま利用することは皆無であり、各社独自の業務処理方法・手順に合わせて当該パッケージソフトを部分的に修正・改良することがよく行なわれている。パッケージソフトを部分的に修正・改良する作業のことを「カスタマイズ」<sup>17)</sup>と呼んでいるが、このカスタマイズ作業もソフトウェア開発作業の一形態といえる。しかし、一般にカスタマイズできる範囲は、当該パッケージソフトの技術的制約等から自ずと狭いので、使用者として企業の要求仕様を完全に満たすことはできない場合が多い。そこで、日本では伝統的に、各企業は独自の要求仕様を満たすソフトウェアを新規に開発しようというニーズが根強いのである。なお、出来合いのパッケージソフトに対して、このようにして新規開発されるソフトウェアのことを「カスタムソフト」と呼んでいる。

これに対して、従来から、欧米、特に米国の企業では、業務処理用ソフトウェアについてもパッケージソフトの利用が主流であり、カスタムソフトの開発が少ないといわれている。そして、ソフトウェア開発取引を巡る法的紛争・トラブル事例がほとんど漏れ聞こえてこない。これは実際に、日本に比べて米国では法的

---

17) 「カスタマイズ (customize)」は、広く顧客 (customer) の要求に合わせて作り変えることを意味するが、ソフトウェア開発においては、パッケージソフト (市販ソフトウェア) を使用者の要求に合わせて部分的に修正・改良することを意味する。通常、パッケージソフトは、オブジェクトコード (機械語) で提供されているので、使用者側でその内容 (プログラム) を修正・改良することはできない。これに対して ERP (統合業務ソフトウェア) や個別ライセンスのパッケージソフトの中には、あらかじめカスタマイズできる部分を設定しているソフトウェアもある。

紛争・トラブルの発生が少ないからであろう。

### 3) ソフトウェア開発市場及び技術者数の日米比較

日本では、なぜ、米国に比べてソフトウェア開発取引を巡る法的紛争・トラブル発生件数が多いのであろうか。この原因を究明するためには、日米両国におけるソフトウェア開発の背景・事情を考察する必要がある。そこで、ソフトウェア開発の背景をなす日米両国のソフトウェア開発市場及び技術者数を比較してみる。

この比較のため、IPA（正式名称「独立行政法人 情報処理推進機構」）の平成23年3月「グローバル化を支えるIT人材確保・育成施策調査」調査報告書<sup>18)</sup>（以下「IPA調査報告書」という。）に掲載された調査データ数値を用いる。

まず、「ITサービス市場規模（2009年）<sup>19)</sup>」について比較すると、米国が約2,945億ドルで世界第1位であるのに対して、日本は約1,070億ドルで、米国が日本の2.7倍となっている。なお、日本の市場規模は世界第2位となっている。また参考までに主要国のITサービス市場規模を掲げてみると、世界第3位が英国（705億ドル）、第4位がドイツ（430億ドル）、第5位がフランス（320億ドル）と続いている。

更に、ITサービスのうち「ソフトウェア開発<sup>20)</sup>」の市場規模は、米国が約855億ドル（米国のITサービス市場全体の29.0%）であり、日本が約363億ドル（日本のITサービス市場全体の33.9%）となっている。このよう日本のソフトウェア開発の市場規模は、日本は米国の42.5%であるが、ITサービス全体に占める割合は、日本の方が若干大きい。

次に、ソフトウェア開発に従事する技術者（いわゆるソフトウェア技術者）の

---

18) 「IPA調査報告書（[www.ipa.go.jp/files/000010609.pdf](http://www.ipa.go.jp/files/000010609.pdf)）」では、ITサービス市場規模や技術者数などについて、米国、中国、インド、ベトナム、韓国、ロシアなどの各国と日本とを比較している。

19) 「IPA調査報告書」にある「ITサービス市場規模（2009年）」のデータの出典は、主として「ガートナー／Market Share: IT Services, Worldwide, 2007-2009（2010年4月）」である。

20) 「ITサービス市場規模（2009年）」におけるサービス分野は、①ハードウェア保守、②ソフトウェア保守、③コンサルティング、④開発・インテグレーション、⑤ITマネジメント、⑥業務マネジメントの6分野に区分されている。そこで本稿では、④開発・インテグレーション市場規模をソフトウェア開発の市場規模とみなしている。

人数について日米両国を比較してみる。なお、ソフトウェア技術者は、一般に、本人が有するソフトウェア開発の技術力や業務経験等によって、プロジェクトマネージャ、システムエンジニア、プログラマ等の職種に区分できる。また、通常、特定の企業に所属して（従業者として）、その企業の一員としてソフトウェア開発業務に従事するケースが多い。そして、これらのソフトウェア技術者が所属する企業は、一般に、ソフトウェアを使用する一般企業（以下「ユーザー」という。）又は、ソフトウェア開発などを行う IT サービス企業（以下「ベンダー」という。）のどちらかに大別できる。

それでは、IPA 調査報告書掲載の調査データ数値を用いて、どれだけの人数のソフトウェア技術者がベンダーとユーザーのどちらの企業に所属しているのかを比較すると下表<sup>21)</sup>のとおりとなる。

表 1. IT 関連業務就業者数の日米比較

所属企業	(a) 米国 (構成比)	(b) 日本 (構成比)	(b) / (a)
ベンダー	941,410 人 (28.5%)	771,426 人 (75.2%)	81.9%
ユーザー	2,362,300 人 (71.5%)	254,721 人 (24.8%)	10.8%
合計	3,303,710 人 (100.0%)	1,026,147 人 (100.0%)	31.1%

なお、上表（表 1.）の人数は、ソフトウェア技術者を含む IT 関連業務就業者の数であるが、その殆どがソフトウェア技術者であろうから、ここでは全員をソフトウェア技術者数とみなしている。

まず、ソフトウェア技術者の総数を比較すると、米国が約 330 万人であり、日本が約 103 万人であるので、米国は日本の約 3.2倍<sup>22)</sup>という豊富なソフトウェア技術者を擁していることになる。

21) この表は、前掲 17) の IPA 調査報告書 (P 44) 「図表 3-14 各国の IT 技術者数 (2009)」に掲載の日米両国の数値をベースに筆者が加工したもの。

22) ソフトウェア技術者数を総人口に占める割合で比較してみると、米国の総人口が約 3 億 1,500 万人 (2013 年 1 月現在)、日本が約 1 億 2,730 万人 (2014 年 10 月現在) であるから、日米両国とも技術者数は、総人口の概ね 1% で、ほぼ同じである。

次に、これらのソフトウェア技術者がユーザー（ソフトウェアを使用する側の企業）、ベンダー（ソフトウェアを開発する側の企業）のどちら側の企業に所属しているかを比較してみる。すると、日本では75.2%のソフトウェア技術者がベンダーに所属し、米国では日本とは全く逆に、71.5%のソフトウェア技術者がユーザーに所属していることがわかる。

#### 4) ソフトウェア開発形態の日米比較

前項の表1. から、日本ではソフトウェア技術者の大半がベンダーに所属し、逆に米国ではユーザーに所属していることがわかった。そこで、このことをユーザーが自企業内で使用するソフトウェア（例えば、給与計算システム等の各種業務処理システム）を開発（パッケージソフトのカスタマイズを含む）する場合に照らして、日米のソフトウェア開発形態の相違について考察してみる。

日本では、当該ソフトウェア開発に必要なかつ十分なソフトウェア技術者を雇用していない場合が多い。すると、大半のユーザーが、当該ソフトウェアの開発を自社内で開発（いわゆる「内作」）できる能力レベルでない（自社開発能力がない）。従って、このようなユーザーは、当該ソフトウェアの開発を、ベンダーへ委託（一種の丸投げ）せざるを得ないことになる。

これに対して、米国では、自企業内で使用するソフトウェアの開発（ソフトウェアの保守を含む。）に必要な人数のソフトウェア技術者を雇用（正規雇用の者だけでなく契約社員などのゆるい雇用関係にある者まで含む。）しているユーザーが多い。そこで、殆どのユーザーがソフトウェア（各種業務処理システム）開発のプロジェクトを立ち上げる際、必要なソフトウェア技術者を自社内から集めて開発（いわゆる「内作」）しているものと思料できる。つまり、米国の企業におけるソフトウェア開発は内作を中心に、技術力が不足・不十分な部分だけを外部委託すればよいことになる。

また、米国は、日本と異なりパッケージソフト利用の文化が根付いている。この点が、企業の基幹業務システムを構築する場合のソフトウェア開発形態においても、日米間では、大きな差異が生じる。

日本では、一般的に「他社のまねをしたがらない」、「独自性を重んじる」という企業文化が根付いている。このことから、日本の企業では、自社内で使用する

ソフトウェアも、その企業独自仕様のカスタムソフトに執着しがちである。従って、新技術を導入した基幹業務システムを構築する場合、それまでの旧システム用のソフトウェアは廃棄して、新システム用ソフトウェアを新規開発することが多い。仮に新システムに適合するパッケージソフトが存在する場合でも、当該パッケージソフトをそのまま使用することはせず、それをカスタマイズするなど何らかの手を加えがちである。つまり、ソフトウェア開発作業が必ずと言ってよいほど生じやすいのであるが、その開発作業に適切なソフトウェア技術者を擁さない日本の殆どのユーザーは、当該ソフトウェア開発をベンダーに委託せざるをえないことになる。

