

バロンくんのQ&Aコーナー

バロンくんが皆様からの質問にお答えします

Q:クリスマスは特別なことをしていますか？



A:質問ありがとうございます！

M.Hさんがしゃべりたがっているから話を聞いてみたよ。
「事務所が昨年8月に移転して、リッツ・カールトン福岡と同じビルに入ることになりました。
その特色を何か福利厚生に活かせないかと考えていたところ、知り合いの方から、スタッフやお取引先にクリスマスケーキをプレゼントしているというお話を聞いて、真似をしてみることになりました。スタッフにも好評でした。今後も(私の身銭を削って?)スタッフに還元しようと思います。
お世話になっている皆様には、毎年恒例のハンカチをお渡しできるよう準備を進めていますので、楽しみにお待ちいただければ幸いです。」

如水ハンカチは4月頃にお渡し予定です。

そのうち、如水口ゴじゃなくてぼうが印刷されたハンカチを作る予定だよ。
質問を募集しているよ。

匿名だから、気軽に質問してみてね！

<https://forms.gle/58kRrS6WqKYy9bdK9>



セミナーのご案内

第17回ミニ法務セミナーのご案内

テーマ:裁判例から学ぶ労務管理

日時 2025年2月19日(水)15時～15時30分

お申込みはこちらからお願いいたします↓↓

<https://vivit.video/s/142/fXH0QQqqqqqo0250116145504>



今回は、裁判例から学ぶ労務管理セミナーとして下記の裁判例についてお話いたします。

事例1 **通勤手段を偽り手当受給、退職願出さずに解雇に**
－ T 大学事件(東京地判令3・3・18)－

事例2 **部長がパワーハラ理由に譴責され処分無効求める**
－ C ホールディングス事件(東京地判令5・1・30) －

事例3 **退職申出後、機密情報を私的に保存したとクビ**
－ 伊藤忠商事ほか 1 社事件 (大阪地判令4・12・26) －

事例4 **同業他社への引抜き計画に関与した3人を懲戒**
－ 不動産研工業事件 (長崎地判令4・11・16) －

☆第18回ミニ法務セミナーのご案内

テーマ:2025年の法改正

日時 2025年3月19日(水)15時～15時30分

弁護士法人如水法律事務所

アソシエイト弁護士 白田晴夏
(福岡県弁護士会所属)
認定IPO実務プロフェッショナル



パートナー弁護士 橋本道成
(福岡県弁護士会所属)
認定IPO実務プロフェッショナル

如水法律事務所機関紙(毎月発行)

システム開発契約

Vol

30

2025年2月

〈編集・発行〉



弁護士法人
如水法律事務所

〒810-0041 福岡市中央区大名2-6-50福岡大名ガーデンシティ11F

TEL 092-738-8760 FAX 092-303-8560

<https://jwater-group.com/law/>

WEBサイトにて
最新情報をお届けしております



今回はシステム開発契約について解説します。

システム開発契約とプロジェクト管理

最初に、システム開発契約とシステム開発の分類について説明します。

システム開発契約とは、**委託者(ユーザー)**が**受託者(ベンダ)**に対し、**システム、ソフトウェアの開発に関する業務を委託する契約**と定義することができます。

システム開発についてどのような分類ができるかという、まず開発の仕方に基づく区別として、典型的なものとしては**ウォーターフォール型**と**アジャイル型**というものがあります。また、元となるパッケージソフトがあるかどうかの区別として、スクラッチ開発、もしくはパッケージ開発というものがあります。また、当該システムを用いて行う業務が現時点のユーザーの業務と同様かどうかに基づく区別としてASISやTO BEというものがあります。

この中で、ウォーターフォールモデルとアジャイル開発について、どのような違いがあるかについて説明します。

ウォーターフォールというのは、英語で滝を意味し、水が上から下に落ちていくというイメージで、システム開発の工程を分割して、それが順次進んでいき、後戻りすることはない。つまり、上から下に水が落ちてそれは戻ることはないというものです。

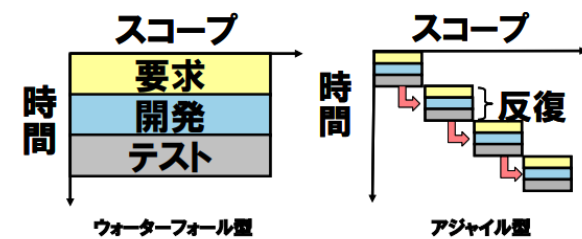
アジャイル開発というのは、それと異なり、開発して、検討して、また開発をしてということを反復継続しながらシステム開発を進めていくというものです。

