

昨今重要視されている要件定義課題を 解決するための勘どころ

IPAセミナー@東京
2019年2月26日

独立行政法人情報処理推進機構（IPA）
社会基盤センター 産業プラットフォーム部

システム化要求WG委員
富士通株式会社
森田 功

■ 第3章で説明している事項

- 昨今特に重要になってきている以下の6つの課題を取り上げ、どのように要件定義を進めればよいのかを解説します。

■ 目次

- 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 業務の複雑さを軽減
- 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

3.1 経営や業務に貢献するITシステムの構築（1）

背景

昨今のITシステムは単なる効率化の道具から経営や業務に直接貢献することが求められている。そのため、要求の価値判断が重要である。

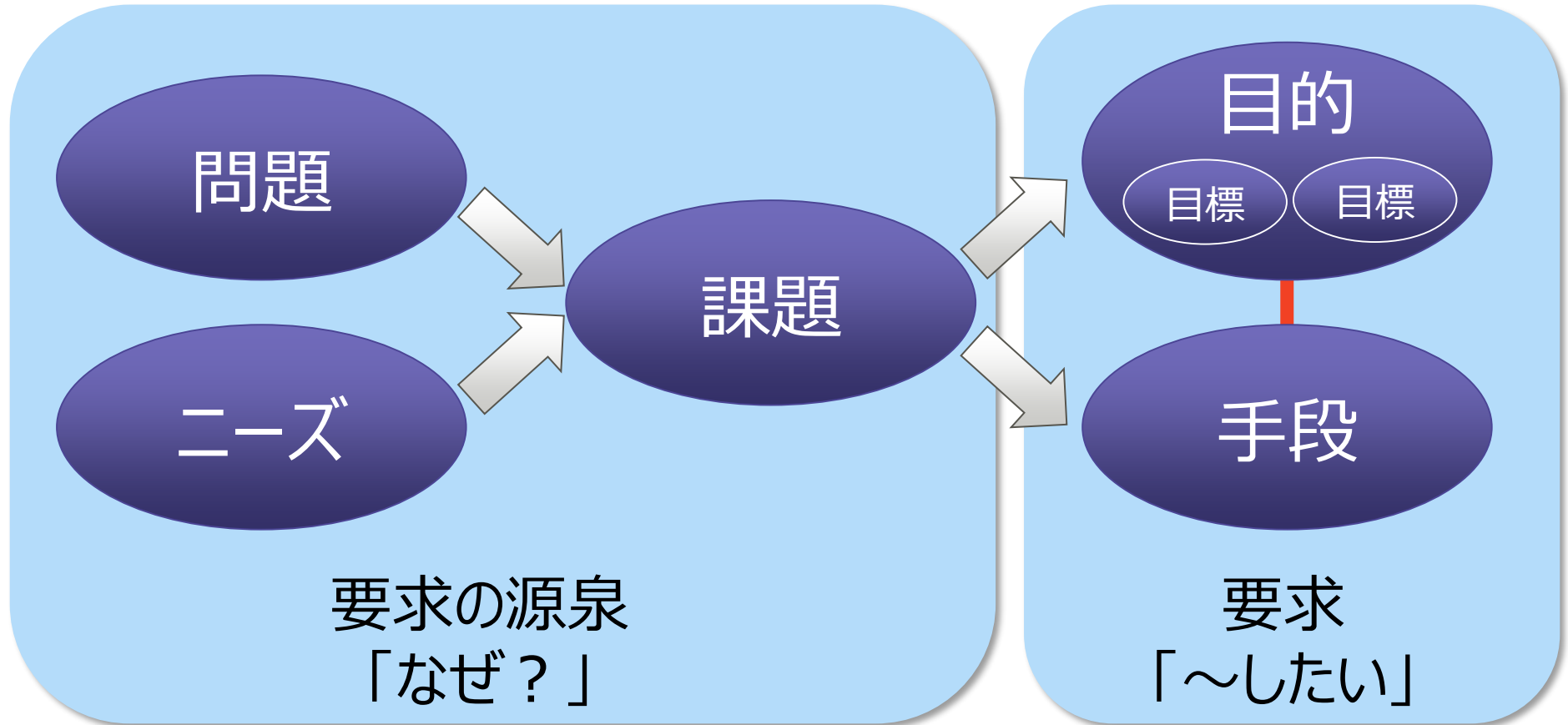
1.3節の課題との関連

- ① ビジネス目的・施策と合致していない
- ② 手段が先行し、「何のために」が理解できていない
- ④ 合意形成が取れていない
- ⑤ 膨らむ要件を抑えきれない
- 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

経営方針や業務目的と整合性がとれた
システム化要求であるかを見極めることが重要

3.1 経営や業務に貢献するITシステムの構築 (2)

～ 要求分析の基礎 ～



この関係をしっかりと捉えることが基本

～ 辞書の意味 ～

そもそもこの意味、違いが分かっているのか？

辞書を引いてみると……

問題：研究・論議して解決すべき事柄

課題：課せられた問題

目的：実現しようとしてめざす事柄

目標：行動を進めるにあたって、実現・達成をめざす水準

手段：ある事を実現させるためにとる方法

分かったような、分からないような……。

～ 問題・課題の違い ～

問題 : 「事実」「あるべき姿と現状との負のギャップ」

課題 : 「自ら設定するもの」「問題を解決するために何をすべきか」

問題

事実

コストの計画値が100であるのに対し、実績値が120だった



課題

意思



Bさんは

「生産性が低い！」

「計画の精度が悪い！」



Aさんは

何を解決すべきなのか : テーマ : チャレンジ

～ 要求と目的・手段の違い ～

要求 : 「～したい」「成し遂げたいもの」



目的 : 状態変化

「～をよくしたい」「価値をあげたい」

経営や業務的になんらかの価値を上げたい、得たい
という要求であり、目指すところ、目指す結果

手段 : どんな手を打つか

「～を実施したい」「～を実行したい」

目的・目標を実現するためにほどこすべき対策を、
実際に行動できる内容で定義したもの

3.1 経営や業務に貢献するITシステムの構築 (6)

