

IT紛争の類型的解決基準とIT契約における合意の効力

(副題) LT式システム紛争解決サービスとIT-ADRセンターの意義

2009. 3. 25

弁護士法人 エルティ総合法律事務所
所長弁護士／システム監査技術者／
公認システム監査人

藤谷 護人

<http://www.it-law.or.jp> <http://www.it-adr.jp>

I. システム開発を巡る幾つかの主要論点についての判例理論 (取り上げた判例一覧)

判例	事件名	判示事項	裁判所	事件番号	判決日付
判例①	請負代金請求控訴事件	建築請負における完成の基準 ＜形式的基準:最後の工程＞	東京高裁	S34(ネ)2336	S36.12.20
判例②	請負代金請求事件 反訴請求事件	未完成債務不履行の判断基準 ＜形式的基準:最後の工程＞と ＜実質的基準:結果無価値＞	東京地裁	H10(ワ)22251 H11(ワ)18926	H14.4.22
判例③	損害賠償請求事件 (本訴事件) 損害賠償請求事件 (反訴事件)	未完成債務不履行の判断基準 ＜実質的基準:行為無価値:プロジェクトマネジメント義務・協力義務＞	東京地裁	H12(ワ)20378 H13(ワ)1739	H16.3.10
判例④	損害賠償等請求、 売買代金等請求事件	「バグ」は、プログラムの瑕疵と 瑕疵担保責任が生じるだけか	東京地裁	H4(ワ)14387 H5(ワ)16569	H9.2.18
判例⑤	報酬金請求事件	開発対象範囲	東京地裁	H14(ワ)2077	H17.4.22

論点1.「未完成・債務不履行の判断基準」

＜形式的基準：最後の工程＞と＜実質的基準：結果無価値＞

- ・請負契約の仕事に何らかの欠陥がある場合、それが法的な評価として、債務不履行（請け負った仕事が完成していないから、報酬請求できない。）に当たるのか、瑕疵（一応報酬請求は可能であるが、別に瑕疵担保責任を負う。）に過ぎないのかの切り分けは、一般の事例でも難しい。
- ・もっとも、この切り分けの抽象的基準については、建築紛争について形成された、「工事が予定された最後の工程」を終了したかという基準でほぼ決着がついている（判例①）。そして、システム開発の類型でも、概ねこれに倣った基準—「請負人が仕事を完成させたか否かについては、仕事が当初の請負契約で予定していた最後の工程まで終えているか否かを基準として判断すべきであり」、「注文者は、請負人が仕事の最後の工程まで終え、目的物を引き渡した時には、単に仕事の目的物に瑕疵があるというだけの理由で請け負い代金の支払いを拒むことはできない」—が用いられている（判例②）。
- ・しかし、システム開発の場合、「予定された最後の工程」が何であるか、明らかでない場合が多い。そもそも、契約において工程を明確に合意しない例も多いし、建築のような場合に比べ、標準工程が確立されているとは言えないためである。
- ・一般的なシステム開発工程モデルによれば、システム開発の最終工程はテスト工程とされているから、結局、ベンダとしてなすべき最後のテスト工程ということになる。しかし、ベンダが担当しなかった上流工程に対応するテスト工程はベンダが行うべき工程なのか、テスト工程にユーザの参画が予定されている場合（ユーザに開発を分担する場合や通常のテストで受入テストを兼ねる場合にこのようなことが起こりがちである。）ベンダとしてどこまで終えていれば「最終工程」を終えたことになるのか、いかなる内容のテストをいかなる密度で行えば当該テスト工程を終えたことになるのか、といったことは一義的には決まらない。まさに、システム開発における作業工程のあり方そのものなのである。
- ・なお、「予定された最後の工程」を一応終えたという外観があったとしても、「システムの機能に軽微とは言えない支障を生じさせる上、遅滞なく補修することができないものであり、又はその数が著しく多く、しかも順次発現してシステムの稼働に支障が生じる場合」（判例④）には、実質的に「予定された最後の工程を終えた」とは言えない、ので完成していないということになる、と解釈される。

