

2011/07/22 財団法人計算科学振興財団 ソフトウェア・エンジニアリングセミナー

ソフトウェア開発の標準プロセス

IPA（独立行政法人 情報処理推進機構）
SEC（ソフトウェア・エンジニアリング・センター）

研究員 室谷 隆

はじめに

ソフトウェアエンジニアリングとは

1. ソフトウェアの開発、運用、および保守における、システムティックであり、ディシプリン(*)に基づいた、定量的なアプローチの適用である。換言すれば、ソフトウェアへの工学の適用である。
2. 1. で示したアプローチに関する研究である。とされている。

(*) ディシプリン：方法論に基いた教育・訓練によって形成された規律

つまり

体系化し、
それに従って作業し、
データを収集し、フィードバックすること。

Abran, Alain (eds.), *SWEBOK; Guide to the Software Engineering Body of Knowledge*, 2004 Version, IEEE Computer Society, 2004

第1部 ソフトウェア開発の標準プロセス ～共通フレーム2007の概要～

第2部 日本独自のプロセス拡張のねらい ～超上流の重視 超上流とはなにか～

～共通フレーム2007の概要～

1. 共通フレームとは
2. なぜ、プロセスが重要なのか？
3. 共通フレームの特徴
4. 共通フレームのプロセス体系
5. 共通フレームの要素と階層
6. 「共通フレームとガイダンス」の見方
7. 規定例 - 「1.5 要件定義プロセス」
8. 修整（テーラリング）の適用について
9. テーラリング方法

1. 共通フレームとは（1／3）

■共通フレームとは

ソフトウェアの構想から開発、運用、保守、廃棄に至るまでのライフサイクルを通じて必要な作業項目、役割等を包括的に規定した共通の枠組み(※1)。

何を実施するべきかが記述されている、

「ITシステム開発の作業規定（プロセス）」である。

■その目的は

日本において、ソフトウェア開発に関係する人々(利害関係者)が、「同じ言葉で話す」ことが出来るようにするため(※2)。

(※1) 歴史的には、1994年に『共通フレーム94』が発表され、1998年の『共通フレーム98(SLCP-JCF98)』を経て、2007年10月に『共通フレーム2007(SLCP-JCF2007)』(第1版)が刊行された。また、第2版が2009年10月1日に発行された。

(※2) より詳しい目的・背景(共通フレームの必要性)が、『共通フレーム2007』(第2版)の p.3～13 で説明されている。

1. 共通フレームとは（2／3）

■その背景は(主として)

- (1) 利害関係者同士の認識のズレによるトラブルの発生がある。
- (2) 取引(主として二者間契約)における作業項目、役割分担等が明確化でなく、取引の適正化がされていない。

■その作成者は

ユーザ企業、ベンダ企業、IPA/SEC、大学、経済産業省からなる開発プロセス共有化部会(2007年10月当時)である(※1)。

(※1) 『共通フレーム2007』(第2版)の p.326 (「第1版 編著者」) を参照。

1. 共通フレームとは（3／3）

■ソフトウェア開発方法論との関係は

ウォーターフォール、スパイラル、プロトタイプ、アジャイル系
すべての開発方法論に共通したものの。

■JIS X 0160(ISO/IEC12207)との関係は

- (1) 共通フレームは、JIS X 0160 を逐次参照している（JIS X 0160 を包含する）。
- (2) IPA/SECの立場としては、共通フレームの参照／利用を推進している（※1）。

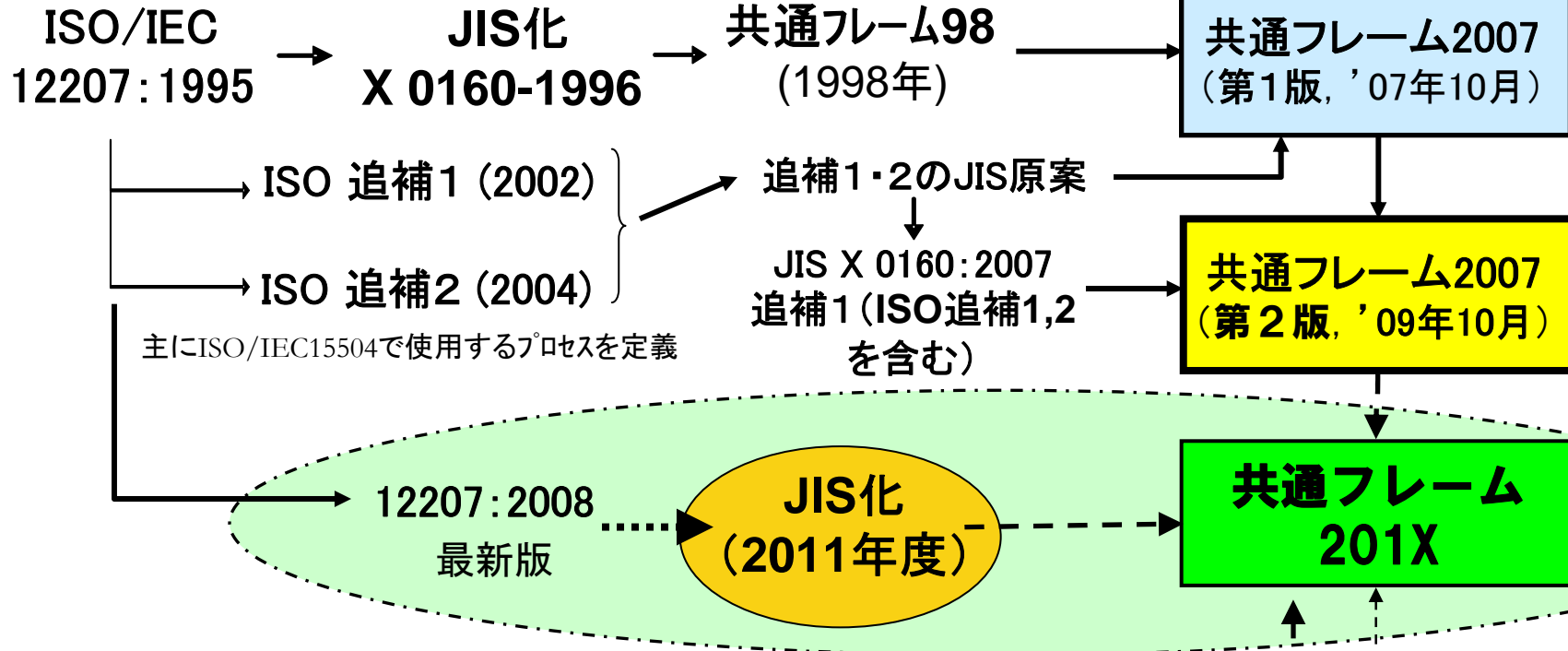
(※1) 経済産業省を始めとして官公庁でも、JIS X 0160 よりも 共通フレームを参照するよう啓蒙している。

また、共通フレームの作成者（企業や諸機関に所属する人々）自身が中心となって、例えば顧客向け講演会等と言及し、その普及／啓蒙に努めている。

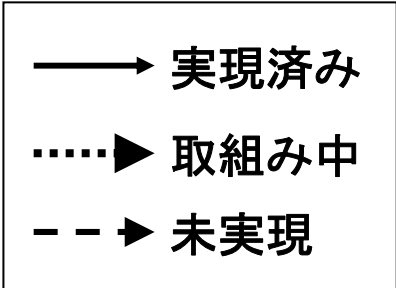
【参考】共通フレーム2007（第1版，第2版）の位置づけ

超上流の本

【ソフトウェアライフサイクルプロセス】



【システムライフサイクルプロセス】



2. なぜ、プロセスが重要なのか？(1/2)