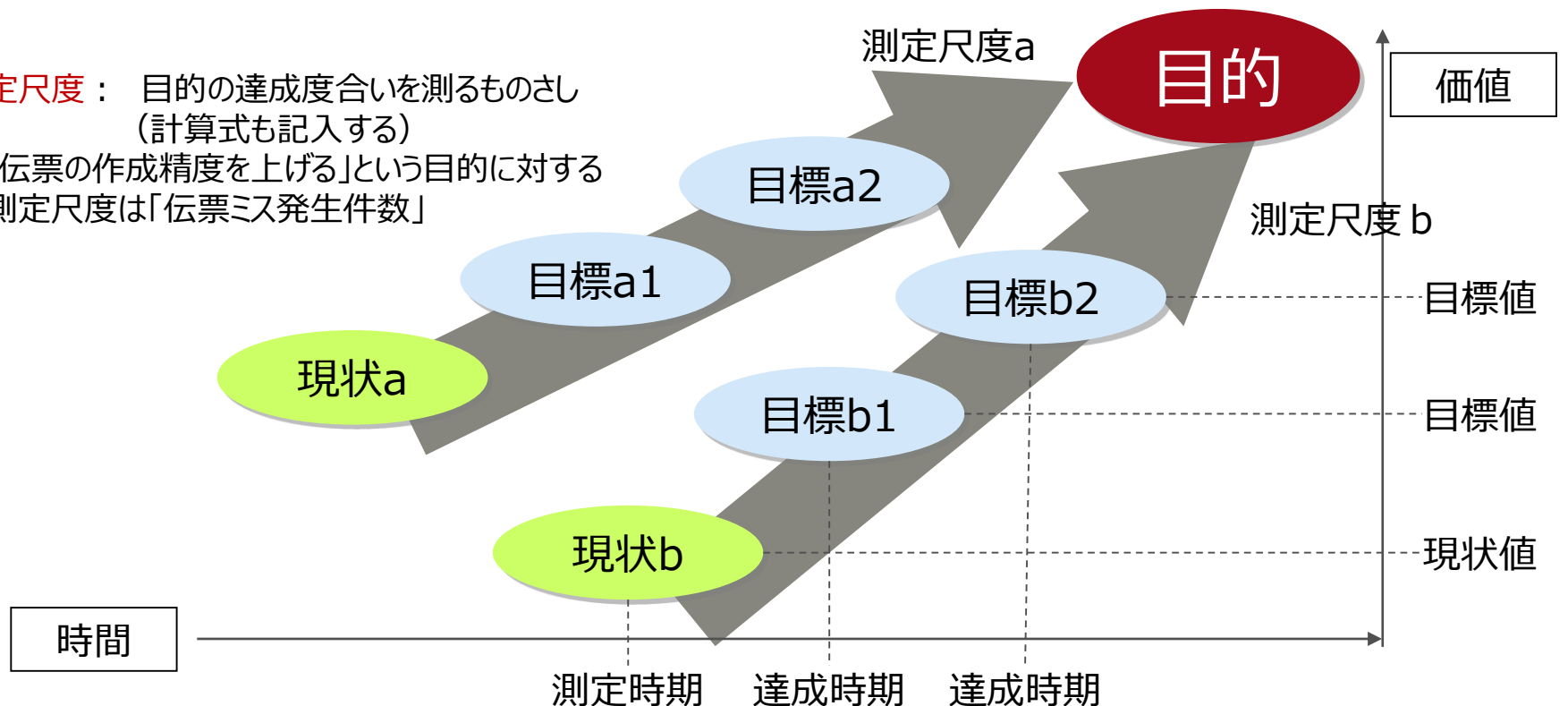
～ 目的・目標の違い ～

目的：経営や業務的になんらかの価値を上げたい、得たいという要求であり、目指すところ、目指す結果

目標：「いつまでに」「どの程度」目的を達成するかを具体的に示したマイルストーン

※測定尺度：目的の達成度合いを測るものさし
(計算式も記入する)

[例]「伝票の作成精度を上げる」という目的に対する
測定尺度は「伝票ミス発生件数」



～ 名言集 ～

- ・「目的」は最終的なゴールであり、「目標」は通過点
- ・「目的」は「目標」の先にある
- ・「目標」は「目的」のためにある
- ・「目標」は過程、「目的」は行き先
- ・「目標」は複数、「目的」はひとつ
- ・「目的」はひとつ大きく、「目標」は複数でより手近に
- ・「目的」は抽象的で、「目標」は具体的
- ・「目標」は見えるモノ、「目的」は見たいモノ
- ・「目的」は見失ってはいけない、「目標」は変化していく
- ・「目標」は諦めても「目的」は諦めない