論点2. 「未完成・債務不履行の判断基準」

＜実質的基準：行為無価値：プロジェクトマネジメント義務・協力義務＞

・本判例は「被告は、システム開発の専門業者として、自らが有する高度の専門的知識と経験に基づき、本件電算システム開発契約の契約書及び本件電算システム提案書に従って、これらに記載されたシステムを構築し、段階的稼働の合意のと通りの納入期限までに、本件電算システムを完成させるべき債務を負っていたものである。」「したがって、被告は、納入期限までに本件電算システム開発の契約書及び本件電算システム提案書において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分析に努め、これを適切に対処すべき義務を負うものと解すべきである。そして、システム開発は注文者と打合せを重ねて、その意向を踏まえながら行う者であるから、被告は、注文者である原告のシステム開発への関わりについても、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原告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原告に働きかける義務(以下これらの義務を「プロジェクトマネジメント義務」という)を負っていたというべきである。」

・「具体的には、(1) 原告における意思決定が必要な事項や、原告において解決すべき必要のある懸案事項について、具体的に課題及び期限を示し、決定等が行われない場合に生ずる支障、複数の選択肢から一つを選択すべき場合には、それらの利害得失等を示した上で、必要な時期までに原告がこれを決定ないし解決することができるように導くべき義務を負い、(2) また、原告がシステム機能の追加や変更の要求をした場合で、当該要求が委託料や納入期限、他の機能の内容等に影響を及ぼすものであった場合等に、原告に対し適時その旨説明して、要求の撤回や追加の委託料の負担、納入期限の延期等を求めるなどすべき義務を負っていたというべきである。」と契約書等に明示の根拠がない場合にも「プロジェクトマネジメント義務」を認めた。

・この判決は、プロジェクトマネジメント義務の不履行を開発作業の遅延や完成に至らなかった原因として位置付けた。しかし、本件契約に原告の「協力義務」の明文があったことから、協力義務にも不適切な部分があると認定し、その結果、いずれの原因とも言えない、とした。しかし、損害金額については2つの義務の不履行の程度を6:4と認定し、金額に反映させている。

論点3. 「バグ」は、プログラムの瑕疵として瑕疵担保責任が生じるだけか

・システム開発委託契約を一応、請負契約の類型に当たると考えても、その「瑕疵」の扱いには通常に見られる特徴が見られる。本来、「瑕疵」とは、「完成された仕事が契約で定めた内容通りでなく、不完全な点を有すること」をいうとされているから、例えばプログラムにバグ、すなわち仕様書どおりに動作しない部分があれば、それは「瑕疵」に該当すると考えるのが素直である。ところが、システム開発においては、ユーザの検査に合格し、本稼働に移った後でも、一定レベルの初期バグの発生は、不可避である。実際、ベンダの出荷基準も、バグがゼロであることではなく、予め定められた一定の基準値以下になること、というような形で定められていることが多い。逆に言えば、ベンダも一定程度のバグの発生は覚悟したうえで、発生後は直ちにこれを修正できる体制を敷くのが通常なのであり、バグの存在が直ちに損害賠償や解除の原因になるとしたのでは、実情にそぐわない。

・裁判例の中にプログラムのバグがソフトの欠陥であるといえるための要件をかなり明確に述べたものがある(判例④)。

(1) コンピュータソフトのプログラムにはバグが存在することがありうるものであるから、コンピュータシステムの構築後検収を終え、本稼働態勢となった後にプログラムにいわゆるバグがあることが発見された場合においても、プログラム納入者が不具合発生 の指摘を受けた後、遅滞なく補修を終え、又はユーザと協議の上相当と認める代替措置を講じたときは、右バグの存在をもってプログラムの欠陥(瑕疵)と評価することはできない。

(2) これに対して、バグといえども、システムの機能に軽微とは言えない支障を生じさせる上、遅滞なく補修することができないものであり、又はその数が著しく多く、しかも順次発現してシステムの稼働に支障が生じる場合には、プログラムに欠陥(瑕疵)があるものといわなければならない。