■製品の品質はプロセスの品質から
(工学(エンジニアリング)の基本)

■プロセス : インプットをアウトプットに変換する,
相互に関連する又は相互に作用する一連の活
動 (JIS Q 9000:2006) (処理する、加工する、手を加える)
活動を役割の観点でまとめている。

例 開発プロセス、運用プロセス、保守プロセス
What to do(何をするか)であり、
How to do(どのようにするか)は決めていない。

2. なぜ、プロセスが重要なのか？(2/2)

■組織が提供する製品(サービス)の品質向上や、経営環境の変化に応じた適正価格対応などの面から、プロセスの確立、標準化及び改善が重要なため。

■もし組織のプロセスが確立されていなければ、どうなる？

- (1)製品(サービス)の品質(Quality)が、安定しない。
- (2)納期(Delivery)が、不確実となる(納期を必ずしも守れない)。
- (3)教育訓練の標準化も図れない(作業手順や方法がバラバラ)。
- (4)プロセスの確立や標準化が図れないため、ITによるシステム化が難しい。
- (5)コスト(Cost)は、(1)~(4)の結果として、増大する。

→ 組織がその顧客に提供する製品(サービス)の品質を維持し、または向上させていくためには(顧客満足度の獲得)、組織の活動基盤となる「プロセスの確立」が重要である。

3. 共通フレームの特徴

(1) 超上流の重視

(2) モジュール性の採用

(3) 責任の明確化

(4) 責任範囲の明確化

(5) 工程、時間からの独立性

(6) 開発モデル、技法、ツールからの独立性

(7) ソフトウェアを中心としたシステム関連作業までを包含

(8) システムライフサイクルプロセスとの整合性

(9) 文書の種類、書式を規定しない

(10) 修整(テーラリング)の採用

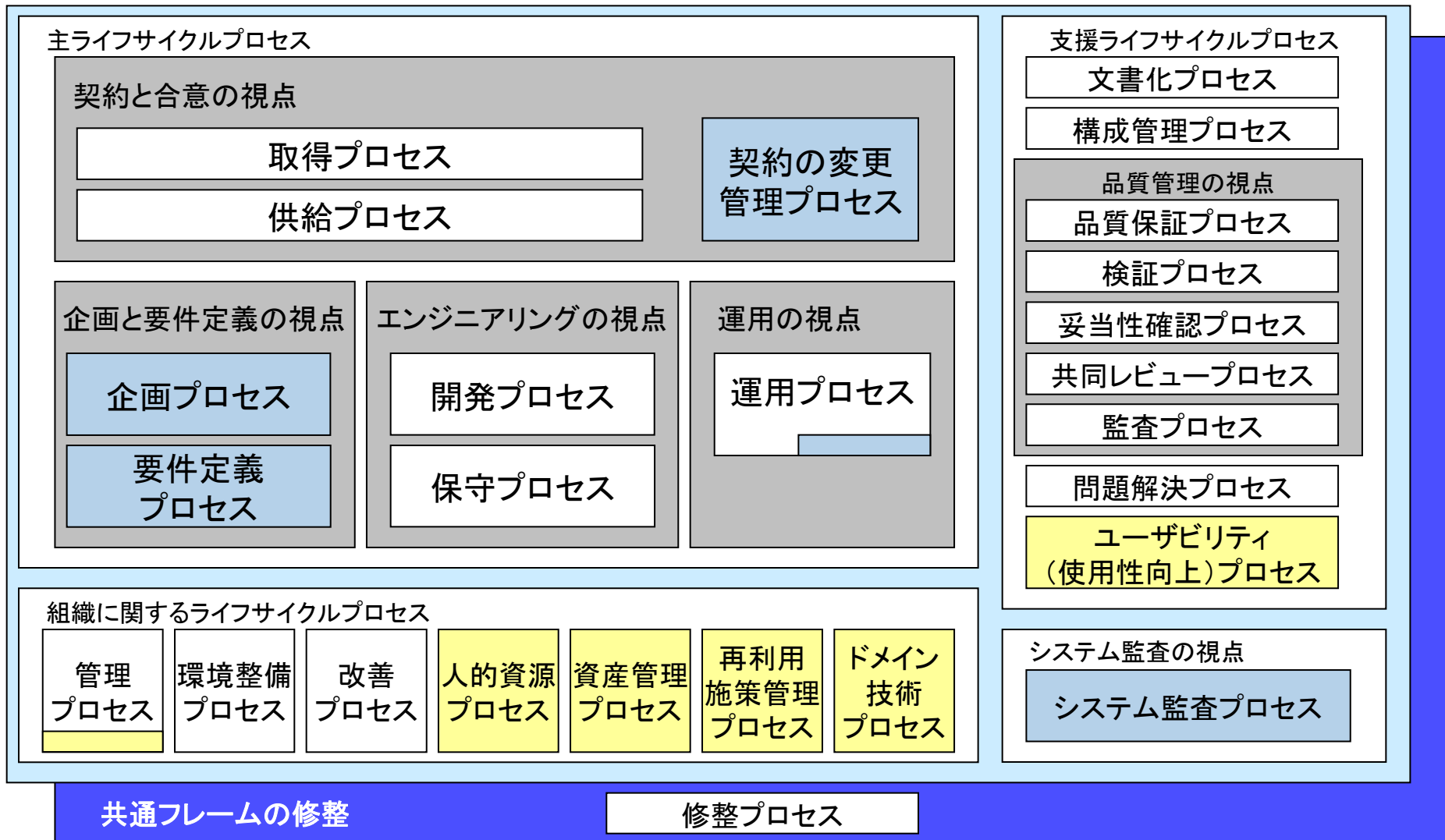
(※1) 左記①は、第2版で明記されたが、その考え方自体は第1版にも含まれていた(第2版で明確化したということ)。