勘どころ

- ① 経営方針やビジネス目的との関係を明確にすべし
 - 目的を体系化する
 - 目的と手段の違いを認識し、関連付ける

- ② 曖昧な目的・手段を具体化すべし
 - 手段の妥当性を検討する
 - 手段の充分性を検討する

- ③ 真の問題を捉えるべし
 - 真の原因を見極める
 - 最適な解決ポイント（課題）を見つける

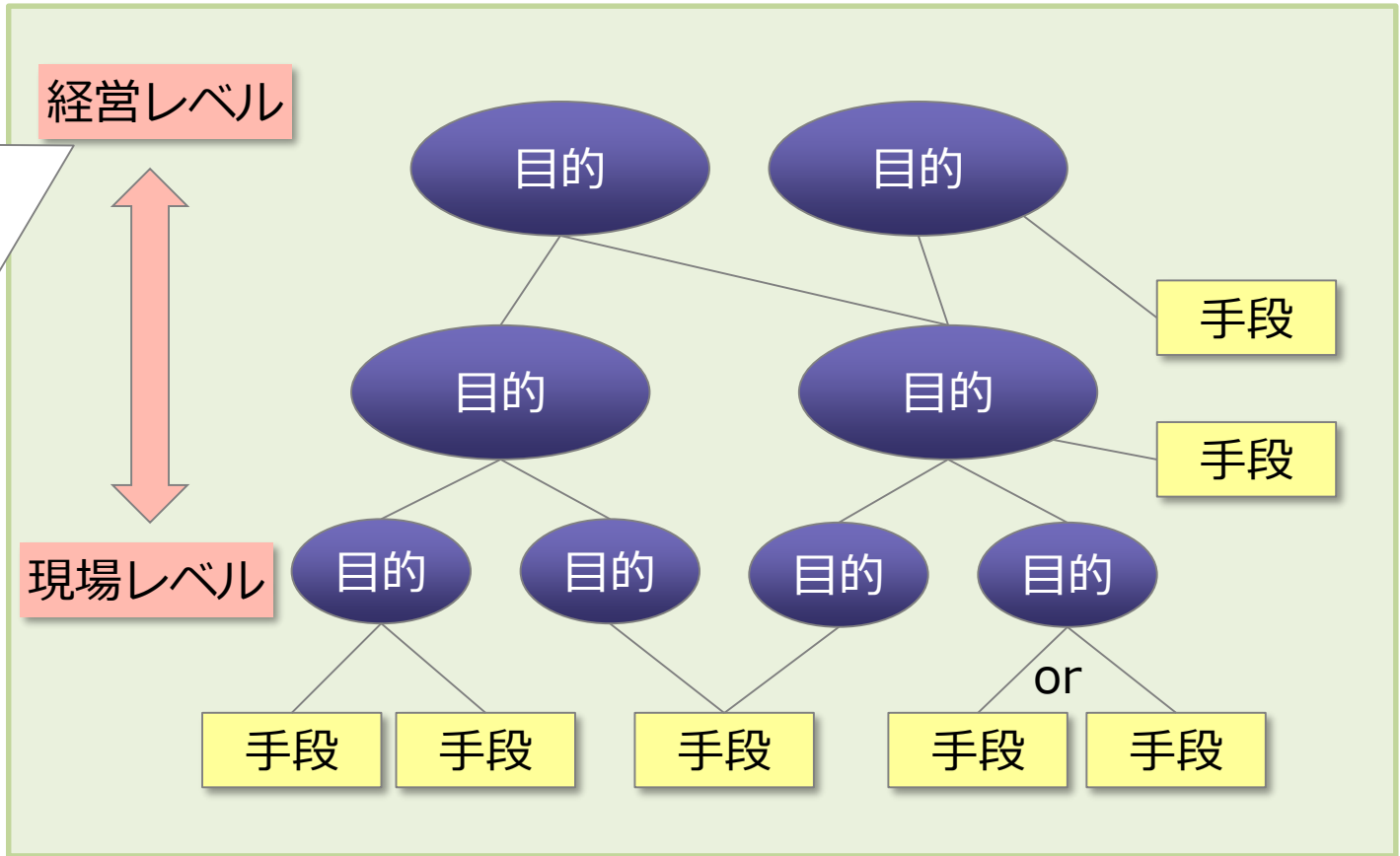
3.1 経営や業務に貢献するITシステムの構築（9）

～勘どころ①：経営方針やビジネス目的との関係を明確にすべし～

目的を体系化する

目的と手段の違いを意識し、
関連付ける

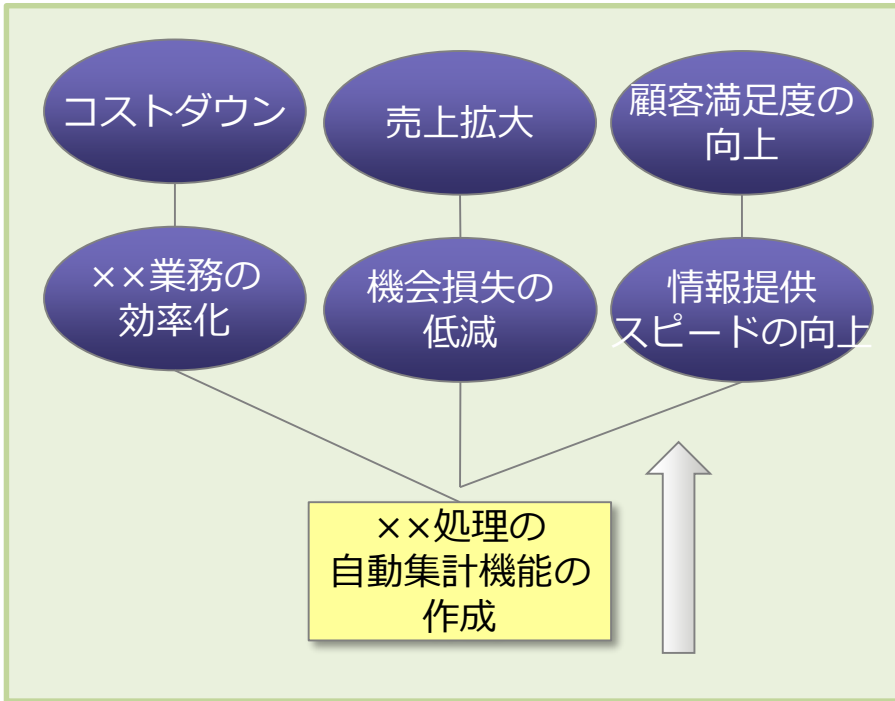
- ・売上を上げる
- ・コストを下げる
- ・品質を向上する
- ・納期を守る
- ・顧客満足度向上
- ・安全性を上げる
- ・リスクを下げる
- ・モラルの向上
- ・モチベーションの向上
- ・コンプライアンスを守る
- ・エコ
- ・社会貢献
-



3.1 経営や業務に貢献するITシステムの構築 (10)

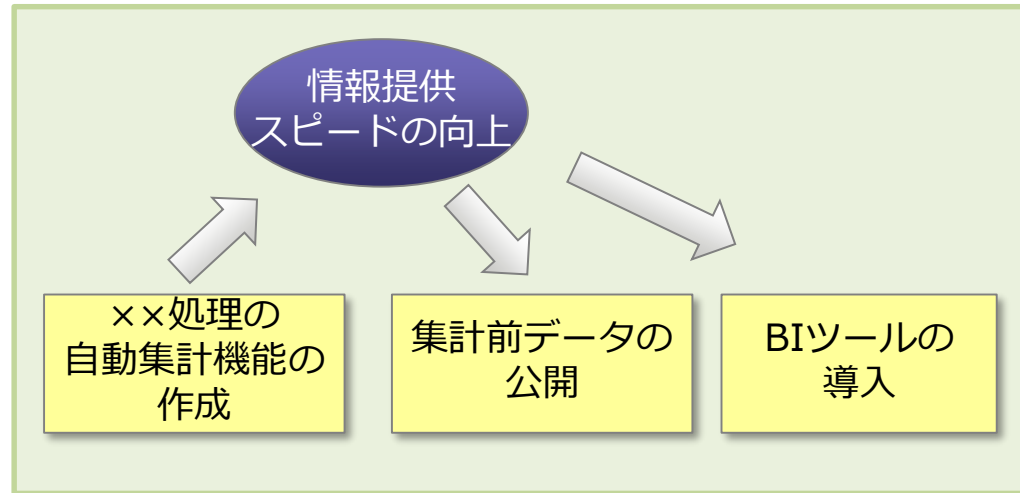
～勘どころ②：曖昧な目的・手段を具体化すべし～

手段の**妥当性**を検討する
この手段は目的にどのくらい貢献しているのか



「なんのため」を問う

手段の**充分性**を検討する
目的を実現するのに他の手段が必要でないか



目的に立ち返って達成手段を考える

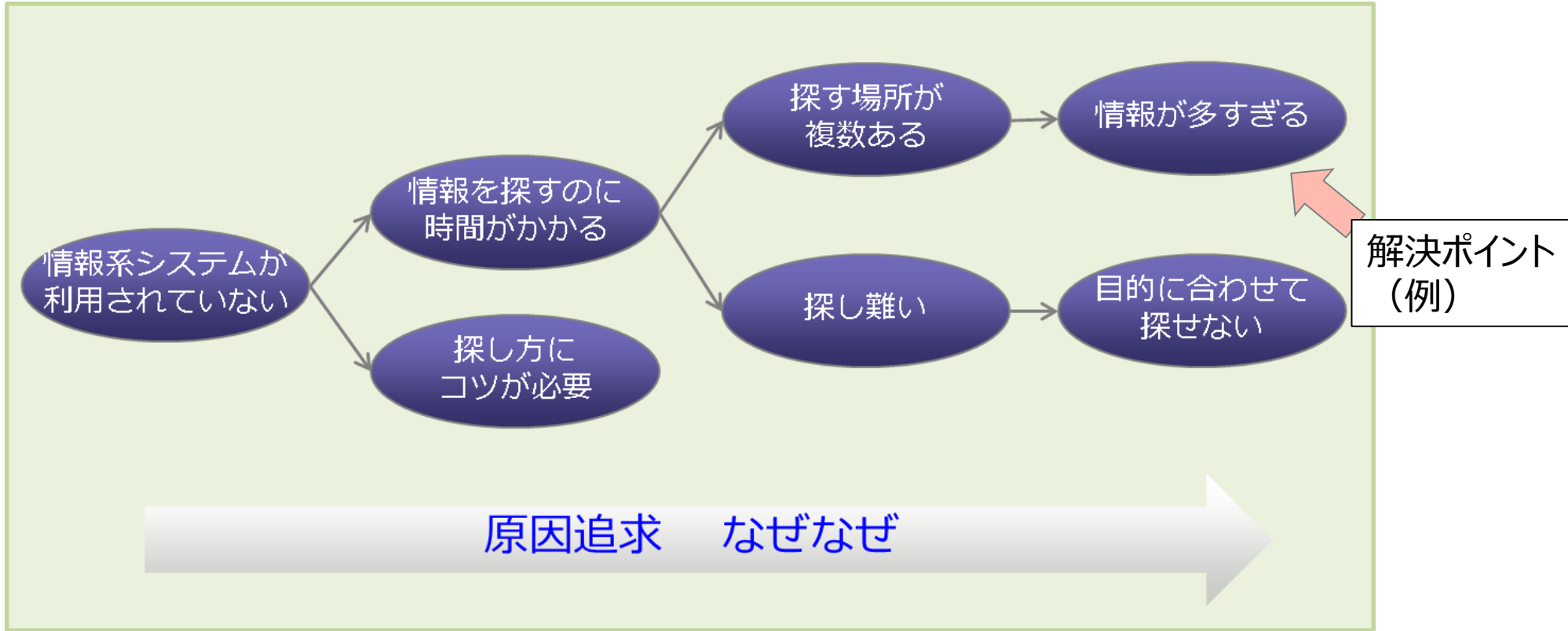
要求の棄却、優先順位の見極めができる

3.1 経営や業務に貢献するITシステムの構築 (11)

～勘どころ③：真の問題を捉えるべし～

真の原因を見極める

最適な課題（解決テーマ）を見つける



真の問題をとらえることにより、正しい課題の設定ができる
納得感、共感を呼ぶ

3.2 膨らむ要求のコントロール（1）

背景

要求が膨らむと、要件定義工程が膨らむだけでなく、開発規模や期間などにも影響を与える。いかに要求の膨らみを抑えるかが重要である。

1.3節の課題との関連

- ④ 合意形成が取れていない
- ⑤ 膨らむ要件を抑えきれない
- 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