・この判断基準を裁判手続において具体的に適用しようとする、バグが瑕疵に当たるかどうかを判断するためには、通常のように個々の欠陥の内容や性質を計るだけでは足りず、バグの存在が全体としてシステムの稼働にどのような影響を与えるのか、ベンダの採る代替策や対応体制が実害発生の抑え込みに十分なのかという、優れて専門技術的判断が必要になる。判例④では、コンピュータプログラムの欠陥の有無に関する事実についての争点を確定するため、当事者間で、原因説明作業実行計画書の作成し、これに基づいて二年余りにわたる検証実験が行われ、これによって争点の確定がなされている。

論点4. 契約の範囲(開発対象範囲)

・システム開発委託契約においては、契約の締結時に完成システムの具体的なイメージが明らかになっていることは殆どなく、開発の進行過程(つまり契約の履行過程)において、徐々に具体化・詳細化されていくのが通常である。このため、契約後に行われたある作業が、当初の契約範囲に含まれていたのかどうか、ベンダからの追加報酬の請求の可否に関して争いとなる(判例⑤)。

・そこで、ある作業が範囲の増加によるのか、単に具体化・詳細化によるのかを切り分ける必要が生じるが、この作業もまた容易でない。オンラインの業務アプリケーションを例に採れば、契約上で幾つかの機能が特定されていたとしても、各機能は何画面にも展開され得るし、幾つかの画面が特定されていたとしても、各画面には何項目でも入り得るし、幾つかの項目が特定されていたとしても、各項目にはいくらでもチェックや演算のロジックが組み込まれ得る。形式的にいえば具体化・詳細化に過ぎないと見られる過程においても、実質的には開発されるシステムの「量」を何倍にも膨らませる要素が含まれている。システム機能の内容にまで踏み込まなければ、十分な切り分けは出来ないのである。

・また、明らかに契約範囲外の作業がなされた場合ですら、一筋縄ではいかない。契約範囲外の作業であれば、ベンダとしてはこれを依頼されても応ずる義務はない。逆に、双方の合意により追加契約が成立しなければ、仮にベンダが作業を行ったとしても、ユーザには代金を支払う義務はない。しかし、これはベンダに応否の自由があった場合の原則である。システム開発の進行中に、システムの完成に不可欠かつ自己以外の者がこれを行うことが事実上不可能な作業の依頼を受けた場合、ベンダがこれを拒否することも事実上不可能に近い。作業拒否すればユーザに多大の損害が発生することは容易に分かるし、将来的に損害賠償責任も負いかねないからである。そのような状況の下で、ベンダが作業を行ったことをもって、無償で受け入れたと考えるのは、ベンダに酷であるばかりか、ベンダに過度に慎重な姿勢をとらせることになって、適時の対応を求めるユーザ自身の期待にも反することになる。裁判例としても、契約範囲外の作業である以上、黙示の合意を緩やかに認定するなどして、ベンダの報酬請求を認めるものが多いが、その前提として、このようなシステム開発における特殊な状況に対する理解が不可欠なのである。

Ⅱ. システム開発請負契約における合意の「特殊性」について

→次頁「システム開発請負契約における合意の構造概念図」参照

1. 「高度複雑専門的」合意

→申込と承諾の「意思の合致」の前提としての意思自治の原則に照らして、当事者間の対等性がユーザ側に十分でない
→「ユーザ側の専門性の不十分性の程度の見極め」と当該程度に応じたベンダ側の「ヒヤリングにおける聞き出し」や「手続における予見可能性の積極的説明」など（「プロジェクトマネジメント義務」の一部）が完成義務の付随的義務として認められる