これに対して米国では、多くのユーザーは、自企業内に基幹業務システムの維持や管理に必要なかつ十分なソフトウェア技術者を擁しているのも、これまでに内作した各種業務システム用のソフトウェア資産も豊富に蓄積していると考えられる。そこで、新技術を導入した基幹業務システムを構築する場合においても、できるだけ適合したパッケージソフトを採用し、これまで内作して蓄積した既存ソフトを改良するなどして利用して、当該新システムのソフトウェアを開発（内作）することが多いと思われる。

## 5) 小括

日本の企業は、一般に「訴訟（裁判）嫌い」である。しかしながら、ソフトウェア委託開発取引においては、ユーザーとベンダーとの間には利害が対立する場面が多いのでトラブルが生じやすい。そこで、本稿の冒頭でも述べた通り、これまでもソフトウェアの委託開発を巡ってユーザーとベンダーとの間で生じたトラブルが裁判まで発展したケースが幾つか散見された。

一方、米国では「訴訟社会」ともいわれるほど裁判が日常茶飯事となっているので、ソフトウェア開発関連訴訟が数多く行なわれていると思われるが、なぜか、ソフトウェアの委託開発を巡る裁判事例が殆ど漏れ聞こえてこない。

この最大の理由は、米国のユーザーは、できるだけパッケージソフトをそのまま利用することでソフトウェアの新規開発を避けるからであろう。また、米国では、多くのユーザーが自社内に必要かつ十分な人数のソフトウェア技術者を擁しているから、新たなソフトウェア開発ニーズ（プロジェクト）が生じた場合でも、

当該プロジェクトに適する自社内のソフトウェア技術者を集めてプロジェクトチームを編成し、一貫して自社内で開発（内作）することができ、外部（ベンダー）にその開発を委託する必要がないからであろう。

この結果、米国でユーザーが外部（ベンダー）へソフトウェア開発の委託を行おうとする場合は、ユーザー自身が当該ソフトウェアと同種のソフトウェア開発の経験を全く有しないとき、又は、当該ソフトウェア開発に必要とされる開発能力（技術等）が不足・不十分であるときなどに限られ。この結果、日本に比べてソフトウェア開発の委託取引自体が少なくなり、自ずと紛争件数も少なくなる。

これに対して、日本のユーザーは、パッケージソフトをそのまま使用するよりも、自社業務の処理手順等に合わせたソフトウェアの開発を好むが、自社内に必要かつ十分な人数のソフトウェア技術者を擁しているユーザーは少ない。

この結果、新たなソフトウェア開発ニーズ（プロジェクト）が生じた場合、それを一貫して自社内で開発（内作）することができないので、外部（ベンダー）にその開発を委託せざるを得ない。しかも、ソフトウェア開発能力（技術等）はもとより、ソフトウェア開発に関する専門的知識も乏しいユーザーが多いので、いわばベンダーに「丸投げ」しがちである。

### 3. ソフトウェア開発におけるプロジェクトマネジメントの問題

#### 1) プロジェクトマネジメントの必要性

一般に「プロジェクト」とは、ある目的を達成（仕事を完成）するための計画の策定とその遂行のことをいうが、情報システム（ソフトウェア開発）の現場では、特定の情報システムの開発（計画）のことを「プロジェクト（project）」と呼んでいる。また、その開発作業の進捗を管理することを「プロジェクトマネジメント（project management）<sup>23)</sup>」といい、プロジェクトを管理する人や役職のこ

23) “プロジェクトマネジメント”とは、「チームに与えられた目標を達成するために、人材・資金・設備・物資・スケジュールなどをバランスよく調整し、全体の進捗状況を管理する手法」（IT用語辞典〈e-Words〉）のことであり、従来型のQCD（品質・コスト・納期）の3つに着目したマネジメント手法と区別する際には、特に“モダンプロジェク

とを「プロジェクトマネージャ (project manager)」と呼んでいる。

通常、ソフトウェア開発作業においては、多数のソフトウェア技術者等が必要であり、それぞれの技術者等がその専門分野の作業を分担して開発作業を行う。また、あらかじめ決められている当該ソフトウェアの使用（新システムの運用）の開始時期に間に合うように開発完了期限も設定されるので、開発から運用開始に至るまでの作業スケジュールも開発作業着手前に策定しておかなければならない。このようにして行われるソフトウェア開発（計画）は、まさしく「プロジェクト」そのものである。

そして、ある程度大きな規模のソフトウェアを開発する場合は、一連の開発工程（プロセス）で所定の順序により諸作業を行わなければならない。そこで、このようなソフトウェアの開発を計画するときは、これらの諸作業のスケジュール（工程表）を確定し、作業環境の整備、必要人員や技術の確保から配置等までの「作業推進体制（いわゆるプロジェクト推進体制）」を確立するなどの計画を策定しなければならない。

このようにして予め策定された計画（作業スケジュール表）に従って開発作業を開始することになるが、この一連の開発工程で所定の順序により行うべき諸作業の進捗状況（予定通りに開発作業が進んでいるかどうか）については、常に把握していく必要があり、万一、作業の進捗を阻害するような問題が生じた場合は、これに速やかにかつ的確に対処する必要がある。これが「プロジェクトマネジメント」と呼ばれるもので、この役割の中心を担う人が「プロジェクトマネージャ」である。

## 2) ソフトウェア開発手法の変化とプロジェクトマネジメント

ソフトウェア開発は、例えば住宅を新築する場合のように、いわば“無”から“有”を創り上げるための一連の技術的な作業である。この開発作業による成果（ソフトウェア）は、知的財産権（著作権、特許権等）で保護される。また、開発されたソフトウェアの良し悪しは、最終的には、それをコンピュータで使ってみなければ評価できないという特徴を持っている。このことから当該ソフトウェア

---

トマネジメント”と呼ばれている。

が所定の使用環境の下で一定の機能を発揮できるものであることは絶対的条件であるが、これに加えて近年は、誰でも（IT技術を持たない者でも）容易に使えるという「使い勝手の良さ（操作性が良いこと）」なども重視されている。

一方、企業の情報システム（基幹業務システム）の使用環境を見ると、情報処理方式が従来の汎用コンピュータ（センター）による一括集中処理型から、今日では多数のクライアント（パソコン）と複数のサーバーによる分散処理型（クライアントサーバーシステム；CSS）へと大きく変化している。これに伴って、ソフトウェアの開発手法も、ウォーターフォール型、スパイラル型、プロトタイプ型などの従来の手法に加えて、アジャイル（agile）<sup>24</sup>と呼ばれる手法も採られるようになってきている。

以上のようにソフトウェアの用途や使用環境等が変化し、開発手法が多様化しても、ソフトウェア開発工程（プロセス）<sup>25</sup>は、基本的には従来のものと変わらない。つまり、どのようなソフトウェアを開発したいのかという計画（ソフトウ

---

24) アジャイル（agile）は、もともと「俊敏な」「すばやい」という意味の英単語であるが、IT用語辞典では「IT業界では、経営環境の変化に迅速に対応できる柔軟な情報システムや、効率的なシステム開発手法などを指す。」とある。自社開発中心の米国で発展した開発手法である。わが国でも、Java アプリ（Java 言語で作成されたアプリケーション・ソフト。通常、このプログラムはインターネット・ブラウザで実行される）などの開発のために導入されたが、一般に委託開発対象である企業業務システムなどのソフトウェア開発には向かないので殆ど採用されず、開発手法としては、依然としてウォーターフォールが主流である。

25) ソフトウェア開発工程（プロセス）については、従来、わが国では各社各様の方法・用語が使われていたので、これを国際規格（ISO/IEC12207:2008）に適合させて標準化したものがIPA（独立行政法人情報処理推進機構）にて策定された「共通フレーム」である。この「共通フレーム 2013」では、大きくは「合意プロセス」の次に「テクニカルプロセス」を置き、この「テクニカルプロセス」を「企画・要件定義の視点」と「開発・保守の視点」とに分けている。そして、「企画・要件定義の視点」としては、「企画プロセス」と「要件定義プロセス」を置き、「開発・保守の視点」としては「システム開発プロセス」、「ソフトウェア実装プロセス」、「ハードウェア実装プロセス」及び「保守プロセス」を置いている。このうち一般にソフトウェア開発の中心となる作業は「ソフトウェア実装プロセス」の部分であろうが、このプロセスは、更に、①「ソフトウェア実装プロセス開始の準備」、②「ソフトウェア要件定義」、③「ソフトウェア方式設計」、④「ソフトウェア詳細設計」、⑤「ソフトウェア構築」、⑥「ソフトウェア結合」、⑦「ソフトウェア適格性確認テスト」、⑧「ソフトウェア導入」そして⑨「ソフトウェア受入れ支援」の9つのプロセスに細分されている。

ウェアの仕様及び作業スケジュールなど)を立てて、その計画に沿ってソフトウェアを設計・製作(最終的にはプログラムを作成しテストする)していくという一連の作業工程<sup>26)</sup>をとらざるを得ない。

特に、企業の基幹業務システムは一般に大規模システムとなっているため、このソフトウェア開発においては、一連の工程に従って様々な種類の作業が、長期間にわたって所定の順序で続いていく。そこで、あらかじめ定められたスケジュール通りに、所定の作業が進んでいるかなどの進捗状況を常時把握していかなければならない。そして、作業の進捗を阻害する要因(障害等)が生じた場合は、それを取り除くなど適切に対処して、所定の期限までにソフトウェア作業を完了させ、目的とするソフトウェアを完成させなければならない。

以上のような一連の工程で進められるソフトウェア開発の各種作業を所定のスケジュール(計画)通り進めていくためには、このプロジェクトの全体を通して作業を統制(コントロール)し、管理するという「プロジェクトマネジメント」は絶対に欠かせない。