ウォーターフォールは、設計の過程で仕様の詳細化をした上で、製作したプログラムの設計プログラムを逆にたどってテストしていく工程が標準的であり、企画、要件定義、ここをしっかりと定めた上で基本設計をして、その詳細設計と仕様の詳細化をしていく。そしてプログラミングをしていく。その後、それが正しくできているかどうかを単体テスト、結合テスト、システムテストという形で一連のテストを行った後、運用・保守に移るものです。

最初に説明したようにウォーターフォール、水が流れ落ちるように、前の工程で決めたものはそこから落ちていくと、後に戻れないということで、結構大規模な開発の場合に用いられることが多いです。

次にアジャイル型ですが、開発対象を多数の機能に分割して、それぞれの機能に優先順位付けをした上で、それぞれ開発していくものです。

典型的なものがスクラムと言われるもので、開発期間を1週間とか1ヶ月程度、スプリントと呼ばれる期間に区切って、さらにそれを1日単位のデイリースクラムというものに区切って、短期間での開発、そして目標、そしてその見直しということを繰り返していくという開発仕様です。



図表 2-1 アジャイル開発を単純化した標準モデル
出典:「非ウォーターフォール型開発WG活動報告書」(平成23年)46頁

図で示していますが、ウォーターフォールの場合は、要求、開発、テストが一つずつそのまま進んでいく。

それに対し、アジャイル型は、要求、開発、テストがあってそれを繰り返し反復していくという形です。

それぞれどんな特徴があるのかというものを表にまとめていますが、ウォーターフォール型の場合にはしっかりと機能を定めて、それに基づいて開発していく。一方で、一度決めた開発内容は変更しにくいし、途中で機能が不要になったり、機能に求めるべき内容が変わったりした場合、柔軟に対応できない。一方でアジャイル型は機能を分割して、それぞれを作って、また反復して違うものをやりながら、というような形で柔軟にできるという特徴があります。

	ウォーターフォールモデル	アジャイル開発
作業工程	すべての機能について同時に要件定義・基本設計・詳細設計等の要求の確定プロセスをこなしたうえで、開発(プログラミング)を行い、単体テスト・結合テスト・システムテスト等のテストを行う	機能を分割して、当該分割した1つ1つの機能について要求確定・プログラミング・テストを順次行っていく
柔軟性	一度決めた開発内容は変更しにくい 途中で機能が不要となったり、機能に求めるべき内容が変わったりした場合に柔軟に対応できない	開発内容を柔軟に変更しやすい 未対応の機能が残っている段階で、最優先で開発すべき機能が出てきた場合には、未対応の機能の優先順位を落として新規機能を開発するなど柔軟な対応ができる

システム開発契約の特徴

次にシステム開発の特徴について説明します。システム開発というのは、契約類型でいうと**業務委託契約の一種**です。

もっとも、システム開発契約の特殊性があります。このシステム開発の特殊性について**二重の専門性**という言い方をされることがあります。**二重の専門性**とは何かということですが、まずシステム開発であるため、**ITの開発については、ベンダ（いわゆるシステム会社）の方が、専門性を有しています**。ただし、システム開発で何を開発するのかというと、**基本的にはユーザーの業務に利用されるシステムを開発**します。そうすると、この**業務の内容というのはユーザーの方がよく知っています**。そのため、ITの開発、システムの開発はベンダの方が詳しいけれど、そのシステム開発によって省力化される業務の内容についてはユーザーの方が詳しいため、それぞれが知識を持ち寄らなければいけないことになります。それによって二重の専門性、すなわち、ベンダに頼るべきところ、ユーザーに頼るべきところというものが出てくるというわけなのです。

その結果、ユーザーの方も、自分は委託しているだけだからではなくて、自分たちの業務内容だとか、どんな風にしたら効率化できるのかをしっかりとベンダに伝えないとなりませんし、ベンダはITの専門家としてどういったものがITで実現できるのか、もしくは、どんなことは苦手なのかということをお互いに意思疎通をはからないといけないということになります。

例えば、よくシステム会社の方から相談を受ける内容として、システムの開発をある程度進めたのに、後で仕様の変更をしてほしいとユーザーの方から言われることがあるそうです。ユーザーの方からすると、ボタンの付け替えだけでしょとか、ちょっと表示を変えるだけでしょというように軽い気持ちで伝えているのかもしれませんが。けれど開発の方からして見ると、その変更に、開発の工数が結構かかりますということがよくあります。

これはお互いの意思疎通がよくできていないことの結果です。特にユーザーというのはベンダにお任せしてしまう傾向にありますので、ベンダ側としてもそうではないということをもっと業務内容をしっかりと伝えてもらわないと開発は難しい、工数がかかるということの啓発からしていかないといけません。

