

失敗しない要件定義とリスク対策 ～システム(再)構築上流工程の強化に向けて～

ソフトウェアジャパン2017 IPAセッション
2017年2月3日

独立行政法人情報処理推進機構 (IPA)
技術本部 ソフトウェア高信頼化センター (SEC)
山下博之

<http://www.ipa.go.jp/sec/index.html>

ICTの役割は、“Support Business”から“Do Business”へと変わってきている。すなわち、今やICTはビジネスの要となっている。さらに、ビジネスのグローバル化が今後一層加速する。このような状況で競争優位を維持し続けるためには、ICTシステムのユーザは、ビジネス環境の変化に機敏に対応すると共に、自ら、そのICTシステムに、ビジネス要求を的確に反映することが必須である。

一方、システム構築の上流工程に起因するトラブル(開発プロジェクトの失敗、サービス開始後のシステム障害の発生)が相変わらず多い。上流工程に起因するトラブルを無くすためには、ユーザ自らが要件を定義し、下流工程でのリスクに関するユーザ／ベンダ双方の合意の上でシステム(再)構築を進めることが必要である。

本講演では、これらの背景と目的のもとにIPA/SECで作成した、「ユーザのための要件定義ガイド」及び「システム再構築を成功に導くユーザガイド」について解説する。

1. IPA/SECにおけるシステム構築上流工程強化の活動
 - 背景と課題
 - 取組み概要
2. ユーザのための要件定義ガイド
3. システム再構築を成功に導くユーザガイド
4. まとめ

1. IPA/SECにおけるシステム構築上流工程強化の活動

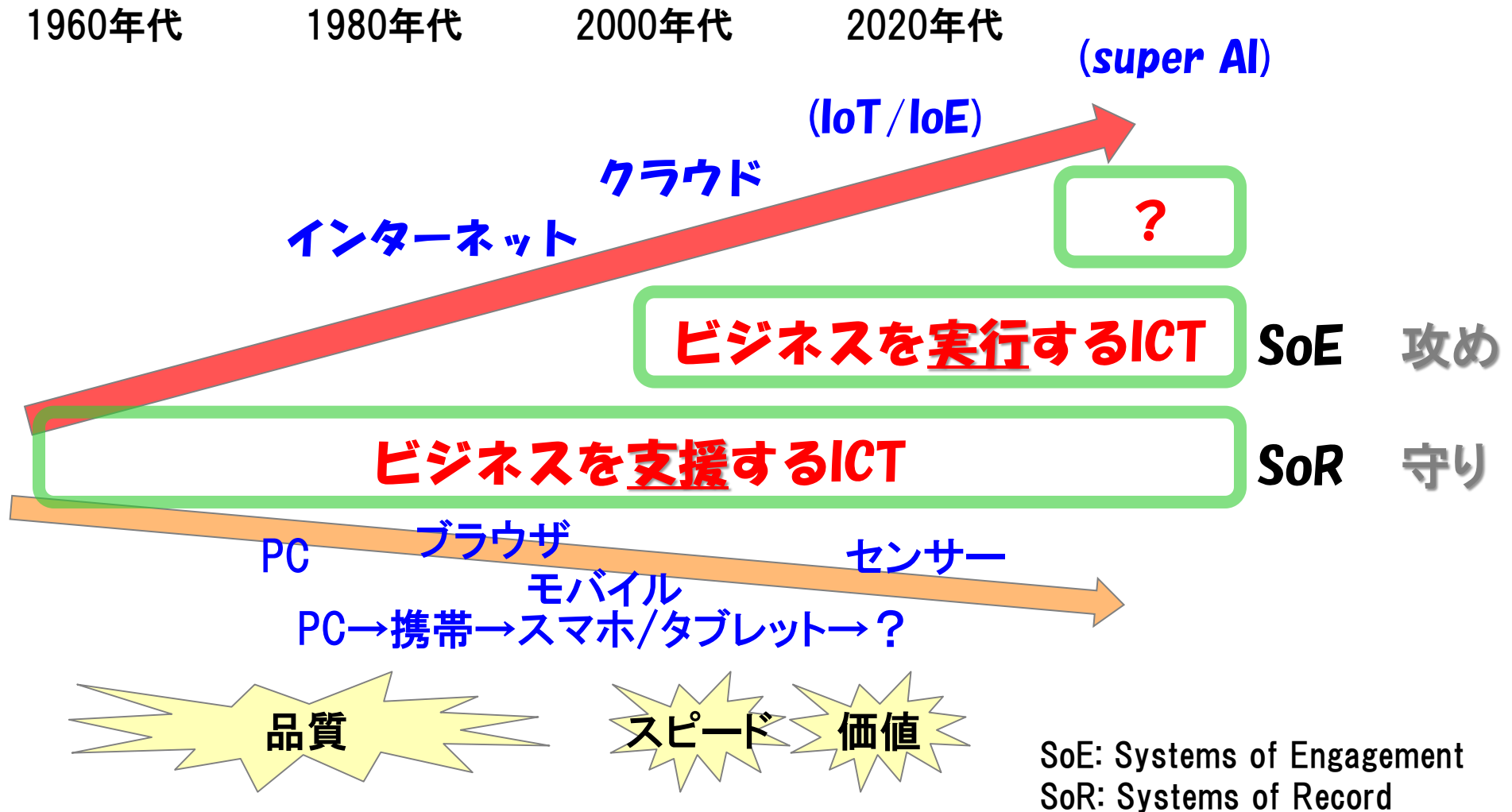
- 背景と課題
- 取組み概要

2. ユーザのための要件定義ガイド

3. システム再構築を成功に導くユーザガイド

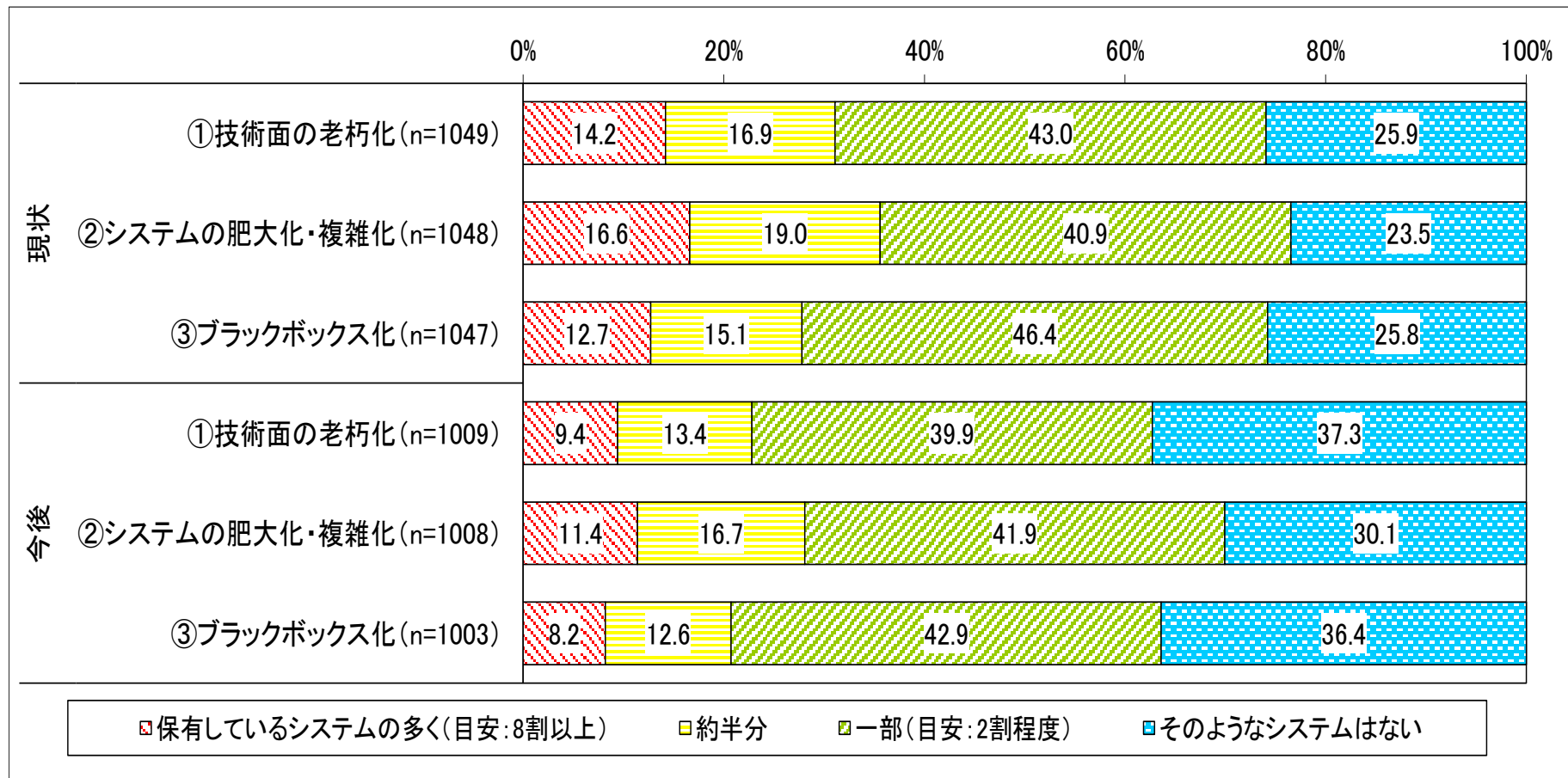
4. まとめ

ICTシステムの役割が、ますます重要に



背景(2) レガシー基幹系システムの保有状況

**システムの半数以上がレガシー化：現状3割・今後2-3割の企業
(2割以上がレガシー化しているのは：現状7割・今後6-7割の企業)**



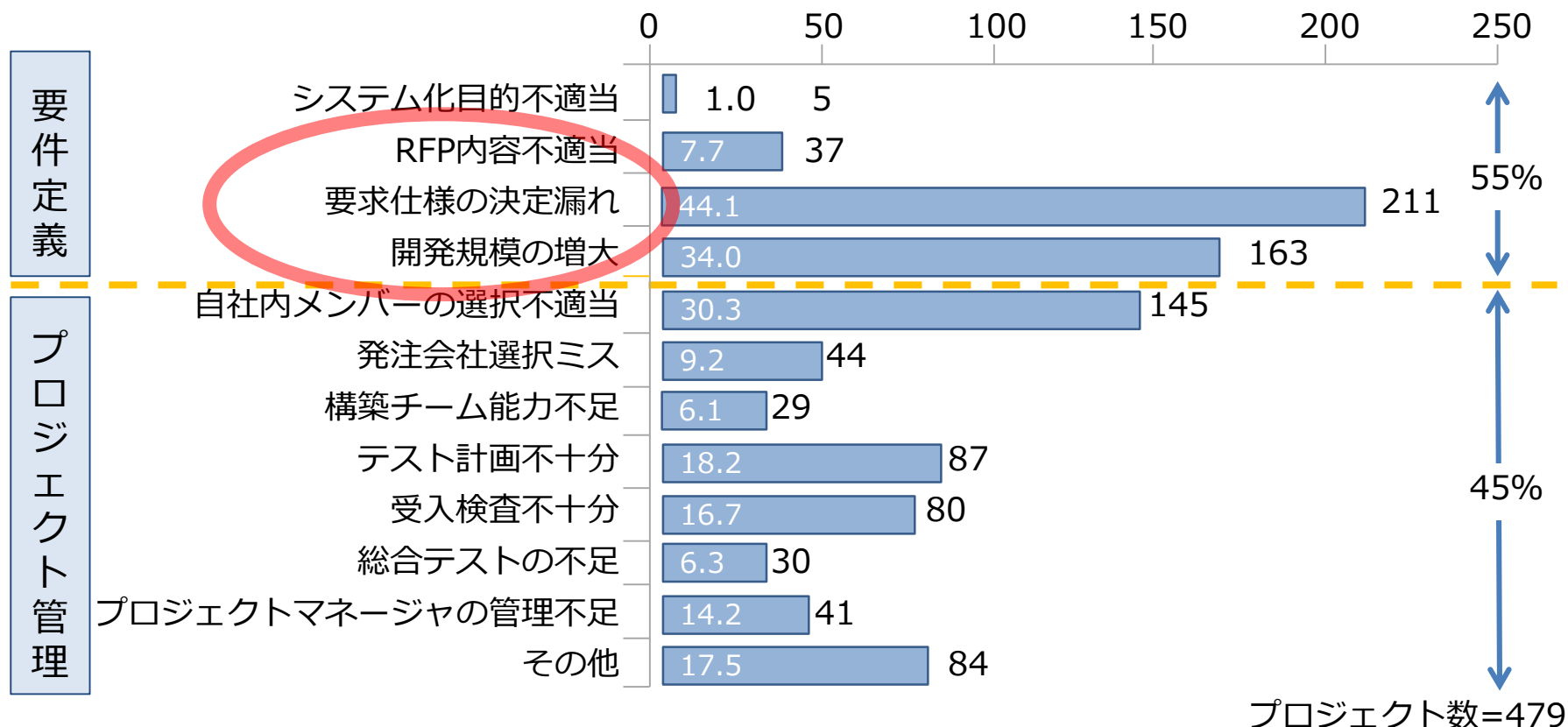
＜出典＞ JUAS・企業IT動向調査2016

ICTの役割は、従来の“Support Business”から、今日では“Do Business”へと変わってきている。すなわち、今やICTはビジネスの要となっている。

また、日本では今後、人口減少に伴うマーケットの縮小が進むものと思われる。その中で企業が生き残るためには、世界中の企業と競争しなければならない。すなわち、ビジネスのグローバル化が加速する。

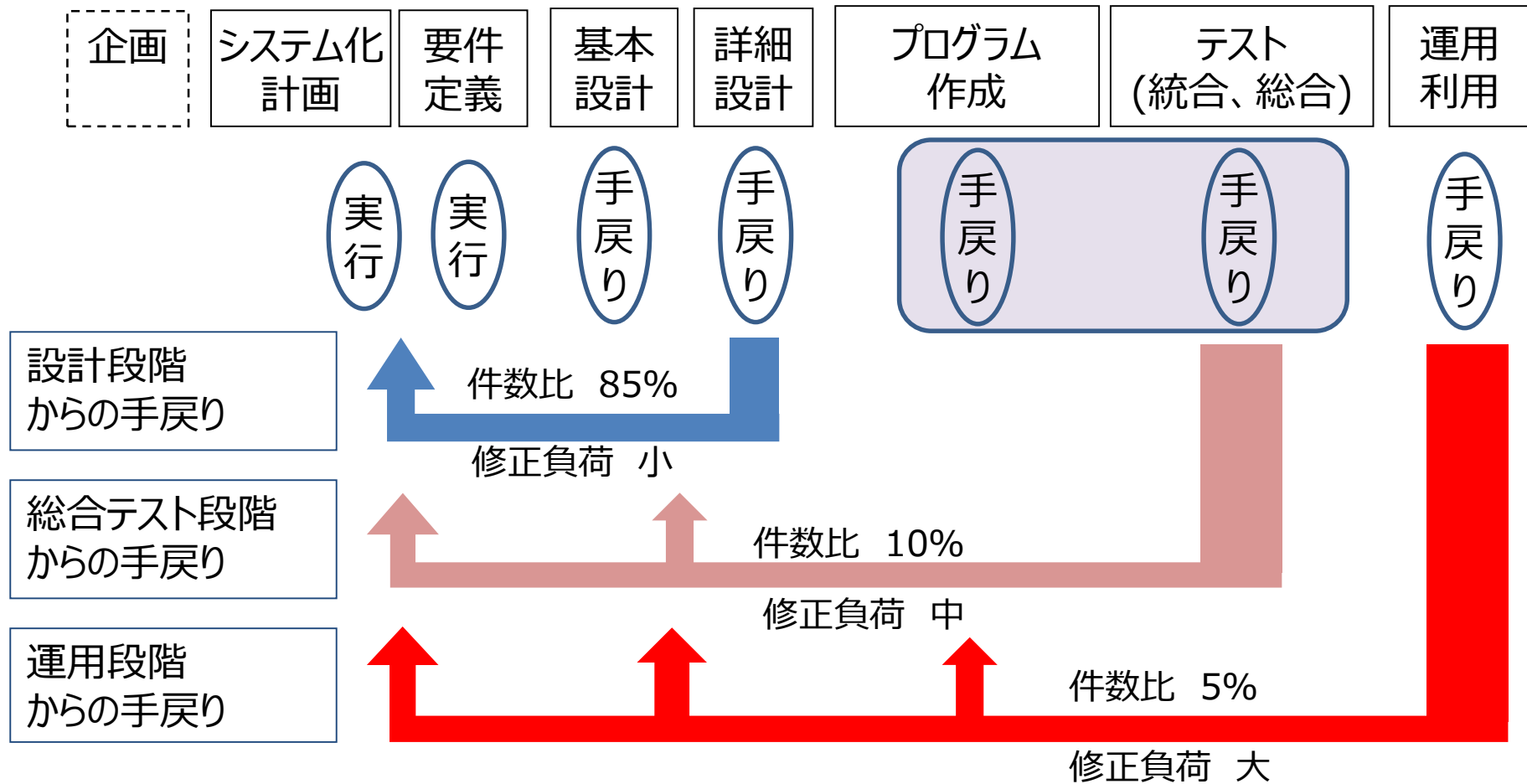
このような状況において競争優位を維持し続けるためには、ICTシステムのユーザは、着実にモダナイゼーションを進めつつ、ビジネス環境の変化に機敏に対応し、自ら、そのICTシステムに、ビジネス要求を的確に反映することが必須である。

工期遅延理由は、全体の55%が、要件定義の問題 (残いの45%がプロジェクト管理の問題)



<出典> JUAS・ユーザー企業ソフトウェアメトリックス調査2016

後工程に行くほど、修正負荷(手戻りのコスト)が増大



<出典> JUAS・ソフトウェアメトリクス調査2014のデータから作成

システム構築の上流工程に起因するトラブル(開発プロジェクトの失敗、サービス開始後のシステム障害の発生)が相変わらず多い。

従来、ユーザ企業は、ICTシステムの要件定義(の相当部分)をベンダ企業に任せるケースが少なくない。また、ユーザ／ベンダ間で後工程でのリスクに関する十分な認識と合意の無いままにシステム構築を進めるケースが、特にシステム再構築において、見られる。

上流工程に起因するトラブルを無くすためには、ユーザ自らが要件をきちんと定義し、下流工程でのリスクに関する双方の合意の上でシステム構築を進めることが必要である。

今回設定した 取組み対象

上流工程に関わる問題



要件定義に関する問題による手戻り

システム再構築に特有な問題による手戻り

主な課題

テーマ

問題

要件定義

ユーザから
開発者に
要求(What)が
正しく
伝わらない

システム
再構築

現状を踏まえた
再構築に伴う
リスクの認識・
合意が不十分

解決する
ために

取り組むべき課題

正しく伝えるための「How」を整理する

※上流における要求からどのように
要件を決め、表現するか

※要件定義の不備(抜け、漏れ、あいまい等)を
無くし、成果物の品質を上げる

「リスクと合意形成の手段」を整理する

※再構築に特有の難しさを踏まえた
工期・コスト・品質への影響を見える化

※要件定義着手前の企画・計画段階で
下流におけるこれらの影響について合意

※リスクを減らすための下流での施策
(やるべきこと)を明確化

上流工程における対策としての活動概要と活動テーマ(目的)

IPA/SEC内にシステム構築上流工程強化部会及び同部会配下にシステム化要求WG、モダナイゼーションWGを設置し、ユーザ系及びベンダ系の企業専門家や経験者による議論と作業の場を設け、議論。その結果をガイドブックとして取りまとめる。

活動概要

システム構築上流工程強化部会



システム化要求WG



モダナイゼーションWG



ユーザ企業、ベンダ企業の経験者を中心に構成

活動テーマ(目的)

要件定義工程に起因する手戻り防止

システム開発の共通的内容を解決する

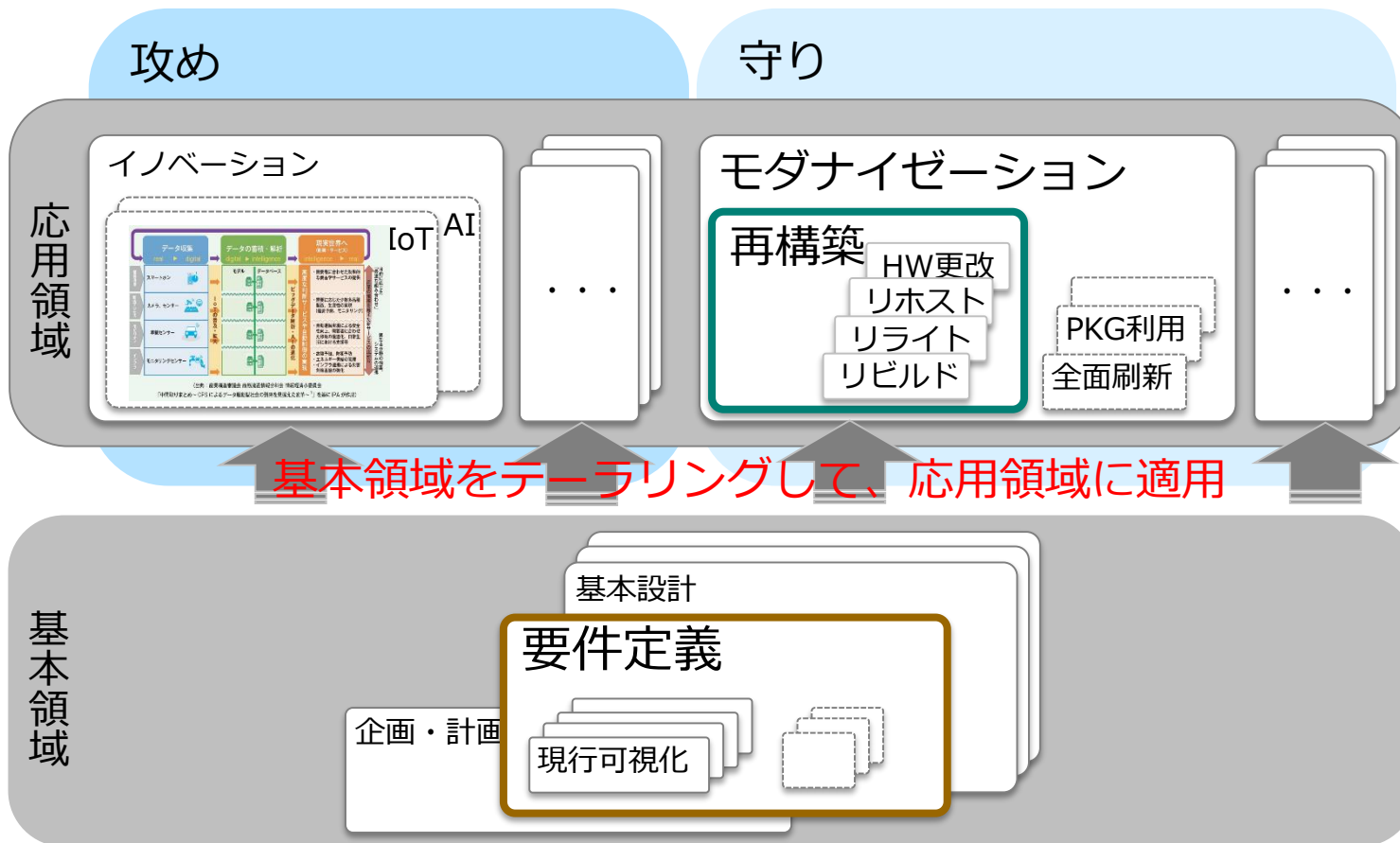
- ・きちんと現行業務を見て、要件として定義する
- ・業務要件とシステム要件の整合性をとる

相互協力

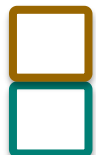
再構築に特有の内容を解決する

- ・リスクを明らかにして、そのリスクを合意する
- ・リスクに対する対応を計画に盛り込む

再構築に関わる問題に伴い発生する手戻り防止



【凡例】



：「ユーザのための要件定義ガイド」のスコープ

：「システム再構築を成功に導くユーザガイド」のスコープ

ユーザのための 要件定義ガイド

～要求を明確にするため
の勘どころ～

IPA Better Life
with IT

2017年1月31日 pdf版公開

<http://www.ipa.go.jp/sec/reports/20170131.html>

2017年3月10日
書籍発行予定

システム再構築を 成功に導くユーザガイド

～ユーザとベンダで共有する
再構築のリスクと対策～

IPA Better Life
with IT

1. IPA/SECにおけるシステム構築上流工程強化の活動

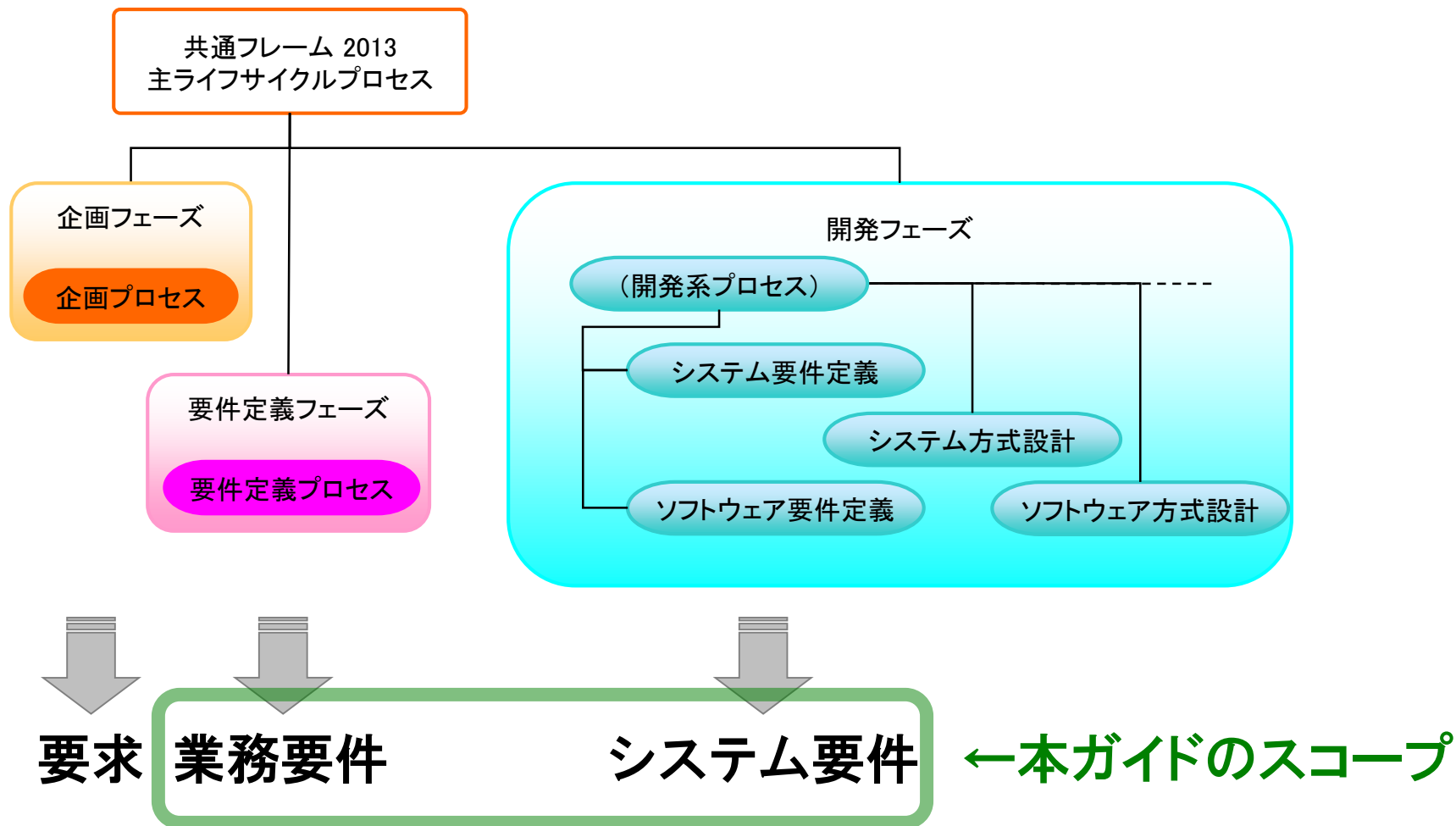
- 背景と課題
- 取組み概要

2. ユーザのための要件定義ガイド

3. システム再構築を成功に導くユーザガイド

4. まとめ

まず、要件定義や前後の工程の目的を理解する

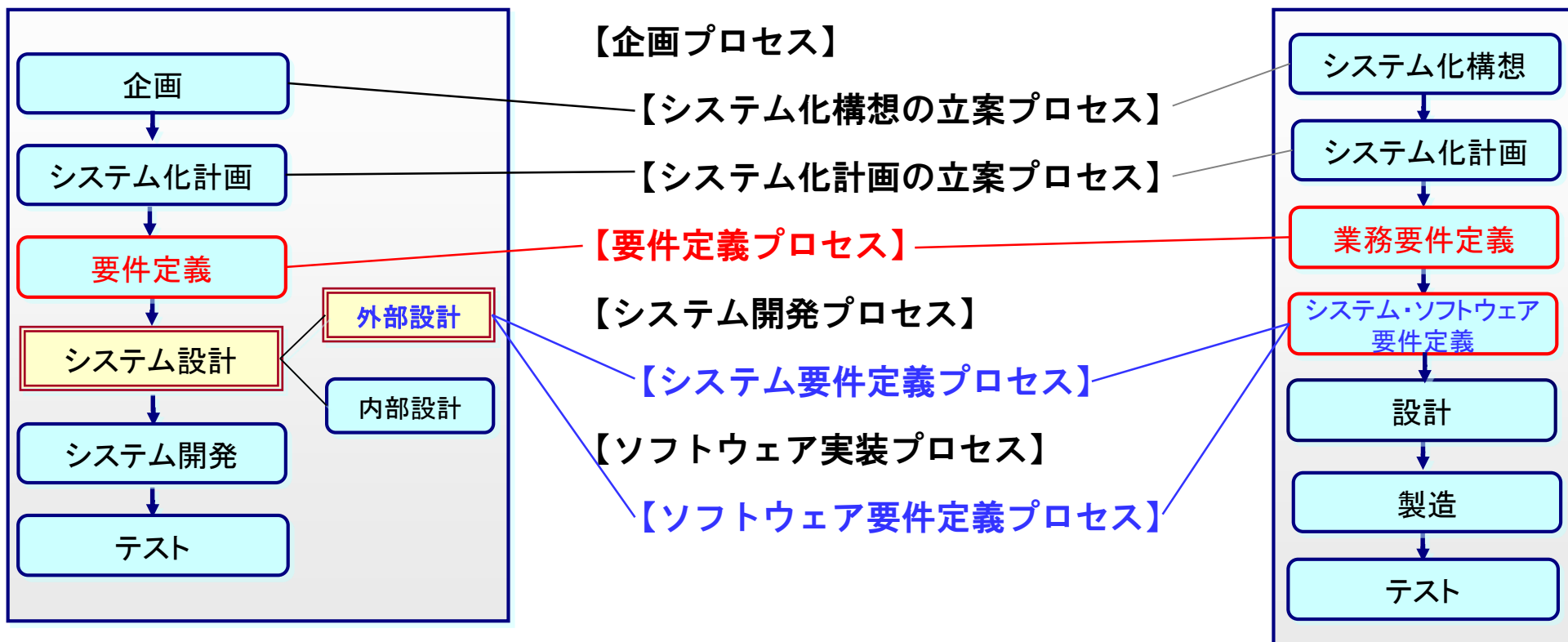


プロジェクトの状況や目的に応じて、工程にプロセスを配置

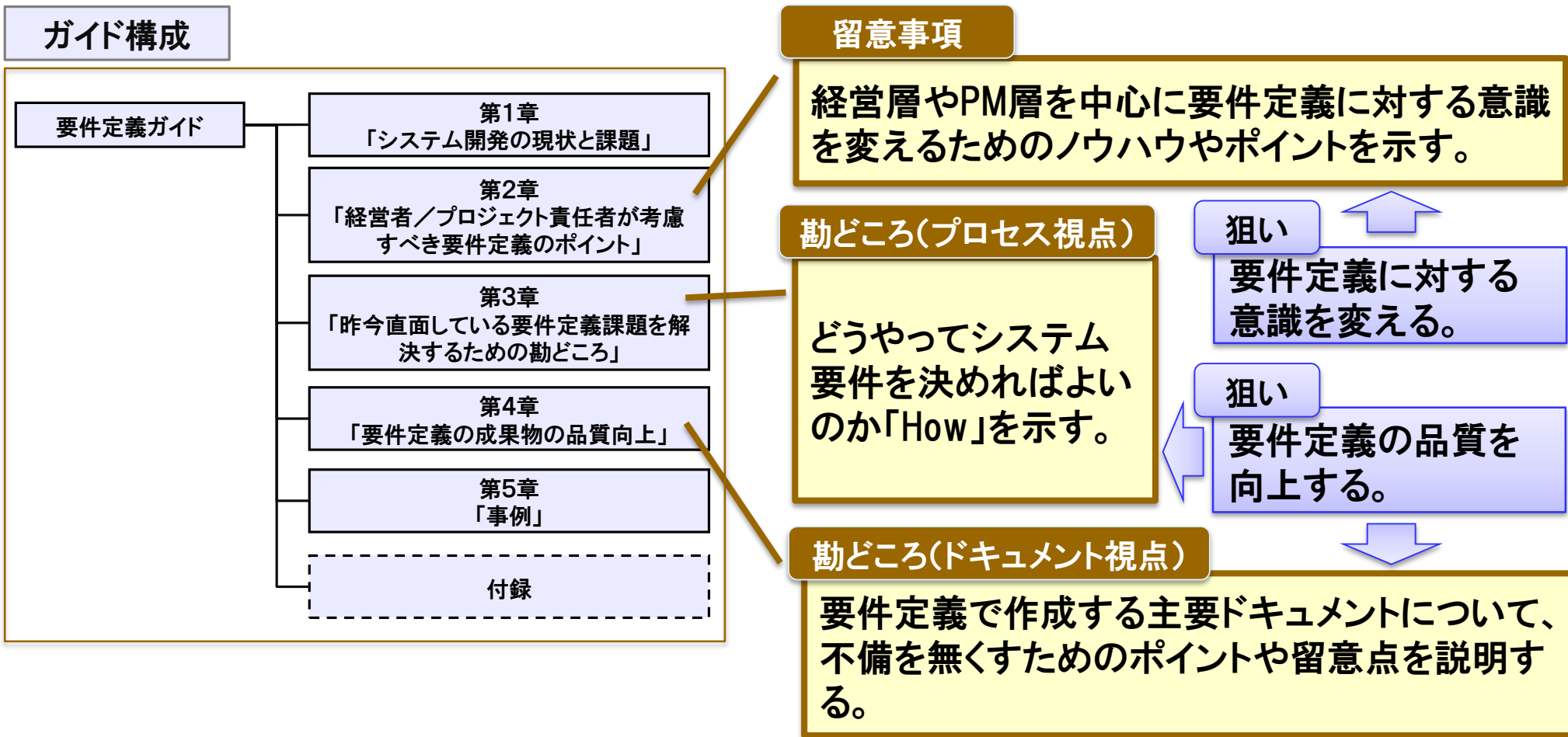
【工程の例－1－】

【共通フレームにおけるプロセス】

【工程の例－2－】



ガイドの構成と、中心となる留意事項と勘どころの内容と狙い



要件定義ガイドの内容(何を)

主なターゲット(誰に)

(※注)
 ・責任を持つステークホルダを表す
 ・他のステークホルダも協力する

留意事項

要件定義に対する意識を変える

ノウハウ・ポイント



ノウハウ・ポイント

勘どころ

プロセス視点

要件定義を不備なく行う

業務要件

システム要件



業務要件
定義活動

システム要件
定義活動

ドキュメント視点

要件定義ドキュメントの品質を上げる

業務要件

システム要件



業務要件
ドキュメンテーション

システム要件
ドキュメンテーション

ユーザ企業

ベンダ企業



経営者
/PM



業務
部門



システム
部門



ベンダ

はじめに

第1章 システム開発の現状と課題

- 1.1 システム開発の現状
- 1.2 日米のIT投資が抱える課題
- 1.3 要件定義を巡る課題
- 1.4 本ガイドの構成

第2章 経営者／プロジェクト責任者が考慮すべき要件定義のポイント

- 2.1 当該システムを開発する妥当性を検討する
- 2.2 システム化の前に業務を分析・整理する
- 2.3 要件定義を確実に効率よく進める
- 2.4 決めるべきことは決めてから次へ進む
- 2.5 ユーザとベンダの役割分担を考える

第3章 昨今直面している要件定義課題を解決するための勘どころ

- 3.1 経営や業務に貢献するITシステムの構築
- 3.2 膨らむ要求のコントロール
- 3.3 業務の複雑性を軽減
- 3.4 要件定義工程からの非機能要件定義
- 3.5 多様化するステークホルダとの合意形成
- 3.6 現行業務やシステムの把握

第4章 要件定義成果物の品質向上

- 4.1 成果物の作成計画時の留意点
- 4.2 主要な成果物と作成上の留意点
- 4.3 成果物に共通の留意点

第5章 事例

- 5.1 (サントリーシステムテクノロジー株式会社)
- 5.2 (株式会社NTTデータ)
- 5.3 (富士通株式会社)
- 5.4 (富士通株式会社)
- 5.5 (セイコーエプソン株式会社)
- 5.6 (株式会社ジャステック)
- 5.7 (株式会社ジャステック)
- 5.8 (株式会社NTTデータ)

おわりに

付録1 共通フレームが規定する要件定義関連のプロセス

付録2 システム要件定義のコツをまとめた「機能要件の合意形成ガイド」

参考文献

留意事項（第2章）



委員から提起された要件定義に関わる課題

- ① ビジネス目的と施策が合致していない
- ② 手段が先行し、「何のために」が理解できていない
- ③ 業務の複雑さが増している
- ④ 合意形成が取れていない
- ⑤ 膨らむ要求を抑えきれない
- ⑥ 要件定義書の不備が多い
(抜け, 漏れ, 曖昧, 不完全, 不整合などへの対策が不十分)
- ⑦ 非機能要件を決めきれない
- ⑧ 要件定義の記述の粒度や深さの基準が不明、内容の評価ができない
- ⑨ As-Isの分析, To-Beの可視化が不十分
- ⑩ 業務部門の参画、理解が不十分
- ⑪ システム部門が要件を引き出せない
- ⑫ 体制, 役割分担の不備での失敗

1. 当該システムを開発することの妥当性を検討する

- 経営課題に対して適切な位置付けのIT投資であるか
- 経営者の視点での問題認識と対策を十分に盛り込む
- 技術進歩を予測し、システムを開発するタイミングは妥当か?
- リスク評価を踏まえた非機能要件について、ユーザの要望を明確にする
- 他社との競争を意識した場合、工期は妥当か?

2. システム化の前に業務を分析・整理する

- 業務部門の責任者が現在のシステムの実態を正しく理解し、要件定義に反映しているか?
- 全社視点で見て、重複作業や無駄を十分に排除したシステムになっているか?

