



ソフトウェア開発を めぐる法律問題

—紛争回避のためのポイント—

2

市毛由美子 (のぞみ総合法律事務所)

ソフトウェア開発におけるトラブル防止 のために

本稿では、ソフトウェア開発の各プロセスの中で、トラブル防止のためにどのような契約処理が望ましいか、誰に（どの機関に）どのような権利が帰属するのか、フリーソフトウェア等のパブリックドメインの権利関係はどうなっているのか、また、他人の権利を侵害しないためにはどのようなことに注意しなければならないのか等、について解説する。

ソフトウェア開発の手法と契約

❖開発手法と仕様の確定

ソフトウェア開発契約の重要な要素として、「仕様」「対価」「納期」が挙げられる。この中でも、「仕様」は最も重要な要素である。「仕様」が決まらなければ「対価」と「納期」も決まらないという関係にあるためである。「仕様」の確定については、ソフトウェア開発の手法により大きく異なる。以下ウォーターフォール型と非ウォーターフォール型を概観する。

ウォーターフォール型は、文字通り滝を意味し、開発プロジェクトを時系列に、「要件定義」「外部設計（概要設計）」「内部設計（詳細設計）」「開発（プログラミング）」「テスト」「運用」などの作業工程（局面、フェーズ）に分割し、原則として前工程が完了しないと次工程に進まないようにして、前工程の成

果物の品質を確保し、手戻りを最小限にするための計画性の高い手法である。これに対し、非ウォーターフォール型は、開発当初には全体的な仕様が確定しておらず、開発の過程で仕様が繰り返し変更され、時には大幅な変更も想定されているプロトタイプ型、スパイラル型やアジャイル型等の手法が典型例である^{☆1}。

この開発手法の違いを契約面で見ると、ウォーターフォール型では、遅くとも外部設計に入る段階でシステム全体の要件定義が決まり、要件定義に基づいた外部設計によりシステム仕様がほぼ確定することになるが、非ウォーターフォール型は、プロトタイピングや反復ごとの単位で新たな仕様（機能）が決まっていくもので仕様変更の繰り返しが想定されている。いずれもシステム仕様の確定は、料金、納期が連動してくることから、一度決まった仕様を後から変更する場合には、料金や納期にも影響が及ぶことが想定され、仕様変更によって契約の主要条件が大きく変わる可能性があることを念頭に置くべきである。

❖請負契約と準委任契約

民法は契約各論の章で、世の中の典型的な契約類型を例示して、各類型に応じた権利義務を定めている。この類型の中でソフトウェア開発の場合にあてはまるのは、「請負契約」と「準委任契約」である。「請負契約」は、一方の当事者がある仕事を完成することを請け負い、相手方がその結果に対して報酬を支

☆1 スパイラル型は、設計とプロトタイピングを繰り返して開発していく手法で、大規模プロジェクトでよく使われる。アジャイル型は比較的小さいプロジェクトで使われ、多くは、反復（イテレーション）と呼ばれる短い期間単位（1週間から4週間が多い）を採用することで、リスクを最小化する。各反復では、それまでに開発した成果物に1つの小さな機能を追加する。計画、要求分析、設計、実装（コーディング）、テスト、文書化といった、ソフトウェアプロジェクトに要するすべての工程を、1つの反復内で行う。

払うことを約束する契約で、原則として請負人は成果物に対する瑕疵担保責任（請負人の仕事の成果として引き渡されたものが、契約で定めた性能、品質等の条件を満たしていない場合の請負人の責任）を負うことになる。他方、準委任契約は、受任者が法律行為以外の事務サービス（コンサルタントサービス、要件定義作成支援サービスや保守サービス等の役務）を提供し、相手方がこれを承諾することを約束する契約で、受任者はサービスを提供する過程で合理的に期待される適時適切なアドバイスや情報提供を行う、預かったものを適切に保管する、等の注意義務（善管注意義務）を負うが、原則として仕事の成果に対して瑕疵担保責任を負うことはない。

