

要件定義概論

Ver1. 3. 0



この作品は [クリエイティブ・コモンズ 表示 - 継承 4.0 国際 ライセンス](https://creativecommons.org/licenses/by-sa/4.0/) の下に提供されています。

要件定義概論

1. 要件とは？
2. 要件定義で何が起こっているか？
3. 要件定義の概念プロセス、成果物
4. 要件定義の基礎知識

「要件」ってなに？

■ 一般的には

【大辞林】

- ✓ 必要な条件。欠くことのできない条件

■ システム工学的には

【IEEE830】 ※抜粋

- ✓ 問題を解決したり、目標を達成するために、ステークホルダーが必要とする条件、あるいは、能力。
- ✓ 契約、標準、仕様、あるいは、その他の正式に要請された文書を満たすために、システム、あるいはシステムコンポーネントが満たすべき条件、もつべき能力。

【共通フレーム2013】 ※抜粋

- ✓ 製品またはプロセスの運用/機能/設計上の特性を識別する文で、曖昧性がなく、テスト可能で、測定可能で、顧客に受け入れられるために必要であるもの。

要件定義工程の位置づけ①

■ 要件定義の目的

ビジネス目的・目標を達成する『手段』としての業務・システムを決定すること

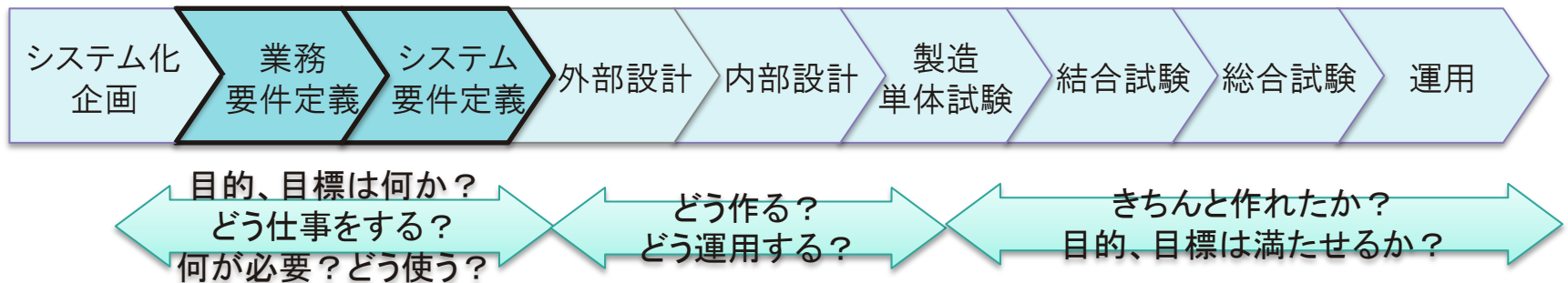
業務要件定義では

ビジネス目的・目標を達成するために、
解決すべき業務課題、実現すべき業務を定義する。

システム要件定義では

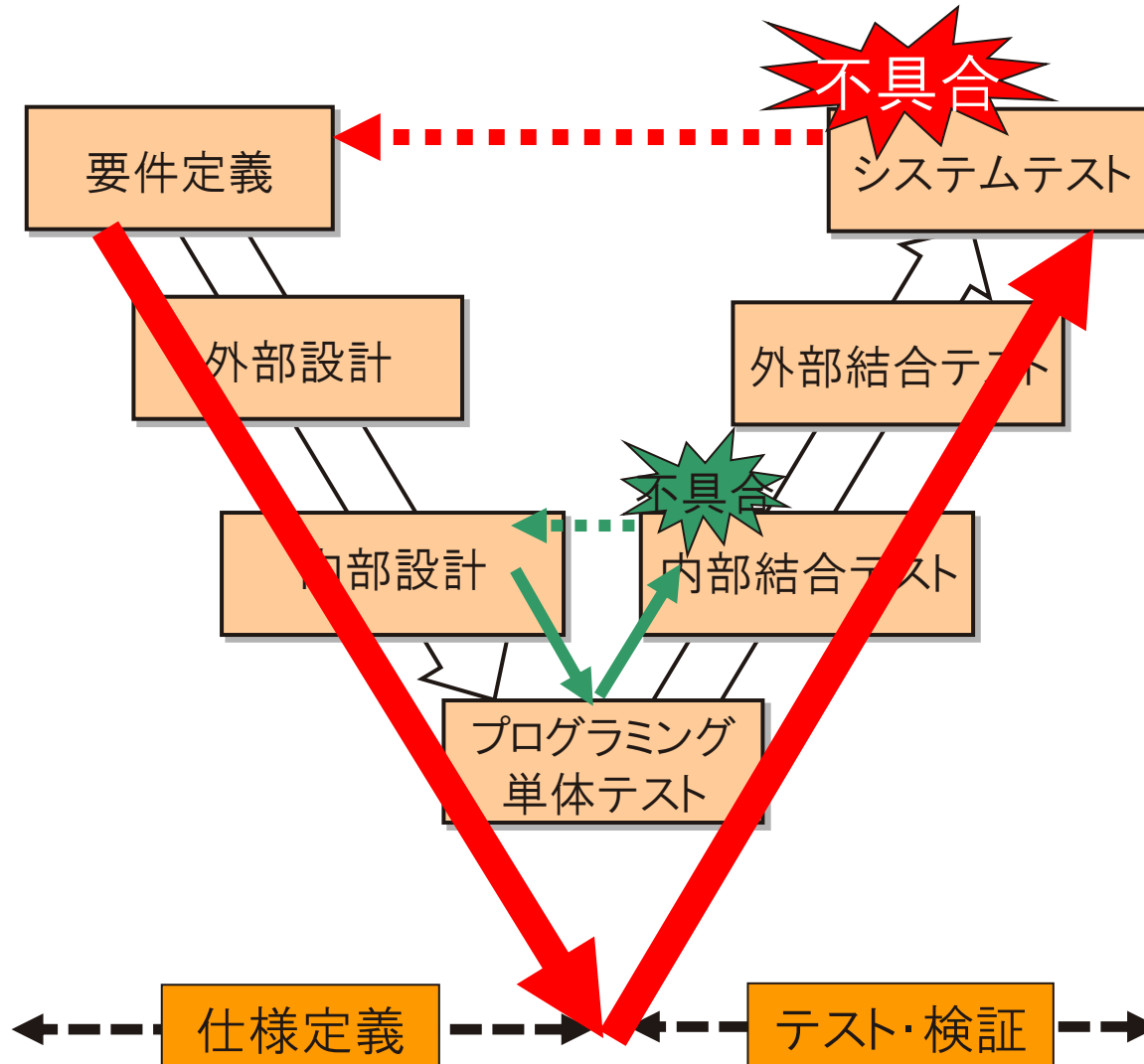
実現すべき業務のために、解決すべきシステム課題、
実現すべきシステムを定義する。