3. 要件定義を確実に効率よく進める

- 合意形成に時間がかかることを前提としてプロジェクトを計画する
- 要求は膨らむことを前提としてプロジェクトを計画する

4. 決めるべきことは決めてから次に進む

- 要件定義工程で決めるべきことを決めるために十分な工期、工数を割り当てる
- 要件定義成果物の品質向上のための課題

5. ユーザとベンダの役割分担を考える

- 要件定義の契約基本形は準委任契約である
- システムの効果を評価する責任は業務部門である
- 仕様変更が生じた場合を想定した予算確保



プロセス視点の勘どころ（第3章）

経営や業務に貢献するITシステムの構築(3.1)


□経営方針やビジネス目的との関係を明確にすべし

- ・目的と手段の違いを意識せよ
- ・目的を体系化せよ
- ・目的と手段を関連付けせよ
- ・目的の重要性を意識せよ

□曖昧な目的を具体化すべし

- ・真の目的を明確にせよ 
- ・目的と目標の違いを意識せよ 
- ・目的に目標を設定せよ
- ・稼働時の目標を明確にせよ

□曖昧な手段を具体化すべし

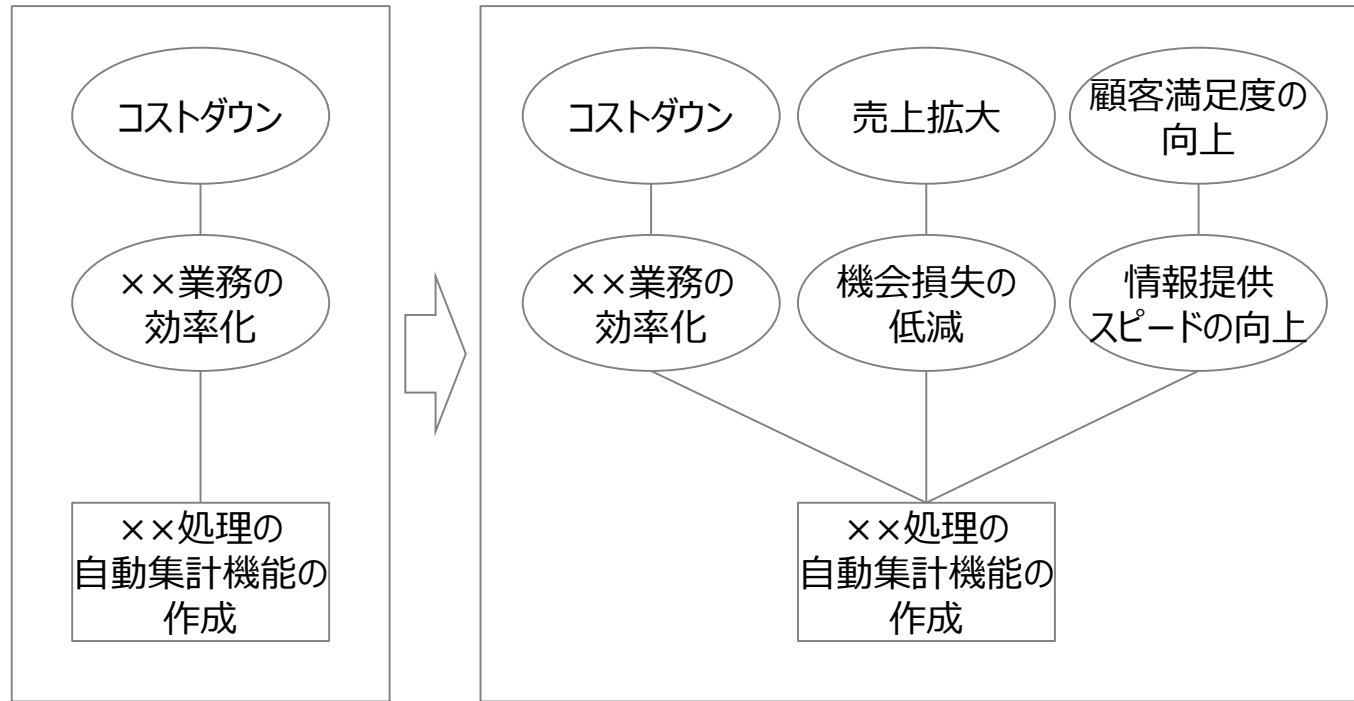
- ・手段の充分性を検討せよ
- ・手段を選択するための優先順位を明確にせよ 

□真の問題を捉えるべし

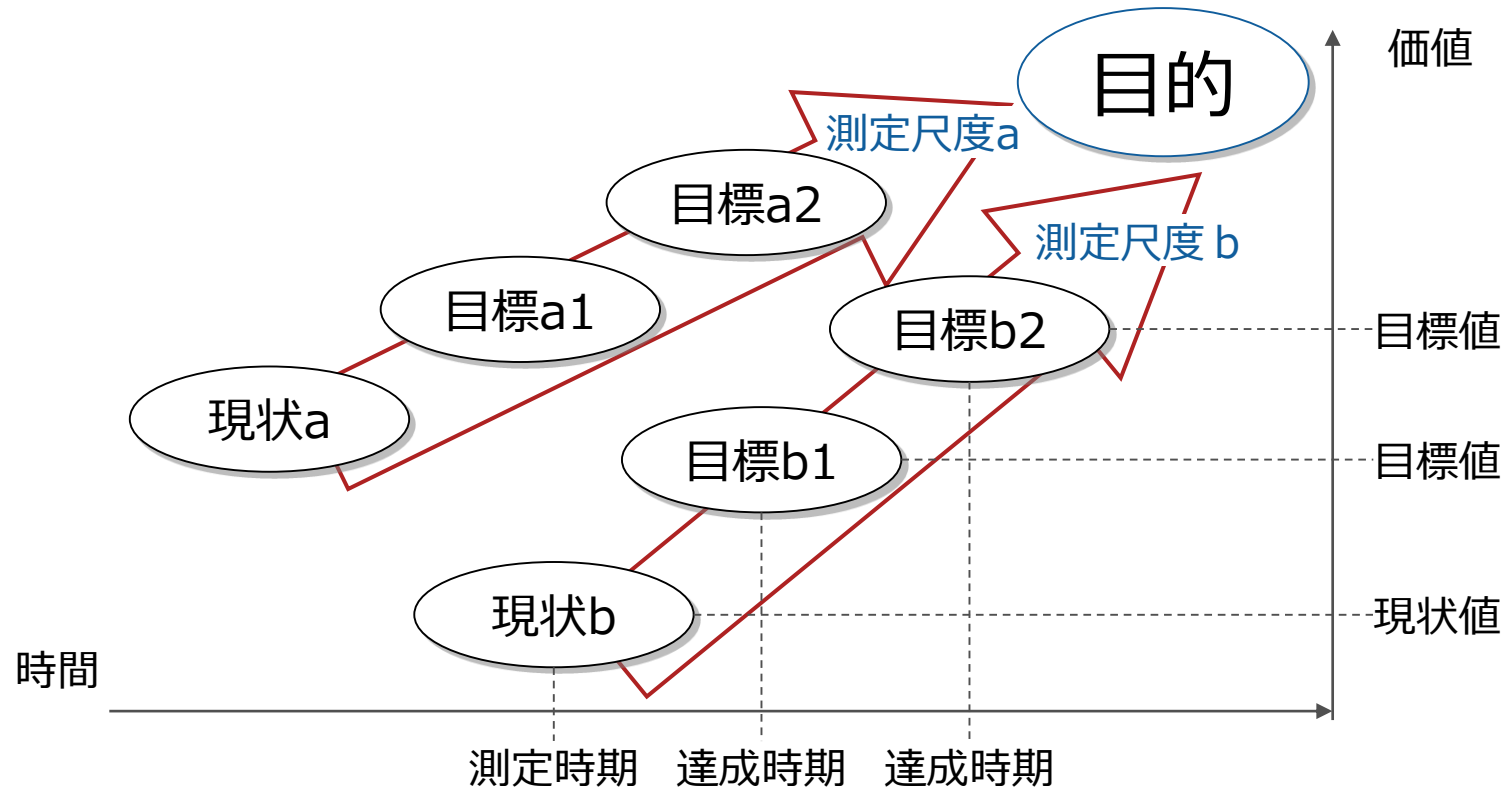
- ・要求の源泉を分析せよ
- ・問題と課題の違いを意識せよ
- ・なぜなぜ分析を行い真の原因を見極めよ
- ・課題/解決テーマを適切に設定せよ



目的を精査した結果のイメージ

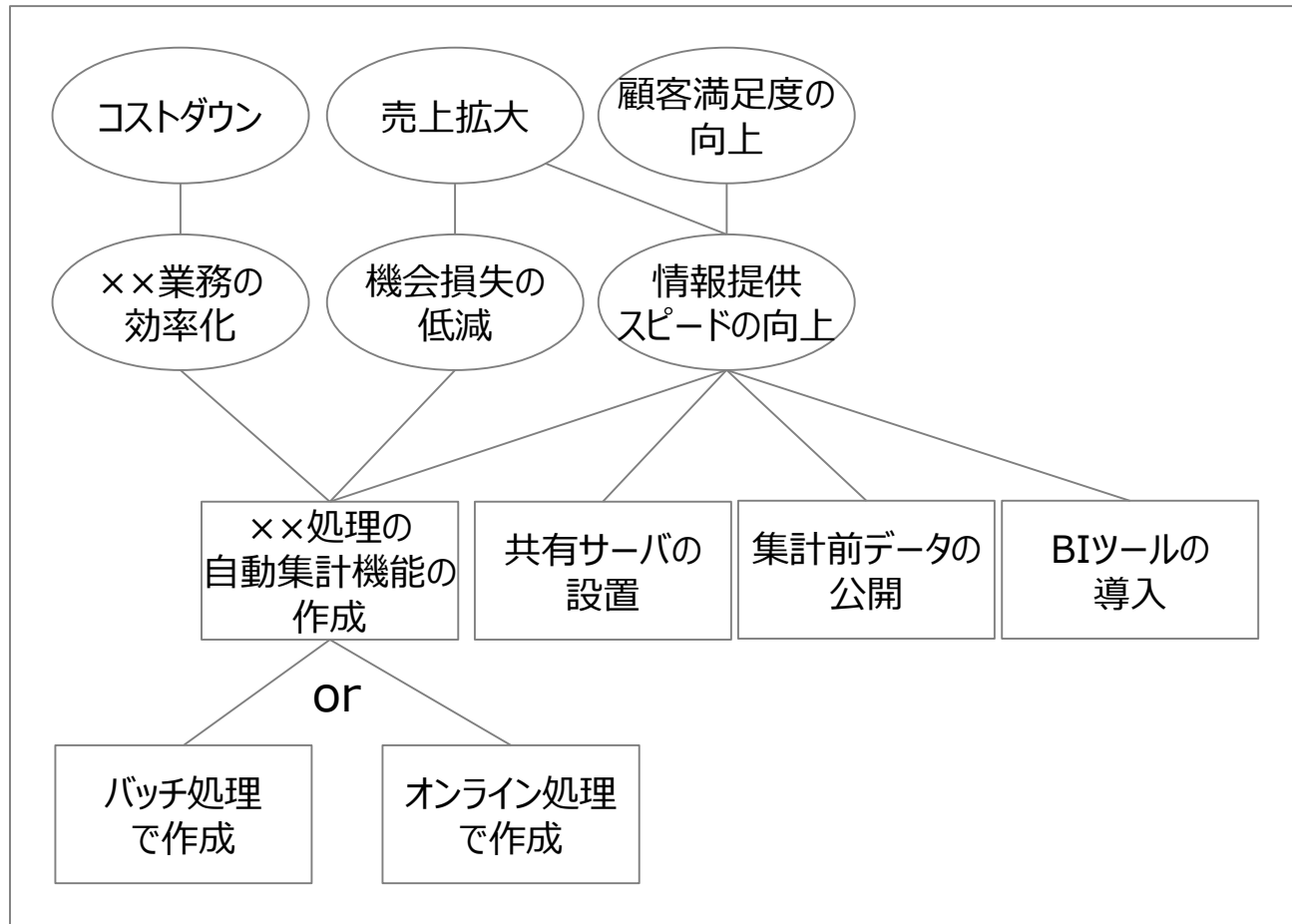


目的が曖昧な状態から、本当の目的を聞き出す



目的に対し目標を設定

目標とは目的を達成するためのマイルストーン



目的を明確にし、手段を明確にし、関係を押さえる

→曖昧だった要求がより具体化

膨らむ要求のコントロール(3.2)

- 要求が膨らみシステム開発が失敗する
- 要求を100%満足した100点のITシステムより60点を早くリリースして良くする時代
- 価値ある要求を見極める
- 過剰要求を捨てる英断

□要件定義内のフェーズを意識して要求をコントロールすべし

- ・膨らませるフェーズと絞り込むフェーズを使い分けよ

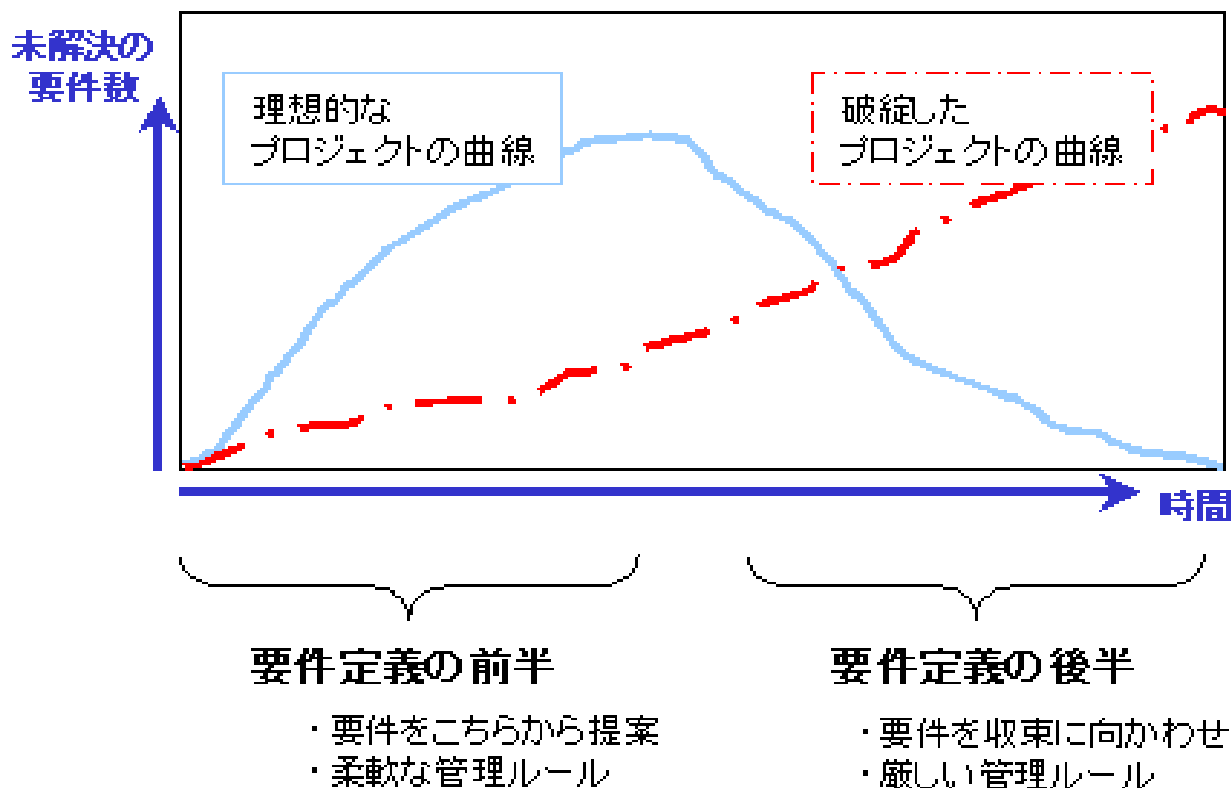


□要求の漏れと過剰要求を是正すべし

- ・要求間の関連を見て充分性を判断せよ
- ・要求の価値を検討して妥当性を判断せよ
- ・過剰要求を捨てよ

□定量化した要求量の中で要求を調整すべし

- ・要求を定量化せよ
- ・ベースラインを合意してから要件定義を進めるべし



要件定義の前半で膨らませた要求も、後半では絞る

業務自体の複雑性を軽減(3.3)

□管理対象のバリエーションを整理すべし

- 管理対象の種類を整理せよ
- 漏れなくダブリなく整理せよ
- 名前の無い分類に名前を付けよ
- 用語の定義として共通認識せよ
- 企業文化に依存するので注意せよ
- 分類の廃止を検討せよ
- 分類の統合を検討せよ

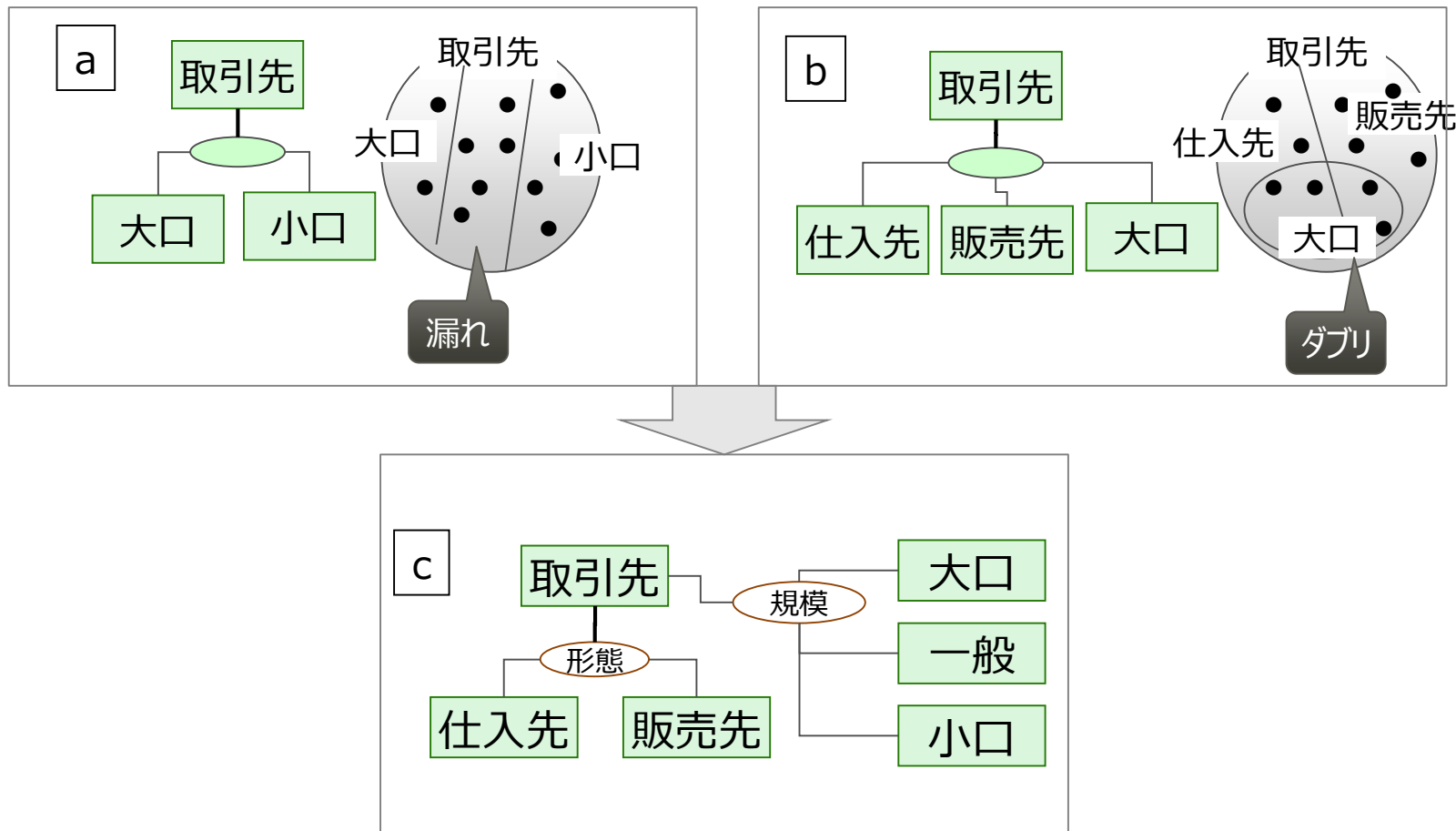
□業務のバリエーションを減らすべし

- 業務プロセスの廃止を検討せよ
- 業務プロセスの集約を検討せよ
- 業務プロセスの連結を検討せよ
- 業務プロセスの並行化を検討せよ
- 一連の業務処理パターンの削減を検討せよ

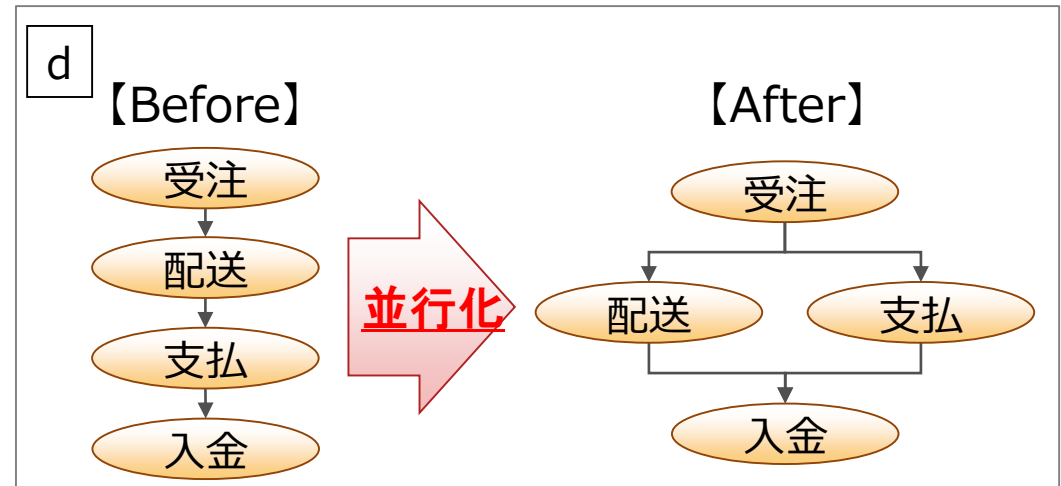
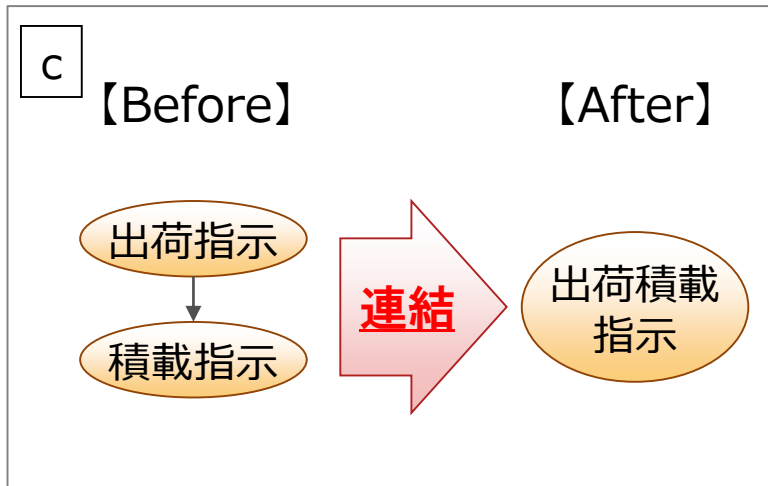
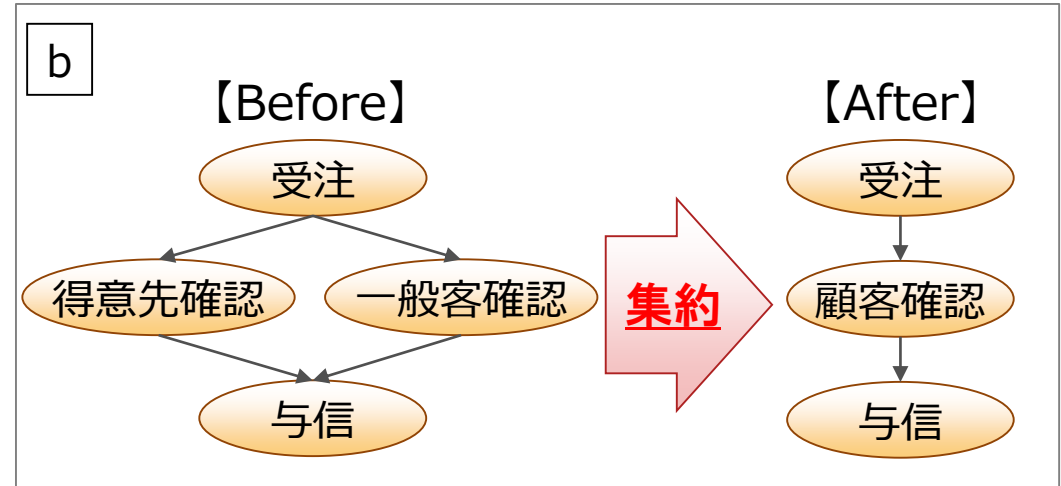
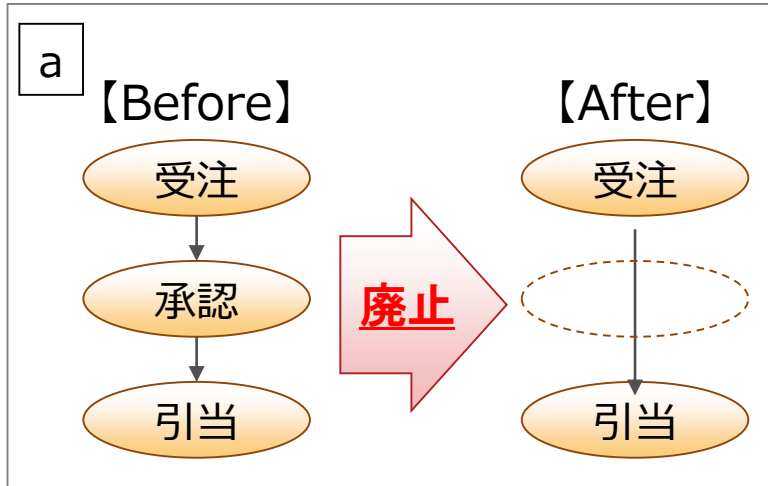
□業務パッケージを適切に利用すべし

- カスタマイズ量に注意せよ
- 業務をパッケージに合わせるという意識を強く持て
- 業務パッケージ導入に対する現場の反対をコントロールせよ
- 経営トップが参画せよ
- 再度業務パッケージの導入の是非を判断せよ





相互に重ならないがなく、全部集めて漏れがないように、まとめる
ミッシー (MECE: Mutually Exclusive and Collective Exhaustive)



業務プロセスや業務パターンを整理し、複雑さを減らす

一連の業務処理のパターンの可視化例



シナリオ名	頻度	商品					顧客					オーダー形態			業務の流れ										
		有形	サービス		単品	セット	代理店	法人		個人	国内	海外	一般	レンタル	無償	顧客確認	見積り	在庫引当	出荷指示	直送	出荷	配送	請求	...	
			定額	従量				大口	小口																
パターン1	30	●			●		●				●					①		②	③		④	⑤	⑥		
パターン2	20	●			●			●			●					①	②	③	④		⑤	⑥	⑦		
パターン3	2	●			●				●	●	●					①		③	④	⑤		⑥	②		
パターン4	1	●			●				●	●	●					①		③	④	⑤		⑥	②		
...

業務処理パターンの統合・削減の前に、表で整理する

要件定義工程からの非機能要件定義(3.4)

□非機能要求グレードを活用すべし

□基本的な情報を要求として引き出すべし

- ・非機能グレードの項目をステークホルダ別に分けよ
- ・聞き出す要求を明確にする

□矛盾、トレードオフを解消すべし

- ・制約や前提条件を明確にせよ
- ・現状を正確に把握せよ
- ・優先順位を判断せよ
- ・要求のレベルでトレードオフ関係を判断せよ
- ・一律でメトリクスを決めるな
- ・実際に体感してもらえ

□関心が高い非機能要求設定に関するポイント

- ・業務継続性
- ・セキュリティ
- ・ユーザビリティ
- ・性能

<参考>

非機能要求グレード

<http://www.ipa.go.jp/sec/softwareengineering/std/ent03-b.html>

非機能要求グレードのイメージ



モデルシステム

大項目	小項目	小項目説明	レベル					備考	達成レベル		達成条件		達成レベル		達成条件		達成レベル	達成条件	
			0	1	2	3	4		5	達成レベル	達成条件	達成レベル	達成条件	達成レベル	達成条件				
可用性	可用性	システムが稼働している状態を維持すること。							2	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%
	可用性	システムが稼働している状態を維持すること。							2	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%
	可用性	システムが稼働している状態を維持すること。							2	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%
	可用性	システムが稼働している状態を維持すること。							2	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%
	可用性	システムが稼働している状態を維持すること。							2	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%
	可用性	システムが稼働している状態を維持すること。							2	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%
	可用性	システムが稼働している状態を維持すること。							2	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%
	可用性	システムが稼働している状態を維持すること。							2	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%
	可用性	システムが稼働している状態を維持すること。							2	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%
	可用性	システムが稼働している状態を維持すること。							2	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%
	可用性	システムが稼働している状態を維持すること。							2	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%
	可用性	システムが稼働している状態を維持すること。							2	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%
	可用性	システムが稼働している状態を維持すること。							2	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%
	可用性	システムが稼働している状態を維持すること。							2	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%
	可用性	システムが稼働している状態を維持すること。							2	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%
	性能・拡張性	性能・拡張性	システムが稼働している状態を維持すること。							2	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%
運用・保守性	運用・保守性	システムが稼働している状態を維持すること。							2	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%
移行性	移行性	システムが稼働している状態を維持すること。							2	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%
セキュリティ	セキュリティ	システムが稼働している状態を維持すること。							2	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%
システム環境・エコロジー	システム環境・エコロジー	システムが稼働している状態を維持すること。							2	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%	可用性99.9%

「項目一覧」
236個のメトリクス

まず、16個の
特徴項目から定義

ビジネス領域の非機能要求抽出シートの例



要求の種類	確認項目
業務特性／サービス時間	サービス提供先(企業内、企業間取引、コンシューマ向けなど)
	サービス提供範囲(特定地域、国内、国外)
	サービス提供時間(通常・特定日 24時間365日)
	サービス特性(繁忙期・閑散期、月間、年間、年度)
	サービス利用者数
	サービス提供拠点数、利用端末数(人/端末)
業務継続	サービス継続範囲(対象業務範囲、計画停止有無、代替業務運用範囲)
	業務停止の許容度(単一障害、多重障害)
業務連携	提携先企業とのシステム連携有無
	提携先企業とのシステム連携方法や条件
業務量	サービス提供業務・取扱量
	サービス拡張指針(利用者数、対象業務)
運用方針／内部統制	サービス運用指針(運用範囲、場所、内部統制有無)
セキュリティ	リスク分析範囲、順守すべき社内規程・ルール・法令・ガイドライン等の有無

シートを用いて、まず、基本的な非機能要求を確認する



業務継続性

- ・長時間ダウン時を考慮せよ
- ・運用を考慮せよ
- ・訓練せよ

セキュリティ

- ・守り以外も考慮せよ
- ・対組織だけでなく、対個人も考慮せよ
- ・IPAの情報セキュリティ読本などを参考にせよ

ユーザビリティ

- ・ユーザビリティ、UXを考慮せよ

性能

(1) 性能に関する障害や要求性能との乖離による不具合への対策

- ・条件による違いを考慮せよ
- ・将来の状態を予測せよ
- ・想定外を検討せよ
- ・当たり前の思い込みによる齟齬を払拭せよ

(2) 昨今の性能に関する要求の高まりを踏まえて注意すべきこと

- ・現行システムと比較してみる
- ・体感する

多様化するステークホルダとの合意形成(3.5)

- 昨今は、グローバルシステム、RRPなど
部分システムの機械化ではない全体最適の見直しになっている
- それに伴い、関係するステークホルダも多岐にわたり合意形成が困難

□ 漏れなく、的確にステークホルダを分析すべし

- ・ 漏れなくステークホルダを抽出せよ
- ・ プロジェクトへの影響度と関心度でステークホルダを分析せよ
- ・ ステークホルダ間の対立・関連を見極めよ

□ 当事者意識を持ってレビューをすべし

- ・ 問題、ニーズ、課題を一人称で扱うべし
- ・ セレモニーでないレビューを行うべし

□ エスカレーションパスを明確にすべし

- ・ プロジェクト外で意思決定できる機関を用意せよ

注) RRP (Resource Requirements Planning): 資源所要量計画

現行業務やシステムの把握(3.6)

- IT化されていない業務はない。ビジネスの環境変化が激しい昨今、ビジネスプロセスのすばやい変更が求められる。
- ゼロから検討するのではなく、
現行の業務やシステムを捉えてからの見直しが必須である。
- 昨今は、業務がITシステムに隠蔽されるなど、
業務知識者や既存IT知識者が減少している中で、
現行の業務やシステムをどう捉えるかが重要になってきている。

□現行業務／システムを可視化すべし

- ・システムを可視化せよ
- ・詳細から全体像に概略化せよ

□フィールドワークを通して現状を理解すべし

- ・埋もれた宝(既存ドキュメント)を発掘せよ
- ・可視化した業務を実態と照らし合わせて確認せよ

□プロジェクト全員で共通認識を構築すべし

- ・プロジェクト全員での共通認識として構築せよ
- ・業務移行を見据えて現状を理解せよ

ドキュメント視点の勘どころ（第4章）

要件定義成果物の作成計画時の勘どころ

前後のプロセスの目的を踏まえて、 プロセスに成果物をマッピング



使用目的(構想立案, システム化計画立案, 業務要件定義, システム要件定義)
に対応し, 要件定義関連成果物を,
どのプロセスで, どのレベル(範囲, 深さなど)で作成するかが決まる
(同種成果物でも, プロセスにより記述レベルが異なる)

成果物全体の関連を明確にする



<参考>

機能要件の合意形成ガイド

<http://www.ipa.go.jp/sec/softwareengineering/std/ent03-a.html>

成果物のプロセスへのマッピング例(1/2)

企画プロセス		要件定義プロセス	システム要件定義プロセス (外部設計:※合意形成ガイド)
構想立案	システム化計画立案		
問題点、課題、ニーズなど一覧	問題点、課題、ニーズなど一覧	問題点、課題、ニーズなど一覧	
要求一覧	要求一覧	要求一覧	
業務機能構成表	業務機能構成表	業務機能構成表	システム化業務一覧
ビジネスプロセス関連図	ビジネスプロセス関連図	ビジネスプロセス関連図	
業務フロー	業務フロー	業務フロー	
システム化業務フロー	システム化業務フロー	システム化業務フロー	システム化業務フロー
	業務処理定義書	業務処理定義書	
		システム化業務説明	システム化業務説明
		画面一覧	画面一覧
		画面遷移	画面遷移
		画面レイアウト	画面レイアウト
		画面入出力項目一覧	画面入出力項目一覧
			画面アクション明細
概念データモデル(ER図)	概念データモデル(ER図)	概念データモデル(ER図)	ER図
		エンティティ一覧	エンティティ一覧
		エンティティ定義書	エンティティ定義書
		CRUD図	CRUD図

成果物のプロセスへのマッピング例(2/2)

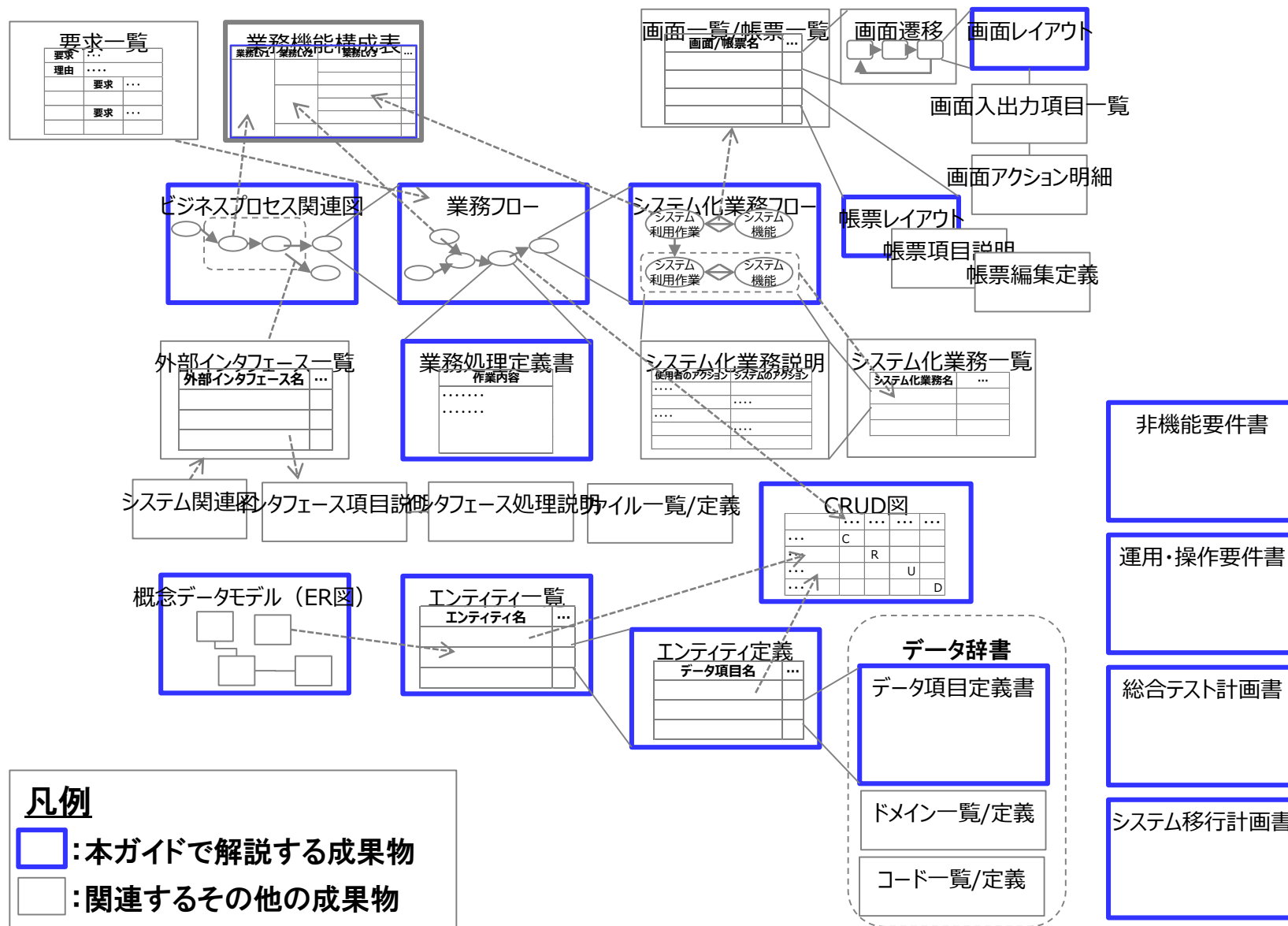


企画プロセス		要件定義プロセス		システム要件定義プロセス (外部設計:※合意形成ガイド)
構想立案	システム化計画立案			
外部システム関連図	外部システム関連図 外部インターフェース一覧	外部システム関連図 外部インターフェース一覧 外部インターフェース項目 定義	外部システム関連図 外部インターフェース一覧 外部インターフェース項目 定義	外部システム関連図 外部インターフェース一覧 外部インターフェース項目 定義 外部インターフェース処理 説明
		バッチ処理一覧	バッチ処理一覧 バッチジョブフロー バッチ処理定義	バッチ処理一覧 バッチジョブフロー バッチ処理定義
	帳票一覧	帳票一覧 帳票レイアウト 帳票概要	帳票一覧 帳票レイアウト 帳票概要 帳票編集定義	帳票一覧 帳票レイアウト 帳票項目定義 帳票概要 帳票編集定義
		データ項目定義書 ドメイン一覧/定義 コード一覧/定義	データ項目定義書 ドメイン一覧/定義 コード一覧/定義	データ項目定義書 ドメイン一覧/定義 コード一覧/定義

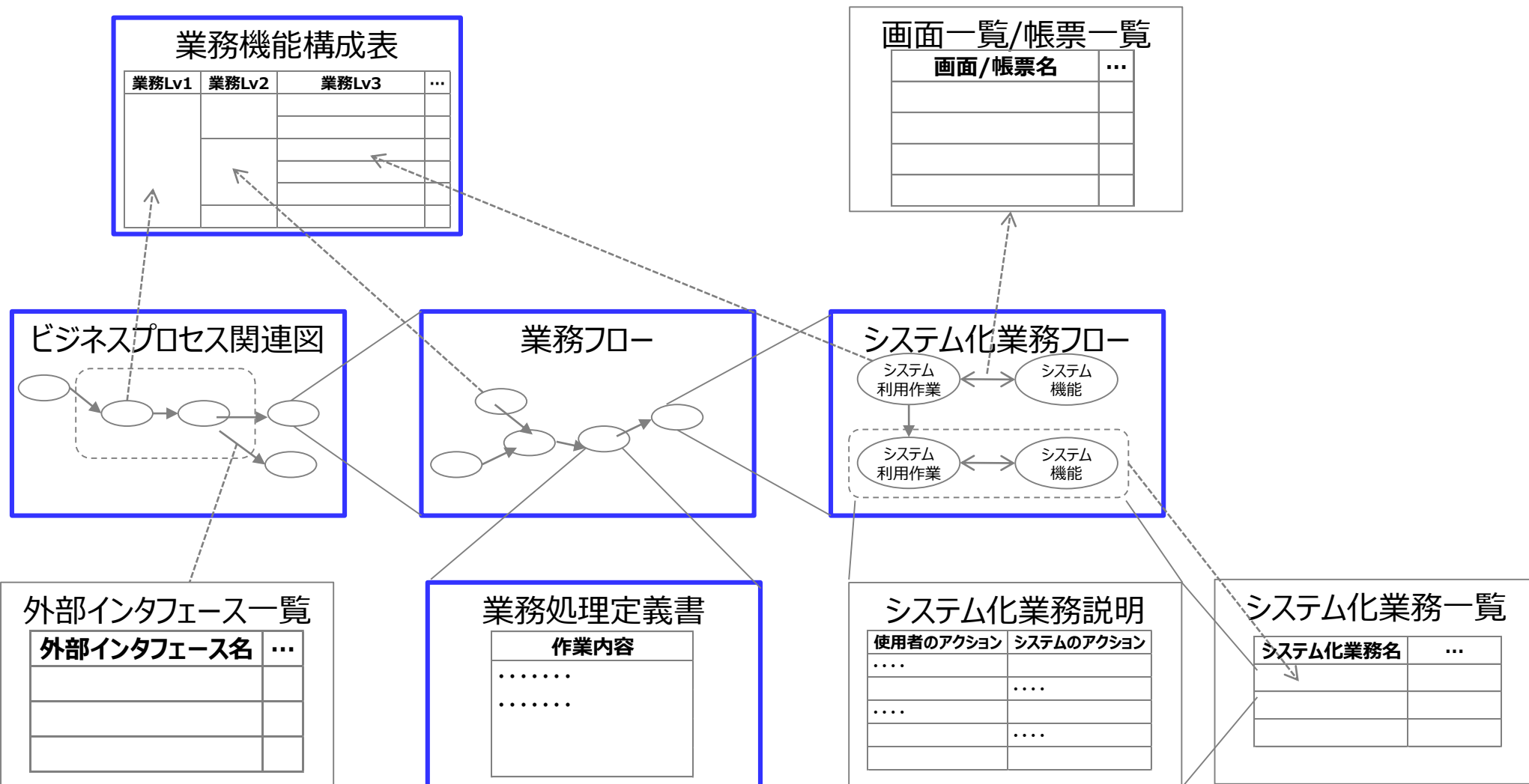
※<関連ガイド>「機能要件の合意形成ガイド」

システム要件定義における発注者(ユーザ企業)-受注者(ベンダ企業)の合意形成に関するコツをまとめたもの

要件定義成果物間の関係(全体概要)



要件定義成果物間の関係(業務フロー中心)