世間で公表されているソフトウェア開発のモデル契約としては、① 2007年4月の経済産業省の「情報システム信頼性向上のための取引慣行・契約に関する研究会」のモデル契約、② 2008年に（社）情報サービス産業協会（JISA）が公表した「ソフトウェア開発委託基本モデル契約書」があり、この2つはいずれもウォーターフォール型開発を前提とした契約であるが、③ 2012年には（独）情報処理推進機構がアジャイル型を前提とした「非ウォーターフォール型開発にふさわしい契約」を公表している。

ウォーターフォール型モデル契約（①②）は、ユーザとベンダが対等の交渉力がある場合の、重要インフラや企業基幹システムの受託開発を前提とする。ここでは、要件定義はユーザが主体となって構築すべきシステムの要求を定める作業であり、ベンダはこれを支援する立場にあることから準委任契約が用いられ、外部設計書の作成については、ベンダが主体になる場合（ユーザの固有事情に比較的左右されないような場合やベンダにおいてユーザの固有性を

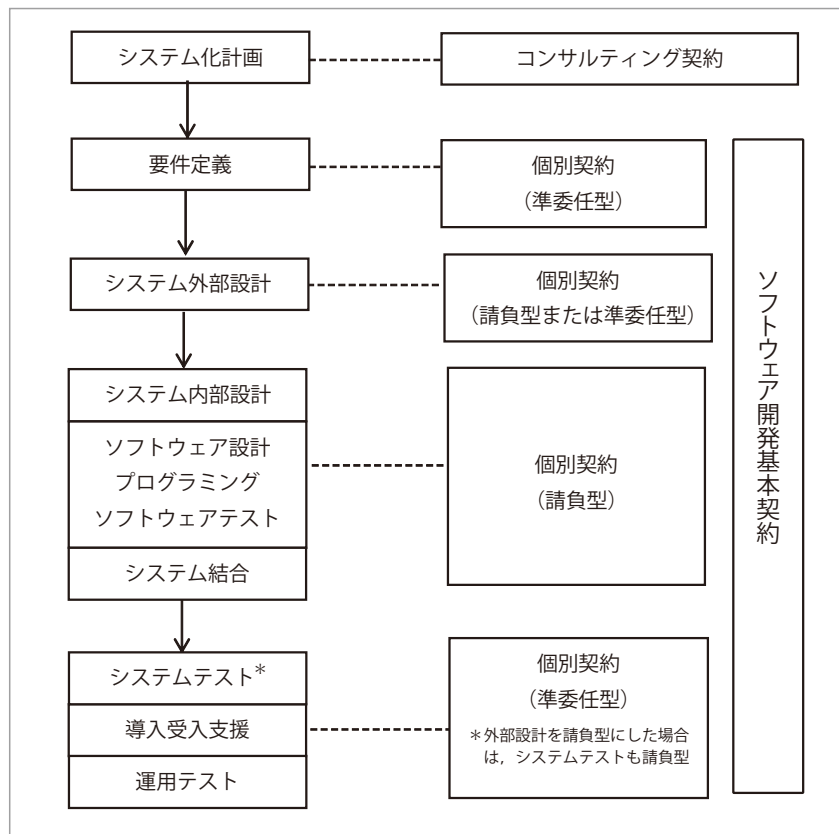


図-1 ウォーターフォール型開発と契約

把握できているような場合)には請負契約、ユーザが主体となる場合（ユーザ業務やそこで使用される画面・帳票の固有性が強い場合）には準委任契約、内部設計以下のソフトウェア開発業務はベンダ主体の請負契約、というように各フェーズごとに契約類型を使い分けている（図-1）。

他方で、非ウォーターフォール型モデル契約（③）は、対等に交渉力がありかつ技術力を有するユーザとベンダ間のアジャイルソフトウェア開発（対象：業務システム、パッケージソフトウェア等）が想定されている。基本契約と個別契約（準委任契約／請負契約）からなり、基本契約では、プロジェクト全体に関する共通事項を規定するが、当該内容は法的拘束力を持たず（内容とかい離する事態になったとしても、裁判になった際の法的責任を問われない）、プロジェクトの過程で変更されることを予定していることが明記されている。ユーザとベンダは協議を行い、ユーザが開発を望む一定の機能群のスコープが確定した時点、またはユーザがある一定期間ベン

