

# 業務要件定義プロセスガイド

第 1.10 版

2018 年 08 月 29 日

TIS 株式会社



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。

## 變更履歷

[illegible]

## 目次

G1 業務要求の収集.....	1
G1-01 現行業務の調査.....	1
G1-01-01 現行業務プロセスの調査.....	2
G1-01-02 現行業務ルール of 調査.....	6
G1-01-03 現行業務データの調査.....	8
G1-02 現行システムの調査.....	9
G1-02-01 現行システムの調査.....	10
G1-03 課題の抽出と原因分析.....	13
G1-03-01 システム化企画資料からの課題抽出.....	14
G1-03-02 ステークホルダーからの課題抽出.....	15
G1-03-03 課題の原因分析.....	16
G1-04 課題解決後のゴール定義.....	17
G1-04-01 課題解決後のゴール定義.....	18
G1-05 課題解決の実現手段検討.....	19
G1-05-01 実現手段の方向性検討.....	20
G1-05-02 実現手段の定義.....	21
G2 業務要求の整理とモデル化.....	22
G2-01 業務要求の整理.....	22
G2-01-01 業務要求のリストアップと整理.....	23
G2-01-02 業務要求の確認.....	25
G2-02 業務要求のモデル化.....	27
G2-02-01 業務要求のモデル化方法の見直し.....	28
G2-02-02 業務プロセスのモデル化.....	29
G2-02-03 業務ルールのモデル化.....	31
G2-02-04 業務データのモデル化.....	32
G2-02-05 システム機能のリストアップ.....	33
G2-03 業務要求の優先順位付け.....	35
G2-03-01 業務要求の優先順位付け基準・方法の見直し.....	36
G2-03-02 業務要求の優先順位付け.....	37
G2-04 業務要求の実現対象決定.....	38
G2-04-01 業務要求の実現対象決定.....	39
G3 業務要件定義書の作成.....	40
G3-01 業務要件定義書の作成.....	40
G3-01-01 業務要件定義書の作成.....	41
G4 業務要件の検証・妥当性確認.....	43
G4-01 業務要件の検証・妥当性確認.....	43
G4-01-01 検証・妥当性確認の基準・方法の見直し.....	44
G4-01-02 業務要件の検証.....	45
G4-01-03 業務要件の妥当性確認.....	47
G4-02 業務要件のお客さま合意と承認.....	48
G4-02-01 業務要件のお客さま合意と承認.....	49
G5 引き継ぎ.....	51
G5-01 システム要件定義への引き継ぎ.....	51
G5-01-01 システム要件定義への引き継ぎ準備.....	52
G5-01-02 システム要件定義への引き継ぎ.....	53

### 【添付資料】

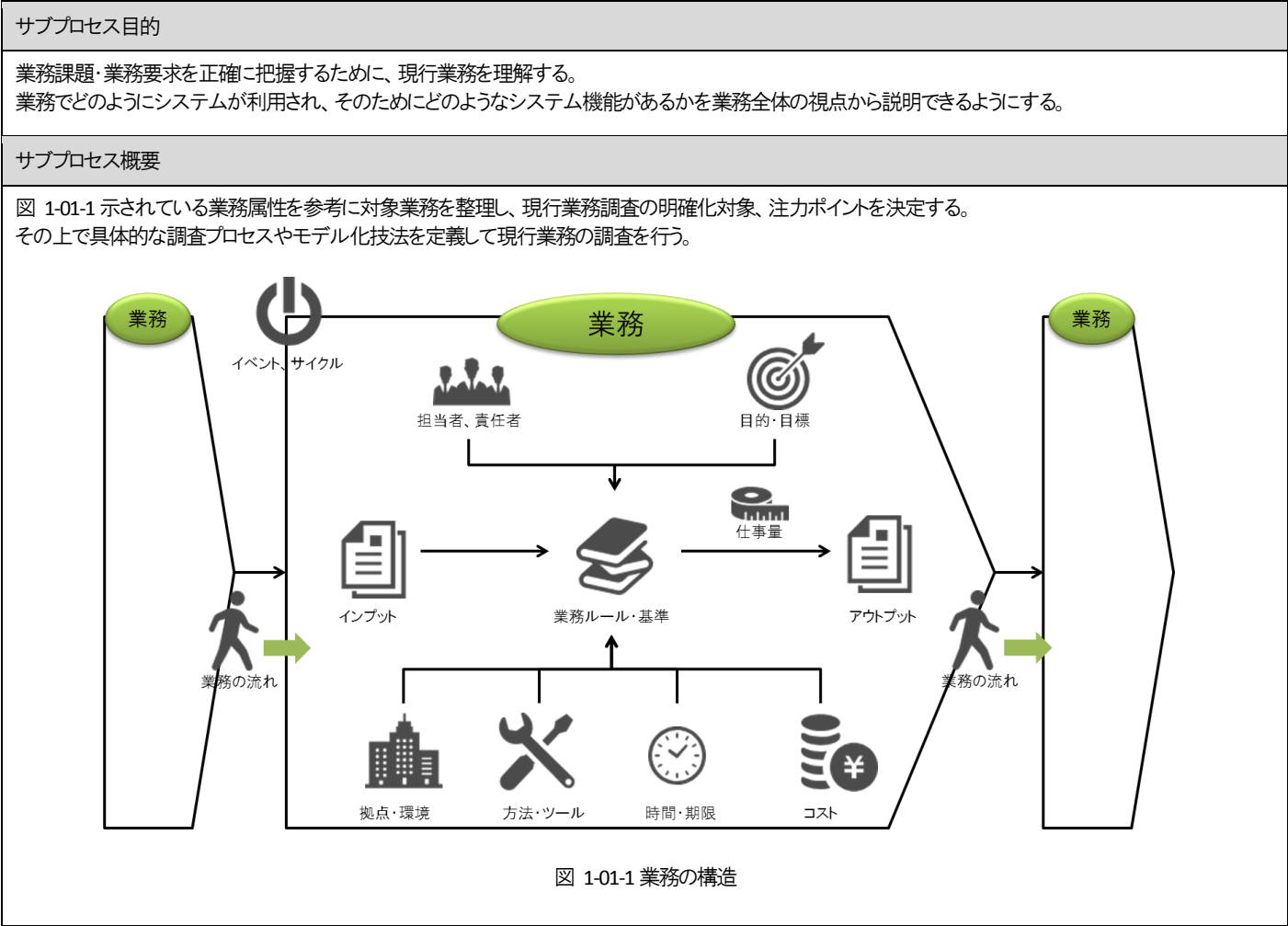
1. 業務要件定義プロセス一覧
2. 業務要件定義プロセスフロー

### 【留意事項】

要件定義フレームワークは、お客さまとベンダーの協働で要件定義を実施することを想定し、ベンダーの視点から要件定義の進め方をガイドしています。この場合、お客さまは要件定義の主体者として課題や要求の抽出、要件を決定する立場となり、ベンダーはそれらの活動を支援する立場となります。ユーザー企業自身で要件定義を実施する場合は、要件定義フレームワークに記述された「お客さま」を「自社」に、「ベンダー」をユーザー企業内で要件定義推進を担う組織・人(例えば、情報システム部門)に読み替えてご利用ください。

G1 業務要求の収集

G1-01 現行業務の調査



## G1-01-01 現行業務プロセスの調査

アクティビティ概要	
業務構造、業務の流れ、業務の依存関係などを理解するために、業務プロセスの視点で現行業務(As-Is)を調査する。	
インプット	アウトプット
<ul style="list-style-type: none"> <li>● 現行業務の資料(As-Is) ※組織構成図、組織一覧、業務用語集、業務一覧、業務マニュアル、業務フロー、業務概要書、実帳票など</li> </ul>	<ul style="list-style-type: none"> <li>● ビジネスモデル定義(As-Is)</li> <li>● 組織一覧(As-Is)</li> <li>● 用語集(As-Is)</li> <li>● 業務俯瞰図(As-Is)</li> <li>● 業務階層定義(As-Is)</li> <li>● データフロー(As-Is)</li> <li>● アクター一覧(As-Is)</li> <li>● 業務フロー(As-Is)</li> <li>● イベント一覧(As-Is)</li> <li>● 状態遷移モデル定義(As-Is)</li> </ul>
手順	
<ol style="list-style-type: none"> <li>1. ビジネスモデルの把握 プロジェクトの目的・目標および取り扱う課題をより深く理解するため、ビジネスモデルキャンバス等を利用してお客さまのビジネスモデルを可視化する。</li> <li>2. 組織構造・役割・所在の把握 業務プロセスに関連するアクターを抜け漏れなく抽出するため、組織一覧等でお客さまの組織構造・役割・所在を可視化する。</li> <li>3. 標準用語の把握 ステークホルダーと共通認識を持った用語を統一的に利用するため、用語集に業務で利用する標準用語を定義する。</li> <li>4. 業務全体の流れ・依存関係の把握 業務の全体構造を理解するため、業務俯瞰図や抽象度を上げた業務フロー等で業務全体の流れや依存関係を可視化する。</li> <li>5. 業務構造の把握 プロジェクトの対象業務範囲を明確化するため、業務階層定義で構造的に業務を可視化する。</li> <li>6. 業務作業担当者の役割(アクター)の把握 業務に関わる担当者の役割を分類し、アクター一覧で可視化する。</li> <li>7. 業務フローの把握 業務階層定義で定義された個々の業務の流れを理解するため、業務フローで可視化する。 この業務フローの中で、システム化対象となる業務作業とその業務作業で利用する画面・帳票などのシステム UI 機能を明確にする。</li> <li>8. 業務上の管理対象となる状態とその状態遷移の把握 業務上の管理対象となる状態(受注状態など)とその遷移を理解するため、状態遷移モデル定義で可視化する。</li> <li>9. 業務イベントの把握 外部とのやり取りを可視化することによる「対象業務範囲の明確化」と、イベントに対応する業務の洗い出しによる「業務階層定義の完全性向上」のため、イベント一覧等で外部から業務に対して発生するイベントや業務内で発生するイベントを可視化する。</li> </ol>	
【重要】上手に進めるためのポイント、注意事項	
全般	
<ul style="list-style-type: none"> <li>➤ 調査目的を明確にし、何を・どのレベル(粒度)で把握し、どのように可視化するのかを標準化する。 目的を明確にせず現行調査を実施すると、プログラム単位の仕様調査に偏り、業務理解が進まない可能性がある。 「現行業務の全体像を把握する」という原則のもとで、どのように可視化するのかを検討する。 また、分析担当者毎の分析レベルのばらつきを防止するために分析対象やレベルを標準化する。5W1Hの観点で検討する。 ※「基礎知識集：3-6. 要件内容を明確化するための『要件の視点』」を参照</li> <li>➤ 分析結果を可視化する手段として、モデルの活用を検討する。 可視化する手段としてモデル(図的な表現形態)を活用することで、ステークホルダー間での現行業務に対する共通理解が得易くなる。 モデル化する場合は、作業生産性・メンテナンス性などを考慮し、モデリングツールの活用も合わせて検討する。</li> <li>➤ 業務毎にお客さま側の有識者・キーマンを明確にする。</li> <li>➤ 「現行踏襲」のプロジェクトでも現行業務分析の必要性を検討する。(保守開発PJ、現行踏襲PJ) 現行保守を担当するシステムであっても業務を体系的に把握していない場合、業務分析が必要となる。 業務分析をスキップしてしまうと、後続工程で実業務と業務要件の乖離を起因とする要件変更多発につながる。 現行踏襲のプロジェクトでも現行業務分析の必要性をお客さまにご理解頂くことが重要である。</li> </ul>	

## 標準用語の把握

- 用語集は、早期にベースラインを作り、継続的に維持・改善する。  
完璧な用語集を作ることは目指さず、早期にステークホルダーと用語を共有し、ステークホルダー間の用語の認識齟齬によるコミュニケーションロス、手戻りを軽減する。

## 業務構造の把握

- 業務階層定義で構造的に業務を可視化する場合、事前に各階層の定義粒度・考え方をステークホルダーと合意しておくことが重要である。業務階層定義で業務を構造化することで、業務全体を部分に分割して理解し易くなり、抜け漏れや重複の検出が容易になる。業務階層定義を作成する際は、各階層の定義粒度・考え方(例えばレベル1:事業、レベル2:組織機能、レベル3:業務等)の定義をステークホルダーと事前に合意しておく。業務構造は、事業や組織単位の異なるステークホルダーと協議して定義する為、定義粒度が未合意の状態が進めると、粒度がバラバラで標準化されていない業務階層定義が出来上がってしまう。結果、個々の業務を単位として作業を進める場合などに、不都合が生じる。  
※各階層の定義粒度の例については、「技法ガイド(DFD 記述ガイド編):業務の可視化範囲定義」の章を参照

## 業務フローの把握

※業務フローの記述方法については、「技法ガイド(業務フロー記述ガイド編)」を参照

- 業務階層定義の階層をベースに業務フローの作成単位を定義する。  
※例えば、図 1-01-2 のように業務階層定義のレベル3単位に業務フローを作成する等

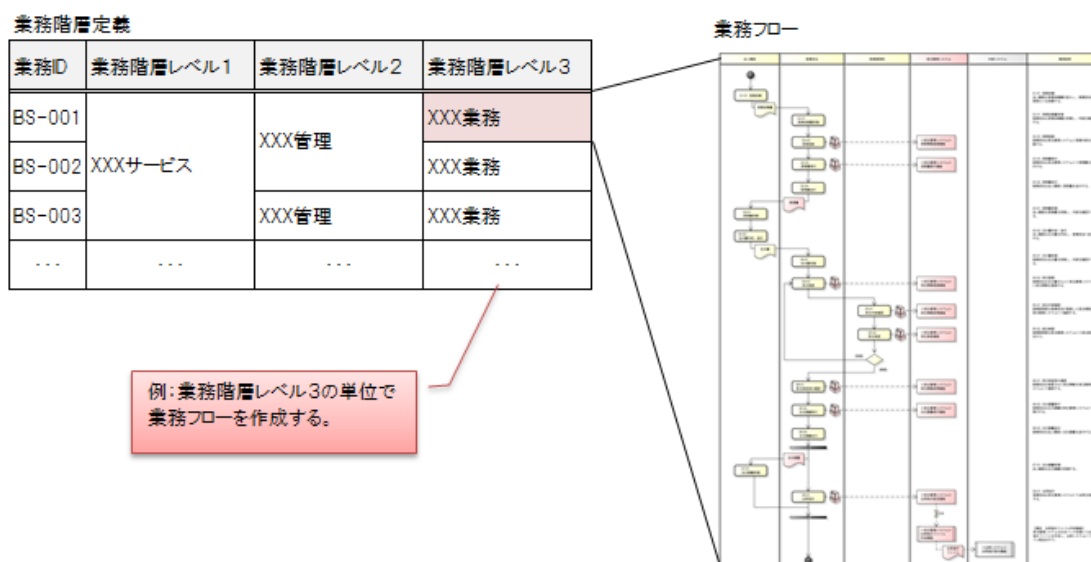


図 1-01-2 業務フロー作成単位の定義イメージ

- 現行システムが存在する場合は、業務作業で使用するシステム機能を明確にする。
- 作成した業務フローを元に業務シナリオを列挙し、業務シナリオのバリエーションが生じる要因を分析することで、業務プロセスに影響を与える前提条件やインプットなどの漏れを防止できる。
  - ✓ 業務シナリオとは、業務プロセスをインスタンス化したもので、1つの業務プロセスに対して、N個の業務シナリオが存在する。業務プロセスの前提条件やインプットに応じて、起こりうる具体的な業務のパス(1つの具体的なシーン)を指す。
  - ✓ 業務特性要素を可視化することで、業務シナリオのバリエーションが生じる要因が深しやすくなる。  
※業務特性要素の例:商品には「標準品」「特注品」がある、納品先には「海外」「国内」がある、など