要件定義では、要求をいかに漏れなく抽出し、かつ納得性をもって絞り込み合意形成することが重要

3.2 膨らむ要求のコントロール (2)

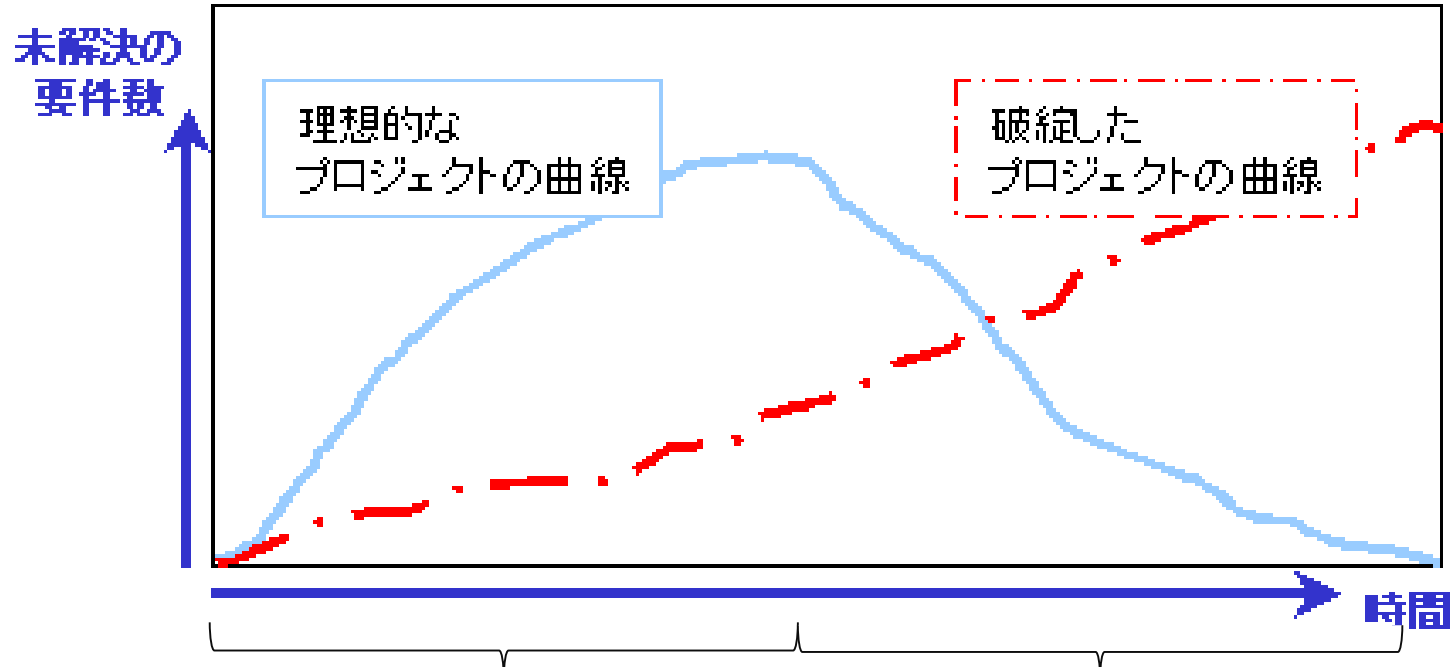
勘どころ

- ① 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
 - 前半では要求を膨らませる
 - 後半では絞り込む
- ② 要求の漏れと過剰要求を是正すべし
 - 評価観点を定め優先度付する
- ③ 定量化した要求量で要求を調整すべし
 - 尺度を使って要求を定量化する

3.2 膨らむ要求のコントロール (3)

～勘どころ①：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



要件定義の前半
(主に業務要求を定義するフェーズ)

- 要求をシステム部門から提案
- 柔軟な要求管理ルール

要件定義の後半
(主にシステム化要求を定義するフェーズ)

- 要求を収束に向かわせる
- 厳しい要求管理ルール

3.2 膨らむ要求のコントロール (4)

～勘どころ②：要求の漏れと過剰要求を是正にすべし～

過剰要求を捨てる

優先度付けに使う評価観点 (例)

- 目的に対する充分性・妥当性→3.1
- 目的に対する効果 (定量・定性)
 - ・ 数値目標だけでなく、計算方法、算出根拠も
- 費用
 - ・ イニシャル、ランニング
- 実現性
 - ・ 技術的実現性、難易度
- 必要性
 - ・ MoSCoW

MoSCoW : Must(必須)、Should(かなり必要)、Could(できたら)、Won't(不要)

3.2 膨らむ要求のコントロール (5)

～勘どころ③：定量化した要求量で要求を調整すべし～

要求を定量化する

要求定量化尺度例とメリット・デメリット

尺度	メリット	デメリット
ファンクションポイント (FP)	<ul style="list-style-type: none"> ・納得性がある ・粒度のバラつきが少ない ・外部データを活用できる 	<ul style="list-style-type: none"> ・計測できる有識者が必要
機能数	<ul style="list-style-type: none"> ・計測が簡易 ・ITに疎くても理解できる 	<ul style="list-style-type: none"> ・粒度にばらつきがあり、精度が低い
画面・帳票数	<ul style="list-style-type: none"> ・計測が簡易 ・ITに疎くても理解できる 	<ul style="list-style-type: none"> ・実現方式により値が異なる

要件の重みにはそれぞれ違いがあることを明らかにする

要求量のベースラインを合意しておく

3.3 業務の複雑さを軽減（1）

背景

変化に柔軟に対応できるITシステムが求められている。システムのアーキテクチャや設計の問題もあるが、要件定義工程では、業務自体の複雑さを少しでも軽減することに着目することも重要である。

1.3節の課題との関連

- ③ 業務の複雑さが増している
- ⑤ 膨らむ要件を抑えきれない
- 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

要件定義では、業務自体の複雑さを低減しよう

3.3 業務の複雑さを軽減（2）

勘どころ

- ① **管理対象のバリエーションを整理すべし**
 - 漏れなく、ダブリなく整理する
 - 不要な分類を廃止する
- ② **業務のバリエーションを減らすべし**
 - 業務プロセスも整理し複雑さを低減する
- ③ **業務パッケージを適切に利用すべし**
 - 業務を標準化するにはパッケージが手っ取り早い
 - しかし注意が必要（パッケージ適用の失敗も多い）

3.3 業務の複雑さを軽減 (3)

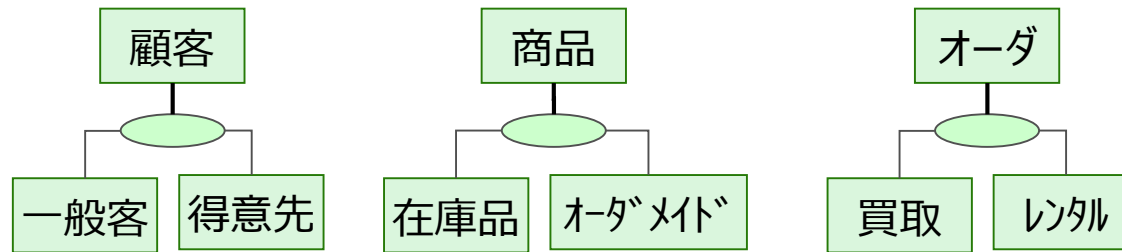
～勘どころ①：管理対象のバリエーションを整理すべし(1/2)～

管理対象の種類とその組み合わせで複雑になっている

■ 管理対象とは：番号を振って管理したくなる“モノ”