■ 要件定義工程の位置付け



要件定義工程の位置づけ②

- 要件定義の誤りを抑えることが、システム開発の重要成功要因のひとつ。

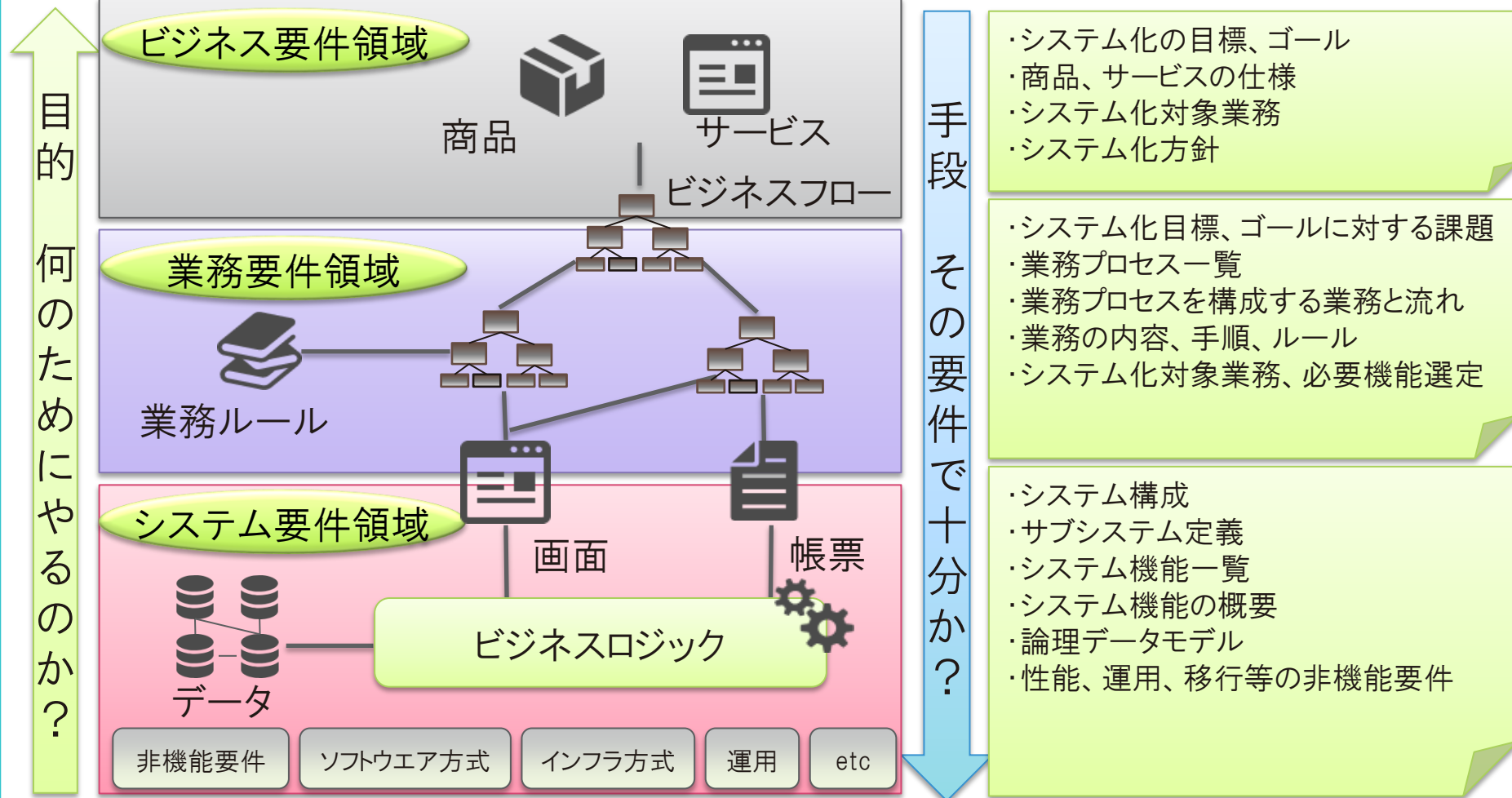


要件定義でやること①

- お客様のビジネスや業務にフィットしたシステム要件を決める。

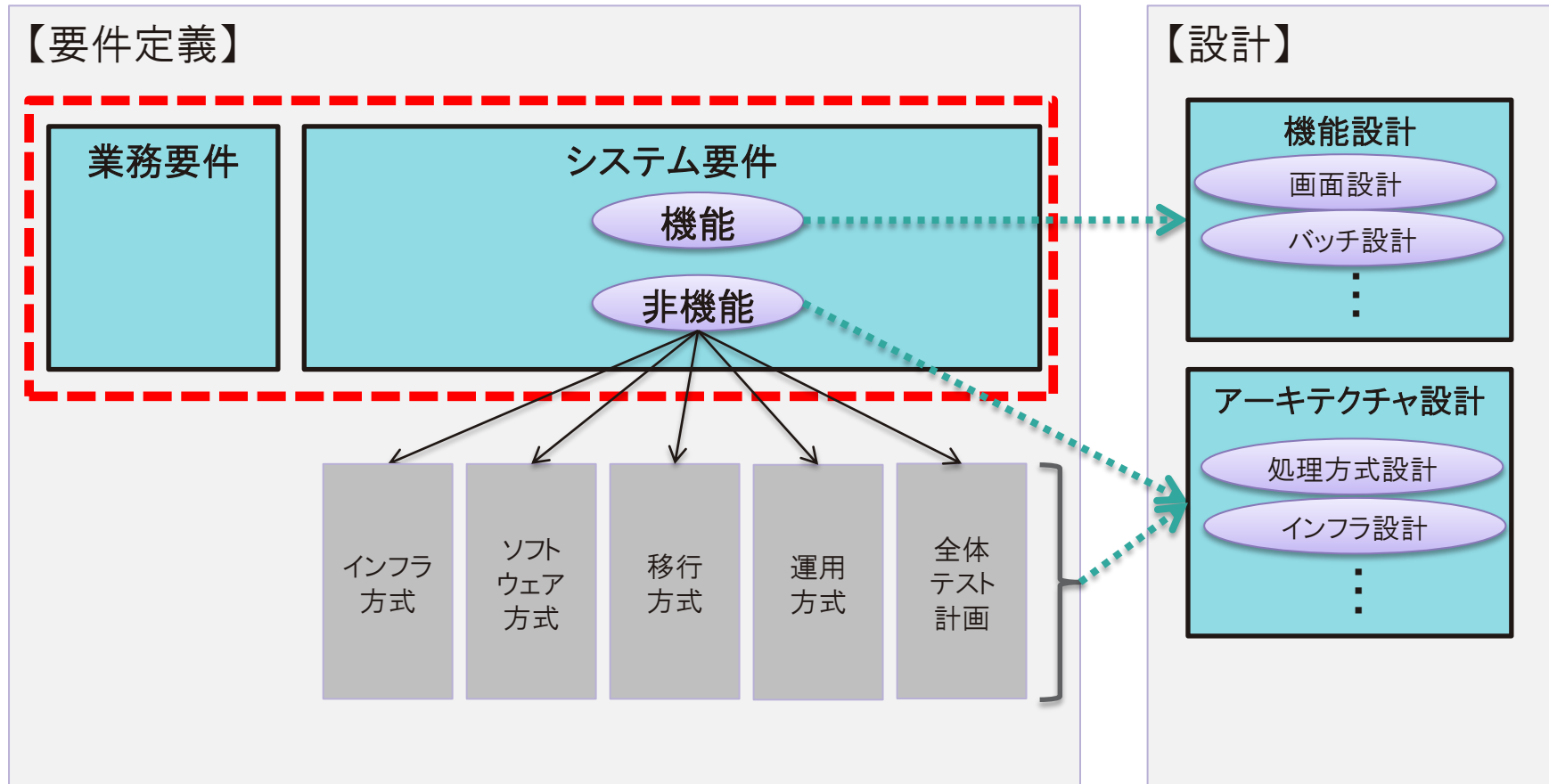
＜各領域の範囲＞

＜主な定義事項＞



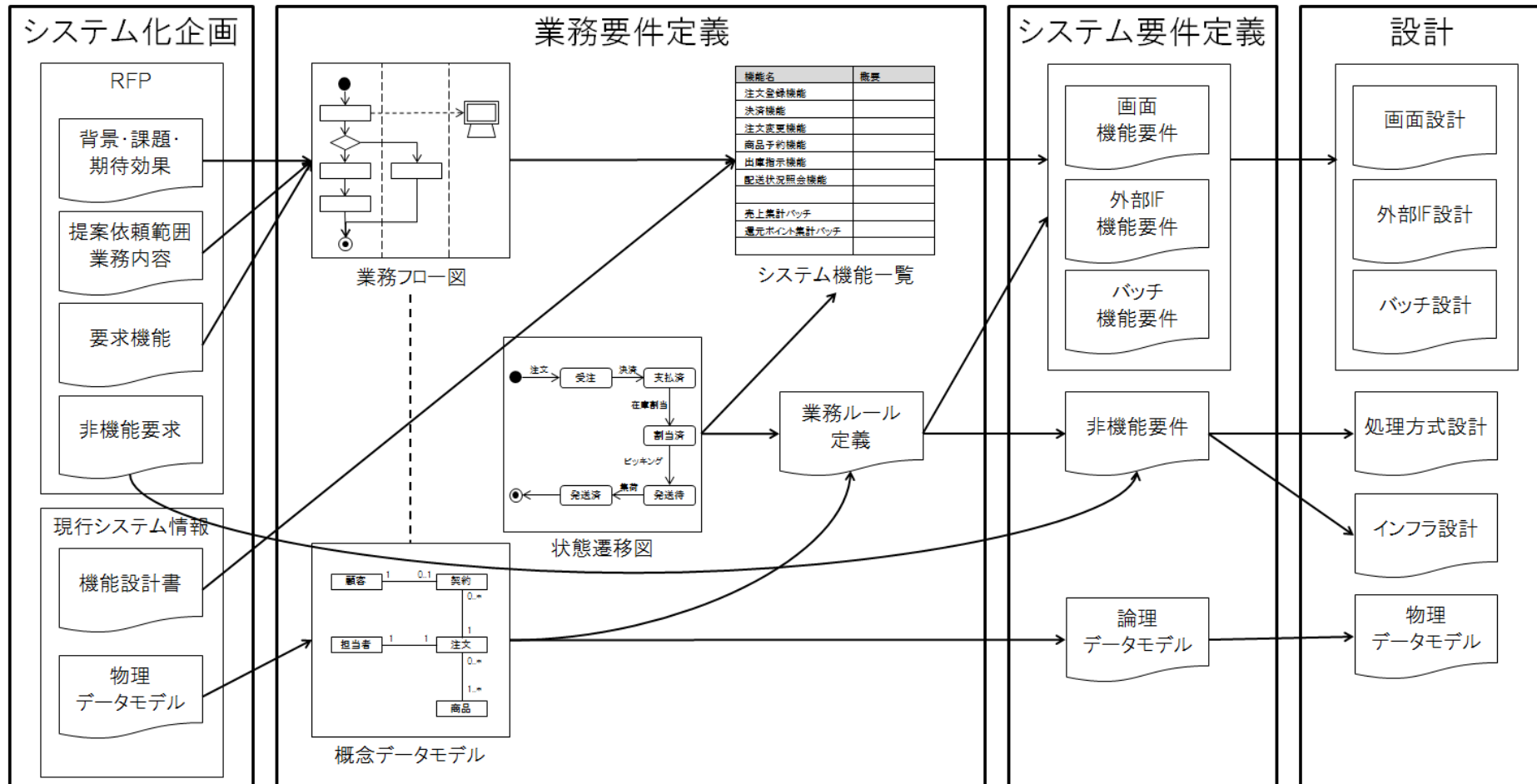
要件定義でやること②

- 業務や業務アプリケーション等の、多岐に渡る要件を定義し、設計へ繋げる。



要件定義でやること③

■ トレーサビリティを確保する。



要件定義概論

1. 要件とは？
2. 要件定義で何が起きているか？
3. 要件定義の概念プロセス、成果物
4. 要件定義の基礎知識

要件定義で起こっている問題

■ インプットに関連する問題

1. 必要なシステム要件を決められない。明確な基準に従った決定・判断ができない。
2. 曖昧なお客さま要求事項の具体化に時間を要し、工数・工期が膨らむ。
3. ステークホルダーの洗い出し、業務上の役割の分析に不備があり、要求収集が漏れる。
4. お客さまからの要件定義へのインプット情報・文書提示が十分でなく、工数・工期が膨らむ。

■ 進め方に関連する問題

5. ステークホルダー間で要求内容、実現する要求事項の調整が折り合わず、合意が遅れる。
6. 決めること、進め方が曖昧。なし崩し的に要件定義が終了し、設計が開始される。
7. お客さま要求事項をそのまま受け入れる「御用聞き」で、後工程で要求の過不足が起こる。
8. お客さま要求事項の増加に歯止めが効かず、予定開発コストを超過する。
9. 工程終盤の要件合意・承認で、急な路線変更や、理不尽な要求が発生する。
10. 設計、実装に入ってから、要件の合意が無効になる。