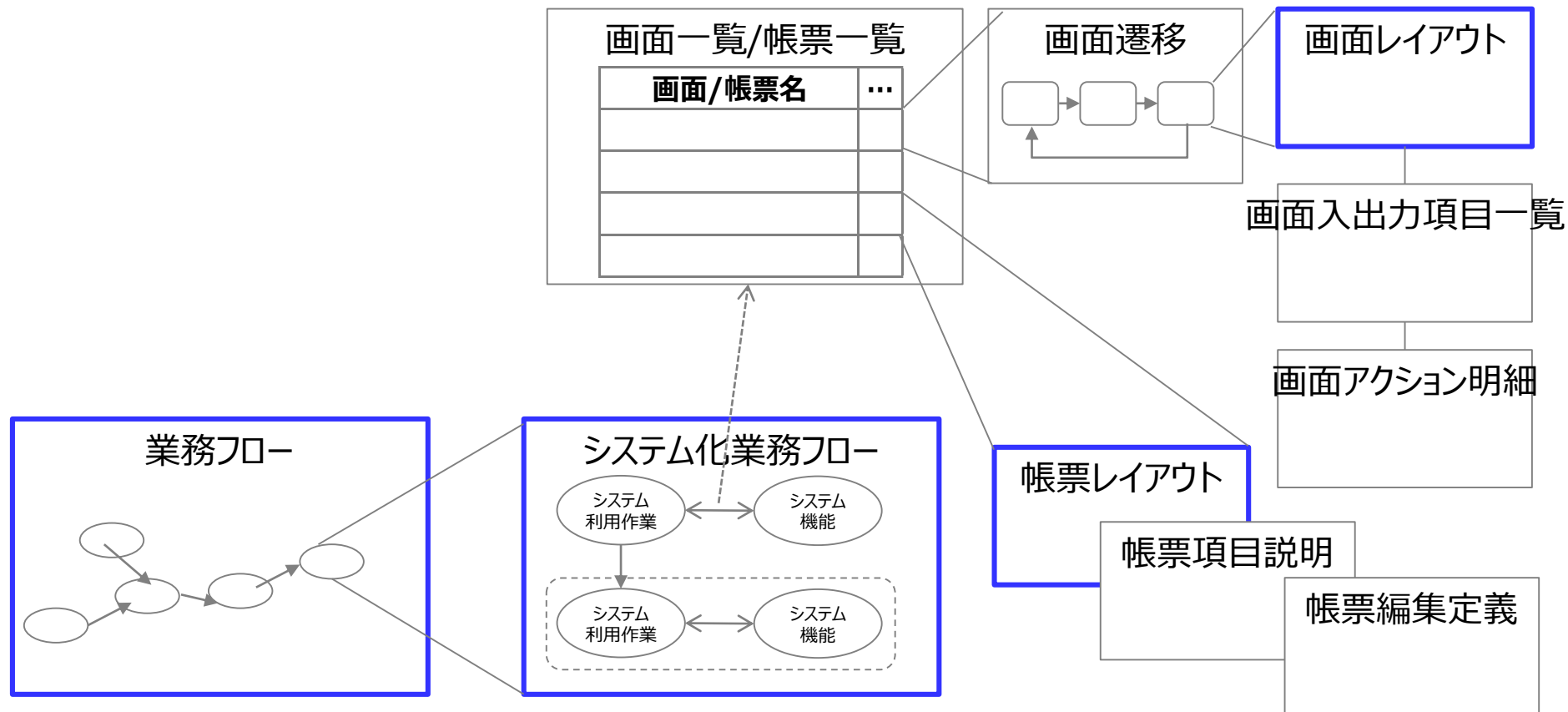


: 本ガイドで解説する成果物



: 関連するその他の成果物

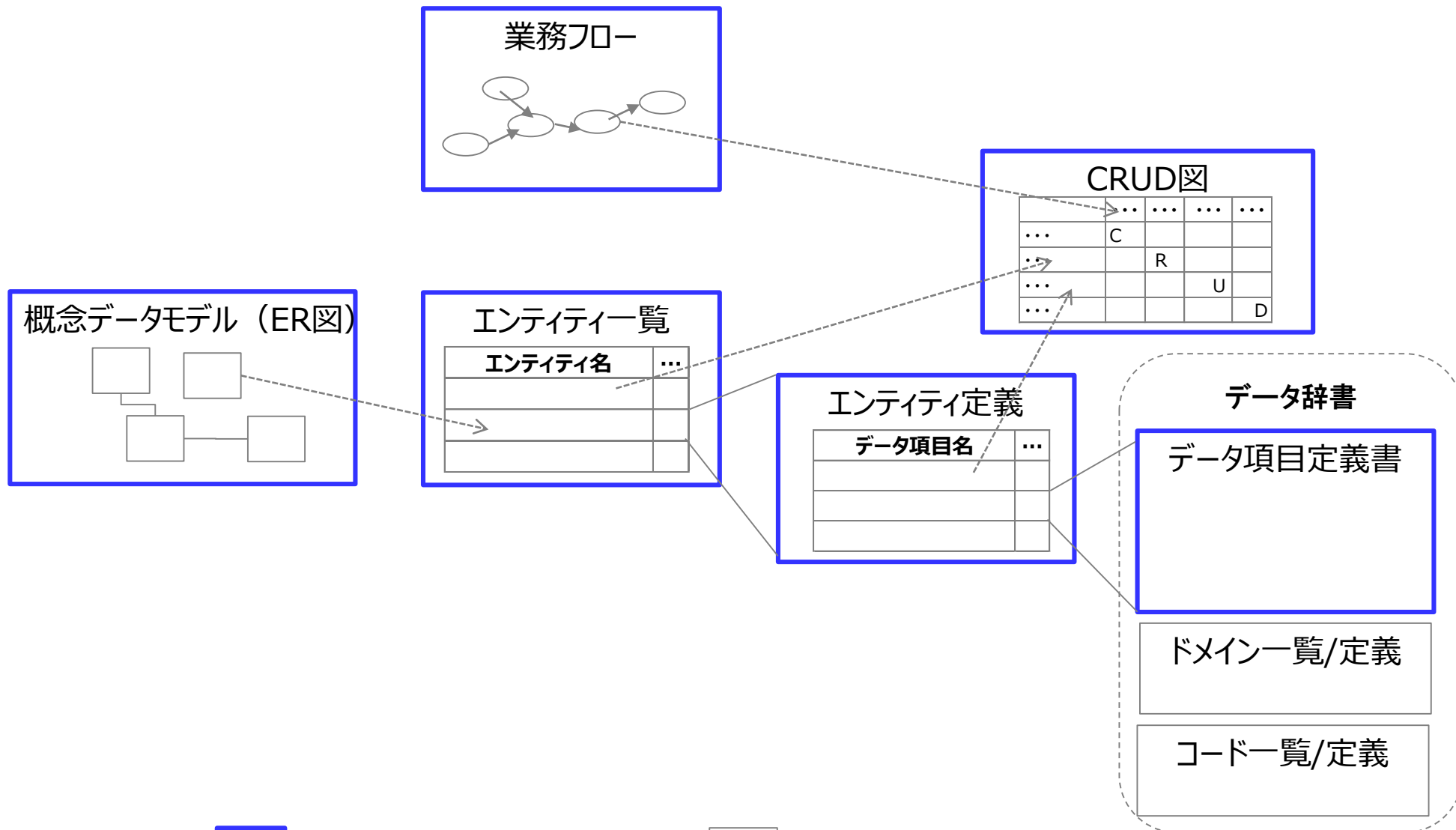
要件定義成果物間の関係(画面/帳票中心)




: 本ガイドで解説する成果物

: 関連するその他の成果物

要件定義成果物間の関係(データ中心)



 : 本ガイドで解説する成果物

 : 関連するその他の成果物

成果物作成上の役割分担

要件定義は、ユーザ企業の利用部門(業務部門)の責任

ユーザ企業の業務部門, システム部門, ベンダ企業が
それぞれの得意とする作業を**分担して進めることが現実的**



各成果物を主体となって作成する部門は,
プロジェクトの状況に応じて異なる

関係者全員が対象業務を理解することが重要

主要な要件定義成果物一覧と作成時の役割分担



項番	成果物	役割分担				補足
		ユーザの部門			ベンダ	
		業務	システム	運用		
4.1.1	ビジネスプロセス関連図	○	△		△	ビジネスプロセスが現状と変わる場合は、業務部門が作成する。 ただし、ビジネスプロセスの変更がなく、システム化対象の明示や外部のインタフェースの明示が主である場合、システム部門が作成してよい。
4.1.2	業務機能構成表	○	△		△	
4.1.3	ビジネスプロセスフロー (業務フロー／システム化業務フロー)	○	△		△	
4.1.4	画面／帳票レイアウト	△	○		△	
4.1.5	業務処理定義書	○	△		△	業務自体の目的や業務内容を明示するため、利用部門が作成する。
4.1.6	概念データモデル (ER図)	△	○		△	業務部門がER図を理解することが困難な場合は、概念データモデルを説明した補足資料 (4.1.6 概念データ コラム参照) を確認する。
4.1.7	エンティティ定義書／ データ項目定義書	△	○		△	
4.1.8	CRUD図	△	○		△	業務理解促進の意味も含めてシステム部門が作成するのが望ましい。
4.1.9	総合テスト計画書	○	○		△	多様な内容が含まれる成果物であり、内容に応じた作成主体を検討し、分担して作成する。 また、計画書であるため、内容に応じた全ての関係者に確認する。
4.1.10	システム移行計画書	○	○		△	
4.1.11	運用・操作要件書	○	○	○	△	業務運用やシステム運用の要件、操作要件など多様な内容が含まれる成果物であり、内容に応じた作成主体を検討し、分担して作成する。 また、内容に応じた全ての関係者に確認する。
4.1.12	非機能要件書	△	○	△	△	非機能要件にも利用部門の業務要件が大きく関係してくるが、技術的な要素も多いため、システム部門が主体で、利用部門から業務要件を聞き出し、成果物の作成を行うことが効率的である。 また、非機能要件は多岐にわたるため、内容に応じた全ての関係者に確認する。

凡例) ○ : 作成主体、△ : 作成支援 (情報提供、確認 (チェック)、アドバイス)

主要な要件定義成果物作成上の留意点

バランスに注意し、プロジェクトに最適な記述形式を選択

- ◆ 要件定義に不慣れな業務担当者にも分かりやすい書式
 - ・要求者(発注者)から開発者(受注者)に要件を正確に伝える
- ◆ 標準の書式
 - ・要件定義内容を第三者に検証依頼
 - ・将来のシステム更改時に初期構築時とは異なる実装者に依頼

業務要件定義とシステム要件定義

要件定義成果物はこれらに対応してそれぞれ存在
プロジェクトや対象システムの特性によっては、
両成果物が1つにまとめられることもある

1. ビジネスプロセス関連図

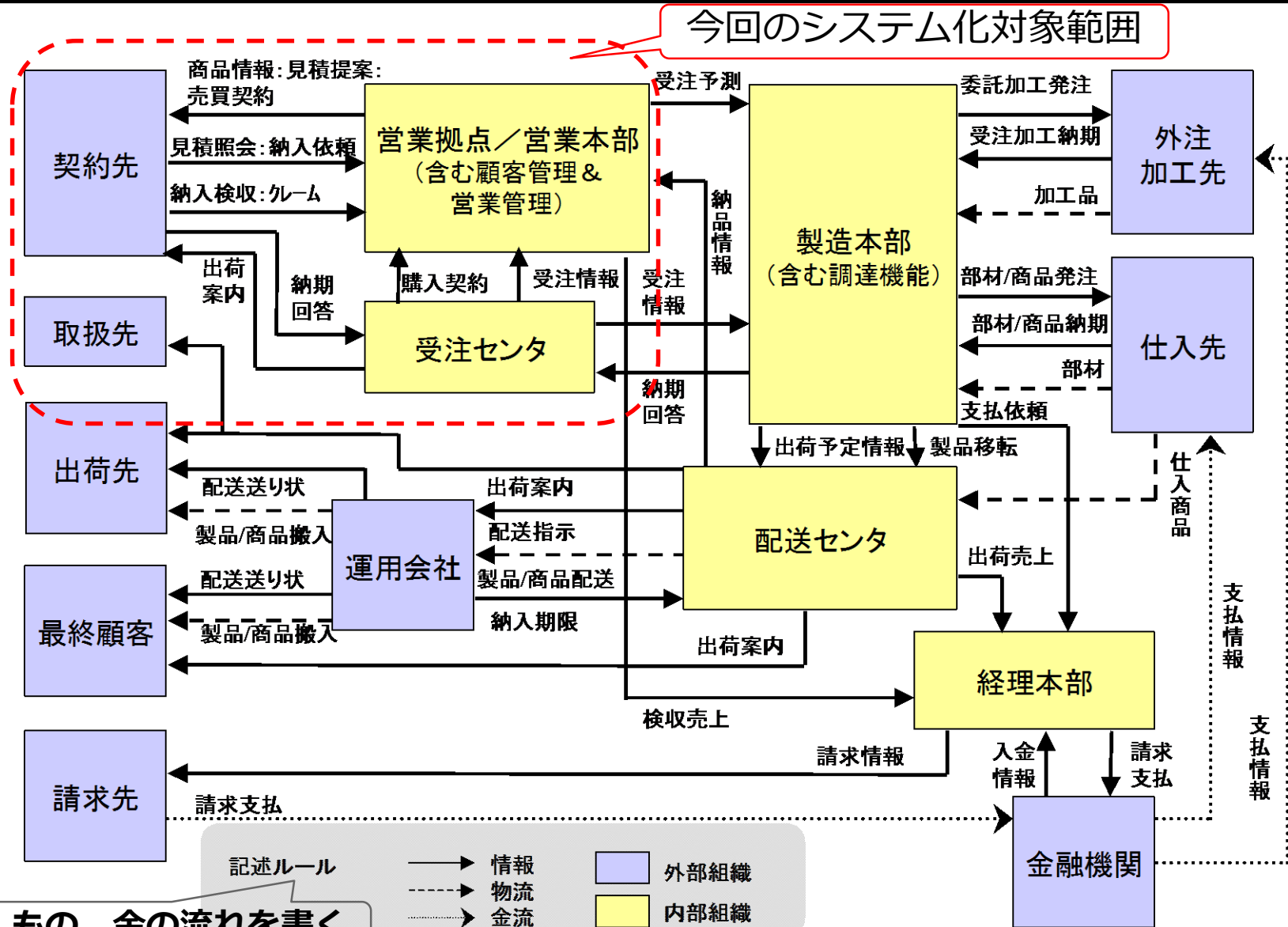
要件定義／システム開発対象の範囲(スコープ)を明確化

→それをもとに、スコープ外とのインタフェースを明確化

ビジネスプロセスの改革内容に関する認識を共有

- システム化対象を明確にする
- 全社システム化計画との整合性を確保する
- ビジネスプロセスの全体像を確認する
- 現場作業への影響範囲を確認する

ビジネスプロセス関連図の例(1)

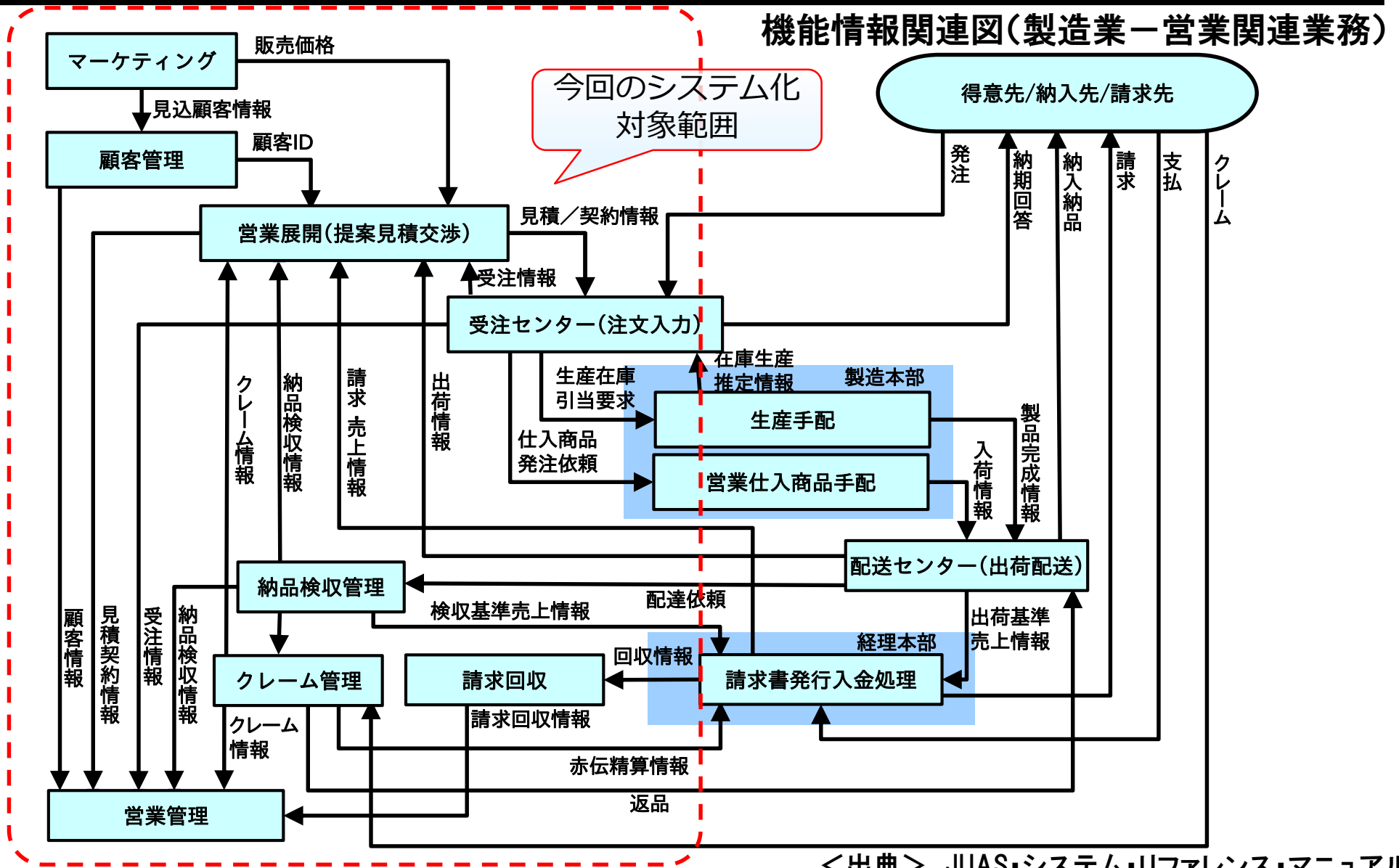


情報、もの、金の流れを書く

<出典> JUAS・システム・リファレンス・マニュアル

ビジネスプロセス関連図の例(2)

機能情報関連図(製造業-営業関連業務)



<出典> JUAS・システム・リファレンス・マニュアル

2. 業務機能構成表

対象の業務全体をまとまりのある機能単位でグループ化し、
機能ごとに
システム化対象か人手で処理を行うプロセスかを区分

- システム化対象を明確にする
- 階層的に表現し、全体を把握する
- 担当者に要求されるスキルレベルを明確にする
- 現在、将来の作業時間を明確にする
- 優先順位、開発規模の調整

業務機能構成表の例(図書館システム)

図書館管理業務機能構成表

業務 レベル 1	業務 レベル 2	業務 レベル 3	スキル	担当者	作業時間(分/月)		システム 化対象	重要度	優先 順位
					現在	将来			
図書管理	図書購入	図書購入依頼、受取	A	田中	120	→	人手	—	—
	新規登録	図書登録作業	B	田中	60	→	○	1	1
	在庫管理	棚卸、除却、修復など	B	田中	120	→	○	2	2
	貸出期間 管理	貸出期間経過の督促など	B	田中	1,200	→	○	2	1
	在庫照会	図書在庫照会への回答など	C	田中	1,200	→	既存	—	—
会員管理	入会管理	会員登録	B	山田	1,200	→	○	1	1
	継続、退会 管理	変更、継続	B	山田	600	→	○	1	1
		退会	B	山田	600	→	○	1	1
	有料会員 管理	請求	B	山田	480	→	○	1	1
		入金確認 督促	B	山田	420 840	→	人手 人手	— —	— —
貸出管理	貸出	新規貸出	C	鈴木	8,400	→	○	1	1
		ネット貸出	B	中村	2,100	4,200	○	2	2
	返却	返却、督促など	B	鈴木	2,400	→	人手	—	—
	期間延長	期間延長受付	B	鈴木	1,200	→	人手	—	—
ネット照会、 予約管理	ネット照会	棚検索、貸出可能照会	C	中村	2,400	4,800	○	3	3
	ネット予約	予約受付、引当通知	B	中村	4,800	9,400	○	3	3
		督促	B	中村	4,800	9,400	○	3	3

スキル A:熟練者 B:AにもCにも当てはまらない業務 C:教えれば誰でも出来る業務

重要度 1:絶対必要 2:次に重要 3:急を要しない,次期システム回しでもよい

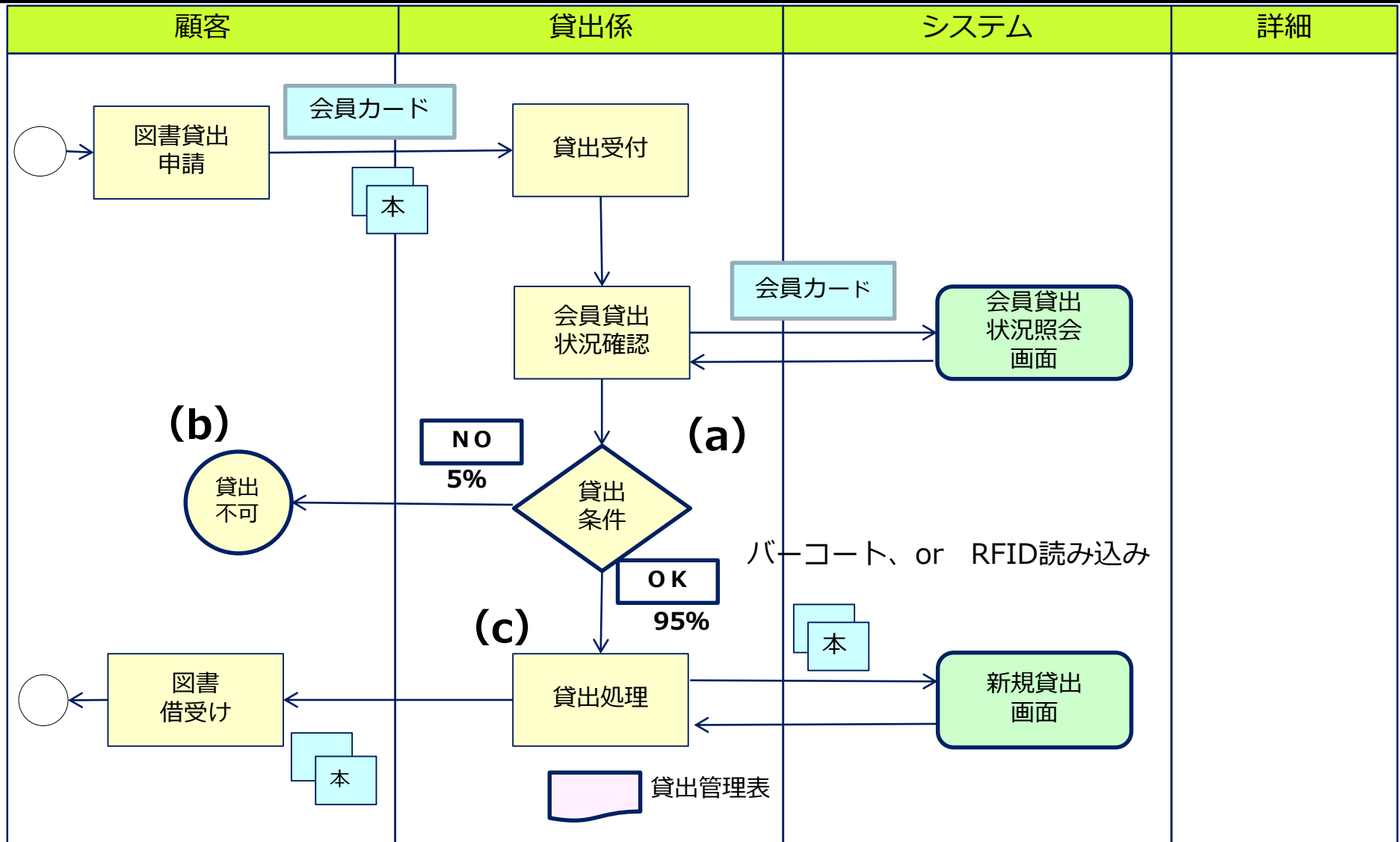
優先順位 1:Must 2:Nice to have 3:効果少ない

3. ビジネスプロセスフロー(業務フロー/システム化業務フロー)

As-Is業務フローを作成して業務の問題点や課題を明確化
→As-Is業務を関係者で理解
新しいビジネスプロセスを創造し,
To-Be業務フローに表して共有

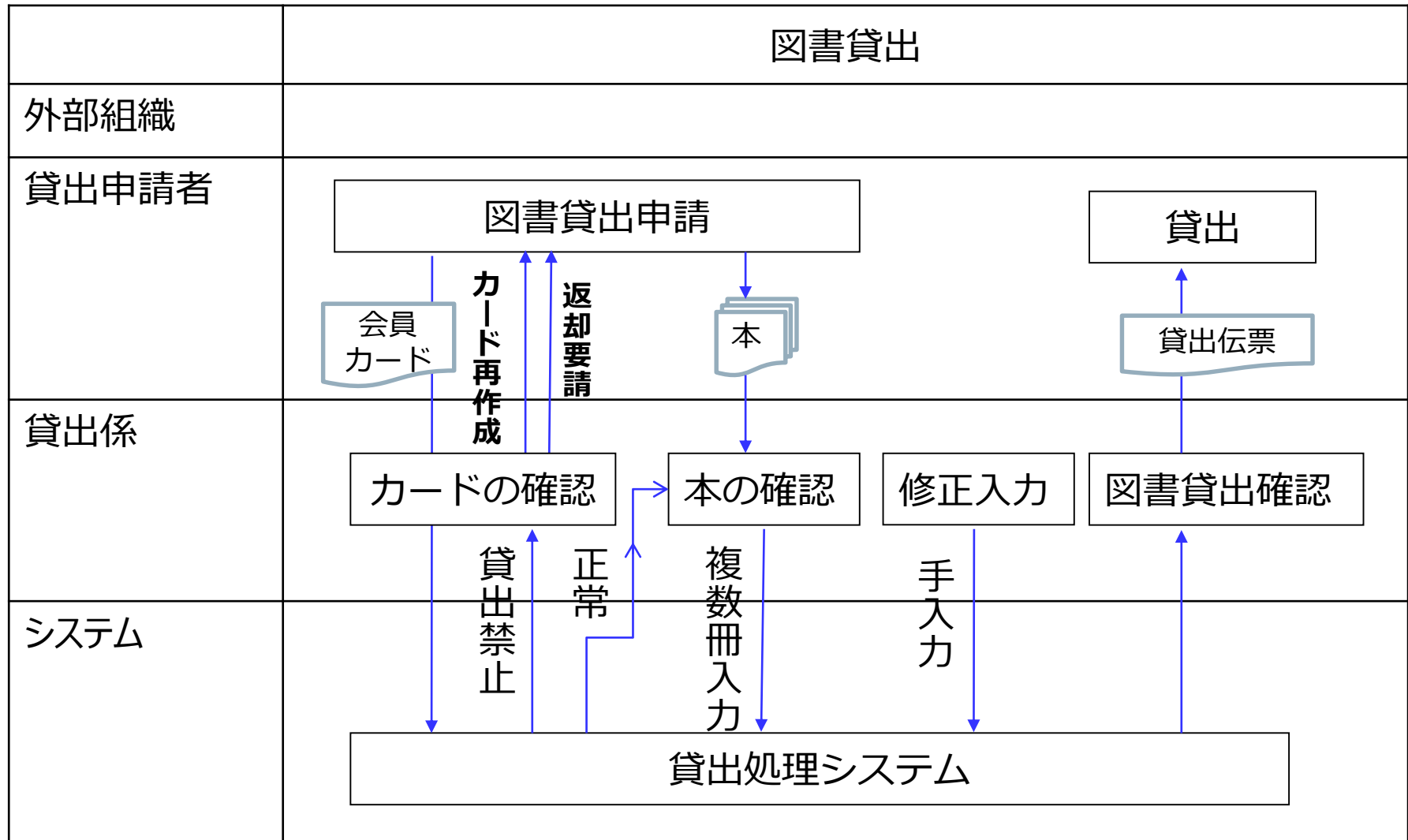
- 業務要件:業務フロー, システム要件:システム化業務フロー
- 業務改革の視点を常に意識する
- 5W2Hに注意して確認する
- 通常処理, 特殊処理, 例外処理を記載する
- あくまでも業務として書く(システム処理フローにしない)
- 記述方法:BPMN(Business Process Modeling Notation), フローチャート, DFD(データフローダイアグラム), UML(Unified Modeling Language)のアクティビティ図, PFD(プロセスフローダイアグラム), など

業務フローの例(1)



図書館の図書貸出業務流れ図(業務フロー)の例(BPMN)

業務フローの例(2)

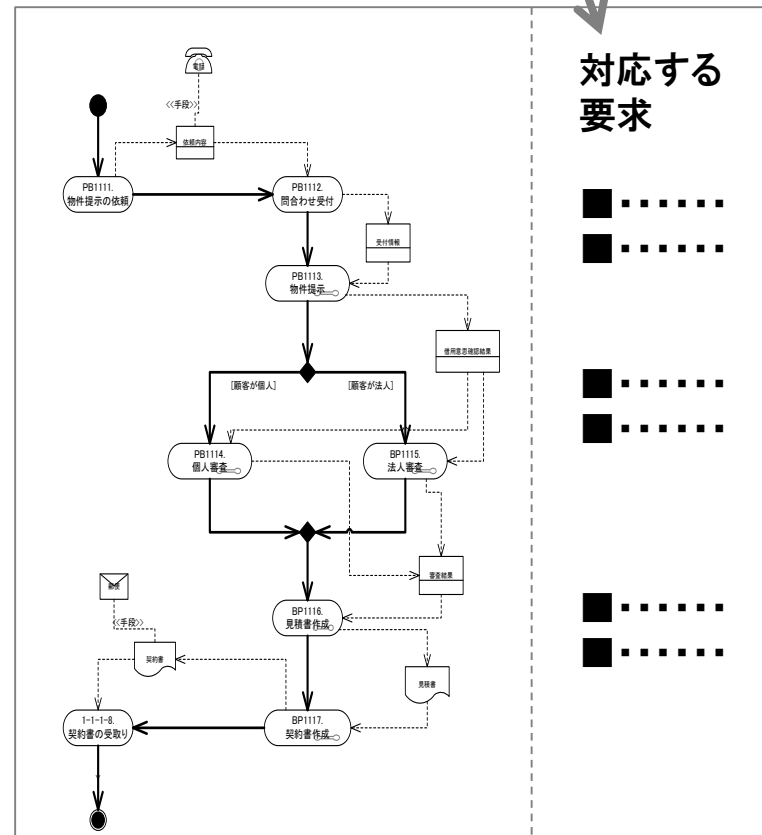
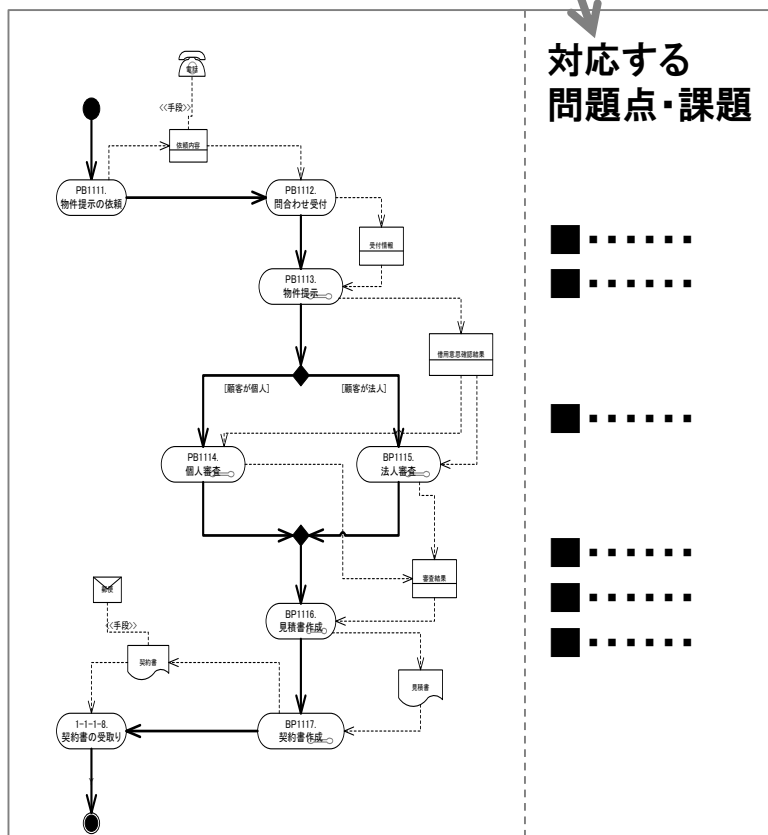


PFD(プロセスフローダイアグラム)による図書貸出業務流れ図(業務フロー)

業務フローの例(3)

問題点・課題一覧

要求一覧



As-Is業務フロー

To-Be業務フロー

4. 画面／帳票レイアウト

画面／帳票レイアウト(画面)

- レイアウトを標準化する
- 画面遷移を標準化する
- データ項目が要求とあっていることを確認する
- データ例で確認する
- 操作方法, 画面遷移が要求と合っていることを確認する
- 要件定義で画面仕様を業務部門に確認し了解を得る
- なぜこのレイアウトになったかの検討経緯を残す
- B2Cはリーガルチェックを行う

画面／帳票レイアウト(帳票)

- 帳票を標準化する
- 出力項目, 表示形式, 表示順序などを業務部門に確認する
- 用途の分かる名前をつける
- データ例で確認する
- 電子化を検討する
- 画面のハードコピー, csvファイルでの対応を検討する
- なぜこのレイアウトになったかの検討経緯を残す
- 顧客提示帳票はリーガルチェックを行う

画面レイアウトの例

操作者が業務実行のために必要とする機能メニュー

新規, 照会, 変更, 削除のうち必要事項を表示

クリックするとポップアップメニューで項目を選択

上段にHeader
下段にDetail
を表示

エラーメッセージ等を表示

The screenshot shows a web application interface with a sidebar menu on the left, a main content area with a form and a table, and a footer with user information and an error message.

お気に入り

- 受注管理
 - 受注登録(ヘッダー・明細形式)
 - 受注一覧(拡張メニュー)
- 受注分析(グラフ1)
- 受注分析(グラフ2)(V2.1)
- 受注分析(マトリックス)(V2.1)
- 受注検索(ノンストップ・スクロール)
- 受注検索(項目代入)
- 製品情報(検索語入力補完)
- 出荷承認・指示(後続イベント)
- 出荷完了報告(後続イベント分割出荷)
- 出荷情報帳票出力(JasperReports帳票出力)

訪問予定

その他の機能

連携機能

管理者用画面

LOGOUT

以下データの操作: 照会 更新 一覧に戻る

更新

受注

注文番号	A004	受注金額(千円)	89000
会社名	KK情報システム	部門名	システム技術グループ
氏名	長谷川 拓海	直送形態	<input checked="" type="radio"/> 直送 <input type="radio"/> 在庫
受注日	2013-01-26	納品日	2013-01-26
連絡事項	連絡事項×××××		
注文書添付	参照...	見積書添付	参照...

4件

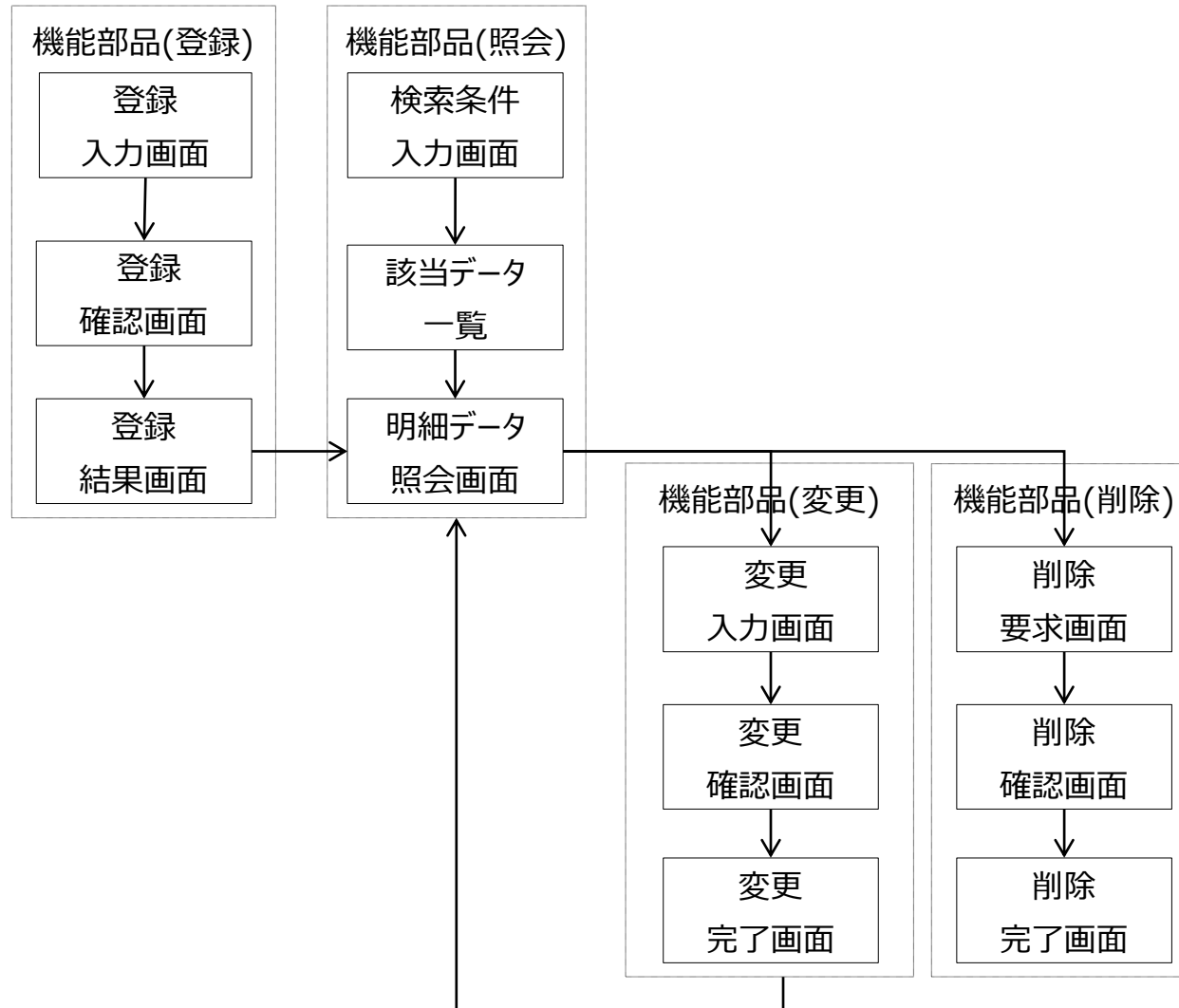
No.	更新	削除	製品コード	製品名	製品分類	受注数量
1	<input type="checkbox"/>	<input type="checkbox"/>	RakFW	楽々Framework II	開発ツール	1
2	<input type="checkbox"/>	<input type="checkbox"/>	RakWF	楽々ワークフロー II	文書管理	4
3	<input type="checkbox"/>	<input type="checkbox"/>	QSPortal	QuickSolution Portal	検索エンジン	1
4	<input type="checkbox"/>	<input type="checkbox"/>	RakFW	楽々Framework II	開発ツール	4
5						
6						
7						
8						
9						
合計						10

更新

demo0110 User: admin(lang0, 127.0.0.1) Date: 2015-10-29 15:43:02(87ms) g IE 7.0 SSO[RakWF21]

<

画面遷移の標準例



5. 業務処理定義書

業務理解促進のため、各業務処理機能の内容を明確化
業務改革や改善の詳細内容の認識を共有

- 背景, 目的, 範囲, 制約事項を明確にする
- 重要な時間や日付, 発生・完了タイミングを確認する
- 重要な業務系の例外まで記載する
- システム例外, データ例外, 運用例外は, 除外してよい
- 表形式で書く
- 「現行どおり」と言われても仕様として記述する
- 新旧の違いを明記する
- セキュリティ制約も記載する
- 障害時の代替方法も決める
- 管理機能も記載する
- IDを付与し関連づける
- [USDM](#)の使用により, [要求の理由](#)を記述できる

図書館システム－会員の状況と貸出延長回数による新規貸出制限

(判定条件3)

(判定条件1) (判定条件2)

			現在貸出中の図書の返却日延長回数				
会員種別	会費 納入状況	判定項目	なし	1回	2回	3回以上	
無料会員	-	返却日	延長可		延長不可		
		貸出	合計3冊まで 貸出可能	追加貸出 不可			
有料会員	滞納なし	返却日	延長可			延長不可	
		貸出	合計5冊まで 貸出可能		追加貸出 不可		
	会費 督促中	返却日	延長可		延長不可		
		貸出	合計3冊まで 貸出可能	追加貸出 不可			

USDMの例(1/3)

貸出延長可否判定	要求	KS01	貸出延長請求、追加貸出請求を受理してもいい状態かどうかの判定と追加貸出可能冊数の算出を、会員情報、図書貸出情報をもとに実施する					
		理由	本の貸出期間の延長可否や追加貸出可能冊数を会員向け貸出規約に基づいて確認したい					
		説明	会員向け貸出条件は次のとおり:		現在貸出中の図書の返却日延長回数			
			会員種別	会費納入状況	判定項目	なし	1回	2回
	無料会員		-	返却日 貸出	延長可 合計3冊まで貸出可能	延長不可 追加貸出不可		
	有料会員	滞納なし	返却日 貸出	延長可 合計5冊まで貸出可能	延長不可 追加貸出不可			
			会費督促中	返却日 貸出	延長可 合計3冊まで貸出可能	延長不可 追加貸出不可		
		要求	KS01.1	会員IDをもとに、有料/無料の区別、会費納入状況、貸出中図書の冊数、貸出中の図書に対する返却期日延長状況を取得する				
	理由		貸出延長を実施できる状態かどうか、および追加貸出可否および追加貸出可能冊数を判定するために、必要な情報を取得したい					
	説明		-					
<判定に使用する情報の取得>								
<会員情報取得>								
□□□	KS01.1-01	カードから読み取った会員IDをキーに会員マスタから以下を取得する						
		<ul style="list-style-type: none"> ・有料/無料の区分 ・会費納入状況を取得する 						
<貸出状況取得>								
□□□	KS01.1-02	カードから読み取った会員IDをキーに図書貸出テーブルから貸出中図書の情報を取得する						
		<ul style="list-style-type: none"> ・貸出中図書の冊数 ・各貸出中図書のID ・各貸出中図書の返却期日延長回数 						
<返却期日延長回数取得>								
□□□	KS01.1-03	貸出中の図書の返却期日延長回数のうち、もっとも多い回数を取得する						

USDMの例(2/3)

貸出延長可否判定

要求	KS01.2	貸出延長請求を受理できる状態かを判定する
	理由	取得した情報と会員向け貸出規約から、本の貸出期間延長可否を知りたい
	説明	—
〈貸出期日延長可否判定〉		
〈可否判定〉		
□□□	KS01.2-01	無料会員かつ貸出中の図書に対する返却期日延長が1回以下 → 延長可能
□□□	KS01.2-02	無料会員かつ貸出中の図書に対する返却期日延長が2回以上 → 延長不可
□□□	KS01.2-03	有料会員かつ会費延滞なし、かつ貸出中の図書に対する返却期日延長が2回以下 → 延長可能
□□□	KS01.2-04	有料会員かつ会費延滞なし、かつ貸出中の図書に対する返却期日延長が3回 → 延長不可
□□□	KS01.2-05	有料会員かつ会費延滞あり、かつ貸出中の図書に対する返却期日延長が1回以下 → 延長可能
□□□	KS01.2-06	有料会員かつ会費延滞あり、かつ貸出中の図書に対する返却期日延長が2回以上 → 延長不可

