

コンピュータ・ソフトウェア取引のADR
ソフトウェア開発を巡る紛争の形態・原因・予防と対策

東京地方裁判所民事22部
調停委員 大石 完一

1. はじめに

IT関連技術は、著しく発展しているが、利用技術に関する一般の人々の知識は低い水準にあるがために、コンピュータの利用が広がるにつれて、IT技術の提供者と利用者間で紛争が生じて訴訟に発展するケースが増加してきているのが現状である。

これらの紛争は、依頼者が情報処理技術に疎いことからソフトハウスに任せっぱなしでシステム構築を依頼していたり、ソフトハウスは依頼者の業務内容を深く知らないまま開発に着手したために依頼者の意図するシステムに仕上がらなかったり、ベンチャービジネスとして発足したソフトハウスがビジネスの基本に疎いために不適切な契約を結び開発を進めて後になって問題が発生したり、などから訴訟に至るケースが多いのが現状である。

そこで、東京地方裁判所における調停手続きによるソフトウェア開発を巡る訴訟の事例概要、紛争に至った原因の分析及び紛争の予防と対策について紹介する。

2. ソフトウェア取引を巡る紛争の形態

ソフトウェア取引を巡る紛争は、次の形態が多い。

- * 納入されたソフトウェアの品質を巡る紛争
- * 納期を巡る紛争
- * システムの処理速度を巡る紛争

1) 納入されたソフトウェアの品質を巡る紛争

利用者の依頼を受けて受注者（ソフトハウス）が作成・納入したシステムについて利用者が意図した通りに仕上がっていないことから開発代金の支払いを拒否されたために開発代金の支払いを求めて訴訟となったケース、すでに支払った開発代金の返還を求めて訴訟となったケースなどが多くある。

当事者双方の主張を整理してみると、どのようなものを作るのかの基本点が明確化さ

れないまま開発に着手したために、完成されたシステムが依頼者の思惑と異なったものとなり、その対応を巡って紛争となったケースが多く見受けられる。

その原因として、何をどのように作るのかの仕様確認が不十分であったことが、第一に挙げられる。利用者の根底に、コンピュータのプロならば当方の意向をシステム化してくれて当然との思いがあり、情報処理の仕組みを理解しないまま発注するケースが目立つ。一方受注者側は、依頼者の業務の知識が乏しいまま依頼内容を自分なりに勝手に解釈して開発に着手したために作成したシステムが依頼者の求めていたものと異なってしまったケースも多く見受けられる。開発に先立ち受注者は、依頼者からの要求仕様の提示を受けて基本設計書を作成し、双方で要求仕様を満足していることの確認をしてから開発代金・納期等を記載した開発契約書を取り交わし開発に着手する必要がある。

第二に、依頼者と受注者との間での打ち合わせ内容が文書化されていないために、「言った・聞いていない」の紛争が多いことである。特に電話による仕様打合せ内容が議事録などの形で保管されていないケースが目立つ。これらの打合せ議事録は事後に双方の確認をとって文書化しておく必要がある。特にコンピュータのソフトウェアは、無形物であるとの認識の基にたって、見積書、契約書、議事録、覚書、要求仕様書、基本設計書、テスト計画書及びテスト結果等は、文書化しておくことが紛争を避ける上で重要となる。

第三に、仕上がったソフトウェアの動作確認が十分にされないまま納品したために、品質問題が生じたケース、納品されたソフトウェアを検証しないまま検収印を押したが重大な欠陥が後になって見つかり処置について問題が生じたケースなどが目立つ。

作成したソフトウェアをテストすることは、開発してきたプログラムの誤りを効率的に見つけ出し修正する手法の他に、委託されたソフトウェア開発の終了を確認するうえで重要な工程となる。即ち、このテストによって依頼先に成果物を納入し開発代金の請求が可能であることの確認上で重要な工程となる。紛争の中でよく見受けられる事項として、テストを実施するにあたって、ただ漫然とテストを行うケース、受託者が下請けに丸投げして自身では確認していないケースがある。特に大切なテストとしては、結合テストや総合テスト（例外処理テスト、限界テスト等）を要求仕様に基づいてソフトウェア開発終了条件を定めておいきテストすることが重要である。