ダに開発支援を委託することが確定した時点で、順次個別契約（準委任契約／請負契約）が締結され、具体的な開発が進められることになる。プロジェクト開始当初は、開発対象が特定されていない場合が多いため、成果物の完成を請け負う請負型契約は適当でなく、準委任型の契約が適切であるとされている。しかし、個別の反復過程において、機能が確定した場合には、その段階で請負型契約になることもある（図-2）。

瑕疵担保責任と責任制限

瑕疵とは取引通念からみて通常であれば同種の物が有すべき品質・性能を欠いていて欠陥が存在することをいい、契約に先立って売主が一定の品質や性能を保証した場合には、その基準に至らなければ瑕疵となる（大審院判大 15・5・24 民集 5 卷 433 ページ）。抽象概念であり、何が瑕疵であるかは紛争になる可能性があるが、要件定義書またはシステム仕様書との不一致がある場合、設計書上で理論的な誤りまたは誤記がある場合、インタフェースが不一致である場合等が挙げられる。

バグがシステム稼働に重大な支障を生じさせる場合は瑕疵が認められることになるが、裁判例では、システムが稼働を開始してから見つかった不具合でも直ちに修補ないし適当な代替措置を講じた場合は、当該不具合は瑕疵にならないとして損害賠償や解除を否定しているものもある（東京地判平 9・2・18 判タ 964・172、東京地判平 14・4・22 判タ 1127・161）。

民法上の請負契約では、瑕疵が発見された場合は、引渡から1年間に限り、発注者は、瑕疵および修補（修理等）を請求し、瑕疵により生じた損害の賠償を請求したりできるほか、契約の目的が達成できない場合には契約の解除ができるとされている（民

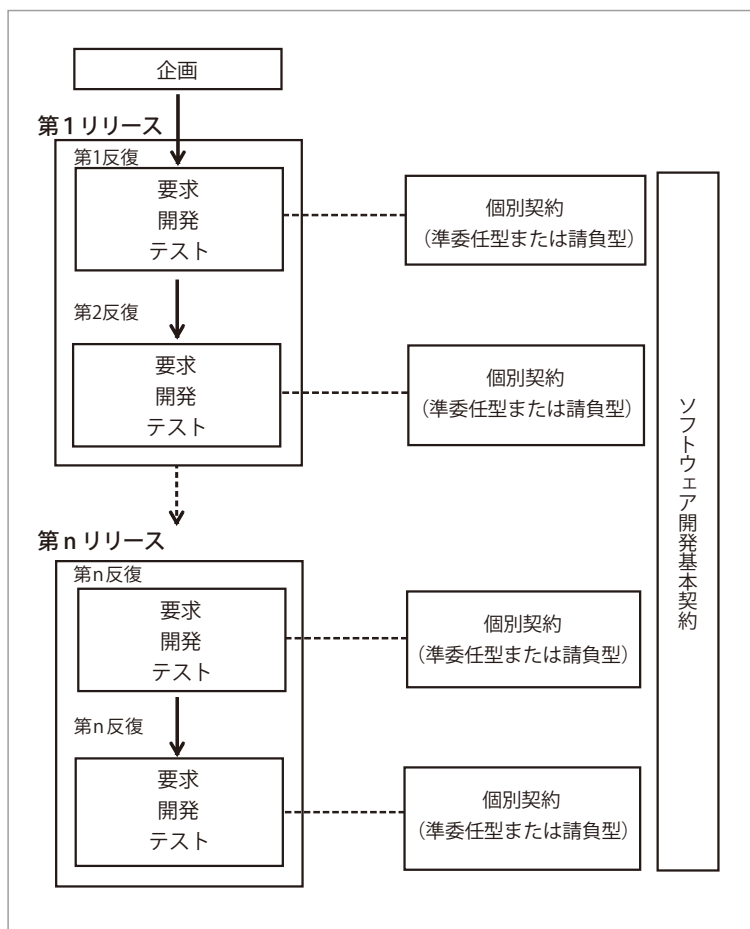


図-2 非ウォーターフォール（アジャイル）型開発の流れと契約

法 634 条、635 条）。

瑕疵担保責任のほかに損害賠償請求がなされることが想定される場面としては、検収で目的物が完成していないことが判明した場合等の不完全履行として債務不履行責任（民法 415 条）や、他人の権利を違法に侵害したことにより当該他人に損害を発生させてしまった場合に生ずる不法行為責任（民法 709 条）が挙げられる。瑕疵担保責任は無過失責任であると解されているが、契約によりベンダの責に帰すべき事由に限るとの責任限定をすることも可能である。