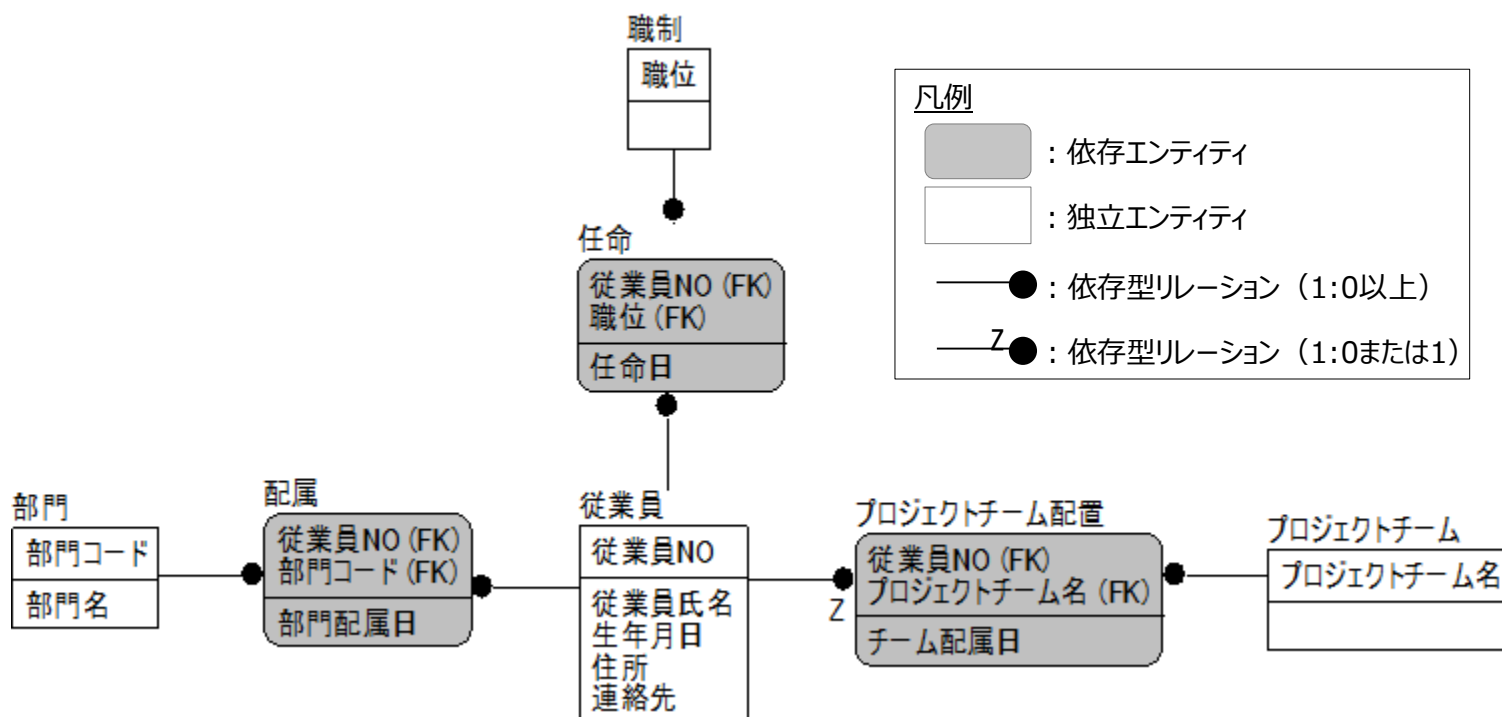
USDMの例(3/3)

貸出延長可否判定	要求	KS01.3	追加貸出可否の判定と追加貸出可能冊数の算出を行う
		理由	取得した情報と会員向け貸出規約から、追加貸出可否と貸出可能冊数を知りたい
		説明	—
	＜追加貸出可能冊数算出＞		
	＜貸出可能冊数取得＞		
	□□□	KS013-01	無料会員かつ図書に対する返却期日延長が0回 → 上限3冊まで貸出可
	□□□	KS013-02	無料会員かつ図書に対する返却期日延長が1回以上 → 追加貸出不可
	□□□	KS013-03	有料会員かつ会費延滞なし、かつ貸出中の図書に対する返却期日延長が1回以下 → 合計5冊まで貸出可
	□□□	KS013-04	有料会員かつ会費延滞なし、かつ貸出中の図書に対する返却期日延長が2回以上 → 追加貸出不可
	□□□	KS013-05	有料会員かつ会費延滞あり、かつ貸出中の図書に対する返却期日延長が0回 → 合計3冊まで貸出可
□□□	KS013-06	有料会員かつ会費延滞あり、かつ貸出中の図書に対する返却期日延長が1回以上 → 追加貸出不可	
＜追加可能冊数取得＞			
□□□	KS01.3-07	追加貸出が可能と判定された場合、以下の式により貸出上限冊数を計算する 追加貸出可能冊数 = 上記により取得した上限冊数 - 貸出中図書の冊数	

6. 概念データモデル(ER図)

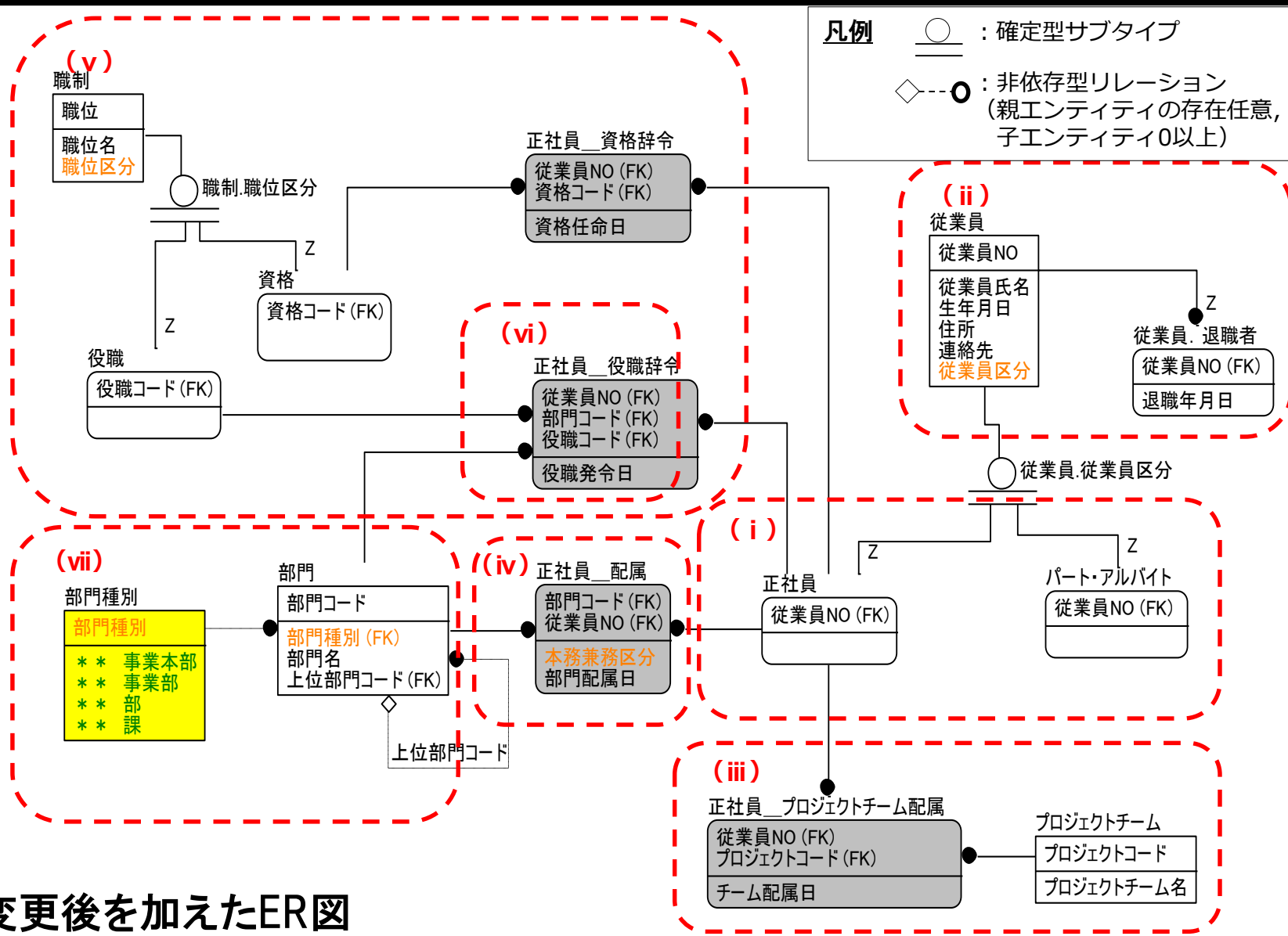
実業務の管理対象や業務ルール等を分析・モデル化し、
企業のビジネス活動の構造を理解

- 企業のビジネス活動を可視化する
- ER図は業務の実態，事実に忠実に描く
- ER図の読み方を業務部門に説明し，両者で確認する
- To-Beとしての必要性も検証する
- 具体的なデータを入れて検証する
- 個々のデータの生成から破棄までを明らかにする



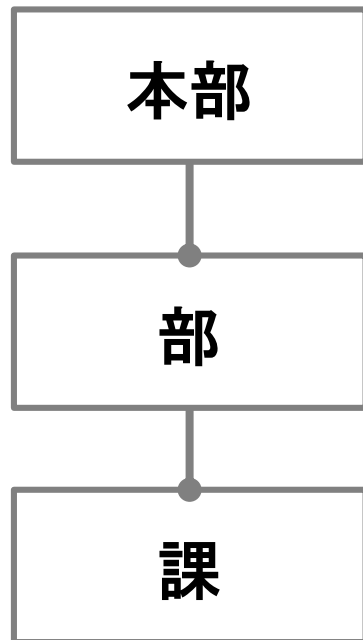
当初のER図

ER図の例(2/2)



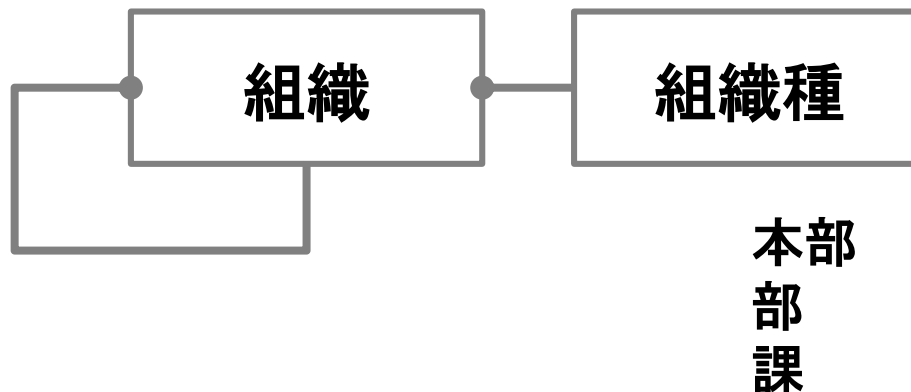
変更後を加えたER図

(a) 分かり易いモデル



組織の構造を
実体どおりの
縦の関係で表現

(b) より抽象化されたモデル



組織の構造を繰り返される関係で表現し、
本部・部・課のどれなのかは「組織種」に持たせる
という抽象化を行った結果

7. エンティティ定義／データ項目定義

管理対象エンティティの管理内容, 管理ルールを
項目レベルで把握

- 主要なデータ項目は業務部門が中心になってまとめる
- 初期値, 範囲, 扱いを明確にする
- 項目名称をルール化
- ひとつの項目に複数の意味を持たせない
- データ管理者を決める

エンティティ名	正社員__役職辞令
説明	正社員に発令した配属、役職任命の履歴を保持する。

No.	属性名	主キー	データ型	桁数 文字数	説明
1	従業員NO	○	文字列	6	従業員を一意に識別する番号
2	部門コード	○	文字列	4	部門を一意に識別するコード
3	役職コード	○	文字列	2	役職を一意に識別するコード
4	役職発令日		日付	-	役職任命を発令した年月日

8. CRUD図

主要エンティティ, データ項目について,
どの機能で作成, 使用, 更新, 消去されるかを確認し,
その主要データ項目のライフサイクルを把握

- 主要データ項目のライフサイクルを把握すると, データ, 機能の抜け・漏れ, 矛盾が
検証できる
- 主要データ項目だけでよい
- 削除については, 業務的な削除(論理削除)で記載する
- エンティティレベルでは確認できないあいまい性を排除できる

CRUD図(エンティティレベル)の例

凡例：C（登録、Create）、R（照会、Read）、U（更新、Update）、D（削除、Delete）

	受注	生産手配	進捗管理	倉庫管理	出荷	売上計上	財務処理	削除処理
受注	C	U	U	U	U	U	U	D
出荷			U	U	C	C	U	D
・ ・								

CRUD図(データ項目レベル)の例

凡例：C（登録、Create）、R（照会、Read）、U（更新、Update）、D（削除、Delete）

	受注	生産手配	進捗管理	倉庫管理	出荷	売上計上	財務処理	削除処理
受注番号	C							D
受注数量	C							D
単価	C					U	U	D
受注金額	C				U	U	U	D
主要仕様 (色など)	C	U			U	U		D
生産数量	-	C	U	U	U	U		D
・ ・								

9. 総合テスト計画書

要件定義で決められたことが、
「システムの機能として不足や誤りなく実装されているか」
「定義した要件で実際の業務が成立するか」
「例外処理も含め、業務が正しく実行できるか」
を確認

- 要件定義で詰めるところは詰めておく
- 総合テストシナリオを記載する
- 総合テストシナリオには、正常系、例外系、特殊系など全てを記載する
- 全体業務を熟知した担当者を作成にあたらせる
- 環境とデータを一部用意し、事前検証してみる
- 要件定義で義務付ける
- 終了条件(品質指標, レスポンスタイムなど)を詰めておく

1. 総合テストの目的

2. 総合テストの概要

(1)対象、範囲

(2)前提条件

(3)環境(ハードウェア、ソフトウェア、ネットワーク)

(4)期間、スケジュール

(5)体制

3. 総合テスト仕様

(1)総合テストシナリオによる一連のつながりの確認(テスト)

- ・通常処理

- ・例外処理

- ・特殊処理

(2)サイクルテスト(日回しのテスト)

(3)締め処理(日、月、年次)

(4)他システムとの連携テスト

(5)性能負荷テスト

(6)障害回復テスト

(7)移行リハーサルテスト

(8)ユーザテスト

4. 総合テストの終了条件

10. システム移行計画書

現行システムから新システムへ、
業務、データ、システム要素を問題なく移行する手順を確認

- 業務、データ、システム要素(Ap, OS, ハードウェア, ネットワーク)を移行する
- 要件定義の段階で移行方法を詰められるところは詰めておく
- 新旧システムの並行稼働の有無を確認する
- 移行中止判断基準・方法を明確にする
- エンドユーザ教育の方針を検討する
- 移行体制を検討する
- 一斉切替えか、スモールスタート(こちらがベター)かを検討する
- 移行データを絞り込む
- 移行時にデータクレンジングを行う
- むやみに移行プログラムの本数を増やさない
- 移行リハーサルを計画する
- 移行チームを独立に設置する
- 訓練・研修を計画する

11. 運用・操作要件書

このシステムを業務が実際に運用・操作でき、
それによって業務が遂行できるための要件を定義

- システムの運用要件(サービス時間など)を決める
- ヘルプ対応を検討する
- インシデント管理, キャパシティ管理などを検討する
- 障害発生時の業務継続などを検討する
- 業務運用要件を決めておく
- 操作手順書の作成は後工程でよく, ここでは操作要件書を作成する
- 運用部門を巻き込む
- 操作性を明確にする
- エラーメッセージの記述基準(原因ではなく指示を記述)を取り決めておく

12. 非機能要件書

業務機能を実現するために必要となる システムの非機能要件を定義

- 要件定義段階で決めておくべき非機能要件を明確化
- 要件定義段階では、最低限必要な項目についての要望でよい
- データ量, レスポンス, 使用性, 稼働率/稼働品質, セキュリティ, 保守性, 移植性を決める
- トレードオフをステークホルダに判断してもらう
- 概算を松竹梅で見積もる

非機能要件の見積もり例

大区分	中区分	小区分	梅	竹	松
信頼性	稼働率	サーバ, 電源, 運転費用 費用の比率 (以下比率)	99.5%未満 1	99.5%以上99.9%未満 2	99.999%以上 3
	障害許容性	ネットワーク (比率)	1	1	1.2
	回復性	障害時業務継続性 (比率)	単一構成 1	コールドスタンバイ 2	ホットスタンバイ 5
	回復性	BCP (災害対策) (比率)	1Wで回復 1	3日で回復 2	1日で回復 5
効率性	レスポンスタイム	プログラム+HW (比率)	3秒以内(画面) 1	2秒以内(画面) 1.2	0.1秒以内 (主要項目) 1秒以内(画面) 2
使用性		画面 (比率)	1	1.2 操作性	2.0 操作性
		帳票 (比率)	1	1.1	1.2
		運用サポート時間 (比率)	9時~17時 1	9時~21時 1.5	24H/365日 3
保守性	プログラム	外部テーブル (比率)	1	1	1.2
		リポジトリ (比率)	1	1.2	1.5
移植性	プログラム	(例) ホスト→クラサバ	1	1.1	1.1
		クラサバ→クラウド	1	1.1	1.1
セキュリティ	攻撃プロテクト	導入ウィルスソフト (比率)	定期更新(~2個) 1	定期更新空(5個) 1.5	常時更新 (6個~) 2.0

上記例は仮設定、数値は概算費用の相対比率

要件定義書のレビュー

要件定義成果物の抜け・漏れ及び成果物間の整合性不備等
に関し、熟視を主体とする公式なレビューによる検査を行い、
品質を担保

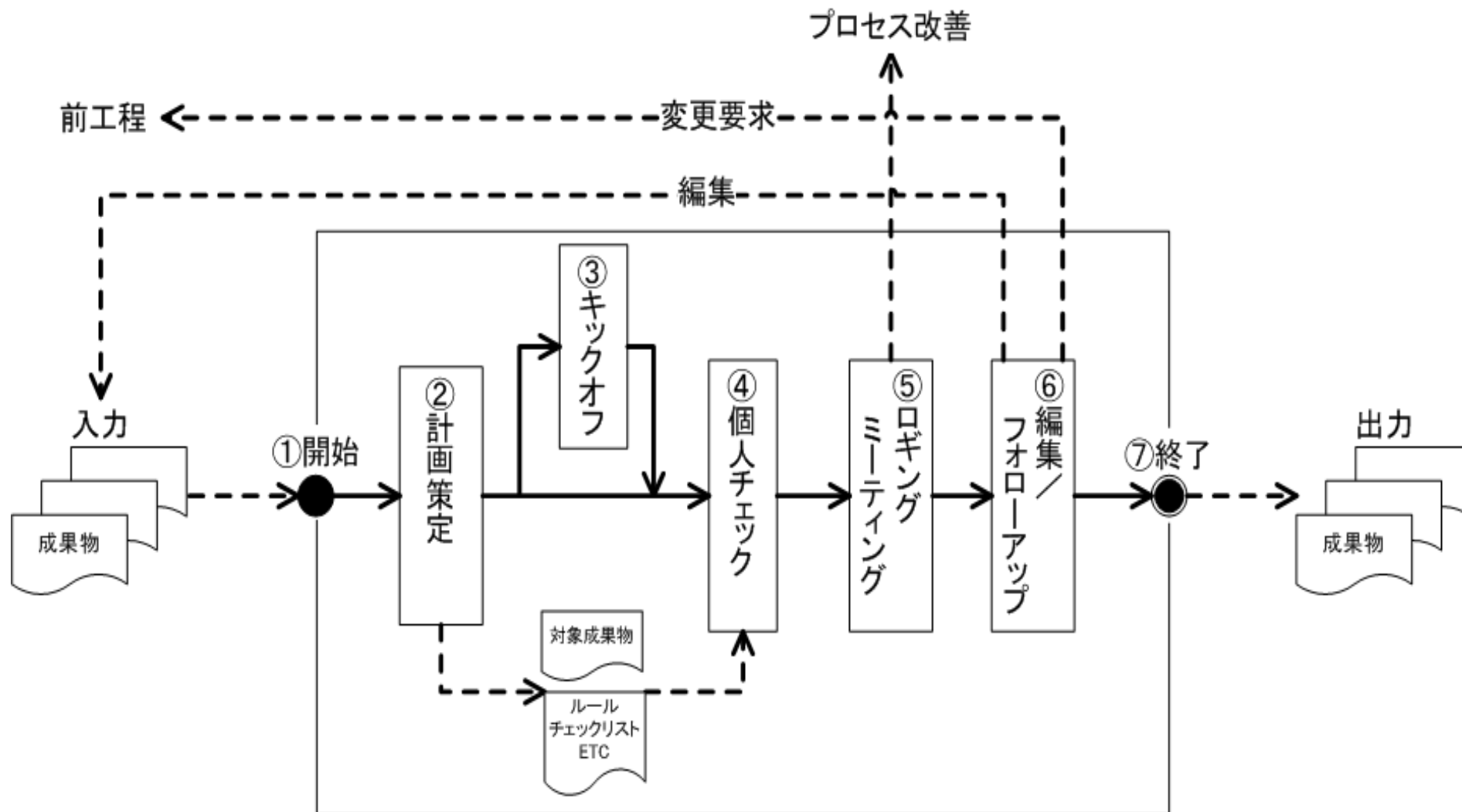
- レビュー手法の提示
- 各レビュー手法による検証内容を確認
- インспекションでの役割の提示
- インспекションのプロセスに従う
- 要件定義でのインспекションは適合性検証が主眼
- 全体を俯瞰した成果物から検証

各レビュー手法に期待される主な効果

効果 手法	欠陥製品の発見	仕様との適合性チェック	標準との適合性チェック	製品の完全性と正しさの検証	理解性と保守性の評価	コンポーネントまたは高リスクの 実証	プロセス改善のためのデータ収集	文書品質の測定	製品に関する他のチームメンバー の教育	対処法についてのコンセンサスの 決定	変更またはバグ修正の正しさの 確認	別の対処法の検討	プログラム実行のシミュレーション	レビューコストの最小化
アドホック・レビュー	●													●
ペア・プログラミング	●				●				●	●		●		
ピア・デスク・チェック	●	●	●									●		●
ピア・レビュー	●	●	●	●					●	●	●			
パス・アラウンド	●	●	●		●				●					
チーム・レビュー	●	●	●		●		●		●	●	●			
ウォークスルー	●			●					●	●	●	●	●	
インスペクション	●	●	●	●	●	●	●	●						

インスペクションでの役割

役割	作業内容など
進行・まとめ役 (Moderator)	<p>インスペクションの主催者である。</p> <p>インスペクションを熟知している必要がある。インスペクション・メンバーの選抜やスケジュール調整、成果物の受領と配布、さらには作業進捗の管理を行う。</p> <p>計画段階で、ある条件が揃った場合にインスペクションの中断か継続可能かのチェックポイントを設けておく。</p> <p>最終的には当該インスペクションの結果報告責任者になる。</p>
レビューを行う役 (Inspector)	<p>欠陥検出の担当で、当該チームの作業内容について既知であるか否かを問わず、高い欠陥発見能力を持つ人が望まれる。</p> <p>インスペクションでは、対象成果物ごとの関連性を考慮して、レビューを行う役などを決める。</p> <p>インスペクションはオーバーヘッドなど非効率性を鑑み、最適人数を決定する。 (経験則として3名～4名まで)</p> <p>インスペクションに関する正式な教育を受けていることが必要である。</p>
説明役 (Reader)	<p>進行に合わせて対象成果物の該当部分を他に説明する。一般的には作成者が担当する。</p>
記録役 (Recorder)	<p>指摘された欠陥と問題を分類、記録する。</p>
作成者 (Author)	<p>インスペクション対象に関わる成果物の作成者または保守者が該当する。</p>



新旧画面対比表の例

[新販売管理業務システム]新旧画面(論理)対比表

新画面	旧画面														
	案件決裁管理(案件情報)【営業秘密】				案件決裁管理(プロジェクト情報)【営業秘密】										
	タブ	タブ	タブ	タブ	タブ	タブ	タブ	タブ	タブ	タブ	タブ	タブ	タブ	タブ	
	サマリー	顧客情報	受注部署情報	システム情報	サマリー	期日情報	顧客情報	受注部署情報	受注工程(実態)	受注額	原価情報	発注内示確認情報	主管部署選択 (受注部署情報タブ内)	売請用主管部署選択 (受注部署情報タブ内)	プロジェクトコード自動採番【営業秘密】

記号の説明:

- ◎ ... 旧画面に対し項目の追加・修正・削除なし
- ▲ ... 旧画面に対し項目の追加・修正・削除あり

新画面		旧画面													
画面名	画面詳細	タブ	タブ	タブ	タブ	タブ	タブ	タブ	タブ	タブ	タブ	タブ	タブ	タブ	タブ
1. 決裁申請・手続承認操作(年度指定/決裁操作者指定)	決裁申請・手続承認操作メイン画面														
1.2. 受注ソフトウェア開発決裁申請(顧客契約承認)	受注ソフトウェア開発決裁申請(顧客契約承認)画面	▲					▲								
	案件情報画面	▲	▲	▲	▲										
	プロジェクト情報画面					▲	▲	▲	▲	▲	▲	▲			
	主管部署選択画面												◎		
	売請用主管部署選択画面													▲	
	決裁種別選択画面														▲
	契約案件集	▲													

業務とワークフローステータスの対比表の例

[新販売管理業務システム]業務とワークフローステータスの対比表

業務名称		ワークフローステータス										
		作成	作成確認	依頼	受付	起案	申請	仮決裁	決裁	結果入力	内容確認	決裁確認
①	(1)受注ソフトウェア開発-①取引開始承認	●			●	●	●	●	●	●	●	●
②	(1)受注ソフトウェア開発-④顧客契約承認	●	●	●	●	●	●	●	●	●	●	●
③	(4)対外提出物	●	●	●	●	●	●	●	●	●	●	●
④	(5)顧客との交際	●	●	●	●	●	●	●	●	●	●	●
⑤	(6)営業-①売上計上報告	●				●	●	●	●	●	●	●
⑥	(6)営業-②売上連携報告	●				●	●	●	●	●	●	●
⑦	(6)営業-③企業ランク見直し	●				●	●	●	●	●	●	●
⑧	(7)請求書対外提出	●				●	●	●	●	●	●	●
⑨	(8)売掛債権回収管理	●									●	●

要件定義における未決定事項の扱い

要件定義の出来栄を評価し、今後の対策を確認

- 要件定義工程で要件の全ては決まらないことが多い
- 完了基準を作成しても、守られることは現実にはない
- 要件の未決定事項をさばく対策を明確化する
- 未決定事項を整理し、リスクとして認識する
- 未決定事項の量によってリスクを評価する
- 基本設計の契約形態にも注意する

要件定義文章の品質向上

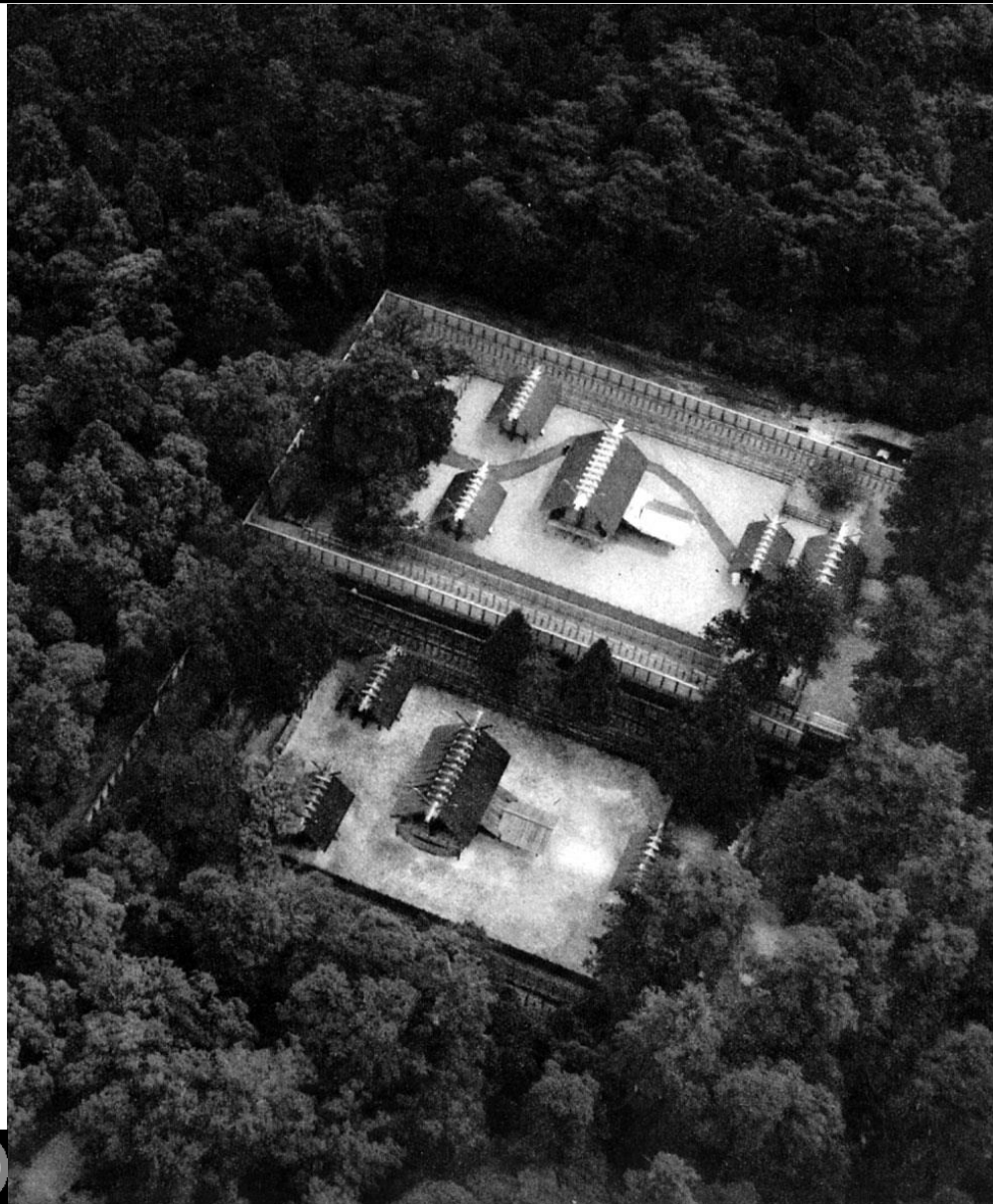
文章の曖昧さにより要件が正しく伝わらないことを避ける

- 仕様を表などで工夫して記載し、あいまい性を低減していくが、要件定義では文章によるドキュメントも多く存在する
- 日本語文章の問題によって、よく齟齬が発生している
- 以外と日本語は難しい
- 記述規約をあらかじめ決めておくことにより、曖昧さを避ける

事例一覧 (第5章)

事例	会社名	第3章 昨今直面している要件定義課題を解決するための勘どころ						第4章 要件定義成果物の品質向上 「4.3 成果物に共通の留意点」		
		3.1	3.2	3.3	3.4	3.5	3.6	レビュー	未決定事項	文章品質
5.1 「ビジネスに貢献する要求を見極める ～ビジネス成果とIT施策の整合性をとる～」の事例	サントリーシステムテクノロジー	✓								
5.2 「要件量を可視化する ～定量化した要件量を使ったスコープ管理～」の事例	NTTデータ		✓							
5.3 「業務バリエーションの整理 ～業務バリエーションを鳥瞰的に把握し整理する業務シナリオマトリクス～」の事例	富士通			✓						
5.4 「非機能要求の進め方 ～各ステークホルダが協力しながら進めていく方法～」の事例	富士通				✓					
5.5 「ステークホルダ分析 ～多様化するステークホルダといかに合意形成を図るか～」の事例	セイコーエプソン					✓				
5.6 「要求の定量化による合意形成と膨らむ要求の制御」の事例	ジャステック		✓		✓	✓	✓			
5.7 「手戻りコストを抑制する要件定義書のレビュー」の事例	ジャステック							✓		
5.8 「要件定義文章の品質向上 ～図表を使った記述技法～」の事例	NTTデータ									✓

1. IPA/SECにおけるシステム構築上流工程強化の活動
 - 背景と課題
 - 取組み概要
2. ユーザのための要件定義ガイド
3. システム再構築を成功に導くユーザガイド
4. まとめ



伊勢神宮・内宮(1953年8月)

出典:ウィキメディア・コモンズ
(Wikimedia Commons)

原典:朝日新聞社「朝日新聞 報道写真傑作集
1954」

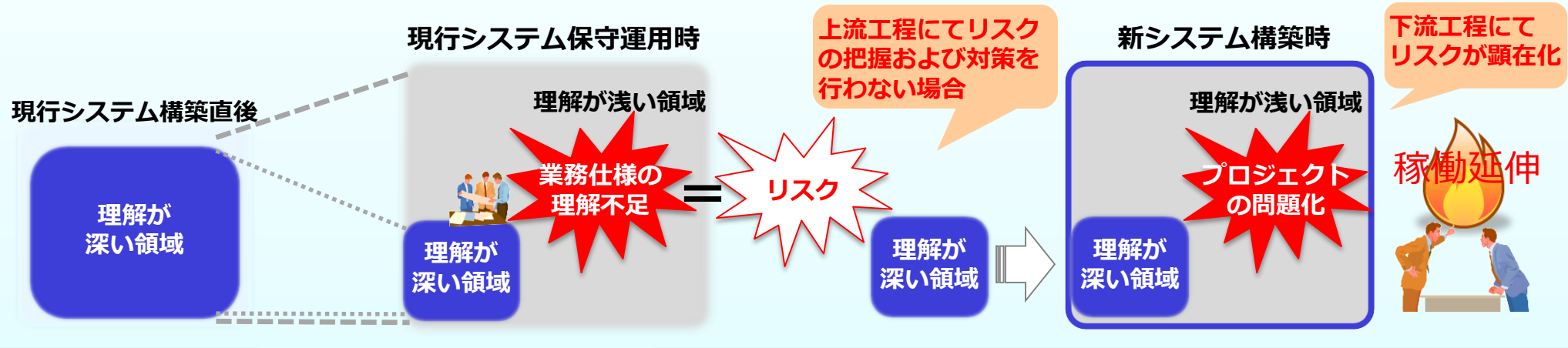


システム再構築の難しさ

長年の保守開発で蓄積された「業務仕様の理解不足」等に伴う**リスク**
上流工程で可視化されず，十分な対策がなされないまま開発に着手
→下流工程において**顕在化**

再構築の難しさ

(例) 業務仕様の理解不足



再構築の
難しさ

人

資産

業務仕様の理解不足

IT人材の後継者不足

ドキュメントの陳腐化

ソースコードの複雑化

乱立する機能・サブシステム

現行
保守
維持

企画・計画

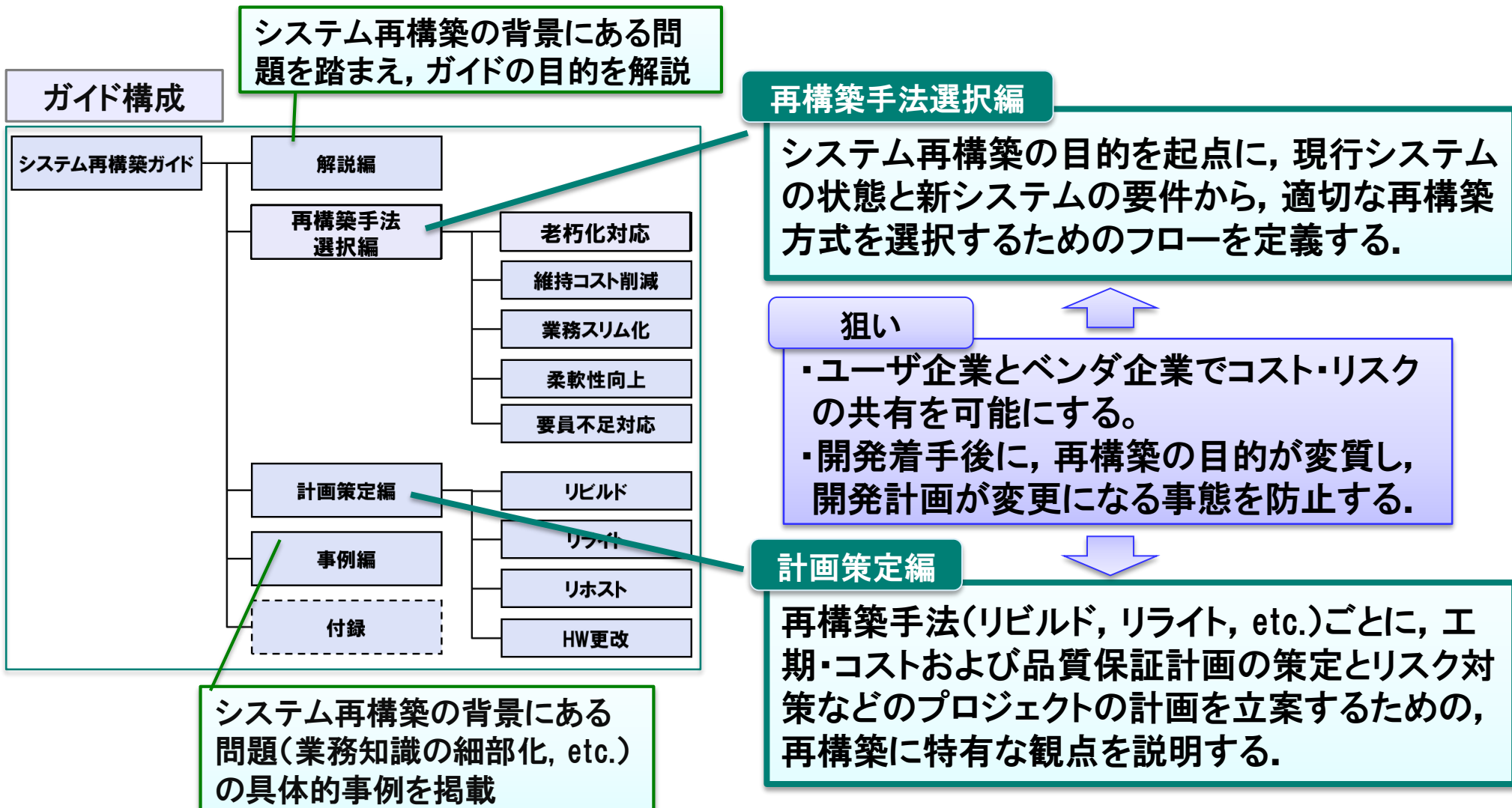
要件定義・
設計

製造

試験

次期
保守
維持

ガイドの構成と、中心となる手法選択・計画策定の内容と狙い



再構築手法選択フロー

対象者(誰が)と内容(何を)

(※注)・責任を持つステークホルダを表す
・他のステークホルダも協力する

- HW SW 老朽化対応
- 維持コスト削減
- 業務スリム化
- 柔軟性向上
- 要員不足対応

システム再構築の目的を
確認



ユーザ企業

新システム
の要求事
項を精査

現行シス
テムの状
態を把握



再構築のリスク抽出後の
共通認識



最適な再構築手法を
選定



新システムの要求事項
分析 (ステップ2)

現行システムの
調査・分析 (ステップ1)

再構築手法の選択
(ステップ3)

再構築手法の候補
を選定する

再構築手法の決定
(ステップ4)

ステップ3で選定した候補の
メリット・デメリット比較と
再構築時のリスクを共有し、
合意形成を図る。

計画策定編へ

ステークホルダの
合意形成を得るまで
繰返し

再構築の種類と特徴

HW更改	リホスト	リライト	リビルド

手法選択は、サブシステムや
業務毎に実施することも可能

はじめに

第1章 解説編

- 1.1 システム再構築をとりまく現状
- 1.2 再構築の現場で何が起きているか
- 1.3 再構築の問題化を防ぐために

第2章 再構築手法選択編

- 2.1 再構築手法選択編概要
- 2.2 現行システム調査・分析(ステップ1)
- 2.3 新システムの要求事項分析(ステップ2)
- 2.4 再構築手法の選択(ステップ3)
- 2.5 再構築手法の決定(ステップ4)
- 2.6 ケーススタディ

第3章 計画策定編

- 3.1 計画策定編概要
- 3.2 要求の確認(観点A)
- 3.3 現行踏襲内容の明確化(観点B)
- 3.4 現行資産活用方針の検討(観点C)
- 3.5 現行業務知識不足への対応(観点D)
- 3.6 品質保証の検討(観点E)
- 3.7 意思決定プロセスの策定(観点F)

- 3.8 データ移行の計画(観点G)
- 3.9 再構築の計画と見積り(観点H)
- 3.10 ケーススタディ

第4章 事例編

- 4.1 (株式会社 NTTデータ)
- 4.2 (富士通株式会社)
- 4.3 (株式会社 日立製作所)
- 4.4 (日本電気株式会社)
- 4.5 (富士通株式会社)
- 4.6 (日本電気株式会社)
- 4.7 (株式会社 ソフトロード)
- 4.8 (株式会社 NTTデータ)
- 4.9 (東京海上日動システムズ株式会社)
- 4.10 (株式会社 日立製作所)
- 4.11 (JFEシステムズ株式会社)

おわりに

付録

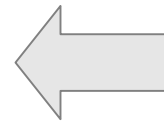
参考文献

解説編 (第1章)