■ アウトプットに関連する問題

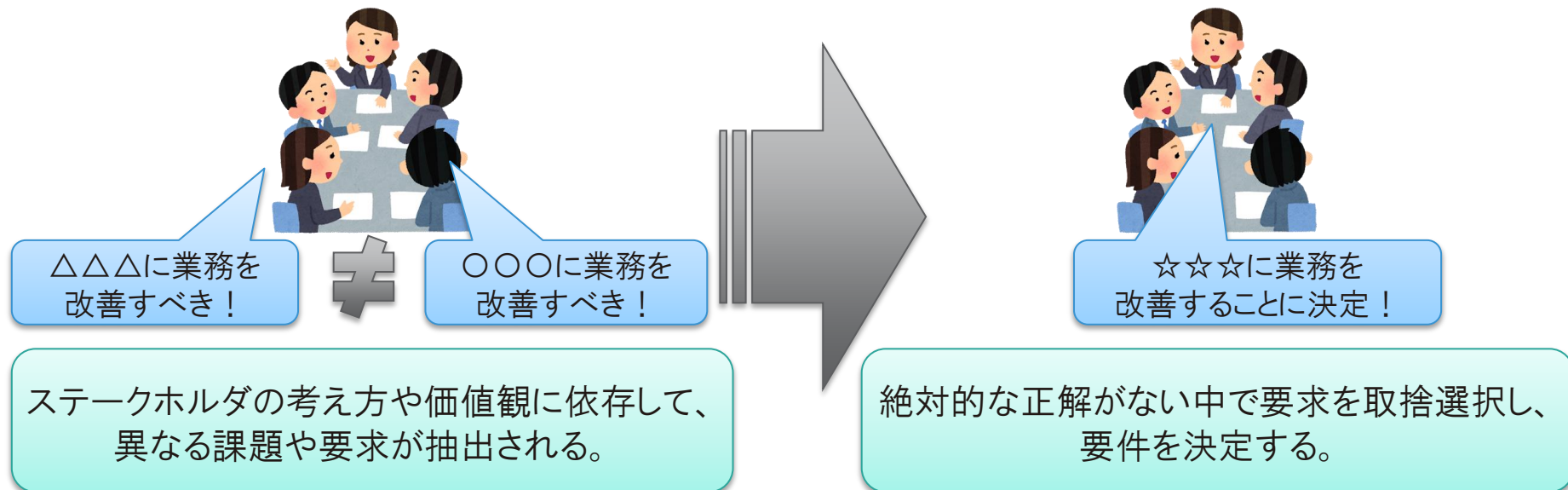
11. 要件の品質を評価できず、適切な工程終了判断が下せない。
12. 読み手依存で解釈がブレるなど、要件の曖昧さがあり、後続工程で手戻りが発生する。
13. 設計開始後も、要件定義相当の課題発生と検討がズルズル続き、設計コストが増加する。
14. 各要件の必要理由が不明になり、設計工程以降で要件スコープ調整の拠り所を失う。
15. 要件通り作ったが、ユーザーテストでユーザーが満足しない、使用に適さないと判断される。

要件定義の難しさ

- 正解が用意されていない(お客さまが意思決定する)
- ステークホルダ間で、ギャップや認識齟齬、解釈の違いが発生しやすい
- 開始時のインプットの品揃えや状態が、プロジェクト(お客さま)ごとに異なる
- お客さまを積極的にリードしないと、うまく進まない
- お客さまの主体的取り組みが不足する

要件定義の難しさ①

■ 正解が用意されていない(お客さまが意思決定する)