いずれの損害賠償においても、その範囲（金額）は、民法上、通常損害および特別事情による損害のうち予見可能な損害が、相当因果関係の範囲として定められている。ただし、ソフトウェア開発の場面における損害の範囲は、予測可能なものであってもきわめて巨額になる可能性があるため、契約により一定

限度に制限することも可能である。責任制限の契約文言例としては、「現実に被った直接かつ通常の損害のみに限り、逸失利益、特別事情による損害や第三者に生じた損害については賠償責任を負わない」「損害賠償責任は、本契約の契約金額を上限とする」「損害賠償責任は、支払い済みの代金相当額を上限とする」等がある。この点、ユーザ側は、「ベンダの責に帰すべき事由によって生じたすべての損害」について責任を負うことを求めることもあり、最終的には交渉により決定される。

責任制限規定のあり方は、どちらが正しいということではなく、民法上の相当因果関係を踏まえて想定される損害拡大リスクをどちらの当事者が負担することになるかの交渉の問題である。当然のことながら、リスクを負担する当事者は将来のコスト（損害賠償による特別損失）の負担を覚悟しなければならないため、理論的にはリスク負担と価格は相関関係に立つはずである。すなわちベンダがリスクを負担してすべての損害を賠償するのであれば対価は高くなり、逆に責任制限の規定を前提にユーザがリスクを負担するのであれば価格は安くなるのが合理的である。しかし、実際には個別具体的な事情と当事者間の力関係によってリスク分担の線引きがなされていると思われる。

仕様の変更管理

仕様変更は契約条件の変更となるため、両当事者の合意が必要な事項であるから、その都度、契約書またはこれに準じた書面で双方当事者が変更された条件を確認するべきである。プロジェクト・マネージャにこの意識が欠けていると、後になってトラブルが発生した場合、追加で発生した作業コストをどちらが負担すべきかの争いになる。先に述べたとおり、仕様と代金と納期は、相互に連動して契約条件を構成するので、この3つのいずれかが変更された場合は、代金や納期は従前のままでよいかについて協議して認識を共通にし、それを当事者双方が確認した旨の日付入り書面に残しておくべきである。ま

た、価格や納期を大幅に変更することができない場合は、すでに合意済みの価格や納期の範囲内で実現可能な仕様を再確定するための協議が必要となる。

裁判が起こされてから、裁判官がバグなのか仕様変更なのかを判断するのに3年もかかった例も報告されている（土木出来形自動作図ソフト事件・大阪地判平14・8・29）が、仕様変更管理の過程を適切に書面化することで、無用な紛争のコストが削減できることになる。

また、モデル契約はいずれも、仕様変更にかかわる協議が不調に終わった場合、個別契約のうち業務未了の部分について本契約または個別契約を解約することができるとしている。この場合、ユーザはベンダがそれまで遂行した個別業務についての委託料等を支払うことになる。

ユーザの協力義務とベンダの説明義務およびプロジェクト・マネジメント義務

ソフトウェア開発は、発注者と開発者との間で、緊密な協力関係が必要不可欠であり、両当事者は開発手法や開発段階のいかにかわらず、相互に協力義務を負っている。ベンダはユーザの業務内容等に必ずしも精通しているわけではないことから、ユーザは、業務内容やソフトウェアに期待される機能や役割、期待する操作性等を適格にベンダに伝え開発過程に積極的に参加する必要がある。いくつかの裁判例でも、ユーザはベンダに対し、ベンダが必要とする正確な情報、資料、機器等を提供し、また、ベンダとの打合せに応じるなどの協力をすべき信義則上の義務があると判示されている（東京地判平9・9・24判タ967・168、東京地判平16・3・10判タ1211・129等）。

他方で、ユーザは必ずしもソフトウェア開発に関する専門的な知識・経験を有しておらず、専門用語にも通じてないことが多い。このような情報の非対称性に鑑み、専門家であるベンダはユーザの信頼にこたえるべく十分な説明をする義務があり（説明義務）、さらには、自らが有する高度の専門的知識と