業務知識が失われる背景

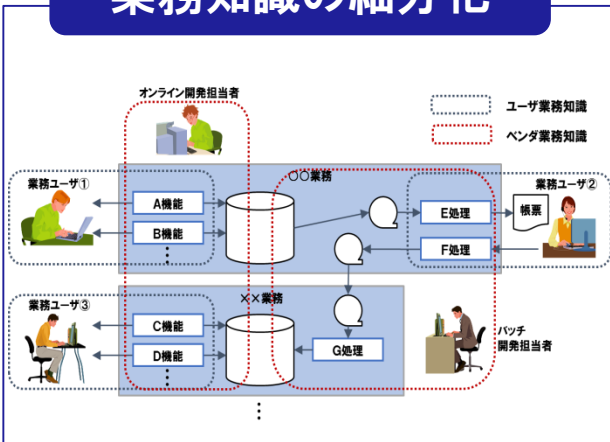
3つの状態：業務知識の

- ◆ 細分化
- ◆ 断片化
- ◆ 領域変化

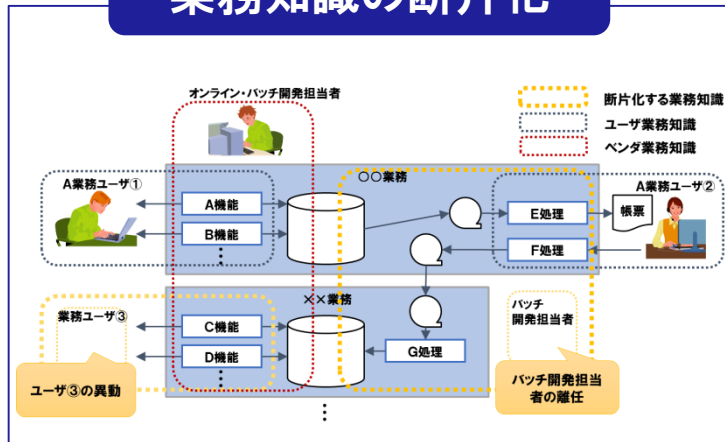


業務仕様の理解不足

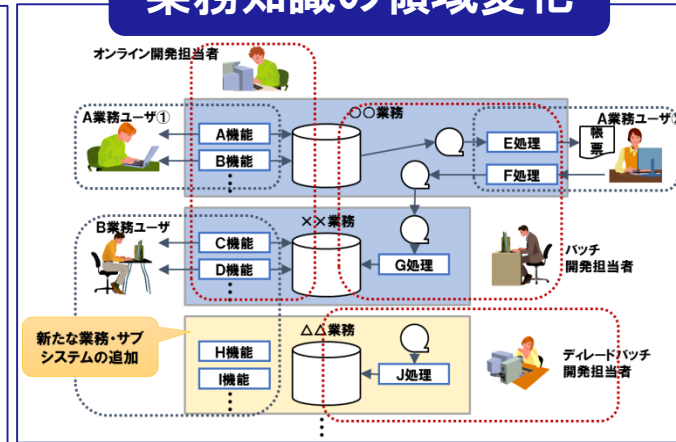
業務知識の細分化

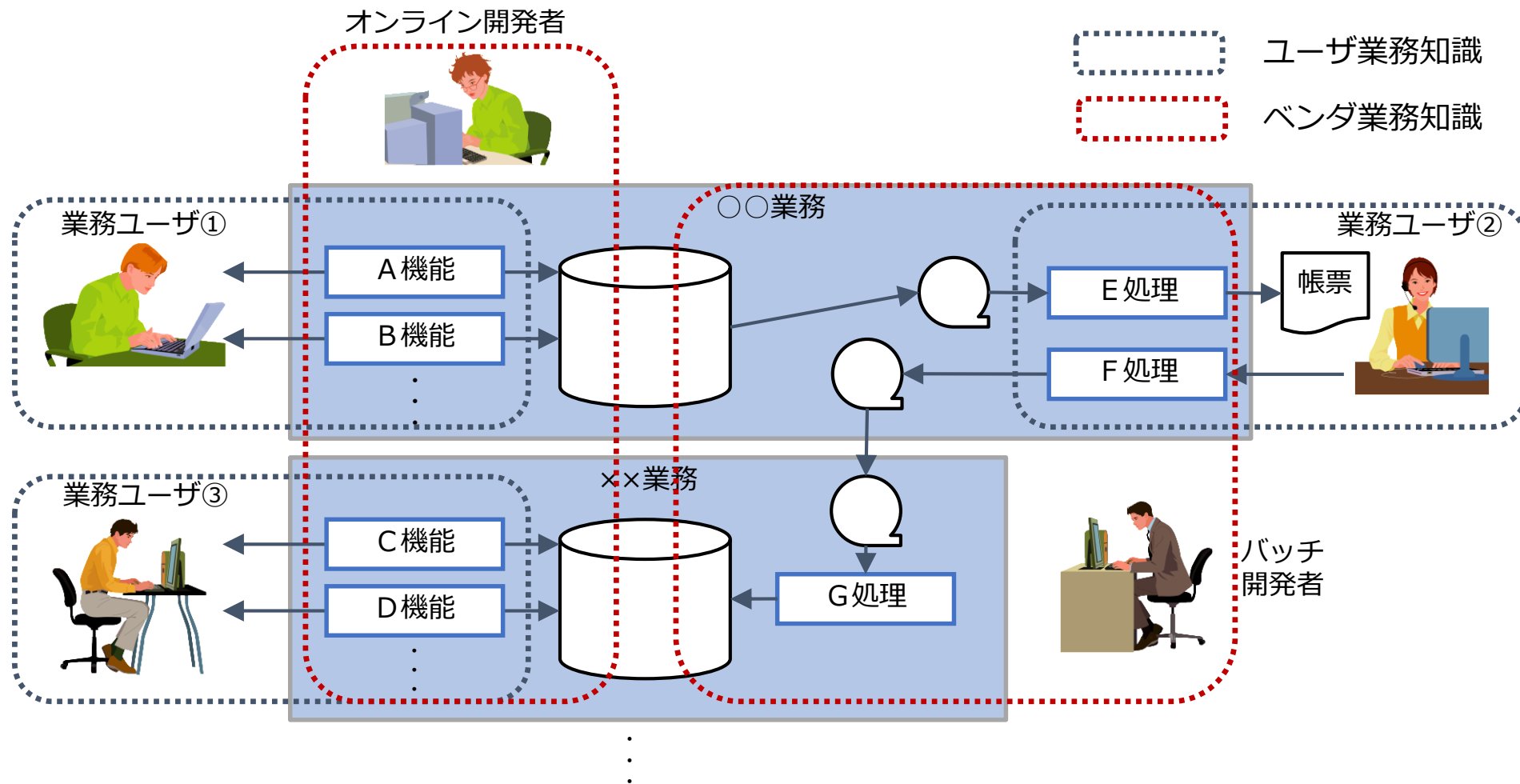


業務知識の断片化



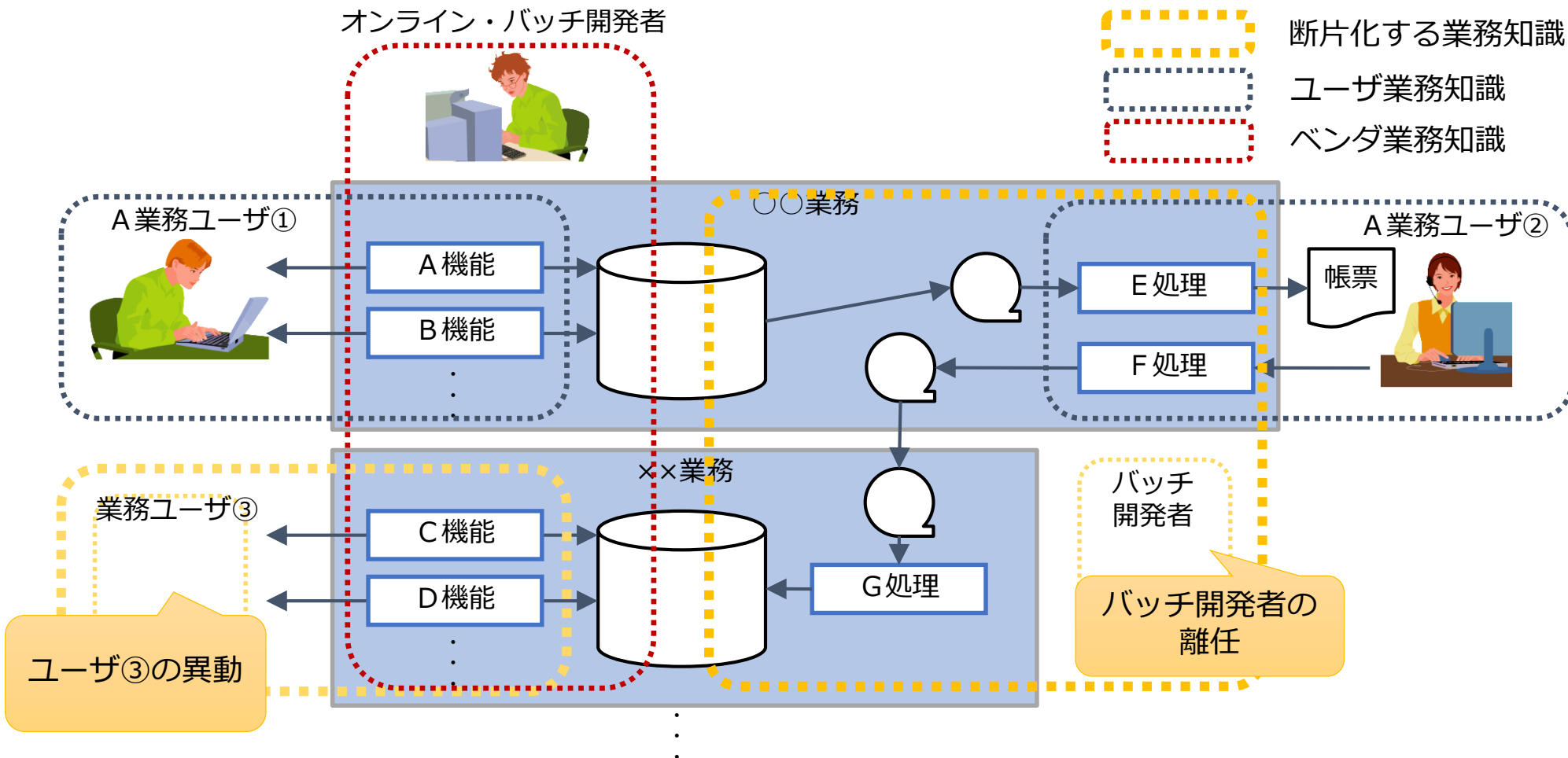
業務知識の領域変化



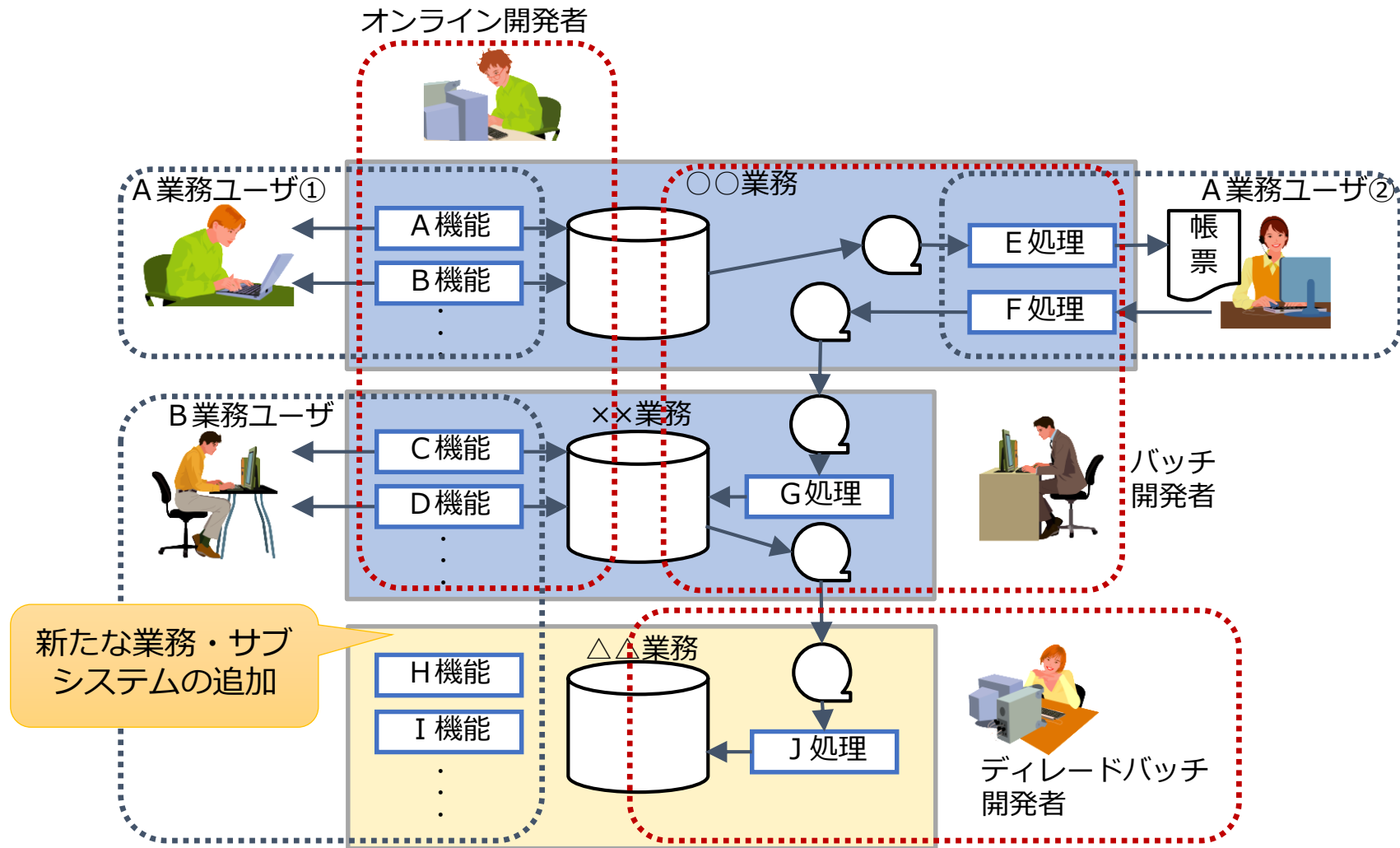


複数のユーザや開発者が業務知識の領域を分けて理解

業務知識の断片化

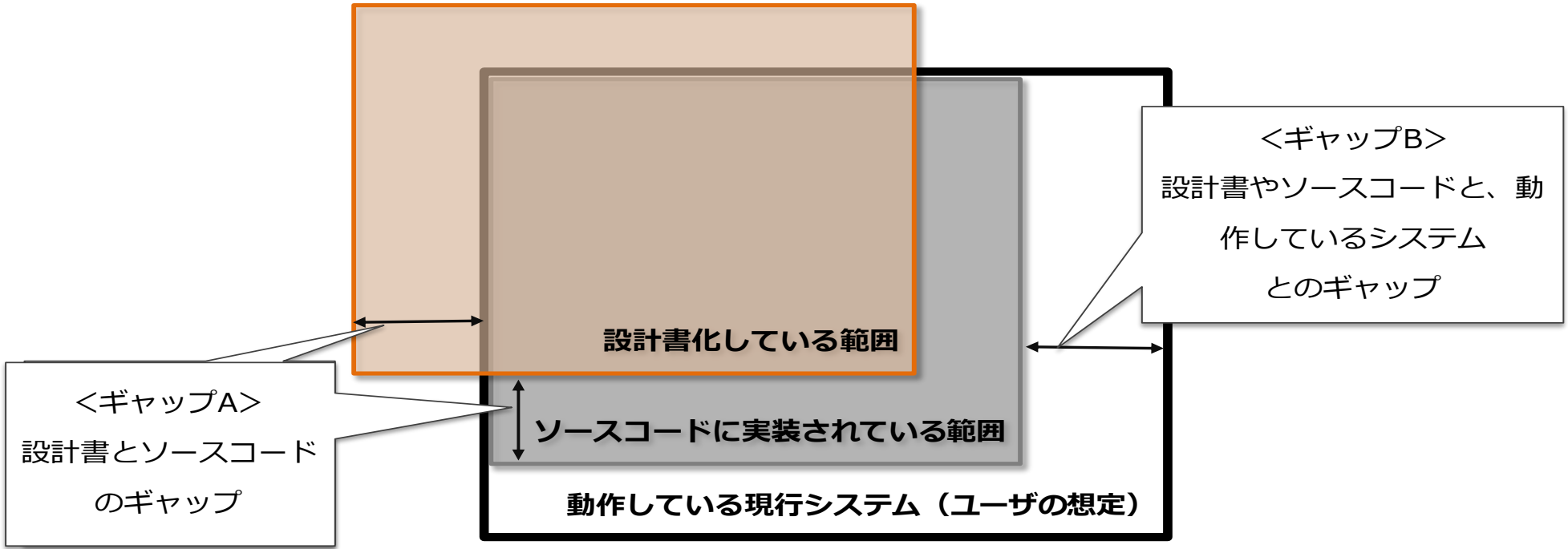


人の交代やメンバーの異動により、一部の業務知識が喪失



ビジネス／技術環境の変化に合わせて、仕様変更を繰り返す

再構築現場の状況

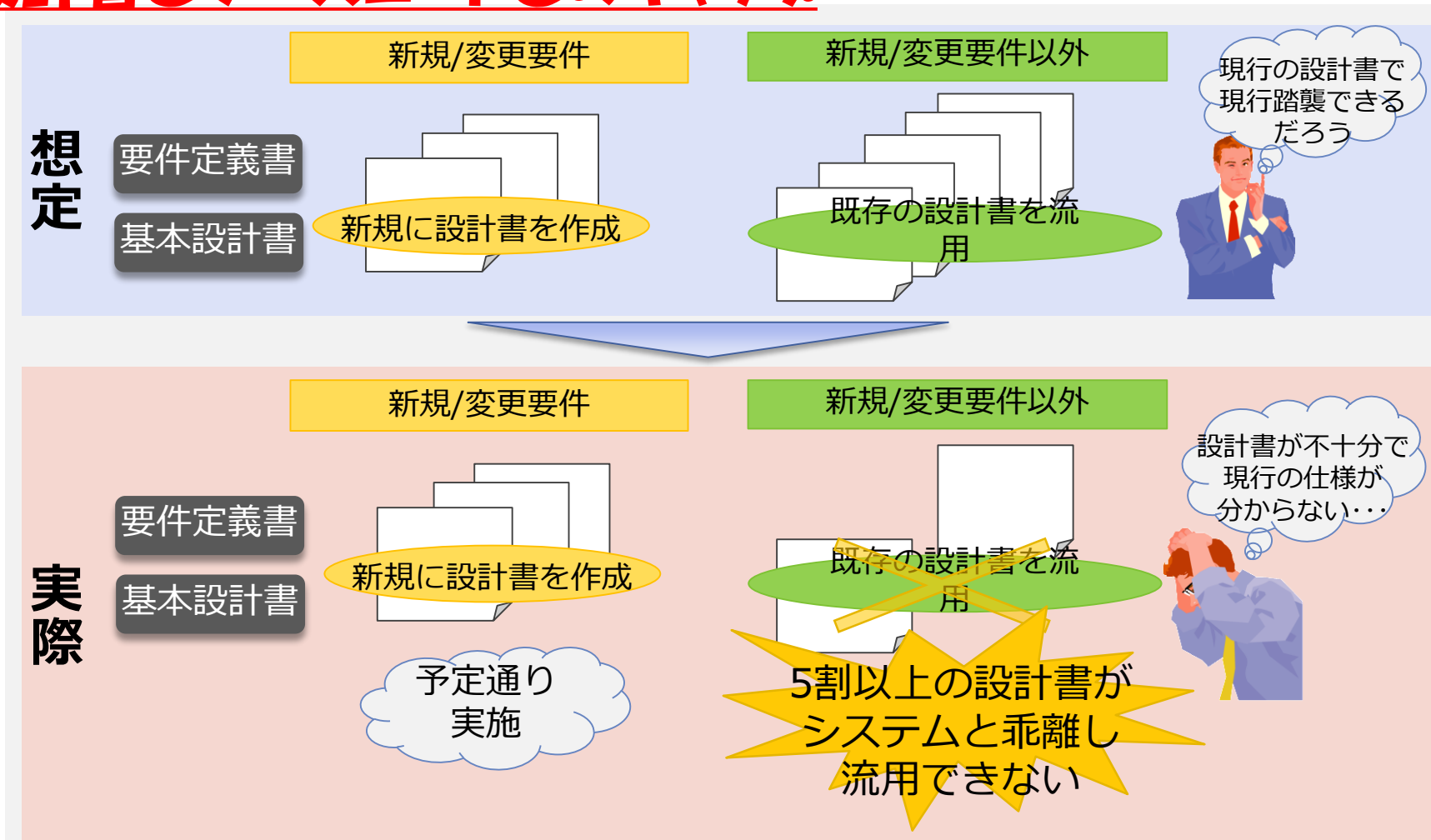


「現行踏襲」の「現行」の認識に、ギャップが発生する場合がある

- ◇ ユーザ企業とベンダ企業との間
- ◇ ユーザ企業内の、システム部門と利用部門との間

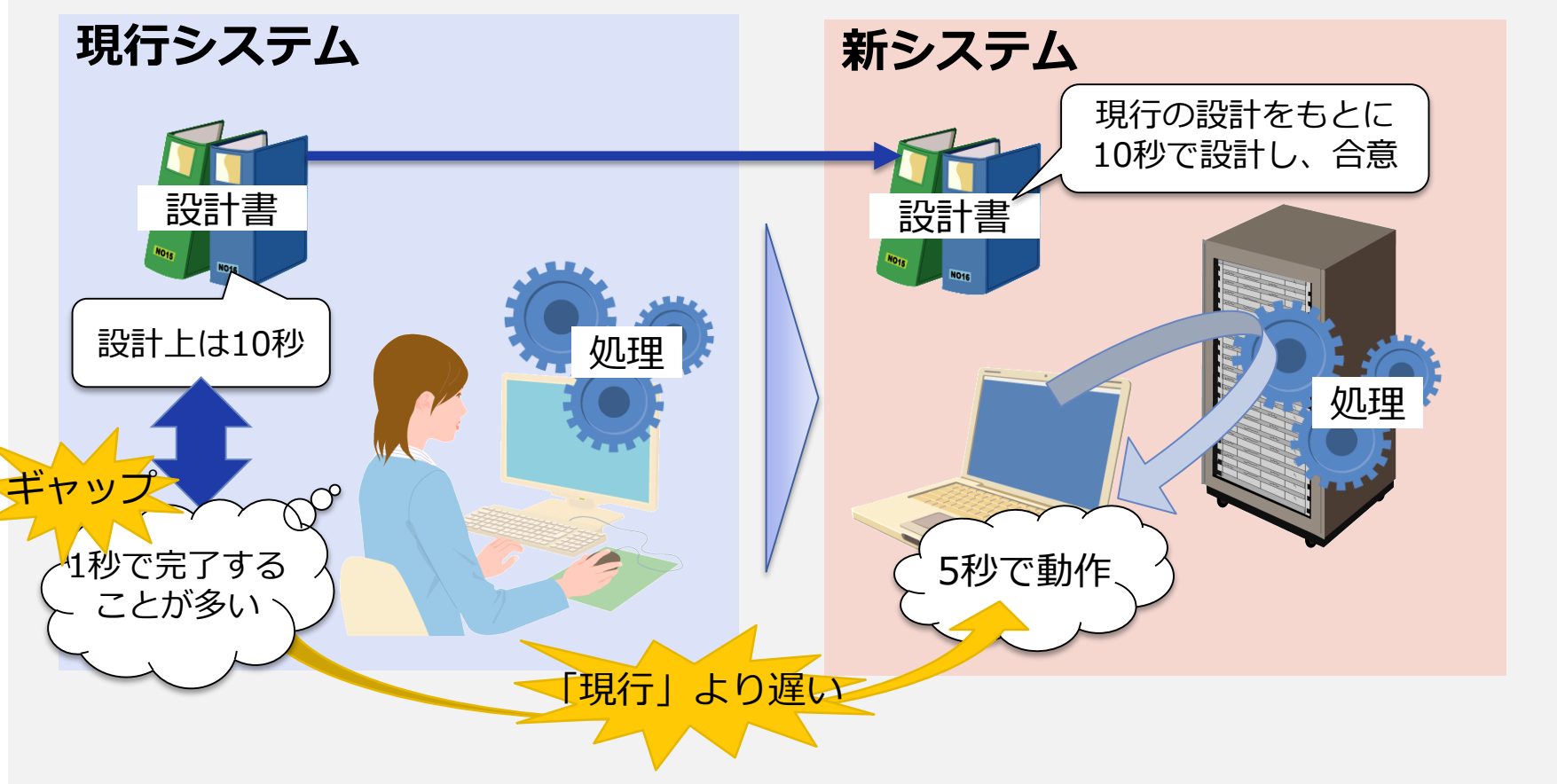
- A 設計書とソースコードとのギャップ
- B 設計書やソースコードと動作しているシステムとのギャップ

設計書とソースコードとのギャップ



機能の実装漏れやユーザ期待と処理結果との不一致が試験工程で多発

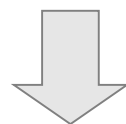
設計書やソースコードと動作しているシステムとのギャップ



**現行より遅く業務が滞るため改善して欲しいという要望が寄せられ、
当該処理を作り直す → 予定していたサービス開始が延期**

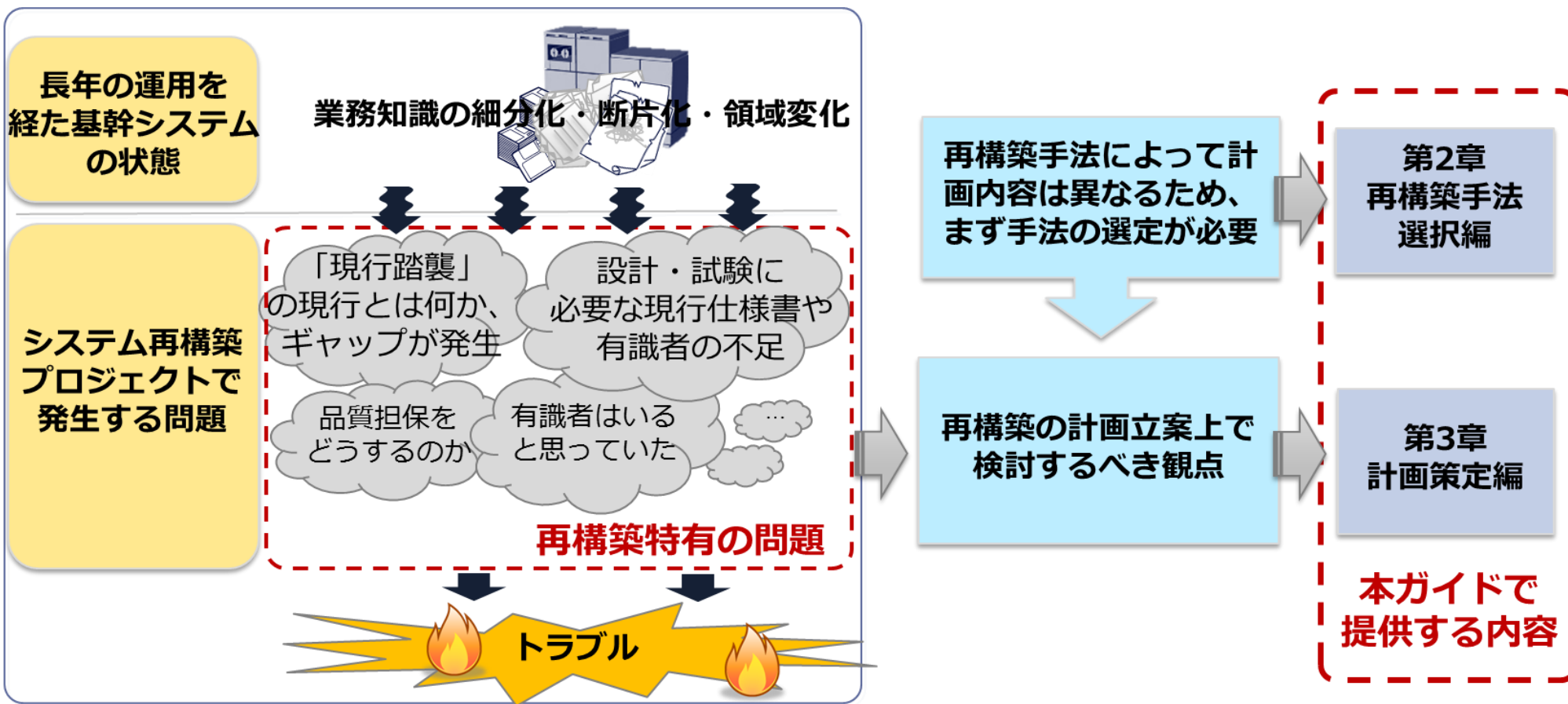
- ① テストをどこまで実施する必要があるかの判断が困難
- ② テスト項目の作成が困難
- ③ トラブル時に解析が困難

テスト項目のインプットとなる設計書類が存在しない
現行保守ベンダはシステム全体にわたる業務知識は有していない



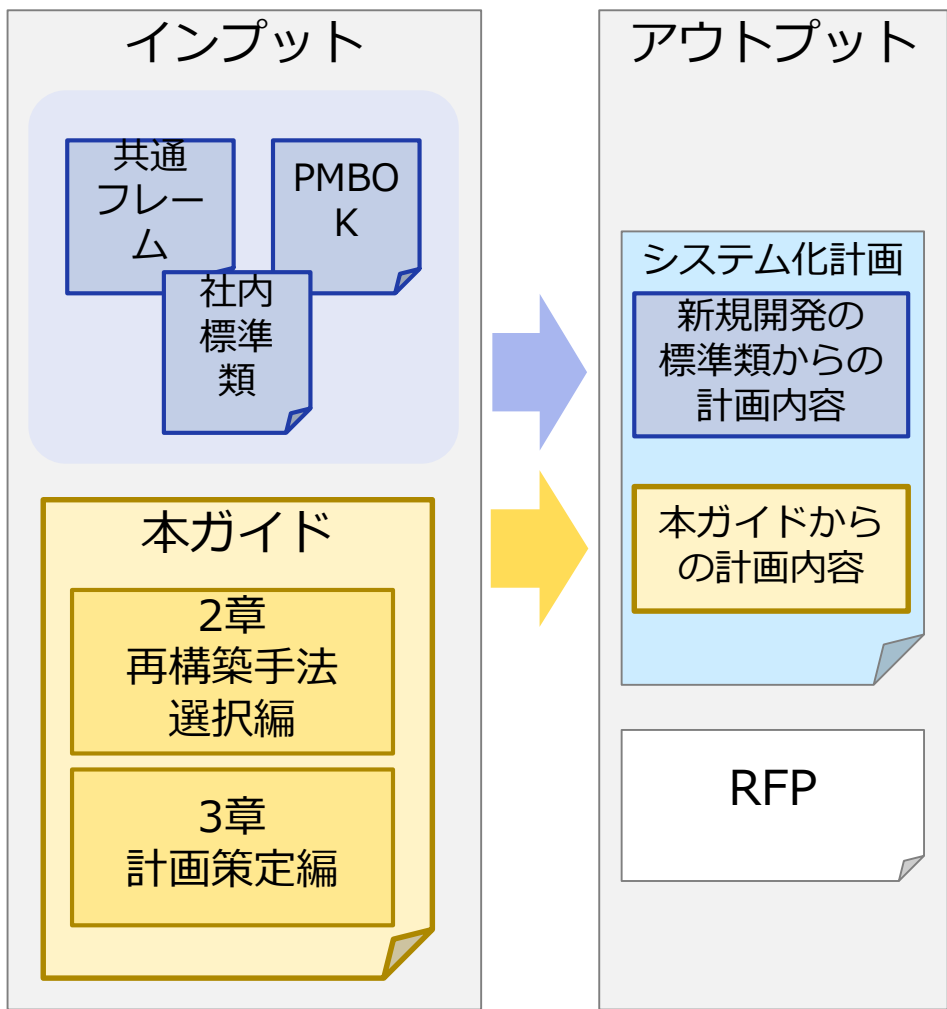
必要なテスト計画が立てられず、数年分のサイクルテスト実施
→テスト期間の大幅延伸、リリース時期延伸

ガイドの目的と内容



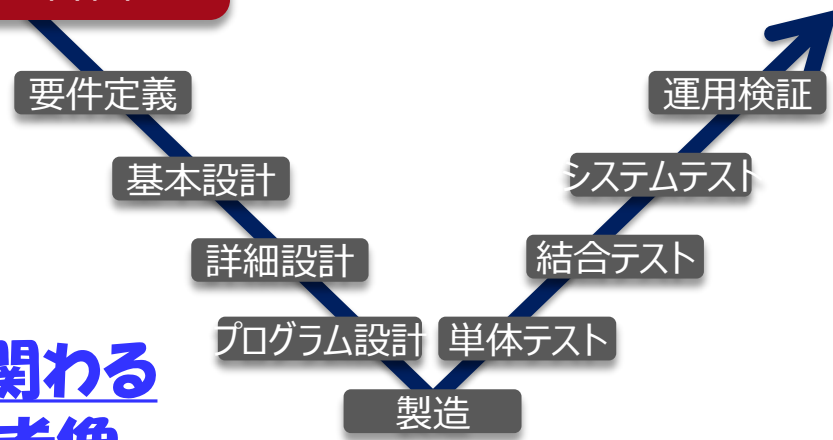
過去の問題化案件の知見から抽出した、再構築の企画段階で検討すべき観点を説明

ガイドの利用イメージ



**システムの再構築の
企画段階において
再構築の構想・計画を
検討する際に利用**

**システム化計画の策定に関わる
ユーザ企業が主対象利用者像**



再構築手法の分類

再構築手法	ハードウェア更改	リホスト	リライト	リビルド
概要	ハードウェアの変更を行う。OS、ミドルウェア製品のバージョンが変わる影響は、プログラムの修正が発生する場合もある。	プログラムは現行と同一の言語で、原則そのまま新規プラットフォームへ移行する。	現行のプログラム設計書をもとに、異なる言語で新たなプログラムを実装する。	現行の基本設計書をもとに新規システム構築。業務要件は変えずに、アプリケーションを事実上作り変える。
要件定義／基本設計				
詳細設計				
プログラム設計				
プログラムソース				
要件定義／基本設計書	現行流用	現行流用	現行流用	現行流用
詳細設計書	現行流用 * 1	現行流用 * 2	現行流用 * 2	再設計
プログラム設計書	現行流用 * 1	現行流用 * 2	現行流用 * 2 言語依存、新規機能分再設計要	再設計
プログラムソース	現行流用 * 1	現行流用 * 2	再生成	再生成

*1 OS、ミドルウェア製品のバージョン変更に伴う修正の場合あり
*2 プラットフォーム変更に伴う修正あり

<凡例> : 非互換の変更
 : 再作成

再構築手法選択編（第2章）

再構築手法選択のプロセス

再構築テーマを起点に、現行システムの状態と新システムの要件から、適切な再構築手法を選択 → 4つのステップ

再構築手法選択プロセス	目的	インプット情報	アウトプット情報	作業概要
ステップ 1	現行システムの把握	関連ドキュメントや運用・保守の状況、有識者の知識など	現行システムの状態	様々な資産の調査や有識者へのヒアリング
ステップ 2	新システムに求める要求の決定	再構築のテーマ	新システム要求事項一覧	再構築の目的を設定して、要求の優先度を定める
ステップ 3	再構築手法の候補を選択	現行システムの状態および新システム要求事項一覧	再構築手法の候補	業務仕様、基盤、開発言語の変更有無を評価
ステップ 4	再構築手法の決定	再構築手法の種類と特徴	決定した再構築手法と根拠	投資効果確認およびリスク把握

1. HW/SW老朽化対応

ハードウェア機種やソフトウェアの販売終了やサポート終了への対応

2. 維持コスト削減

現行システムの維持管理コスト(運用・保守コストなど)の削減

3. 業務スリム化

肥大化, 複雑化した基幹システムのスリム化, 業務改善, 運用の改善

4. 柔軟性向上

新しいビジネスとの連携や新サービスへの対応を容易とする柔軟性の向上

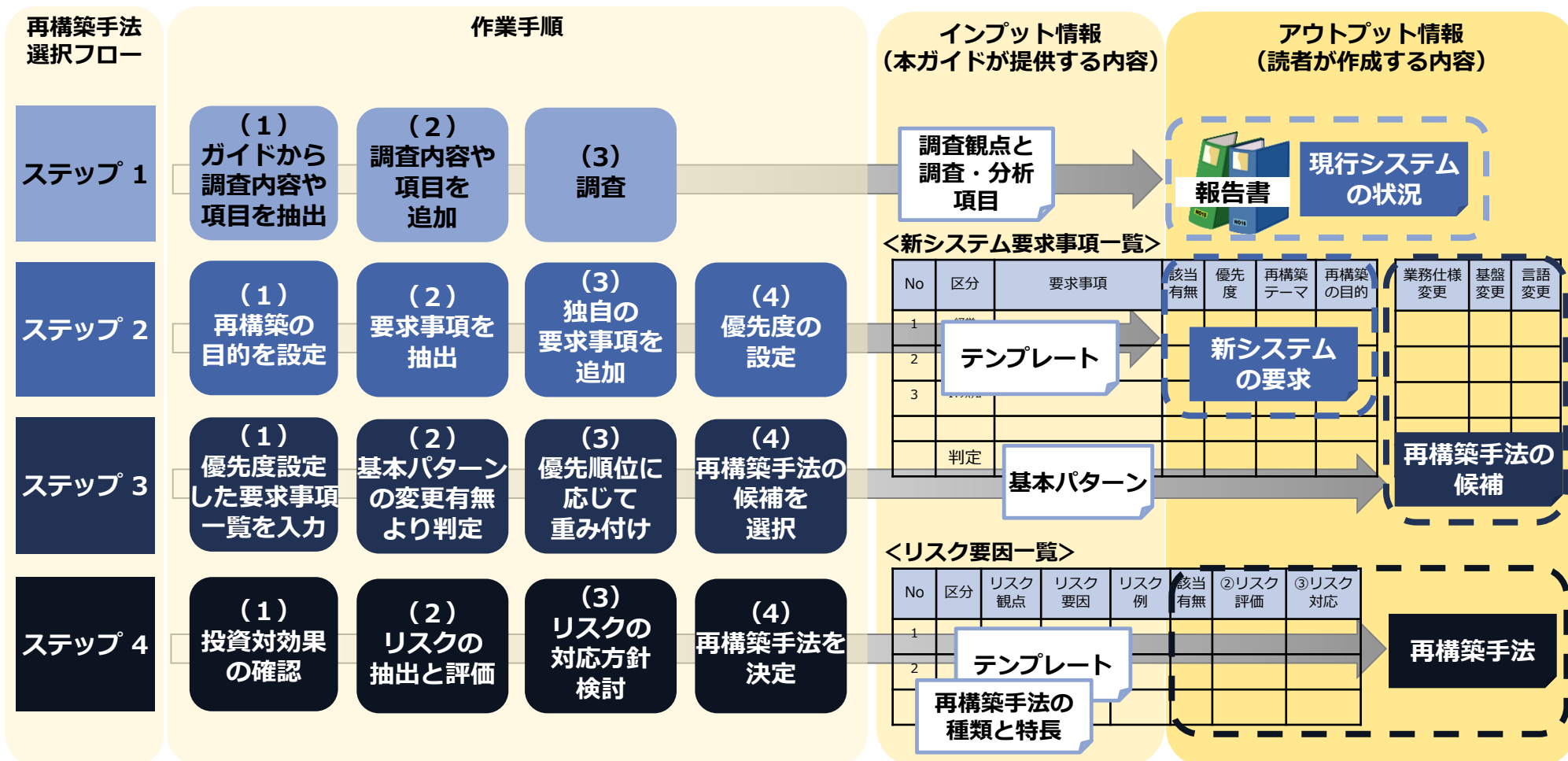
5. 要員不足対応

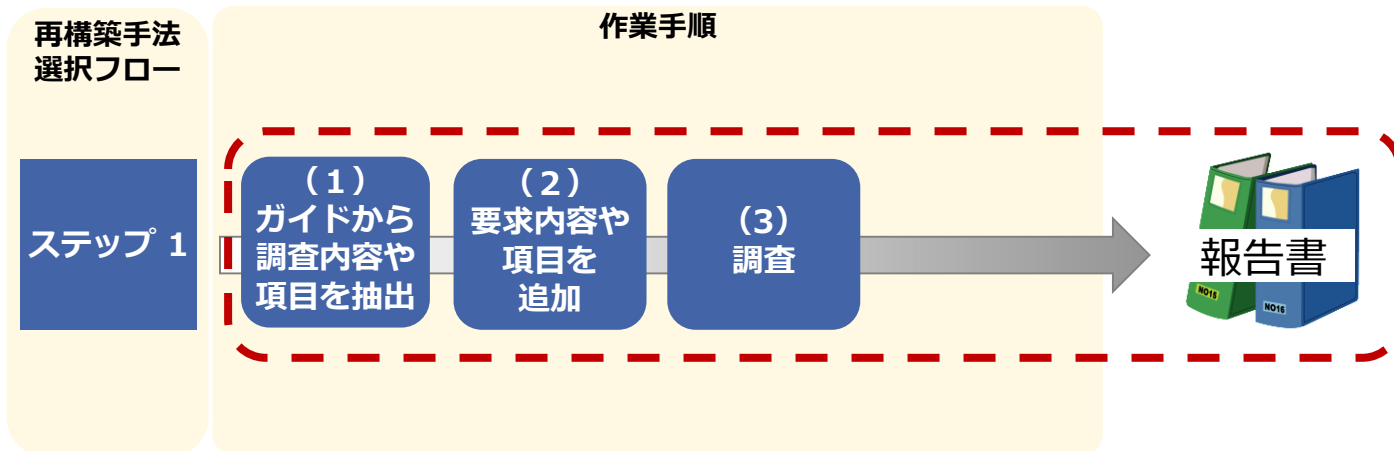
現行システムの保守要員の高齢化や基幹システムの技術者の減少への対応

各企業の現状やニーズに応じ, 独自の目標や目的を設定

再構築手法選択のステップ(概要)

- (ステップ1) 現行システムの調査・分析
- (ステップ2) 新システムの要求事項分析
- (ステップ3) 再構築手法の選択
- (ステップ4) 再構築手法の決定





現行システムの稼動状況や現行資産は膨大

再構築のテーマや目的をもとに分類し、効率的に調査・分析作業を行う

- ・老朽化対応: システム構成やソフトウェア構成を中心に調査・分析
- ・維持コスト削減: 運用・保守の状況を中心に調査・分析

(ステップ1) 現行システムの調査・分析(2/2)

現行システムの調査観点と調査・分析項目例

NO	調査観点	区分	具体的な調査・分析項目の例
1	資産	システムの規模	業務毎の資産規模、本数、言語種別、画面数、帳票数と属性
2		構成	システム構成、ミドルウェア、データベース管理システム、運用ソフトウェア、文字コード体系
3		特性	アプリケーションの複雑度、類似度
4		稼働/未稼働	現運用で稼働している業務資産、未稼働資産の選別
5	業務知識	要員(質的な面)	有識者(業務仕様、基盤、運用・保守など)のスキルセット
6		//(量的な面)	再構築時に必要な有識者(業務仕様、基盤、運用・保守など)を揃えられるか、限定されるか
7		ドキュメント	業務ドキュメントの有無(業務フロー、機能仕様書、処理フローなど)
8		//	上記のドキュメントの最新性が保たれているか
9	処理方式	オン/バッチ	アプリ動作環境、オンライン処理方式、バッチ処理の形態、DBアクセス方法、帳票出力方法
10		外部接続	外部接続先、メッセージ連携、ファイル連携方法、コード変換の有無、種類
11		バックアップ	バックアップの対象、種類、タイミング、リカバリの方法
12	非機能要件	可用性	現行システムの稼働時間、停止時間、稼働率
13		運用・保守性	運用時間、運用監視、異常時運用、運用スケジュール定義方法、システムオペレーション方法
14		性能・拡張性	オンラインレスポンス、バッチ処理時間、スループット、長時間ジョブ情報
15		信頼性	冗長構成の有無、多重化の対象
16		セキュリティ	システム利用に対する有資格者のチェック方法
17		システム環境	プラットフォームやミドルウェアのバージョン、及びサポート切れになるハード・ソフトウェアとその時期
18	業務データ	データ量	ファイル数、テーブル数、データ容量、トラフィック数、伸び率
19		データ構造	論理データ構造、データレイアウト
20	保守	保守状況	システム保守、アプリケーション保守の計画と実績、業務毎のバックログ数、対応期間
21		品質状況	業務毎のインシデント数、傾向分析
22		保守環境	アプリケーション保守環境(言語、ツールなど)、開発規約、資産管理方法、リリース管理