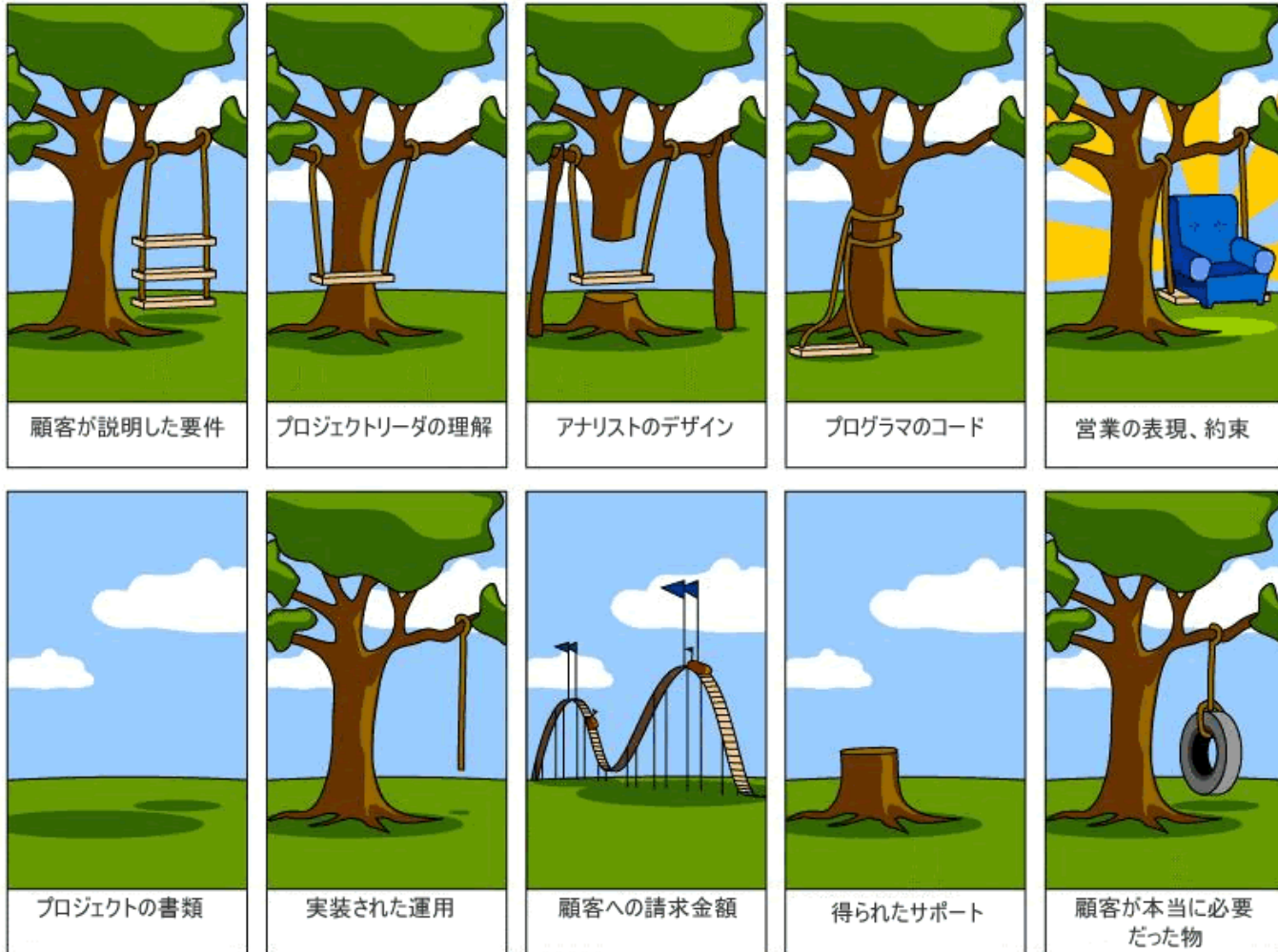


「お客さまが過不足なく正しい要求を持っている」とは考えない

誰が、何を以って、どう意思決定するかを決める

要件定義の難しさ②

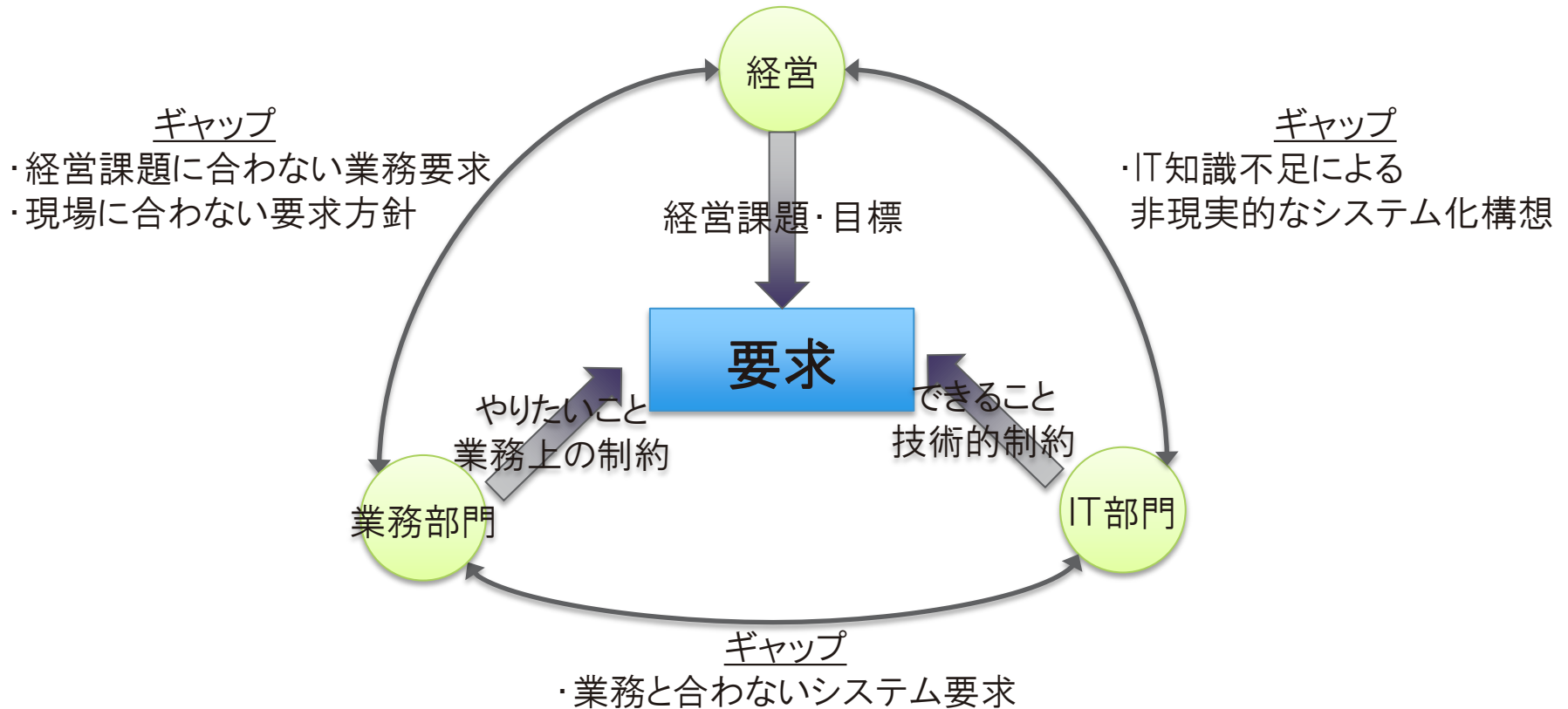
- ステークホルダ間で、ギャップや認識齟齬、解釈の違いが発生しやすい



"I know that's what I said, but it's not what I meant!" by CNX under CC BY 4.0

要件定義の難しさ②

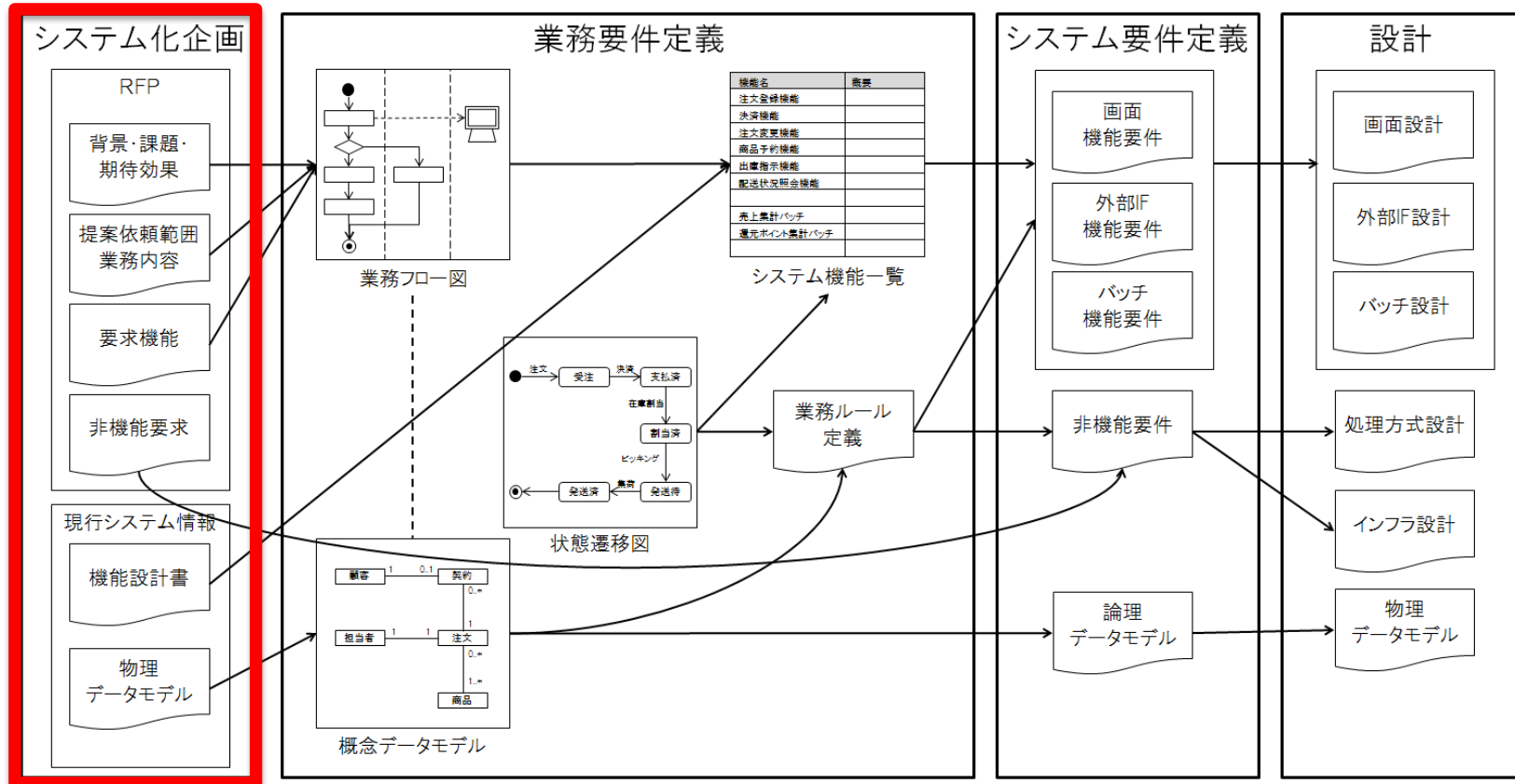
- ステークホルダ間で、ギャップや認識齟齬、解釈の違いが発生しやすい



ステークホルダ間でゴールを共有し、ゴールと要求のトレーサビリティを確保する。

要件定義の難しさ③

■ 開始時のインプットの品揃えや状態が、お客さまやプロジェクトごとに異なる



インプット情報の充足度、品質に合わせて、「進め方」「成果物」を決める

要件定義の難しさ④

■ お客さまを積極的にリードしないと、うまく進まない

進 捗

- ・不確定要素が多く、進めながら作業内容、実行計画を見直す必要がある
- ・お客さま、自社ともに、現業優先の参画者が多いと、順調に進捗しない
- ・ステークホルダ間での利害対立調整が進捗を阻害する

規 模

- ・要件定義方針が不明確だとスコープが拡大し、ユーザの要求が膨らむ
- ・お客さまは、要件確定より費用確定を優先する傾向にある
- ・ユーザ要求を単純に積み上げてしまうと規模が膨れる

品 質

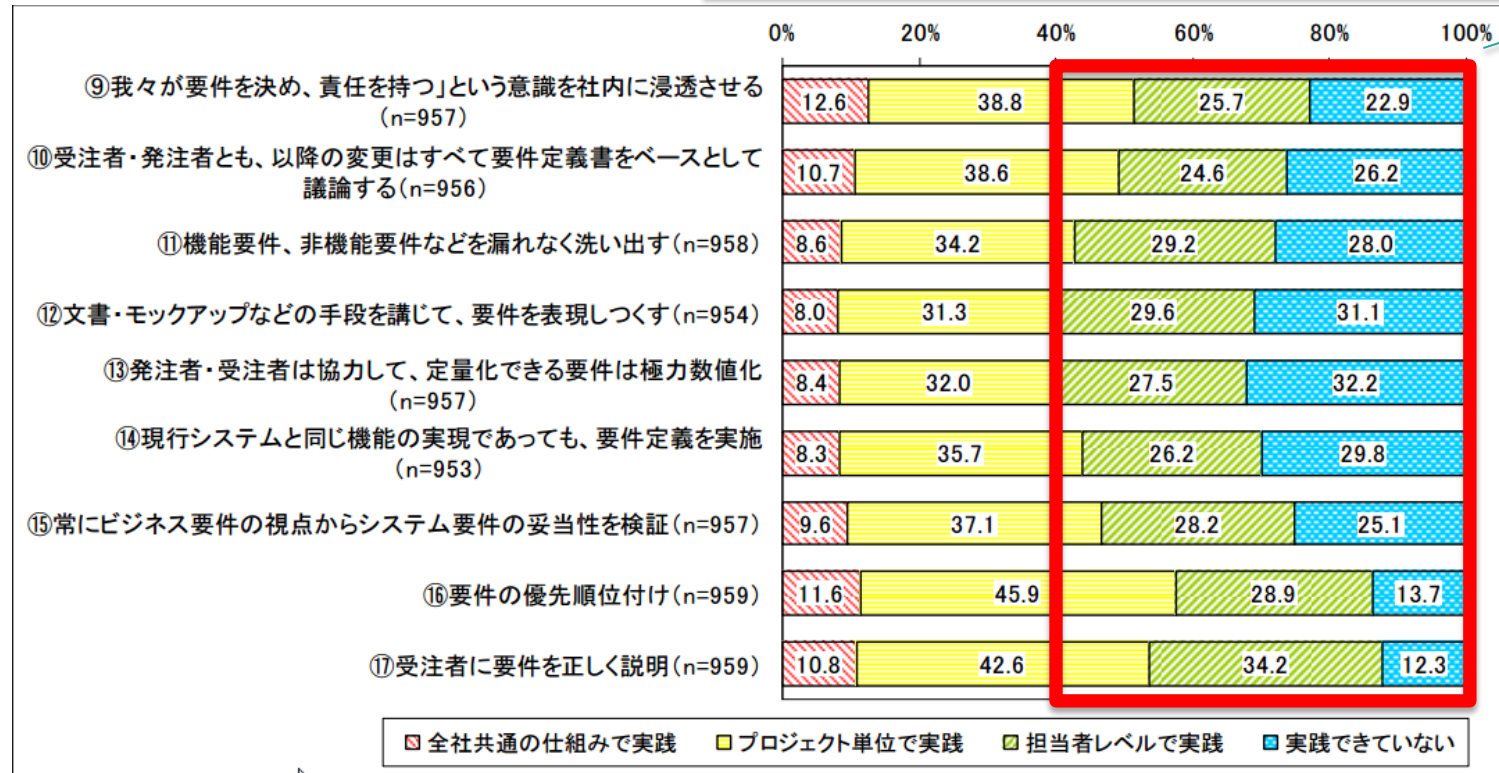
- ・他工程に比べると、品質が注視される度合いが低い
- ・要件品質を確保するために確認すべき観点が不明確
- ・要件定義の品質評価項目設定や測定方法が不明確

お客さまに内容を理解頂いた上で「要件定義計画」を合意し、状況に合わせて見直す

要件定義の難しさ⑤

■ お客さまの主体的取り組みが不足する

60%弱のユーザー企業が要件定義に十分に取り組めていない



※ JUAS「企業IT動向調査2012報告書」から引用

ステークホルダ分析と根回し・準備を行い、必要なキーマンを巻き込む

要件定義の難しさ⑤

■ お客さまの主体的取り組みが不足する

- 原理原則【1】ユーザとベンダの想いは相反する
- 原理原則【2】取り決めは合意と承認によって成り立つ
- 原理原則【3】プロジェクトの成否を左右する要求確定の先送りは厳禁である
- 原理原則【4】ステークホルダ間の合意を得ないまま、次工程に入らない
- 原理原則【5】多段階の見積もりは双方のリスクを低減する
- 原理原則【6】システム化実現の費用はソフトウェア開発だけではない
- 原理原則【7】ライフサイクルコストを重視する
- 原理原則【8】システム化の方針・狙いの周知徹底が成功の鍵となる
- 原理原則【9】要求定義は発注者の責任である
- 原理原則【10】要求定義書はバイブルであり、事あらばここに立ち返るもの
- 原理原則【11】優れた要求定義書とはシステム開発を精緻にあらわしたもの
- 原理原則【12】表現されない要求はシステムとして実現されない
- 原理原則【13】数値化されない要求は人によって基準が異なる
- 原理原則【14】「今と同じ」という要求定義はありえない
- 原理原則【15】要求定義は「使える」業務システムを定義すること
- 原理原則【16】機能要求は膨張する。コスト、納期が抑制する
- 原理原則【17】要求定義は説明責任を伴う