なお、後述するが、「超上流」の範囲は、「企画プロセス」と「要件定義プロセス」からなる。

(※2) 上記(1)～(10)の詳細説明については、『共通フレーム2007』(第2版)の p.21～29 を参照。

なお、プロセスを修整(テーラリング)する上で、上記(5)(6)(9)を強く認識しておく必要がある。

4. 共通フレームのプロセス体系

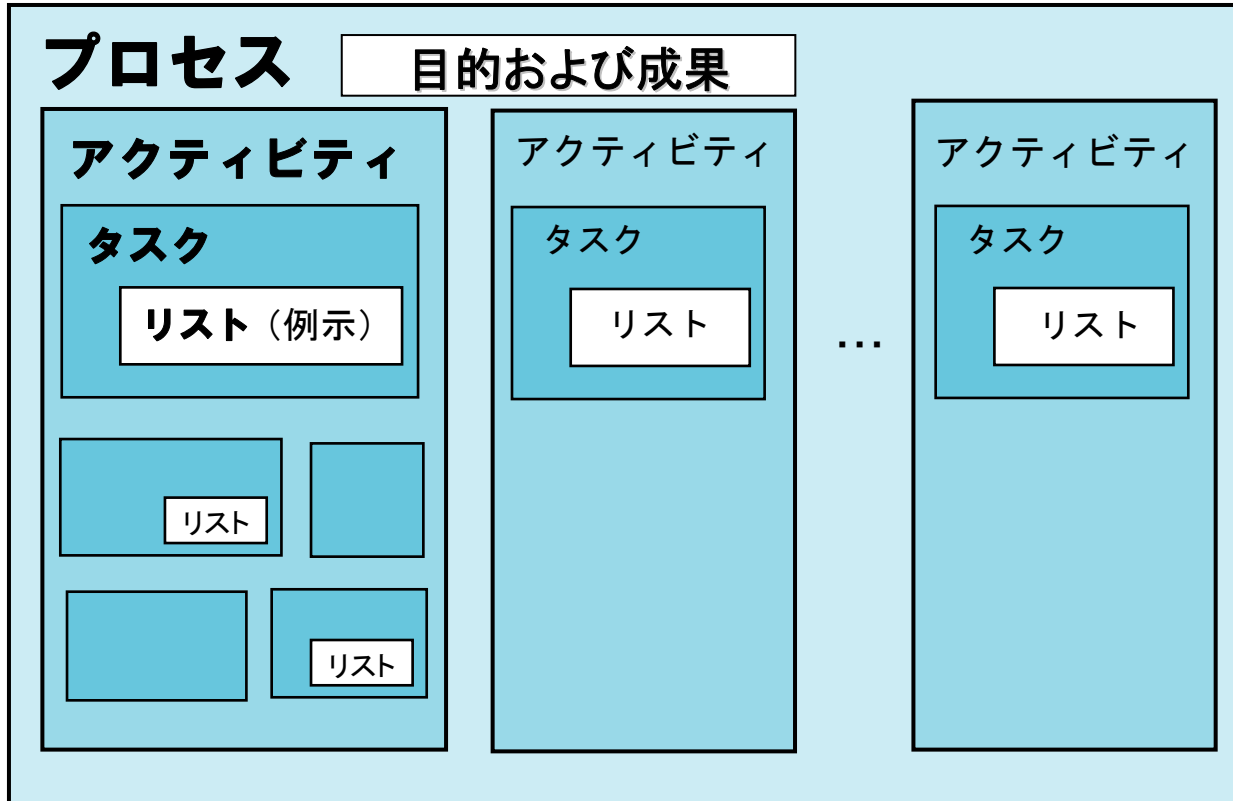


: 規格部分
 : 追補で変更, 追加された部分

: 共通フレームで拡張した部分
 (※1) 上記の図は、『共通フレーム2007』(第2版)の p.38 に掲載。

5. 共通フレームの要素と階層

●次の図のように、4つの要素が階層化されている。



■ プロセスとは、
システム開発作業を役割の
観点でまとめたもの (※1)。

■ アクティビティとは、
相関の強いタスクをまとめた
タスクの集合のこと (※1)。

■ タスクとは、
アクティビティを構成する個々の
作業のこと (※1)。

■ リストとは、
タスクを構成する要素のこと。
なお、JIS規格でも、共通フ
レームでも、リストは「例示」と
して取り扱う。

(※1) 上記の図は、『共通フレーム2007』(第2版)の p.39 に掲載。

また、プロセス、アクティビティ、タスクの規格上の定義については、『共通フレーム2007』(第2版)の p.38 を参照。

6. 「共通フレームとガイダンス」の見方

(本体の形式)

1.6.10.4 システム適格性確認テストの準備

システムの適格性確認要求事項ごとに、システム適格性確認テストを行うため、一連のテスト、テストケース(入力、出力及びテスト準備)及びテスト手順を作成し、文書化する。開発者は、結合したシステムがシステム適格性確認テストを実施できる状態にあることを確認する。

テスト実施にあたって各種マスタファイルのデータ、トランザクションデータを作成し、テスト環境に登録する。

ガイダンス

1.6.10.4: データは本稼働で用いるデータにできる限り近いものを設定する。現行システムのデータが存在する場合は、セキュリティを考慮し移行して利用する。

(青色の囲み) 共通フレーム定義体を表す。

(文字種) 国際標準: **太字**、国際標準の追補: **太字**/*斜体*、国内での追加部分: 細字。

(ガイダンス) 国内で追加した解説を表す。

国際標準との差異を明示している(※2)。

(※1) 上記の「見方」は、『共通フレーム2007』(第2版)の p.77 に掲載。

(※2) 文字種について、書籍上では、「太字」はゴシック体に、細字は「明朝体」による表示となっている。

7. 規定例 – 「1.5 要件定義プロセス」(1/2)

目的：

要件定義プロセスの目的は、新たに構築する（あるいは再構築する）業務、システムの仕様を明確化し、それをベースにシステム化範囲とその機能を具体的に明示することである。また、関連する組織およびシステムに対する制約条件を明確にし、定義された内容について取得者側の利害関係者間で合意することである。

(※1) ここで示す規定例は、第2版に基づく。

成果：

要件定義プロセスの実施に成功すると次の状態になる。

取得者側の利害関係者間で

- (1) 対象システムを含む業務、組織に関する要件が定義され、かつ合意される。
- (2) システムに対する要件及び制約事項が定義され、かつ合意される。
- (3) 定義された要件と、プロセスの入力となったニーズとの整合性が保たれる。

...

(※2) 「成果」とは、プロセスの実施が成功した状態のことをいう。

7. 規定例 – 「1.5 要件定義プロセス」(2/2)

アクティビティ一覧

このプロセスは、次のアクティビティからなる。

- (1) プロセス開始の準備
- (2) 利害関係者要件の定義
- (3) 利害関係者要件の確認

1.5.2.2 業務要件の定義

要件定義者は、新しい業務のあり方や運用をまとめた上で、業務上実現すべき要件を明らかにする。業務要件には例えば次のようなものを記述する。

- (a) 業務内容（手順、入出力情報、組織、責任、権限など）
- (b) 業務特性（ルール、制約など）
- (c) 業務用語
- (d) 外部環境と業務の関係、授受する情報

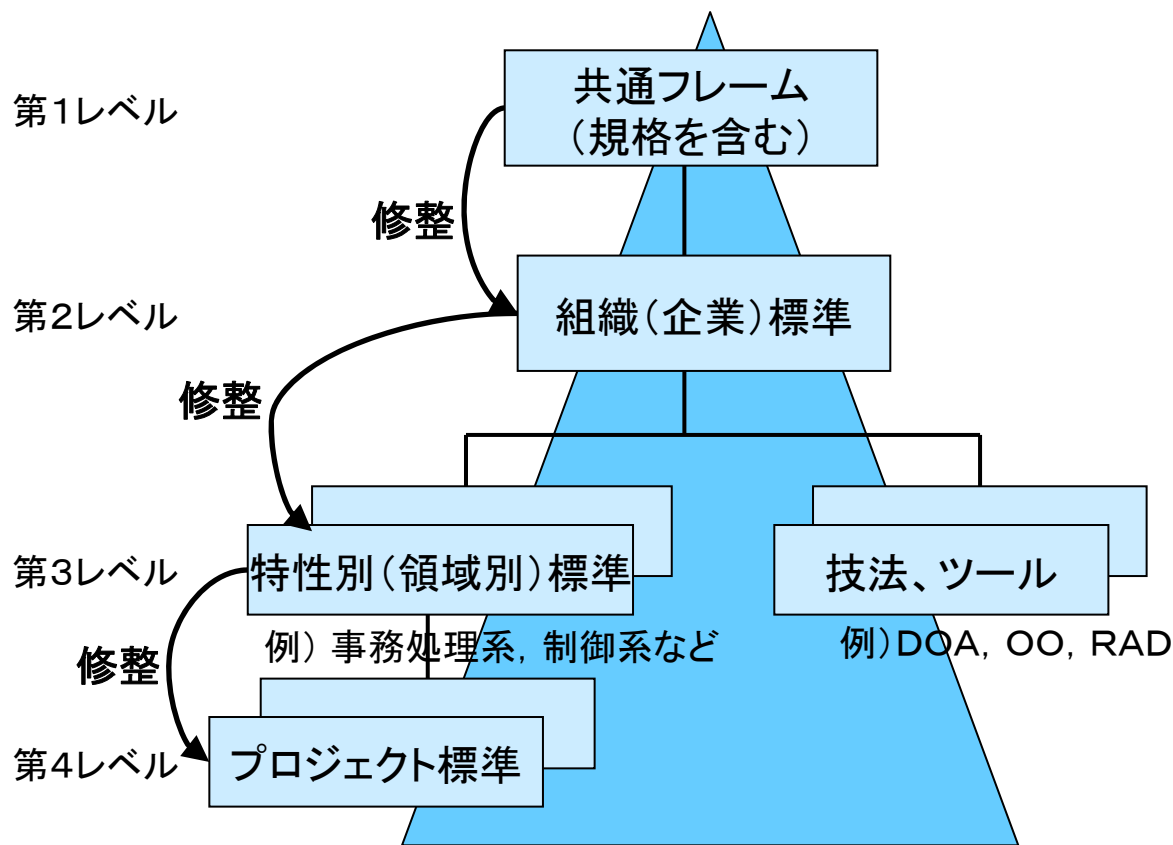
- 1.5.1 プロセス開始の準備
 - 1.5.1.1 要件定義作業の組立て
 - 1.5.1.2 必要な支援プロセスの組込み
 - 1.5.1.3 利害関係者の定義と役割の確認
 - 1.5.1.4 要件合意及び承認ルールの決定
 - 1.5.1.5 要件定義環境の準備
 - 1.5.1.6 要件定義プロセス実施計画の作成