(ステップ2) 新システムの要求事項分析(1/3)_1

再構築手法
選択フロー

ステップ 2

作業手順

- (1) 再構築の目的を設定
- (2) 要求事項を抽出
- (3) 独自の要求事項を追加
- (4) 優先度の設定

インプット情報
(本ガイドが提供する内容)

<新システム要求事項一覧> (2)

No	区分	要求事項	該当有無	優先度	再構築テーマ		再構築の目的
					再構築テーマ	再構築の目的	
1							
2		テンプレート					新システムの要求
3							
	判定						

アウトプット情報
(読者が作成する内容)

(1)

(2)
再構築の目的を起点に、要求事項例の中から適合する要求事項を抽出する。

(1)
再構築テーマの中から、起点となる再構築の目的を定める。ここでは、例のため全てを記載している。各企業独自の目的を設定し、追加、変更することも可能。

<凡例> 該当有無(◆:該当あり、-:該当なし)

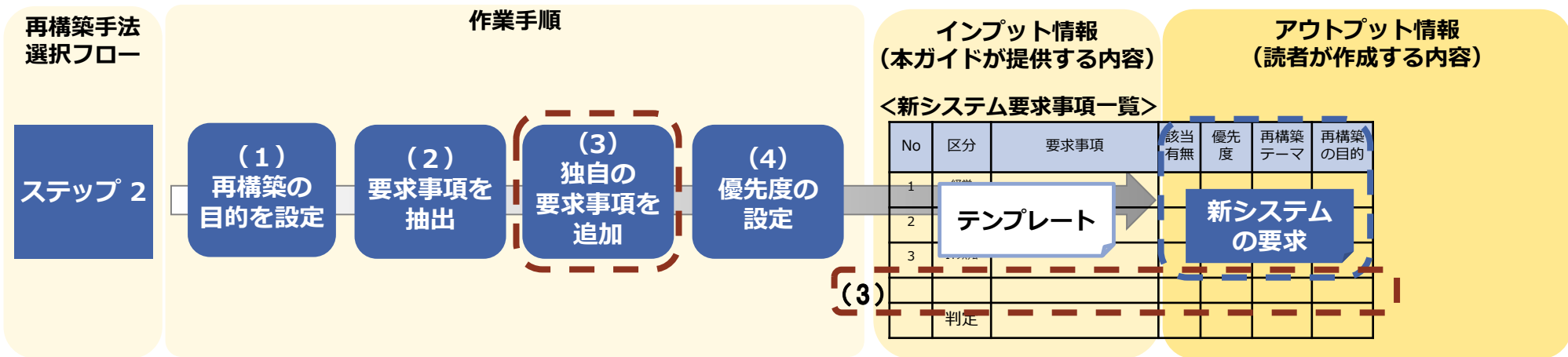
NO	区分	新システムの要求事項	該当有無	優先度	再構築テーマ				
					老朽化対応	維持コスト削減	業務スリム化	柔軟性向上	要員不足対応
1		業務プロセスの効率化(省カ化 業務コスト削減)	◆			レ	レ		

(ステップ2) 新システムの要求事項分析(1/3)_2

(2) (1)
 <凡例> 該当有無(◆:該当あり、-:該当なし)

NO	区分	新システムの要求事項	該当有無	優先度	再構築テーマ				
					老朽化 対応	維持コ スト削減	業務 スリム 化	柔軟性 向上	要員不 足対応
1	経営	業務プロセスの効率化(省力化、業務コスト削減)	◆			レ	レ		
2		事業競争力、営業力の強化	◆					レ	
3		迅速な業績把握、情報把握(リアルタイム経営)	-				レ	レ	
4		開発・運用コストの削減	◆			レ	レ		
5		広域災害対応機能を強化したい(BCP)	-						
6		セキュリティ機能の強化	-						
7		ベンダーロックインを解消したい	-		レ	レ			
8		基盤要員の若手を育成し、既存言語の要員減へ対応したい	-						レ
9		一時的な再構築・移行コストは極力抑えたい	◆			レ			
10	業務	業務開発生産性、アプリケーションの保守性を向上させたい	◆			レ		レ	
11		オンライン業務は、新サービス対応、新チャネル追加のため業務仕様を見直したい	◆					レ	
12		バッチ業務は、基幹系処理であり、業務仕様は踏襲したい	◆						レ
13		業務プロセスのスピードアップ(リードタイム短縮など)	-				レ	レ	
14	ITシステム	最新のIT技術や新機能を使って、運用の効率化、自動化を図りたい	-		レ			レ	

(ステップ2) 新システムの要求事項分析(2/3)_1



(3) 業務部門、システム部門、利用部門などからそれぞれの要求事項を追加し、経営、業務、ITシステムに区分し、全体として再構築の目的に合致するか、方向性や整合性を検証する。

- 【各企業に固有な情報】
- ・ 経営方針、事業目標
 - ・ 業務課題、改善事項
 - ・ システム上の課題、問題点
- など

(ステップ2) 新システムの要求事項分析(2/3)_2

(3)

<凡例> 該当有無(◆:該当あり、-:該当なし)

NO	区分	新システムの要求事項	概要有無	優先度	再構築テーマ				
					老朽化対応	維持コスト削減	業務スリム化	柔軟性向上	要員不足対応
15	ITシステム	現行システムの堅牢な基盤を継続して利用したい	-		レ				
16		現行システムの老朽化した基盤から、最新のインフラに変更したい	◆		レ	レ			
17		現行システムより可用性や信頼性を高めるため、冗長化を実現した新たな基盤に乗せ換えたい	-					レ	
18		現行の開発言語の保守要員の定年から、新しい言語に変更したい	◆			レ			レ
19		開発・保守環境は維持したいので、開発言語は変更する必要ない	-		レ				
20		ホストのセンタオペレーション負荷を削減したい	-			レ			
21		システム運用や監視方法を簡易化したい	-			レ		レ	
22		WSの画面・メニュー構成や操作性(ユーザインターフェース)を改善したい	◆						レ
追加		経営	業務プロセスの質・精度の向上(ミス、欠品削減など)	◆				レ	レ

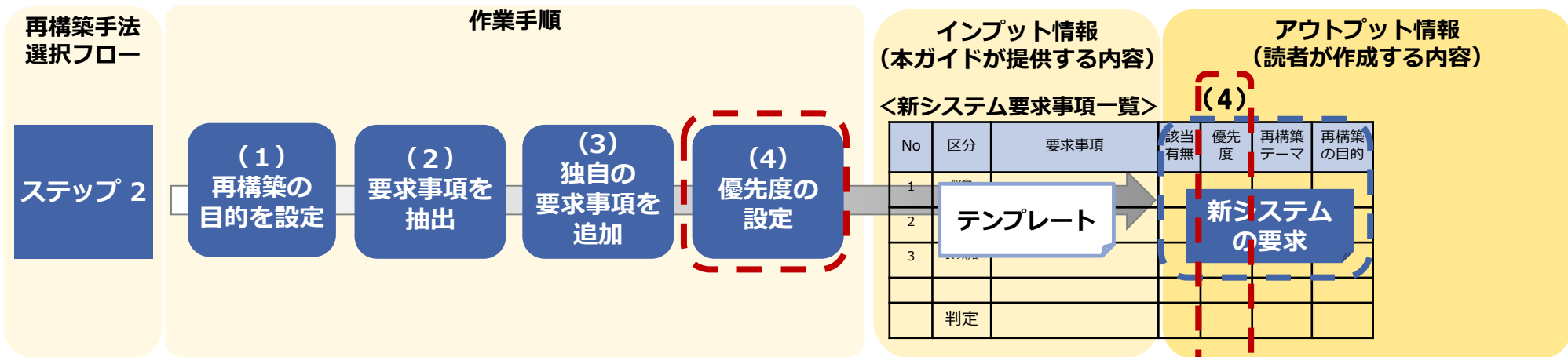
【各企業に固有な情報】

- ・経営方針、事業目標
- ・業務課題、改善事項
- ・システム上の課題、問題点

など

(3)

(ステップ2) 新システムの要求事項分析(3/3)_1



(4) -1
要求内容の重要性、緊急性などにより、優先付けを行い、アウトプットする。
下表では、3段階（◎：必須、○：推奨、△：できれば実現したい）を例示している。

(4) -2
優先度の設定について再構築の目的との関連ではない要因がある場合には、自由記入欄を追加して、背景や理由等を記載する。

<凡例> 該当有無◆：該当あり、-：該当なし
優先度 ◎：必須、○：推奨、△：できれば

NO	区分	新システムの要求事項	該当有無	優先度	再構築の目的				
					老朽化対応	維持コスト削減	業務スリム化	柔軟性向上	要員不足対応
1		業務プロセスの効率化(省力化、業務コスト削減)	▲	◎		レ	レ		

(ステップ2) 新システムの要求事項分析(3/3)_2

(4) -1

<凡例> 該当有無◆:該当あり、-:該当なし
優先度 ◎:必須、○:推奨、△:できれば

NO	区分	新システムの要求事項	該当有無	優先度	再構築の目的				
					老朽化対応	維持コスト削減	業務スリム化	柔軟性向上	要員不足対応
1	経営	業務プロセスの効率化(省力化、業務コスト削減)	◆	◎		レ	レ		
2		事業競争力、営業力の強化	◆	◎				レ	
4		開発・運用コストの削減	◆	◎		レ	レ		
9		一時的な再構築・移行コストは極力抑えたい	◆	○		レ			
追加		業務プロセスの質・精度の向上(ミス、欠品削減など)	◆	○			レ	レ	
10	業務	業務開發生産性、アプリケーションの保守性を向上させたい	◆	○		レ		レ	
11		オンライン業務は、新サービス対応、新チャネル追加のため業務仕様を見直したい	◆	○				レ	
12		バッチ業務は、基幹系処理であり、業務仕様は踏襲したい	◆	○					レ
16	ITシステム	現行システムの老朽化した基盤から、最新のインフラに変更したい	◆	○	レ	レ			
18		現行の開発言語の保守要員の定年から、新しい言語に変更したい	◆	○		レ			レ
22		WSの画面・メニュー構成や操作性(ユーザインターフェース)を改善したい	◆	△				レ	

(ステップ2) 新システムの要求事項分析(3/3)_3

(4) -2

優先度の設定について再構築の目的との関連ではない要因がある場合には、**自由記入欄**を追加して、背景や理由等を記載する。

<凡例> 該当有無:◆:該当あり、-:該当なし
優先度 :◎:必須、○:推奨、△:できれば

NO	区分	新システムの要求事項	該当有無	優先度	再構築の目的
1	経営	業務プロセスの効率化(省力化、業務コスト削減)	◆	◎	不要なプロセスを排除し、運用コスト削減のため
2		事業競争力、営業力の強化	◆	◎	事業課題、経営トップの方針
4		開発・運用コストの削減	◆	◎	運用、保守費用の削減
9		一時的な再構築・移行コストは極力抑えたい	◆	○	一時コスト含めたTCO削減のため
追加		業務プロセスの質・精度の向上(ミス、欠品削減など)	◆	○	業務生産性の向上
10	業務	業務開発生産性、アプリケーションの保守性を向上させたい	◆	○	制度改正等の保守費用の削減
11		オンライン業務は、新サービス対応、新チャネル追加のため業務仕様を見直したい	◆	○	新業態、新サービスへの柔軟で、迅速な対応
12		バッチ業務は、基幹系処理であり、業務仕様は踏襲したい	◆	○	オープン環境への安全で、確実な移行
16	ITシステム	現行システムの老朽化した基盤から、最新のインフラに変更したい	◆	○	現ハードウェアの保守期間、サポート切れ
18		現行の開発言語の保守要員の定年から、新しい言語に変更したい	◆	○	オープン環境への開発要員のシフト
22		WSの画面・メニュー構成や操作性(ユーザインターフェース)を改善したい	◆	△	エンドユーザの利便性の向上

(ステップ3) 再構築手法の選択_1

再構築手法
選択フロー

作業手順

インプット情報
(本ガイドが提供する内容)

アウトプット情報
(読者が作成する内容)

<新システム要求事項一覧>

No	区分	要求事項	該当有無	優先度	再構築テーマ	再構築の目的	再構築手法の候補		
							業務仕様変更	基盤変更	言語変更
1	経営								
2	業務								
3	ITシステム								
	判定								

新システムの
要求

基本パターン

再構築手法の
候補

ステップ 3

(1)
優先度設定
した要求事項
一覧を入力

(2)
基本パターン
の変更有無
より判定

(3)
優先順位に
応じて
重み付け

(4)
再構築手法の
候補を
選択

<凡例> 該当有無(◆:該当あり、-:該当なし)
優先度(◎:必須、○:推奨、△:できれば)

Step2のアウトプット例から

NO	区分	新システムへの要求事項	該当有無	優先度	業務仕様変更		基盤変更		言語変更	
					あり	なし	あり	なし	あり	なし
1		業務プロセスの効率化(省力化、業務コスト削減)	◆	◎	レ					
18	ITシステム	現行の閉鎖言語の保守委員の不足から、新しい言語に変更したい	◆	○					レ	
22		WSの画面・メニュー構成や操作性(ユーザインターフェース)を改善したい	◆	△	レ		レ		レ	
		◎:必須(3ポイント)、○:推奨(2ポイント)、△:できれば(1ポイント)で集計した場合			11	4	11	2	5	4

(ステップ3) 再構築手法の選択_2



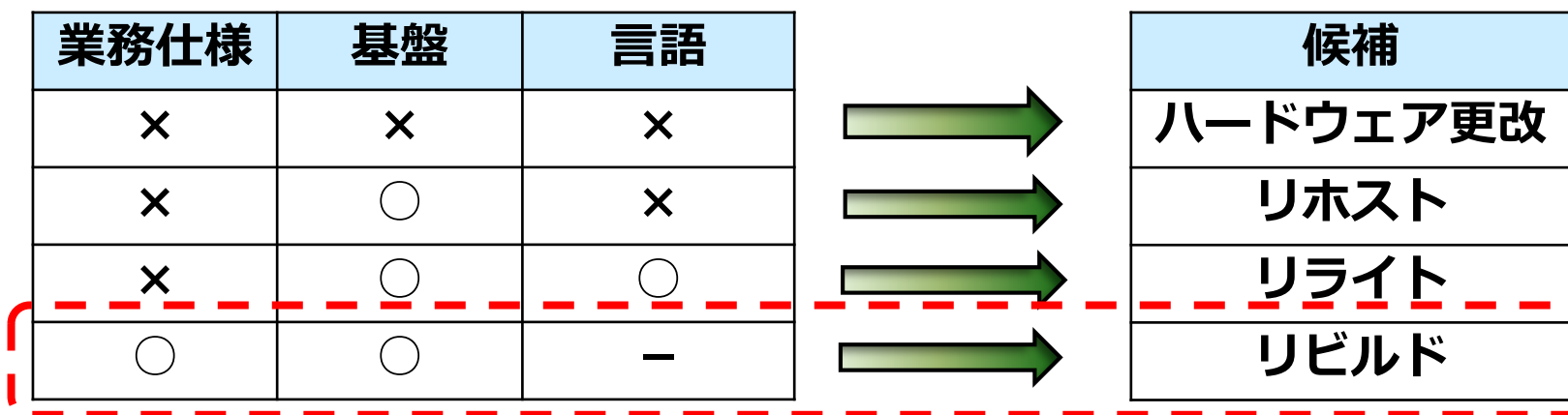
<凡例> 該当有無 (◆: 該当あり, -: 該当なし)
優先度 (◎: 必須, ○: 推奨, △: できれば)

Step2のアウトプット例から

NO	区分	新システムへの要求事項	該当有無	優先度	業務仕様変更		基盤変更		言語変更		
					あり	なし	あり	なし	あり	なし	
1		業務プロセスの効率化(省力化、業務コスト削減)	◆	◎	√						
18	システム	現行の閉鎖言語の保守委員の定年から、新しい言語に変更したい	◆	○					√		
22		WSの画面・メニュー構成や操作性(ユーザインターフェース)を改善したい	◆	△	√		√		√		
◎: 必須(3ポイント)、○: 推奨(2ポイント)、△: できれば(1ポイント)で集計した場合						◇11	4	◇11	2	◇5	4

○ ○ ○

<凡例> (○: 変更あり, ×: 変更なし, -: どちらでも可)



再構築におけるリスク要因例

NO	区分	リスク観点	再構築におけるリスク要因	リスク例	① 該当有無	②リスク評価(注)			
						HW 更改	リホ スト	リラ イト	リビ ルド
1	要求	現行踏襲 (機能)	・現行から受け継ぐべき業務運用や画面・帳票、データ仕様、外部インターフェースなどの対象や範囲は明確か？	・作り込み品質の不良 ・サービス開始の延伸 ・作業工数の増大					
2	要求	現行踏襲 (非機能)	・移行先のシステムの運用方法や性能、操作性、可用性、信頼性などの非機能要件は明確か？	・テスト、運用段階になり、利用者から「使えない」、「レベルダウン」とのクレームが起こる。					
3	要求	再構築 サービスの 選定	・適用したいサービスやソリューションは、現行と同じ機能、運用を全て実現できるか？(特にリホストの場合) (例. 現行システムを充分エミュレートし、全て非互換対応できるか?) ・役割分担や制限、制約事項は明確になっているか？	・想定どおり動作しない不具合の多発による手戻り ・サービス開始の延伸 ・作業工数の増大					
4	要求	アーキテク チャ	・最新技術の適用や初めてのアーキテクチャ採用時には、十分な適用検討と技術検証を行っているか？ ・アーキテクチャが混在することはないか?(例. 現新混在)	・想定どおり動作しない不具合の多発による手戻り、対応工数の増大 ・保守要員の増加 ・保守環境の複雑化					

(ステップ4) 再構築手法の決定(0/3)_リスク要因(2/2)

N O	区分	リスク 観点	再構築におけるリスク要因	リスク例	① 該当 有無	②リスク評価(注)			
						HW 更改	リホ スト	リラ イト	リビ ルド
5	現状	システムの規模	・現行システムは肥大化・複雑化していないか？ (例えば、未稼働資産が多数存在する)	・業務仕様の把握不足による手戻り ・作業工数の増大					
6	現状	業務知識不足(要員)	・現行業務有識者は、後継者不足になっていないか？(現行システム仕様は掌握できるか。) ・再構築時の要員は確保可能か？	・業務仕様の把握不足による手戻り ・作り込み品質の不良 ・試験による検証の長期化 ・サービス開始の延伸 ・作業工数の増大					
7	現状	// (ドキュメント)	・業務ドキュメントとアプリケーションは一致しているか？ ・ブラックボックス化により影響調査の精度が低下していないか？	・業務仕様の把握不足による手戻り ・作り込み品質の不良 ・サービス開始の延伸 ・作業工数の増大					
8	現状	資産管理	・現行稼働中資産と移行資産のアンマッチ(例. 実行モジュールとソースコードのアンマッチ)や ・未検証資産の移行(開発中資産)は、ないか？	・想定どおり動作しない不具合発生による手戻り、対応工数の増大					
9	現状	データ移行	・データ仕様の把握不足によるデータ変換の誤り(フォーマット、文字コード)や ・異常データの混入は考えられないか？	・現行システムと新システムで結果が一致しない。 ・移行データ調査や対応工数の増大					

(ステップ4) 再構築手法の決定(1/3)_1

再構築手法
選択フロー

作業手順

ステップ 4

(1)
投資対効果
の確認

(2)
リスクの
抽出と評価

(3)
リスクの
対応方針
検討

(4)
再構築手法を
決定

インプット情報
(本ガイドが提供する内容)

アウトプット情報
(読者が作成する内容)

<リスク要因一覧>

No	区分	リスク 観点	リスク 要因	リスク 例	該当 有無	②リスク 評価	③リスク 対応
1							
2							

テンプレート

再構築手法の
種類と特長

再構築手法

NO	区分	リスク 観点	再構築におけるリスク要因	①該当 リスク例	① 該当 有無	②リスク評価		
						リホスト	リライト	リビルド
1	要求	現行踏襲 (機能)	・現行から受け継ぐべき業務運用や画面・帳票、データ仕様、外部インターフェースなどの対象や範囲は明確か？	・作り込み品質の不良や ・サービス開始の延伸 ・作業工数の増大	あり	△	○	◎
2	要求	現行踏襲 (非機能)	・移行先のシステムの運用方法や性能、操作性、可用性、信頼性などの非機能要件は明確か？	・テスト、運用段階になり、利用者から「使えない」、「レベルダウン」とのクレームが起こる。	あり	◎	○	○
3	要求	再構築 サービスの選 定	・適用するサービスやソリューションは、現行と同じ機能、運用を全て実現できるか？(特にリホストの場合) ・役割分担や制限、制約事項は明確になっているか？	・想定どおり動作しない不具合の多発による手戻り ・サービス開始の延伸 ・作業工数の増大	なし	—	—	—

①該当リスクの特定

リスク例

②リスク顕在化の相対評価

(ステップ4) 再構築手法の決定(1/3)_2

NO	区分	リスク観点	再構築におけるリスク要因	リスク例	① 該当 有無	②リスク評価		
						リホスト	リライト	リビルド
1	要求	現行踏襲 (機能)	・現行から受け継ぐべき業務運用や画面・帳票、データ仕様、外部インターフェースなどの対象や範囲は明確か？	・作り込み品質の不良や ・サービス開始の延伸 ・作業工数の増大	あり	△	○	◎
2	要求	現行踏襲 (非機能)	・移行先のシステムの運用方法や性能、操作性、可用性、信頼性などの非機能要件は明確か？	・テスト、運用段階になり、利用者から「使えない」、「レベルダウン」とのクレームが起こる。	あり	◎	○	○
3	要求	再構築 サービスの選 定	・適用するサービスやソリューションは、現行と同じ機能、運用を全て実現できるか？(特にリホストの場合) ・役割分担や制限、制約事項は明確になっているか？	・想定どおり動作しない不具合の多発による手戻り ・サービス開始の延伸 ・作業工数の増大	なし	—	—	—
4	要求	アーキテク チャ	・最新技術の適用や初めてのアーキテクチャ採用時には、十分な適用検討と技術検証を行っているか？ ・アーキテクチャが混在することはないか？(例、現新混在)	・想定どおり動作しない不具合の多発による手戻り、対応工数の増大 ・保守要員の増加 ・保守環境の複雑化	なし	—	—	—
5	現状	システムの規 模	・現行システムは肥大化・複雑化していないか？(例えば、未稼働資産が多数存在する)	・業務仕様の把握不足による手戻り ・作業工数の増大	なし	—	—	—
6	現状	業務知識不足 (要員)	・現行業務有識者は、後継者不足になっていないか？(現行システム仕様は掌握できるか。) ・再構築時の要員は確保可能か？	・業務仕様の把握不足による手戻り ・作り込み品質の不良 ・試験による検証の長期化 ・サービス開始の延伸 ・作業工数の増大	あり	○	○	◎
7	現状	” (ドキュメント)	・業務ドキュメントとアプリケーションは一致しているか？ ・ブラックボックス化により影響調査の精度が低下していないか？	・業務仕様の把握不足による手戻り ・作り込み品質の不良 ・サービス開始の延伸 ・作業工数の増大	あり	△	○	◎
8	現状	資産管理	・現行稼働中資産と移行資産のアンマッチ(例、実行モジュールとソースコードのアンマッチ)や ・未検証資産の移行(開発中資産)は、ないか？	・想定どおり動作しない不具合発生による手戻り、対応工数の増大	なし	—	—	—
9	現状	データ 移行	・データ仕様の把握不足によるデータ変換の誤り(フォーマット、文字コード)や ・異常データの混入は考えられないか？	・現行システムと新システムで結果が一致しない。 ・移行データ調査や対応工数の増大	なし	—	—	—

(ステップ4) 再構築手法の決定(2/3)

再構築手法
選択フロー

作業手順

ステップ 4

(1)
投資対効果
の確認

(2)
リスクの
抽出と評価

(3)
リスクの
対応方針
検討

(4)
再構築手法を
決定

インプット情報
(本ガイドが提供する内容)

アウトプット情報
(読者が作成する内容)

<リスク要因一覧>

No	区分	リスク 観点	リスク 要因	リスク 例	該当 有無	②リスク 評価	③リスク 対応
1							
2							

テンプレート

再構築手法の
種類と特長

再構築手法

(3) リスク対応の観点や区分は、テーラリングして使用する

<凡例> リスク評価:リスクが顕在化する相対的な度合い(◎:高、○:中、△:低)
リスク対応:●検討対象とするリスク対応方針)

No	区分	リスク 観点	再構築におけるリスク要因	リスク例	① 該当 有無	②リスク評価			③リスク対応			
						リホ スト	リラ イト	リビル ド	回避	軽減	受容	転嫁
1	要求	現行踏襲 (機能)	・現行から受け継ぐべき業務運用や画面・帳票、データ仕様、外部インターフェースなどの対象や範囲は明確か？	・作り込み品質の不良 ・サービス開始の延伸 ・移行作業工数の増大	あり	△	○	◎	●	●		
2	要求	現行踏襲 (非機能)	・移行先のシステムの運用方法や性能、操作性、可用性、信頼性などの非機能要件は明確か？	・テスト、運用段階になり、利用者から「使えない」、「レベルダウン」とのクレームが起こる。	あり	◎	○	○	●	●		
6	現状	業務知識不足 (要員)	・現行業務有識者は、後継者不足になっていないか？(現行システム仕様は掌握できるか？) ・再構築時の要員は確保可能か？	・業務仕様の把握不足による手戻り ・作り込み品質の不良 ・試験による検証の長期化 ・サービス開始の延伸	あり	○	○	◎		●		

(ステップ4) 再構築手法の決定(3/3)_1

再構築手法
選択フロー

作業手順

インプット情報
(本ガイドが提供する内容)

アウトプット情報
(読者が作成する内容)

ステップ 4

(1)
投資対効果
の確認

(2)
リスクの
抽出と評価

(3)
リスクの
対応方針
検討

(4)
再構築手法を
決定

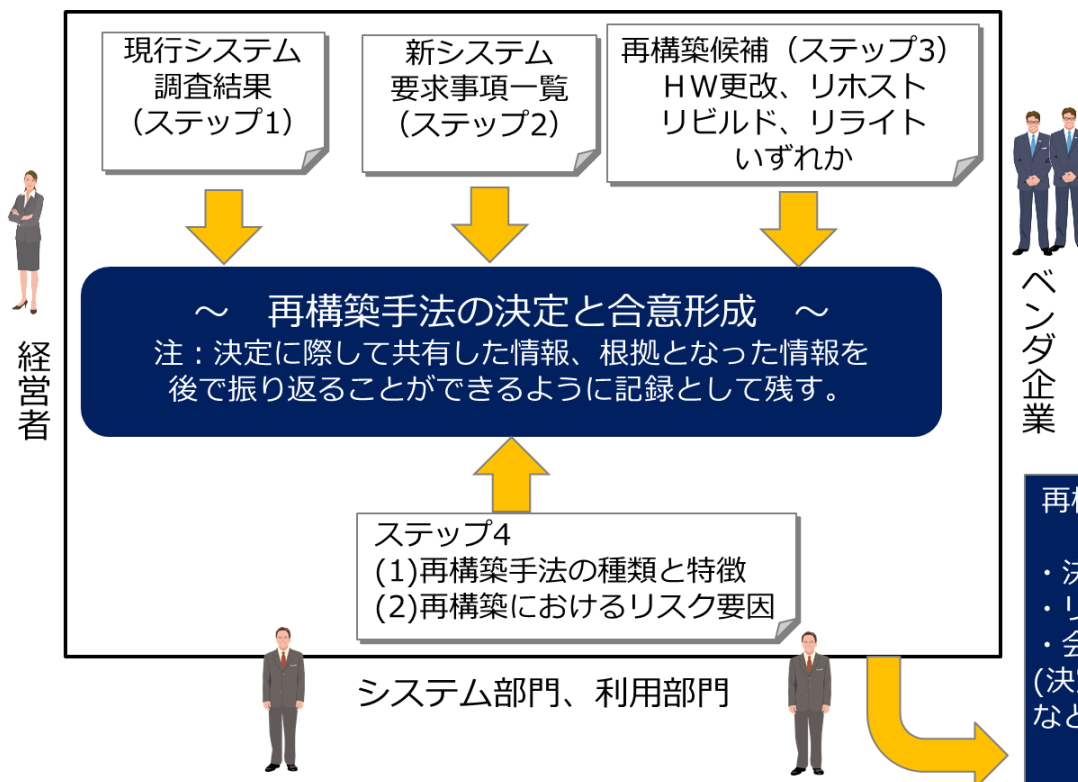
<リスク要因一覧>

No	区分	リスク 観点	リスク 要因	リスク 例	該当 有無	②リスク 評価	③リスク 対応
1							
2							

テンプレート

再構築手法の
種類と特長

再構築手法



再構築手法のメリット・デメリットや投資・効果および発生し得るリスクの共有と対応方針，考え方について，ステークホルダ(経営者，業務部門，システム部門，運用部門など)で合意形成を図り，再構築手法を決定

主な審議内容

- (a) 現行システムの調査・分析結果による，現状の認識合わせ
- (b) 新システムの要求事項の優先度が再構築の目的と方向性が合致しているか，選択した再構築手法と整合しているか
- (c) 再構築手法のメリット・デメリットや再構築への投資と構築後の効果
- (d) 再構築において発生し得るリスク要因の共有とリスクへの対応方針

計画策定編（第4章）

計画策定編の利用イメージ

計画策定時の対策検討・実施の観点 8個

再構築特有の観点

再構築手法選択編 ステップ4で抽出するリスク

NO	区分	リスク観点	再構築におけるリスク要因 (ハザード)	リスク例	該当有無
1	要求	現行踏襲 (機能)	・現行踏襲する資産、機能、移行条件、動作条件は明確か？ ・変える、変えないは明確か？	・移行作業品質の低下 ・サービス開始の延伸 ・移行作業工数の増大	有り
2	要求	現行踏襲 (非機能)	・移行先のシステムで性能、操作性、可用性、信頼性などの非機能要件は明確ですか？	・テスト、運用段階になり、利用者から「使えない」、「レベルダウン」とのクレームが起こる。	有り
6	現状	業務知識不足 (要員)	・現行業務有識者は、後継者不足になっていませんか？ (現行システム仕様は掌握できるか。) ・再構築時の要員は確保可能ですか？	・業務仕様の把握不足による手戻り ・移行作業品質の低下 ・サービス開始の延伸 ・移行作業工数の増大	有り
7	現状	" (ドキュメント)	・業務ドキュメントとアプリケーションは一致していますか？ ・ブラックボックス化により影響箇所調査の精度が低下していませんか？	・業務仕様の把握不足による手戻り ・移行作業品質の低下 ・サービス開始の延伸 ・移行作業工数の増大	有り

計画策定編

観点A

再構築手法選択編で抽出した要求の整理

観点B~G

リスク対策の検討

観点H

リスク対策の検討結果を基にしたシステム化計画・見積りの策定

通常開発の観点

共通フレーム

社内標準類

システム化計画

通常開発の標準類を基にした計画内容

計画策定編を基にした計画内容

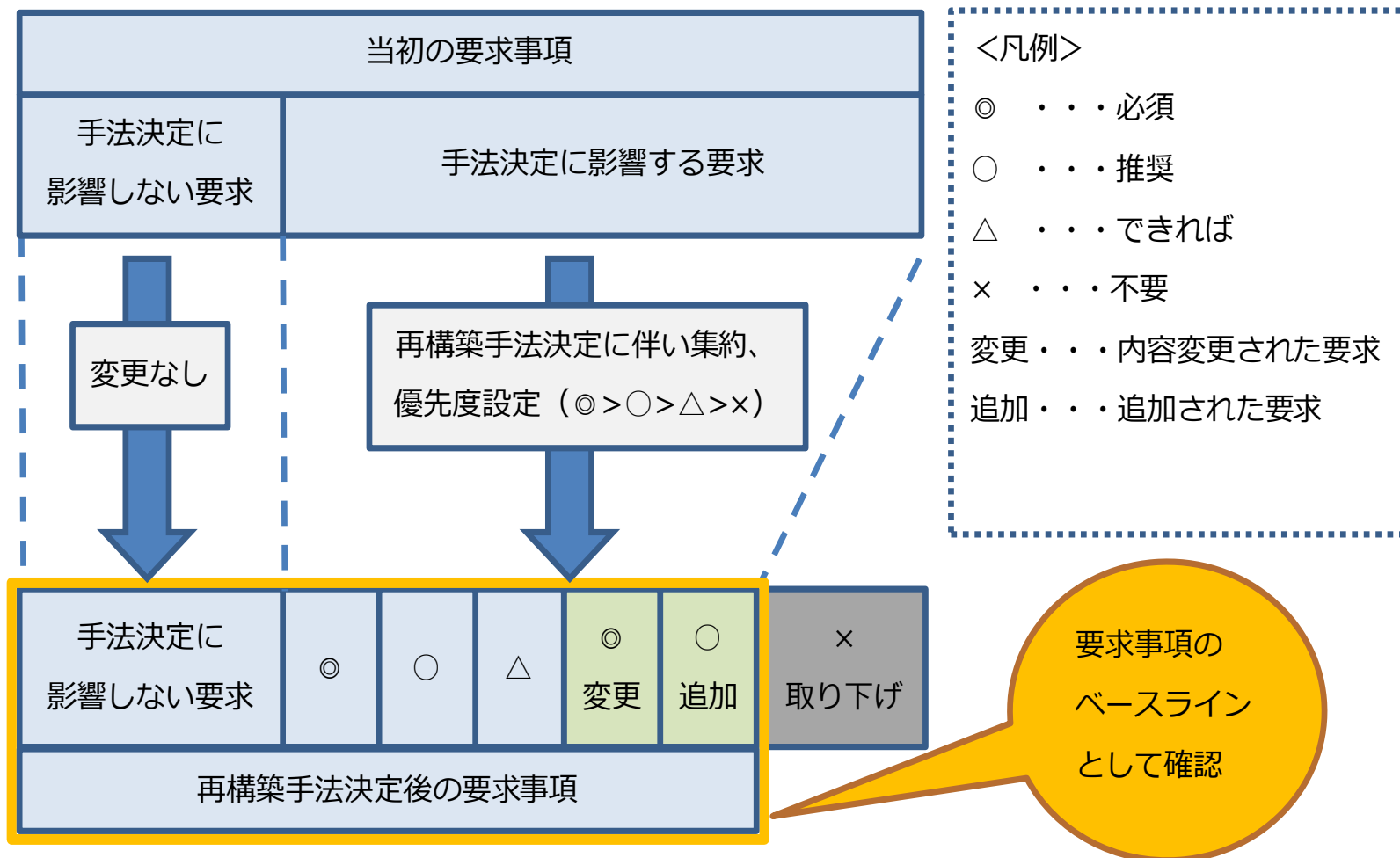
①標準に基づくシステム化計画の検討

②標準に基づいたシステム化計画に再構築のリスク対策を反映

リスクと対応策観点のマッピング例

再構築手法選択編 Step4 「再構築におけるリスク要因(ハザード)」		計画策定編 観点							
		A 要求の 確認	B 現行踏襲 内容の 明確化	C 現行資 産活用 方針の 検討	D 現行業 務知識 不足へ の対応	E 品質保 証の検 討	F 意思決 定プロ セスの 策定	G データ 移行の 計画	H 再構築 の計画 と見積り
リスク観点	再構築におけるリスク要因 (ハザード)								
現行踏襲 (機能)	・現行から受け継ぐべき業務運用や画面・帳票、データ仕様、外部インターフェースなどの対象や範囲は明確か？		○			○			
現行踏襲 (非機能)	・移行先のシステムの運用方法や性能、操作性、可用性、信頼性などの非機能要件は明確か？		○			○			
再構築サービスの 選定	・運用するサービスやソリューションは、現行と同じ機能、運用をすべて実現できるか？(特にリホストの場合)		○			○			
	・役割分担や制限、制約事項は明確になっているか？						○		
アーキテクチャ	・最新技術の適用や初めてのアーキテクチャ採用時には、十分な適用検討と技術検証を行っているか？		○			○			
	・アーキテクチャが混在することはないか？(ex新旧混在)		○			○			
システムの規模	・現行システムの肥大化・複雑化していないか？				○				
業務知識不足 (要員)	・現行業務有識者は、後継者不足になっていないか？(現行システム仕様は掌握できるか。)				○				
	・再構築時の要員は確保可能か？				○				
業務知識不足 (ドキュメント)	・業務ドキュメントとアプリケーションは一致しているか？		○	○	○				
	・ブラックボックス化により影響調査の精度が低下していないか？				○				
資産管理	・現行稼動中資産と移行資産のアンマッチはないか？(ex.実行モジュールとソースコードのアンマッチ)		○	○	○				
	・未検証資産の移行(開発中資産)はないか？			○					
業務データ移行	・データ仕様の把握不足によるデータ変換の誤り(フォーマット、文字コード)や異常データの混入は考えられないか？							○	

再構築手法選択で抽出した新システムへの要求を、システム 化計画立案にあたり再度整理し、ユーザ企業内で合意



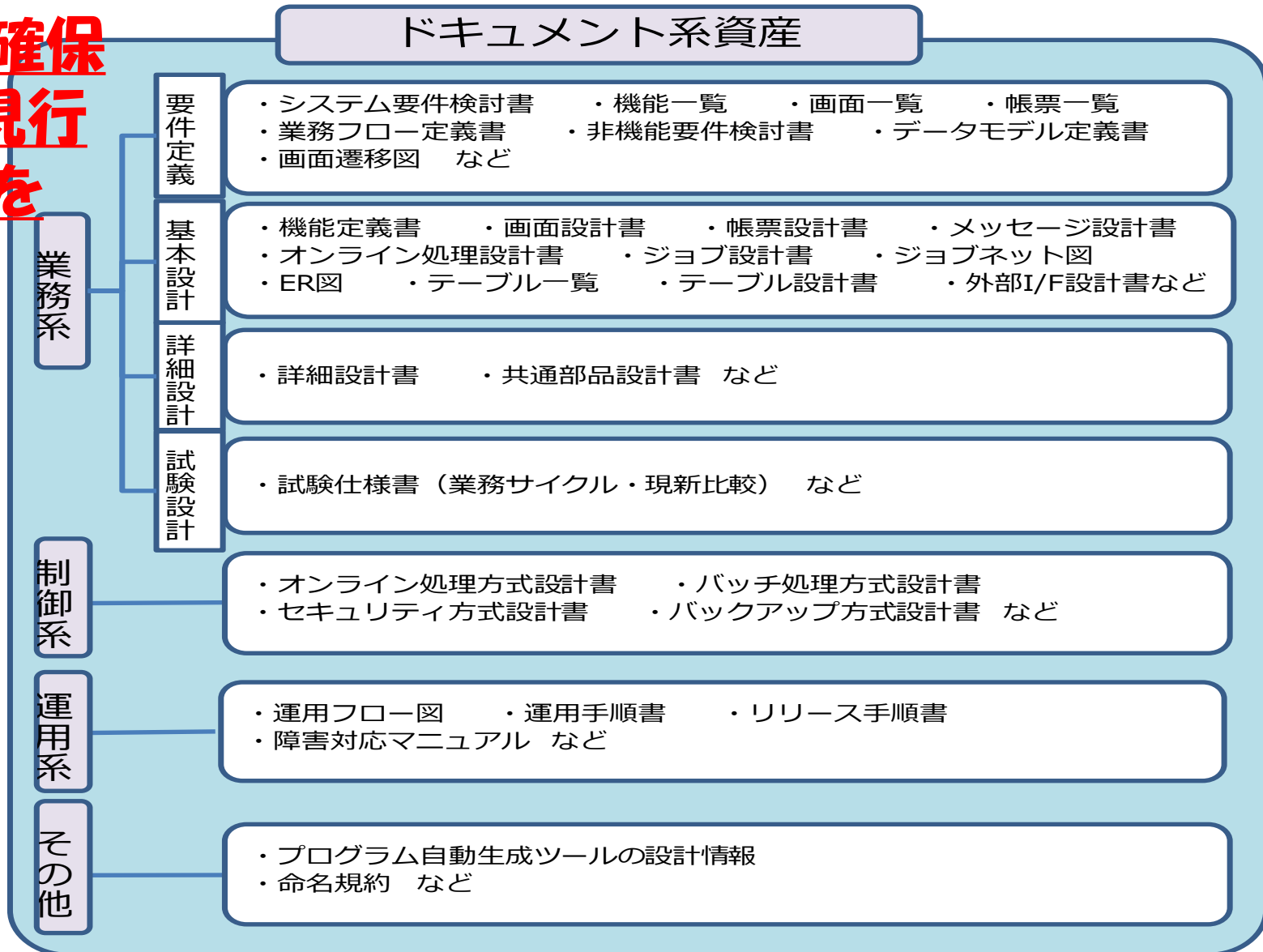
ユーザ企業とベンダ企業の「現行踏襲」のギャップを踏まえた上で、段階的に現行踏襲内容を明確化し曖昧さを排除

- ・ユーザ企業やベンダ企業といった各ステークホルダと調整・合意した内容を明確化
- ・機能要件のみではなく、必ず非機能要件も対象に
- ・本取組みにより、設計書やソースコード、システム間の情報の乖離を是正(適切な「現行」を明確化)

- ・現行踏襲内容は設計ドキュメントに仕様として明記
- ・レビュー・承認時には責任を持って実施
- ・利用部門との認識齟齬軽減のために、現行踏襲内容が変更になる際の利用部門へのレビュー実施や、サンプルプログラム(プロトタイプ)の早期提供による使用感の確認実施などの取組みをシステム化計画に組み込む