※ 超上流工程の『原理原則17ヶ条』 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～」から引用

要件定義概論

1. 要件とは？
2. 要件定義で何が起こっているか？
3. 要件定義の概念プロセス、成果物
4. 要件定義の基礎知識
5. 特定種類のPJでの要件定義

火事

か

確認

じ

準備

歌舞伎

か

獲得

ぶ

分析

き

記述

消せ火

け

検証

せ

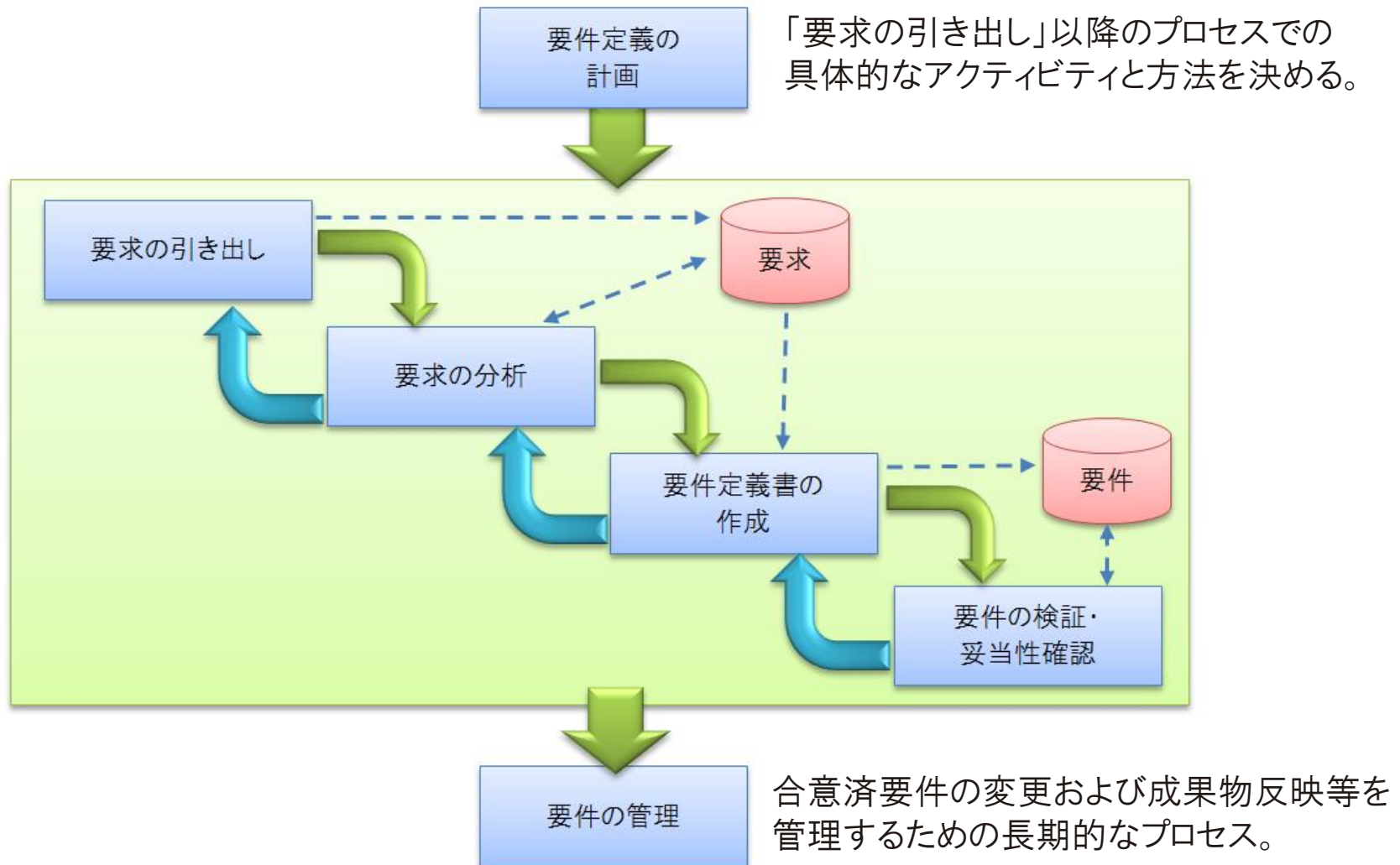
設定

ひ

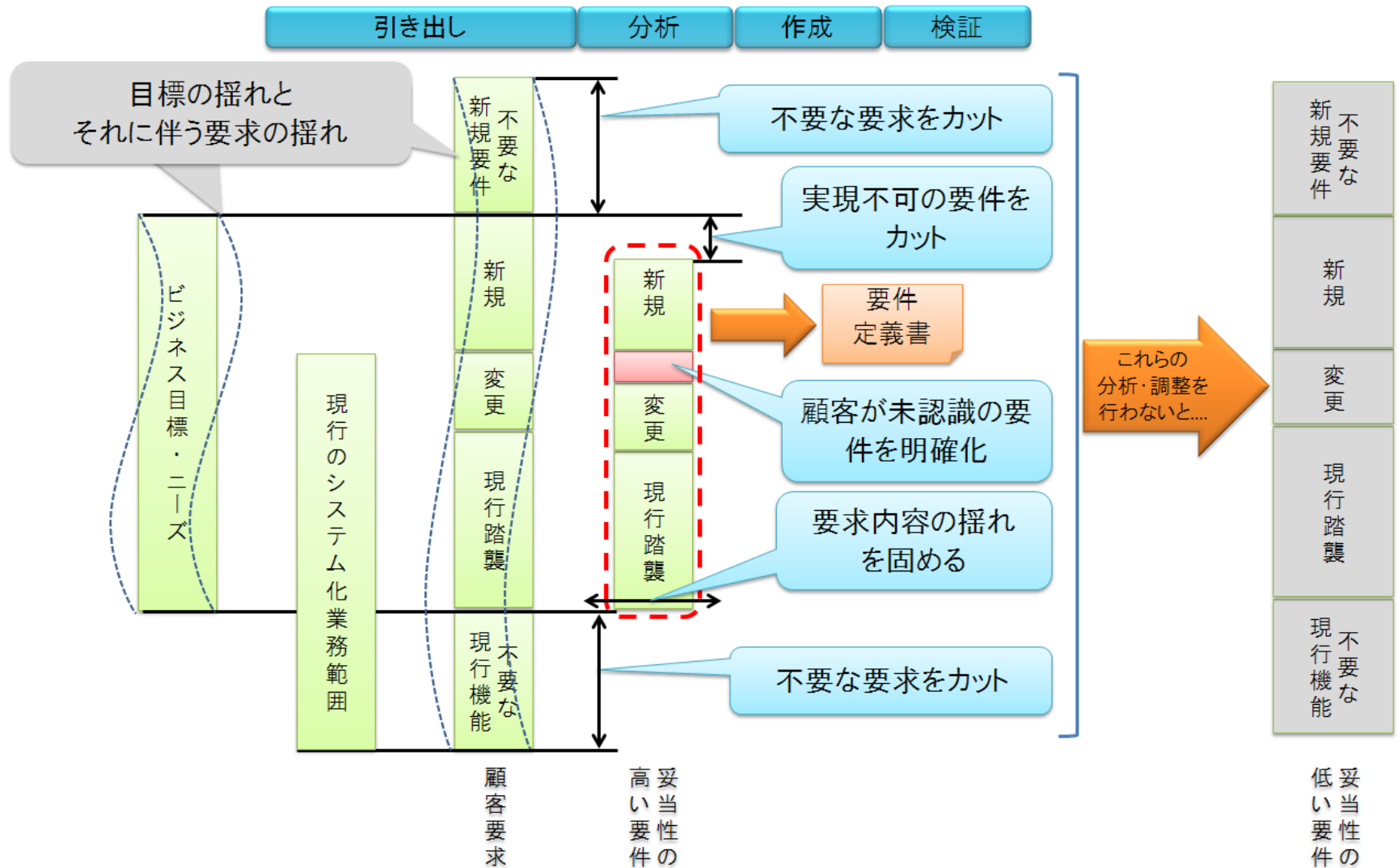
引継

要件定義の概念プロセス

■ 6つの要件定義プロセス領域

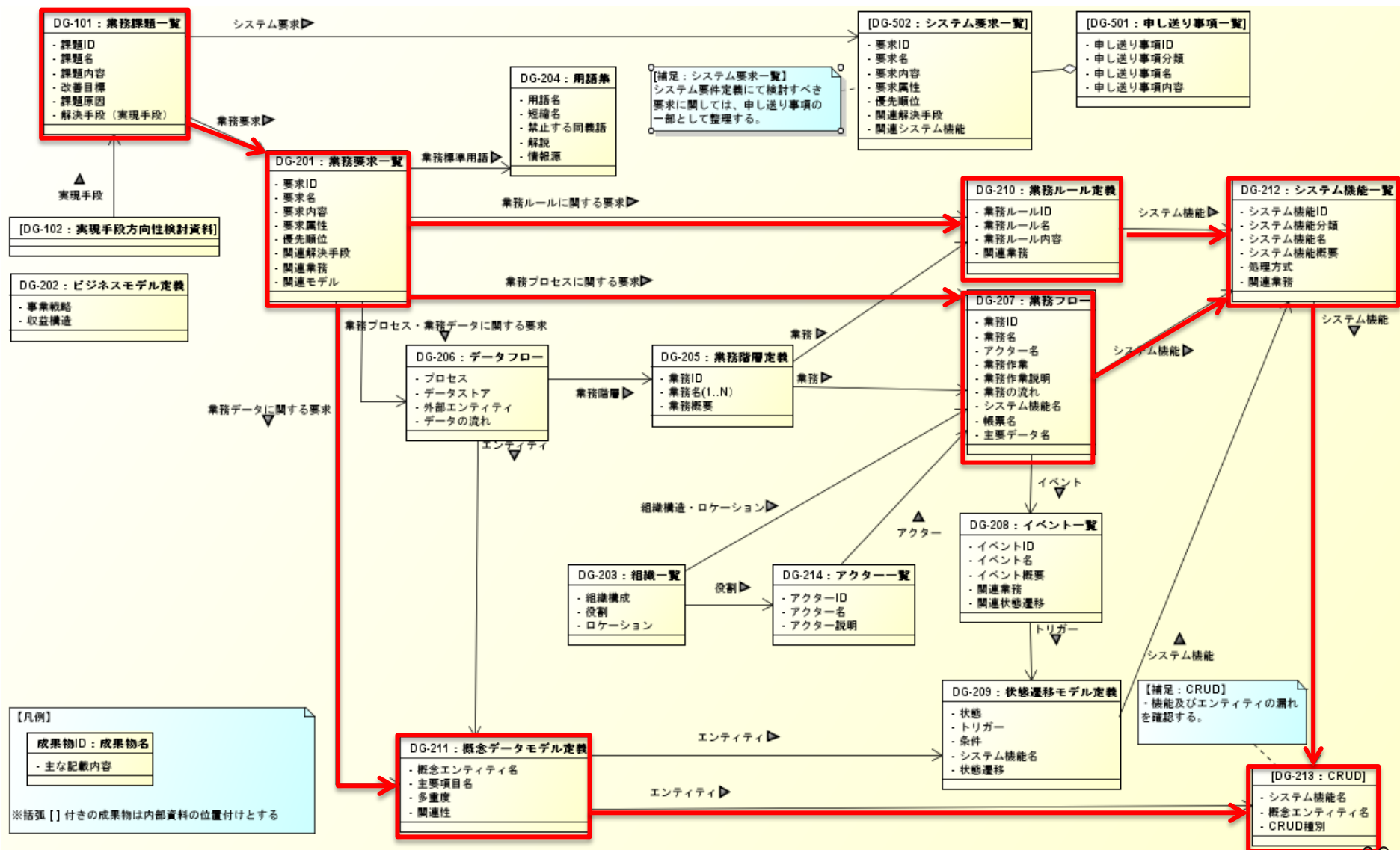


プロセス進行に伴う要求の変化



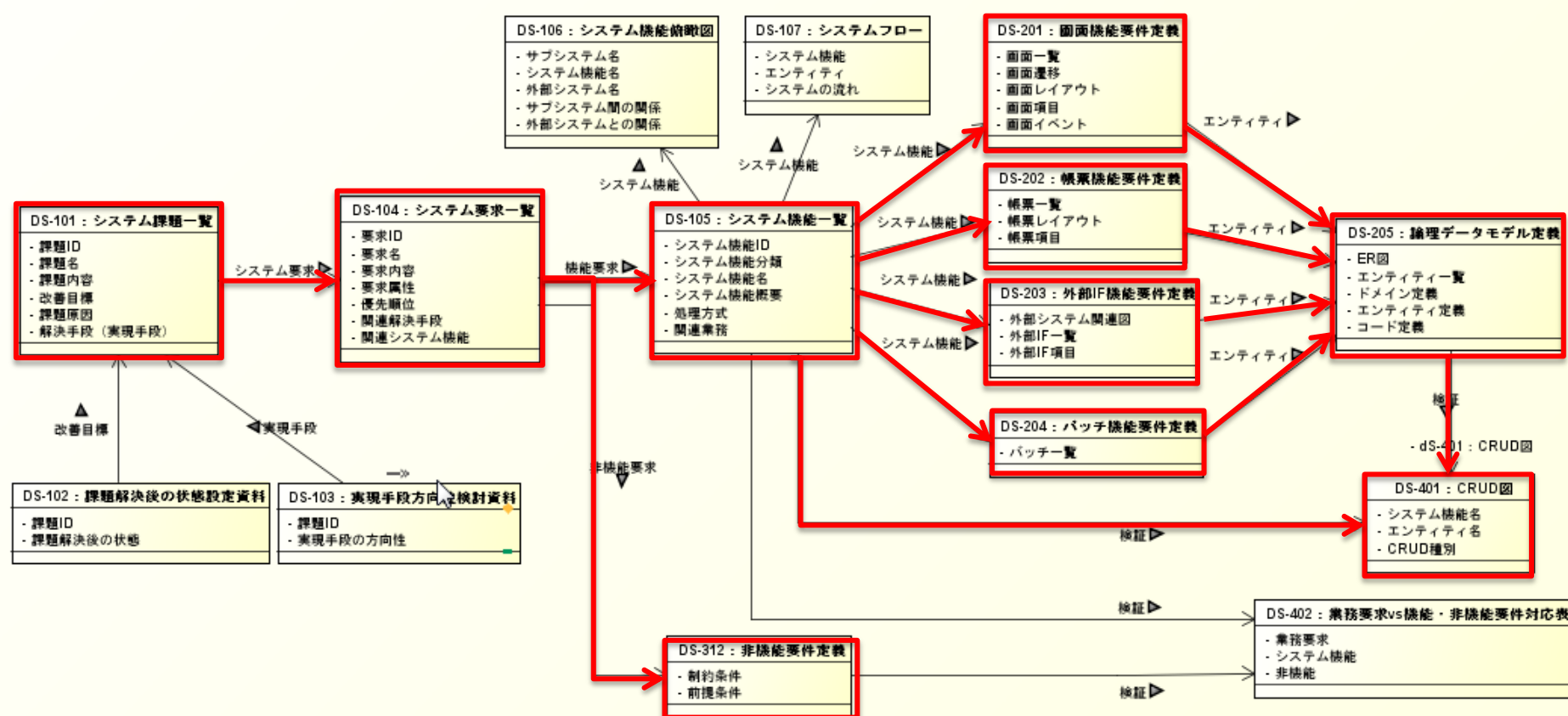
業務要件定義の成果物(『要件定義フレームワーク』に基づく作成例)

(要件定義フレームワーク: <https://fintan.jp/?p=233>)



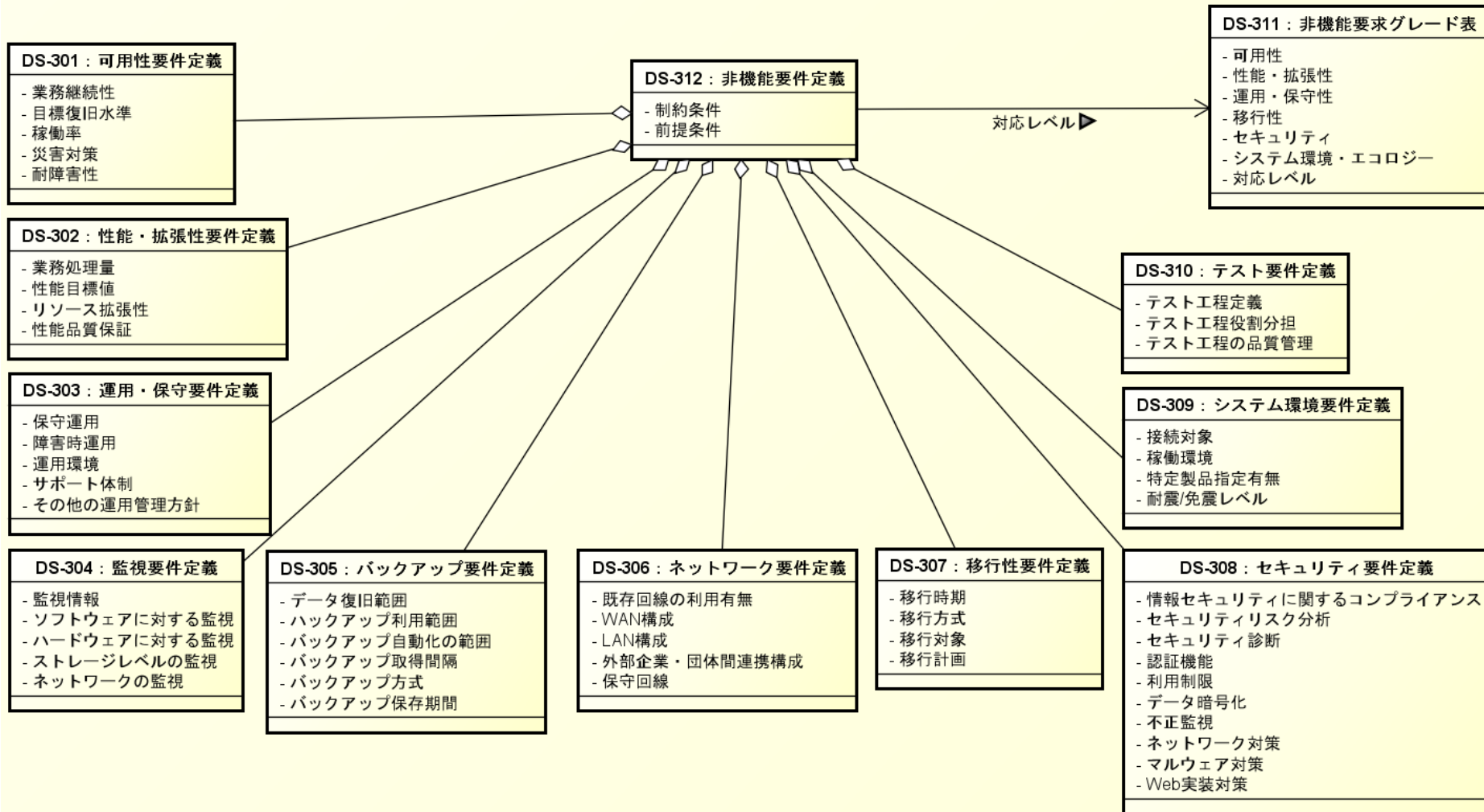