これを法的にとらえると、**ユーザーの場合は協力義務**、ベンダに協力する義務として構成することができ、一方で、**ベンダの方はPM義務（プロジェクトマネジメント義務）**に分類することができます。これに基づいて裁判でも、ユーザーは協力義務を果たしたのか、ベンダはPM義務を果たしたのか、といった観点から判断されます。

このようなシステム開発の特徴というものを見ていたときに、業務委託全般については以前もミニ法務セミナーや事務所報でも解説していますが、業務委託といっても、その性質が請負だったり、準委任だったり、様々という解説をしました。



簡単に請負とは何かというと、**請負は仕事を完成させるというものです**。これに対して**準委任というのは、完成ではなくて、ある業務を遂行するというものです**。

システム開発契約は請負でも準委任でもどちらもあり得ますが、ユーザーからすると、きちんとシステムを完成させてほしいという意味で請負を好みます。一方で、ベンダ側は、完成とすると、ユーザーの要望が拡大した場合などに、どこまでしたら完成なのか、判断が難しい場合があるので、準委任の方を好むという傾向があります。

システム開発契約の特有条項

それでは、システム開発契約の特有条項について見ていきたいと思います。
まず、**役割分担**です。
ユーザー側に有利な条文例というのを見ていくと、第1項は、**確認事項**や**協力事項**について書いています。
この他に、ベンダが分担作業を遅延した場合とか、実施しない場合に、それによって損害が生じたら、それに対して責任を負うというように、ベンダのプロジェクトマネジメント義務に違反するような場合には損害賠償責任が発生するという規定を入れています。

■役割分担
【ユーザ側】 ユーザ: 甲 ベンダ: 乙

第〇条（役割分担） 1. 甲及び乙は、本件業務の円滑かつ適切な遂行のためには、乙の有するソフトウェア開発に関する技術及び知識の提供と甲による仕様書の早期かつ明確な確定が重要であり、甲乙双方による共同作業及び各自の分担作業が必要とされることを認識し、甲乙双方による共同作業及び各自の分担作業を誠実に実施するとともに、相手方の分担作業の実施に対して誠意をもって協力する。 2. 乙は、共同作業及び乙が実施すべき分担作業を遅延し又は実施しないことによって、甲に損害が発生した場合、かかる遅延又は不実施について甲に対して責任を負う。
--

一方でベンダ側からすると、ベンダ側だけに規定するのは不利ではないかいうことで**双務的に修正する**ということを考えます。

その場合には、下の条項案で**下線で消している部分や青色付けしている部分を修正していくことが考えられます**。

■役割分担
【ベンダ側】 ユーザ: 甲 ベンダ: 乙

第〇条（役割分担） 1. …… 2. 甲及び 乙は、共同作業及び 乙が 各自の実施すべき分担作業を遅延し又は実施しないことによって、 甲に 損害が発生した場合、 それにより相手方に生じた損害の賠償も含め 、かかる遅延又は不実施について 甲に 相手方に対して責任を負う。

■役割分担
【ユーザ側】【ベンダ側】 ユーザ: 甲 ベンダ: 乙

第〇条（仕様書） 1. 本件ソフトウェアの仕様書を乙が作成するにあたり、乙は甲に要件の提示を求め、甲は乙の求めに応じて迅速に要件を提示しなければならない。 2. 乙が仕様書の作成を完了した場合、甲は、仕様書の記載内容が本件ソフトウェアの仕様書として適合するか点検を行い、適合することを確認した証として甲乙双方の責任者が仕様書に記名押印する。 3. 第2項の点検の結果、仕様書が本件ソフトウェアの仕様書として適合しないと判断された場合、乙は、協議の上定めた期限内に修正した仕様書を作成し、甲及び乙は再度前項の点検及び確認手続を行う。 4. 甲乙双方の責任者による記名押印をもって、仕様書は確定する。
--

次に、仕様書に関する記載を見ていきます。
ここでは仕様書の確認の流れを書いています。
① **まずユーザーが要件を提示して、**
② **それに基づいて仕様案としてベンダが提案する。**
③ **その提案されたものをユーザーが確認して承認して確定**
④ **これを確定したことによって、この仕様書に基づいて作業が進んでいく**
ことが決まります。
ただ、仕様書の内容について変更していくことがあります。
例えばユーザー側からすると、**業務との適合性の確認が不足していた**場合がありますし、ベンダ側からすると、**検証してみると技術的に難しいものだった**ということがありえます。
ウォーターフォール型では、まず仕様をきっちり決めるという説明をしましたが、実際には仕様変更は頻繁に見られます。
そのため契約レビューでは、この仕様変更についてはある程度柔軟にできるようにします。
ただし、仕様変更が費用とか、時間（工数）に影響する可能性があります。ですので、**その負担をどうするのかというところで有利不利というのが変わってきます**。
例えば、ベンダ側は、下のように仕様変更により工数が変わる場合は、追加の委託料を請求できるようにしておきたい。