## 上手く進めるためのポイント、注意事項

### 全般

- 分析作業フローを検討する。  
何をインプットとし、どのように分析を進めれば期待する内容、レベルの業務関連情報を引き出すことができるかを検討する。

その結果を分析作業フローとして可視化する。  
また、業務分析にステークホルダーの協力は不可欠であるため、業務説明・情報提供・分析結果レビュー等の依頼事項を明確にするとともに、そのタイミングに関しても分析フローの中で明確にする。

#### ビジネスモデルの把握

- 関連文書が存在しない場合には、ベンダー側の有識者(営業担当、提案担当など)に作成を支援してもらう。  
これらの有識者は提案段階でお客様のビジネスモデルをある程度把握している。

#### 標準用語の把握

- 業務メイン標準の用語集をベースに、お客さま固有の用語集を整理する組織で業務メインごとに標準用語集を整備し、お客さま固有の用語を反映することで、効率的に標準用語が定義できる。
- 要件定義成果物は、用語集に定義された用語を利用する。

#### 業務全体の流れ・依存関係の把握

- 組織機能や業務を、実行順序性でつないだ概要レベルの業務フローを書くことで、事業や組織内の業務の流れを俯瞰して理解できる。
- 組織機能や業務を、授受情報でつないだ業務俯瞰図を書くことで、業務間の依存関係を俯瞰して理解できる。

#### 業務構造の把握

- 階層的な業務構造の整理に、DFDの活用を検討する。  
業務で取り扱う情報、外部と業務の間で発生するイベント情報や業務間で発生するイベント情報、などの主要情報の整理にDFDを活用することで、階層的な業務構造整理を並行して行うことができる。  
※「技法ガイド(DFD記述ガイド編)」を参照
- 業務の候補は、以下のような方法を活用するなどして抽出する。

No	抽出方法の例	補足
1	各種資料(会社説明資料、IR資料、業務マニュアル、新人教育資料など)から業務を抽出する。	自治体業務など、業務整理のベースとなる参照モデルが存在する場合もある。
2	概念エンティティのステータス分析を行うことで、ステータス遷移のトリガーから業務を抽出する。	
3	業務上のイベントを分析することで、イベントから起動される業務を抽出する。	

#### 業務フローの把握

- 業務で実行される作業、実行者、実行タイミング、作業の順序性/前後関係、分岐条件などを、業務フローで整理することで、システム化対象の業務プロセスで行われている仕事内容が把握できる。
- 業務の通常フローにあわせて、その業務で発生する例外的なフローに関しても把握する。  
※例えば、通常フローでは、提出されたA文書の内容を確認した上で後続の業務作業を行うことになっているが、緊急時はメール文面での通達を元に後続の業務作業を先に行い、A文書の確認は後ほど行うといったような例外がある場合など。
- 作成した個々の業務フローを比較することで、業務プロセスの共通性(業務バリエーション)を把握する。
- 認識したアクター基点で関与する業務フローを確認することで、業務フローの認識漏れ等を検知できる。

#### 業務上の管理対象となる状態とその状態遷移の把握

- 状態遷移モデルでは、まず状態遷移トリガーとなる業務イベントを整理し、次にその状態遷移トリガーで利用するシステム機能を明確にする。

#### 業務イベントの把握

- 業務イベントの抽出は、以下のようなイベント種類を考慮して行う。

No	イベント種類	イベント例
1	ユーザーがトリガーとなる「ビジネスイベント」	注文依頼、注文変更依頼(出荷前)など
2	日時等がトリガーとなる「タイマーイベント」	カード利用決済の締め日など
3	外部システム等がトリガーとなる「システムイベント」	出荷管理システムからの出荷完了通知など

- 業務イベントの把握は、以下のような手順で行う。

No	手順内容
1	イベント一覧の属性項目と設定値を定義する。
2	画面や帳票での入出力の発生源となるイベントを抽出し、イベント一覧へ追加する。
3	業務の「始まり」と「終わり」やその間の業務フローから、トリガーとなるイベントを抽出し、イベント一覧に追加する。
4	業務上管理が必要な状態をモデル化した状態遷移モデル定義からイベントを抽出し、イベント一覧に追加する。
5	抽出したイベントをグルーピングして分析し、抜け漏れを補完し、重複を削除する。
6	イベントをトリガーとして実行される業務・アクティビティを確認し、業務・アクティビティの抜け漏れを洗い出す。 その結果を作成済みの業務階層定義・業務フローに反映する。

#### 適用技法

- ビジネスモデルキャンバス(ビジネスモデルを把握するために有効)
- 業務階層定義(業務の全体像を把握するために有効)
- DFD(階層的な業務構造の把握や、業務で扱う情報の流れを把握するために有効)
- 業務フロー図(業務の流れを把握するために有効)
- 業務俯瞰図(業務間の依存関係を俯瞰して把握するために有効)
- ステートマシン図(業務で発生するイベントを把握するために有効)



## G1-01-02 現行業務ルールの調査

アクティビティ概要																									
業務を遂行するにあたり適用され、守らなければならないルールを理解するために、業務ルール視点で現行業務(As-Is)を調査する。																									
インプット	アウトプット																								
<ul style="list-style-type: none"><li>● 現行業務の資料(As-Is) ※業務概要書、業務ルール定義、業務フロー、業務マニュアル 法令、社内規定、など</li></ul>	<ul style="list-style-type: none"><li>● 業務ルール定義(As-Is)</li></ul>																								
手順																									
<div>1. 業務ルールの抽出・把握準備<div>1.1. 業務ルール分析のインプット資料整理 業務ルール分析の為にインプット資料を集めて一覧化する。各資料に記載されている事項などを整理し、ルール抽出作業で何を把握したいとき何を確認すればよいかを、標準化する。</div><div>1.2. 業務ルール抽出単位の決定 業務ルールの抽出を実施する単位を決め、各抽出作業の範囲やボリュームを可視化する。 ※業務階層上の一定の階層にある業務単位、業務フロー上の業務作業単位、など。</div><div>1.3. 業務ルールの熟練者の確認 特定の業務ルール分析対象や特定の業務ルール種類に関する熟練者が、お客さまやベンダー内にいないかを確認する。 存在する場合は、業務ルールの概要説明依頼や業務ルールの網羅性チェック依頼等の必要な支援内容を検討して要請する。</div><div>1.4. 分析アプローチの決定 業務ルール抽出の実施対象毎に、適切な分析手法を決定する。</div></div> <div>2. 業務ルールの抽出・把握<div>2.1. 業務マニュアル等からの業務ルールの抽出・把握 業務マニュアル、約款、標準、商習慣等を確認し、事実・制約・計算・判断・契機、等の分類で業務ルールを抽出・把握する。</div><div>2.2. 業務フローからの業務ルールの抽出・把握 業務フローの業務作業の内容を確認し、業務作業のインプットをアウトプットに変換する過程から内部にある業務ルールを抽出・把握する。</div><div>2.3. 設計書からの業務ルールの抽出・把握 ビジネスロジックに変換された業務ルールが記述されている可能性のある設計文書を特定し、該当の設計文書の内容から業務ルールを抽出・把握する。</div><div>2.4. 概念エンティティからの業務ルールの抽出・把握 概念エンティティが取り得る状態遷移と遷移リガーとなるユーザーやシステムの振る舞いと条件を整理することで、業務ルールを抽出・把握する。</div></div> <div>3. 業務ルール抽出・把握結果確認抽出した業務ルールが最新で正しい内容であることをステークホルダーに確認し、必要であれば更新する。 ※例えば、規約変更等で不要になった業務ルールが混入している可能性がある。</div>																									
【重要】上手に進めるためのポイント、注意事項																									
※「G1-01-01 現行業務プロセスの調査」の【重要】上手に進めるためのポイント、注意事項 《全般》も参照。																									
➤ 業務ルールは以下のように分割でき、分類毎に具体的な業務ルールの種類が定義できる。																									
<table><tr><th>No</th><th>分類</th><th>定義</th><th>業務ルール例</th></tr><tr><td>1.</td><td>事実</td><td>業務に関する一般的な真実</td><td><ul style="list-style-type: none"><li>「優良顧客」「一般顧客」「見込顧客」の3種の顧客分類がある。</li><li>商品配送は「未発送」「手配中」「手配完了」「発送済」の何れかの状態を取る。</li></ul></td></tr><tr><td>2.</td><td>制約</td><td>業務・アクティビティに対して、その実行や内容の制限をかける。</td><td><ul style="list-style-type: none"><li>一般顧客に対する値引き販売は、事前に上長承認を得なければならない。</li><li>販売日報は翌営業日の午前 10:00 までに営業本部に提出しなければならない。</li></ul></td></tr><tr><td>3.</td><td>計算</td><td>特定の数式やアルゴリズムを使用して、何かしらの情報を導き出す。</td><td><ul style="list-style-type: none"><li>一般顧客の商品購入に対する付与ポイント計算式</li></ul></td></tr><tr><td>4.</td><td>判断</td><td>特定の情報を用いて、何かしらの選択を行う。</td><td><ul style="list-style-type: none"><li>1回の購入金額が1万円を超える場合は、送料を無料とし、顧客に請求しない。</li><li>請求元のカード会社ごとに専用の様式で利用者向けの請求書を発行する。</li></ul></td></tr><tr><td>5.</td><td>契機</td><td>特定の状況で、何かしらの業務・アクティビティを実行する。</td><td><ul style="list-style-type: none"><li>月次請求締め時に年間購入金額が5万円超の顧客に、VIPカードを送付する。</li></ul></td></tr></table> <p>『ビジネスルールを可視化する要件定義の図解術』[4] P17 の表 1-1 ビジネスカテゴリ一覧を参考に作成]</p>		No	分類	定義	業務ルール例	1.	事実	業務に関する一般的な真実	<ul style="list-style-type: none"><li>「優良顧客」「一般顧客」「見込顧客」の3種の顧客分類がある。</li><li>商品配送は「未発送」「手配中」「手配完了」「発送済」の何れかの状態を取る。</li></ul>	2.	制約	業務・アクティビティに対して、その実行や内容の制限をかける。	<ul style="list-style-type: none"><li>一般顧客に対する値引き販売は、事前に上長承認を得なければならない。</li><li>販売日報は翌営業日の午前 10:00 までに営業本部に提出しなければならない。</li></ul>	3.	計算	特定の数式やアルゴリズムを使用して、何かしらの情報を導き出す。	<ul style="list-style-type: none"><li>一般顧客の商品購入に対する付与ポイント計算式</li></ul>	4.	判断	特定の情報を用いて、何かしらの選択を行う。	<ul style="list-style-type: none"><li>1回の購入金額が1万円を超える場合は、送料を無料とし、顧客に請求しない。</li><li>請求元のカード会社ごとに専用の様式で利用者向けの請求書を発行する。</li></ul>	5.	契機	特定の状況で、何かしらの業務・アクティビティを実行する。	<ul style="list-style-type: none"><li>月次請求締め時に年間購入金額が5万円超の顧客に、VIPカードを送付する。</li></ul>
No	分類	定義	業務ルール例																						
1.	事実	業務に関する一般的な真実	<ul style="list-style-type: none"><li>「優良顧客」「一般顧客」「見込顧客」の3種の顧客分類がある。</li><li>商品配送は「未発送」「手配中」「手配完了」「発送済」の何れかの状態を取る。</li></ul>																						
2.	制約	業務・アクティビティに対して、その実行や内容の制限をかける。	<ul style="list-style-type: none"><li>一般顧客に対する値引き販売は、事前に上長承認を得なければならない。</li><li>販売日報は翌営業日の午前 10:00 までに営業本部に提出しなければならない。</li></ul>																						
3.	計算	特定の数式やアルゴリズムを使用して、何かしらの情報を導き出す。	<ul style="list-style-type: none"><li>一般顧客の商品購入に対する付与ポイント計算式</li></ul>																						
4.	判断	特定の情報を用いて、何かしらの選択を行う。	<ul style="list-style-type: none"><li>1回の購入金額が1万円を超える場合は、送料を無料とし、顧客に請求しない。</li><li>請求元のカード会社ごとに専用の様式で利用者向けの請求書を発行する。</li></ul>																						
5.	契機	特定の状況で、何かしらの業務・アクティビティを実行する。	<ul style="list-style-type: none"><li>月次請求締め時に年間購入金額が5万円超の顧客に、VIPカードを送付する。</li></ul>																						
➤ 業務ルール分類毎にインプットとする文書や抽出対象が異なるため、業務ルール分類を意識して作業を行うことが重要である。																									
上手に進めるためのポイント、注意事項																									

※「G1-01-01 現行業務プロセスの調査」の「上手く進めるためのポイント、注意事項」《全般》も参照。

- 業務ルールはプログラムのロジック(入力チェック、画面表示のための計算、テーブルへの値設定、画面遷移条件など)として様々なところに埋め込まれており、可視化されていない業務ルールも存在することに留意する。  
※「G1-02-01 現行システムの調査」も参照

#### 適用技法

- デシジョンテーブル(複雑な条件の業務ルールを整理するために有効)