2. 「契約締結時における目的物特定の抽象性」→「目的物の段階的順次特定性」

→一般契約における「特定」時点は内部設計段階★

3. システム開発請負契約における「合意の構造」について

→「源・全般的契約」と2つの「派・全般的契約」

→「追加・変更契約」

→「保守・運用契約」

→源・全般契約の2つの「発展的合意」

→「工程順次的/部分的/補完的/具体的合意」

→「各工程内の/部分的/補完的/具体的合意」

※2つの発展的合意の中で、源・全般契約の開発対象範囲外であると合意したものが「追加・変更契約」

4. 解釈上の問題点

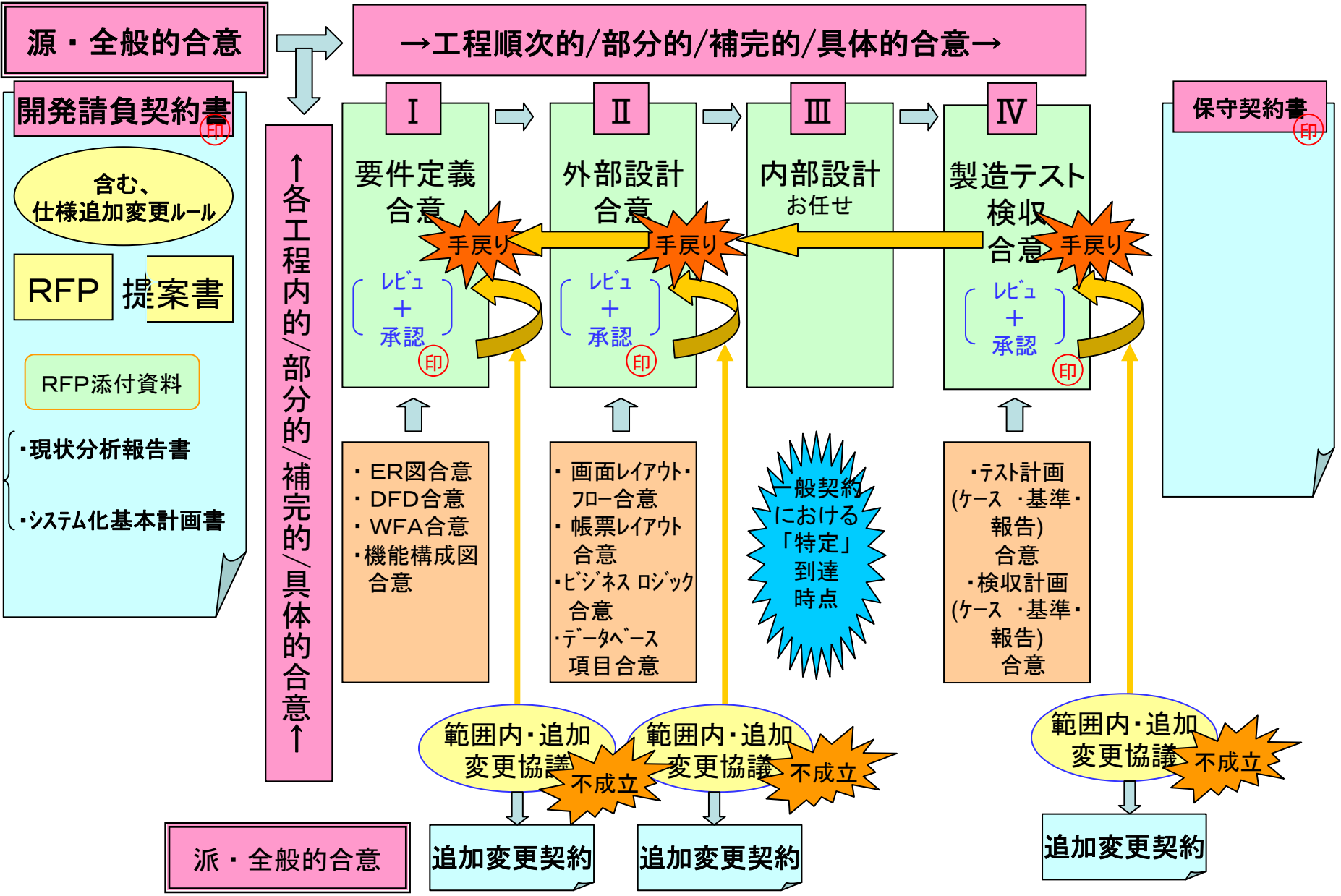
(1) システム開発請負契約の成立時期は何時か

(2) 一般契約における「特定」時点以前のコミュニケーションミスマッチに基づいて、開発対象範囲内か外かについての合意が困難な場合の合理的解釈基準は何か

(3) (2)の基準に照らして、例えば「ベンダが追加費用を認めしてくれないならそれ以後の開発は継続しない、と主張した場合」、あるいは「ユーザが開発対象範囲の内なのに製造されていない機能があるので未完成であり、検収しないし代金も支払わない、と主張した場合」に、どう解釈するか

(4) 「保守運用契約において、ベンダが見積もりどおりで認めなければ保守に応じない、と主張した場合」に、どう解釈するか

「システム開発請負契約における合意の構造概念図」



Ⅲ. IT紛争について、裁判からADRへ、さらにCPRへ

ADR (Alternative Dispute Resolution)

裁判外紛争解決手続(民営調停仲裁手続)

2008.8.1 softicソフトウェア紛争解決センター

2008.9.1 IT-ADRセンター <http://www.it-adr.jp>

■IT-ADR委員会

■IT-ADRセンター運営審査委員会

■IT-CPR研究会

CPR (Conflict Prevention Resolution)

紛争「予防」解決手続

LT式システム紛争解決サービスの概要 <http://www.lt-law.or.jp>

当事務所では、IT紛争の事後的解決としての「システム関係訴訟弁護サービス」、未然防止のための「LT式システム開発監理サービス」、そして紛争のプロジェクト内解決をも可能にする「IT-ADRセンターの運営」を用意しています。法とシステム技術との専門的融合により、御社のIT紛争を戦略的に解決する豊富なメニューを提供いたします。