紛争の中で依頼者に多く見受けられるケースとして、納入されたソフトウェアの検収

方法が挙げられる。紛争となった検収の事例として、検収テストをしないで検収印を捺したが重大な欠陥が後になって見付き対応について紛争となったケース、要求仕様と異なる検収テストによって不合格を宣言されたケース等がある。

2) 納期を巡る紛争

約束した納期になっても依頼したソフトウェアが完成しないことから計画した業務に支障をもたらしたとして損害賠償を請求する事件が多くある。

依頼者は「約束した期日までに依頼したものが仕上がらない。」「意図したシステムに仕上がっていないので作り直しを要求した結果、納期が大幅に遅れ、かつ、仕上がらなかった。」といった苦情に対して受注者は、「開発途中で仕様が頻繁に変わり、都度、作り直しが生じたので開発が予定通り進まなかった。」「仕様が中々決まらず、開発期間が定まらなかった。」「要求仕様どおりに作ったが、依頼者の強い要望ゆえに作り直した結果で落ち度はない。」と反論する。また、「仕様変更が相次ぎ予定工数が大幅に超過したので、開発代金の見直しを要望したが受入れられなかった。」ことから紛争に至るケースもある。

情報処理システムの場合、開発コストに占める人件費の割合は、きわめて高い。そのために、納期が予定通りに守られないと、赤字幅が増大の一途をたどることになるのは必然である。仕様変更によって発生するコストは、開発の工程が進むにしたがって増大する性質がある。したがって、ソフトウェア業界では、開発の途中で発生する仕様変更に対処しなければ命取りになるとも言われている重要な事項である。

最近のソフトウェア開発では、「クライアント・サーバー方式」を採用したシステム開発での紛争事件が増えてきている傾向にある。この方式では「プロトタイプ技法」を使って情報処理を開発することが多い(図1参照)。

「プロトタイプ技法」は、完全な形の要求仕様書や基本設計書を作らずに、システムやプログラムの試作版を作り、依頼者に対してパソコンの画面を見せ仕様を固めながらシステムを組上げていく手法で、コンピュータ・システムをよく理解していない依頼者にとって分かり易く、開発者にとっても依頼者の要求仕様を理解する上で有効な手段という特長があり、採用するソフトハウスが増えてきている。

しかし、反面、仕様が中々決まらず、だらだらと開発時間が長引き、契約納期内に仕上がらず、それが紛争の火種となっているケースが増えてきている。その上、ソフトハ

ウスにとっては、工数が増え利益が得られなくなるリスクがあることを知る必要がある。依頼者にとって納期は重要事項となるので、ソフトハウスは、理由の如何を問わず納期を厳守するよう工程管理する義務のあることに留意すべきである。