顧客、商品、オーダ、口座、……など

■ 管理対象の種類（バリエーション）とは：



■ 管理対象の種類組み合わせとは：

一般客 は オーダメイト を オーダ できない
 オーダメイト は必ず 買取 である

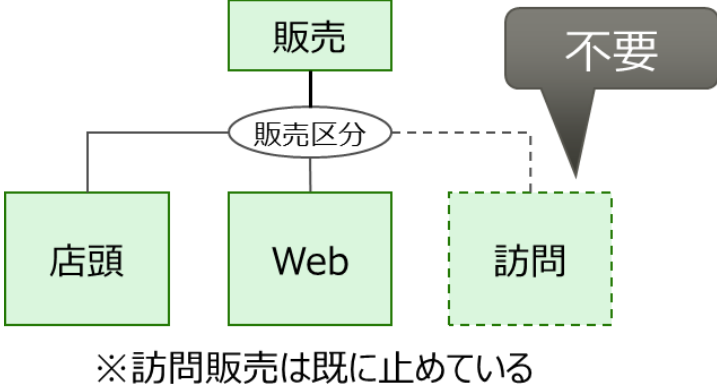
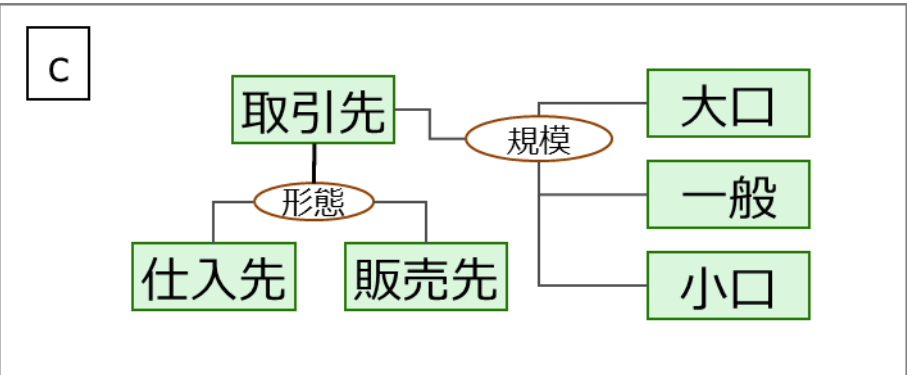
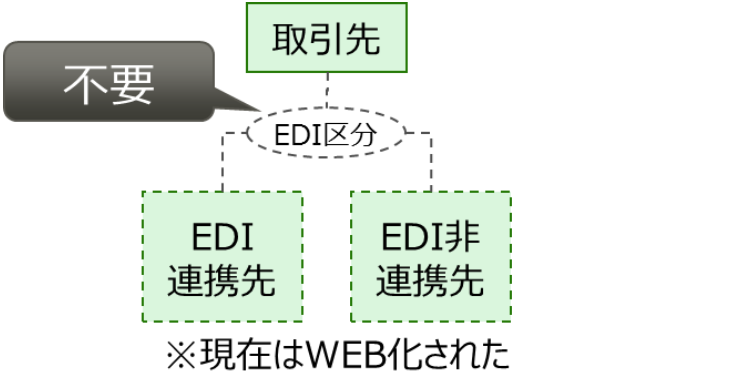
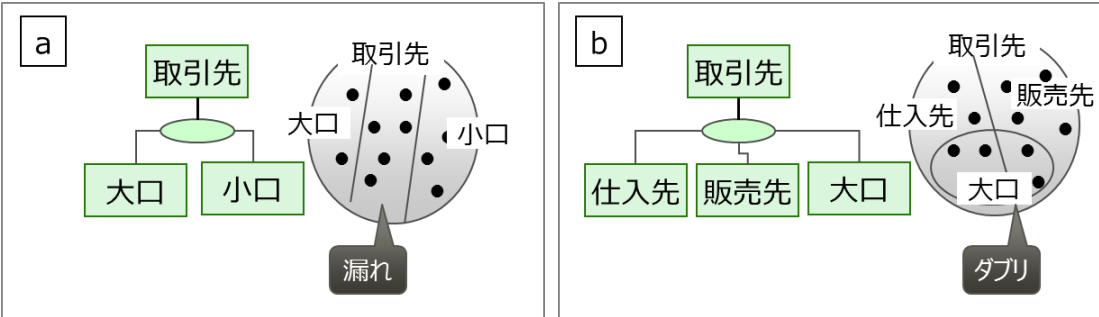
少しでもバリエーションを減らす努力が必要

3.3 業務の複雑さを軽減 (3)

～勘どころ①：管理対象のバリエーションを整理すべし(2/2)～

漏れなく、ダブリなく整理する

不要な分類を廃止する



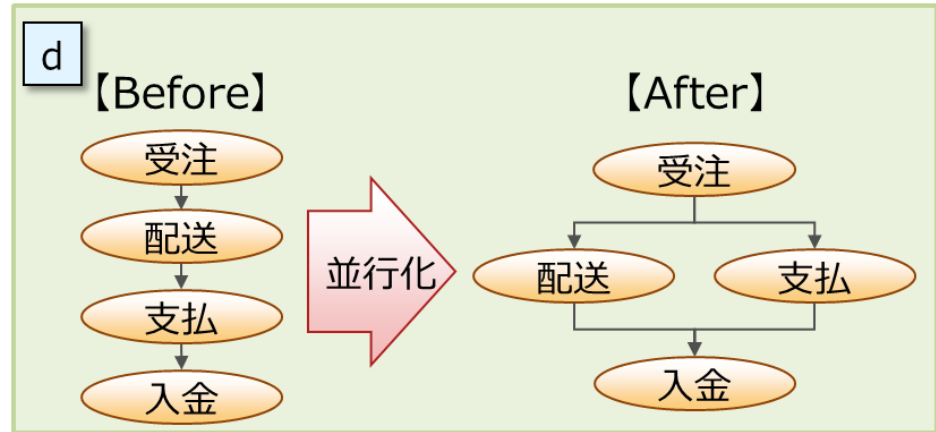
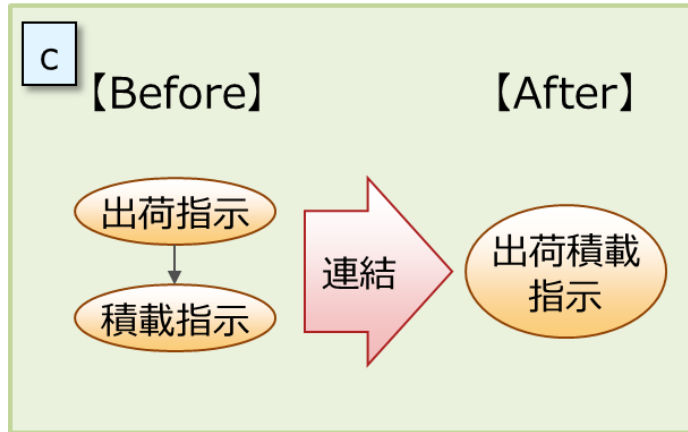
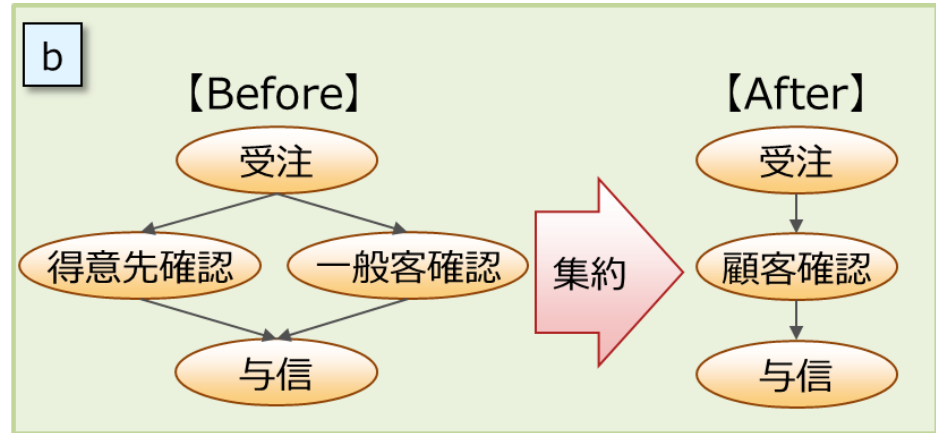
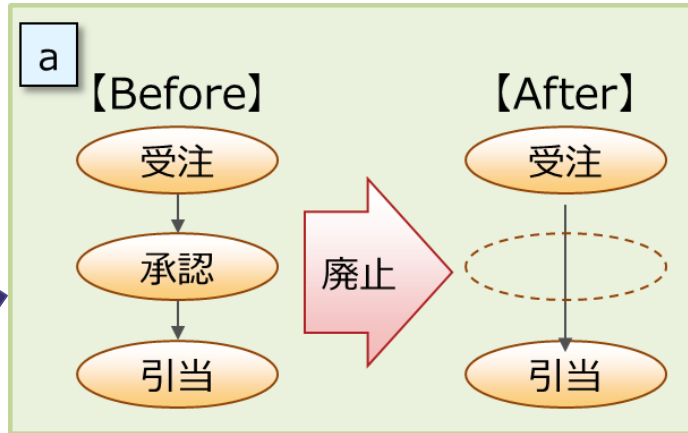
複雑さが減る