Law
法的側面

法とシステム
技術の専門的融合

(1) 一方の代理人型業務

■システム関係訴訟弁護サービス

(2) コンサルティング型業務または中立型業務

■システム訴訟セカンドオピニオン・サービス

■システム訴訟鑑定サービス

■LT式システム開発監理サービス

RFP作成/業者選定/開発請負契約作成/開発作業の開始から稼働までの進捗監査・監理/運用保守契約

(3) 中立型業務

■IT-ADRセンターの運営

- 3ヶ月主義
- 専門的納得が得られる解決
- 公正中立性/厳格な非公開・守秘

「別添パンフ」
参照

- ・法による拘束性
- ・紛争の事後的解決

戦略的な解
決メニューを
提供

- ・合意による自律性
- ・紛争の未然防止
- ・紛争のプロジェクト内
解決

Technology
技術的側面

IV. モデル取引・契約書の問題点について

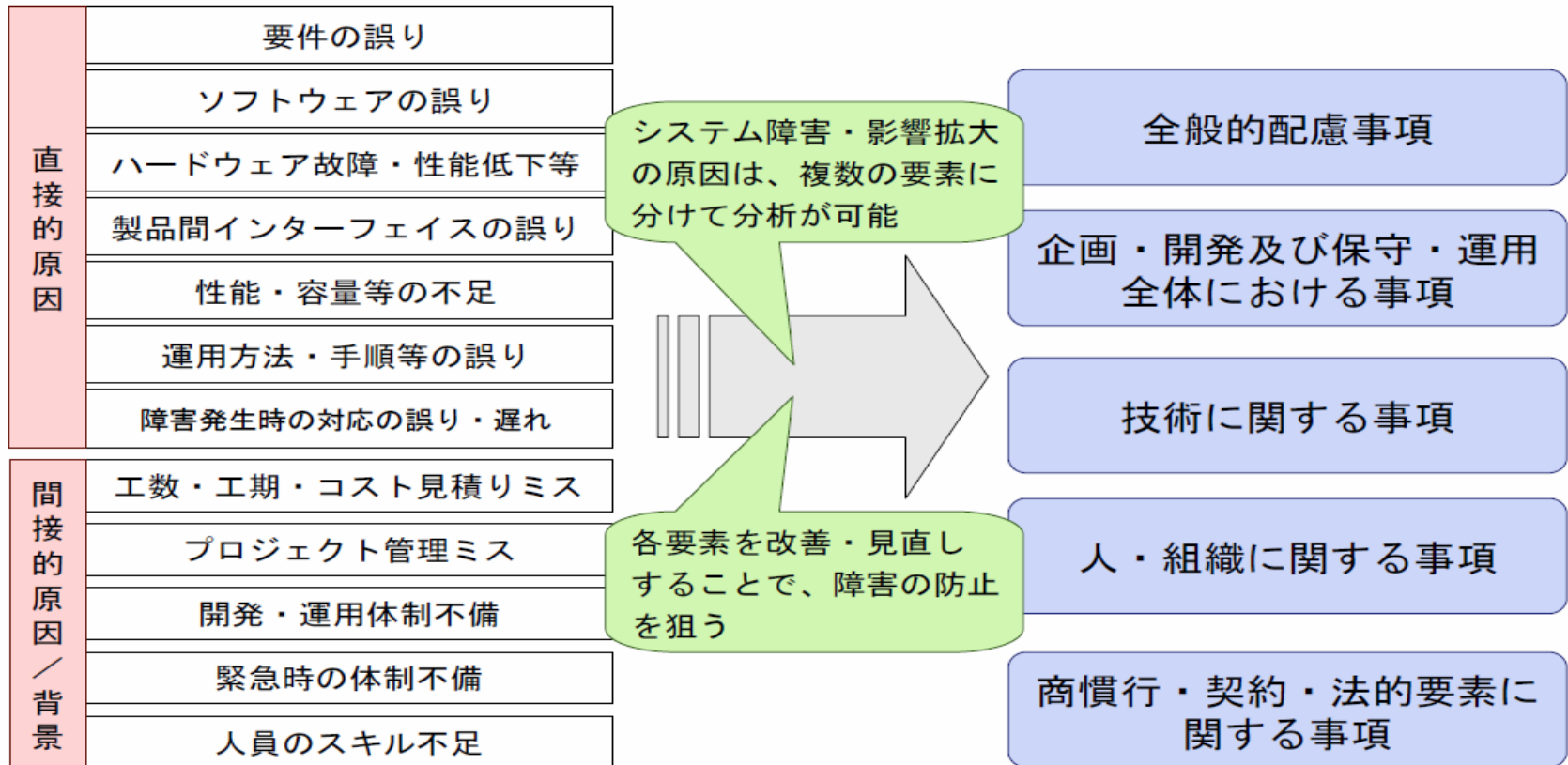
1. モデル取引・契約書の概要



信頼性・安全性向上に向けた方向性

システム障害・影響拡大の原因及び背景

改善すべき要素

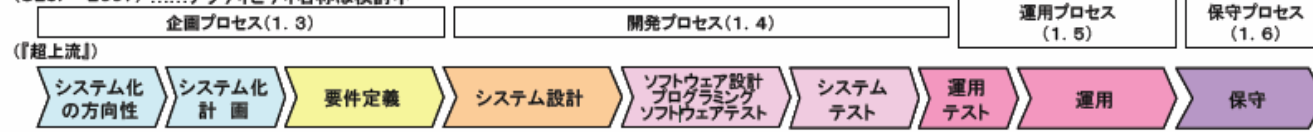


信頼性向上・取引可視化のための「モデル取引・契約書」の全体像

《フェーズ》

(SLOP-2007).....アクティビティ名称は検討中

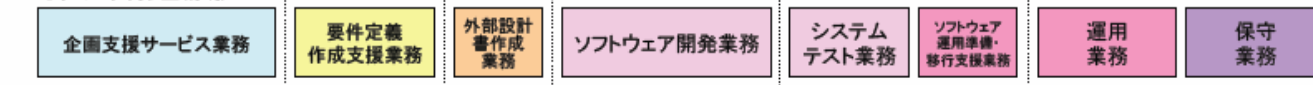
(各フェーズにおける論点のマッピング)