- 1.5.2 利害関係者要件の定義
 - 1.5.2.1 利害関係者のニーズの識別と制約事項の定義
 - 1.5.2.2 業務要件の定義
 - 1.5.2.3 組織及び業務環境要件の具体化
 - 1.5.2.4 機能要件の定義
 - 1.5.2.5 非機能要件の定義
 - 1.5.2.6 スケジュールに関する要件の定義
 - 1.5.2.7 実現可能性とリスクの検討

- 1.5.3 利害関係者要件の確認
 - 1.5.3.1 要件の合意と承認
 - 1.5.3.2 要件変更ルールの決定

リストとしての例示。

8. 修整(テーラリング)の適用について (1/3)



(注1) DOA : データ中心のアプローチ
(注2) OO : オブジェクト指向の方法論, 技法など
(注3) RAD : 短期間アプリケーション開発技法

■修整(テーラリング)とは、

共通フレームをそのまま適用するのではなく、組織(企業)やプロジェクトの特性(例えば開発モデル)に合わせて、共通フレームで規定されているプロセス/アクティビティ/タスクを取捨選択したり、繰り返し実行できるように、又は複数を一括して実行できるように組み替えたりする作業をいう。

この修整(テーラリング)プロセスが、共通フレームに含まれていることによって、共通フレームの適用が、非常に柔軟なものとなる。

(※1) 上記の図は、『共通フレーム2007』(第2版)の p.32 に掲載。

8. 修整(テラリング)の適用について (2/3)

(※1)『共通フレーム2007』(第2版)の p.268~269 参照。

●テラリングのポイント

(1) 「共通フレームに書いてあることをすべてそのまま実行しなければならない」ということではない。

(2) 「共通フレームに書いてあること」を、妥当と判断した場合には、省略してもよい。

(組織(企業)標準やプロジェクト標準に加えなくてもよい、ということ。)

(3) 「共通フレームに書いていないこと」を、組織(企業)標準やプロジェクト標準に追加してもよい。

→ 組織やプロジェクトの特性(例えば二者間契約の内容もその一つ)に合わせて、できるだけ最適と思われる作業の組み立て(「プロセス設計」)を行うために必要な活動が、修整(テラリング)である。…「修整」であり、「修正」ではない!

●もし修整（テラリング）をしなければ、どうなるのか？

- (例) ・ 小規模プロジェクトにとって冗長的な作業項目が含まれ、生産性が低下する。
- ・ 安全性が特に求められるシステムを構築する場合は、品質保証活動（レビュー／検証／妥当性確認／監査など）が不足することとなり、システムの信頼性について確実な確信が持てなくなる。
- だからこそ、組織やプロジェクトの特性に合わせ、適切な修整（テラリング）が必要となる！

9. テーラリング方法（1 / 3）

（1）作業工程を定義する

- ・ 時間軸（管理の区切り）を取り入れて、組織やプロジェクトの作業に必要なプロセス、アクティビティ、タスクを時間軸にマッピングして工程定義を行う。
- ・ 他のプロセス、アクティビティ、タスクとの関連を時間軸（PERT図など）で表現する。

例えば、運用プロセスの移行・準備作業は、開発工程が終了した後の運用工程から始めるのではなく、開発工程内で実施する。

- ・ 各プロセスには、それぞれ「プロセス開始の準備」というアクティビティが定義されているので参照されたい。

9. テーラリング方法（2 / 3）

（2）開発モデルを選択する

- ・開発モデルに依存していないため、プロジェクトの特性に応じた開発モデルを選択し、共通フレームにあるタスクを組み立てる。

例えば、プロジェクト全体では、ウォーターフォールモデルを採用するが、企画・要件定義段階では、繰り返し型や一部プロトタイピング型の開発モデルを使ってシステム化の実現性を調査する。

- ・開発モデルが異なっても、実施するタスクは同じである。どの時点でどう実施するのかの違いである。

9. テーラリング方法（3 / 3）

（3）プロセスの利用者を具体化する。

- ・ 共通フレームは、各プロセスの実施をどういった立場や資質の人間がなすべきかを適用主体者として定義している。
- ・ 実際の利用では、これを参考に組織から利用者を選定する必要がある。