■仕様の変更
【ベンダ側】 ユーザ: 甲 ベンダ: 乙

第〇条（仕様の変更） 1. …… 2. …… 3. …… 4. …… 5. 本条により、仕様の変更が確定した場合には、乙は、甲に対して、仕様の変更が確定するまでに乙が行った本件業務の遂行割合に応じて、追加で委託料の支払いを請求することができる。

次に、**第三者ソフトウェアを利用する場合**、ユーザー側は、下の第3項に入れているように、第三者ソフトウェアを使っても問題はないけれど、**きちんと使えるものかどうかについては、ベンダ側で確認してほしい**、という要望が出てきます。

■第三者ソフトウェアの利用
【ユーザ側】 ユーザ: 甲 ベンダ: 乙

第〇条（第三者ソフトウェアの利用） 1. 乙は、本件業務遂行の過程において、システム機能の実現のために、第三者ソフトウェア（フリーソフトウェア及びオープンソースソフトウェアを含む）を利用するには、甲の承諾を得なければならない。 2. 前項に基づき甲が第三者ソフトウェアの利用を承諾した場合、乙は、乙の費用と責任において、乙と当該第三者との間で当該第三者ソフトウェアのライセンス契約及び保守契約の締結等、必要な措置を講じる。 3. 乙は、第三者ソフトウェアの契約不適合、当該ソフトウェアが第三者の権利を侵害していないこと及び性能が十分であることについて調査を行う。
--

一方でベンダ側からすると、**開発を進めているすべての機能に適合しているかどうかの調査というものにも時間（工数）がかかります**。そもそも、第三者のソフトウェアが使えるという前提で開発を進めるから、開発費用をある程度抑えられて安く済むはずなのに、その安く抑えた費用で調査まで行えというのはムシがいいのではないかという思いを抱きます。
そうすると、第三者ソフトウェアに関する保証はしたくないということになります。

検収についての条項ですが、ユーザーからすると合格して検収終了するというのが当たり前の規定と考えます。
一方で、ベンダ側からすると、**合格でないのであれば、何が合格でないのか合理的に説明してもらわないと改修のしようもない**。使い勝手が悪いとか、そういう曖昧な基準では困るということになります。
ですので、下の条項に挙げていますが、合格検査書を交付しないのであれば、書面で具体的かつ合理的な理由を明示して異議を述べてもらう。
そうでないと合格したものとみなすという規定にしています。
ベンダ側からすると、受領拒否や検収拒否などされると、いつまで経っても納品できないとか代金を払ってもらえないという宙ぶらりんの状態になりますので、いつ、どういう場合に合格とみなすのかということが重要になります。

■検収
【ベンダ側 請負】 ユーザ: 甲 ベンダ: 乙

第〇条（検収） 1. …… 2. …… 3. 合格検査書が交付されない場合であっても、第1項の検査期間内に甲が書面で具体的かつ合理的な理由を明示して異議を述べない場合は、納品物は、本条所定の検査に合格したものとみなされる。 4. ……

最後に**完了確認**について見ていきます。
これもベンダ側からすると、**業務の完了報告書を提出して完了したという形でベンダがイニシアティブを持つのが望ましい**。
一方でユーザー側からすると、**ベンダが報告書を提出して完了ではなく、ユーザーの確認というプロセスを入れてほしい**という要望が当然に出てきます。
ですので、このような確認をどの程度するのか、ということを含めてベンダとユーザーの調整をして、システム開発契約が締結できるということになります。

■完了確認
【ユーザ側 準委任】 ユーザ: 甲 ベンダ: 乙

第〇条（検収） 1. 乙は、本件業務の作業完了日から遅滞なく〇営業日以内に、甲に対して本件業務の完了報告書を提出する。 2. 前項の完了報告書の提出をもって、乙による本件業務の履行は完了とする。 甲は、完了報告書を受領後遅滞なく、本件業務の完了を確認する。 甲は、善管注意義務違反がある場合そのほかの本件業務が本契約に基づき遂行されていない場合には、乙の選択に従い、本件業務の履行の追完を求めることができる。 この場合、受託者は、甲が定める期限内に無償で、本件業務の履行の追完をしなければならない。 3. 甲は、前項の確認の結果、本件業務の完了を認めた場合には、乙に対し、完了した旨を書面又は電磁的方法で通知し、当該通知を持って本件業務の完了とする。