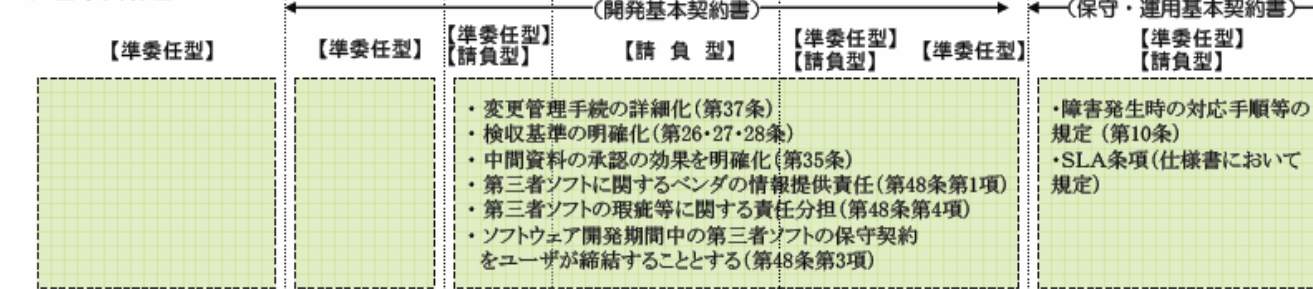
《契約プロセス》(例)



《モデル契約書雛形》



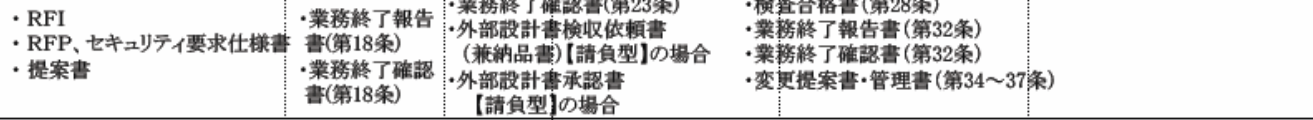
○ 基本契約書



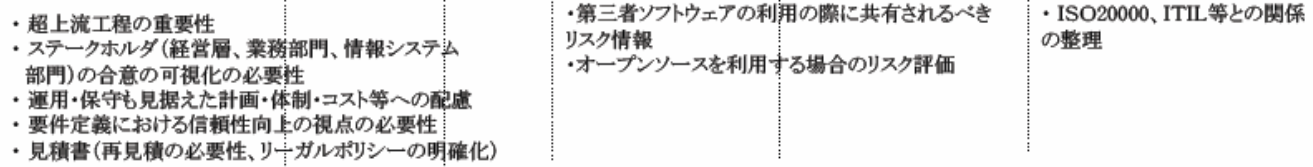
○ 個別契約書



《モデルドキュメント》



《契約プロセスガイド/条文解説の記載事項》



○ 各フェーズ共通事項等

- ・プロセス・用語の定義の明確化
- 《契約プロセス・手続き規定》
 - ・多段階契約と再見積りの考え方を採用
 - ・LOI(仮発注合意書)モデルを策定
 - ・マルチベンダ方式、分割発注時の考慮事項の整理
- 《責任関係》
 - ・ユーザ・ベンダの役割分担の明確化(第8条関係〈作業責任分担〉)
 - ・プロジェクトマネジメントの責任(第13条)
 - ・未決事項の確定手続・時期の明確化(第36条)
 - ・セキュリティ対策の責任の明確化(第50条)
- 《主要論点整理》
 - ・フェーズの分類と契約類型
 - ・再委託におけるユーザ承諾の要否(第7条)
 - ・損害賠償責任(第53条)
 - ・著作権の帰属(第45・46条)
 - ・第三者ソフトウェアに関わる瑕疵(第48条)
- 《その他》
 - ・工程分割発注分割発注を前提とした規定(例)要件定義書の精査・修正、変更の協議不調に伴う契約終了
- ・ドキュメントの定義の明確化(参考:第2条)
- ・プロジェクト推進体制図(第9・10条関連)
- ・作成者(ユーザ(内)、ベンダ)役割分担の明確化
- ・各フェーズの契約の性質、内容等
- ・役割分担における留意点(特にグレーゾーン)
- ・請負の瑕疵担保責任の性質(債務不履行の特則)
- ・第三者による監査の視点、内部統制