例えば企画プロセスの利用者は企画者であるが、実際の組織に当てはめると、業務部門であったり、企画部門であったりする。

- ・ 誰の責任で実施すべきか、どのタスクを誰がいつ実施すべきかを、組織、プロジェクト、開発モデルの特性に合わせる。

第1部 終わり

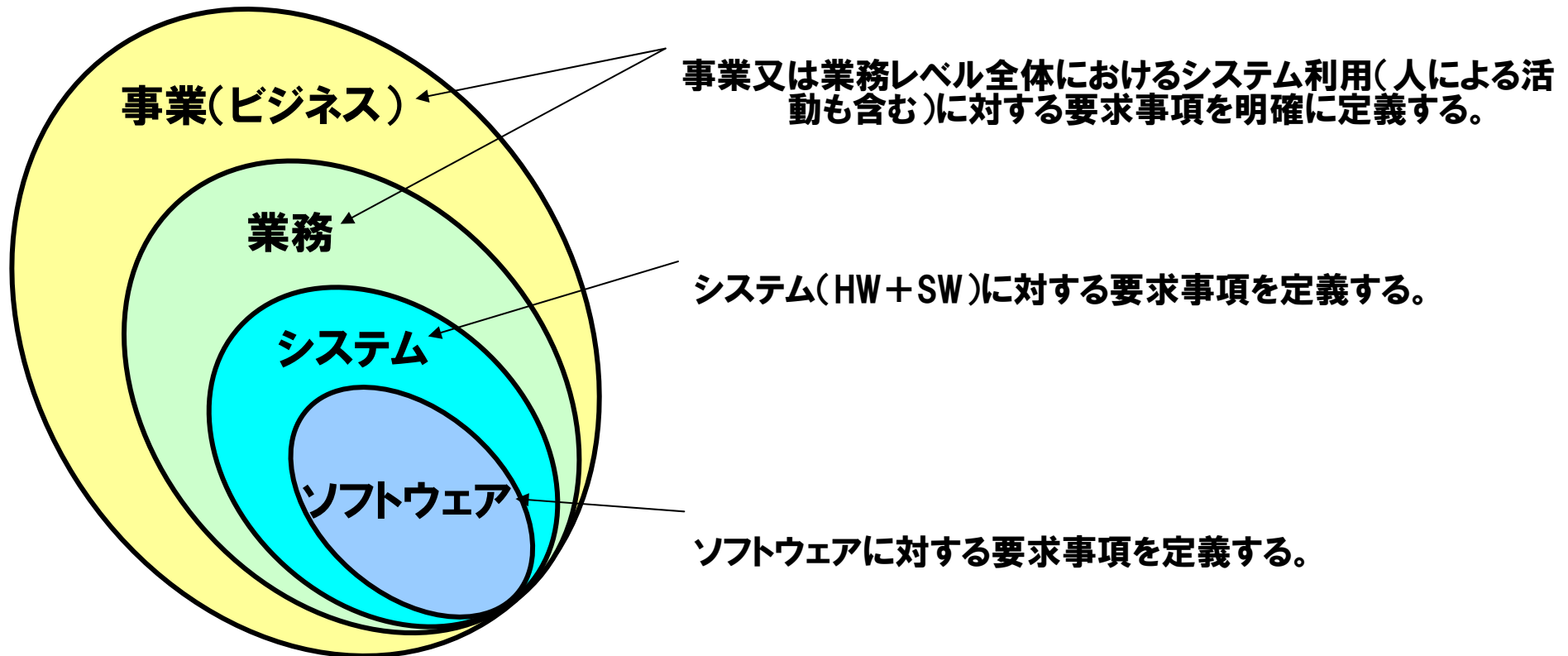
～ 超上流の重視 超上流とは～

1. プロセス拡張のねらい
2. 企画プロセスと要件定義プロセス
3. 超上流とは？
4. もし超上流を軽視したら？
5. 超上流でのトラブルの要因は？
6. 共通フレームに含まれている主な考え方

1. プロセス拡張のねらい

ITシステムは、事業(ビジネス)又は業務で使われるために開発される。

事業／業務における利用目的を明らかにし、その利用目的に応じて、システムに対する要求事項を定義することが非常に重要である。ここを疎かにしてしまうと、利用目的が曖昧となる。結果、「使い勝手の悪いシステム」や「利用されないシステム」等が出来上がってしまう恐れがある。共通フレームはこの考えを導入した。



2. 企画プロセスと要件定義プロセス

●開発に入る前の「要求品質の確保」



システムは、事業(ビジネス)を実現するために開発される。すなわち、開発に入る前の要求品質を確保することが重要になってくる。このため、「超上流」と呼んでいる「企画」「要件定義」のプロセスが追加されたのである。

3. 超上流とは？

■ 「共通フレーム2007」（第1版）の著編者は、その発行前に以下の書籍を刊行している。

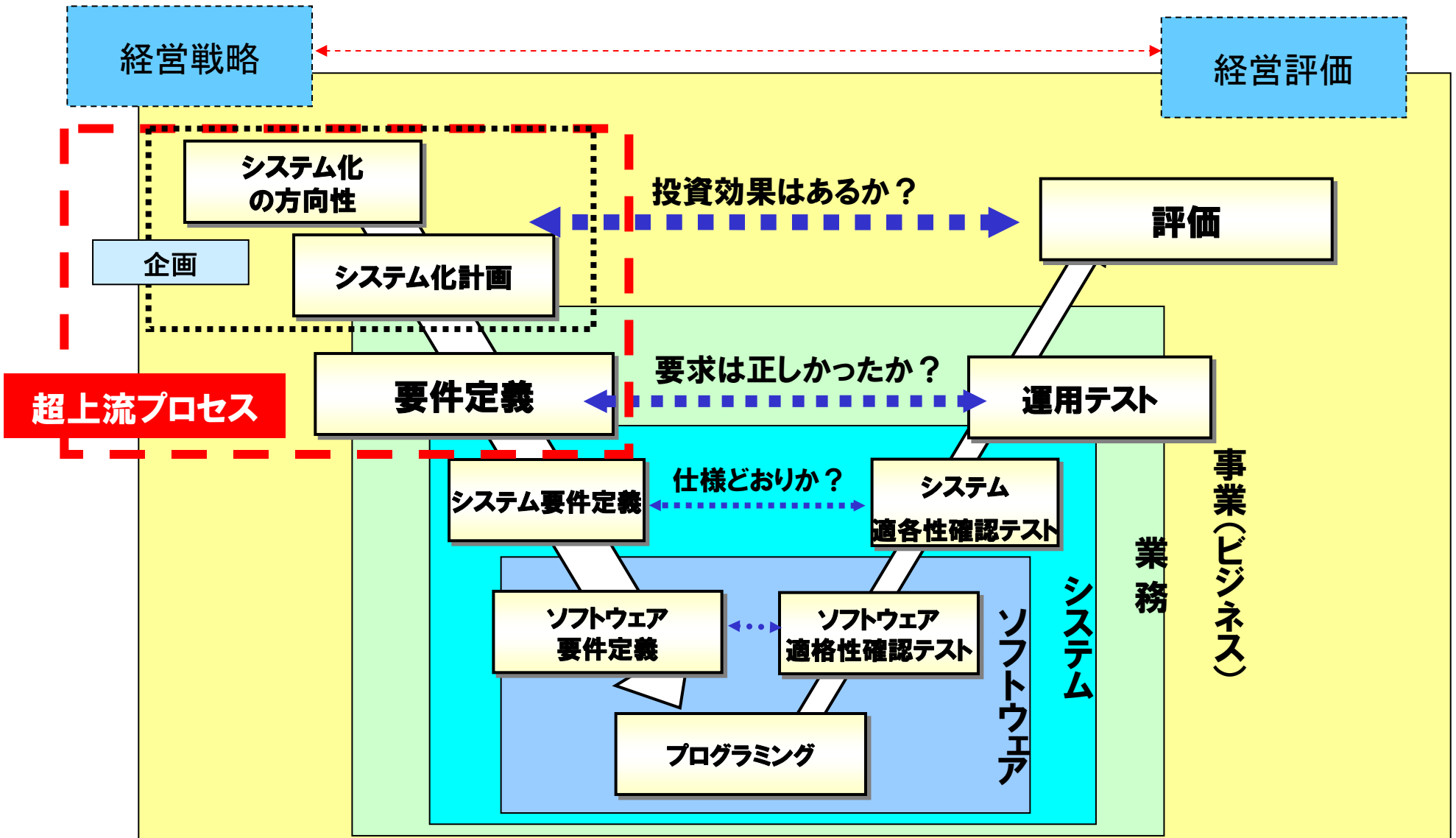
- 「**経営者が参画する要求品質の確保
～ 超上流から攻めるIT化の勘どころ ～**」
（第1版：2005年、第2版：2006年）
⇒ これ以降、本資料では『**超上流の本**』と呼ぶ。