既に品質が確保 されている現行 資産の活用を 検討

ドキュメント系 資産の対象例



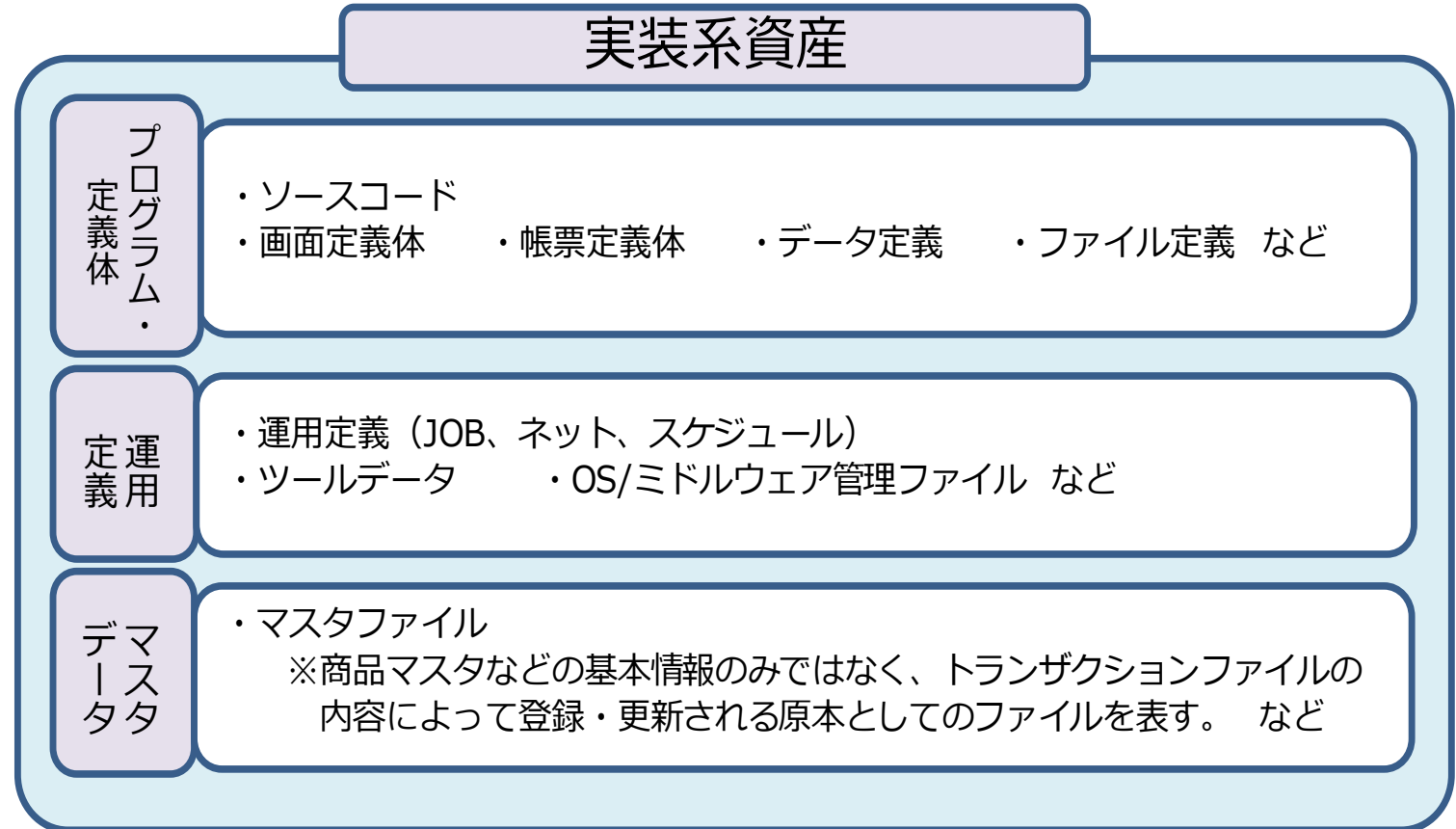
観点C. 現行資産活用方針の検討(2/3)

凡例：現行流用 再作成

業務系 要件定義 基本設計 詳細設計 制御方式 運用系	ハードウェア更改	リホスト	リライト	リビルド
「業務系」				
業務要件・業務仕様とともに変更なしの為、全体的に活用範囲となる。		ハードウェア更改とほぼ同じ。	リホストと同様だが、言語変更に伴うプログラム構成単位の変更などで基本設計レベルの変更が想定される。	業務要件以外は全て再作成となる為、活用範囲は要件定義ドキュメントのみ。
「制御方式」				
ハードウェア変更に伴い変更される場合もある。		各手法に共通して、アーキテクチャが変更される為、制御方式は検討しなおす必要があり、活用範囲はなし。		
「運用系」				
運用はほぼ変更なしの為、活用範囲が広い。		各手法に共通して、アーキテクチャが変更される為、運用ドキュメントは再作成となり活用範囲はなし。		

ドキュメント系資産の活用範囲の例

実装系資産の 対象例

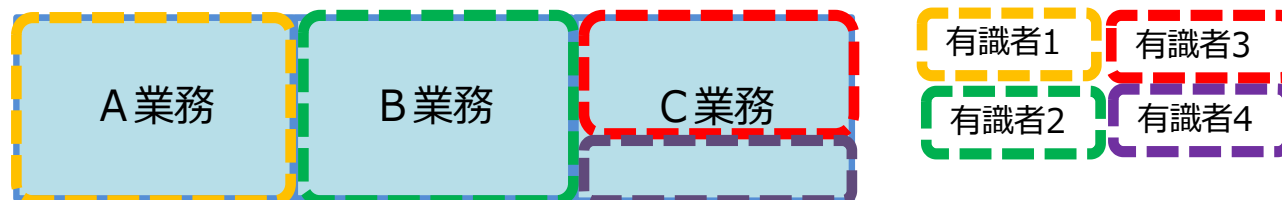


現行業務知識不足に備え、企画工程で、「不足部分の整備計画」を策定し、「整備不可の部分については意思決定プロセスで要件判断を行う方針を決定」しておく

現行業務不足により引き起こされる問題例

要件定義・設計	試験
<ul style="list-style-type: none">要件漏れ現行踏襲内容の明確化不可	<ul style="list-style-type: none">試験シナリオ作成不可試験網羅性検証不可故障発生時の本来あるべき姿の提示が不可要件面からの修正方針検討不可

有識者のカバー範囲のイメージ

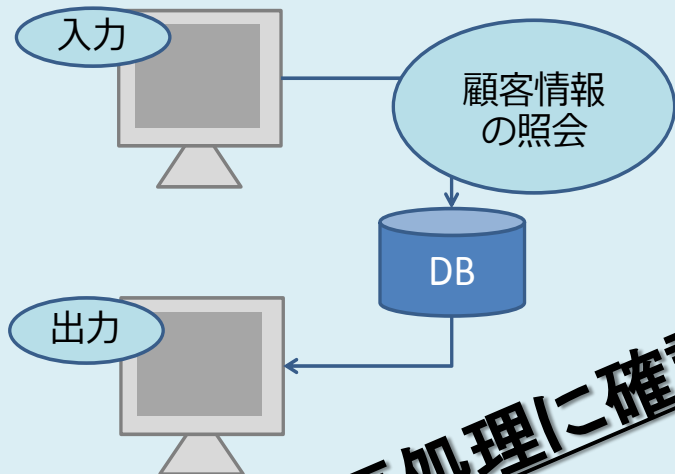


**有識者不足への対応
有識者の質は要確認**

「業務継続性の担保」を実現するための、 再構築のプロセス全体を通じた品質保証の取組み

例えば、保険業界のシステムの場合

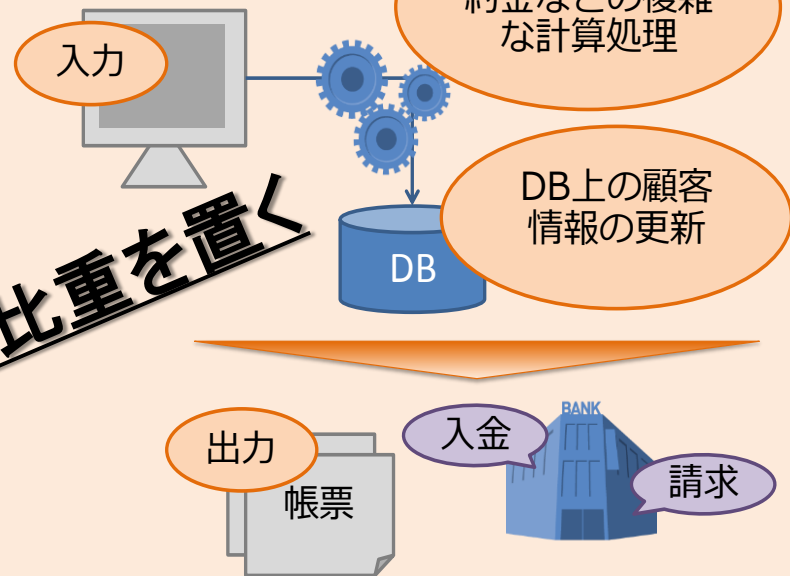
契約内容照会業務



重要処理に確認の比重を置く

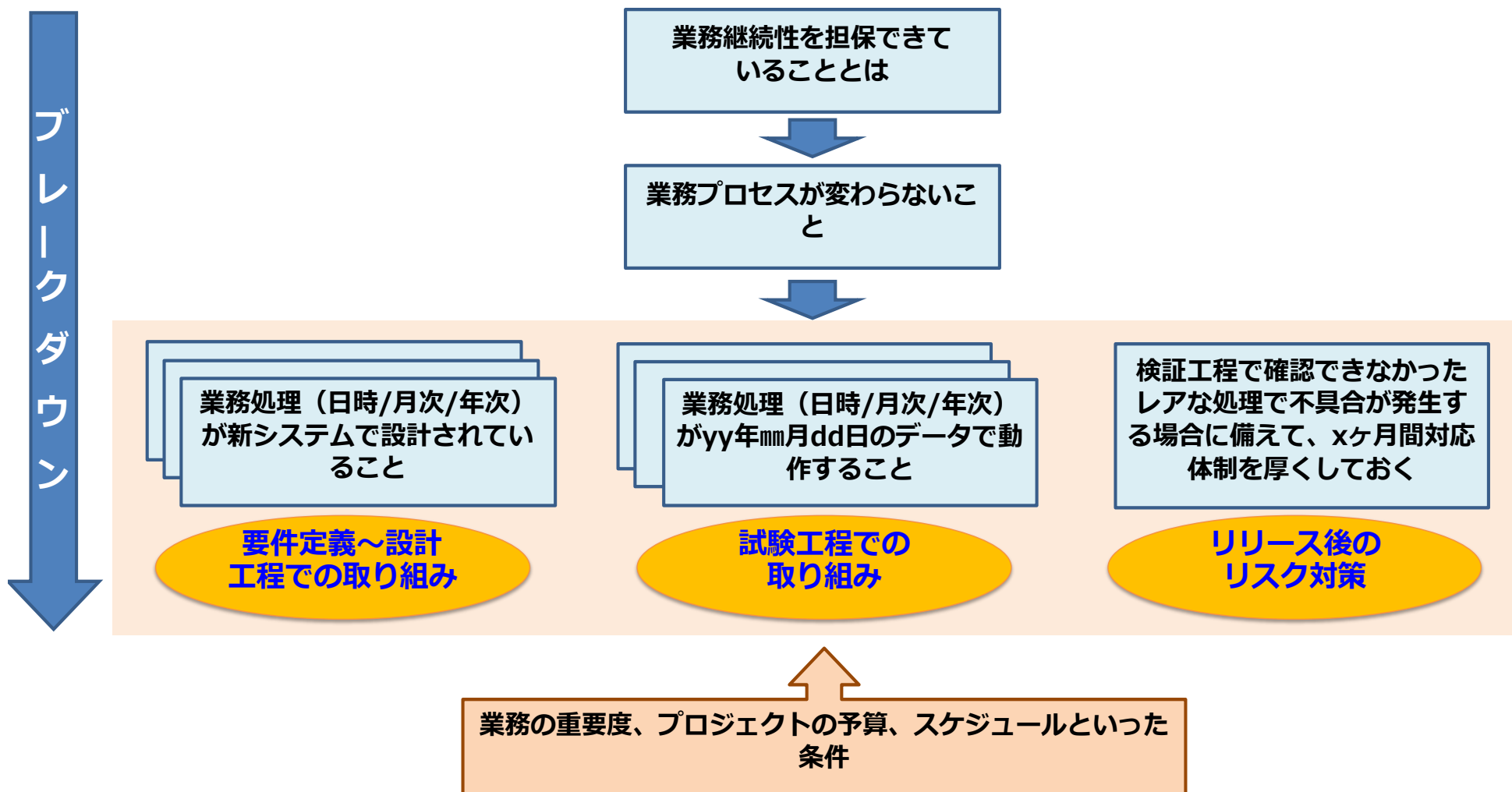
難しい処理が含まれないため、
入出力の確認レベルでよい

解約業務

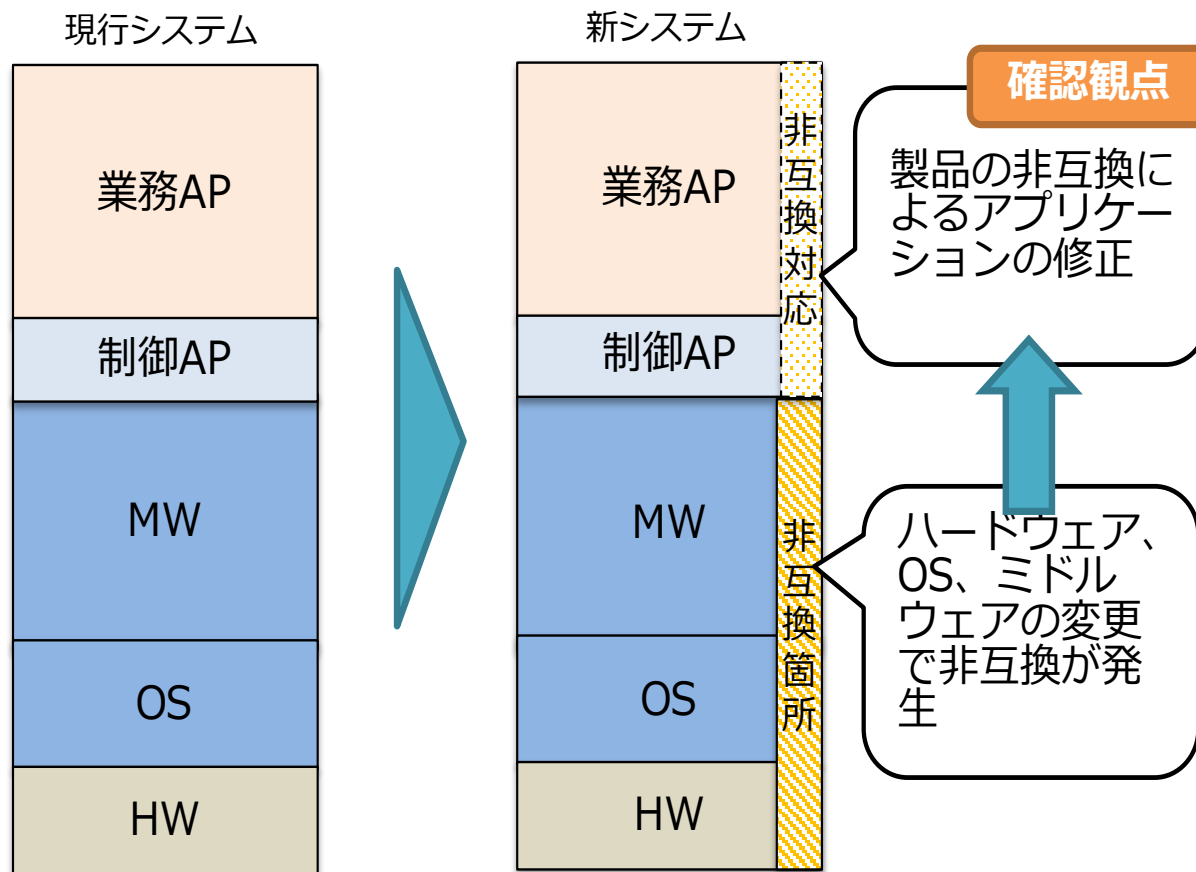


複雑かつ重要な処理が含まれている
ため、確実に確認する必要がある

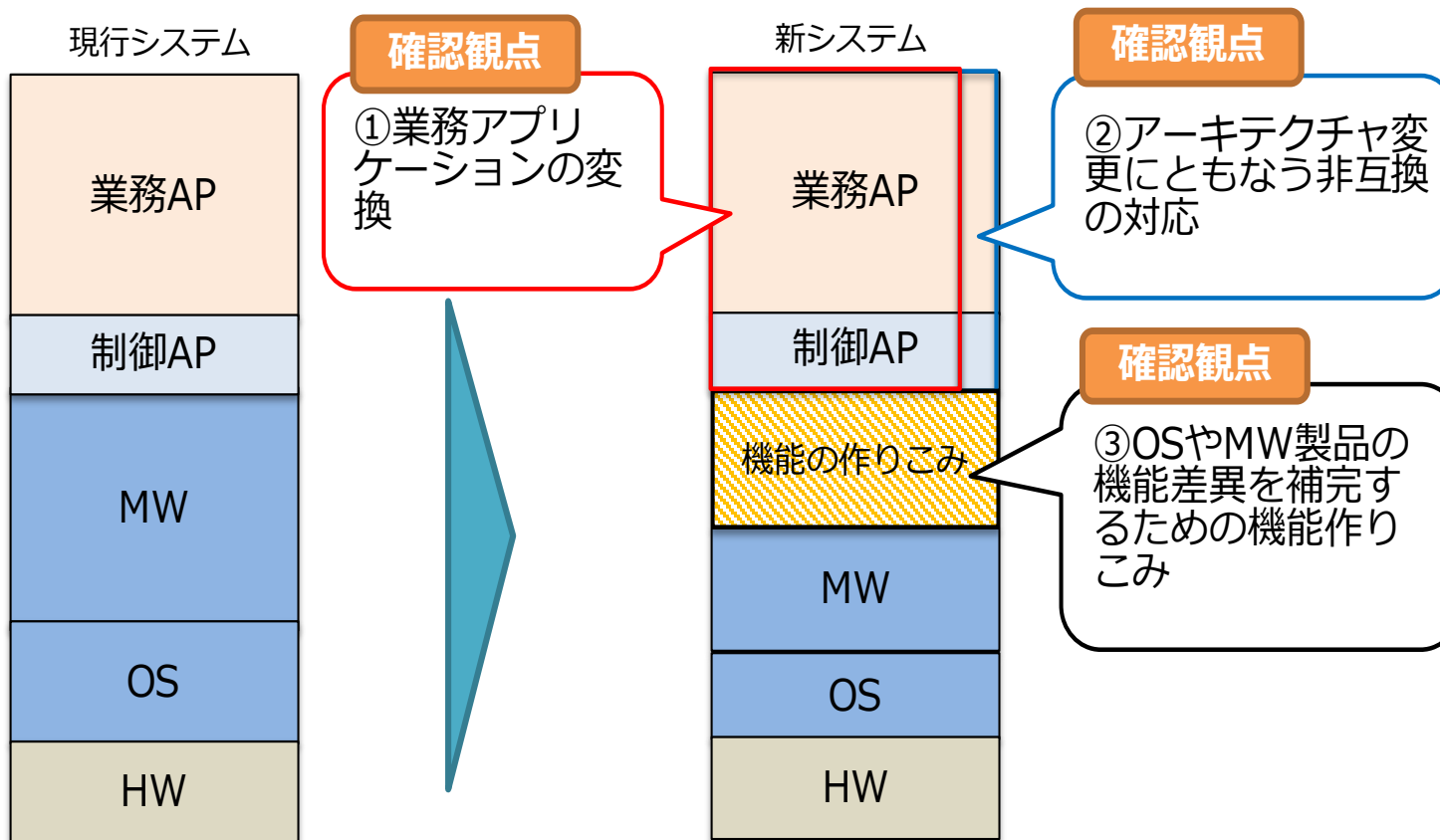
業務継続性担保の具体化



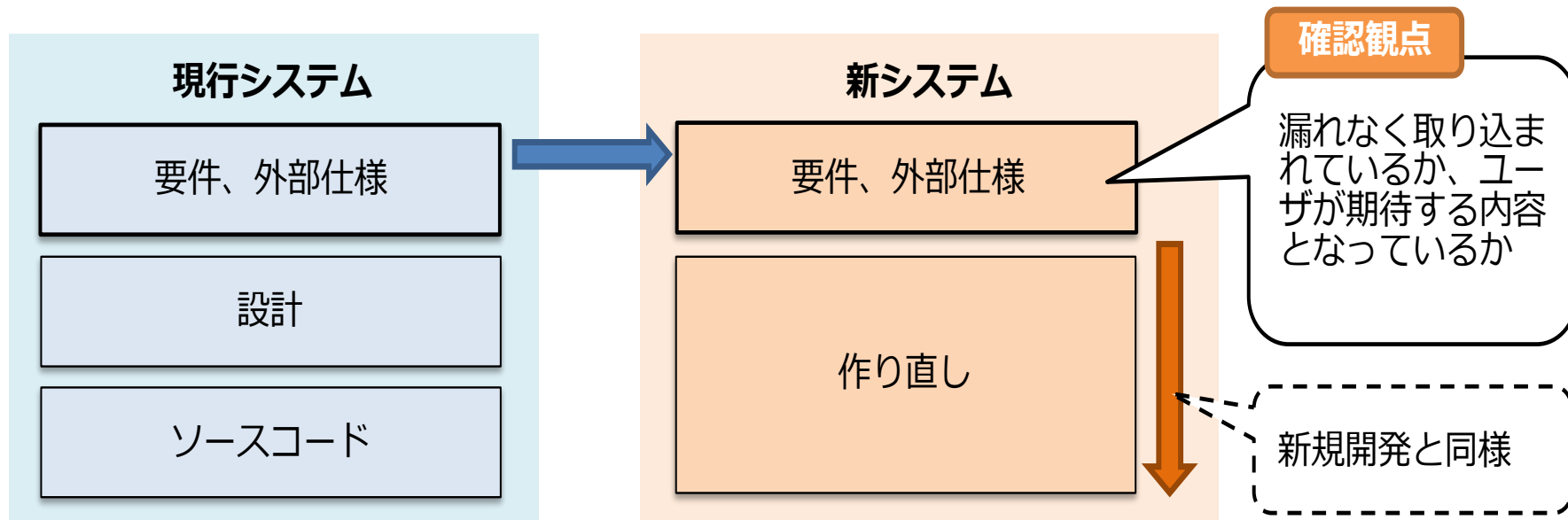
要件定義～設計工程では、現行踏襲内容を漏れなく、正しく検討し、それを設計に取り込むこと、および取り込まれていることを確認



ハードウェア更改の確認観点の例

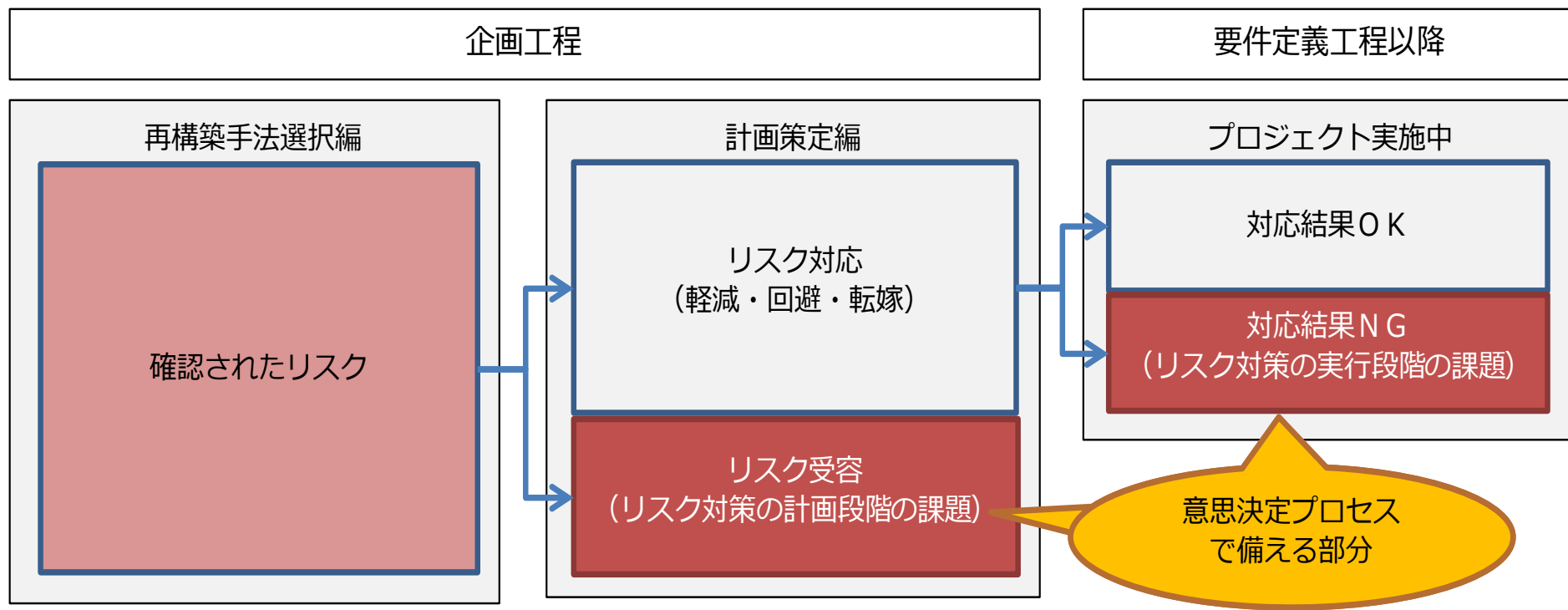


リホスト, リライトの確認観点の例



リビルドの確認観点の例

リスクを受容するにあたり方針の決定が必要な場合や、
計画したリスク対策をプロジェクトで実施した際に対応結果が
想定外となし課題化する場合に備え、
意思決定を行うプロセスを策定



検討すべき確認事項を明確にし、 開発と並行して実施する計画を早期に立てる

確認すべき観点

- 移行対象の整理
- レイアウトの確認
- データクレンジングの実施
- 文字コードのマッピング
- 移行方法の確認
- 移行実施結果の確認

ポイント

- データ移行で必要な確認観点の項目を早期に抽出する
- 問題が発生した際の解消期間も計画に盛り込む
- 試験工程までにデータ移行が完了している

観点A～G で述べた、
再構築で起こり得るリスクと、その対策としてのタスクを
見積りに盛り込む

例題(ケーススタディ)

1. 現行踏襲内容の明確化

- (1) 企画段階での明確化
- (2) 要件定義段階での明確化の計画
- (3) 追加コスト・時間が必要な部分への対応判断

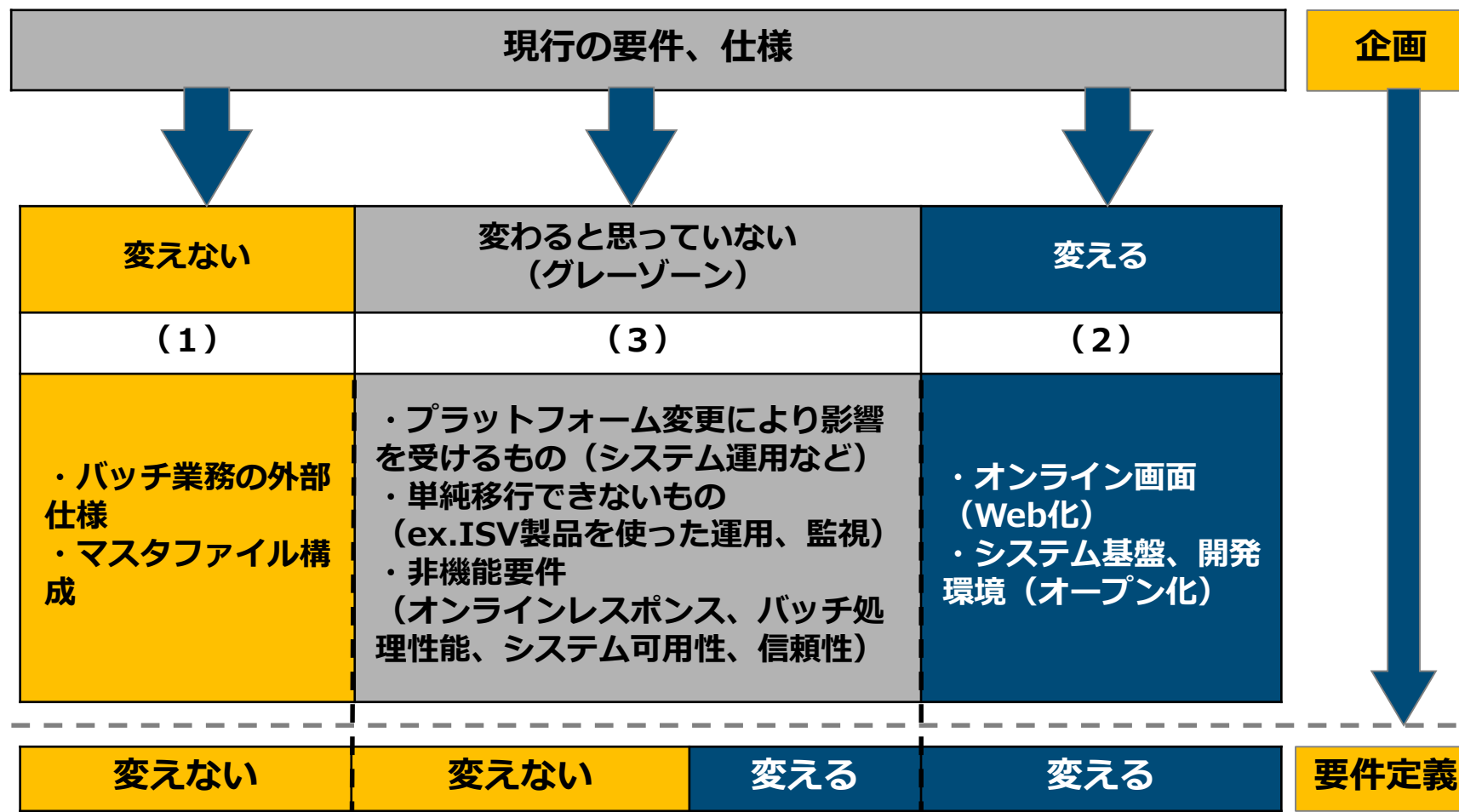
2. 品質保証(の考え方)の検討

- (1) 業務継続性担保の具体化
 - ・上記を検証する際の品質目標(品質確保方針)の具体化
- (2) 実施方針の検討
 - ・作り込み(設計)工程での実施方針
 - ・検証(各テスト)工程での実施方針
 - ー完了基準の明確化, 最終的なユーザ受入テスト(と早期参画)

3. システム化計画(まとめ)

- ・工程計画
- ・ユーザとベンダの役割分担例

企画段階での現行踏襲内容の明確化イメージ



再構築手法決定段階での現行システムの調査・分析, 新システムへの要求およびリスク対応方針の結果と「現行踏襲」の拠り所のイメージ図のギャップについてのステークホルダ間の認識合わせの結果から, 企画段階での再構築方針を定める。

リホストの要件定義段階の計画例

【リホスト】	役割分担		実施内容	入力情報	出力情報
	ユーザ	ベンダ			
	○		現行資産の提供 ・各種プログラム資産 ・JCL ・画面定義体 ・帳票定義体 ・オーバーレイ ・ADL	稼働資産一覧	現行資産
		○	・処理方式分析(画面、帳票) ・言語非互換分析 ・ISV製品分析	現行資産	非互換一覧
		○	移行対象資産の過不足調査	現行資産	資産一覧 不足資産一覧
	○		不足資産の提供	不足資産一覧	現行資産
	○	○	・リホスト対象資産の確定 ・リホスト非対象資産の確定 ⇒対応方針確定	資産一覧	移行対象資産一覧
	○	○	・非互換部分の移行方法検討 ・資産内容、変換方式に関するQ/A対応、ISV製品の情報提供	Q/A票 打合資料(移行方法案 など)	移行方法検討結果 Q/A票
		○	・Shell方式設計 ・COBOL方式設計 ・非互換対応方式設計	分析結果一覧	移行方法設計書
	○	○	実行時非互換の事前説明 レビュー及び承認	移行方針仕様書 移行方法設計書	レビュー記録
					移行方針仕様書 移行方法設計書

リビルドの要件定義段階の計画例

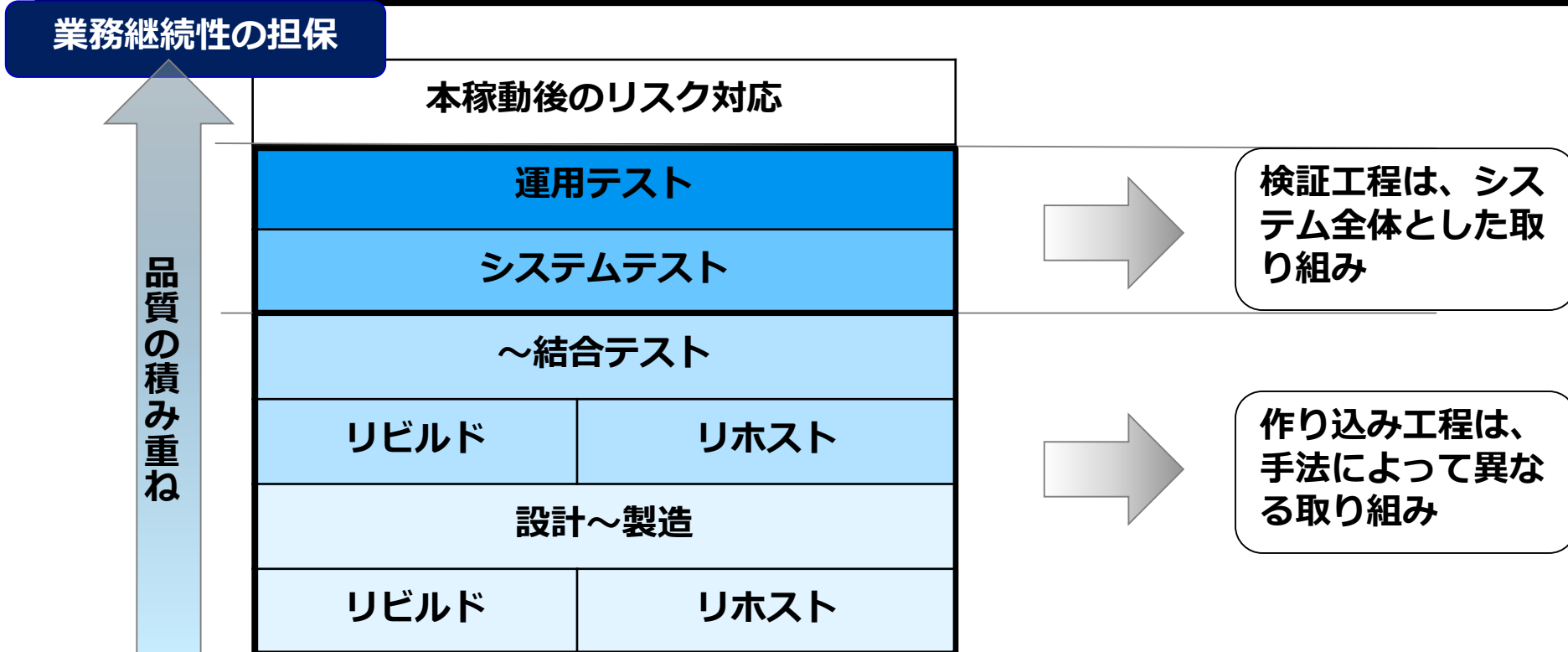
【リビルド】	実施者		実施内容	入力情報	出力情報
	ユーザ	ベンダ			
	○		<ul style="list-style-type: none"> ・現行オンラインの対象調査 ・画面遷移調査 ・オンライン機能調査 	現行資産	システム化業務フロー 画面レイアウト図 画面遷移図
	○	○	<ul style="list-style-type: none"> ・UNIXの基礎 ・Java/Servlet、SQL、shell基礎 	基礎研修教材	受講報告
	○	○	<ul style="list-style-type: none"> ・Javaフレームワークの研修 ・オンライン開発実践演習 	開発演習教材	受講報告 サンプル画面
	○		<ul style="list-style-type: none"> ・命名規約、画面設計規約の検討 ・オンライン処理方式の検討 	オンライン業務一覧 現行画面	設計規約、開発標準の検討資料
	○		<ul style="list-style-type: none"> ・レビュー 及び 承認 ・オープン環境研修の達成度評価 	設計規約、開発標準の検討資料 研修状況一覧	設計規約、開発標準研修評価資料
	○				

運用・システム基盤の要件定義段階の計画例

【運用】	実施者	実施内容	入力情報	出力情報	
					ユーザ
<p>現行運用情報の提供</p>	○	<p>現行システム運用資料の提供</p> <ul style="list-style-type: none"> 運用/保守マニュアル 保守作業内容 臨時作業内容 運用/保守補完ツール一覧 		現行システム運用資料	
<p>新運用案の検討</p>		<p>検討観点の抽出と実現案の作成</p>	現行システム運用資料	課題一覧 実現方式案	
<p>運用方式の検討</p>	○	<p>実現方式の検討</p> <ul style="list-style-type: none"> ミドルウェア代替 OS機能代替 新規開発 	課題一覧 実現方式案	議事録 検討結果	
<p>運用方式設計</p>		<p>運用方式の設計</p>	課題一覧 実現方式案 検討結果	運用方式設計書	
<p>レビュー 及び 承認</p>	○	<p>レビュー 及び 承認</p>	運用方式設計書	レビュー記録	
<p>判定</p> <p>ドキュメント修正</p>	○				
<p>運用要件定義確定</p>				運用方式設計書	
【基盤】		○	<p>非機能要件の定義(可用性、信頼性、性能・拡張性、運用・保守性、セキュリティ)</p>	現行システム運用資料 実現方式案	非機能要件定義書
	<p>非機能要件定義</p>		<p>インフラ要件の定義(システム構成、適用ミドルウェア、システム環境)</p>	現行システム運用資料 実現方式案	システム構成要件定義書
	<p>レビュー 及び 承認</p>	○	<p>レビュー 及び 承認</p>	基盤要件定義書	レビュー記録
	<p>判定</p> <p>ドキュメント修正</p>	○			
	<p>基盤要件定義確定</p>				基盤要件定義書

- ・バックアップ
- ・リカバリ
- ・スケジューラ
- ・媒体運用
- ・帳票出力/配信
- ・外部接続
- ・システム監視
- ・資産管理
- ・リリース管理
- ・ユーザ管理
- ・文字コード管理

業務継続性担保の考え方例_1



作り込みから検証までの各工程での品質検証の積み上げの結果、最終的な業務継続性を担保する考え方をステークホルダ間で合意形成し、各工程での取組み、検証計画を策定

- ・設計～結合テストまでは、手法の違いに着目し、業務継続性を担保するための品質の基礎としての実施方針(品質確保方針)を策定
- ・システムテスト以降は、再構築手法の違いによらずシステム全体として業務継続性を検証

設計～結合テストまでの取り組み例

区分		設計	開発	～結合テスト
リビルド	検証観点	「新規開発と同様」 (注)		<ul style="list-style-type: none"> 設計書に元づいた、画面遷移や画面への出力内容およびマスタファイルの登録内容等の妥当性検証(ホワイトボックステスト)
	検証内容			<ul style="list-style-type: none"> オンライン画面遷移 マスタファイルの更新結果 画面メッセージの内容 など
リホスト	検証観点	<ul style="list-style-type: none"> 非互換調査の充分性 移行方法、変換仕様、基盤部品の妥当性 	<ul style="list-style-type: none"> 変換後資産が、変換仕様どおりになっているかの検証 	<ul style="list-style-type: none"> 現行システムとの同値性 変換仕様パターンの網羅性
	検証内容	<ul style="list-style-type: none"> 設計レビュー及び処理方式と変更仕様パターンの妥当性確認を目的としたパイロット検証 	<ul style="list-style-type: none"> 変換後資産のレビュー(ツール変換はサンプル、手修正は全箇所) 	<ul style="list-style-type: none"> 現行システムとの処理結果の比較による、新システム側の処理結果の妥当性。(現新比較検証) マスタファイルの出力結果 帳票の出力結果

システムテスト以降の取り組み例

区分	システムテスト1	システムテスト2 (サイクルテスト)	運用テスト (並行稼動)
検証観点	<ul style="list-style-type: none"> 各業務毎のシナリオの処理結果が正当であること。 業務シナリオによる機能検証（正常系/異常系） 各種運用・保守ツールの運用検証（リリース検証、障害通知検証など） 	<ul style="list-style-type: none"> 数日間の全主要業務が完了するまで。（YY.MM.DD～YY.MM.DD+n日） 日付を合わせた移行データで、処理結果の数値（売上等）が現行と同値であること。 繁忙期の性能検証、トラブル時の信頼性検証 	<ul style="list-style-type: none"> 現行と同じ運営運用を新システムで継続して維持できること。 日々の業務運営の妥当性、正当性を検証するため、システム運用を自動実行し、他システム連携を含む、各処理間の整合性、情報収集、提供タイミングを評価する。
検証内容	<ul style="list-style-type: none"> 各業務イベント（シナリオ検証） 全オンライン 全バッチジョブ 全帳票検証 全サーバ連携 	<ul style="list-style-type: none"> オンライン業務検証（コンペア検証） バッチ業務検証（ 〃 ） 数値検証（日次、月締め等でのデータ内容の妥当性とデータ間の整合性） 主要帳票検証 内部サーバ連携検証 非機能要件（冗長化、性能、バックアップ、リカバリ、システム監視） 	<ul style="list-style-type: none"> 本番アクセス権での通常運営、異常時運用検証 外部データ連携検証 性能検証（レスポンス、夜間バッチ） データ移行検証 現場キーパーソン教育 マニュアル検証
実施主体	ベンタ	ユーザ	ユーザ

テスト完了の判断基準の例

判定項目	オン	バッチ	完了基準	判定根拠等
業務機能	△	○	<ul style="list-style-type: none"> ・オンライン:全画面の検証が完了している。 ・バッチ:全バッチ処理の検証が完了している。 ・日常の業務オペレーション(月末、期初イベント含む)ができる。 	
運用管理			<ul style="list-style-type: none"> ・トラブル時のリカバリ対応ができる。(切替えが目標時間内にできる) ・各種障害を検知し、想定とおりに通知される。 	
性能			<ul style="list-style-type: none"> ・オンライン:目標レスポンスで応答している。 ・バッチ:目標時間(現行より早く)、バッチ処理が完了している。 	
データ移行			<ul style="list-style-type: none"> ・オンライン:全テーブル移行できている。 ・バッチ:過去分含めて全データ移行できている。過去分はデータ検証が完了している。 	
習熟度			<ul style="list-style-type: none"> ・基本的な操作ができる。 ・現行のレベル(時間/品質)で通常対応、トラブル対応ができる。 	