システム要件定義の成果物①(『要件定義フレームワーク』に基づく作成例)

■ 機能要件定義



システム要件定義の成果物②(『要件定義フレームワーク』に基づく作成例)

■ 非機能要件定義



要件定義概論

1. 要件とは？
2. 要件定義で何が起きているか？
3. 要件定義の概念プロセス、成果物
4. 要件定義の基礎知識

要件のスコープ①

要件は階層構造を持つ。
業務要件からシステム要件が導かれる。

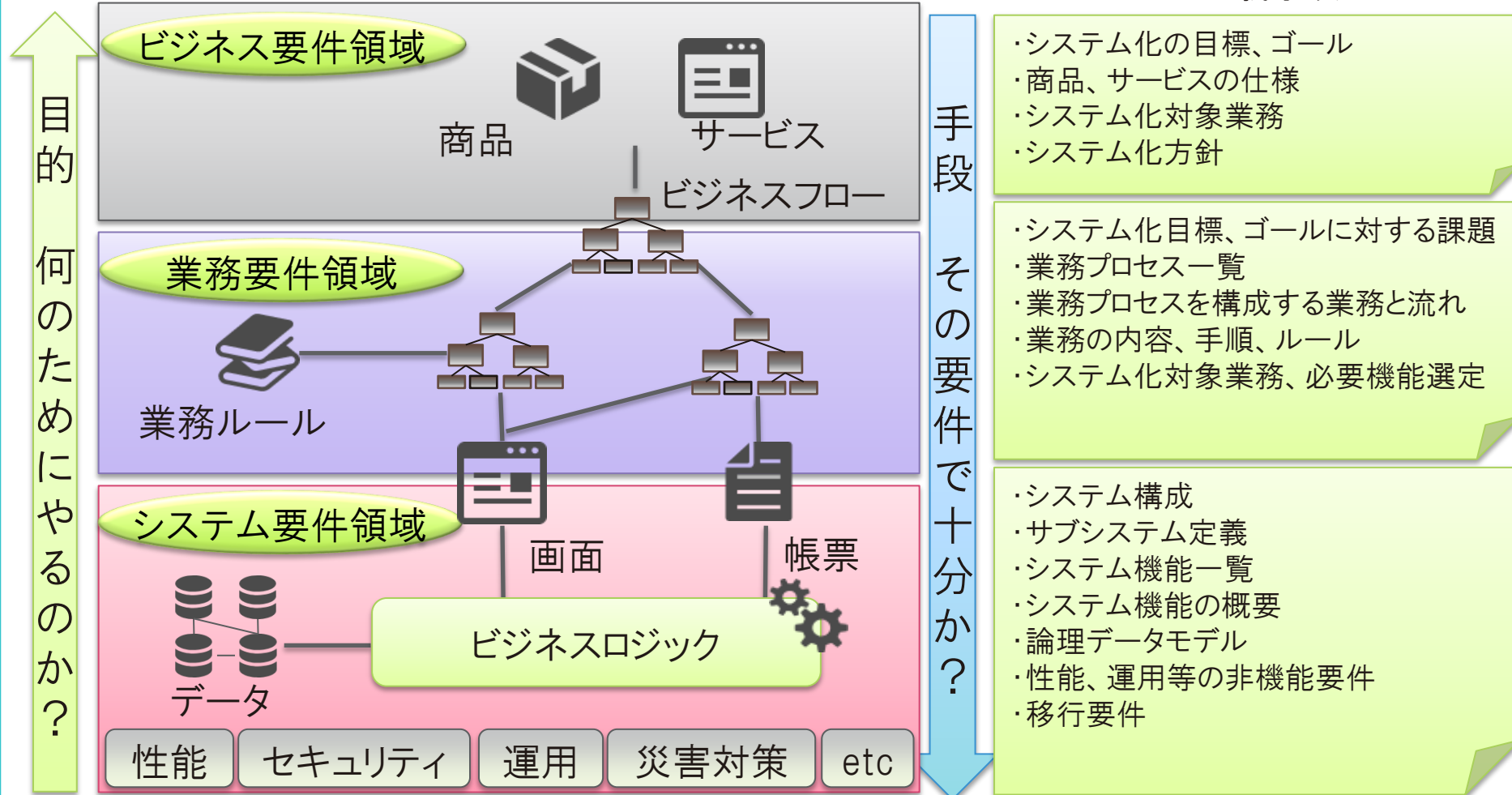
要件スコープ	対象領域	内容
ビジネス要件	お客さまの 経営、ビジネス	<ul style="list-style-type: none">• 企業や組織が果たすべきミッション、目標、ゴール• 企業が提供する商品やサービス
業務要件	お客さまの 業務機能	<ul style="list-style-type: none">• ビジネス要件を前提として、業務プロセスや概念構造など複数の視点から分析されたお客さま業務。• お客さま業務を支援するシステム機能の品揃え
システム要件	業務で活用する システム機能	<ul style="list-style-type: none">• 業務要件を前提として、外部設計以降の設計、実装の工数を見積もれる粒度を目安として具体化、詳細化されたシステム機能、非機能