## G1-01-03 現行業務データの調査

アクティビティ概要	
業務で取り扱う情報とそれらの関連を理解するために、業務データの視点で現行業務(As-Is)を調査する。	
インプット	アウトプット
<ul style="list-style-type: none"> <li>● 現行業務の資料(As-Is) ※概念データモデル、論理データモデル、実帳票、実 DB など</li> </ul>	<ul style="list-style-type: none"> <li>● 概念データモデル定義(As-Is)</li> </ul>
手順	
<ol style="list-style-type: none"> <li>1. 情報の特定 モデル化した業務プロセスや業務ルールから、使用する画面や帳票を特定し、そこで参照する情報を特定する。</li> <li>2. 情報の整理 特定した情報から、概念エンティティ・情報項目・エンティティ間の関連を整理する。</li> <li>3. 概念データモデルの作成 “手順2”で整理した内容をもとに概念データモデルを作成する。</li> <li>4. 概念データモデルの検証 作成した概念データモデルに対して、業務ルールとしての矛盾・不整合・漏れがないかを検証する。</li> </ol> <p>※上記の手順は、データは業務で活用するために存在するため、業務で使用する画面や帳票で必ず参照されるという前提で記述している。 ※他に業務標準用語や業務ルールから抽出した名詞や動詞を元に概念レベルの情報構造を把握するというアプローチ方法もある。</p>	
【重要】上手く進めるためのポイント、注意事項	
※「G1-01-01 現行業務プロセスの調査」の「【重要】上手く進めるためのポイント、注意事項」《全般》も参照。	
上手く進めるためのポイント、注意事項	
<p>※「G1-01-01 現行業務プロセスの調査」の「上手く進めるためのポイント、注意事項」《全般》も参照。</p> <ul style="list-style-type: none"> <li>➤ 概念エンティティ間の依存関係・多重度から情報構造に関する業務ルールを把握できる。</li> <li>➤ 現行システムが存在する場合は、上記手順を踏まずに、現行の論理データモデルから概念データモデルを作成する方法もある。 この場合、論理データモデルから設計・実装上の都合(システム都合)で定義されたエンティティを除外したり、一部を抽象化したりするなどし、概念データモデルを作成する。</li> </ul>	
適用技法	
<ul style="list-style-type: none"> <li>● ER 図</li> <li>● クラス図</li> </ul>	

## G1-02 現行システムの調査

サブプロセス目的
業務課題・業務要求を正確に把握するために、現行システムを理解する。 どのようなシステム機能があり、そのシステム機能が業務でどのように利用されているかをシステム全体の視点から説明できるようにする。
サブプロセス概要
現行業務・現行システムが抱える業務課題を明らかにするための準備作業として、お客さまから入手した資料を元に現行システムの調査を行う。

## G1-02-01 現行システムの調査

アクティビティ概要	
以下を目的に、システム UI 機能・外部インターフェース機能・バッチ機能、およびデータモデルの視点で現行システム(As-Is)を調査する。 ✓ ビジネス・業務とシステムの繋がりの明確化 ✓ システム機能の構成要素と全体構造の把握 ✓ システム実装に埋もれた業務ルールの抽出	
インプット	アウトプット
<ul style="list-style-type: none"> <li>● 現行システムの資料(As-Is) 各種システム機能設計書、論理データ設計書、実画面、実帳票、実 DB 情報など</li> <li>● 業務フロー(As-Is)</li> <li>● 業務ルール定義(As-Is)</li> <li>● 概念データモデル定義(As-Is)</li> </ul>	<ul style="list-style-type: none"> <li>● システム機能一覧(As-Is)</li> <li>● CRUD(As-Is)</li> <li>● システム機能俯瞰図(As-Is)</li> </ul>
手順	
<ol style="list-style-type: none"> <li>1. 現行システムに関する情報の収集               <ol style="list-style-type: none"> <li>1.1. システム UI 機能の収集 現行システムの資料(As-Is)の画面設計や帳票設計などからユーザーのインターフェースとなる画面機能・帳票機能を洗い出す。</li> <li>1.2. 外部インターフェース機能の収集 現行システムの資料(As-Is)の外部 IF 設計などから外部システムとのインターフェース機能を洗い出す。</li> <li>1.3. バッチ機能の収集 現行システムの資料(As-Is)のバッチ設計などからバッチ機能を洗い出す。</li> <li>1.4. データ情報の収集 現行システムの資料(As-Is)の ER 図やエンティティ定義やデータベーススキーマ情報などから、エンティティを洗い出す。</li> </ol> </li> <li>2. 収集した現行システムに関する情報の分析               <ol style="list-style-type: none"> <li>2.1. システム機能の整理 1.1～1.3 で収集した情報を元に、業務上意味のある単位に抽象化したシステム機能をシステム機能一覧で整理する。</li> <li>2.2. 情報構造の整理 1.4 で収集した情報を元に、概念レベルのエンティティに抽象化する。 その上で、「G1-01-03 現行業務データの調査」で作成した概念データモデル定義に反映し、概念エンティティ間の依存関係・多重度から情報構造で表現される業務ルールを整理する。</li> <li>2.3. システム機能とデータの関連整理 2.1 で整理したシステム機能と 2.2 で整理した概念エンティティとの関連を CURD(Create、Read、Update、Delete)で整理し、データライフサイクル視点での矛盾(Create がないのに Read がある、など)がないことを確認する。</li> <li>2.4. 業務作業とシステム UI 機能の関連整理 「G1-01-01 現行業務プロセスの調査」で可視化された業務作業とシステム UI 機能の関連を整理する。</li> <li>2.5. 業務ルールの整理 システム設計や実装に落とされ、業務分析では可視化されない計算式や基準などの業務ルールを抽出し、「G1-01-02 現行業務ルールの調査」で作成した業務ルール定義に反映する。</li> </ol> </li> </ol>	

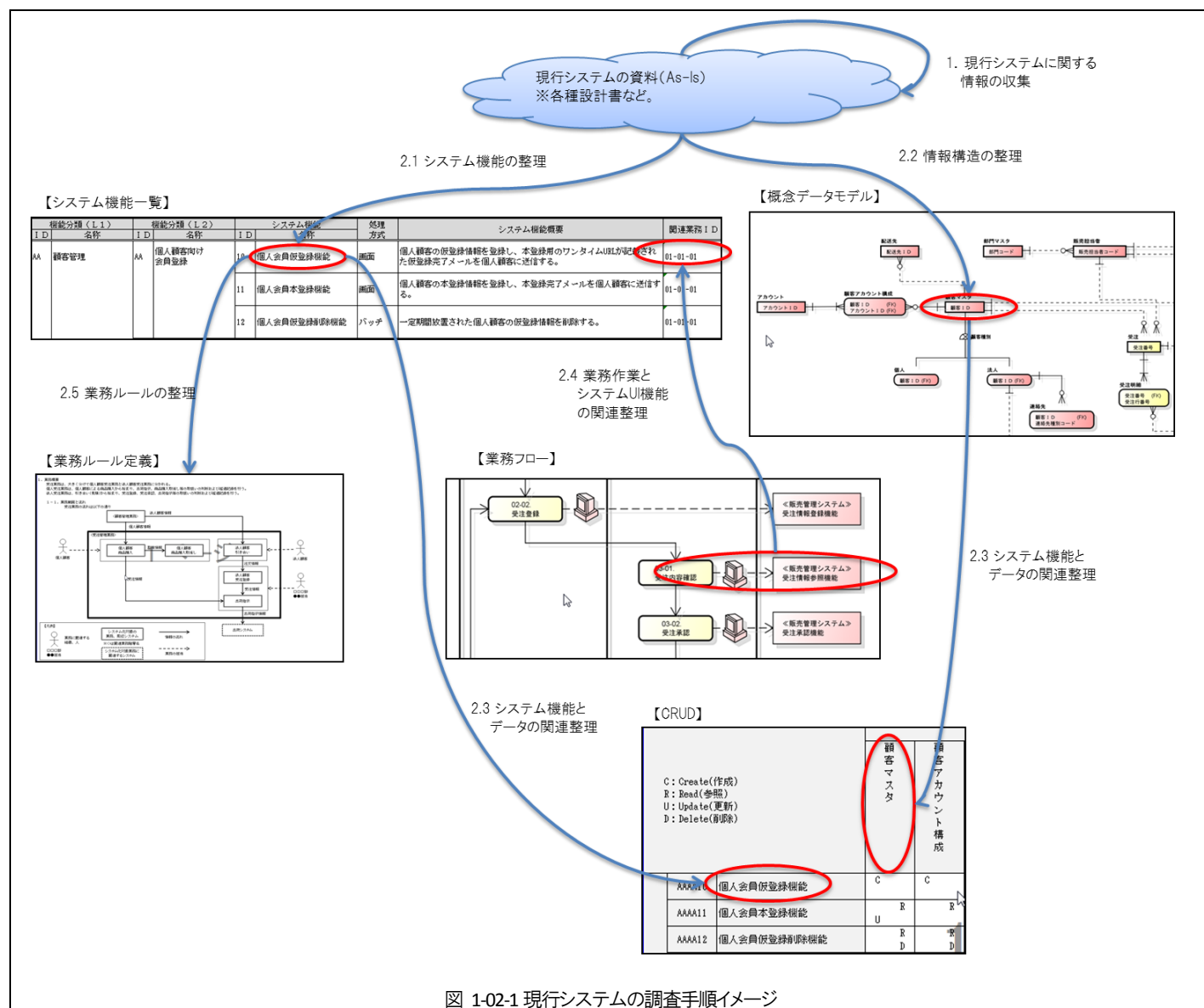


図 1-02-1 現行システムの調査手順イメージ

## 【重要】上手に進めるためのポイント、注意事項

- 調査目的を明確にし、何をどのレベル(粒度)で把握し、どのように可視化するかを標準化する。むやみに調査を行わず、調査結果の活用目的を明確にし、必要な事と必要なレベルまで調査する。画面や帳票などの機能を羅列しても、後続作業で活用できない。例えば次期システムでアーキテクチャが全面刷新されるならば、現行システムのアーキテクチャを調査する必要がないケースもあり得る。サブシステム/機能構造、業務との関係、データエンティティとの関係、などを全体像で把握することが原則である。また、分析担当者毎の分析レベルのばらつきを防止するために分析対象やレベルを標準化する。5W1Hの観点で検討すると良い。※「基礎知識集：3-6. 要件内容を明確化するための『要件の視点』」を参照

## 上手に進めるためのポイント、注意事項

- 業務ルールに基づいた本質的な機能要件を明確にする。システム機能には、業務ルールに基づいた本質的な機能要件の他に、システムの構造・方式・技術等の制約から生まれた実装が含まれている。現行システム調査では、次期システムに引き継ぐべき前者の把握が重要。後者は、次期システム開発における制約で再検討される為、この段階では調査対象外として良い。
- マイグレーションプロジェクトの場合は、「既存システムと新システムの機能をマッチングして、システム機能統廃合の妥当性を検討する」という観点も含めて、現行システム調査の成果物を検討する。
- システム機能がインプットをどう加工してアウトプットするかを考えることで、業務ルールの候補が洗い出される。例えば、インプットされた条件を元に集計結果をアウトプットするようなシステム機能の場合、集計するための計算式や基準が業務ルールとして抽出される。
- バッチ機能を構成するプログラムの実行順序を決定している要因を確認し、背後にある業務ルールの有無を確認する。

例えば、出荷情報取り込みバッチ後に売上計上バッチが起動しているような場合、出荷基準で売上を計上するという業務ルールが抽出されることがある。

#### 適用技法

- （特になし）

G1-03 課題の抽出と原因分析

サブプロセス目的
プロジェクト目的・目標を達成するために解決すべき業務上の課題を明確にする。 また、適切な実現手段を検討するため、業務課題の根本原因を特定する。
サブプロセス概要
システム化企画段階で既に表明されている業務課題、およびステークホルダーから抽出した業務課題を取りまとめ、業務課題の原因分析を行う。 本サブプロセスでは、図 1-03-1 要求抽出の概念構造の「課題」と「原因」を取り扱う。

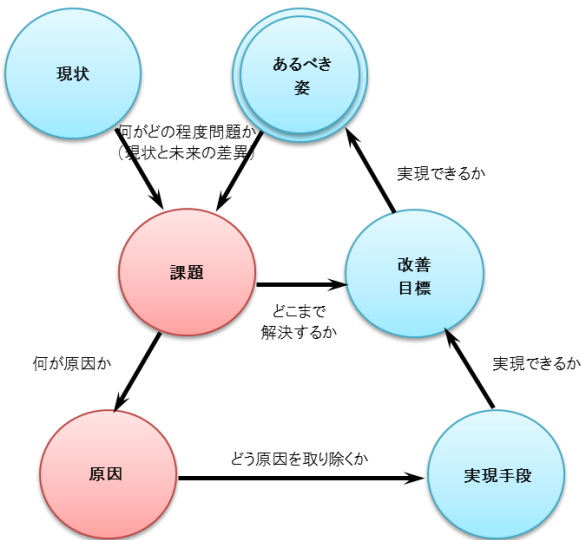


図 1-03-1 要求抽出の概念構造



## G1-03-01 システム化企画資料からの課題抽出

アクティビティ概要	
システム化企画資料から既に表明されている課題を抽出・確認する。	
インプット	アウトプット
<ul style="list-style-type: none"> <li>● システム化企画資料(現行課題、要求事項等)</li> <li>● RFP(現行課題、要求事項等)</li> <li>● 現行業務の資料(As-Is)</li> <li>● 現行システムの資料(As-Is)</li> </ul>	<ul style="list-style-type: none"> <li>● 業務課題一覧</li> </ul>
手順	
<ol style="list-style-type: none"> <li>1. システム化企画資料、RFP などから既に表明されている業務課題を抽出する。</li> <li>2. 抽出した課題の解決による、プロジェクト目標達成の実現性を評価する。</li> <li>3. 十分な実現性がない場合、当該目標達成に関わる仮説課題を分析する。</li> <li>4. 現行業務分析や現行システム分析の結果から仮説課題の裏付けを取り、解決による目標達成への寄与度合いや解決の難易度、解決の実現性などから、多角的に解決対象の課題を選別する。</li> </ol>	
【重要】上手に進めるためのポイント、注意事項	
<ul style="list-style-type: none"> <li>➤ どのプロジェクト目的・目標に対する課題であるかを明確にする。 プロジェクト目的・目標に紐付かない課題が存在する場合、プロジェクト目的・目標の不足、もしくは本質的ではない課題である可能性があるため、プロジェクト目的・目標との関連性が明確になるレベルまで、ステークホルダーを巻き込んで課題を深掘りする。</li> <li>➤ 業務・システムの要求事項のみ記述されている場合は、その要求の背景をお客さまに確認し、根本の課題を特定する。 課題の正しい理解が、要求の正しい理解に繋がるため、要求の背景となる課題を特定することは重要である。 しかし、システム化企画資料や RFP には要求事項のみ記述され、その背景となる課題が記述されていないことが多いため、注意する。</li> <li>➤ 課題の範囲(業務・システム)で分類する。 業務課題とシステム課題に仕分けし、システム課題の分析についてはシステム要件定義で行う。 これは、業務課題・要求を理解した後の方が、システム課題を業務と関連付けて理解でき、より良い解決策の検討が可能になるためである。</li> </ul>	
上手に進めるためのポイント、注意事項	
<ul style="list-style-type: none"> <li>➤ 業務レベルの課題は、「G1-01 現行業務の調査」で整理した業務フロー・業務ルール・概念データモデルを参考に課題箇所、内容を具体化する。</li> </ul>	
適用技法	
<ul style="list-style-type: none"> <li>● (特になし)</li> </ul>	