経験に基づき、納入期限までにシステムを完成させるようにユーザに提示し、ユーザ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、これに適切に対応すべき義務や、ユーザのシステム開発へのかかわりについても適切に管理するなどの行動をすべき義務（プロジェクト・マネジメント義務）を負っていると解されている（図-3）。

最近話題になっているケースとして、銀行業務全般をつかさどる情報システムの構築に関するパッケージソフトのカスタマイズによる開発について、基本合意および個別契約が締結されたが、結果として当該システム開発にかかる

プロジェクトが中止に至った事案で、第一審の東京地裁は、ベンダは「パッケージの選定にあたり、パッケージの機能・性能、設定・導入の容易性、導入実績、パッケージの提供者の導入実績、経営の安定性、技術力、カスタマイズへの積極性、その他関連する諸事情を考慮して、ユーザが構築しようとしているシステムに最適のパッケージを選定した上、これに適した開発方法を採用しなければならない」と判示し、ベンダにプロジェクト・マネジメント義務違反があると判断した（東京地判平 24・3・24）。

さらに、この事件の控訴審（東京高裁）では、企画・提案段階でのプロジェクト・マネジメント義務を「契約締結に向けた交渉過程における信義則に基

づく、不法行為法上の義務」と定義して、双方が契約を結んでいない段階でも、当該義務は発生し得ることを明確にする一方で、企画・提案段階ではシステム開発は一定の不確実性があるのも事実であるため、ユーザ企業についても「業務の分析とベンダの説明を踏まえ、システム開発について自らリスク分析することが求められる」と言及している。そのうえで、要件定義によって、当初予定の費用や開発スコープの達成が不可能と判明したにもかかわらず、ベンダが「システムの抜本的な変更、中止を含めた説明や提言、具体的なリスクの告知を行わないまま最終合意書を結んだこと」が、義務違反に当たると判断している（東京高判平 25・9・26）。