### 3) 分散処理型システム開発とプロジェクトマネジメント

今日の企業の基幹業務システムは、Web対応の分散処理型のコンピュータシステム(分散処理型システム)が主流となっている。この分散処理型システムの代表的な方式が「クライアントサーバーシステム(Client Server System)」又は単に「CSS」<sup>27)</sup>と呼ばれる方式である。

26) ソフトウェア開発における実際の作業は、一般に「要件定義」に基づいて外部設計(基本設計)し、以後、内部設計(機能設計)、プログラム設計(詳細設計)、「プログラミング(コーディング及び単体テスト)」の工程(プロセス)の順で各工程の作業が進められ、当該ソフトウェア(プログラムの集合体)として具体化されていく。この各工程(作業)を、川の流れるに例えて、外部設計、内部設計を上流工程といい、具体的にプログラムを製作する「プログラミング」を下流工程という場合がある。また、上流工程の成果が「基本仕様書(外部設計書)」や「機能仕様書(内部設計書)」「詳細仕様書(プログラム設計書)」などと呼ばれるものである。

27) CSSとは、「分散型コンピュータシステムの一つ。プリンタ、モデムなどのハードウェア資源や、アプリケーションソフト、データベースなどの情報資源を集中管理する「サーバー」と呼ばれるコンピュータと、サーバーの管理する資源を利用するコンピュータ(クライアントと呼ばれる)が接続されたコンピュータネットワークのこと」(IT用語辞典〈e-Words〉)である。

このCSSにおいて実行されるソフトウェアを実際に使用する者（クライアント）は、当該企業の末端（現場）で仕事している多数の従業員等であり、これらの者は、必ずしも必要かつ十分なITの知見を持っているとは限らない。

このように多数の者がクライアントとして使用するCSS用のソフトウェアは、所定の機能を備えることは当然であるが、それが“良いソフトウェア”として高い評価を受けられるための重要なポイントは、そのソフトウェアを“誰（ITの知見がない者）でも簡単に使う（操作する）ことができるか”ということだろう。このようなソフトウェアの操作容易性は、一般に多数のクライアントが当該システムを使用する際に必ず見ることになる「ディスプレイ（入出力）画面」等のGUI<sup>28)</sup>に依存することが多い。従って、CSS用ソフトウェアの開発においては、実際の使用者（クライアント）となる末端（現場）の従業員等による操作性（例えば、従業員等が容易に操作できるようにした入出力画面）などに配慮した仕様（specifications）となっていなければならない。

ソフトウェア（プログラム）の仕様の確定は、一連の開発プロセス（工程）の中では上流工程の「外部設計」や「内部設計」の段階で行われる（この作業の成果は、一般に「基本仕様書（基本設計書）」、「機能仕様書」、「詳細仕様書（プログラム仕様書）」として具体的に細部まで確定されていく）が、所定の仕様通りとなっているかどうかの最終的な確認は、下流工程の「プログラミング及びテスト」の段階にならないと実施できない。つまり、この工程において、具体的な形としてプログラムが完成するので、このプログラムを実際にコンピュータで実行（テストラン）することで、仕様通りかどうかの確認ができるからである。この確認作業について、入出力画面（プログラム）の確認を例にとると、その画面は見やすいデザインとなっているか、必要なデータ項目に洩れはないか、入力データに対する処理出力結果には間違いがないかなど様々な事項があるが、実際には、当該入出力プログラムをクライアント（末端のコンピュータ）の画面を見ながら操作する形で確認を行うことになる。この確認作業は、当該プログラムを実際に作

---

28) GUI（「Graphical User Interface」の略）とは、「コンピュータやソフトウェアが利用者に情報を提示したり操作を受け付けたりする方法（ユーザインターフェース）の類型の一つで、情報の提示に画像や図形を多用し、基礎的な操作の大半を画面上の位置の指示により行うことができるような手法のこと」（IT用語辞典〈e-Words〉）である。

成した技術者（プログラマー等）が行うよりも、当該仕様を要求した者（クライアントとして当該プログラムを使用する者）が行う方が、見落としなく確実に確認することができる場合もある。

一般に、CSS などの分散処理型システム（ソフトウェア）は、クライアント用プログラム、サーバー用プログラム、通信用プログラムなど多数の多種多様なプログラム群で構成されている大規模システムである。また、このシステムを構成する多種多様なプログラムは、全てを新規開発（オーダーメイド）するのではなく、その大部分は、既存システムを構成した同種プログラムの修正やパッケージソフトのカスタマイズなどによって当てられることが多い。いずれにせよソフトウェア開発は、当該システムを構成する多種多様なプログラム群の個々のプログラムを、新規作成だけでなく、修正やカスタマイズなどによって完成させ、これらのプログラムを組み合わせ、あるいは結合して全体システムとして完成させるものである。

以上の通り、今日の主流となっている分散処理型システム（ソフトウェア）の開発は、一般に大規模システムであるので、一連の工程に従って様々な作業が行われることになる。すると、これらの諸作業が予定通り進められているか、阻害要因はないかなど全体の作業を統制しながら管理していくこと、すなわちプロジェクトマネジメントが欠かせないことになる。

#### 4) 多段階契約とプロジェクトマネジメント

わが国では、殆どのユーザーが、自社内でソフトウェア開発するために必要かつ十分な能力（ソフトウェア開発技術力等）を有していない。そこで、このようなユーザーが新たにソフトウェアを開発しようとする場合は、一般に、適切な（当該ソフトウェアの開発能力を有する）ベンダーを選定し、そのベンダーに開発委託せざるを得ない。この場合、ユーザーはベンダーに「このような内容のソフトウェアを開発したい」という要求事項を正確に伝えなければ、ベンダーは、その要求をきちんと反映した内容のソフトウェアを開発することができない。

ソフトウェア委託開発の取引実務では、ベンダーに対して、ユーザーが開発してもらいたいソフトウェアの内容等についての要求事項を取りまとめた書類として「RFP」<sup>29)</sup>が作成されることが求められる。この RFP は、通常、大規模で技術的

に難易度が高いソフトウェアの委託開発を行う際、ユーザーが複数のベンダーから提案書や概算見積書を提出してもらうための前提として作成されるものである。従って、ユーザーによる RFP の作成作業は、共通フレームのプロセスでいうと、「開発プロセス」に入る前の段階（企画・要件定義プロセス）で行われるべきものである。

しかし、ソフトウェア開発をベンダーに委託するユーザー（特に、一連のソフトウェア開発工程の全体作業を丸投げしてくるユーザー）は、一般的に、ソフトウェアに関する専門的知識等を殆ど有していないこと（いわゆる“素人”）が多い。このようなユーザーは、単独では RFP を満足に作れないから、ユーザーが要求する内容を十分かつ正確にベンダー側に伝えることができず、結果的には、そのプロジェクトは失敗に終わる可能性が高くなる。

このような背景を受けて、ソフトウェア委託開発の取引実務では、従来から委託するソフトウェア開発作業（委託業務）全体を包括する内容の「基本契約書」を締結したうえで、開発工程（プロセス）を念頭に置いて、例えば、設計業務、プログラム作成業務のように委託業務を段階的に区分<sup>30)</sup>して、それぞれの業務毎に個別契約を締結していくという方式の「多段階契約」<sup>31)</sup>が行われている。これ

---

29) RFP（「Request for proposal」の略）は、「提案要求仕様書」とも呼ばれている。一般に、ベンダーがユーザーに提出する提案書や見積書を作成するための前提となる書類である。つまり、開発目的のソフトウェア内容について、ユーザーの要求事項が記述された書類であり、通常、この RFP の提示を受けたベンダーは、その要求事項を満たした内容のソフトウェアを開発するとしたら、こういう条件で、このくらいの開発期間・費用がかかるなどを取りまとめた「提案書 (Proposal)」や「概算見積書」をユーザーに提出する。一般に、ユーザーは、複数ベンダーから提出された提案書等の内容を比較検討したうえで委託先（ベンダー）の選定を行い、当該ベンダーに開発の委託をすることになる。

30) 「多段階契約」により、ユーザーが新基幹業務システムの開発を委託する場合の業務区分としては、第 1 段階として「現行システムの問題分析から新システムについての RFP 作成までの業務」、第 2 段階として「当該 RFP 内容を満たした設計仕様書の作成業務」、第 3 段階「当該設計仕様書に基づきプログラム作成からテストまでの業務」などが考えられる。

31) 内布光「ソフトウェア開発委託取引の適正化に関する一考察」現代法学 第 17 号 2009 年 2 月、125 頁では、この多段階契約（モデル契約）についても考察している。古くは（社）情報サービス産業協会（JISA）が平成 14 年に公表した「ソフトウェア開発委託モデル契約書（JISA 新モデル契約書）」で、①システム仕様書作成業務と②ソフトウ

は大きなビルを新築する場合、そのビルの設計を設計業者に委託し、その成果である設計図を基にした建築工事を施工業者に委託して、ビルを建築することに似ている。この多段階契約では、ユーザーが委託する相手方は、それぞれの段階ごとに当該業務を得意とする別のベンダーにすることもできる。

いずれにせよ、ソフトウェア委託開発において最大の関心事は、最終的に、ベンダーがユーザーの要求通りのソフトウェアを完成できるかどうかである。しかも、これが判明するのは「開発」プロセスの下流工程で行われる「プログラミング」後の「テスト」の段階まで作業が進まないといわれない。このソフトウェア開発の特殊性（欠点）を考慮すると、委託開発においては、一定の段階ごとに成果の確認を行いながら、次の段階の作業に入る多段階契約は、この欠点をカバーできる契約方式といえる。