削除すると、どこに影響するか心配・・・
勇気を持って廃止しよう！

3.3 業務の複雑さを軽減 (4)

～勘どころ②：業務のバリエーションを減らすべし～

業務プロセスの整理 4つの観点



廃止

集約

連結

並行化

3.3 業務の複雑さを軽減（5）

～勘どころ③：業務パッケージを適切に利用すべし～

パッケージに業務を合わせる

パッケージに業務を合わせることでシンプルにするための近道

業務パッケージ適用の失敗原因はカスタマイズが多くなることである！

注意点：

- 業務をパッケージに合わせるという意識を強く持つ！
- 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
- 経営トップの参画が重要
- 業務パッケージ導入の是非を再度判断せよ

【噂】
カスタマイズ量が20%で
新規開発と変わらない
と言われている

3.4 要件定義工程からの非機能要件定義（1）

背景

非機能要求は、ややもすると、設計工程に入ってから実施しがちである。昨今は、災害やセキュリティ事故などは経営の重要な関心事となっている。多岐に渡る非機能要求を要件定義工程からどう捌いていくかが重要である。

1.3節の課題との関連

- ⑦ 非機能要件を決めきれない
- 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

経営や業務に重要になってきている非機能を
要件定義工程から開始しよう

3.4 要件定義工程からの非機能要件定義（2）

勘どころ

- ① **非機能要求グレードを活用すべし**
→要件定義工程で決めなければならない非機能要求が236項目列挙されている。（知っていますか？）
- ② **基本的な情報を要求として引き出すべし**
→非機能要求項目をステークホルダ別に分けよ
- ③ **矛盾、トレードオフを解消すべし**
→ステークホルダごとに要求を聞きだすと「依存関係」「矛盾関係」「トレードオフ関係」になることがある
これらの解消に注力しなければならない（詳細は省略）

3.4 要件定義工程からの非機能要件定義 (3)

～勘どころ①：非機能要求グレードを活用すべし～

非機能要求グレードのイメージ

大項目	中項目	小項目	重要項目	メトリクス(指標)	レベル						運用コストへの影響	社会的影響が殆ど無いシステム		社会的影響が限定されるシステム		社会的影響が極めて大きいシステム			
					0	1	2	3	4	5		選択レベル	選択時の条件	選択レベル	選択時の条件	選択レベル	選択時の条件		
可用性	継続性	運用スケジュール	○	運用時間(通常)	規定無し	定時内(9時～17時)	夜間のみ停止(9時～21時)	1時間程度の停止有り(9時～翌朝8時)	若干の停止有り(9時～翌朝8時55分)	24時間無停止	2	夜間のみ停止(9時～21時)	夜間を実施する業務はなく、システムを停止可能。 [-] 運用時間をもっと限って業務を稼働させる場合 [-] 停止やリポート処理等ののみを考える場合	4	若干の停止有り(9時～翌朝8時55分)	24時間無停止での運用は必要ないが、極力システムの稼働は継続させる。 [-] 夜間のアクセスは認めないなど、長時間運用を停止する場合 [-] 24時間無停止で運用する場合	5	24時間無停止	システムを停止できる時間帯が存在しない。 [-] 1日のスケジュールで定期的に運用を停止する時間帯が存在する場合
				運用時間(特定日)	規定無し	定時内(9時～17時)	夜間のみ停止(9時～21時)	1時間程度の停止有り(9時～翌朝8時)	若干の停止有り(9時～翌朝8時55分)	24時間無停止	2	夜間の停止(特定日)	運用時間となる特定日 [-] 休日/バックアップ運用など、通常とは異なる運用時間特定日が存在する場合	5	24時間無停止	バックアップ運用のための、システムを停止できる時間帯が存在しない。			
				計画性	無	○	スケジュールの変更不可	スケジュールの変更不可					0	計画停止有り(運用スケジュールの変更不可)	事前の合意があれば、停止可能 [-] 運用時間外での停止だけ可能な場合				
業務継続性	対象業務範囲	内部向け業務	○	内部向け業務	内部向け業務	内部向け業務	内部向け業務	外部向け業務	外部向け業務	全ての業務	2	内部向け業務	内部向けの業務が主要業務であり、内部向け全業務が稼働していることがシステム稼働の条件となる。	3	外部向け業務	[-] 24時間無停止が要求される場合 外部とのバッチな処理で業務が主要業務であり、内部向けの業務および外部とのバッチな業務が稼働していることがシステム稼働の条件となる。 [-] 外部との業務が必要ない場合 [-] 業務継続に、外部とのリアルタイムでの処理が必要な場合	4	外部向け業務	外部とのリアルタイムでの処理が主要業務であり、外部向けオンライン業務が稼働していることがシステム稼働の条件となる。 [-] 業務継続に、外部とのリアルタイムでの処理が必要とならない場合
				外部向け業務	外部向け業務	外部向け業務	外部向け業務	外部向け業務	外部向け業務	外部向け業務	外部向け業務	外部向け業務	外部向け業務	外部向け業務	外部向け業務	外部向け業務	外部向け業務	外部向け業務	外部向け業務

レベルから選択

2 3 6メトリクス

3つのモデルシステム
 ・社会的影響が殆ど無いシステム
 ・社会的影響が限定されるシステム
 ・社会的影響が極めて大きいシステム

6大項目

大項目	内容
可用性	情報システムを継続的に利用可能とするための要件 【例】業務継続性、目標復旧水準、稼働率、災害対策 など
性能・拡張性	情報システムの性能と将来の拡張に関する要件 【例】業務量、レスポンス・スループット、リソース拡張性 など
運用・保守性	情報システムの運用と保守のサービスに関する要件 【例】通常運用、障害時運用、運用環境、運用管理 など
移行性	現行システム資産の移行に関する要件 【例】移行時期、移行方式、移行対象、移行計画 など
セキュリティ	構築する情報システムの安全性の確保に関する要件 【例】リスク分析・診断、データの秘匿、不正監視 など
システム環境・エコロジー	情報システムの設置環境やエコロジーに関する要件 【例】拠点数、CO2排出量、適合規格、耐震/免震 など

ダウンロード(無料)
<http://www.ipa.go.jp/sec/softwareengineering/reports/20100416.html>

3.4 要件定義工程からの非機能要件定義 (4)