モデル取引・契約書（＜第一版＞及び＜追補版＞）の概要

	モデル取引・契約書＜第一版＞ （平成19年4月公表）	パッケージ、SaaS/ASP活用、保守・運用＜追補版＞ （平成20年4月公表）
契約当事者	対等に交渉力のあるユーザ・ベンダ	ITの専門知識を有しないユーザと業として情報サービスを提供するベンダ
対象モデル	ウォーターフォールモデル	パッケージ、SaaS、ASP
対象システム	重要インフラ・企業基幹システムの受託開発（一部企画を含む）、保守・運用	一般業務システム
特徴	<ul style="list-style-type: none"> ✓初のユーザ・ベンダ双方が議論の上策定。 ✓フェーズごとのユーザ・ベンダ間の責任の明確化（準委任・請負） ✓SLCP2007準拠（共通プロセス） ✓仕様の変更管理手続の明確化 ✓マルチベンダ・工程分割発注への対応 	<ul style="list-style-type: none"> ✓重要事項説明書を用いた契約合意 ✓ITコーディネータや中小企業診断士を始めとする外部専門家やコンサルタントの参画を前提 ✓システム構築後のプロセスの重視（保守、運用等） ✓パッケージソフトウェアの取扱いについてのベンダの責任明確化。 ✓著作権のベンダへの帰属。 <p>※上記以外の点について第一版の特徴は原則、追補版でも踏襲</p>

2. 「モデル取引・契約書」の問題点

1. 契約の「解釈基準」性と「行為規範」性

- ◆モデルのベンダへの偏頗性→ユーザとベンダの等距離中立性の回復
- ◆適用対象を無視した「モデル契約書」の「モデル性の濫用」の戒め

2. 工程と契約

- ◆工程毎の多段階契約と再見積りを前提とするのは、ユーザに、費用も納期も不明確な状態のまま、後戻りのできないシステム開発プロジェクトに入ることを強いることになる

3. 再委託

- ◆原則として再委託可 とするのは、ユーザの信頼が損なわれるおそれ

4. 協働・協力関係

- ◆素人としてのユーザの責任を不当に強化するものであって、法的権利義務関係の構成として不適切

(「モデル取引・契約書」の問題点つづき)

5. プロジェクトマネジメント

◆ユーザにプロジェクトマネジメント責任があるとまでいうのは判例を逸脱

6. 支援業務

◆超上流工程はユーザの業務に密接に関わるからユーザの責任、と単純に割り切るのは問題

7. 変更管理

◆容易に契約問題に転化する構造。しかも、協議不調であれば、ベンダは報酬を確保した上で契約関係から離脱することができてしまうのは問題

8. 著作権

◆ベンダ帰属を原則とするのは、ユーザは開発コストを全面的に負担しながら何の権利も取得できない、またノウハウ確保も不十分

(「モデル取引・契約書」の問題点つづき)

9. 第三者ソフト

- ◆ベンダが契約上の債務として選定した場合にも責任がないのはおかしい

10. 期限の利益喪失

- ◆実質的にユーザ側に不利な片面規定となっている

11. 損害賠償

- ◆瑕疵についても過失を必要とし、短期の期間制限を設け、賠償額の制限を設けるなどユーザ側に不利な片面規定となっている

12. 違約罰

- ◆設計・開発契約において違約罰が無い場合には、ベンダによる無責任な契約上の義務放棄を認めることになる