そして、ソフトウェア委託開発におけるプロジェクトマネジメントは、通常、当該ソフトウェア全体の開発を一括して請け負ったベンダーが行うことになるが、多段階契約で委託開発を行う場合は、これとは別の方法も考えられる。例えば、大きなビルの建築においては、設計を委託した業者とは別の施工業者にビルの建築工事を委託することがある。この場合、設計図通りに建築工事が行われているかについての「監理」業務を設計した業者に委託することが行われている。これに倣って、ソフトウェア開発プロセスにおける上流工程の“設計”に係る業務と下流工程の“製作（プログラミング、テスト）”に係る業務とに分けて、先ず“設計”業務をA業者に委託して、これが完了した後に、“製作”業務をB業者に委託するという方法である。この方法でソフトウェア開発を行う場合、A業者に対して開発プロジェクト全体に対する「監理」業務も委託することで、当該プロジェクトの成功確率が高まる可能性がある。

## 5) ソフトウェア委託開発におけるユーザーの役割

通常（請負型）のソフトウェアの委託開発においては、ベンダーは、所定の期

---

ウェア作成業務の2つの個別業務に分けて、段階的に個別業務についての契約を結ぶ方法を提唱した。その後、平成19年に経済産業省が「情報システム・モデル取引・契約書（第一版）」を公表したが、この契約書では①要件定義、②外部設計、③ソフトウェア開発、④ソフトウェア運用準備・移行の4個別業務に区分している。

限までに、ユーザーが要求したソフトウェア開発を完了（＝ソフトウェアを完成）させ、当該ソフトウェアをユーザーに引き渡さなければならない。この場合、ベンダーは、この仕事（ソフトウェア開発）を完遂させるための体制（いわゆる「プロジェクト推進体制」）を確立する等の準備万端を整えた上で、開発作業に着手し、当該プロジェクト全体についてプロジェクトマネジメントを行うことになる。しかし、ベンダーは、このプロジェクトマネジメントを独力で行うことはできない。つまり、ソフトウェア開発作業の計画・遂行自体は、ベンダー主導で行わなければならないが、ユーザーの業務処理用ソフトウェアを開発するという性質上、ユーザーからの必要情報の提供やユーザーの協力（例えば、入出力画面はユーザーが要求した仕様とおりになっているかの確認<sup>32)</sup>などが絶対に欠かせないからである。

ベンダーにソフトウェア開発を委託するユーザーは、“ベンダーにすべてを任せた”という意識から、当該ソフトウェア開発（プロジェクト）への参画が消極的になりがちである。しかし、“このプロジェクトはベンダーとの共同作業”というくらいの認識を持つべきである。このためにはユーザー自身も、当該プロジェクトの進捗状況などを把握するために定期的にベンダー側と十分な意思疎通を図ることができるような体制を整備するなど、当該ソフトウェア開発（プロジェクト）の推進に積極的に参画することが大切である。

もちろん、中には、開発委託に際して簡単な内容のRFPすら作れず、ベンダーが作成した提案書の内容も満足に理解できないユーザーがいるかもしれない。つまり、ユーザーによっては、その有するソフトウェア開発に関する知識や経験（いわゆる専門的知見）には大きな差がある。このようなユーザーの事情は、ベンダーがプロジェクトマネジメントの一環としてユーザーに協力等の要請をするときに大きく影響する。

そこで、ベンダーとしても、後日、ユーザーから必要な協力等が得られるよう

---

32) (社)情報サービス産業協会 法的問題委員会契約部会編 (JISA)「新しいソフトウェア開発委託取引の契約実務」商事法務、2002年、52頁では、今日のソフトウェア開発委託取引の主流であるWeb対応・分散処理型のシステム開発においては、ユーザーの作業参画が必須とされているが、特に、ユーザーの作業としては“仕様の確定”と“検取”が重要としている。

にするため、受託前の提案書の内容説明の段階から、当該プロジェクトに関連する情報を適宜提供し、特に、ソフトウェアに関して専門的知見を殆ど有していないユーザーに対しては、できるだけ専門用語を用いず、わかりやすい説明に徹するなど、ユーザーの理解が得られるように努めるべきである。

## 4. ソフトウェア委託開発におけるベンダーの義務

### 1) ソフトウェア委託開発の契約類型

「プロジェクトマネジメント」の言葉自体は、かなり古くからソフトウェア開発の現場（ソフトウェア技術者間）では普通に使われてきた。

このプロジェクトマネジメントが法的紛争の争点として問題になるのは、一般に、ソフトウェア開発を外部に委託する場合（いわゆる「外作＝アウトソーシング」の場合）だけである。つまり、ソフトウェア開発を自社内で行う場合（いわゆる「内作」の場合）は、仮に、当該ソフトウェアが予定通りに完成しなかった（この結果、開発費用が大幅に予算オーバーした）としても、それ自体は、当該企業内部の問題（このプロジェクトマネージャなどの関係者に対する評価等の問題）であるにすぎない。

わが国において、ユーザーがソフトウェア開発の全部又は一部を外部（ベンダー）に委託する場合の契約類型は、請負型、委任型（準委任）<sup>33)</sup>及び派遣型<sup>34)</sup>の3類型に大別できる。それぞれの契約の特徴を比較すると、下表（表2.）のような

33) ソフトウェア開発の委託取引契約の内容は、民法の典型契約である請負（民法632条）や委任（民法632条）にぴったり当てはまる場合もあるが、一般に、これらの契約を中心に、その他の契約の性質を併せ持った契約（混合契約）が多い。従って、本稿では、契約内容が請負に近い性質の契約を「請負型」、委任（ソフトウェア開発は法律行為でないので、正確には「準委任」）に近い性質の契約を「委任型」の言葉を使っている。

34) ソフトウェア委託開発の契約類型のうち請負型及び委任型は、それぞれ民法の典型契約である請負契約及び準委任契約を基本としているが、派遣型は、「労働者派遣事業の適正な運営の確保及び派遣労働者の保護等に関する法律（通称；労働者派遣法）」に基づき結ばれる契約である。この契約では、ユーザーが、自社従業員と同様に指揮命令して派遣労働者を使用することができるが、一方では使用者責任を負わなければならない。従って、派遣契約は、大手ベンダーが、自社従業員だけでは必要となるソフトウェア開発要員に不足が生じた場合に、中小（いわゆる下請け）ベンダーとの間で結ばれることが多い。

表 2. ソフトウェア委託開発契約の類型

契約類型の区分 (根拠法)	請負型 (民法 632 条)	委任型 (民法 656 条)	派遣型 (労働者派遣法)
契約の目的	仕事の完成 (開発完了)	事務委託 (開発技術の指導)	技術者供給
開発の主導権	ベンダー	ユーザー	ユーザー
ベンダー側の 義務・責任	完成責任／開発済 ソフトウェアの引渡	善管注意	技術者供給

相違点がある。

これらの3種類のいずれの契約で取引するかは、一般に委託者であるユーザー側の事情に依存する。つまり、ユーザーが有するソフトウェア開発についての専門的知見（経験や技術など）の程度、保有設備環境などに依存する。例えば、ソフトウェア開発について豊富な経験や高度な技術等を有しているユーザーは、ユーザー自身が主導的にソフトウェア開発を推進できる。そこで、ユーザーは、当該ソフトウェア開発に不足するものだけをアウトソーシング（外部委託）すればよい。例えば、ある分野の技術が不足する場合は、当該技術を補完するために外部委託すればよく、これに適した委任型や派遣型の契約で委託取引が行われることになる。

これとは逆のユーザー、すなわちソフトウェアに関する技術的知見を殆ど有さないユーザーは、ソフトウェア開発の全てをベンダーに依存（いわば丸投げ）せざるを得ない。この場合は、請負型の契約で委託取引が行われるが、特に、開発プロセスの初めから終わりまでの作業を丸投げするような内容の契約は、一般に「一括請負契約」と呼ばれている。

## 2) ベンダーが負うべき契約上の義務

前項の表 2. からわかるように、ソフトウェア開発の委託取引をどの契約類型で行うかによって、当該ソフトウェア開発（プロジェクト）についての主導権を、ユーザーあるいはベンダーのどちらが握るかが決まる。

一般にプロジェクトマネジメントは、一連のソフトウェア開発作業を主導的に

進める者が行うことになる。そして、ソフトウェア開発委託取引において法的紛争が生じやすいのは、一般に「請負型」(特に、ソフトウェア開発を丸投げする一括請負契約)で契約した場合である。この場合は、ベンダーが当該ソフトウェア開発の主導権を持つことになる半面、ベンダーの義務・責任は重くなる。特に、ベンダーが、ソフトウェア開発を一括請負契約(丸投げ)で受けた場合に当該プロジェクトを成功に導くためには、どうしても当該プロジェクトの全工程におけるマネジメントを確実にしていかなければならない。

これに対して委任型の委託取引契約では、ユーザーが当該ソフトウェア開発について主導権を持つ場合が多い。この場合は、ユーザー自身が当該プロジェクトについてマネジメントを行うので、ベンダーとの間でプロジェクトマネジメントに関する法的紛争が生じる余地は殆ど生じない。

なお、派遣型(労働者派遣契約)は、労働者派遣法に基づく契約であり、請負や準委任とは全く異質である。この契約では、派遣労働者(自社従業員でない)に対する使用者責任などをユーザーが直接負わされる。そこでユーザーは、一般に派遣型のソフトウェア開発委託取引を避けがちであり、派遣型での委託は、ソフトウェアを自社開発(内作)する際に必要な技術者が不足する場合に、当該技術者を派遣型にするときなど、かなり限定された場合に限られる。