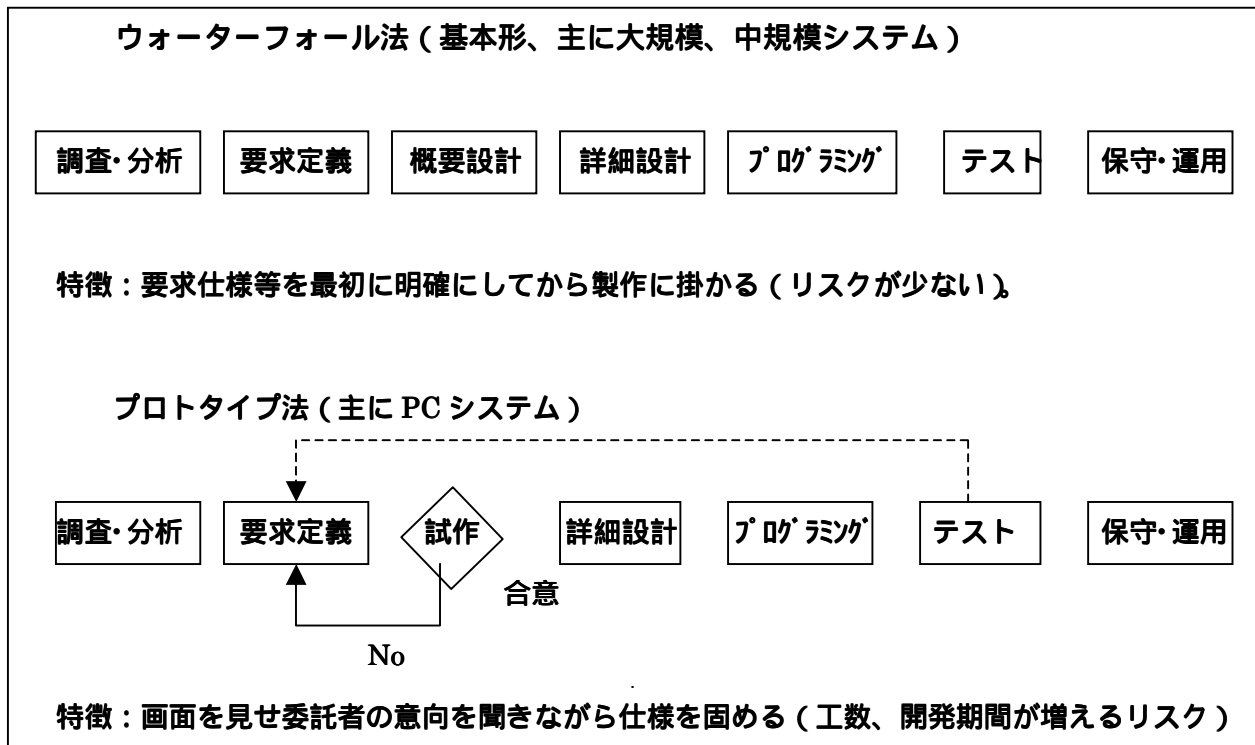


図1 ソフトウェア開発技法「ウォーターフォール」と「プロトタイプ」の比較

この原因は、一言で言えば、システム開発者の「プロジェクト・マネジメント」の不備に起因していることになる。これはソフトウェア開発にとって、「工程管理」を正しく行うことがきわめて重要な要素となる。約束期限内に要求仕様書どおりの機能を有するシステムに仕上げることが出来るか否かは、開発プロジェクトを管理するプロジェクト・マネージャーの能力にかかっている。

また、この任に当たる開発者側のプロジェクトマネージャーは、依頼者側との意思の疎通に隔たりのないよう努めることも重要な任務となる。依頼者は一般にコンピュータの知識に疎く、ソフトウェア開発者は依頼者の業務に疎いことから依頼者が要望するシステムを作り上げるためには、両者の意思の疎通を図ることもプロジェクトマネージャーの責務となる。

開発途上での仕様変更の原因として、依頼者側の関連部署間の要望の取り纏め方に起因

して発生するケースも見受けられる。このようになる原因としては、依頼者側のプロジェクトに対するマネジメントの不備が起因しているケースも多く見受けられる。

ソフトウェア開発を依頼する側にも紛争を起こさせないために、プロジェクトを遂行するマネージャーの資質が問われてくる。社内の関連部署との調整を図り要求仕様を明確にすること、開発導入スケジュールや開発者との役割分担を調整すること、納入されたソフトウェアが要求仕様に対しての仕上がり具合を見て速やかに検収すること、などが依頼者側のプロジェクトマネージャーの責務となる。

3) 構築したシステムの処理性能を巡る紛争

納入されたシステムを本番業務に移行したところ、処理速度が遅いとか異常終了して業務が遂行できなくなるなどの障害が出て、その対応を巡って紛争に至ったケースも多く見受けられる。

この種の紛争では、大半が受入れ検査（検収）は問題なく通過し運用段階になってから発見された問題が主体となる。開発したソフトハウスに改善を求めたが、一向に改善されないとして開発代金の返還や損害賠償の訴訟に至ったケース、開発時に予想したデータ量を大幅に上回っているためソフトウェアを作り直す必要があるとのソフトハウスの言い分から改善費用を請求されたがデータ量の見積もりの甘さはソフトハウス側にあるとして、開発代金の返還や損害賠償の訴訟に至ったケースなどである。