## G1-03-02 ステークホルダーからの課題抽出

アクティビティ概要	
<p>ステークホルダーからプロジェクト目的・目標に対する課題を抽出する。</p> <p>また、ここまで抽出した全ての課題が、プロジェクト目的・目標に対する課題になっていることを要件定義計画で定めたステークホルダーと合意する。</p>	
インプット	アウトプット
<ul style="list-style-type: none"> <li>（特になし）</li> </ul>	<ul style="list-style-type: none"> <li>業務課題一覧</li> </ul>
手順	
<ol style="list-style-type: none"> <li>課題抽出の基準となるプロジェクト目的・目標を説明する。</li> <li>既出の課題内容とプロジェクト目的・目標との関連性を説明し、合意を得る。</li> <li>各課題の解決による、プロジェクト目的・目標達成の見通しを立て、合意を得る。</li> <li>達成の見通しが付かない場合、ステークホルダーを巻き込んで、課題抽出を行う。（以降、”手順 2～4”を繰り返す）</li> </ol>	
【重要】上手く進めるためのポイント、注意事項	
<ul style="list-style-type: none"> <li>➤ ステークホルダーが直接的な要求を提示された場合は、その要求が発生した理由、必要な理由から、関連する課題を明らかにする。課題を明らかにすることで、ステークホルダーが提示した要求が課題に対して妥当な要求であるかの判断や、別の解決策を検討することが可能になる。</li> <li>➤ 抽出された課題がプロジェクト目的・目標と比較し、妥当な課題であるかを確認する。ステークホルダーの意見を取り入れているうちに、現場の課題を解決し、かつ使いやすいシステムになったが、プロジェクト目的・目標を達成できなかった（ビジネス上の課題を解決できなかった）ということも起こりうる。そのため、抽出された課題がプロジェクト目的・目標と整合性が取れていない場合、課題の却下もしくはプロジェクト目的・目標の見直しが必要になる。</li> <li>➤ 抽出した課題は、関係するステークホルダーと合意する。課題定義は、課題の原因分析、課題解決の実現手段検討を経て、具体的な要件定義へ辿り着く出発点となるため、課題に対してステークホルダーと共通認識を持つこと、およびステークホルダーから課題内容の合意を得ることが重要である。</li> </ul>	
上手く進めるためのポイント、注意事項	
<ul style="list-style-type: none"> <li>➤ ベンダーが主体となってステークホルダーから課題を引き出す。その為、システム化企画資料からの課題抽出で、多面的に分析された仮説課題を立てておくことが重要である。</li> <li>➤ 業務やシステムに関連するステークホルダーの役割を意識することで、そのステークホルダー視点での課題が洗い出しやすくなる。例えば、RACI マトリクスで役割を洗い出し、各役割を網羅するように視点を変えて課題分析する。</li> <li>➤ 「G1-03-01 システム化企画資料からの課題抽出」で抽出した課題内容の妥当性をステークホルダーに確認し、認識不足や誤認識を補正する。</li> </ul>	
適用技法	
<p>ステークホルダーから課題を引き出すためには、以下のような適用技法が有効である。</p> <ul style="list-style-type: none"> <li>● インタビュー</li> <li>● アンケート</li> <li>● ワークショップ</li> <li>● プレインストーミング</li> </ul>	

## G1-03-03 課題の原因分析

アクティビティ概要	
ステークホルダーを巻き込み、抽出された課題の原因分析を行う。	
インプット	アウトプット
● 業務課題一覧	● 業務課題一覧(原因分析済み)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
<ul style="list-style-type: none"> <li>➤ 課題の根本原因を特定することで、効果的な対策を検討できる。</li> <li>➤ 個々の課題のスコープ(業務・システム)を識別した上で、根本的な原因を分析する。</li> <li>➤ 課題と原因は1対1の関係とは限らないことに留意する。 複数の課題が1つの原因に紐付くことも1つの課題が複数の原因によって発生している場合もある。</li> <li>➤ 課題理解のために、課題を放置した場合に何が起こるかを洞察することも有効である。</li> </ul>	
適用技法	
<ul style="list-style-type: none"> <li>● 問題分析</li> <li>● KI 法</li> <li>● なぜなぜ分析</li> <li>● 特性要因分析</li> </ul>	

G1-04 課題解決後のゴール定義

サブプロセス目的
適切な実現手段の検討を可能とするため、課題解決後のゴールを明確にする。
サブプロセス概要
<p>抽出した業務課題に対する課題解決により達成または維持すべき、業務上のゴールを定義する。 本サブプロセスでは、図 1-04-1 の「改善目標」を取り扱う。</p> <div><pre>graph TD; J[現状] -- "何がどの程度問題か (現状と未来の差異)" --&gt; K[課題]; A[あるべき姿] -- "実現できるか" --&gt; K; K -- "どこまで解決するか" --&gt; M[改善目標]; K -- "何が原因か" --&gt; R[原因]; M -- "実現できるか" --&gt; S[実現手段]; R -- "どう原因を取り除くか" --&gt; S;</pre></div> <p>図 1-04-1 要求抽出の概念構造</p>

## G1-04-01 課題解決後のゴール定義

アクティビティ概要	
期待効果を得られる適切な実現手段を検討するために、課題解決後の状態を明確に定義する。	
インプット	アウトプット
<ul style="list-style-type: none"> <li>● 業務課題一覧(原因分析済み)</li> </ul>	<ul style="list-style-type: none"> <li>● 業務課題一覧(ゴール定義済み)</li> </ul>
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<ul style="list-style-type: none"> <li>➤ プロジェクト目的・目標、および制約・前提事項に矛盾のないゴールを定義する。</li> <li>➤ 定量的効果、定性的効果を意識してゴールを定義する。</li> </ul>	
上手く進めるためのポイント、注意事項	
<ul style="list-style-type: none"> <li>➤ 効果の計測方法や成功判断基準を明確に定義する。</li> <li>➤ 課題解決後のゴールを定義することで、ステークホルダーの抱える課題をより正確に把握できる。</li> </ul>	
適用技法	
<ul style="list-style-type: none"> <li>● ゴール指向分析</li> <li>● CATWOE 分析</li> <li>● SMART フレームワーク</li> </ul>	

## G1-05 課題解決の実現手段検討

サブプロセス目的

業務課題を解決し、ゴールが達成された状況またはゴールが維持されている状況を実現するための手段を明確にする。

## サブプロセス概要

業務課題を解決し、ゴールが達成された状況またはゴールが維持されている状況を実現するための方向性を検討する。その上で実現手段を定義する。本サブプロセスでは、図 1-05-1 の「実現手段」を取り扱う。

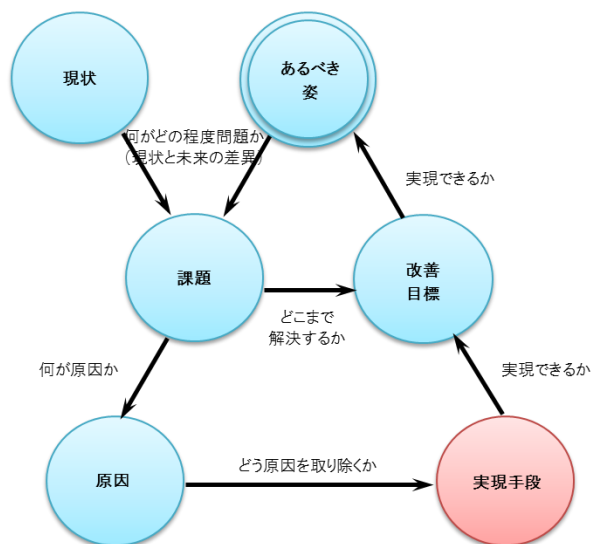


図 1-05-1 要求抽出の概念構造

## G1-05-01 実現手段の方向性検討

### アクティビティ概要

定義された「課題解決後のゴール」の実現に向けた具体的な手段を検討する前に、改善方針を見定める。  
例えば、図 1-05-2 のような実現手段の方向性を検討する。

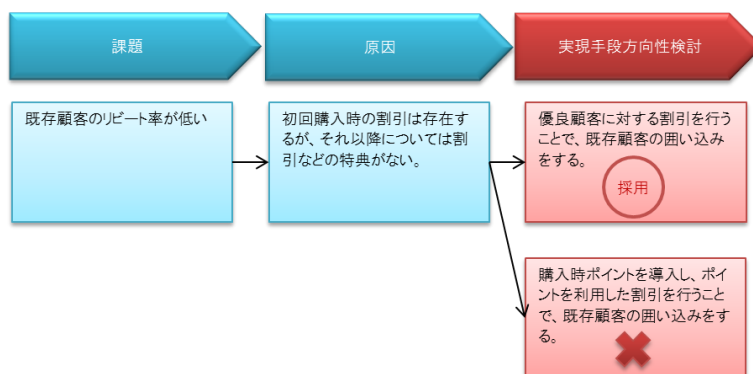


図 1-05-2 例: 実現手段の方向性検討

インプット	アウトプット
<ul style="list-style-type: none"> <li>業務課題一覧(ゴール定義済み)</li> <li>要件定義計画書(プロジェクトの目的・目標、制約条件・前提条件)</li> </ul>	<ul style="list-style-type: none"> <li>実現手段方向性検討資料</li> </ul>
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<ul style="list-style-type: none"> <li>➤ 「業務をどう変えるのか」の視点で、実現手段の方向性を検討する。 「システムをどう変えるのか」という視点になりやすいが、「業務をどう変えるのか」を検討した上で、その業務要件を実現するために「システムをどう変えるのか」を考える必要がある。</li> <li>➤ 要件定義計画段階で確認した、プロジェクトの目的・目標および制約条件・前提条件に整合した、実現手段の方向性を検討する。</li> </ul>	
上手く進めるためのポイント、注意事項	
<ul style="list-style-type: none"> <li>➤ 個々の課題に対する個別最適な方向性でなく、関連する課題間や業務全体において最適な方向性を検討する。</li> <li>➤ システム化以外の実現手段も選択肢として検討する。</li> <li>➤ 後続で実施する「G1-05-02 実現手段の定義」の範囲を絞り込むため、実現手段の方向性は、ステークホルダーに確認して頂く。</li> </ul>	
適用技法	
<ul style="list-style-type: none"> <li>プロセス改善キーワード</li> </ul>	

## G1-05-02 実現手段の定義

### アクティビティ概要

課題原因を解消し、ゴールを達成する実現手段を定義する。  
例えば、図 1-05-3 のような実現手段を定義する。

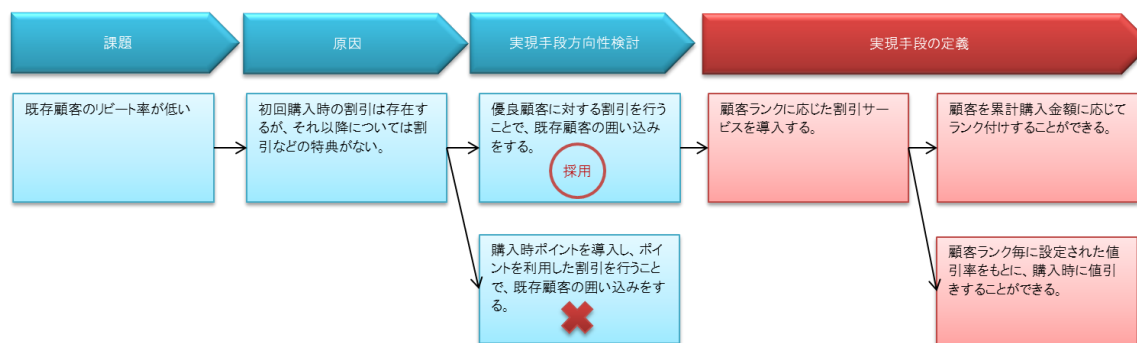


図 1-05-3 例: 実現手段の定義

### インプット

- 業務課題一覧(ゴール定義済み)
- 実現手段方向性検討資料

### アウトプット

- 業務課題一覧(実現手段定義済み)

### 手順

1. 実現手段の方向性に基づき、具体的な実現手段の案を検討する。
2. 実現手段に関する制約・前提事項を整理する。
3. 実現手段を選択する為の評価項目を設定する。
4. 評価項目、方法に従い、実現手段を評価し、決定する。
5. 課題→原因→実現手段→課題解決後のゴールのトレーサビリティを整理する。
6. 課題解決後のゴールに対して十分な効果を見込める実現手段になっていることを検証する。
7. 課題解決・改善目標(ゴール)を達成するために、十分な実現手段が、内容の間違いなくリストアップされていることをステークホルダーに確認する。

### 【重要】上手く進めるためのポイント、注意事項

- 実現手段は、プロジェクトスコープや現行業務や現行システムを意識せずに抽出する。プロジェクトスコープ等を意識すると検討範囲が狭くなり、最適な実現手段にならない可能性があるため。
- 課題→原因→実現手段→課題解決後のゴールのトレーサビリティは、ロジックツリーなどを活用し、成果物として残す。十分な実現手段が定義されていることをステークホルダーに確認するケース(手順 7 に該当)や、設計工程以降で要件の必要性や背景を再確認するようなケースなどに対応するため。

### 上手く進めるためのポイント、注意事項

—

### 適用技法

- ロジックツリー
- マインドマップ



## G2 業務要求の整理とモデル化

### G2-01 業務要求の整理

サブプロセス目的
ステークホルダーから抽出した要求の元になっているのは、ステークホルダーが個人の視野でとらえたローカルな業務であり、業務の全体像ではない。そのため、このプロセスで、抽出した要求の候補を整理・分類することによって、業務要求の全体整合性を確保する。
サブプロセス概要
これまでに抽出された実現手段の中からプロジェクトの範囲内の業務要求の候補をリストアップする。 その上で、業務要求を重要度・緊急度・難易度・使用頻度等の要求属性で分類し、網羅性チェック・矛盾チェック等の観点で確認する。