このようにソフトウェア開発委託取引は、一般に、請負型と委任型に区分できるが、表2. に示した通り、請負、委任の契約の違いによってベンダーの義務や責任には雲泥の差が生じる。そこで、ベンダーにとって、ソフトウェア開発について完成責任を負わないという準委任(委任型)が有利となるので、ベンダーは、できるだけこの契約で受託しようとする。一般に「コンサルテーション契約」や「システムエンジニアリング契約」は、通常、その業務内容から委任型に区分されているが、実務上、一番多い「業務委託契約」として交わされた契約は、その委託業務の範囲・内容などから、請負、委任のどちらの性質に近いかが判断されることになる<sup>35)</sup>。

35) 東京地裁平成22年9月21日判決(判例タイムズ1349号136頁)は、「システム開発のために必要な業務分析等を委託業務内容とするコンサルテーション契約について準委任契約であるとしても請負契約の要素を含む」と判示した。

### 3) 開発作業におけるプロジェクトマネジメントの意義

ソフトウェア開発の委託を請負型の取引契約で行う場合、特に、ユーザーがソフトウェア開発をベンダーに丸投げする「一括請負契約」で行う場合は、ベンダーは当該開発プロジェクトの主導権を完全に握ることになる。このような契約で委託開発が行われると、請負契約は「仕事の完成」を目的とする契約であるから、請負人であるベンダーには、「仕事の完成」である当該ソフトウェアの開発完了（完成）が義務付けられる。

そこで、ベンダーは、契約目的を達成（目的とするソフトウェアを開発完了）するため、一連の開発プロセス（工程）に従って所定のスケジュールで行われる各作業について、進捗状況の把握・統制、管理すること（プロジェクトマネジメント）が不可欠である。つまり、請負型のソフトウェア委託開発においては、プロジェクトマネジメント義務がベンダーに負わされることになる。

しかし一方では、このようにベンダーが主導的に進めていくソフトウェア開発においても、所定の作業の中には、どうしてもユーザーからの協力等がなければ先に進められない作業等も出てくる。このような場合、ベンダーはユーザーに対して当該協力等を要請せざるを得ない。

ベンダーから、このような協力等の要請がなされたとしても、特に、ソフトウェア開発をベンダーに丸投げしているユーザーの中には、ソフトウェア開発についての専門的知見を有しないユーザーも多いので、これらのユーザーからは「そちら（ベンダー）はプロなので、そちらの方で適当にやってくれ」と非協力であったり、そのプロジェクトへの参画に消極的であったりする。ユーザーが所定の役割分担に従って協力してくれないと、開発作業そのものがストップすることもあり、作業スケジュールにも大きな狂いが生じる。

この結果、これまでのソフトウェア開発関連訴訟では、ユーザー側から「注文したシステムが期限までに納品されなかった。納品されたが内容が注文通りでなかった。」<sup>36)</sup>と主張されることが多い。

そこで、ベンダーは、ユーザーからソフトウェア委託開発を受ける際に、「本ソフトウェア開発では、ユーザーとの共同作業が欠かせず、必要な資料等の提供は

36) コンピュータ訴訟研究会「コンピュータ紛争Ⅱケース研究と予防対策」200年11月、尚文社、230頁

もとより、様々な協力等をお願いすることがあります」<sup>37)</sup>ということを受託条件の一つとして、ユーザーの同意を得ておかなければならない。

そして、ユーザーからの協力等を円滑に得るための前提として、ベンダー・ユーザー間で様々な連絡や協議等をしやすくする体制（例えば、窓口責任者や連絡協議会の設置など）等を双方で確立しておくことが不可欠となる。この連絡協議体制をユーザー側も確立させるために、委託取引契約書には、ユーザーの義務として盛り込むべきである。このような体制等が整備されることで、互いの意思疎通が密となり、当該ソフトウェア開発に必要な情報をお互いに共有できることになる。

以上の通り、請負型のソフトウェア委託開発（特に一括請負契約）では、一般にベンダー主導で行われるので、プロジェクトマネジメント義務は、第一義にはベンダーが負うべきであろう。しかし、ソフトウェア開発作業は、いわば両者の共同作業という性質を持つことから、従来から判例<sup>38)</sup>でも、ユーザーにも協力義務等の相応の義務があることが認めている。このことをユーザーに認識してもらうためにも、委託取引契約書にユーザーの義務として明記すべきといえる。

#### 4) ソフトウェア委託開発を巡る法的紛争の背景

ソフトウェア委託開発におけるプロジェクトマネジメントを巡り裁判で争われるような法的紛争が発生する背景としては、ソフトウェア委託開発の特性として、所定期限までにユーザーが要求する内容（当初計画した仕様通り）のソフトウェアを開発する（完成させる）ことが困難であるからである。特に、ベンダーにとって、これまでに開発したことがないような（同種内容のソフトウェア開発経験が殆どない）ソフトウェアの開発、つまり、これまでに蓄積した開発技術や経験が活用できないような内容のソフトウェアの開発においては、この開発失敗のリスクは増大する。

37) 前掲注36)（「コンピュータ紛争IIケース研究と予防対策」）でも「開発はユーザーとベンダーの共同作業であることを認識する」としている。

38) 東京地裁平成9年9月24日判決（平成6年（ワ）第8866号）判例タイムズ967号168頁。この判決は買主がコンピュータ（ソフト含む）を導入する以上、買主にも「自ら積極的に売主との打ち合わせに応じ、売主に協力すべき信義則上の義務がある。」と判示した。

そして、ユーザー自身が自社内でソフトウェア開発を行う場合（部分的に不足する技術・技術者等を準委任契約や派遣契約で外部調達する場合を含む）には、このような事態（ソフトウェア開発の未完成）が生じたとしても、それがすぐに裁判等により表沙汰になることは殆どない。しかし、ユーザーがベンダーにソフトウェア開発を委託した場合、特に、請負型で丸投げした場合には、このようなソフトウェアが完成できないという事態に陥ると、形式上、ベンダーは、ユーザーから債務不履行責任等を問われて（法的紛争となり）裁判等により表沙汰になりやすい。このことは、過去に争われた裁判における請求原因（主要争点）を見ても、債務不履行ないしは瑕疵担保責任に関する事例が大半を占めていることからわかる<sup>39)</sup>。

ところで、日本人は、もともと“裁判（訴訟）嫌い”という国民性があり、わが国の企業も、法的紛争が生じたとしても、裁判で企業情報が公開されるのを嫌い、できるだけ裁判まで発展しないように“協議解決”しようと努力する。従って、裁判で争われるような事件（法的紛争）は、当事者間で協議解決できなかったものである。つまり、裁判まで発展した事件は、一般に事実関係が複雑で、しかも有力な証拠等が殆どなく立証が困難である場合が多い。特に、ソフトウェア委託開発を巡って裁判まで発展した法的紛争（事件）を見ると、そこに至った原因は多種多様であり、しかもこれらは複雑に絡み合っているため、一般に、ソフトウェア委託開発を巡る裁判の審理が難しく長引くのは、ここに由来している。

例えば、債務不履行の構成要件である帰責事由を例にとっても、ベンダー側だけに帰責事由がある場合は少なく、ユーザー側にある場合も多い。このようにユーザー側にも帰責事由が生じる背景としては、一言でいえば、ユーザーがソフトウェア開発に関する専門的知識・経験を殆ど有していないことに起因している。特に、Web 対応・分散処理型のソフトウェアの委託開発における仕様確定や検取などの重要な作業においては、ユーザーの参画が必須とされる。このユーザーによるソフトウェア開発作業への積極的な参画が得られないと、それ以降の作業にも入ることができず、所定の期限までに開発完了することができない（形式上は

39) 前掲 内布「ソフトウェア開発委託契約紛争事例の研究（1）」現代法学 第10号 179頁では、裁判事例26件中18件が債務不履行又は瑕疵担保責任に関して主要争点であると分析している。

ベンダーの債務不履行) という結果を招くことになる。

## 5. プロジェクトマネジメント義務に関する判例動向

ソフトウェア開発作業で行われているプロジェクトマネジメントは、判例上、どのような内容で捉えられているのであろうか。ベンダーが負うべき義務として“プロジェクトマネジメント義務”という言葉が、判例上、初めて使われたのは、東京地裁平成 16 年 3 月 10 日判決（以下、「健保組合システム開発事件」という。）<sup>40)</sup>においてである。その後、同地裁平成 24 年 3 月 29 日判決（以下、「スルガ銀行対日本 IBM 事件」<sup>41)</sup>という。）では、再度、このプロジェクトマネジメント義務が一大争点となった。

なお、このプロジェクトマネジメントに似た争点として、同地裁平成 21 年 12 月 4 日判決（「ジェイコム株式事件」<sup>42)</sup>）では、株式取引の誤発注を巡り証券取引所の取引参加者規定との関係で“市場管理者責任”が争われている。

本稿では、「健保組合システム開発事件」と「スルガ銀行対日本 IBM 事件」の二つの裁判で争われた“プロジェクトマネジメント義務”に焦点を当て、それぞれの判決においてプロジェクトマネジメントをどのように捉えているかを考察する。

---

40) 「健保組合システム開発事件」東京地裁平成 16 年 3 月 10 日判決、判例タイムズ 1211 号 129 頁

41) 「スルガ銀行対日本 IBM 事件」東京地裁平成 24 年 3 月 29 日判決（平成 20 年（ワ）第 35320 号／反訴；平成 20 年（ワ）第 24303 号、判例タイムズ 1405 号 254 号）、この控訴審は、東京高裁平成 25 年 9 月 26 日判決（平成 26 年（ネ）第 3612 号）、上告棄却；最高裁平成 27 年 7 月 8 日決定