要件のスコープ②

- ビジネス要件、業務要件、システム要件にトレーサビリティを持たせる。

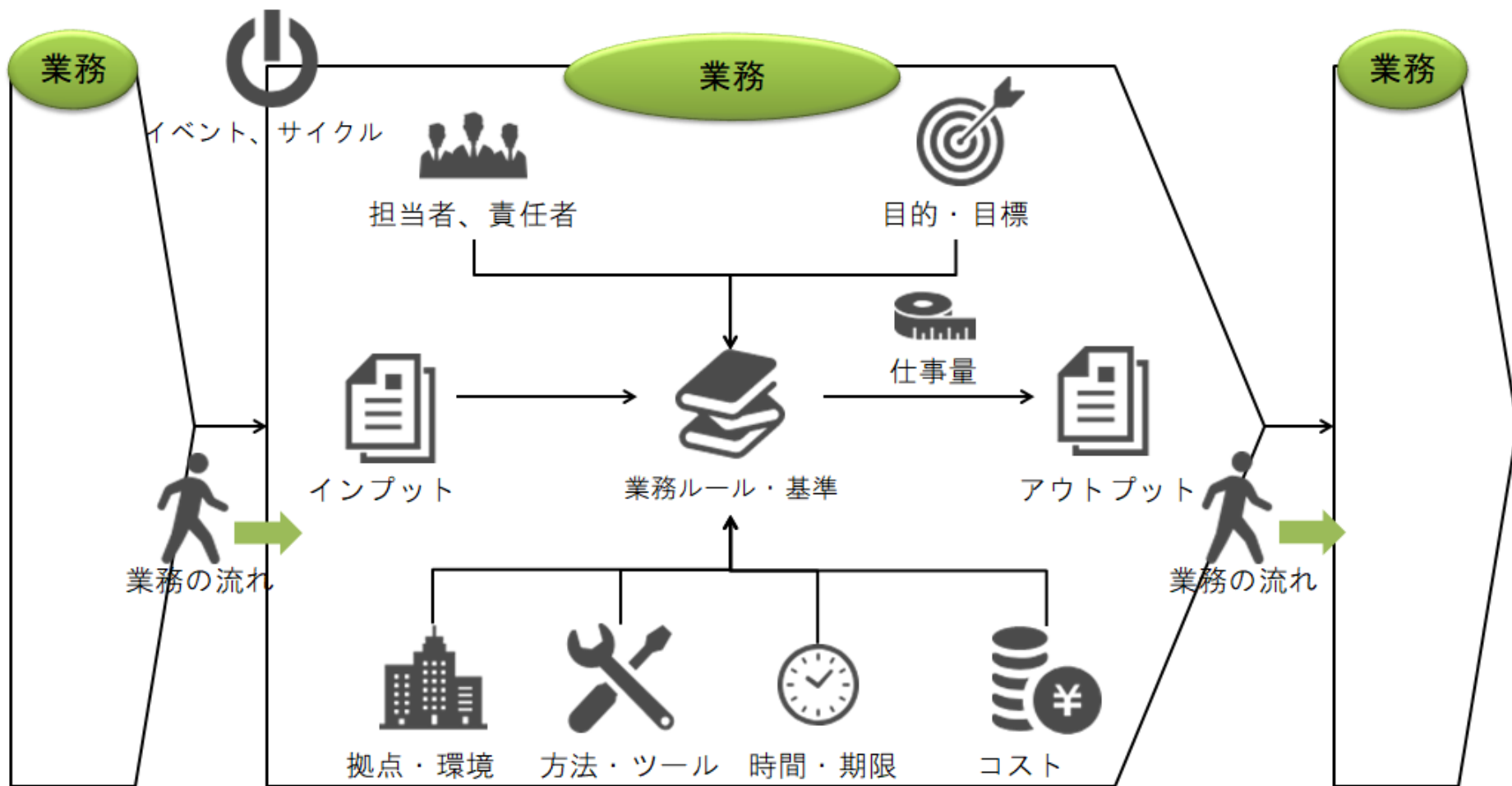
＜各領域の範囲＞

＜主な定義事項＞



要件のスコープ③

- 業務要件を構成する要素 ⇨ あるべき業務として明らかにすべき要素



要件の分類①

- 非機能要件は、ユーザーの関心が部分的で、要求が曖昧になりやすい。

分類	説明	例
機能要件	利用者が目的を遂げるためにITシステムが提供するサービス。	前日に発生したクレジットカード利用実績をカード番号単位で集計し、〇〇センターへ電送する。
非機能要件	ITシステムの機能要件に付随して必要となる品質要件や制約事項。	クレジットカード利用実績集計は、3万件/日の実績データを00:00から01:00の間で集計する。 ※非機能要求グレードのサブ項目「業務処理量」観点の要件

業務特性に応じた適切な非機能要件を定義するには、以下が欠かせない。

- 非機能要件に含まれる要素の体系的な理解
- お客さまをリードして具体化する

要件の分類②

- 網羅的な非機能要件定義には、「非機能要求グレード」等のフレームワークが有効

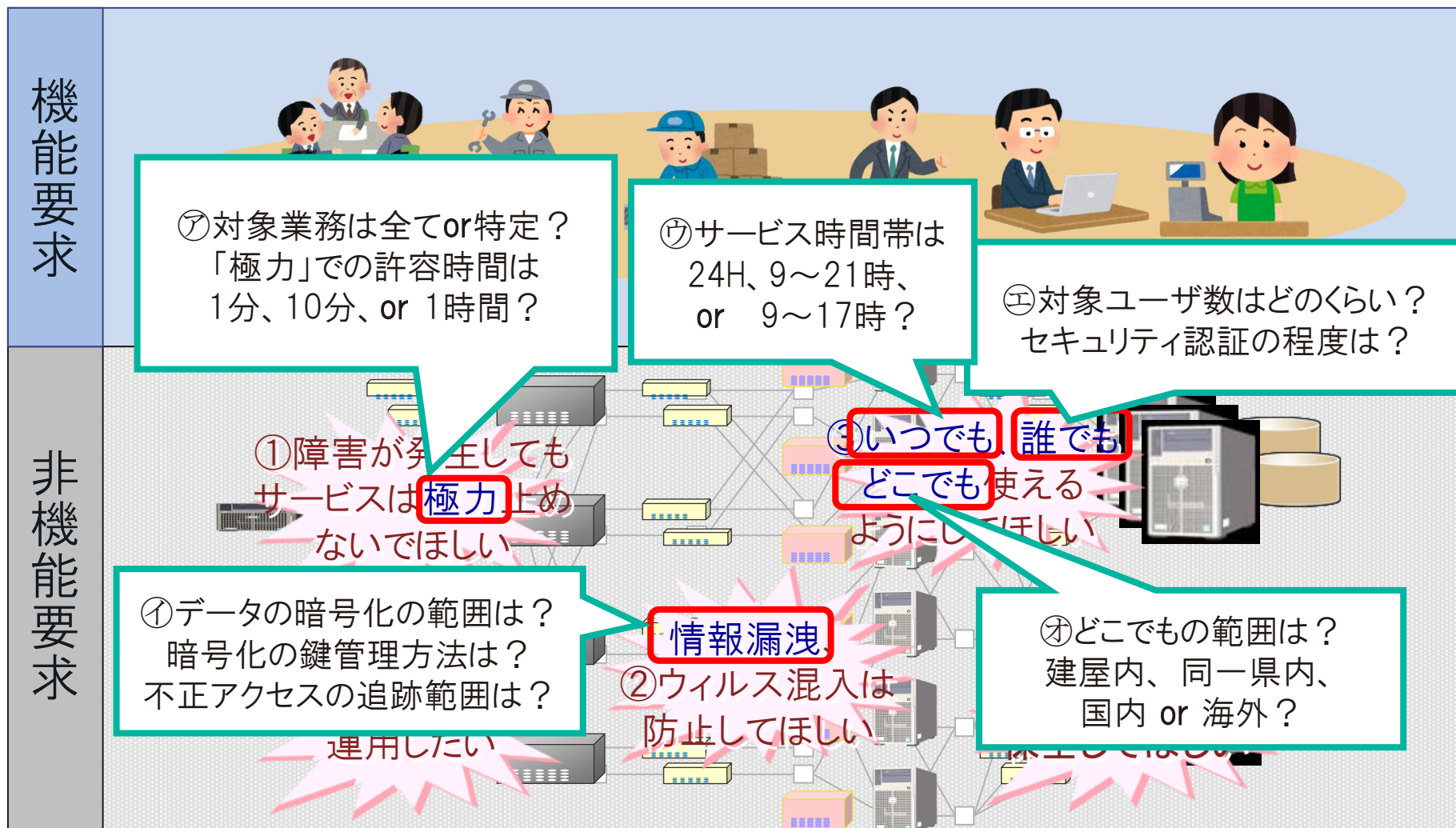
大項目	説明	例
可用性	システムサービスを継続的に利用可能とする要求	運用スケジュール(稼働時間・停止予定など) 障害、災害時における可動目標
性能・拡張性	システムの性能および将来のシステム拡張に関する要求	業務量および今後の増加見積 システム化対象業務の特性(通常時/ピーク時)
運用・保守性	システムの運用と保守サービスに関する要求	運用中に求められるシステム稼働レベル 問題発生時の対応レベル
移行性	現行システム資産の移行に関する要求	新システムへの移行期間、移行方法 移行対象資産の種類および量
セキュリティ	情報システムの安全性の確保に関する要求	利用制限 不正アクセスの防止
システム環境・エコロジー	システムの設置環境やエコロジーに関する要求	耐震/免震、温度/湿度などのシステム環境 CO2排出量などのエコロジー関連