## G2-01-01 業務要求のリストアップと整理

アクティビティ概要	
業務要求候補の検証(要求の漏れ、整合性の確認等)を可能とするため、「G1-05 課題解決の実現手段検討」で抽出された実現手段から、プロジェクトスコープ内の業務要求候補をリストアップし、業務要求候補の粒度を合わせ、整理する。	
インプット	アウトプット
<ul style="list-style-type: none"> <li>業務課題一覧(実現手段定義済み)</li> </ul>	<ul style="list-style-type: none"> <li>業務要求一覧</li> <li>システム要求一覧</li> </ul> ※システム要求一覧は「G5-01 システム要件定義への引継ぎ」で利用
手順	
<ol style="list-style-type: none"> <li>業務要求候補のリストアップ 「G1-05 課題解決の実現手段検討」で抽出された実現手段から、プロジェクトスコープ内の業務要求をリストアップする。</li> <li>業務要求候補の具体化 リストアップした業務要求候補を、業務プロセス・業務ルール・業務データへの影響やシステム要求の超概略が把握できるレベルに具体化する。</li> <li>業務要求候補の属性定義 業務要求候補を分析し、要件定義計画で定めた要件の優先度評価方法を踏まえた、管理属性項目と属性値を設定する。</li> </ol>	
【重要】上手く進めるためのポイント、注意事項	
<b>業務要求候補のリストアップ</b> <hr/> <ul style="list-style-type: none"> <li>「G1-05 課題解決の実現手段検討」では、プロジェクトスコープや現行業務や現行システムを意識せずに実現手段を検討しているため、要求のリストアップでプロジェクトスコープ内の業務要求の候補を抽出する。</li> </ul>	
<b>業務要求候補の具体化</b> <hr/> <ul style="list-style-type: none"> <li>現行業務・現行システムの要件を踏襲する部分は、お客さまとの間で踏襲要件を何で確認し、合意するのかを明確にする。</li> <li>抽象度の高い要求と具体化された要求のトレーサビリティを記録しておく。</li> </ul>	
<b>業務要求候補の属性定義</b> <hr/> <ul style="list-style-type: none"> <li>ステークホルダーと実現対象の要求を合意する際は、メリットだけでなくデメリットも説明した上で合意する。後続工程で、デメリットが明らかになった場合、要件変更発生の要因になり得るため。</li> </ul>	
上手く進めるためのポイント、注意事項	
<b>業務要求候補のリストアップ</b> <hr/> <ul style="list-style-type: none"> <li>ビジネス・業務の目標に対して効果が高いなどの理由で、スコープを変更して対応する要求が発生し得る。その際は、要件定義計画(予算、体制、スケジュール)の見直し等、必要な調整を行う。</li> <li>現行踏襲要件の確認には、現行業務の可視化が有力な手段となる。 「お客さま内に有識者がいる」という理由で現行業務可視化をしないのは高リスクと考える。 要件定義開始後に、有識者不在や期待レベル未達などが分かり、要件定義計画実行に支障を及ぼす。 運用期間が長くなるにつれ、現行業務・システムの把握を人に依存するリスクは高まる。 よって、現行踏襲要件の確認は、現行調査結果の活用を前提に考えるべきである。</li> <li>業務要求を検討する過程で、付随的に洗い出されたシステム要求候補は、システム要求一覧に記録する。 業務要求に関するステークホルダーとの議論では「業務」に焦点を当てるが、業務要求に付随する形でシステム要求が表明される場合がある。 その場合、システム要求に関する詳細な議論・検討は行わずに、システム要求一覧へ記録し、システム要件定義へ引き継ぐ。</li> </ul>	
<b>業務要求候補の具体化</b> <hr/> <ul style="list-style-type: none"> <li>具体化のレベル(詳細度)は、ステークホルダーと合意しておく。</li> </ul>	

- 1つの文章で1つの要求を説明するように単一化を図る。  
1つの文章に複数の要求を含んでしまうと、後続で行う「G2-01-02 業務要求の確認」が難しくなる。
- 文章表現による一意な要求内容の伝達が難しい要求に対しては、モデルを使うことが有効である。
- 業務要求候補の具体化の過程で、不明事項や検討事項を洗い出す。

#### 業務要求候補の属性定義

- 要求属性および要求属性値の例は、「要件定義計画プロセスガイド：C2-02-03 優先順位付け基準・方法の設定」を参照。

#### 適用技法

- KI 法
- マインドマップ

## G2-01-02 業務要求の確認

アクティビティ概要																				
整理した業務要求候補に対して網羅性チェック・矛盾チェック等を行うことで、業務要求が全体として過不足なく整合性の取れた状態で抽出できていることを確認する。 ※要求の漏れや誤り等が判明した場合は、「G1 業務要求の収集」プロセスに戻る。																				
インプット	アウトプット																			
● 業務要求一覧	● 業務要求一覧(要求確認済み)																			
手順																				
1. 確認観点・方法の定義 整理された業務要求候補に対する確認観点と確認方法を定義する。																				
2. 業務要求候補の確認 定義した確認観点と確認方法で、業務要求候補を確認する。																				
3. 業務要求候補のステークホルダー確認 具体化した業務要求内容が妥当であることをステークホルダーに確認頂き、訂正事項を業務要求一覧に反映する。																				
【重要】上手く進めるためのポイント、注意事項																				
確認観点・方法の定義																				
➤ 以下を参考にプロジェクト特性に合わせた確認観点を定義する。 ※確認観点については、「技法ガイド(検証ガイド編)」も参照																				
<table><tr><th>No</th><th>確認観点</th><th>概要</th></tr><tr><td>1</td><td>重複チェック</td><td>内容が重複している要求がないかをチェックをする。</td></tr><tr><td>2</td><td>矛盾チェック</td><td>要求間の要求内容に矛盾がないかをチェックする。</td></tr><tr><td>3</td><td>網羅性チェック</td><td>上位の目的・目標を充足する観点から、要求に不足がないかをチェックする。</td></tr><tr><td>4</td><td>一貫性チェック</td><td>課題解決の方向性に準拠した要求内容になっているかをチェックする。</td></tr><tr><td>5</td><td>実現可能性チェック</td><td>技術的・コスト的な観点から、明らかに実現困難な要求はないかをチェックする。</td></tr></table>			No	確認観点	概要	1	重複チェック	内容が重複している要求がないかをチェックをする。	2	矛盾チェック	要求間の要求内容に矛盾がないかをチェックする。	3	網羅性チェック	上位の目的・目標を充足する観点から、要求に不足がないかをチェックする。	4	一貫性チェック	課題解決の方向性に準拠した要求内容になっているかをチェックする。	5	実現可能性チェック	技術的・コスト的な観点から、明らかに実現困難な要求はないかをチェックする。
No	確認観点	概要																		
1	重複チェック	内容が重複している要求がないかをチェックをする。																		
2	矛盾チェック	要求間の要求内容に矛盾がないかをチェックする。																		
3	網羅性チェック	上位の目的・目標を充足する観点から、要求に不足がないかをチェックする。																		
4	一貫性チェック	課題解決の方向性に準拠した要求内容になっているかをチェックする。																		
5	実現可能性チェック	技術的・コスト的な観点から、明らかに実現困難な要求はないかをチェックする。																		
業務要求候補のステークホルダー確認																				
➤ 各回の範囲やテーマに合わせた、適切なステークホルダーの参加が重要になる。																				
➤ リストアップした要求候補の全てがプロジェクトで実現する要求ではなく、「G2-02 業務要求のモデル化」以降のプロセスを経て、実現すべき要求(＝要件)に昇華することをステークホルダーにご理解頂く。																				
➤ 「G2-02 業務要求のモデル化」以降のプロセスに進める要求(受理する要求)とそうでない要求(却下する要求)をステークホルダーに取捨選択して頂く。却下する要求についてはその理由を記録する。																				
➤ ステークホルダー確認時に未知の要求を獲得した場合、プロジェクトの目的・目標に整合した価値ある要求候補であることを確認するため、既知の課題・解決ゴール等とのトレーサビリティを整理・確認することが重要である。 確認した未知の要求状態によって、以下の対応を行う。																				
<table><tr><th>No</th><th>獲得した要求と既知課題との関係</th><th>対応内容</th></tr><tr><td>1</td><td>既知の課題・解決ゴールに整合する場合</td><td>未知の要求と既知の課題・解決ゴールのトレーサビリティを確保する。</td></tr><tr><td>2</td><td>既知の課題・解決ゴールに整合しない かつプロジェクトの目的・目標と整合する場合</td><td>このケースは、課題の抽出漏れであるため、「G1-03 課題の抽出と原因分析」以降のプロセスを再実施し、課題抽出の精度をあげる。</td></tr><tr><td>3</td><td>既知の課題・解決ゴールに整合しない かつプロジェクトの目的・目標と整合しない場合</td><td>ステークホルダーと要求の取り下げを合意するか、上位の目的・目標との依存関係がないことを明確にした状態で要求として管理する。</td></tr></table>			No	獲得した要求と既知課題との関係	対応内容	1	既知の課題・解決ゴールに整合する場合	未知の要求と既知の課題・解決ゴールのトレーサビリティを確保する。	2	既知の課題・解決ゴールに整合しない かつプロジェクトの目的・目標と整合する場合	このケースは、課題の抽出漏れであるため、「G1-03 課題の抽出と原因分析」以降のプロセスを再実施し、課題抽出の精度をあげる。	3	既知の課題・解決ゴールに整合しない かつプロジェクトの目的・目標と整合しない場合	ステークホルダーと要求の取り下げを合意するか、上位の目的・目標との依存関係がないことを明確にした状態で要求として管理する。						
No	獲得した要求と既知課題との関係	対応内容																		
1	既知の課題・解決ゴールに整合する場合	未知の要求と既知の課題・解決ゴールのトレーサビリティを確保する。																		
2	既知の課題・解決ゴールに整合しない かつプロジェクトの目的・目標と整合する場合	このケースは、課題の抽出漏れであるため、「G1-03 課題の抽出と原因分析」以降のプロセスを再実施し、課題抽出の精度をあげる。																		
3	既知の課題・解決ゴールに整合しない かつプロジェクトの目的・目標と整合しない場合	ステークホルダーと要求の取り下げを合意するか、上位の目的・目標との依存関係がないことを明確にした状態で要求として管理する。																		
上手く進めるためのポイント、注意事項																				

## 確認観点・方法の定義

- 以下を参考に確認観点に合わせた確認方法を定義する。

No	方法	確認内容
1	要求のグループ化による確認 (*1)	要求に付与した属性値を類似の属性値でグループ化し、各確認観点で個々の要求内容を確認する。 ※例:対象業務分類・要求の分類・利用者などの属性でグループ化
2	文章確認	要求内容説明文章の主語・述語が明確であることを確認する。
		副詞や形容詞の対象が明確であること、定量化されていることを確認する。

(\*1):特定の観点(曖昧性チェック等)は、主要な要求をサンプリングして確認するといった方法もある。

## 業務要求候補のステークホルダー確認

- リストアップした要求候補は、後続の「G2-02 業務要求のモデル化」を経て、全体最適解となる。
- ベンダーが整理した業務要求一覧に対する、ステークホルダーからの変更内容と変更理由を記録する。

## 適用技法

- (特になし)

## G2-02 業務要求のモデル化

サブプロセス目的
業務要求および業務要求間のつながりをモデルとして表現することで、自然言語では発見しづらい要求の漏れ・矛盾・曖昧さをなくす。 また、モデル化することで、将来のシステムに必要となるシステム機能を明確にする。
サブプロセス概要
複数の視点で要求事項をモデル化する。 この作業の過程で要求の漏れ・矛盾・曖昧などが検出された場合は、「G1 業務要求の収集」に戻り、要求を再収集する。 また、各モデルから導出されたシステム機能をリストアップし、システム機能一覧を作成する。

## G2-02-01 業務要求のモデル化方法の見直し

アクティビティ概要	
要件定義計画時に設定したモデル化方法や成果物構成が、ステークホルダーと要件を合意する目的に対して最適か確認し、必要に応じて見直しを行う。	
インプット	アウトプット
<ul style="list-style-type: none"> <li>● 要件定義計画書(成果物定義)</li> <li>● 業務要求一覧(要求確認済み)</li> </ul>	<ul style="list-style-type: none"> <li>● 要件定義計画書(成果物定義見直し済み)</li> </ul>
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<p>※要件定義計画プロセスガイドの「C2-01 成果物定義」の「【重要】上手く進めるためのポイント、注意事項」も参照。</p> <p>➤ ベンダーが考える必要な成果物とその理由をお客さまに納得頂き、適正な成果物定義を合意する。 例えば、以下のように不適切な成果物定義にならないように留意する。</p> <ul style="list-style-type: none"> <li>✓ 「成果物を作らない」と誤解した、アジャイルアプローチを部分的に適用する。</li> <li>✓ プロジェクト特性・制約等を度外視して、フルセットのプロセス、成果物を適用する。</li> </ul>	
上手く進めるためのポイント、注意事項	
<p>※要件定義計画プロセスガイドの「C2-01 成果物定義」の「上手く進めるためのポイント、注意事項」も参照。</p> <p>➤ 下記観点でモデル化方法と目的の適合性を確認する。</p> <ul style="list-style-type: none"> <li>✓ お客さまのITリテラシーやシステム開発知識、社内文化</li> <li>✓ お客さま業務や業務要求の複雑度や難易度</li> <li>✓ お客さま業務に対する、ベンダー側メンバーの理解度</li> </ul>	
適用技法	
<ul style="list-style-type: none"> <li>● (特になし)</li> </ul>	