～勘どころ②：基本的な情報を要求として引き出すべし～

非機能要求をステークホルダ別に振り分けた例

ビジネス領域

経営層や利用部門の関心の高い領域。
ビジネスの要求として捉えるものであり、
判断に困ったときのよりどころとなる重要な要求。

ソリューション領域

利用部門や運用部門の関心が高い領域。
ITシステムが提供するサービスに対する要求。

テクニカル領域

システム部門が検討し提示していくIT専門
技術要素の強い領域。
技術や構成要素に対する要求。

領域	大項目	中項目
ビジネス領域	ビジネス要求/ 基本情報	業務特性、サービス時間、業務継続、 業務連携、業務量、運用方針、内部 統制、セキュリティポリシー
ソリューション 領域	可用性	業務復旧
	性能・拡張性	拡張指針、性能目標、性能検証指針
	運用・保守性	運用監視、運用自動化、システム連携、 ライフサイクル、サポート体制、システム 保守、システム復旧、システム運用環境、 システム運用管理
	移行性	移行計画、システム移行
	セキュリティ	情報セキュリティ
	システム環境・ エコロジー	システム環境、設備規格
	ユーザビリティ	ユーザビリティ目標、ユーザビリティ指針
テクニカル領域	可用性	資源冗長化、データ保護対策
	性能・拡張性	資源拡張性、帯域保証
	運用・保守性	バッチ運用指針
	セキュリティ	セキュリティ対策
	システム環境・ エコロジー	設備環境
	ソフトウェア 保守・移植性	ソフトウェア保守・移植性

(出展) 富士通「要件定義手法 Tri-shaping」

3.4 要件定義工程からの非機能要件定義 (5)

～勘どころ③：矛盾、トレードオフを解消すべし～

要求のレベルの整合性を判断する

制約・前提条件

ビジネス上の制約

法令、商慣習、予算

技術的な制約

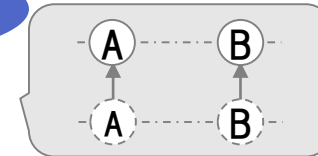
特定技術採用、技術人材、再利用素材

現在の状況

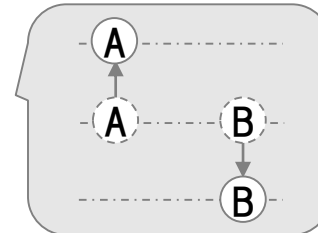
- 要求レベルがジャンプアップしていないか
- (許容できない範囲で) 悪くないか

整合性の判断

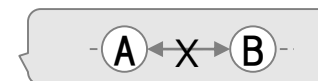
- 依存関係
例) サービス利用者数—システム処理量
- トレードオフ関係
例) セキュリティ対策—性能目標
- 矛盾関係
例) サービス時間—システム保守



要求Aのレベルを**上げる**と
要求Bのレベルも**上がる**



要求Aのレベルを**上げる**と
要求Bのレベルが**下がる**



要求Aのレベルと
要求Bのレベルが**成立しない**

3.5 多様化するステークホルダとの合意形成（1）

背景

昨今の要件定義では、関係するステークホルダが多岐に広がっている。それらステークホルダと確実に合意形成を行うことが重要である。

1.3節の課題との関連

- ④ 合意形成が取れていない
- 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

多様化するステークホルダを漏れなく抽出、分析することが重要

勘どころ

- ① 漏れなく、的確にステークホルダを分析すべし
 - 漏れのないステークホルダの抽出
 - プロジェクトへの影響度と関心度で分析する
- ② 当事者意識を持ってレビューをすべし
 - 問題、ニーズ、課題や成果物を一人称で扱う
- ③ 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する

3.5 多様化するステークホルダとの合意形成 (3)

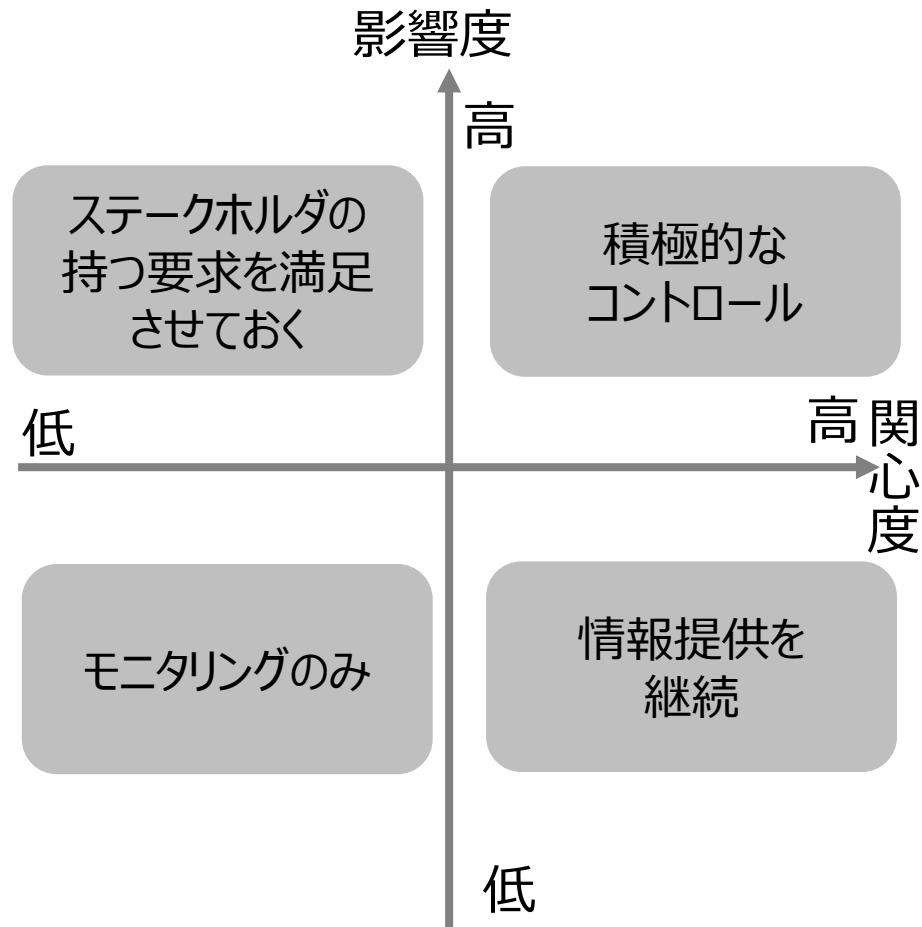
～勘どころ①：漏れなく、的確にステークホルダを分析すべし～

ステークホルダの抽出

ステークホルダの抽出観点

- 直接的な関わりを持つ関係者
 - 対象業務の運用者
 - 対象システムの運用者
 - 対象業務・システムの最終利用者 (お客様やエンドユーザなど)
- 業務の結果で評価される人 (経営層など)
- 間接的な関わりを持つ関係者
- 対象業務と関連する顧客、取引業者
- 対象業務と関連する業務の運用者
- 対象業務と関連する業務の結果で評価される人 (経営層など)
- 対象システムと関連する外部システムの運用者
- 対象業務の主管部門以外の組織の人
- 対象システムの外部システム

プロジェクトへの影響度と 関心度で分析する



3.5 多様化するステークホルダとの合意形成 (4)