【この本のポイント】

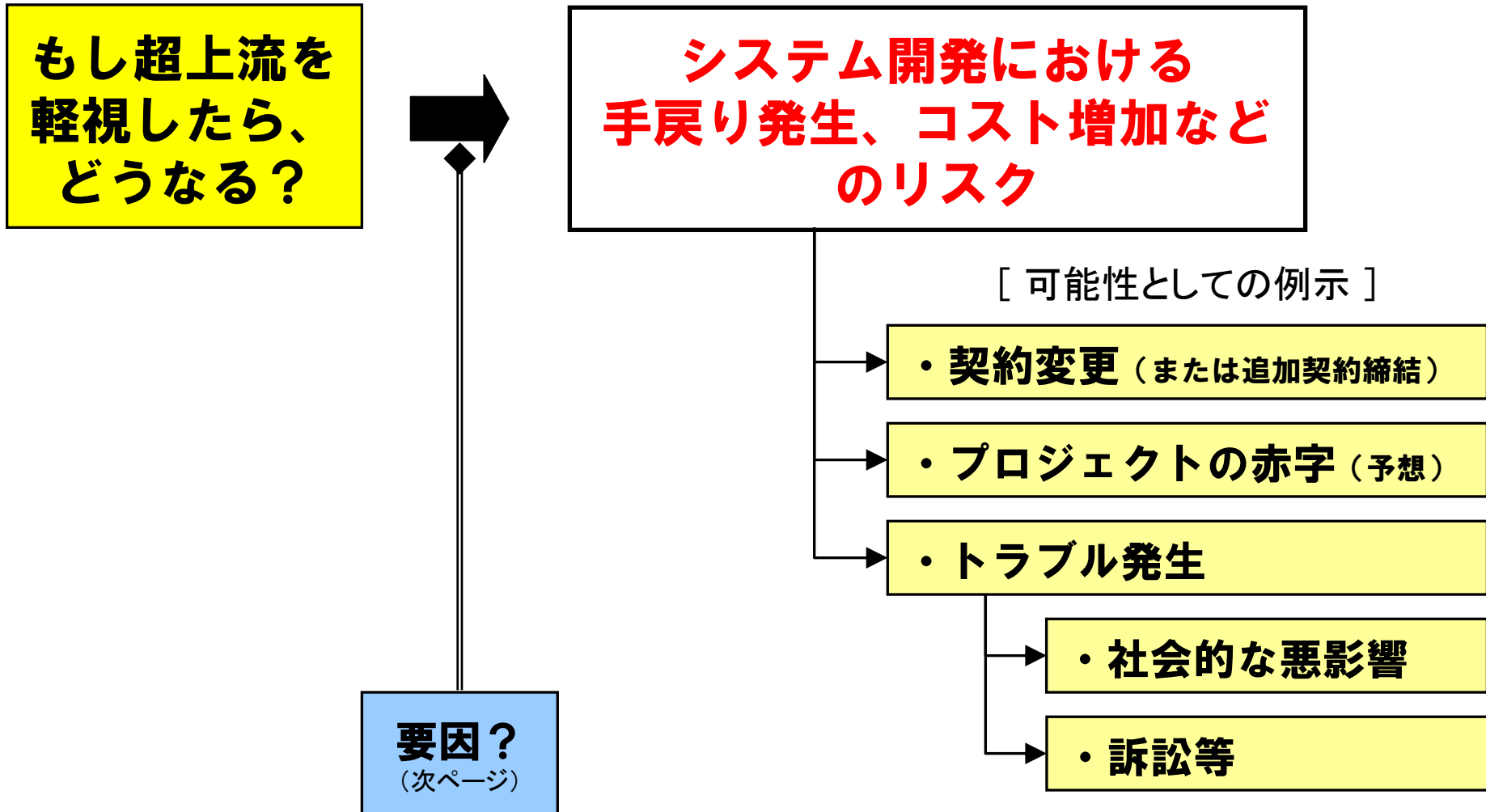
- ① 超上流の重視を説いている。
- ② 経営者の参画を（経営者としての役割があると）説いている。
- ③ 原理原則17ヶ条の活用による問題解決を提唱している。



3. 超上流とは？



4. もし超上流を軽視したら？



5. 超上流でのトラブルの要因は？

- (1) システム化の方針・目的があいまいである。または、周知徹底されていない。
- (2) 要件が固まらない。要件定義書が作成されない。実現可能性について、十分に検討されていない。要件について合意されていない。
- (3) あいまいな見積りのまま、開発段階に入ってしまう、実際の開発規模にズレが生じる(大概は、規模が増大する)。また、要件(機能等)が膨らむ。
- (4) 要件が確定しないのに、次工程に進んでしまう。あるいは、発注者の合意や承認を取らずに、次工程に進んでしまう。

6. 共通フレームに含まれている主な考え方

- (1)「利害関係者の役割と責任分担の明確化」を提唱
- (2)「多段階の見積り方式」を提唱
- (3)「V字モデルの採用」を提唱
- (4)「超上流における準委任契約の採用」を提唱
- (5)「要件の合意及び変更ルールの事前確立」を提唱
- (6)「非機能要件の重要性を認識すること」を提唱
- (7)「運用・保守を含めたSLCPを考えること」を提唱

(1) 「利害関係者の役割と責任分担の明確化」を提唱

【参照先】 『超上流の本』:p.37 の 3.2(1)項、p.41 の 4.1

『共通フレーム』(第2版):p.5 の第1部 2.(1)項、p.8 の同(3)項、p.22 の第2部 5.1(4)項、補足説明集の 1.1及び 1.2

事業要件、業務要件、システム要件を定義できるのは、それぞれ経営層、業務部門、情報システム部門である。それぞれが責任をもって自らの役割を果たすことで、要件を適切に定義できる。

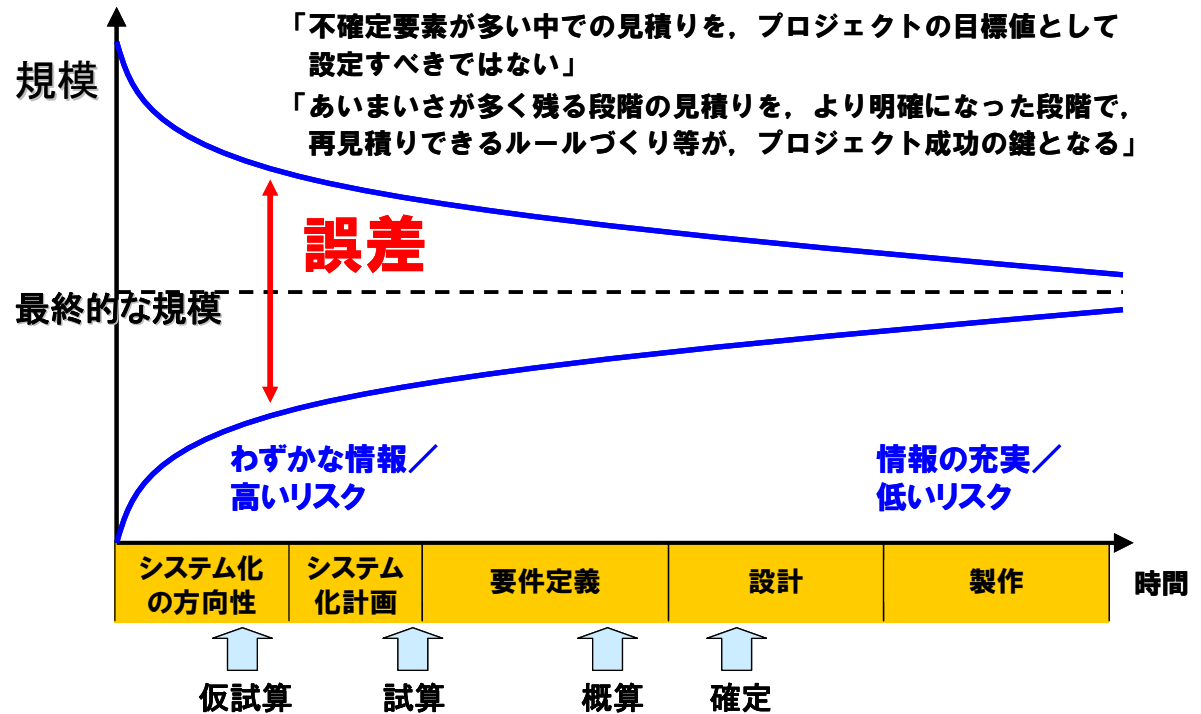
部署等／役割（ロール）		要件の定義内容		
経営層	社長	事業要件定義	業務要件定義	
	担当役員			
業務部門	部門長			
	業務推進担当			
	システム推進担当			
	関連会社			
情報システム部門	部門長			システム要件定義
	システム開発担当			
	システム子会社			
ベンダ	元請けベンダ			
	アウトソーサ			
	サブベンダ			