<凡例> ○ : 完了している、△ : 一部未完了 (完了見込みあり、不安なし)、× : 完了していない (不安がある)

次の構築段階での工程計画(マスタスケジュール)や体制, ユーザ企業とベンダ企業の役割分担などの立案

	要件定義	設計	開発	～結合 テスト	システムテスト～	リリース後	
リビルド	現行オンライン 調査 【ユーザ】 オープン 環境教育 【共同】	新規と同様な設計・開発・テスト 【ユーザ】					
運用 基盤	現行運用調査 【共同】	システム運用・基盤の再設計・構築 【ベンダ】			システム テスト 【共同】	サイクル テスト & 運用検証 【ユーザ】	本稼動後 対応 【ユーザ】
リホスト	非互換調査 【ベンダ】	移行方法検討 変換仕様設計 パイロット検証 【ベンダ】	資産変換& 現新比較テスト 【ベンダ】				

事例一覧（第4章）

再構築手法選択編の事例

- 4.1（株式会社 NTTデータ）
- 4.2（富士通株式会社）
- 4.3（株式会社 日立製作所）
- 4.4（日本電気株式会社）

計画策定編の事例

- 4.5「現行踏襲内容の明確化(観点B)」(富士通株式会社)
- 4.6「現行資産活用方針の検討(観点C)」(日本電気株式会社)
- 4.7「現行業務知識不足への対応(観点D)」(株式会社 ソフトロード)
- 4.8「品質保証の検討(観点E)」(株式会社 NTTデータ)
- 4.9「意思決定プロセスの策定(観点F)」(東京海上日動システムズ株式会社)
- 4.10「再構築の計画と見積り(観点H)」(株式会社 日立製作所)
- 4.11「再構築の計画と見積り(観点H)」(JFEシステムズ株式会社)

再構築プロジェクトを通じて、新システムでは、業務知識不足や、ドキュメントやソースコード・動作しているシステムの間での乖離の解消が行われると考えられる。

しかし、新システムでも再び「システム再構築をとりまく現状」で述べたようなレガシー化が進むおそれがある。

本ガイドの対象は再構築プロジェクトの企画段階で検討すべき内容であることから説明はしていないが、

新システム稼動後の再レガシー化を防止するような取組みを検討することも重要。

1. IPA/SECにおけるシステム構築上流工程強化の活動
 - 背景と課題
 - 取組み概要
2. ユーザのための要件定義ガイド
3. システム再構築を成功に導くユーザガイド
4. まとめ

開発プロジェクトの失敗やサービス開始後のシステム障害発生を低減

ユーザのための要件定義ガイド

～要求を明確にするための勘どころ～

システム再構築におけるリスクをステークホルダ間で合意し、対策

システム再構築を成功に導くユーザガイド

～ユーザとベンダで共有する再構築のリスクと対策～

2017年1月31日 pdf版公開

2017年3月10日 書籍発行予定

第10回要求シンポジウム

要求定義の今と未来

～イノベーション／モダナイゼーション時代の要求定義のヒントを事例から探る～

<http://sec.ipa.go.jp/seminar/20170310.html>

共催	独立行政法人情報処理推進機構 (IPA) 株式会社NTTデータ 富士通株式会社
後援	一般社団法人 日本情報システム・ユーザー協会 (JUAS) 特定非営利活動法人ITコーディネータ協会 (ITCA)
開催日時	2017年3月10日(金)13:30～17:50
開催場所	NTT DATA 駒場研修センター 東京都目黒区駒場2-18-2
募集対象	要求定義に興味を持つ方(企業の業務部門／システム部門の方、 IT業界関係者の方、教育機関の方)

[IPAが上記の課題認識のもとに行った取組みの成果を紹介](#)

<プログラム>

13:30～13:35 開会のご挨拶 IPA/SEC 所長 松本 隆明

13:35～14:15 基調講演:

「[全社視点の要求マネジメント ～システム論的要求の扱い～](#)」

IPA/SEC システム構築上流工程強化部会 主査/国立大学法人 名古屋大学 教授 山本 修一郎 氏

14:15～14:35

「[ユーザのための要件定義ガイド ～要求を明確にするための勘どころ～](#)」のご紹介

IPA/SEC システム化要求WG 主査/一般社団法人アドバンスト・ビジネス創造協会 岩佐 洋司 氏

14:35～14:55

「[システム再構築を成功に導くユーザガイド ～ユーザとベンダで共有する再構築のリスクと対策～](#)」のご紹介

IPA/SEC モダナイゼーションWG 主査/株式会社NTTデータ 大山 宏 氏

14:55～15:25 モダナイゼーション事例:

「[再構築プロジェクトだからこそ知っておきたい、上流工程におけるコミュニケーションのツボ](#)」

IPA/SEC モダナイゼーションWG 委員/東京海上日動システムズ株式会社 内藤 克郎 氏

15:40～16:25 招待講演:

「[イノベーションの機会発掘とシステムケーパビリティ ～新たなアイデアがビジネスとして実装されるまで～](#)」

特定非営利活動法人 CeFIL デジタルビジネス・イノベーションセンター 副代表 兼 チーフディレクター

東京理科大学 客員教授/九州工業大学 客員教授 小西 一有 氏

16:25～17:05 イノベーション事例1:

「[Made by imagination ～様々なアイデアやデザインを実現できる環境作り～](#)」

シタテル株式会社 代表取締役 河野 秀和 氏

17:05～17:45 イノベーション事例2:

「[IoT時代に求められる要求定義](#)」

Qrio株式会社 営業企画部 マネージャー 佐藤 竜斗 氏

17:45～17:50 閉会のご挨拶 株式会社NTTデータ 生産技術部 技術部長 戸村 元久

ご清聴, ありがとうございます ございました

システム構築上流工程強化の取組み

http://www.ipa.go.jp/sec/system/sys_upper.html

ガイドブックのダウンロード

<http://www.ipa.go.jp/sec/reports/20170131.html>



参考



主査	山本 修一郎	国立大学法人名古屋大学
	岩佐 洋司	一般社団法人アドバンスト・ビジネス創造協会
	大山 宏	株式会社エヌ・ティ・ティ・データ
	小野 修一	株式会社エヌ・ティ・ティ・データ
	加藤 一郎	日本電気株式会社
	崎本 壮	株式会社日立製作所
	澁谷 裕以	株式会社日本取引所グループ
	西村 光司	一般社団法人日本情報システム・ユーザー協会
	細川 泰秀	一般社団法人アドバンスト・ビジネス創造協会
	森田 功	富士通株式会社

<http://www.ipa.go.jp/sec/about/committee.html#014>



主査	岩佐 洋司	一般社団法人アドバンスト・ビジネス創造協会
	太田 忠雄	株式会社ジャステック
	後藤 将夫	一般社団法人アドバンスト・ビジネス創造協会
	坂巻 雅貴	東京ガス株式会社
	崎山 直洋	株式会社エヌ・ティ・ティ・データ
	桜井 新	新日鉄住金ソリューションズ株式会社
	清水 淳史	セイコーエプソン株式会社
	高橋 康介	大日本印刷株式会社
	高橋 実雄	サントリーシステムテクノロジー株式会社
	中村 伸裕	住友電気工業株式会社
	藤本 礼久	全日本空輸株式会社
	森田 功	富士通株式会社

<http://www.ipa.go.jp/sec/about/committee.html#015>



主査	大山 宏	株式会社エヌ・ティ・ティ・データ
	御魚谷 かおる	富士通株式会社
	小林 茂憲	日本電気株式会社
	小林 豊	株式会社菱化システム
	崎本 壮	株式会社日立製作所
	千田 正一	富士通株式会社
	内藤 克郎	東京海上日動システムズ株式会社
	渡邊 崇	JFEシステムズ株式会社

<http://www.ipa.go.jp/sec/about/committee.html#016>

2016年の活動



システム構築上流工程強化
部会

全体の運営方針、扱うべき課題等の議論

Step1 目線合わせ 報告・共有 Step2 議論・検討 報告・共有 Step3 成果物作成/レビュー 報告・共有 Step4 まとめ・普及展開 報告・共有

システム化要求
WG



課題共有



課題検討



成果物作成



事例共有、問題・課題・解決策の議論後、
成果物のまとめ方、構成、内容を議論

モダナイゼーション
WG



課題共有



課題検討



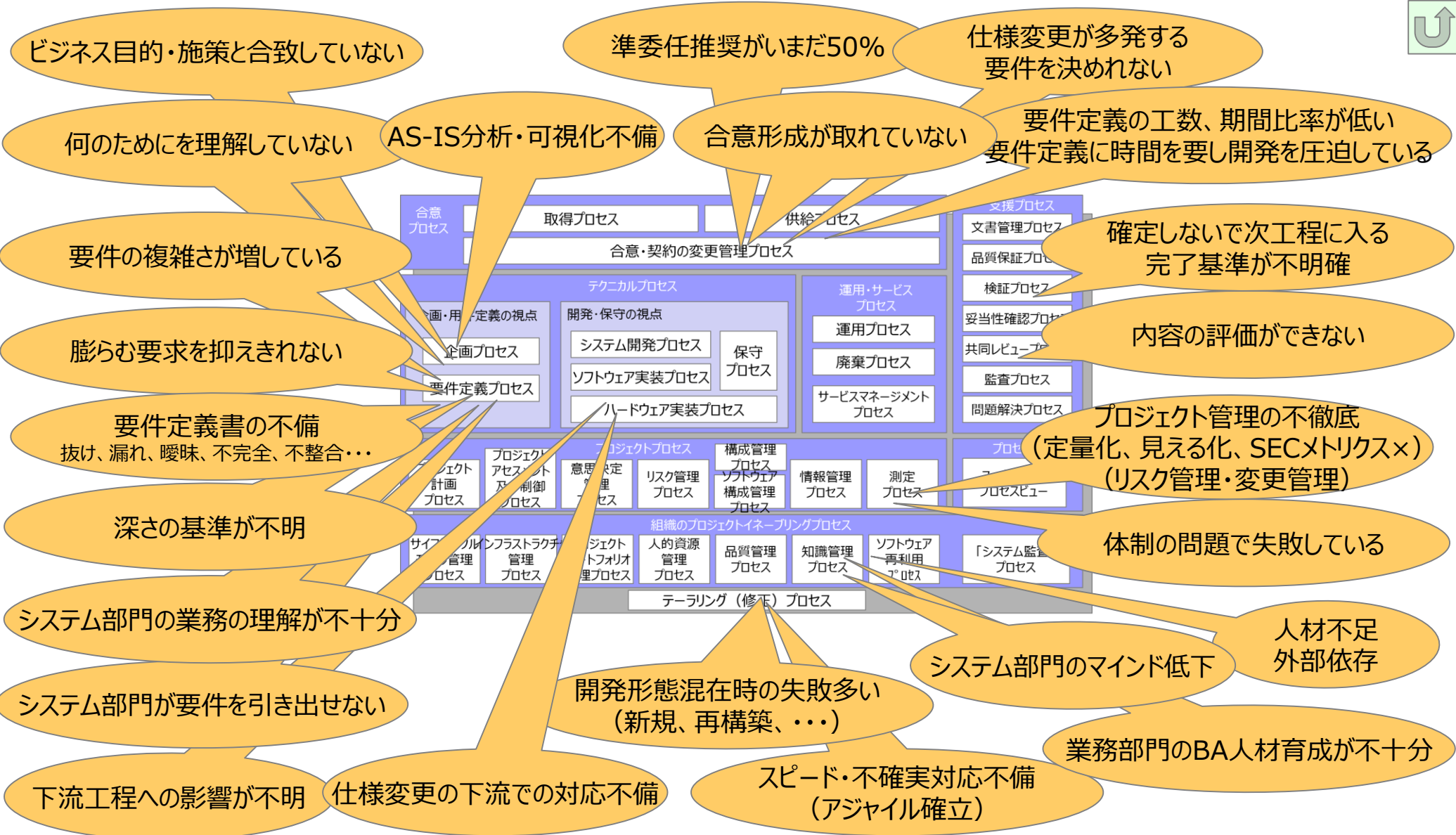
成果物作成



成果物公開

議論の経緯

～委員から出された課題(共通フレームに照らして)～



議論の経緯 ～課題の整理～



- 1.(留)要件定義の役割分担、特に業務部門の参画が不十分
- 2.(留)準委任推奨がいまだ50%
- 3.(留)要件定義の工数、期間比率が低い/要件定義に時間を要し開発を圧迫している
- 4.(留)確定しないで次工程に入る(完了基準が不明確)
- 5.□(留)仕様変更が多発する、要件を決めれない(合意プロセス、要件定義プロセス)
- 6.□(留)内容の評価ができない。
- 7.□ビジネス目的・施策と合致していない
- 8.□手段が先行し、「何のために」が理解できていない。
- 9.□業務の複雑さが増している
- 10.□合意形成が取れていない
- 11.□膨らむ要求を抑えきれない
- 12.□要件定義書の不備(抜け、漏れ、曖昧、不完全、不整合・・・)
- 13.□非機能要件を決めきれない、不備
- 14.□深さの基準が不明
- 15.□AS-IS分析、可視化不備
- 16.□システム部門の業務の理解が不十分
- 17.□システム部門が要件を引き出せない
- 18.■プロジェクト管理の不徹底(定量化、見える化、SECメトリクス)(リスク管理・変更管理)
- 19.■体制の不備での失敗
- 20.■業務部門のBA人材育成が不十分
- 21.■仕様変更の下流での対応不備、下流工程への影響が不明(トレーサビリティ)

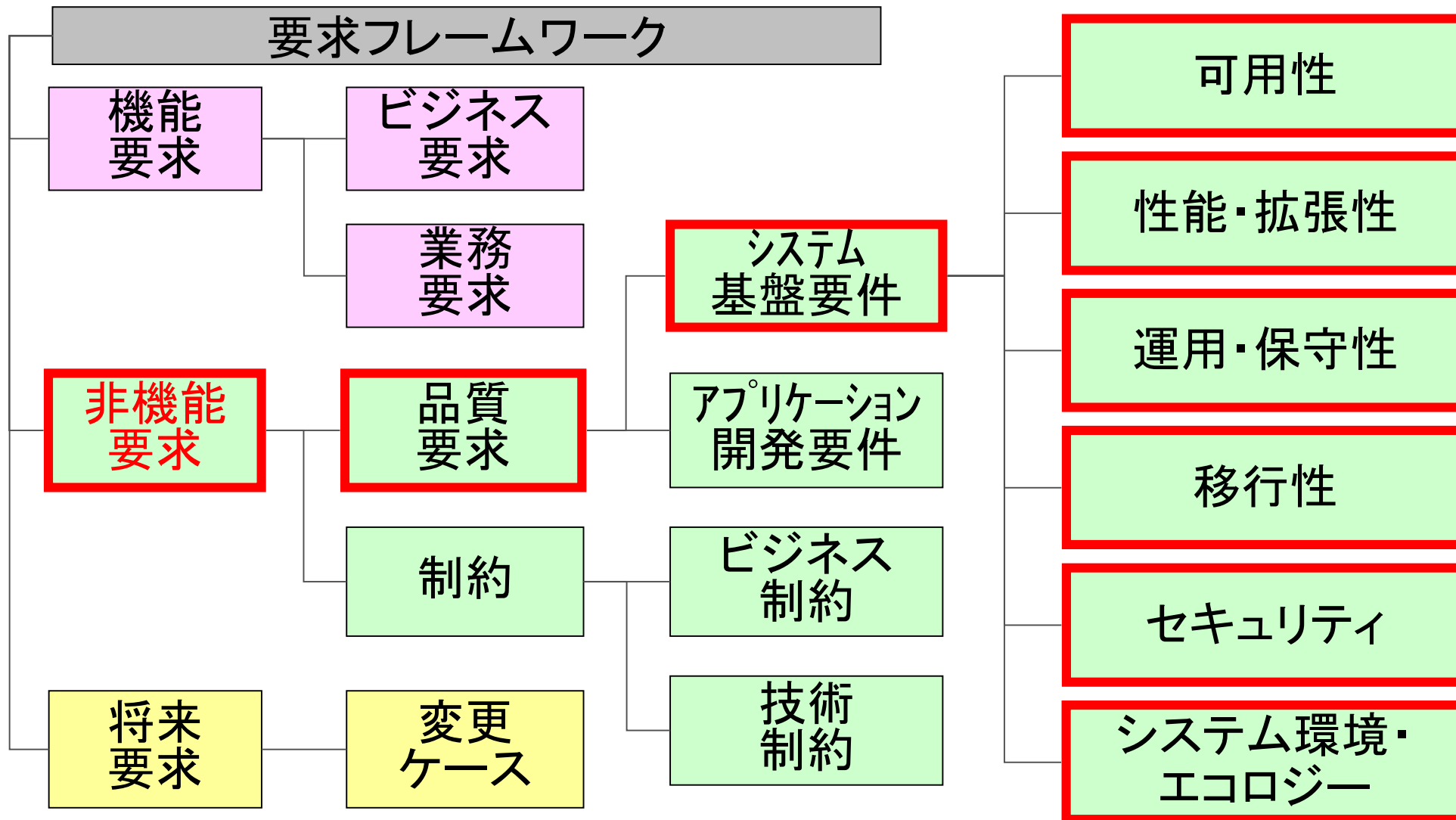
凡例:

(留)留意事項として言う

□勘どころを示唆する

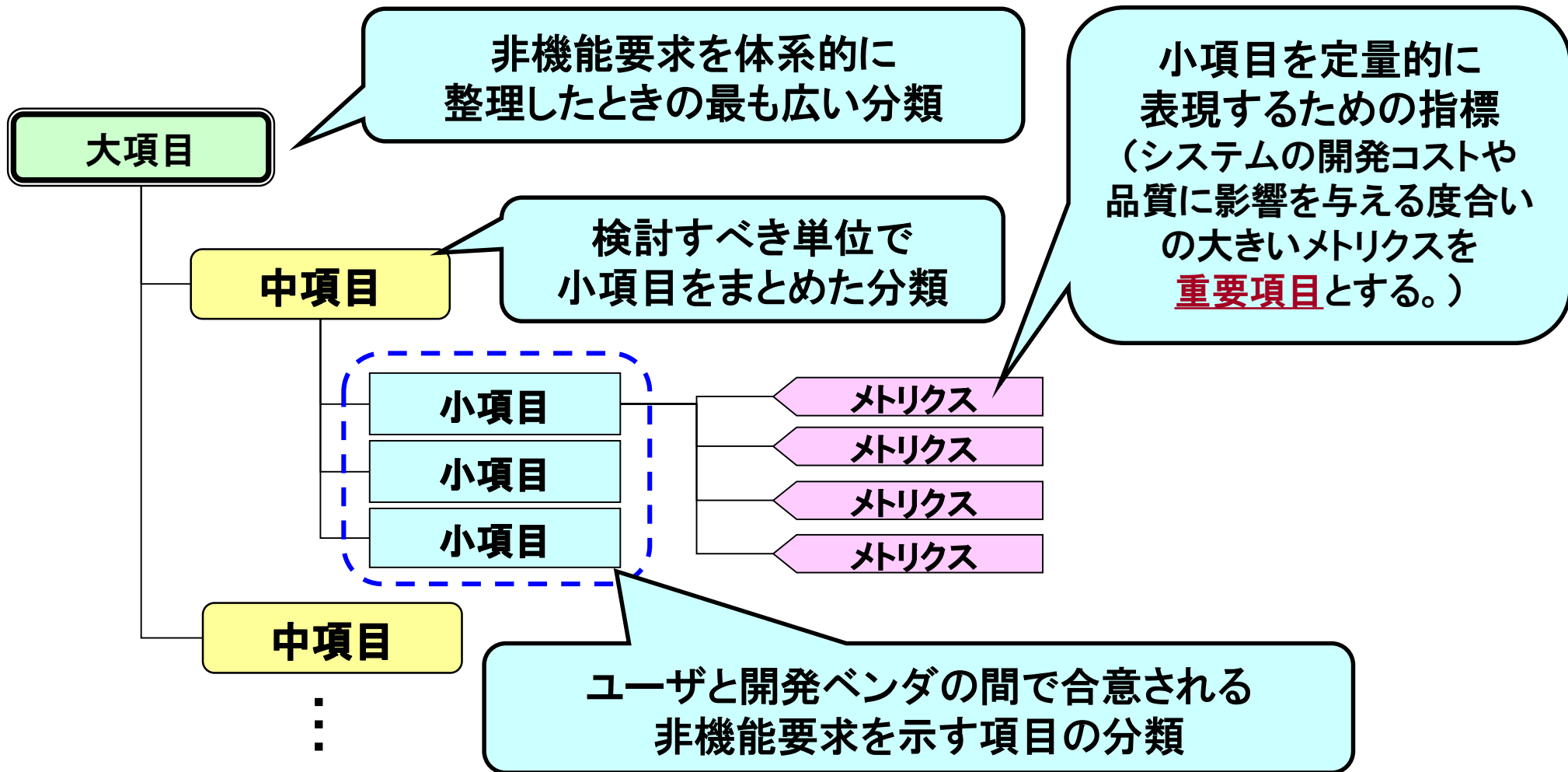
■今回対象外候補

情報システムに対する要求には、「機能要求」と「非機能要求」がある



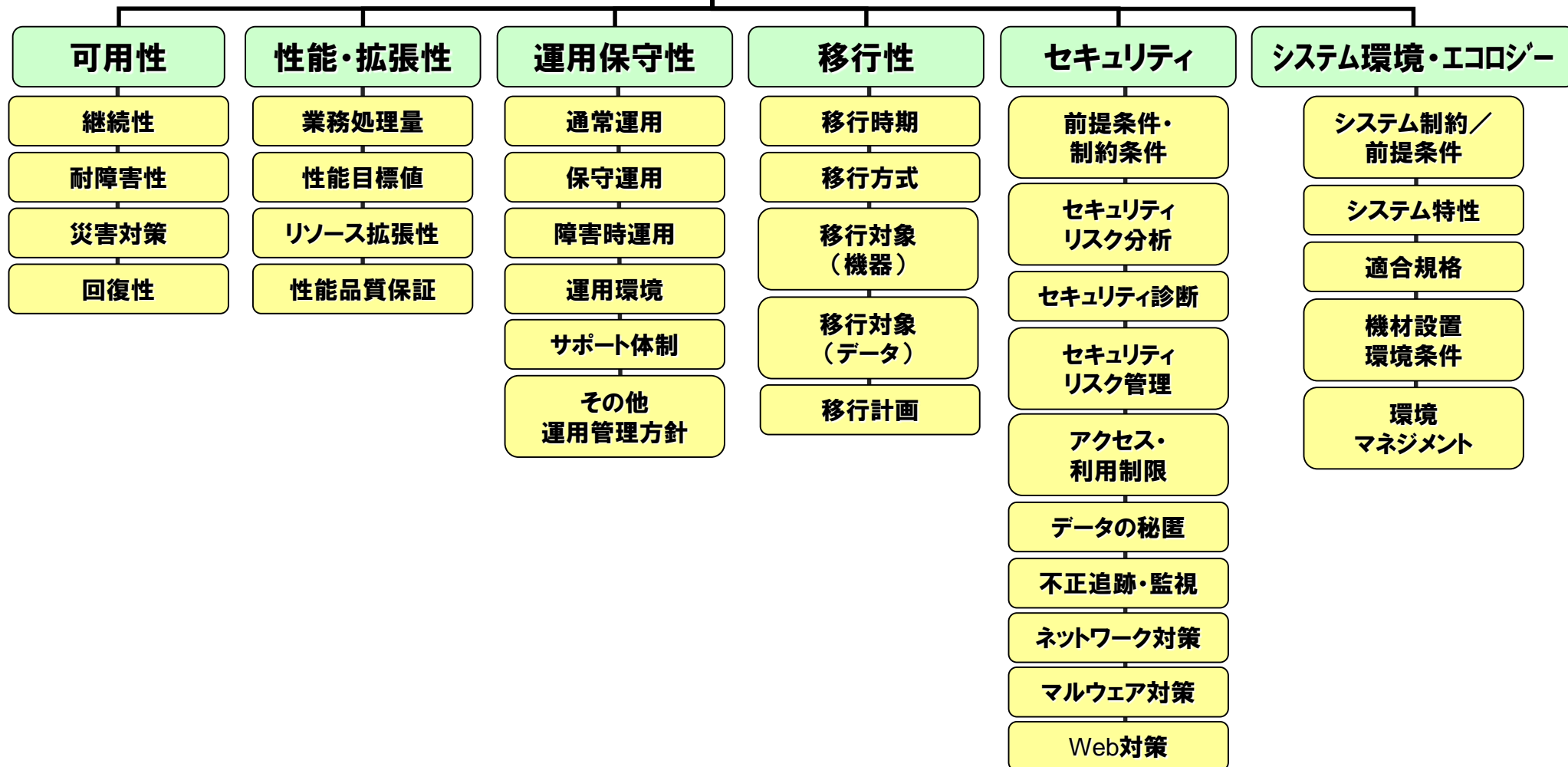
<http://www.ipa.go.jp/sec/softwareengineering/std/ent03-b.html>

■ 非機能要求項目とメトリクスの関係



■ 非機能要求を6大項目, 34中項目, 116小項目, 236メトリクスに体系化

非機能要求項目



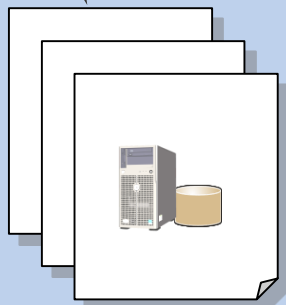


■ ユーザ視点から要求の実現レベルを見える化するため、段階的に詳細化しながら、「早期に」「誤解なく」「漏らさずに」確認できるようにする

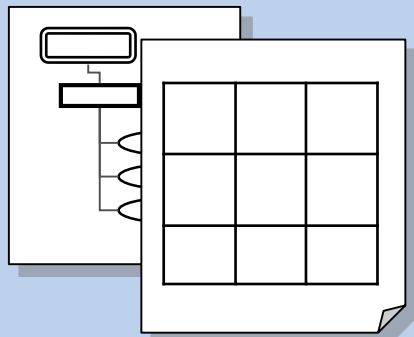
方向性の確認

コストに大きな影響がある
要求項目を決定

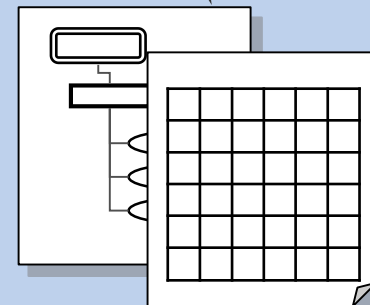
開発開始までに必要な
要求項目を決定



モデルシステムシート
(グレード表)



グレード表・樹系図



項目一覧・樹系図

イメージレベル

特徴16項目

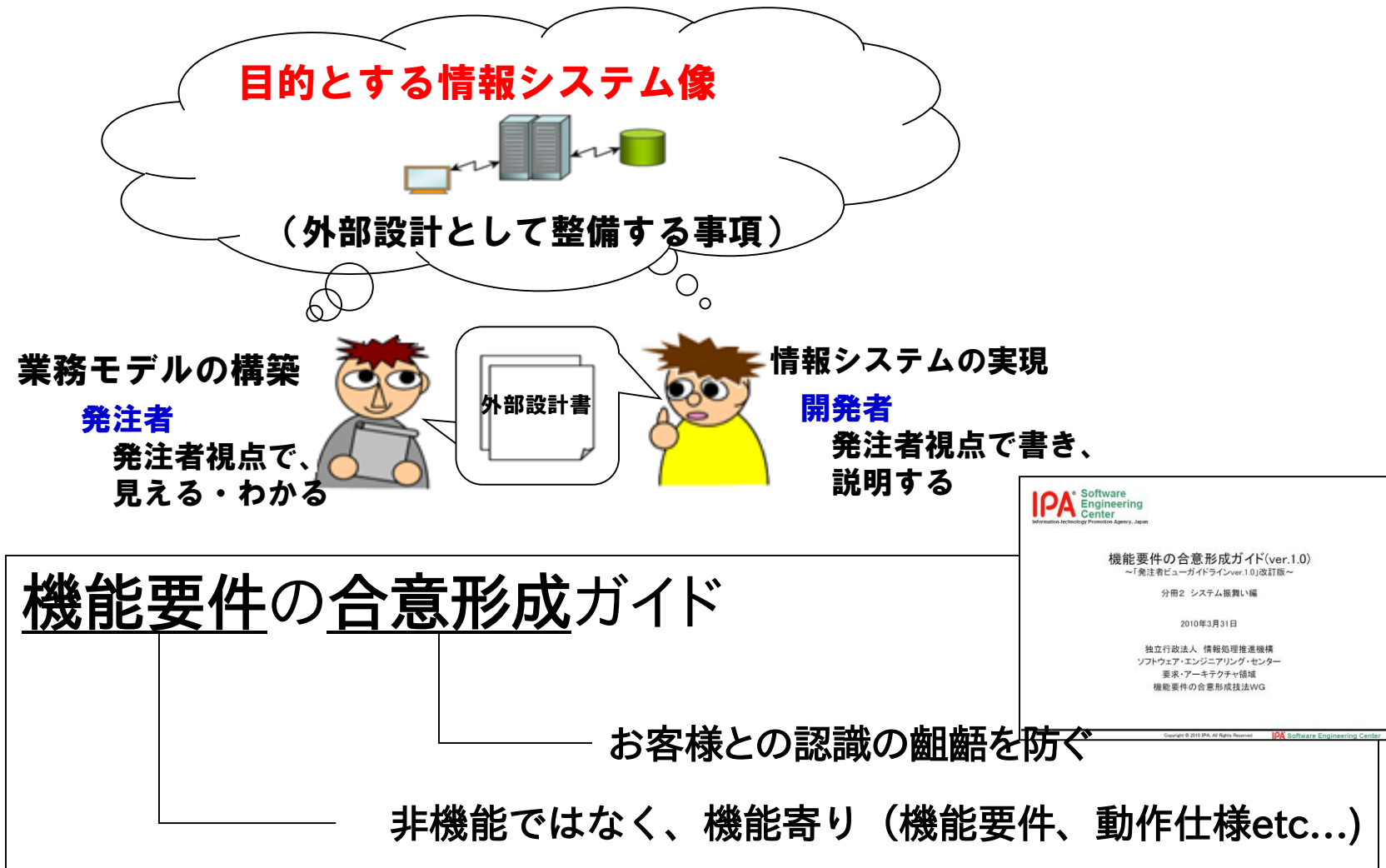
重要項目

92項目

詳細項目

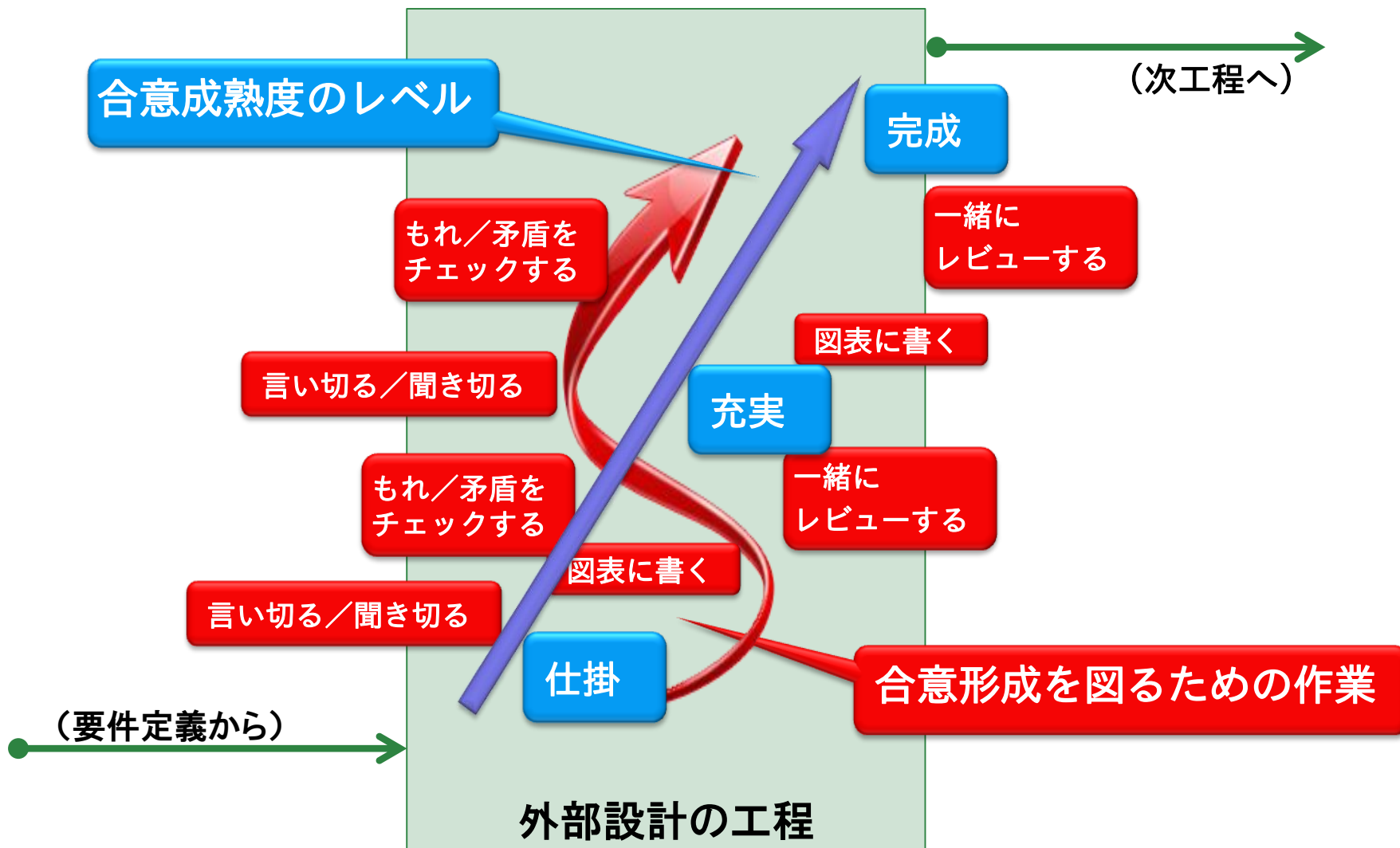
236項目

目的:要件定義におけるお客様との認識の齟齬を防ぐ



<http://www.ipa.go.jp/sec/softwareengineering/std/ent03-a.html>

■ 合意形成の進め方(合意成熟度と4つの作業)

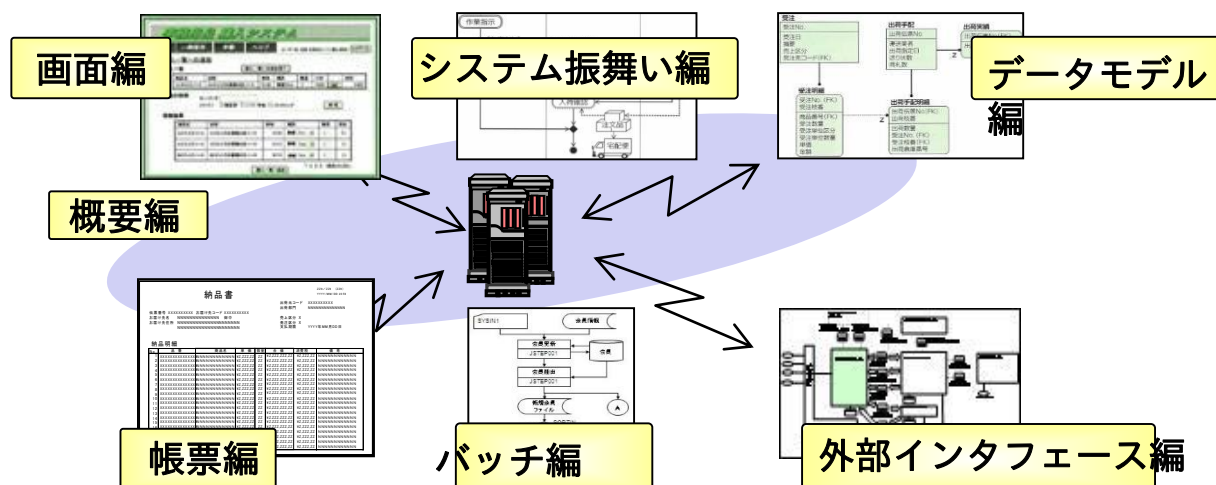


機能要件に着目し、上流工程で実現したい情報システム像を伝え、発注者と開発者との不十分な合意形成が原因で発生する下流工程の手戻りを防止するためのコツを集めたもの

実現したい情報システム像について発注者と開発者が合意形成するために、伝える側が漏れなく正確に情報を提供するためのコツ

発注者と開発者との不十分な合意形成が原因で下流工程で発生する手戻りを防止するための先人の開発者のコツ

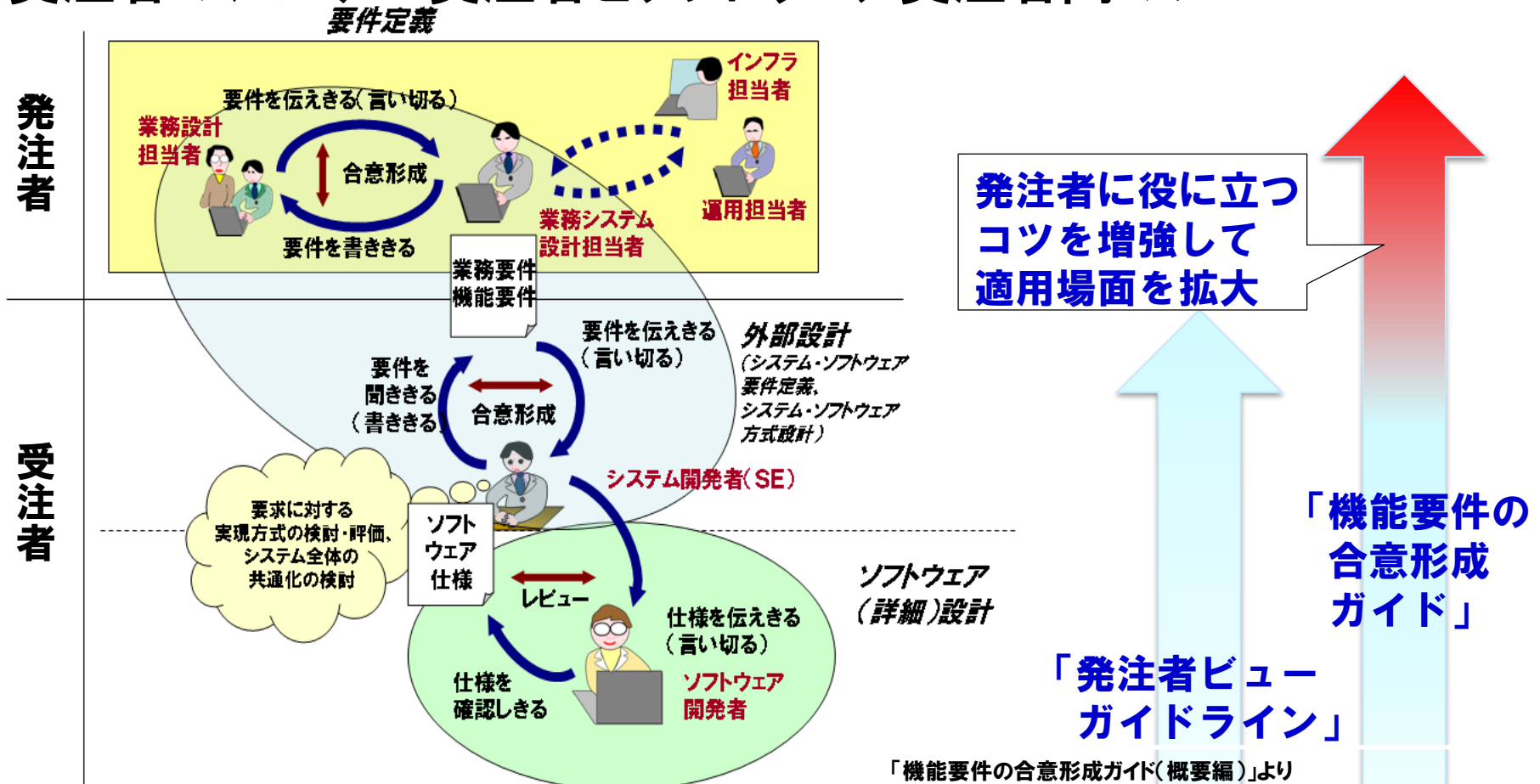
概説編および
6つの技術領域毎の編
の計7冊から構成





本ガイドのコツの使用場面:

- 発注者と受注者間の業務要件・機能要件の定義とレビュー
- 発注者の業務設計担当者と業務システム設計担当者間のレビュー
- 受注者のシステム受注者とソフトウェア受注者間のレビュー



(参考) システム再構築手法の種類と特徴

比較項目		再構築				
		ハードウェア更改	リホスト	リライト	リビルド	
業務要件の変更		変更なし	変更なし	変更なし	見直す	
基盤の変更 (アーキテクチャなど)		変更なし	一部変更あり(アーキテクチャをエミュレート又は非互換と関連部分の再設計)	一部変更あり (アプリ構造、基盤は再設計)	変更あり (再設計)	
プログラム資産の変更		変更なし	変更小 (自動変換+手修正)	異言語で書き換える部分は変更中～大 (アーキテクチャに合わせてリライト)	変更大 (再設計)	
再構築 に必要な投資	検証、品質確保の方法	疎通テストレベル	プラットフォームや言語非互換を中心にしたブラックボックステスト(現新比較テスト)	左記に加えて、言語書き換え(COBOL→java等)の部分はホワイトボックステスト	再設計後の設計書に基づいたホワイトボックステスト	
	業務有識者の 関与度 合い	量的な面	小	小～中	中	大
		質的な面	現行運用・保守の 範囲内	業務シナリオ作成、 テスト結果の妥当性判断	業務シナリオ作成、 テスト結果の妥当性判断	現行運用、業務仕様の網羅的な把握 業務シナリオ作成、 テスト結果の妥当性判断
	再構築コスト	小	小～中	中	大	
	再構築期間	短期間	短～中期間	中～長期間	長期間	
再構築 後の 効果	新しいプラットフォームの活用	不可	可能	可能	可能	
	運用コストの削減	限定的	取り組みは可能	取り組みは可能	可能	
	新規業務への柔軟な対応/業務の効率化	不可	不可	不可	可能	
	アプリケーションの保守性	向上しない(現行の状態を引き継ぐ)	向上しない (現行の状態を引き継ぐ)	基本的に向上しない、 言語書き換え部分は一部向上する	向上する (再設計でスリム化・最適化する)	
	保守コストの削減	不可	不可	部分的な取り組みは可能	可能	