問題の主体は、処理データ量の予測の誤算やシステムテスト、受入れテストの甘さが原因となっている。これらの問題は、構築したソフトウェア自身の改善で解決する場合と使用しているハードウェアがらみの問題となる場合がある。いずれにせよ改善するためには多額の費用と時間を要することになる。処理データ量の予測誤算の責任が依頼者側にあったのか開発者側にあったのかが争点となる。依頼者側では、データ量によって処理速度が異なることを認識していなかったために要求仕様に記載せずソフトハウス側でも気に留めなかったケース、依頼者側の予測誤りのケースなど原因は様々である。いずれにせよ、一概に依頼者側の落ち度だけとはいえない。ソフトハウスは、プロという立場で、データ量によって処理性能がどのように変化するのかを予測して、依頼者に説明した上で要求仕様書や基本設計書を確定する義務があることに留意する必要がある。

また、コンピュータによる情報処理システムは、導入されれば永久に使用できると思

っているユーザが多い。導入後放任されたシステムは、日常発生する操作誤りや異常データの入力等によって発生するシステムの誤動作や異常終了に対処できなくなる。これを防ぐためには、社内に運用・保守体制を組織し異常発生のと適切な処置を施さなければならないことに留意する必要がある。

3 . 紛争事件発生の予防と対策

コンピュータ・ソフトウェア開発を巡る紛争を未然に防ぐための基本事項について整理する。

1) システム開発は依頼者と受注者との共同作業

システム開発を効率よく執り行うためには、「依頼者と受注者との共同開発で、双方がそれぞれの役割と責任を負う」という基本認識に立つことが重要となる。そのためには、双方にプロジェクトを管理遂行するプロジェクト・マネージャーを任命し、双方の連絡を密にして分担した役割を遂行する必要がある。プロジェクト・マネージャーの資質としては、社内でのプロジェクトに関する権限と実行力を持つことの他に、お互いの業務に関する基礎知識を持っていることが必要不可欠となる。依頼者側は情報処理に関する基礎知識を、受注者側は構築しようとする業務の基礎知識を持つことである。例えば、両者の話す言葉が異なる場合、お互いに話している内容が理解できないのでは意思疎通が図れないのと同じである。

最初の責務は、依頼者は構築して欲しい仕様（要求仕様）を明確に受注者に伝えることである。受注者は、依頼者からの要求仕様を正しく理解してシステム設計に反映しているかを基本設計書に纏め、依頼者の確認を取って業務契約に移行する必要がある。仕様が不明確のまま開発してしまうと前述のように紛争のもとになる。

紛争を起こさないためには、双方の打合せの全てを文書化し、双方で保管することも大切となる。

プロジェクト・マネージャーは、次の基本的な責務を負うべきである。

依頼者側：

- * 構築するシステムについて社内の関連部署との調整を的確に行う。
- * 構築するシステム内容を明確に示し、要求仕様を受注者に示すか協議して作成する。
- * 受注者に導入予定を明確に伝え、或いは協議して決定し、管理する。

- * システム開発から運用に至る双方の役割分担を明確にする。
- * 納品されたシステムは要求仕様に基づいて作成されたか否かを速やかに検収する。
- * 導入後のシステムの運用・保守体制を確立し、実行する。
- * その他。

受注者側：

- * 依頼者の意図を汲取る技量・知識を有する。
- * 依頼者と共に要求仕様を検討し基本設計書を作成し、依頼者の合意をとる。
- * 開発スケジュールに従った工程管理及び予算管理を実施する。
- * プロジェクトチームの適切な指揮・管理（下請けを使う場合、その管理を含む）
- * その他。