～勘どころ②：当事者意識を持ってレビューすべし～

問題、ニーズ、課題や成果物を一人称で扱う

要求抽出ワークショップ

- 問題・ニーズのブレインストーミング
- 模造紙や付箋を使った
 - 問題ニーズの深掘り、課題抽出
 - 目的、手段の構造化、詳細化

成果物レビュー

- 業務フローに沿ったシミュレーション



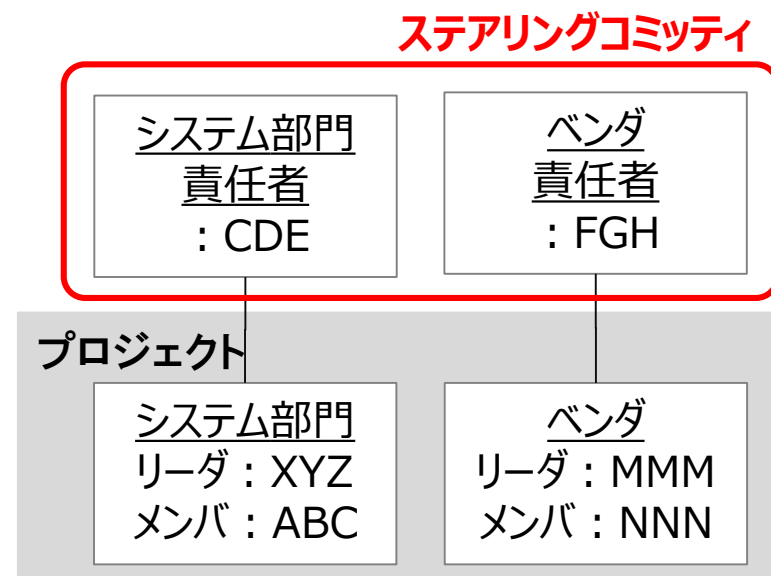
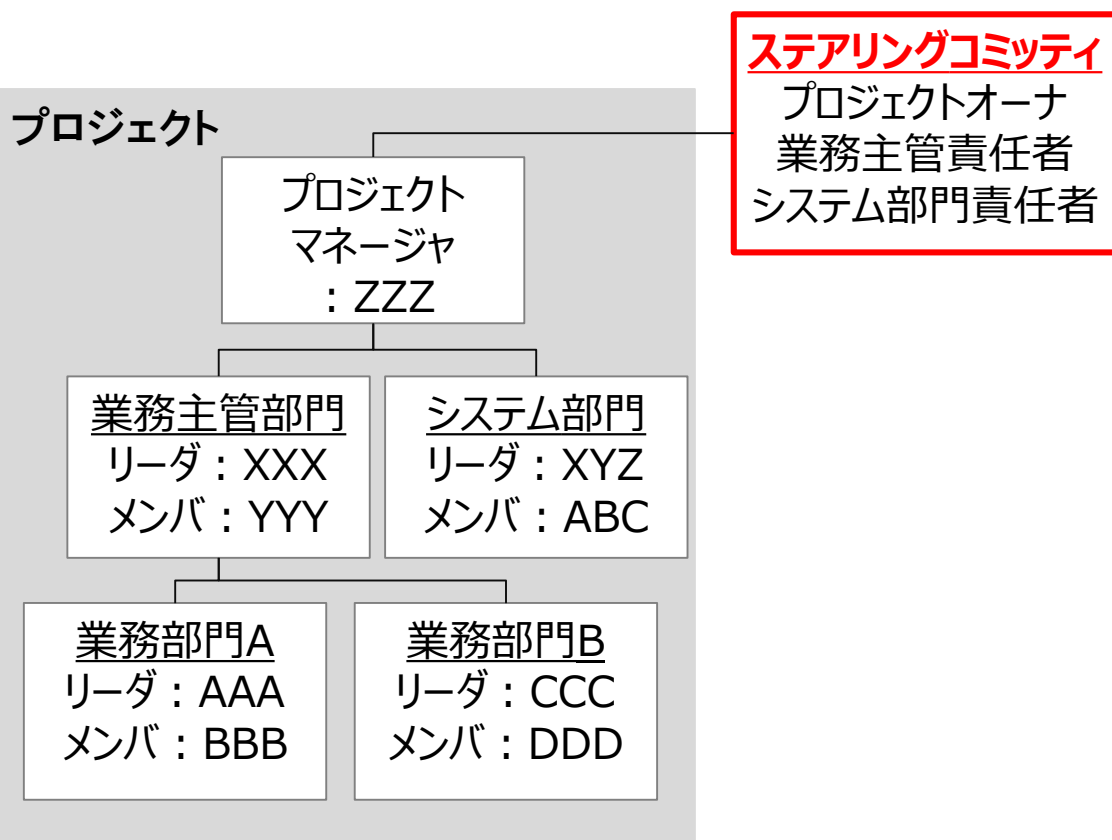
3.5 多様化するステークホルダとの合意形成 (5)

～勘どころ③：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する

業務部門／システム部門の体制図

システム部門／ベンダの体制図



3.6 現行業務やシステムの把握（1）

背景

昨今のITシステム開発では、現行システムが存在していないことが極めて少ない。逆に、業務がシステムに隠蔽されていて見えなくなっている。いかに現行業務やシステムを可視化し関係者で共有するかが重要である。

1.3節の課題との関連

- 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

現行業務やシステムを実態と照らし合わせて
理解・共有することが重要

3.6 現行業務やシステムの把握 (2)

勘どころ

- ① 現行業務／システムを可視化すべし
 - システムを可視化する
 - 詳細から全体像に概略化する

- ② 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）を発掘する
 - 可視化した業務を実態と照らし合わせて確認する

- ③ プロジェクト全員で共通認識を構築すべし
 - 読み合わせやウォークスルーで関係者間の認識を合わせる
 - 稼働後、安定的に業務を運用するための業務移行も念頭に置く

3.6 現行業務やシステムの把握 (3)

～勘どころ～

① 現行業務 / システムの可視化

システムを可視化することが現行業務、システム理解の第一歩
概略化し、全体像を文書化することが必要

→俯瞰的に整理することで全体像を理解しやすくなる

② フィールドワークを通じた現状の理解

既存ドキュメントや現行業務担当者からの再学習（リラーン）
を通じて可視化結果の補完を行う

机上だけでなく、フィールドワークを通して確認することが重要
→潜在する問題、ニーズ、課題を自分事に捉えられる

③ 共通認識の構築

整理した結果をプロジェクト全員で読み合わせやウォークスルー
する

→関係者間での認識違いや共通認識を確認し、同じ方向性
でプロジェクトを進められる

価値ある業務／システムの要件定義を進める上での勘どころを解説した

第3章で紹介した勘どころ

テーマ	おもな勘どころ	テーマ	おもな勘どころ
3.1 経営や業務に貢献するITシステムの構築	経営方針やビジネス目的との関係を明確にすべし	3.4 要件定義工程からの非機能要件定義	非機能要求グレードを活用すべし
	曖昧な目的・手段を具体化すべし		基本的な情報を要求として引き出すべし
	真の問題を捉えるべし		矛盾、トレードオフを解消すべし
3.2 膨らむ要求のコントロール	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	3.5 多様化するステークホルダとの合意形成	漏れなく、的確にステークホルダを分析すべし
	要求の漏れと過剰要求を是正すべし		当事者意識を持ってレビューをすべし
	定量化した要求量で要求を調整すべし		エスカレーションパスを明確にすべし
3.3 業務の複雑性を軽減	管理対象のバリエーションを整理すべし	3.6 現行業務やシステムの把握	現行業務／システムを可視化すべし
	業務のバリエーションを減らすべし		フィールドワークを通して現状を理解すべし
	業務パッケージを適切に利用すべし		プロジェクト全員で共通認識を構築すべし

「なぜ～？」「～のために」をユーザ部門／システム部門／ベンダが協調して考え抜くことが要件定義成功の鍵

ご清聴ありがとうございました

SEC BOOKS : 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～

<http://www.ipa.go.jp/sec/publish/tn16-008.html>

システム構築の要件定義に役立つポイント集を公開
～家づくりを例に、要求を明確にする勘どころ概説～

<https://www.ipa.go.jp/sec/reports/20180327.html>