42) 「ジェイコム株式事件」東京地裁平成 21 年 12 月 4 日判決（平成 18 年（ワ）23958 号）判例タイムズ 1322 号 149 頁。この控訴審は、東京高裁平成 25 年 7 月 24 日判決（平成 22 年（ネ）481 号ほか）判例タイムズ 1394 号 93 頁。この事案は、みずほ証券がジェイコム株式売り注文において株数と株価とを取り違えて誤発注した後、取消注文したが、東京証券取引所の株式売買システムの不具合により、取消処理がされなかったことなどから、400 億円超の売却損が生じたので、みずほ証券と東京証券取引所との間で、主として市場管理者としての責任について争われたものである。

## 1) 健保組合システム開発事件

### (1) 本件の概要

本件は、国民健康保険法第 17 条に基づき設立の許可を受け、労働組合を母体として設立された国民健康保険組合（注文者）が、開発業者（請負人）に対して委託した新システム（ソフトウェア）開発が完成できなかったことにより生じた。

国民健康保険組合は、従来のシステムが古くなったので、これに替わる新システムを開発することを計画し、開発業者との間で委託料 3 億 4650 万円のシステム開発委託契約を締結し、この委託料のうち 2 億 5200 万円を一種の前払いをした。しかし、所定の納入期限までに、開発業者は新システム開発を完成させることができなかった。

そこで、第一審（東京地裁）では、国民健康保険組合（原告）は、この契約を解除したうえで、開発業者（被告）に対して、支払済委託料（2 億 5200 万円）の返還とともに債務不履行による損害賠償金として約 3 億 5 千万円の支払いを求めた。（以上が本訴）

この反訴として、開発業者は、本件システムが納入期限までに完成しなかったのは国民健康保険組合が協力義務に違反したことが原因であるとして、委託契約の解除による報酬及び損害賠償請求権に基づく損害金約 5278 万円の支払いを求めた。

この判決では、本訴原告（国民健康保険組合）の主張の一部を認めて、本訴被告（開発業者）に対して過失相殺後の損害賠償金（1 億 1340 万円）及びこの遅延損害金の支払いを命じた。

### (2) プロジェクトマネジメント義務の内容

本件判決で注目すべき点は、第一に、受託者（開発業者）には、受託したシステム開発についてプロジェクトマネジメント義務があることを認め、そして第二に、委託者側（国民健康保険組合）にも、当該システム開発への協力義務があることを認めた点である。なお、民法 641 条に基づく請負契約の解除における損害賠償金<sup>43)</sup>については、委託者側に協力義務違反により 40% の過失があるとし

43) 民法 641 条は「請負人が仕事を完成しない間は、注文者は、いつでも損害を賠償して契約の解除をすることができる。」と定めているが、この場合の注文者の損害賠償の範

て過失相殺<sup>44)</sup>の類推適用を認めている。

このようにソフトウェア委託開発においては、ベンダーにはプロジェクトマネジメント義務があり、ユーザーにも協力義務があることが明確にされた。

ここでは、プロジェクトマネジメント義務だけに的を絞って、もう少し判決内容を詳しく見てみる。

本件判決では、プロジェクトマネジメント義務とは、次の①及び②の二つを合わせた義務としている。

- ①被告（開発業者）は、納入期限までにシステムを完成させるように契約書等において注文者の協力を提示した開発手順や開発手法、作業工程等に従って開発作業を進めるとともに、常に進捗状況を管理し、開発作業を阻害する要因の発見に努め、これに適切に対処すべき義務を負う。
- ②注文者（国民健康保険組合）のシステム開発へのかかわりについても適切に管理し、システム開発について専門的知識を有しない注文者によって開発作業を阻害する行為がされることのないよう注文者に働きかける義務を負っている。

このように本件判決では、以上の①の義務と②の義務とを合わせたものが「プロジェクトマネジメント義務」と定義しているが、①の義務は、従来（この判決が出される前）から、一般にソフトウェア開発の実務で理解されていたプロジェ

---

囲は、幾代通・広中俊雄編「新版注釈民法（16）債権（7）」177頁によると、一般に「債務不履行に基づく損害賠償の範囲と同じように民法416条の規定によるところの相当因果関係の範囲内における積極的損害及び消極的損害の賠償でなければならない」とされ、「請負人が、請負契約が解除されたことにより仕事を完成する義務を免れたために、費用の支出を節約することができ、また、未工事部分の仕事のために手配された労働力や材料を他に使用あるいは売却することによって、その対価を取得することができた場合には、損益相殺の原理により、これらを控除する必要があることはいうまでもない」とされている。

44) 過失相殺について、この判決では、「一切の事情を斟酌すれば、被告（開発業者）に生じた損害について6割の過失相殺（民法418条の類推適用）をするのが相当である。従って、被告が被った損害は3億4650万円であり、原告（国民健康保険組合）が賠償すべき金額は、その4割に相当する1億3860万円（3億4650万円×40%）であると認められる」とした。この結果、被告は原告から委託料として既に2億5200万円を受取っているから、1億1340万円（2億5200万円－1億3860万円）が過払いとなっているとして、被告にこの金額の支払いを命じたものである。

クトマネジメントの内容とほぼ同じである。そして、②の義務は、①の義務を完全に果たすためにはユーザーからの協力が不可欠とされているので、この協力を得るためには、ベンダーが当然に行うべきことを明確にただけである。従って、①の義務は、プロジェクトマネジメント義務の本来的内容であり、②の義務は、①に附随して当然に生じる内容といえる。

### (3) 本件判決の実務への影響

従来から契約実務上、ソフトウェア開発が請負型（特に一括請負契約）で委託された場合、ベンダーは、当該ソフトウェア開発の主導権を握ることになるので、そのプロジェクトマネジメント義務を当然に負い<sup>45)</sup>、同時にユーザーは、当該ソフトウェア開発が円滑に進められるように必要な「協力義務」を負うものとされてきた。本件判決は、このベンダーが負うべきプロジェクトマネジメント義務の具体的内容を明確にするとともに、ベンダーが負うべき「プロジェクトマネジメント義務」の内容として、ユーザーから協力を得るために、積極的にユーザーに働きかけなければならないという内容も含ませたことに大きな意義がある。

そこで、ベンダーは、ユーザーからの必要な協力を得てプロジェクトマネジメント義務を果たしていくための前提として、当該プロジェクトの委託を受ける際（委託契約交渉段階から）、当該ソフトウェア開発についてユーザーがどの程度の専門的知識・経験等を有しているか（＝知見レベル）を見極めておくことが大切である。そして、ベンダーは、ユーザーに対しては、その知見レベルに合わせて理解できるような平易な言葉を用いて接するなど、日頃から意思疎通を密にして、ユーザーが当該プロジェクトへ参画しやすい関係を樹立しておかなければならない。つまり、ベンダーは、当該ソフトウェア開発（プロジェクト）におけるユーザーの役割分担の重要性を理解させておくことが重要であり、その上で、適宜、

---

45) ベンダーがシステム開発を一括請負（いわば丸投げ）で請け負う場合は、ベンダーは、その開発の主導権を握るので、そのプロジェクトマネジメント義務を当然に負う。しかし、ユーザーが複数の業者に分けて分割発注する場合、すなわちベンダーが全体システム開発の一部だけを請負う場合には、全体システム開発の主導権はユーザーが握るので、このプロジェクトマネジメントの責任はユーザー自身が負わざるを得ない。2007年経済産業省が公表した「ソフトウェア開発委託基本モデル契約」第13条では、このことを明確に規定している。

ユーザーの準備期間を考慮しながら様々な協力要請（例えば、必要な資料の提供要請、連絡体制の整備要請など）等の働きかけをしなければならない。

このようなユーザーに対して必要な説明や要請などの働きかけをするというベンダーの義務は、もともとソフトウェア開発の専門事業者として当然に負うべき信義則上の義務であるという考えが従来から実務界ではされてきた<sup>46)</sup>。また、ユーザーの協力義務については、かなり古くから裁判所は（下級審ではあるが）、信義則上の義務として認めている<sup>47)</sup>が、本件判決では、これより更に踏み込んで、特に、ソフトウェア開発に関する専門的知見のレベルが低いユーザーが、開発作業を阻害することのないように働きかける義務を負うとし、プロジェクトマネジメント義務の一部分を構成する義務として組み入れた点が注目できる。

## 2) スルガ銀行対日本 IBM 事件

### (1) 本件の概要

最高裁は、今年（平成 27 年）7 月 8 日、本件の控訴審判決を不服とした上告を棄却・不受理する決定<sup>48)</sup>をした。これにより、ベンダー（日本 IBM）に対してユーザー（スルガ銀行）に約 42 億円の損害賠償の支払いを命じた控訴審判決<sup>49)</sup>が確定した。

この事件は、ベンダーが業界有数の企業であり、海外製パッケージソフトを利用（カスタマイズ）した金融基幹システムの開発プロジェクトが失敗に終わったことに端を発した事件であったので、社会的関心が高かったが、裁判の争点としては、第一審（東京地裁）、第二審（東京高裁）の審理を通して、ベンダーが負うべきプロジェクトマネジメント義務の適用範囲が明確にされた。