## G2-02-02 業務プロセスのモデル化

アクティビティ概要	
<p>システムが実現・支援する業務のプロセスを明確化するために、To-Be 業務プロセスモデルを作成する。</p> <p>To-Be 業務プロセスモデルをベースに、システム化方針や要求に沿ったシステム化対象業務、システム機能を定義する。</p> <p>※要求の漏れや誤り等が判明した場合は、「G1 業務要求の収集」プロセスに戻る。</p>	
インプット	アウトプット
<ul style="list-style-type: none"> <li>● ビジネスモデル定義(As-Is)</li> <li>● 組織一覧(As-Is)</li> <li>● 用語集(As-Is)</li> <li>● 業務俯瞰図(As-Is)</li> <li>● 業務階層定義(As-Is)</li> <li>● データフロー(As-Is)</li> <li>● アクター一覧(As-Is)</li> <li>● 業務フロー(As-Is)</li> <li>● イベント一覧(As-Is)</li> <li>● 状態遷移モデル定義(As-Is)</li> <li>● 業務要求一覧(要求確認済み)</li> </ul>	<ul style="list-style-type: none"> <li>● ビジネスモデル定義(To-Be)</li> <li>● 組織一覧(To-Be)</li> <li>● 用語集(To-Be)</li> <li>● 業務俯瞰図(To-Be)</li> <li>● 業務階層定義(To-Be)</li> <li>● データフロー(To-Be)</li> <li>● アクター一覧(To-Be)</li> <li>● 業務フロー(To-Be)</li> <li>● イベント一覧(To-Be)</li> <li>● 状態遷移モデル定義(To-Be)</li> </ul>
手順	
<ol style="list-style-type: none"> <li>1. ビジネスモデルの可視化 お客さまのビジネスモデルの変更内容を洗い出し、「G1-01-01 現行業務プロセスの調査」で作成したビジネスモデルに反映する。</li> <li>2. 組織構造・役割・所在の変更内容把握 お客さまの組織構造や役割、所在の変更内容を洗い出し、「G1-01-01 現行業務プロセスの調査」で作成した組織一覧に反映する。</li> <li>3. 業務標準用語の可視化 業務要求内容の説明文章から、新たな概念等を抽出し、業務標準の呼称、意味を用語集に追加する。</li> <li>4. 業務全体の流れ・依存関係の可視化 業務要求内容から「G1-01-01 現行業務プロセスの調査」で作成した業務俯瞰図や抽象度を上げた業務フロー図への変更点を洗い出し、反映する。</li> <li>5. 業務構造の可視化 業務要求内容からお客さま業務の追加・廃止、業務間の関係、上下階層との関係等を洗い出し、「G1-01-01 現行業務プロセスの調査」で作成した業務階層定義に反映する。</li> <li>6. 業務作業担当者の役割(アクター)の可視化 組織一覧などから業務作業担当者の役割の変更内容を洗い出し、「G1-01-01 現行業務プロセスの調査」で作成したアクター一覧に反映する。</li> <li>7. 業務フローの可視化 業務プロセスに関する業務要求事項を洗い出し、業務フローを作成する。 その上で、業務を支援するシステム機能の概略を検討し、業務フローに反映する。 ※既存業務プロセスに関する要求の場合は、「G1-01-01 現行業務プロセスの調査」で作成した業務フローに反映し、 新規業務プロセスに関する要求の場合は、新規に業務フローを作成する。</li> <li>8. 業務上の管理対象となる状態とその状態遷移の可視化 業務上の管理対象となる状態(受注状態など)の変更内容を洗い出し、「G1-01-01 現行業務プロセスの調査」で作成した状態遷移モデル定義に反映する。状態遷移モデルでは、状態遷移トリガーとなる業務イベントを整理し、次にその状態遷移トリガーで利用するシステム機能を可視化する。</li> <li>9. 業務イベントの可視化 業務要求内容を基にイベント分析を行い、外部から業務に対して発生するイベントや業務内で発生するイベントを洗い出し、「G1-01-01 現行業務プロセスの調査」で作成したイベント一覧に反映する。</li> </ol>	
【重要】上手く進めるためのポイント、注意事項	
<p>※「G1-01-01 現行業務プロセスの調査」の「【重要】上手く進めるためのポイント、注意事項」も参照</p>	
業務構造の可視化	
<ul style="list-style-type: none"> <li>➤ 追加する業務だけでなく、廃止する業務についても洗い出すことが重要である。 廃止する業務の洗い出しを漏らした場合、誰にも使われない不要なシステム機能が実装される可能性がある。</li> <li>➤ 業務階層定義の内容をステークホルダーに確認頂き、この業務構造を前提に業務プロセスや業務ルールの整理を進めることを了承頂く。</li> </ul>	
業務フローの可視化	
<p>※業務フローの記述方法については、「技法ガイド(業務フロー記述ガイド編)」を参照</p>	



- 業務プロセスの内容変更による影響を受けるアクターを整理する。  
当該のアクターには、業務プロセスの変更内容の妥当性確認および業務要求内容詳細化のサポートをしてもらう。
- 業務プロセスの実現性を確認する。  
例えば、業務で参照する業務データが、前方の業務で作成される流れになっているか？など、ロジカルに業務が流れる業務フローであることを確認する。

#### 上手に進めるためのポイント、注意事項

※「G1-01-01 現行業務プロセスの調査」の「上手に進めるためのポイント、注意事項」も参照

#### 業務作業担当者の役割(アクター)の可視化

- 「G2-01-02 業務要求の確認」の文章確認で、要求内容説明文章の主語・述語が明確にされていると、業務上の新たなアクターが抽出しやすい。

#### 業務フローの可視化

- アクターの役割変更、所在変更などに伴う業務プロセスの変更点に関しても業務フローに反映する。
- 業務フローの作業を行う組織が不明確である場合、まずは、どのような役割があれば業務フローが効率的に回るかを考え、役割ベースで業務フローを作成する。  
その上で、最終的に各役割をどの組織にマッピングするのかを明らかにする。

#### 適用技法

- ビジネスモデルキャンバス(ビジネスモデルを把握するために有効)
- 業務階層定義(業務の全体像を把握するために有効)
- DFD(階層的な業務構造の把握や、業務で扱う情報の流れを把握するために有効)
- 業務フロー図(業務の流れを把握するために有効)
- 業務俯瞰図(業務間の依存関係を俯瞰して把握するために有効)
- ステートマシン図(業務で発生するイベントを把握するために有効)

## G2-02-03 業務ルールのモデル化

アクティビティ概要	
<p>システムが実現・支援する業務に適用されるルールを明確化するために、To-Be 業務ルールモデルを作成する。</p> <p>※要求の漏れや誤り等が判明した場合は、「G1 業務要求の収集」プロセスに戻る。</p>	
インプット	アウトプット
<ul style="list-style-type: none"> <li>● 業務ルール定義(As-Is)</li> <li>● 業務要求一覧(要求確認済み)</li> <li>● 業務階層定義(To-Be)</li> <li>● 業務フロー(To-Be)</li> <li>● 用語集(To-Be)</li> </ul>	<ul style="list-style-type: none"> <li>● 業務ルール定義(To-Be)</li> </ul>
手順	
<ol style="list-style-type: none"> <li>1. 業務ルールの抽出 業務ルールの抽出を実施する単位(業務単位、業務作業単位、要求事項単位など)を定義し、抽出単位毎に必要と推察される業務ルールの候補をベンダー側で洗い出す。それを基にステークホルダーから業務ルールを引き出し、業務ルール定義に反映する。</li> <li>2. 業務ルールの検証 新たに抽出した業務ルールと既存の業務ルールとの間に、衝突や矛盾がないかを確認する。</li> <li>3. 業務プロセスの見直し 抽出した業務ルールから業務プロセスへの影響を検討し、必要に応じて、業務フロー等への反映を行う。 ※例えば、「ユーザーが行為をするために、資格を保有している必要がある」という業務ルールの場合、資格有無情報をシステムが保有する場合と、システムは資格情報を持たず資格有無を上席者が確認し、システムに「行為の許可」を入力する場合とでは、業務プロセスや必要なシステム機能が異なってくる。</li> </ol>	
【重要】上手く進めるためのポイント、注意事項	
<p>※「G1-01-02 現行業務ルールの調査」の「【重要】上手く進めるためのポイント、注意事項」も参照。</p> <p>➤ 既存の業務ルールと異なり、把握したい新規の業務ルールは、まだ業務マニュアル・約款等で可視化されていないため、ステークホルダーから引き出す。</p>	
上手く進めるためのポイント、注意事項	
<p>※「G1-01-02 現行業務ルールの調査」の「上手く進めるためのポイント、注意事項」も参照。</p> <p>➤ 業務要求の背後にある根拠、必要な理由を考えることで、業務ルールの候補を抽出できる。 ※例えば、「個人属性情報に生年月日を追加する」という要求の背景・理由を確認した結果、「誕生月の場合、10%の値引きを行う」といった業務ルール候補が抽出される。</p> <p>➤ 同じ業種や業務範囲のプロジェクトの要件定義で可視化した業務ルール事項を参考とすることが、効率よく網羅性の高い業務ルール定義に役立つ。</p> <p>➤ 永続的に文書で維持・管理されている業務ルールは、当該文書への参照リンクと概要説明に留めることも検討する。 原本を参照する形式を採ることで、内容同期が容易になる。</p> <p>➤ 新規ビジネスの立ち上げなどに関する要件定義の場合、要件定義と並行でお客さまによる業務プロセス・ルールの検討が行われることがある。 そのような場合、ベンダー側で可能な範囲で決定必要事項をブレイクダウンし、お客さまと状況共有しながら、業務プロセス・ルールの分析作業の実施タイミングを調整する必要がある。</p>	
適用技法	
<ul style="list-style-type: none"> <li>● デシジョンテーブル(複雑な条件の業務ルールを整理するために有効)</li> </ul>	

## G2-02-04 業務データのモデル化

アクティビティ概要	
システムが実現・支援する業務が管理する情報構造を明確化するために、To-Be 概念データモデルを作成する。 ※要求の漏れや誤り等が判明した場合は、「G1 業務要求の収集」プロセスに戻る。	
インプット	アウトプット
<ul style="list-style-type: none"> <li>● 概念データモデル(As-Is)</li> <li>● 業務要求一覧(要求確認済み)</li> <li>● 業務フロー(To-Be)</li> <li>● 業務ルール定義(To-Be)</li> <li>● データフロー(To-Be)</li> <li>● 用語集(To-Be)</li> </ul>	<ul style="list-style-type: none"> <li>● 概念データモデル定義(To-Be)</li> </ul>
手順	
<ol style="list-style-type: none"> <li>1. 変更対象情報の特定 モデル化した業務プロセスや業務ルールから、現行業務に対する追加・変更・廃止される業務を洗い出し、その業務で必要または不要になる情報を特定する。</li> <li>2. 変更対象情報の整理 変更対象情報から、変更対象となる概念エンティティ・情報項目・エンティティ間の関連を整理する。</li> <li>3. 概念データモデル(To-Be)の作成 「G1-01-03 現行業務データの調査」で作成した現行の概念データモデルに”手順 2”で整理した内容を反映する。</li> <li>4. 概念データモデル(To-Be)の検証 作成した概念データモデルに対して、変更部分と既存部分の関連や多重度に矛盾・不整合・漏れがないかを検証する。</li> </ol>	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
—	
適用技法	
<ul style="list-style-type: none"> <li>● ER 図</li> <li>● クラス図</li> </ul>	

## G2-02-05 システム機能のリストアップ

アクティビティ概要	
システム要件の検討対象を明確化するために、モデル化した業務要求からシステム機能を導出し、一覧化する。	
インプット	アウトプット
<ul style="list-style-type: none"> <li>● システム機能一覧(As-Is)</li> <li>● CRUD(As-Is)</li> <li>● 業務俯瞰図(To-Be)</li> <li>● 業務フロー(To-Be)</li> <li>● 業務ルール定義(To-Be)</li> <li>● 状態遷移モデル定義(To-Be)</li> <li>● 概念データモデル定義(To-Be)</li> </ul>	<ul style="list-style-type: none"> <li>● システム機能一覧(To-Be)</li> <li>● CRUD(To-Be)</li> </ul>
手順	
<p>※現行システムが存在しない新規開発の場合は、このタイミングでシステム機能一覧を作成する。</p> <ol style="list-style-type: none"> <li>1. システム UI 機能(画面、帳票)の抽出 業務フローや状態遷移モデルから現行システムに対するシステム UI 機能(画面、帳票)の追加・変更・廃止を洗い出し、「G1-02-01 現行システムの調査」で作成したシステム機能一覧に反映する。</li> <li>2. 外部インターフェース機能の抽出 業務俯瞰図や業務ルール定義などから現行システムに対する外部システムとのインターフェース機能の追加・変更・廃止を洗い出し、「G1-02-01 現行システムの調査」で作成したシステム機能一覧に反映する。</li> <li>3. バッチ機能の抽出 業務ルール、外部インターフェース(授受するファイル)、イベントやシステム UI 機能の実現方法の観点からバッチ機能を洗い出し、「G1-02-01 現行システムの調査」で作成したシステム機能一覧に反映する。</li> <li>4. システム機能とデータの関連を整理する。 抽出したシステム機能とデータとの関連を「G1-02-01 現行システムの調査」で作成した CRUD に反映し、データライフサイクル視点での矛盾(Create がないのに Read がある、など)がないことを確認する。 システム機能が漏れていた場合は、システム機能一覧への反映とそのシステム機能を利用する業務作業の漏れがないか確認する。 概念エンティティが漏れていた場合は、概念データモデルに反映する。</li> </ol>	
【重要】上手に進めるためのポイント、注意事項	
全般	
<ul style="list-style-type: none"> <li>➤ システム機能の抽出方法を検討する。 「業務を可視化したモデルからシステム機能を抽出する」ことが原則であるが、それだけでは不十分であるため、それを補う抽出方法を検討する。 例えば、バッチ機能をどう抽出するか、など</li> <li>➤ システム機能は、プログラムの単位ではなく業務上意味のある機能単位で洗い出す。</li> </ul>	
上手に進めるためのポイント、注意事項	
システム UI 機能(画面、帳票)の抽出	
<ul style="list-style-type: none"> <li>➤ 「G2-02-02 業務プロセスのモデル化」で業務フローに追加・変更・廃止するシステム機能を反映する際に、システム機能一覧にも合わせて反映する。</li> </ul>	
外部インターフェース機能の抽出	
<ul style="list-style-type: none"> <li>➤ システムが保有するデータを接続先システムで活用するような、ニーズが外部システム側にある場合は、ステークホルダー分析や業務俯瞰図、要求から対象外部システムを特定し、外部インターフェース機能を抽出する。</li> </ul>	
バッチ機能の抽出	
<ul style="list-style-type: none"> <li>➤ 以下のような多面的な視点で、バッチ機能を漏れなく抽出する。</li> </ul>	

No	抽出元	抽出方法
1	外部インターフェース	システム機能一覧に定義した外部インターフェース機能に対して、バッチ機能の可否を検討する。
2	業務イベント	業務フローや状態遷移モデル定義に定義された業務イベントに対して、バッチ機能の可否を検討する。
3	システム UI 機能	システム UI 機能(画面、帳票)の実現に必要な概念データを抽出し、その概念データを作成するバッチ機能の可否を検討する。
4	業務ルール	業務ルール定義に定義された業務ルールの、実装対象機能を検討する。

適用技法
<ul style="list-style-type: none"> <li>● (特になし)</li> </ul>

## G2-03 業務要求の優先順位付け

サブプロセス目的
限られたプロジェクトリソース(コスト、期間、人など)内で実現すべき業務要求をステークホルダー間で調整・合意するため、業務要求の優先順位を明らかにする。
サブプロセス概要
業務要求の優先順位付けとは、整理・分類された業務要求に対して相対的な優先順位をつける作業である。 ここでは計画した優先順位付けの基準・方法に則り、各業務要求の優先順位付けを行う。