2) 情報処理システムの開発工程

情報処理システムの開発は、おおよそ次のような工程となる。

表1 情報処理システムの開発工程及び各工程の開発工程に占める割合

工程	開発内容	開発工程に占める割合
計画	現状調査・分析	30%
設計	要求仕様・基本設計・詳細設計	
制作	プログラム作成	20%
テスト	単体・結合・システムのテスト	40%
検収	検収テスト・システムの移行	
運用	システム運用・保守	

表1に示すように、仕様の取り決めから確認を経て設計に至る工程に占める割合は開発工程の30%に及び最も重要な段階となる。また、作成したシステムの検査から依頼者に納入しての検収・システム移行するまでの工程は40%にもおよぶ。作成したシステムのテストを厳密に行うことによって受注者のシステム完了を宣言できることになる重要な工程となる。なお、プログラム作成の開発工程に占める割合は20%にすぎない。

依頼者は、開発してもらうシステム要件を可能な限り詳細に文書で受託者に伝える必要がある。表2に記載すべき項目を示す。

表2 要求仕様書の構成綱目

システム化の目的	入出力情報一覧表
システム化適用業務	ファイル一覧表
システム化機能定義	開発スケジュール
システム論理定義	(入札条件) その他

受注者は、依頼者から受けた要求仕様を完全に理解してシステム設計に反映していることの確認するために基本設計書を作成し依頼者の合意を得て開発に着手する。基本設計書の構成項目を表3に示す。

表3 基本設計書の構成項目

システム概要	入出力設計	コード設計	ファイル設計
適用業務フロー	画面遷移図	コード体系	マスタファイル
システム構成図	画面レイアウト	区分体系	データファイル
サブシステム定義	出力用件仕様		集計ファイル
開発スケジュール	出力項目仕様		ファイル一覧表

作成したシステムの動作確認、仕上がり状況を確認するための受注者自身が行うテスト及び納入後に依頼者が行うテストの概要を、表4に示す。

表4 テスト概要

<p>納入前に受注者側で行うテスト</p> <ul style="list-style-type: none"> * 単体テスト：サブシステムごとの動作確認 * 結合テスト：全サブシステムを結合して、システムとしての動作確認 * 総合テスト：データ量の大小、異常データ挿入時の動作、異常操作時の動作等の確認
<p>納入後に依頼者が行うテスト</p> <ul style="list-style-type: none"> * 検収テスト：要求仕様どおりに仕上がっていることの確認(受注者の立会いが望ましい) * システム運用テスト：システム立ち上げ試行(受注者の立会いが望ましい)

なお、ソフトウェアの開発手順については、国際的に規格化されており、日本においても国際規格に順じた日本工業規格（JIS）が制定されている。この規格は、ソフトウェア製品及びサービスに関して、受注から供給までのプロセスを中心に、ソフトウェアの構想からその廃却に至るライフサイクルの管理について規定しているものである。したがって、この規定に基づいた開発工程の基準を守って、依頼者との情報交換を密に開発を進めれば紛争は避けられるはずである。

日本工業規格：ソフトウェア・ライフサイクル・プロセス JIS X 0160：1996

（国際規格：ISO/IEC 12207：1995 に準拠）

4. おわりに

以上、ソフトウェア開発を巡る訴訟の事例概要、紛争に至った原因の分析及び紛争の予防と対策について紹介してきた。ソフトウェア開発を巡る紛争を予防する対策を次に纏める。

- 1) 依頼者からの要求仕様を受けて、受注者が正しく理解していることを基本設計書によって双方で確認することからシステム開発が始まる。
- 2) 契約内容、双方の打合せ結果等は文書で残す習慣をつける。
- 3) システム開発を成功させるためには、依頼者と受注者の守備範囲を明確にし、共同作業という認識のもとで、双方のプロジェクト・マネジメントによるところ大きい。
- 4) 作成したシステムは、納入前に、要求仕様どおりに動作することの確認テストが大切。
- 5) 納品されたシステムは、速やかに要求仕様に沿って検収テストを実施する。
- 6) システム導入後のシステムの運用・保守体制を確立する。
- 7) システム開発は、JISの規定に沿って行うことが望ましい。