契約書においては、両当事者の義務や役割分担を、できる限り具体的に明確にしておくことが望ましい。

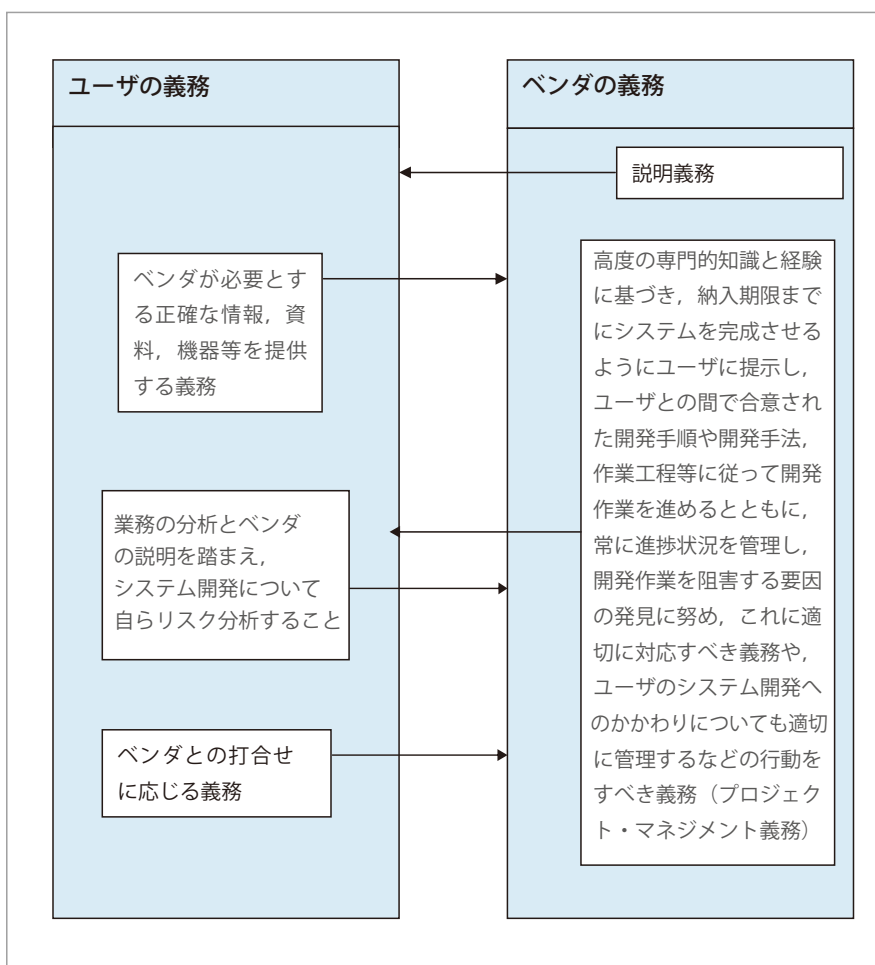


図-3 ユーザの義務 vs ベンダの義務

知的財産権の取扱い

ソフトウェア開発を巡り問題となる知的財産権は、主に特許権と著作権である。一般的に、ユーザ側は費用を負担するので、成果物の権利はすべて取得するのが当然であり、また、成果物にはユーザのノウハウや有用な情報が混入されているのですべて引き渡してもらわないと困ると考え、逆にベンダ側は、成果物の中に当該ユーザとの契約締結前から有していたモジュールやルーチンが入っており、また、新たに開発するものであっても、他の開発案件にも流用することが効率的なモジュール等があり、ビジネスの効率性からこれらの権利はぜひとも留保したいと考える傾向にある。

❖ 特許権の帰属

特許を受ける権利は、発明者に帰属し、発明者がベンダまたはユーザの従業員である場合には、通常定められている職務発明規定に従って雇用主であるベンダまたはユーザが権利を承継することになる。ベンダとユーザが共同で行った発明については、特許を受ける権利は共有となる。特許権または特許を受ける権利について上記と異なる取決めをするためには、その旨の特約を契約書で明確にすべきである。

なお、ベンダ側に特許権が帰属する場合、ユーザまたはユーザから成果物の使用許諾を受けた第三者等が開発成果物を使用することが特許権侵害とならないよう、ユーザ側が当該ソフトウェアを使用するのに必要な範囲で、ベンダからユーザに通常実施権を許諾することを明記し、かかる許諾料は開発委託契約の対価（委託料）に含まれていることを明確にしておく必要がある。ベンダが契約締結前から有していた特許権等についても、当該ソフトウェアを使用するために必要な範囲で、実施許諾を明記しておくべきである。

さらに、ユーザまたはベンダが通常実施権の許諾をする権限を確実に有することを担保するため、モデル契約書等では、職務発明の承継手続（職務発明規定の整備、運用およびこれに基づく譲渡手続等）がきちんと履践されていることを確認する条項が入れられている。

❖ 著作権の帰属

著作権は、著作物を創作した者に帰属し、法人著作の要件を満たすと雇用主である会社（請負契約類似の形態の契約ではベンダ）が原始取得する。ユーザ・ベンダ間では、ユーザのソフトウェアの有効活用とソフトウェア産業の技術力および生産力の向上のための技術の蓄積の要請等から、当事者の合意により決定することになる。経済産業省モデル契約では、A) ベンダにすべての著作権を帰属させるもの、B) 汎用的な利用が可能なプログラム等の著作権をベンダ、それ以外はユーザに帰属させるもの、C) 汎用的な利用が可能なプログラム等の著作権をベンダ、それ以外はユーザとベンダの共有にするもの、の3つが提案されている。なお、2007年には、国の委託研究の成果にかかる特許権等を受託者に帰属させることを可能とする「日本版バイ・ドール法」が産業技術力強化法の改正により盛り込まれ、経済産業省から「ソフトウェアに係る日本版バイ・ドール制度に係る運用ガイドライン」が公表されている^{☆2}。

また、契約条項で留意しなければならないのは、著作権法 27 条の翻訳、変更、翻案権等と同 28 条の二次的著作物に関する原作者の権利が譲渡の目的として特掲されていないときは、譲渡人に権利が留保されるとの著作権法の規定（同法 61 条 2 項）があるため、契約書にこれらの権利が譲渡の対象であることを明記しておくべきことである^{☆3}。また、