## G2-03-01 業務要求の優先順位付け基準・方法の見直し

アクティビティ概要	
要件定義計画で設定した要求の優先順位付け基準・方法に問題がないか確認し、必要に応じて見直しを行う。	
インプット	アウトプット
<ul style="list-style-type: none"> <li>要件定義計画書</li> <li>業務要求一覧(要求確認済み)</li> </ul>	<ul style="list-style-type: none"> <li>要件定義計画書(優先順位付け基準・方法見直し済み)</li> </ul>
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<p>※「要件定義計画プロセスガイド：C2-02-03 優先順位付け基準・方法の設定」の「【重要】上手く進めるためのポイント、注意事項」も参照。</p> <p>➤ 要件定義を進める過程で、「優先順位付けの基準や方法」で考慮すべき新たな価値観や要求属性が明らかになった場合、ステークホルダーと協議した上で、同基準・方法を再検討する。 例えば、ビジネス要件実現に欠かせない根幹業務が明確となったため、当該業務に関連する要求を特定する管理属性を追加し、該当する要求は無条件にトッププライオリティを付ける、など</p>	
上手く進めるためのポイント、注意事項	
※「要件定義計画プロセスガイド：C2-02-03 優先順位付け基準・方法の設定」の「上手く進めるためのポイント、注意事項」を参照。	
適用技法	
<ul style="list-style-type: none"> <li>プライオリティ方式</li> <li>優先順位付けマトリクス</li> <li>100ドルテスト</li> <li>イエス／ノー投票</li> <li>4 象限方式(例: 重要度×緊急度)</li> <li>MoSCoW(Must、Should、Could、Won't)</li> </ul>	

## G2-03-02 業務要求の優先順位付け

アクティビティ概要	
設定した基準・方法で業務要求の優先順位付けを行う。	
インプット	アウトプット
<ul style="list-style-type: none"> <li>要件定義計画書(優先順位付け基準・方法見直し済み)</li> <li>業務要求一覧(要求確認済み)</li> <li>業務モデル(To-Be)の成果物</li> </ul> ※「G2-02 業務要求のモデル化」で作成	<ul style="list-style-type: none"> <li>業務要求一覧(優先順位付け済み)</li> </ul>
手順	
1. 業務要求の管理属性値の再設定 「G2-03-01 業務要求の優先順位付け基準・方法の見直し」に伴い、管理属性値を再設定する。 2. 業務要求の優先順位付け 各要求の管理属性値を元に、要件定義計画で定めた方法に基づいて、優先順位付けを行う。	
【重要】上手く進めるためのポイント、注意事項	
※「要件定義計画プロセスガイド：C2-02-03 優先順位付け基準・方法の設定」の「【重要】上手く進めるためのポイント、注意事項」を参照。	
上手く進めるためのポイント、注意事項	
※「要件定義計画プロセスガイド：C2-02-03 優先順位付け基準・方法の設定」の「上手く進めるためのポイント、注意事項」を参照。	
適用技法	
<ul style="list-style-type: none"> <li>プライオリティ方式</li> <li>優先順位付けマトリクス</li> <li>100ドルテスト</li> <li>イエス／ノー投票</li> <li>4 象限方式(例: 重要度×緊急度)</li> <li>MoSCoW(Must, Should, Could, Won't)</li> </ul>	



## G2-04 業務要求の実現対象決定

サブプロセス目的
優先順位付けした業務要求とプロジェクトリソース制約(コスト、期間、人)等を考慮し、実現対象の業務要求を明らかにする。
サブプロセス概要
業務要求リストの要求内容に認識齟齬がないことをステークホルダーに確認する。その上で、業務要求を優先順に並び替え、プロジェクトリソース制約(コスト、期間、人)の範囲内に収まる実現対象の業務要求を選定し、要件定義計画で定めたステークホルダーと合意する。

## G2-04-01 業務要求の実現対象決定

アクティビティ概要	
業務要求内容をステークホルダーに説明し、適切であることを確認する。その上でプロジェクトリソース制約に収まるよう、優先順位をベースに実現対象の業務要求を選定し、ステークホルダーと合意する。	
インプット	アウトプット
<ul style="list-style-type: none"> <li>● 業務要求一覧(優先順位付け済み)</li> <li>● 業務モデル(To-Be)の成果物 ※「G2-02 業務要求のモデル化」で作成</li> </ul>	<ul style="list-style-type: none"> <li>● 業務要求一覧(実現対象合意済み)</li> </ul>
手順	
<ol style="list-style-type: none"> <li>1. 業務要求内容の変動ポイントのステークホルダー確認 業務要求一覧レベルで、当初要求をベースラインとした業務要求内容の変動ポイント概要をステークホルダーに説明し、内容の妥当性をご確認頂く。</li> <li>2. 業務モデル(To-Be)のステークホルダー確認 一定粒度の業務単位、または単一・複数の業務要求単位などで、要求を具体化した業務プロセス・業務ルール・業務データ・システム機能一覧をステークホルダーに説明し、内容の妥当性をご確認頂く。</li> <li>3. 優先順位付けのステークホルダー合意 要件定義計画で定めたステークホルダーを集め、優先度別に分類された各要求を確認し、各要求の優先度のステークホルダー合意を得る。</li> <li>4. システム要件定義工程以降の概算見積もり システム機能一覧をインプットに、システム要件定義以降の工数・コスト・期間の概算見積もりを行う。 非機能要件定義やインフラ要件など未確定の要素を除外または仮定し、可能な限りの概算見積もりを行うことで、コスト等の増加問題の早期解消(手順5)を可能にする。</li> <li>5. 実現対象の業務要求のステークホルダー合意 プロジェクトリソース制約と、手順4のシステム要件定義工程以降の概算見積もりを突き合わせて、要求の実現性をステークホルダーと確認する。 プロジェクトリソース制約に収まらない場合は、要求の取り下げや要求内容の軽量化などの対応をステークホルダーと協議し、合意した要求内容や実現対象を業務要求一覧に結果を反映する。</li> </ol>	
【重要】上手く進めるためのポイント、注意事項	
<p>➤ 後続工程での要件変更リスクを軽減するため、合意に求められる知識・能力・権限を保有するステークホルダーのキーマンから組織やプロジェクトのルールに則り、合意を得ることが重要である。</p>	
上手く進めるためのポイント、注意事項	
<p>➤ 全ての業務要求が、一定の優先順位付け基準・方法で設定した優先度に基づいて合意されるとは限らない。 個別の理由・事情で、優先順位付け基準・方法が無視した優先度を設定することもあり得る。このような場合は、判断理由を明確にした上で、記録に残す。また、例外的判断が過度に多い場合は、優先順位付けの基準・方法の不備の可能性もあり得る。 このような場合は、判断理由などを参考に優先順位付け基準・方法の見直しを行い、優先順位付けの再実施を検討する。</p>	
適用技法	
<ul style="list-style-type: none"> <li>● (特になし)</li> </ul>	

## G3 業務要件定義書の作成

### G3-01 業務要件定義書の作成

サブプロセス目的
ステークホルダーからの合意とお客さまの承認者(主にプロジェクトオーナー)からの承認を得た業務要件をもとにシステム要件定義を実施するため、ベースラインとする業務要件定義書を作成する。
サブプロセス概要
作成済のモデル等を活用して、実現対象の業務要求を業務要件として業務要件定義書にまとめる。

## G3-01-01 業務要件定義書の作成

アクティビティ概要																																						
実現対象の業務要求(＝業務要件)を対象に、これまで検討してきたモデルを活用して業務要件定義書にまとめる。																																						
インプット		アウトプット																																				
<ul style="list-style-type: none"><li>● 業務要求一覧(実現対象合意済み)</li><li>● 業務モデル(To-Be)の成果物 ※「G2-02 業務要求のモデル化」で作成</li></ul>		<ul style="list-style-type: none"><li>● 業務要件定義書</li></ul>																																				
手順																																						
<div>1. 業務要件定義書の目次構成・内容骨子を纏め、ステークホルダーと合意する。</div> <div>2. これまでに作成した業務モデルの成果物を業務要件定義の構成(目次)に合わせて整理し、業務要件定義書として完成させる。</div>																																						
【重要】上手く進めるためのポイント、注意事項																																						
要件定義書の作成																																						
<div>➤ 要件定義書は、異なる役割を持つステークホルダー間のコミュニケーションツールの役割を果たす。 要件を漏れなく定義することも必要であるが、ステークホルダー間の認識齟齬を避ける表現方法(*1)・詳細度にする必要がある。</div> <div>(*1):要件の表現方法は、以下の通り。</div> <table><tr><th>No</th><th>分類</th><th>内容</th></tr><tr><td>1</td><td>自然言語</td><td>十分に構造化して、使用する用語や文章の明確さなどに注意を払った自然言語で表現する。</td></tr><tr><td>2</td><td>モデル</td><td>プロセス、状態変化、静的/動的な関係、などを表すモデルで表現する。</td></tr><tr><td>3</td><td>自然言語＋モデル</td><td>多くの場合、自然言語での定義を、モデルで補完する形で表現する。</td></tr></table> <div>➤ 要件を読み易く定義するためのポイントは、以下の通り。</div> <table><tr><th>No</th><th>要件を読み易く定義するためのポイント</th></tr><tr><td>1</td><td>必要な情報全てが盛り込まれ、構造的に整理された、適切なテンプレートを使用する。</td></tr><tr><td>2</td><td>読者が必要な情報を見つけやすい文書構造とし、目次をつける。</td></tr><tr><td>3</td><td>章・節・項に、構造と整合した統一感があるラベルを付ける。</td></tr><tr><td>4</td><td>太字・下線・斜体・文字色・フォントは、要件定義書全体で統一した使い方をしよう、文書標準などでルールを定義しておく。</td></tr><tr><td>5</td><td>図、表には、番号と表題をつけ、他からは番号で参照する。</td></tr><tr><td>6</td><td>可能な限りモデルを使用して、要件を表現する。</td></tr><tr><td>7</td><td>統一された用語を使用する。</td></tr><tr><td>8</td><td>すべての要件に、一意で永続的な識別子を付与する。 ※要件変更時や上位要件・下位要件とのトレーサビリティ検証などで、対象の要件を一意に特定できるようになる。</td></tr><tr><td>9</td><td>自然言語を使う場合は、主語、述語、目的語、修飾語の対象を明確にする。 ※一般的に、受動態で書くより、能動態で書くほうが、アクターやトリガーが明確になりやすい。</td></tr><tr><td>10</td><td>自然言語を使う場合は、文章は短く簡潔に、適切な句読点を用いて、ひとつの要件のみを説明する。 箇条書き、表なども活用する。</td></tr><tr><td>11</td><td>曖昧な用語の使用を避ける</td></tr></table> <div>『ソフトウェア要求 第3版』[2] 第10章を参考に作成</div>			No	分類	内容	1	自然言語	十分に構造化して、使用する用語や文章の明確さなどに注意を払った自然言語で表現する。	2	モデル	プロセス、状態変化、静的/動的な関係、などを表すモデルで表現する。	3	自然言語＋モデル	多くの場合、自然言語での定義を、モデルで補完する形で表現する。	No	要件を読み易く定義するためのポイント	1	必要な情報全てが盛り込まれ、構造的に整理された、適切なテンプレートを使用する。	2	読者が必要な情報を見つけやすい文書構造とし、目次をつける。	3	章・節・項に、構造と整合した統一感があるラベルを付ける。	4	太字・下線・斜体・文字色・フォントは、要件定義書全体で統一した使い方をしよう、文書標準などでルールを定義しておく。	5	図、表には、番号と表題をつけ、他からは番号で参照する。	6	可能な限りモデルを使用して、要件を表現する。	7	統一された用語を使用する。	8	すべての要件に、一意で永続的な識別子を付与する。 ※要件変更時や上位要件・下位要件とのトレーサビリティ検証などで、対象の要件を一意に特定できるようになる。	9	自然言語を使う場合は、主語、述語、目的語、修飾語の対象を明確にする。 ※一般的に、受動態で書くより、能動態で書くほうが、アクターやトリガーが明確になりやすい。	10	自然言語を使う場合は、文章は短く簡潔に、適切な句読点を用いて、ひとつの要件のみを説明する。 箇条書き、表なども活用する。	11	曖昧な用語の使用を避ける
No	分類	内容																																				
1	自然言語	十分に構造化して、使用する用語や文章の明確さなどに注意を払った自然言語で表現する。																																				
2	モデル	プロセス、状態変化、静的/動的な関係、などを表すモデルで表現する。																																				
3	自然言語＋モデル	多くの場合、自然言語での定義を、モデルで補完する形で表現する。																																				
No	要件を読み易く定義するためのポイント																																					
1	必要な情報全てが盛り込まれ、構造的に整理された、適切なテンプレートを使用する。																																					
2	読者が必要な情報を見つけやすい文書構造とし、目次をつける。																																					
3	章・節・項に、構造と整合した統一感があるラベルを付ける。																																					
4	太字・下線・斜体・文字色・フォントは、要件定義書全体で統一した使い方をしよう、文書標準などでルールを定義しておく。																																					
5	図、表には、番号と表題をつけ、他からは番号で参照する。																																					
6	可能な限りモデルを使用して、要件を表現する。																																					
7	統一された用語を使用する。																																					
8	すべての要件に、一意で永続的な識別子を付与する。 ※要件変更時や上位要件・下位要件とのトレーサビリティ検証などで、対象の要件を一意に特定できるようになる。																																					
9	自然言語を使う場合は、主語、述語、目的語、修飾語の対象を明確にする。 ※一般的に、受動態で書くより、能動態で書くほうが、アクターやトリガーが明確になりやすい。																																					
10	自然言語を使う場合は、文章は短く簡潔に、適切な句読点を用いて、ひとつの要件のみを説明する。 箇条書き、表なども活用する。																																					
11	曖昧な用語の使用を避ける																																					
上手く進めるためのポイント、注意事項																																						
要件定義書の作成																																						

- 要件定義書に記述された要件は、基礎知識集の「要件が持つ特性」を満たしていることが理想的である。
- 文章校正支援ツールの適用も検討する。  
機械的なチェックや修正により、作業負荷軽減や作業品質向上効果が見込める。

#### 適用技法

- （特になし）

## G4 業務要件の検証・妥当性確認

### G4-01 業務要件の検証・妥当性確認

サブプロセス目的
定義した業務要件の検証および妥当性確認を行い、業務要件の品質を確保する。
サブプロセス概要
要件の検証とは、定義した要件が「要件の特性(単一性・完全性・一貫性等)」に照らして正しいことを確認することである。 要件の妥当性確認とは、定義した要件の実現を通じて、プロジェクト目的・目標を達成できるかを確認することである。 このサブプロセスでは、定義した業務要件に対して、検証と妥当性確認を行う。