IPA/SEC 非機能要求グレード

[:https://www.ipa.go.jp/sec/softwareengineering/std/ent03-b.html](https://www.ipa.go.jp/sec/softwareengineering/std/ent03-b.html)

要件の分類③

■ 非機能要求は様々で曖昧



要求の状態①

- 要件の品質を高めるには、3つの「要求の状態」を意識することが重要

状態	説明
表明された要求	文書や言葉で表明された要求
示唆された要求 (暗黙の要求)	既存業務・システムや社会・業界・企業内の文化、常識、習慣から、そうであるべきと考えられる要求
<u>認識されていない要求</u>	<u>お客さまが気づかない、認識していない要求</u>

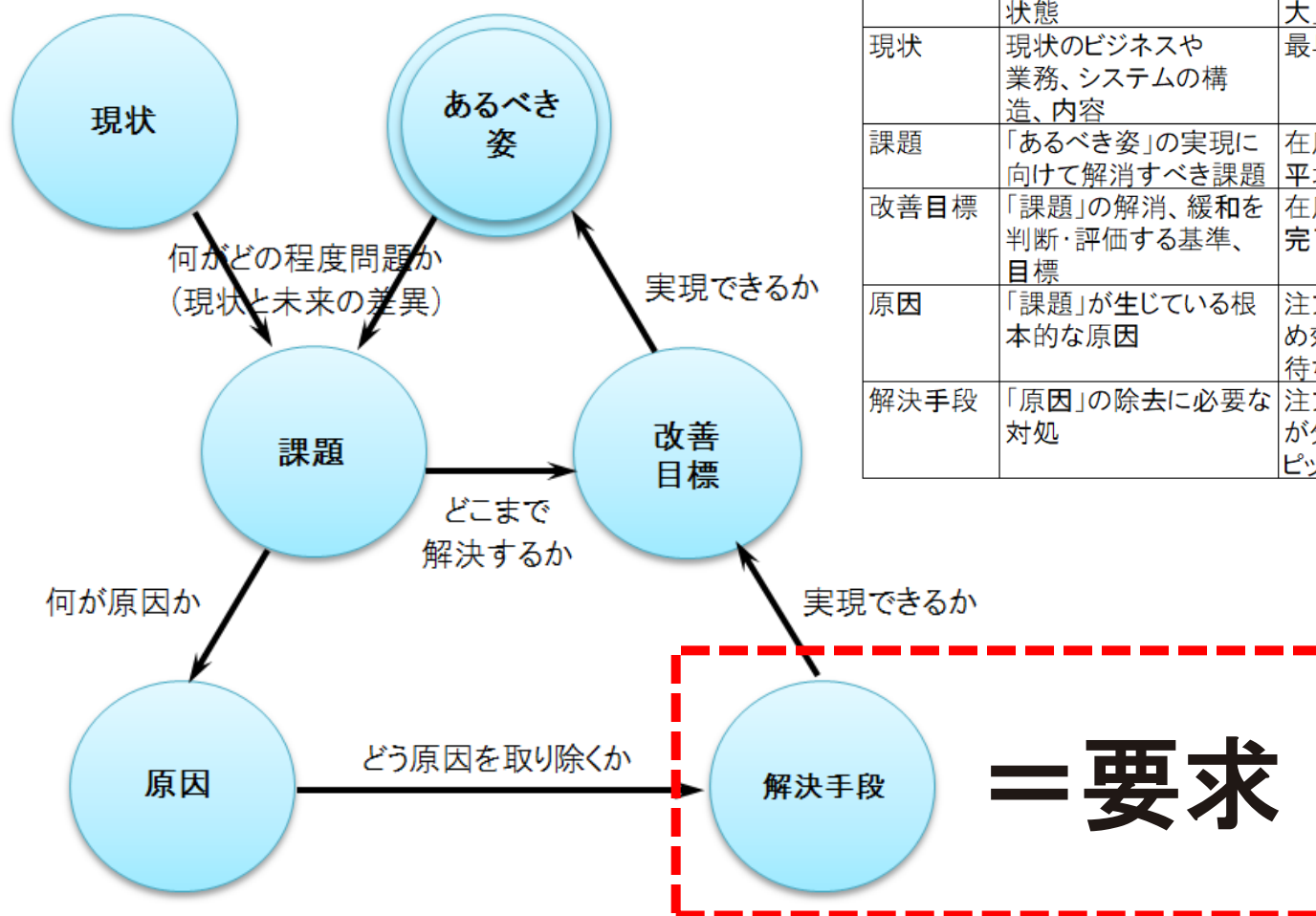
「表明された要求」を集約、文書化する要件定義は失敗する。(＝御用聞き)
表明された要求だけでは、完全性、実現性等に不備を残すことが多く、
要求を「引き出す」「分析する」アプローチによる精錬が欠かせない。

要求は始めからあるものではなく開発する ＝ 要求開発

要求の状態②

■ 要求抽出の抜け漏れを抑制する考え方

「要求の概念構造」



概念	説明	例
あるべき姿	プロジェクトゴール達成に向けて実現すべき状態	購入商品がお客さまに即日配送される(プロジェクトゴールは「リピート顧客の拡大」)
現状	現状のビジネスや業務、システムの構造、内容	最早でも翌日配送
課題	「あるべき姿」の実現に向けて解消すべき課題	在庫割当からピッキング開始までに平均2時間の待ち状態が発生している
改善目標	「課題」の解消、緩和を判断・評価する基準、目標	在庫割当から30分以内にピッキングを完了する
原因	「課題」が生じている根本的な原因	注文単位に担当1名がピッキングするため効率が悪化し、ピッキング担当の空き待ちが発生している
解決手段	「原因」の除去に必要な対処	注文1件を商品分類別のピッキング担当が分担する方式に変更し、1件あたりのピッキング所要時間を短縮する。

要件の属性

- 要件の属性情報を明確にすることで要件を補強し、要件を管理する。

No	要求属性の例	要求属性値の例
1	要求スコープ	業務要求・システム要求・移行要求・運用要求など
2	対象業務分類	事業名・組織機能名など
3	要求の分類	機能要求・非機能要求など
4	重要度	高・中・低（業務の継続性や事業価値への貢献で設定）
5	緊急度	高・中・低など
6	難易度	高・中・低など（既存ソースのSLOCなどで設定）
7	使用頻度・実行頻度	XX回/月など
8	要求内容の安定度（確度）	高・中・低など（要求変更リスクレベルなどで設定）
9	所要コスト	XX人月など
10	メリット/デメリット	削減運用費用、増加売上額、業務プロセスの複雑化
11	リスク	ビジネス上のリスクと利用技術の持つリスクなどで設定
12	オーナー、利害関係者	XX部XX担当など
13	要求依存関係	依存する要求IDなど
14	関連業務	XX業務など

要件の特性①

- 検証の観点として、要件の特性を活用する。

検証:
(Verification)

正しく、適切な手段であること

妥当性確認:
(Validation)

目的・目標を達成できること

要件の特性②

■ 検証の観点として、要件の特性を活用する。

特性	説明	欠陥の例
単一性	要件の対象がひとつであること。	特定システム機能要件の説明内で、他のシステム機能要件に触れている。
完全性	要件および要件説明に抜け漏れがなく、顧客のニーズを満たすこと。	異常時の業務フローが定義されていない。
一貫性	複数の要件間、上位要件間および要件全体に矛盾がないこと。	一方の要件実現により、他方の要件が実現できない。
法令遵守	法律や規制に準拠していること。	業界の法令に反する顧客要求事項をそのまま要件として定義している。
独立性	要件内容が不適切または暗黙の認識を前提としていないこと	システム機能要件が、理由なく特定のミドルウェアに依存した内容になっている。
追跡可能性	前後の工程で定義した要件、設計との関連性が明確であること。	ビジネス目標・ゴールと業務要件の関連が確認できない。 システム機能要件と業務プロセスの関連が確認できない。
最新性	最新の条件を元に行っていること。	現行システム仕様の古いベースラインを元に要件定義されている。
実現可能性	要件が各種制約等のもとで、実現可能であること。	実務で運用不可能な業務フローが定義されている。 技術的に実現不可能な機能要件が定義されている。
無曖昧性	複数解釈が成立する曖昧さがなく、理解できること。	未定義の用語が要件説明に使用されている。 要件定義文内の修飾語の対象が不明確。
必要性	要件が必要とされる理由が明確であり、過剰でないこと。	システム機能要件を必要としている業務が特定できない。
検証可能性	要件が実現されたこと、効果を検証および測定が可能であること。	検証の可否判定観点や基準が明確に定義されていない。

まとめ

トピック	内容
要件とは？	<ul style="list-style-type: none">問題解決、目標達成に欠かせない条件効果とその手段としての業務とシステムを決めること
要件定義の難しさ	<ul style="list-style-type: none">「答えが用意されていない」「お客さまの主体性欠如」「インプット不足」「認識齟齬」「積極的なコントロール」
要件定義の概念プロセス	<ul style="list-style-type: none">“火事 歌舞伎 消せ火”で要求を洗練する
要件のスコープ	<ul style="list-style-type: none">ビジネス要件・業務要件・システム要件トレーサビリティを確保
要件の分類	<ul style="list-style-type: none">機能要件・非機能要件
要求の状態	<ul style="list-style-type: none">「明示された」「示唆された」「認識されていない」要求は始めからあるものでなく、開発する
要件の属性	<ul style="list-style-type: none">分類、重要度、緊急度、メリット/デメリット、オーナー
検証と妥当性確認	<ul style="list-style-type: none">「実現性」「無曖昧性」「必要性」等で正しさを確認上位要件や目的・目標に対する効果を確認