(2) 「多段階の見積り方式」を提唱

【参照先】 『超上流の本』:p.38 の 3.2(2)項、原理原則15

『共通フレーム』(第2版):p.9 の第1部 2.(6)項

わずかな情報で見積ること自体、リスクが高い。それ故、それだけで、プロジェクトの目標としてはならない。



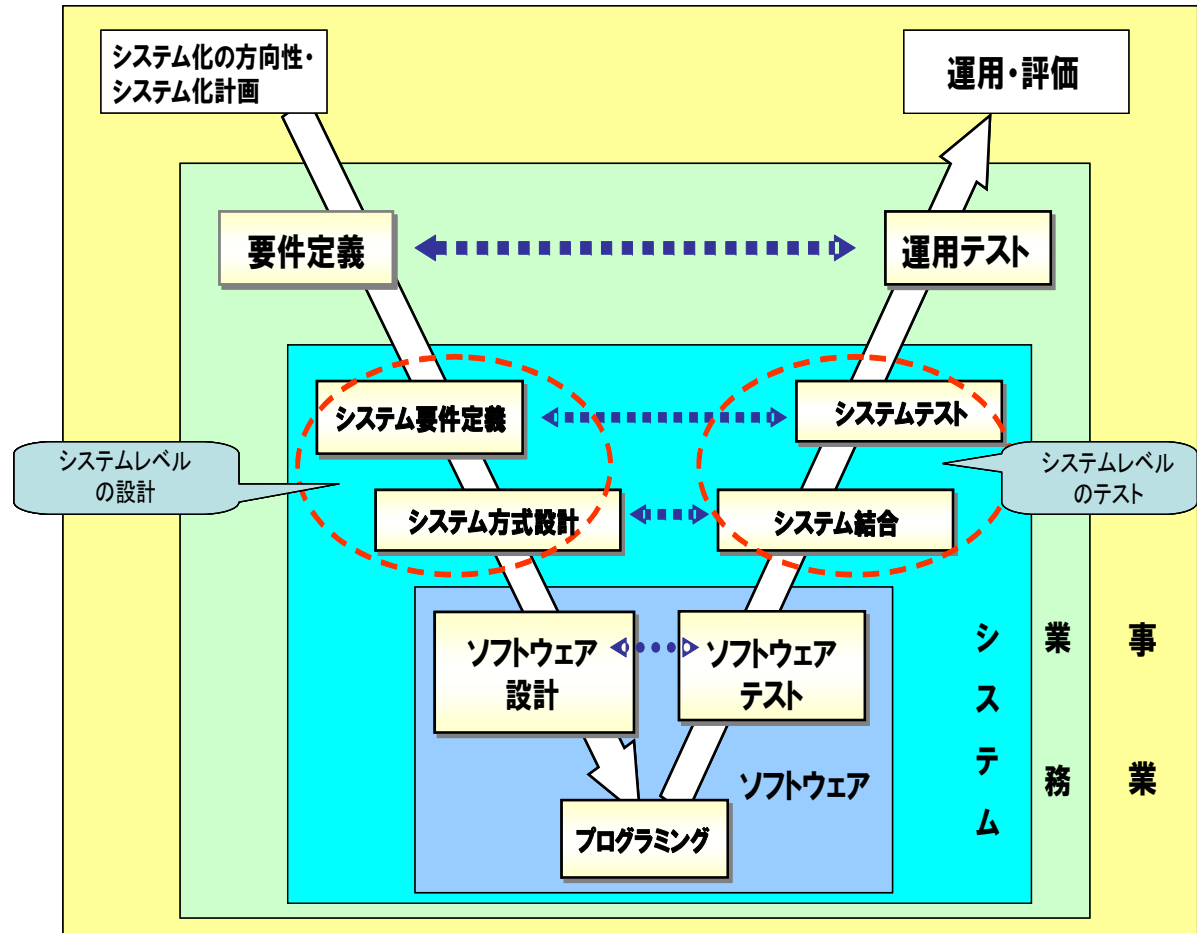
※SEC BOOKS「経営者が参画する要求品質の確保 ～超上流から攻めるIT化の勘どころ～ (第2版)」より引用・一部改修

(3) 「V字モデルの採用」を提唱

【参照先】 『超上流の本』:p.24 の 図2.3

『共通フレーム』(第2版):p.22 の 図2-2

設計(品質の埋め込みプロセス)とテスト(品質の検証プロセス)とを対応させることにより、プロダクト品質を確保する。



(4)「超上流における準委任契約の採用」を提唱

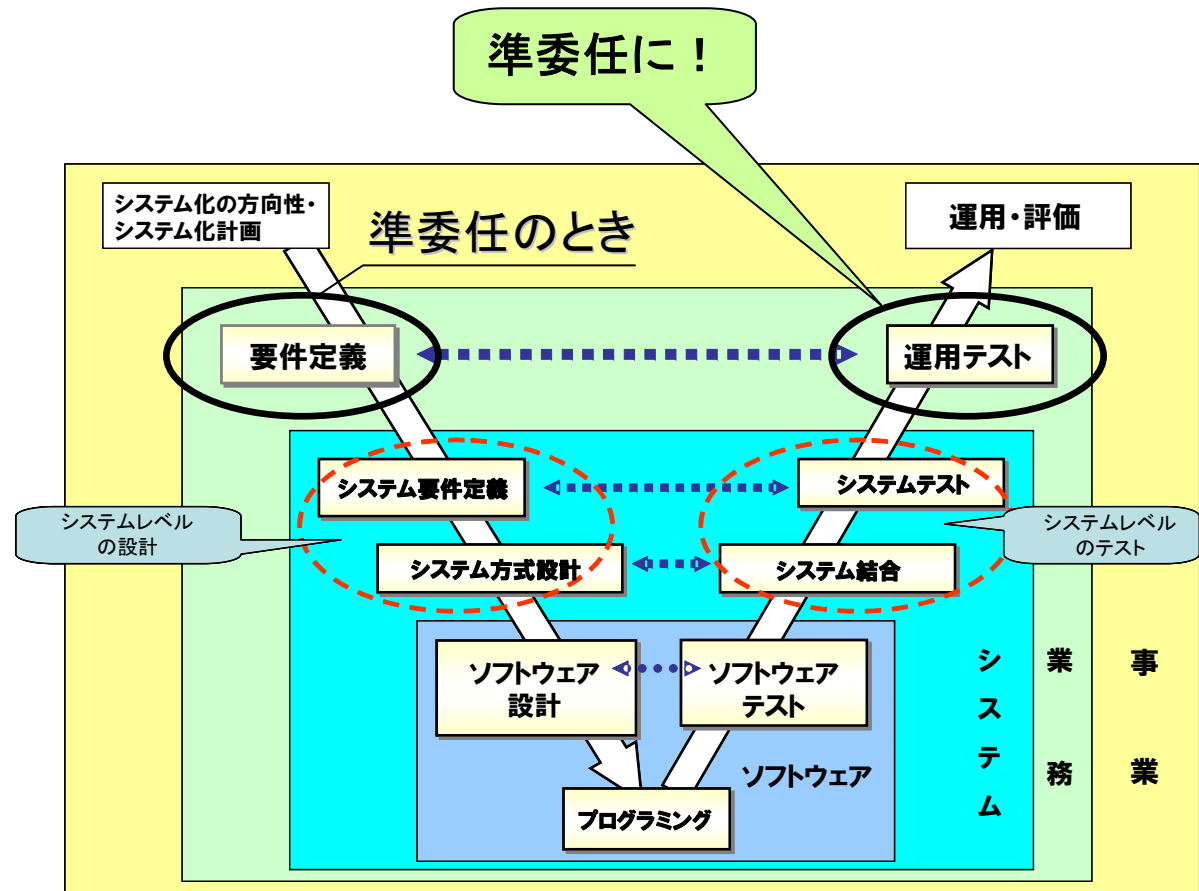
【参照先】 経済産業省「情報システムの信頼性向上のための取引慣行・契約に関する研究会」報告書
”～情報システム・モデル取引・契約書～”（2007年4月13日 公表）