本件の概要は、以下の通りである。

---

46) 前掲注 31) の JISA「新しいソフトウェア開発委託取引の契約と実務」P 147 でも、「債務には、何らかの信義則上の附随義務が伴うのが通例である。……当該プログラムが稼働すべき環境に制限がある場合は、助言義務が附随することもある。……瑕疵の存在についてユーザーに告知することが要求される情報提供義務が附随することもある。」とある。

47) 東京地判平成 9 年 9 月 24 日判決、平成 6 年（ワ）8866 号

48) 最高裁平成 27 年 7 月 8 日決定、平成 26 年（オ）201 号/26 年（受）261 号

49) 東京高裁平成 25 年 9 月 26 日判決、平成 20 年（ネ）3612 号

元来、ベンダーは、ユーザーの情報システムの管理・運用支援等を行っていたが、平成12年頃、ユーザーから次期基幹系システムである「新経営システム（以下「本件システム」という。）」の提案の依頼を受けた。なお、ユーザーは、ベンダーとは別の複数業者に対しても本件システムの提案を依頼していたが、これまでのベンダーとの関係から、ベンダーからの提案（海外ベンダー FIS 社の基幹パッケージソフト「Corebank」をベースに開発）の採用を決定した。

平成16年9月29日、ベンダーに対して本件システムの開発を総額95億円で委託する「基本合意書」を締結し、平成20年1月のサービスインに向けて、開発プロジェクトがスタートした。

しかしその後、計画・要件定義作業により明らかになる開発スコープ等を踏まえて締結する予定だった最終合意の内容をなかなか固めることができず、その合意時期が延期された。

平成17年9月30日に、この計画・要件定義の成果を待つことなく、両社は、「最終合意」を締結し、プロジェクトの基本的な運営に関する覚書を締結した。

その後の進捗は難航し、合意した開発日程通りに本件システムを完成させることができなかった。

平成19年4月、この状況を打破するため、ベンダーは、「Corebank」を他のパッケージソフトに変更する提案をした。ユーザーは、これに激怒し、ユーザーからのベンダーへの信頼が失われた。その後、損害賠償について両者協議するも進展しなかった。

結局、ユーザーは、ベンダーに対して、平成19年7月17日付け内容証明郵便により最終合意及びこれに関連する全ての個別契約を解除する通知をし、本件プロジェクトは中止された。

平成20年2月29日、ユーザーは、ベンダーに損害賠償等を求めて東京地裁に対して本訴を提訴した。

## (2) 裁判の経緯

本件裁判は、最高裁まで争われたので、裁判の経緯<sup>50)</sup>を整理すると以下の通り

50) スルガ銀行は、2015年7月9日付で「日本アイ・ピー・エム株式会社に対する損害賠償請求訴訟の上告審決定に関するお知らせ」と題して同社ホームページでニュースリ

である。

平成 20 年 2 月 29 日 ユーザーがベンダーに対して、ベンダーの契約違反（完成義務、プロジェクトマネジメント義務及び説明義務の違反）により開発プロジェクト失敗したことによる損害及び逸失利益など総額 115 億 8 千万円の損害賠償を求めて東京地裁に本訴を提起。これに対してベンダーは、未払い契約代金とその遅延損害金など総額約 125 億 5 千万円の支払い等を求めて反訴。

平成 24 年 3 月 29 日 東京地裁（第一審）は、以下の判決

ベンダーに対して、ユーザーに約 74 億 1300 万円の賠償金及び遅延損害金の支払いを命じ、ベンダーの反訴請求については全面的に棄却した。

〈これに対してベンダーが控訴〉

平成 25 年 9 月 26 日 東京高裁（控訴審）は、以下の判決

第一審判決を一部変更し、ベンダーに対して、ユーザーに約 41 億 7 千万円の賠償金及び遅延損害金の支払いを命じ、ベンダーの反訴請求は棄却した。

〈これに対してユーザー及びベンダーが上告及び上告受理申立〉

平成 27 年 7 月 8 日 最高裁判所は、ユーザー・ベンダー双方の上告を棄却し、両者の上告受理申立てを不受理とする決定をした。（控訴審判決が確定）

### (3) 本件プロジェクト失敗の背景

本件プロジェクトの失敗の原因は、ベンダーによる海外パッケージソフトのカスタマイズ（当該ソフトウェアを特定のユーザーの要求仕様に合わせて改変すること）ができなかったことに尽きる。ベンダーは、世界的に著名な IT 事業者であるので高度なソフトウェア開発力を有しており、かつ、長年にわたりユーザー

---

リースしており、ここに裁判の経緯が紹介されている。（URL; <http://www.surugabank.co.jp/surugabank/kojin/topics/150709.html> 参照）

情報システムの管理等を行っているので本件システムに必要な機能等についても熟知していると思われる。しかし、それでも本件開発プロジェクトは失敗した。つまり、本件は、パッケージソフトをカスタマイズすることが、どんなに困難な作業であるかを如実に証明している。

そして、本件のように有名なパッケージソフトの殆どは海外製であり、その海外パッケージソフトが、どんなに海外で実績があったとしても、日本でカスタマイズして使おうとすれば、その前提として日本語化（ローライゼーション）<sup>51)</sup>しなければならない。すると、開発の難易度が更に高くなる。

そもそもパッケージソフトは、不特定多数のユーザーが様々な用途に使えるように設計されているので、本来、カスタマイズに馴染まない。つまり、一切の改変等変更することなく、そのまま使うことを想定しているので、カスタマイズを許す場合であっても、その範囲は一部に限定され、各種制約が課せられることが多い。従って、パッケージソフトをカスタマイズすることができるのは、当該パッケージソフトの制作者及びその製作者から改変権を許諾され、かつ特別に技術指導を受けた者に限られる。

ベンダーは、本システムの中核ともいべき「Corebank」と邦銀の業務との間には大きな乖離があったので、大規模なカスタマイズが必要だということが分かっていた。しかし、ベンダーは、このパッケージソフトの改変権を得ていなかっただけでなく、その提供者である FIS 社を、当初から開発プロジェクトに参画させてもいなかった。そして、途中から参加させたが、ベンダーは FIS 社との利害調整をうまくマネジメントできず、役割分担の協議にも失敗した。なお、ベンダーは、プロジェクト開始前に、本件プロジェクトには以上のような固有なリスクがあることを、ユーザーに一切説明していなかった。

---

51) 今日のソフトウェア（プログラム）は、通常、画面を見ながら使用する。パッケージソフトは海外で開発されたものが多く、例えば、マイクロソフト社製品（Windows、Microsoft-Office など）は、もともと米国の本社で開発されている。このように外国で開発されたものはその国の言語の仕様となっているので、これを実際に使用する国の言語の仕様に変更することを「ローライゼーション（Localization）」といい、日本語仕様に変更（改変）することを「日本語化」と言っている。

#### (4) プロジェクトマネジメント義務の内容

本件裁判（第一審・控訴審）では、プロジェクトマネジメント義務の内容そのものについて争われておらず、前述の「健保組合システム開発事件」判決を受けて訴訟当事者双方が主張しているだけで、新たに大きな変更を加えていない。

ユーザー（第一審本訴原告）は、本件プロジェクトが中止するに至ったのは、ベンダー（第一審本诉被告）が、システム開発業者として納入期限までにシステムを完成させるようにユーザーに提示し、ユーザーとの間で合意された開発手順や開発手法、作業工程等に従って開発作業を進めると共に、常に進捗状況を管理し、開発作業を阻害する要因の発見に努め、これに適切に対処する義務（プロジェクトマネジメント義務）を負うのに、これに違反したからだと主張<sup>52)</sup>した。これに対して、ベンダーは、本件プロジェクトが中止に至ったのは、ユーザーの協力義務違反によるものであると主張した。

以上について、裁判所（第一審判決）は、プロジェクト開始前の企画・提案段階から当該パッケージソフトのカスタマイズ可否等の必要な検討・検証をせず、開発スタート以後幾つかの協議・合意の段階を経て最終合意書締結に至るまで経緯の中で、ベンダーがユーザーへの適切な（カスタマイズの困難性やリスクなどの）説明していない、カスタマイズ作業を適切に実施するための体制を整備していない、ベンダーが当該パッケージソフトを改変するために必要な役割分担、作業量・作業時間費用等に関してFIS社との十分な協議を整えていないなどの事情を挙げて、ベンダーはプロジェクトマネジメント義務に違反をしているとし、ベンダーが主張したユーザーの協力義務違反に対しては、これを認めるに足りる証拠がないなどとして否認した。

#### (5) プロジェクトマネジメント義務の範囲

これまで、プロジェクトマネジメント義務は、請負契約の完成義務に附随して派生する契約上の義務として捉えられてきた。

本件判決では、契約成立前（開発プロジェクト開始前）の企画・提案段階から、

---

52) この原告（ユーザー）の主張は、健保組合システム開発事件（東京地裁平成16年3月10日判決）で示されたプロジェクトマネジメント義務の内容のうち本来的義務と同じ内容である。

ベンダーは、プロジェクト成功のため必要とされるパッケージソフトのカスタマイズの検討・検証やカスタマイズの困難性・リスクなどのユーザーへの適切な説明なども行うべきであり、これもベンダーのプロジェクトマネジメント義務の範囲内に含めた<sup>53)</sup>。このような内容は、契約成立後に目的を達成する（仕事を完成し、プロジェクトを成功に導く）ために、契約成立前の提案段階で、IT 専門家としてのベンダーは当然に行うべき内容であるが、契約が成立していない以上、この義務に違反しても、すぐに契約違反を問うことはできない。