^{☆2} http://www.meti.go.jp/policy/innovation_policy/sangiho/8GL.pdf

「国等におけるソフトウェアの自由な利活用（複製、翻案又は改変等を含む）の確保と、秘密の保持ができれば、必ずしも特許権等を国等に帰属させる必要はなく、むしろ、当該特許権等を事業者に帰属させることにより、ソフトウェアの産業の技術力および生産性が向上するとともに、当該知財の対価の分だけ発注金額が低下する」とされ、国等が発注するソフトウェア開発では、その成果の特許権等を受託者に帰属させることが可能となった。

^{☆3} SCE 事件（東京地裁平 16・4・23）<http://www.courts.go.jp/search/jhsp0030?hanreiid=10370&hanreiKbn=07>

SCE（被告）からプレイステーション用のライブラリ・プログラムの開発委託を受けた原告が、開発したライブラリ・プログラムの著作権は原告に帰属していると主張して、プレイステーションの差止および損害賠償を求めた事件で、裁判所は契約書の取り交わしがなくても、著作権譲渡および著作者人格権不行使の黙示の合意があったと認定した。

著作者人格権（公表権、氏名表示権、同一性保持権）は著作者の一身に専属し、譲渡することができない（著作権法 59 条）が、ユーザ側が自己使用の目的で改変等することを想定し、ベンダ側が権利を行使しないという条項（不行使特約）を定めるのが一般的である^{☆4}。

なお、成果物に含まれるユーザのノウハウ等の営業秘密は、開発契約中の秘密保持義務の規定により保護され得るが、これらの営業秘密をベンダモジュールのコードの中には残さない工夫をすることも必要である。

オープンソースとフリーソフト

オープンソース・ソフトウェア（OSS）は、著作権のあるソフトウェアについて、ソースコードが開示され誰に対しても自由に改変や再頒布等の利用が許諾されているものをいう。GNU General Public License（GNU，一般公衆利用許諾書 / GPL）は Free Software Foundation という非営利の民間団体が提唱しているライセンス体系で、ソースコードの公開を原則としており、使用者に対してソースコードを含めた再配布や改変の自由を認めているものをいう。OSS は、優れたソフトウェアを生み出すためには、ソースコードを公開して有用な技術を共有し、世界中の誰でも自由にソフトウェア開発に参加できるようにすることが望ましいという発想に基づいている。フリーソフトウェアも、一般的には OSS と同様の意味で使用されているが、OSS とあわせて、

FOSS または FLOSS と表現されることもある。

GPL のもとでは、プログラムのソースコードをそのまま複製して再配布することができるが、著作権表示、無保証である旨の表示、GPL ライセンスの写しの添付等が要求される。また、ソースコードを利用してプログラムの全部または一部を改変し、これを複製または頒布することができるが、改変された旨および改変日付を表示することや、その派生的著作物を全体として無償で GPL に従ってライセンスすることが必要となるため、改変部分のソースコードを開示することが求められる。

JISA モデル契約においては、FOSS を利用するときには、変更管理手続きによるものとし、ベンダが提案する場合は、当該 FOSS の利用許諾条項、機能、開発管理コミュニティの名称・特徴など FOSS の性格に関する情報、当該 FOSS の機能上の制限事項、品質レベル等に関して適切な情報を、書面によりユーザに提供するよう努力するものとされている。ただし、FOSS 採用の決定権限はユーザにあることから、保守や障害対応に必要な措置はユーザの責任と費用で講じることとされている。

(2013 年 11 月 18 日受付)

● 市毛由美子 yichige@nozomisigo.gr.jp

1983 年中央大学法学部卒業。1989 年第二東京弁護士会登録、同年より 1994 年まで日本アイ・ビー・エム（株）にて企業内弁護士として勤務、都内法律事務所を経て 2007 年よりのぞみ総合法律事務所パートナー。

^{☆4} 宇宙戦艦ヤマト事件（東京地判平 13・7・2）<http://www.courts.go.jp/search/jhsp0030?hanreiid=12402&hanreiKbn=07>
宇宙戦艦ヤマトのアニメーション製作者である原告が、著作権は譲渡したものの著作者人格権は自己に留保されているとして、当該アニメーションを利用したゲームソフトの販売会社（被告）に差止および損害賠償を求めた事案で、裁判所は契約上明記されていなくても著作者人格権の不行使特約の合意があったことを認め、さらに、諸般の事情から原告の著作者人格権の行使は権利濫用であるとして、請求を棄却した。