『共通フレーム』（第2版）：p.9 の第1部 2.(5)項、p.20 の「(注)サービスメニュー」の項

超上流は、基本的には、ユーザ責任であるため、ベンダにとって準委任契約とするのが合理的である。(もし請負契約にすると、ユーザの事情に大きく影響されるため、リスクが大きい)。

【例】

- ・超上流 → 準委任ならば
- ・運用テスト → 準委任に
- ・ソフトウェア開発 → 請負



(5)「要件の合意及び変更ルールの事前確立」を提唱

【参照先】 『超上流の本』:p.39 の 3.2(4)項

『共通フレーム』(第2版):p.8 の第1部 2.(4)項、p.10 の同(7)項、p.35 の第2部 5.4.2(1)(c)項

ソフトウェア開発においては、時の経過に伴って「要件は変わるもの」であり、ユーザとベンダとが事前にルールを策定し合意(確定)しておかないと、いざトラブルが発生した時に、速やかな対応が取れない。

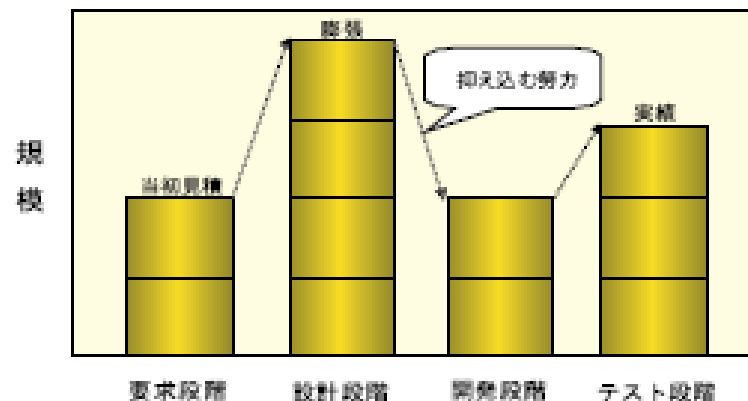


図 2.9 2-4-2-3の法則

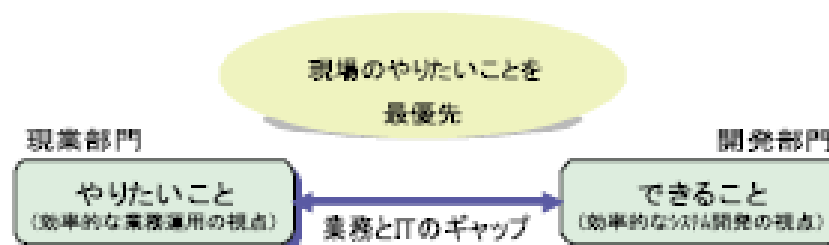


図 2.10 規模が膨張する力学

【出所】 『超上流の本』p.31 より。

(6)「非機能要件の重要性を認識すること」を提唱

【参照先】 『超上流の本』:p.11 の 1.3(3)②項、p.39 の 3.2(3)項

『共通フレーム』(第2版):p.113 の第3部 1.5.2.5(非機能要件の定義)とそのガイダンス

運用テストの段階に至って、問題をもたらす要因は、機能要件のみならず、むしろ深刻な事態になりがちな非機能要件の方であるため、早い段階で「非機能要件の重要性」を認識し、何かしらの対応策を講じることが望ましい。

●機能要件 とは

システムに実装する機能に関する要件のこと。

●非機能要件 とは

運用要件、移行要件、性能要件、セキュリティ、機密情報保護対策など、機能要件以外の要件のこと。

【出所】 『超上流の本』 p.43 より。

【注意】 業務部門(システムの利用部門等)にとっては、業務要件こそが重要である。
なお、業務要件に、機能要件、非機能要件も含まれる。

(業務要件については、『共通フレーム』(第2版):p.112 の第3部 1.5.2.2 参照)

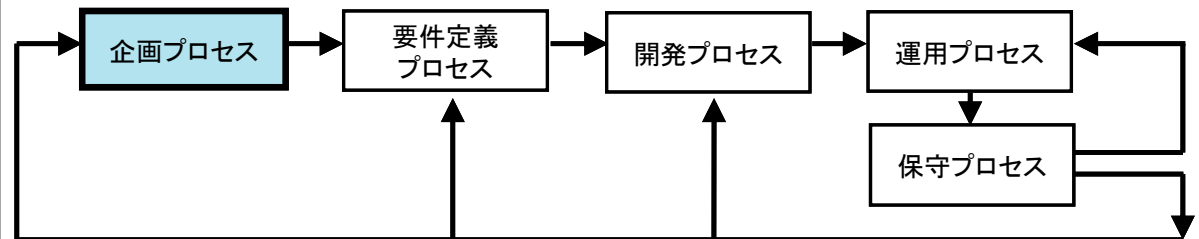
(7)「運用・保守を含めたSLCPを考えること」を提唱

【参照先】 『超上流の本』:原理原則6、7

『共通フレーム』(第2版):p.11~12 の第1部 2.(8)(9)(12)項

(注) "SLCP": システムライフサイクルプロセスの略記。また、ソフトウェアライフサイクルプロセスの略記でもある。

システムは生きもの。作って終わりではない。顧客との取引が継続する限り、または事業や業務が続く限り(ITシステムを必要とする限り)、システムライフサイクル全般に目配せしてシステム化計画(企画)や要件定義を行うことが、結局は、適正コストで「使えるシステム」を実現できる。



第2部 終わり

(1) 共通フレーム2007の概要を説明しました。

(例) 共通フレーム2007は、例えば、情報処理技術者試験でも基礎知識として取り入れられています。

(2) システム開発又は保守などの組織(企業)標準を整備する上で、IT 専門家の見識が詰まった『共通フレーム2007』を参照すると有用です。

(3) システムを調達する場合、特定ベンダの開発方法論に依存していない『共通フレーム2007』の「言葉」を使用し、RFP(提案依頼書)や調達仕様書等を記述することによって、中立性を確保することができます。

(4) 『超上流の本』を一読して頂き、ご活用することを強く推奨します。共通フレーム2007に込められた考え(ねらい)を理解する上で大変役立つと共に、IT化の勘どころとして、単独に使用しても有用です。

ご清聴ありがとうございました