そして、控訴審判決では、この第一審判決のベンダーの義務を詳しく展開し、次のとおり明確にした。

第一に、ベンダーは、企画・提案段階において、自ら提案するシステムの機能、ユーザーのニーズに対する充足度、システムの開発手法、受注後の開発体制等を検討・検証し、そこから想定されるリスクについて、ユーザーに説明する義務がある。

第二に、ベンダーは、システム開発過程において、適宜得られた情報を集約・分析して、ベンダーとして通常求められる専門的知見を用いてシステム構築を進め、ユーザーに必要な説明をし、その了解を得ながら、適宜必要とされる修正、調整等を行いつつ、本件システム完成に向けた作業を行う義務を負う。

第三に、システム開発は当初想定した通り進むとは限らないので、当初とは異なる状況変化により、想定していた開発費用、開発スコープ、開発期間等に相当の修正を要すること、更には、その修正内容がユーザーの開発目的等に照らして許容限度を超える事態が生じることもあり得る。そこでベンダーは、そのような局面に応じて、ユーザーのシステム開発に伴うメリット、リスク等を考慮し、適切に開発状況の分析、開発計画の変更の要否とその内容、更には開発計画の中止の要否とその影響等について、ユーザーに説明する義務を負う。

控訴審判決では、ソフトウェア開発という特殊な契約関係におけるベンダーの

---

53) 本件の第一審判決と控訴審判決の双方とも、プロジェクトマネジメント義務の範囲の捉え方（開発着手前の企画・提案段階から生じる）や法的根拠（不法行為上の信義則に基づく義務）は同じである。しかし、この段階でのベンダーの義務違反を第一審では認め、控訴審ではベンダーの過失を認めなかったことなどから、賠償額では第一審の4割減という金額的に大きな差が出た。

義務の内容を、以上の通り、三つに分けて明らかにした。

このうち第二の義務の内容は、ベンダーが完成責任を全うするために当然に負うべきであり、本来的なプロジェクトマネジメント義務の内容とほぼ同じであるが、第一及び第三の義務内容は、第二の本来的な契約上の義務（プロジェクトマネジメント義務）に付随・付帯して派生する内容であり、契約当事者の信頼関係から必然的に導き出される義務として捉えることができる。

特に、第一の義務は、契約締結前の段階（企画・提案段階）における内容であるので、契約から生じる義務ではない。そこで、控訴審判決では、この義務を「契約段階に向かう交渉過程における信義則に基づく不法行為上の義務である」と位置づけた。

いずれにせよ、本件判決では、プロジェクトマネジメント義務及びこれに付随・付帯するベンダーの義務は、契約締結前の段階（企画・提案段階）からでも発生し得ることを明らかにしたといえる。

## 5) 小括

判例上、ベンダーのプロジェクトマネジメント義務が認められたのは、東京地裁平成 16 年 3 月 10 日判決（健保組合システム開発事件）が最初であるが、その後、東京地裁平成 24 年 3 月 29 日判決（スルガ銀行対日本 IBM 事件）により、システム開発着手前の企画・提案段階においてもベンダーはプロジェクトマネジメント義務（ユーザーへの説明義務）が及ぶことが明らかにされた。

この義務の本質的な内容は、請負契約の中核をなす完成義務から導き出されるものであるから、取引実務でも、「健保組合システム開発事件」判決の内容と同様に、ユーザーの協力義務とのセットで当然の義務として認識されてきた。ところが、スルガ銀行対日本 IBM 事件の控訴審（東京高裁平成 25 年 9 月 26 日）判決（その後、最高裁平成 27 年 7 月 8 日決定で確定）が、企画・提案段階においてユーザーへの説明義務があるとした第一審（東京地裁平成 24 年 3 月 29 日）判決を維持し、これを信義則に基づく不法行為上の義務と位置付けた。すでに、この判決の取引実務への影響等について議論がされているが<sup>54)</sup>、不法行為上の義務と位

54) 鼎談（三村量一・上山浩・桶田大介）「情報システムの開発・運用と法務」NBL1050 号 4 頁、座談会（大谷和子・竹内規・野々垣典男・松島淳也）「スルガ銀行対日本 IBM /

置付けたことには、従来からのプロジェクトマネジメント義務の本来的な内容から見ると多少違和感を禁じ得ない。

特に、企画・提案段階でのユーザーへの説明義務は、契約が成立しなければ問われない義務であるが、契約成立を前提にした義務であるから、「契約締結上の過失」の法理<sup>55)</sup>に似ている。相手方が契約成立前には予想できなかったことを成立後に告げることは、一種の「だまし討ち」と解釈できるので、控訴審判決では、これを「不法行為上の義務」と位置付けたのであろう。確かに、民法 709 条は、非常に広い要件で不法行為責任を認めているが、これまで「契約締結上の過失」も、契約責任の一環として認めてきているので<sup>56)</sup>、このような義務は、単に「信義則に基づく義務」と位置付けてもよいのではないだろうか。

## 5. おわりに

わが国企業の基幹業務情報システムが、現在では、Web 対応分散処理型のシステム（この典型が「クライアントサーバーシステム」）が主流となっても、依然として、当該システム用ソフトウェアは、パッケージソフト利用よりも独自開発ソフトウェア（カスタムソフト）に対するニーズが非常に高い。しかし、その殆どのユーザーは、ソフトウェア開発の能力を持たないので、独力で内作できず、ベンダーに委託開発（中には、企画・提案段階からの丸投げ委託）をせざるを得ない。すると、ユーザー・ベンダー間のソフトウェア開発に対する意識・知見等に大きなギャップが生じることが多いので、それだけにこの委託取引を巡ってトラブルが生じやすく、裁判沙汰となる可能性も高くなる。

---

みずほ証券対東京証券取引所 判決内容から実務への影響を探る」Business Law Journal 2014 年 1 月号 31 頁、など

55) この「契約締結上の過失」は、一方当事者の過失によって契約が原始的不能で無効であることを知らない相手方が、契約有効と信じて蒙った損害（信頼利益）を、過失ある当事者に賠償させることが公平に適うという考え方で、元々ドイツ法で認められていた法理。

56) 内田貴「民法Ⅲ [第 2 版] 債権総論・担保物権」東京大学出版会（2004 年 1 月）26 頁では、「契約締結上の過失」などという条文にない特別な法理を持ち出さなくても不法行為で処理できる。しかし、契約締結後とのバランスを考慮し、契約責任の一環として処理するために、このような法理が肯定されてきた。」としている。

これまでのソフトウェア開発関連訴訟を見ると、ベンダーの債務不履行責任や瑕疵担保責任などを争う事案が多かった。このうち債務不履行は、ベンダーが請負契約の完成義務に違反した結果(=プロジェクト失敗)であるが、この原因(帰責事由)は、必ずしも全てがベンダーにあるとは限らない。つまり、特に、Web対応分散処理型システムのソフトウェア委託開発における作業の特性上、ユーザーによる確認作業等の協力が欠かせないからである。そこで、従来からベンダーは、このプロジェクト失敗という最悪の結果を招かないようにするため、一連のソフトウェア開発プロセスにおいてプロジェクトマネジメントに努めているが、このプロジェクトマネジメントは、ユーザーからの協力があることを前提としている。

このプロジェクトマネジメント義務の内容について、裁判所が初めて明らかにしたのは、「健保組合システム開発事件」判決においてであるが、ソフトウェア開発の企画・提案段階から生じると、この義務の範囲を広げたのが「スルガ銀行対日本IBM事件」判決(第一審・控訴審)においてである。

特に、「スルガ銀行対日本IBM事件」は、世界有数のベンダーであっても、海外パッケージソフト利用のソフトウェア開発(カスタマイズ)がいかに困難であるかを如実に物語った典型的事件である。わが国では、従来から健保組合システム開発事件のようなカスタムソフト開発が中心であるので、海外パッケージソフトのカスタマイズを中心としたソフトウェア委託開発は、これまで殆ど行われてこなかった。なお、これまで海外ベンダー提供のERP(統合型基幹業務パッケージ)の導入も行われてきているが、一般にERPは、カスタマイズできる範囲がかなり狭く、しかも当該ERPの技術内容に精通した技術者だけがカスタマイズできるなどの大きな制約を課してライセンスされるので、これら制約条件を前提にして行うソフトウェア開発の失敗はほとんど生じない。

いずれにせよ、これまでの取引実務では、ベンダーが企画・提案段階でユーザーに対して適切な説明をすることは当然のこととされてきた。これは、ベンダーがユーザーから受注をするために行う提案内容(=受注条件)は、受注後、そっくりそのまま契約内容(条件)となることを前提にしているからである。なお、パッケージソフトを利用したソフトウェア開発は、カスタムソフトの開発に比べて、一般的に大幅なコストダウンや開発期間が短縮化できるなどのメリットがあ

るが、この反面、様々な制約があり、ユーザーの要求仕様を満たしにくい、カスタマイズ作業が難しいなどの大きなリスクを抱えていることも多い。

一般的に、ベンダー側に受注優先の意識が強くある場合、ユーザーにとって良いことだけしか提案（説明）せず、リスクやマイナス面は矮小化しがちである。このことからスルガ銀行対日本 IBM 事件が生じた背景を推測すると、日本 IBM にとってスルガ銀行は重要顧客であり、他社に奪われたくないという意識が強く働き、本件プロジェクトについてのマイナス面やリスクについての検討・説明が後回しになったであろう。

このように事件が生じる背景や経緯を見ると、事件ごとに独特な事件発生要因があることが多いので、プロジェクトマネジメント義務の内容や範囲を一律に決めつけることは避けるべきであり、ケース・バイ・ケースで判断すべきものといえる。

(完)