## G4-01-01 検証・妥当性確認の基準・方法の見直し

アクティビティ概要	
要件定義計画時に設定した要件の検証・妥当性確認を行う基準・方法で問題ないかを確認し、必要に応じて見直しを行う。	
インプット	アウトプット
<ul style="list-style-type: none"> <li>● 要件定義計画書</li> <li>● 業務要件定義書</li> </ul>	<ul style="list-style-type: none"> <li>● 要件定義計画書(検証・妥当性確認の基準・方法の見直し済み)</li> </ul>
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<p>※「要件定義計画プロセスガイド：C2-02-04 検証の基準・方法の設定」、「要件定義計画プロセスガイド：C2-02-05 妥当性確認の基準・方法の設定」の「【重要】上手く進めるためのポイント、注意事項」も参照。</p> <p>➤ ここまでの要件定義活動を通して知り得た、プロジェクト成功に向けた要件定義品質の重要検証ポイントを踏まえて、計画していた要件定義成果物の検証・妥当性確認の基準・方法の見直し要否を検討する。 ※見直しを実施する場合の具体的な検証・妥当性確認の観点候補は、「技法ガイド(検証ガイド編)」および「技法ガイド(妥当性確認ガイド編)」を参照。</p>	
上手く進めるためのポイント、注意事項	
<p>※「要件定義計画プロセスガイド：C2-02-04 検証の基準・方法の設定」、「要件定義計画プロセスガイド：C2-02-05 妥当性確認の基準・方法の設定」の「上手く進めるためのポイント、注意事項」も参照。</p> <p>➤ 要件定義活動で新たに認知したステークホルダーに、検証の目的や参加の必要性などを説明し、検証活動参加の合意を得る。</p>	
適用技法	
<ul style="list-style-type: none"> <li>● (特になし)</li> </ul>	

## G4-01-02 業務要件の検証

アクティビティ概要	
設定した基準・方法で業務要件の検証を行う。 ※要求の漏れや誤り等が判明した場合は、「G1 業務要求の収集」プロセスに戻る。	
インプット	アウトプット
<ul style="list-style-type: none"> <li>要件定義計画書(検証・妥当性確認の基準・方法の見直し済み)</li> <li>業務要件定義書</li> </ul>	<ul style="list-style-type: none"> <li>業務要件定義書(検証済み)</li> <li>検証結果</li> </ul>
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
※「要件定義計画プロセスガイド：C2-02-04 検証の基準、方法の設定」の「【重要】上手く進めるためのポイント、注意事項」も参照。 ※検証方法・観点については、「技法ガイド(検証ガイド編)」を参照。	
検証の実施	
<ul style="list-style-type: none"> <li>➤ 検証の実施前に、要件定義計画で決めた各ステークホルダーの役割・責任・範囲などを改めて認識合わせをする。毎回のレビューミーティングの始めで、簡易な確認を行う。</li> <li>➤ レビューミーティングを開く場合は、参加者にレビュー対象の事前確認と課題事項・指摘事項・質問事項の事前抽出を義務付ける。こうした準備がなければ、レビューは考えることにミーティング時間を使ってしまい、重要な課題を見逃すリスクが高まる。</li> <li>➤ 要件定義書の検証は、要件定義期間を通して、対象範囲や観点を分割した段階的な実施を検討する。分厚い要件定義書をまとめてレビューすると、レビューの集中力が持続できないなどの弊害によって、品質の高い検証の実現が難しくなる。</li> <li>➤ 要件定義を進める中で知り得た高リスクな部分(認識齟齬や実現性が疑われる部分など)は、確認ポイントを整理し、重点的にレビューを行う。逆に低リスクの部分は一部のステークホルダーによる査読などで軽量のレビューに留めて、メリハリをつける。リスク影響度・顕在化度のレベルは、お客さまとベンダーで認識を合わせる。(一般的なリスク管理の範疇なので説明省略)</li> </ul>	
上手く進めるためのポイント、注意事項	
※「要件定義計画プロセスガイド：C2-02-04 検証の基準、方法の設定」の「上手く進めるためのポイント、注意事項」も参照。	
全般	
<ul style="list-style-type: none"> <li>➤ 要件の見落としがないことやステークホルダーからの要件追加の終わりを確実に知る方法はない。「完璧な要件」を目指すよりも、後続工程での手戻りリスク抑制を目的に、プロジェクト特性に合ったアプローチで要件内容の品質を高める。例：法令対応案件で関係法規との整合性に注力、複雑度が高い業務限定で全ステークホルダー参加のウォークスルーを実施</li> </ul>	
検証の実施	
<ul style="list-style-type: none"> <li>➤ 要件定義計画に定めたレビュー計画を超える、多数のレビュー参加者が集まる場合は、レビューを効率的に行うために、以下のような方法を検討する。               <ul style="list-style-type: none"> <li>✓ 参加者の立場を確認した上で、1つの立場から1名のレビューアに絞り込みを行う。</li> <li>✓ 複数のチームに分け、レビュー対象を分担し、最後に指摘事項のすり合わせを行う。など</li> </ul> </li> <li>➤ 定義した要件に対して、ステークホルダーとの認識齟齬が発生した場合は、認識差異が発生した根本原因分析と対処も実施する。例えば、ステークホルダーの要求内容変化が原因で生じた要件内容認識齟齬は、指摘箇所の修正だけでなく、「G1 業務要求の収集」まで立ち戻った検討作業が必要になる。</li> </ul>	



適用技法
<ul style="list-style-type: none"><li>● ウォークスルー</li><li>● チェックリスト</li></ul>

## G4-01-03 業務要件の妥当性確認

アクティビティ概要	
設定した基準・方法で業務要件の妥当性確認を行う。 ※要求の妥当性に問題がある場合は、「G1 業務要求の収集」プロセスに戻る。	
インプット	アウトプット
<ul style="list-style-type: none"> <li>要件定義計画書(検証・妥当性確認の基準・方法の見直し済み)</li> <li>業務要件定義書(検証済み)</li> </ul>	<ul style="list-style-type: none"> <li>業務要件定義書(妥当性確認済み)</li> <li>妥当性確認結果</li> </ul>
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
※「要件定義計画プロセスガイド：C2-02-05 妥当性確認の基準、方法の設定」の「【重要】上手く進めるためのポイント、注意事項」を参照。 ※妥当性確認方法・観点については、「技法ガイド(妥当性確認ガイド編)」を参照。	
上手く進めるためのポイント、注意事項	
※「要件定義計画プロセスガイド：C2-02-05 妥当性確認の基準、方法の設定」の「上手く進めるためのポイント、注意事項」を参照。	
適用技法	
<ul style="list-style-type: none"> <li>ロジックツリー(上位要件と下位要件との関連性を表現するために有効)</li> </ul>	

## G4-02 業務要件のお客さま合意と承認

サブプロセス目的
定義した業務要件をステークホルダーと合意し、要件定義計画に定めたお客さまの承認者(主にプロジェクトオーナー)から承認を得ることで、後続工程以降のベースラインを確定する。
サブプロセス概要
プロジェクトの目的・目標を達成できる業務要件が、漏れや誤りなく業務要件定義書に記述されていることをステークホルダーに確認して頂き、ステークホルダーから合意を得る。 また、ステークホルダーと合意した業務要件定義書に対して要件定義計画に定めたお客さまの承認者(主にプロジェクトオーナー)から承認を得る。

## G4-02-01 業務要件のお客さま合意と承認

アクティビティ概要	
プロジェクトの目的・目標を達成できる業務要件が、漏れや誤りなく業務要件定義書に記述されていることをステークホルダーと確認・合意し、要件定義計画に定めたお客さまの承認者(主にプロジェクトオーナー)から承認を得る。	
インプット	アウトプット
<ul style="list-style-type: none"> <li>業務要件定義書(妥当性確認済み)</li> </ul>	<ul style="list-style-type: none"> <li>業務要件定義書(承認済み)</li> <li>申し送り事項一覧</li> <li>合意記録</li> <li>承認記録</li> </ul>
手順	
<ol style="list-style-type: none"> <li>業務要件定義の活動の中で合意に至らず、システム要件定義活動へ申し送りとした事項、課題を、申し送り事項一覧に整理する。</li> <li>計画に定めたステークホルダーと、業務要件定義書内容およびシステム要件定義へ申し送りする課題と対応を合意する。</li> <li>ステアリングコミッティなどを公式な承認の場とし、事前に関係者へ説明を行い、内諾を得る。</li> <li>計画に定めた承認者(主にプロジェクトオーナー)に、“手順 2”で合意済みの業務要件定義書を以って、業務要件定義工程を終了すること、および業務要件のベースラインに設定し、変更管理を開始することを承認してもらう。</li> </ol>	
【重要】上手く進めるためのポイント、注意事項	
<p>全般</p> <ul style="list-style-type: none"> <li>合意・承認をしないステークホルダーがいる場合は、合意・承認の障壁となっている課題を特定し、解決する。</li> </ul> <p>業務要件の申し送り事項の整理</p> <ul style="list-style-type: none"> <li>申し送り候補の課題は、申し送りとする理由や内容を精査し、影響範囲や影響度合いやリスクを見極めて妥当性を判断する。工程 Exit 判定通過が過度に優先され、重要な業務要件の合意を保留し、申し送り課題にされた結果、後続工程の工数見積り誤り、システム要件検討の遅れなどの問題に繋がるケースがある。</li> </ul> <p>業務要件の合意</p> <ul style="list-style-type: none"> <li>ステークホルダーの実質的な合意を得ずに要件定義を終了しない。要件に合意しない人に対する「いついつまでに返答がない場合は、要件に合意したものとみなします」というアプローチは、実質的な合意が得られていない。合意しない/できない理由、箇所、感じている不安などを確認し、そこにアプローチしたほうが良い。また、こうした状態が続きプロジェクトが停滞しないよう、要件を変更する手続き(要件変更管理)があることを伝え、現在の業務要件定義をベースラインとして、システム要件定義を開始することを承認してもらう。</li> </ul> <p>業務要件の承認</p> <ul style="list-style-type: none"> <li>ベースラインとする業務要件定義書のバージョンを明確にし、要件変更管理の運用を開始する。</li> </ul>	
上手く進めるためのポイント、注意事項	
<p>全般</p> <ul style="list-style-type: none"> <li>合意や承認の議事録を取り、その場で合意・承認内容、条件や課題を再確認し、合意する。会議後の回覧形式での確認ではタイムラグが発生する為、その場で議事録作成と確認を行う。</li> </ul> <p>業務要件の合意</p> <ul style="list-style-type: none"> <li>合意形成の元となる要件内容は、検証や妥当性確認の活動を通して、ステークホルダーと共有するのが望ましい。</li> </ul> <p>業務要件の承認</p> <ul style="list-style-type: none"> <li>要件変更管理ルールやツールが、お客さまと未合意の場合は、遅くとも要件定義の承認までに合意する。無秩序な変更要求の発生による要件ベースラインの混乱や非効率な管理などの悪影響が発生する。</li> </ul>	

適用技法
● （特になし）

- （特になし）

## G5 引き継ぎ

### G5-01 システム要件定義への引き継ぎ

サブプロセス目的
システム要件定義への引き継ぎが必要な事項を整理し、引き継ぎを行うことで、システム要件定義でのシステム要求の検討漏れや業務要件に対する認識齟齬が発生しないようにする。
サブプロセス概要
業務要件定義で解決せず先送りとした課題や、業務要件定義で導出されたシステム要求事項を、申し送り事項として整理する。 承認された業務要件と整理した申し送り事項を、システム要件定義担当者へ説明し、引き継ぎとする。

## G5-01-01 システム要件定義への引継ぎ準備

アクティビティ概要	
業務要件定義で解決せず先送りとした課題や、業務要件定義で導出されたシステム要求事項が、システム要件定義で漏れなく取り扱われるように整理する。	
インプット	アウトプット
<ul style="list-style-type: none"> <li>● 業務課題一覧</li> <li>● 申し送り事項一覧</li> </ul>	<ul style="list-style-type: none"> <li>● システム要求一覧</li> </ul>
手順	
<ol style="list-style-type: none"> <li>1. 業務要件定義の活動の中で、ステークホルダーから提示され、詳細な検討を保留していたシステム要求事項をシステム要求一覧に整理する。</li> <li>2. 「G4-02-01 業務要件のお客さま合意と承認」で作成した申し送り事項について、お客さま側担当者、経緯・背景、保留・申し送り理由、など、必要な属性情報が付いているか、確認する。</li> </ol>	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
全体	
<ul style="list-style-type: none"> <li>➤ 保留としていたシステム要求事項、先送りとした業務要件課題のいずれも、業務要件定義プロセスの過程で課題管理を運用しておき、業務要件定義終盤で課題の整理、集約を行う、のが現実的な進め方。</li> </ul>	
既出のシステム要求の整理	
<ul style="list-style-type: none"> <li>➤ ステークホルダー個人の主観に基づくシステム要求が多く含まれている場合は、上位の業務要件との関連を整理した上で、ステークホルダーと要求の取り下げを交渉することも検討すること。</li> </ul>	
適用技法	
<ul style="list-style-type: none"> <li>● （特になし）</li> </ul>	

## G5-01-02 システム要件定義への引継ぎ

アクティビティ概要	
業務要件定義とシステム要件定義で体制が異なる場合は、システム要件定義の担当者への申し送り事項の引継ぎを行う。	
インプット	アウトプット
<ul style="list-style-type: none"> <li>● 業務要件定義書(承認済み)</li> <li>● 申し送り事項一覧</li> <li>● システム要求一覧</li> </ul>	<ul style="list-style-type: none"> <li>● (特になし)</li> </ul>
手順	
<ol style="list-style-type: none"> <li>1. システム要件定義の担当者へ、承認された業務要件定義内容を説明する。 実現する業務仕様と、そのために必要とされるシステム機能を確実に引き継ぐ。</li> <li>2. システム要件定義の担当者へ、申し送りとした業務要件定義課題事項、既出のシステム要求事項を説明し、システム要件定義で確実に継続検討されるよう引き継ぎを行う。</li> </ol>	
【重要】上手く進めるためのポイント、注意事項	
<b>業務要件の引き継ぎ</b> <hr/> <ul style="list-style-type: none"> <li>➤ 取り下げになり業務要件定義書から除外した業務要件についても、システム要件定義担当者へ理由や経緯と合わせて引き継ぐ。 システム要件定義で再度検討対象に挙がるケースや、お客さまから業務要件の漏れとして問題視されるケースがあるため。</li> </ul>	
上手く進めるためのポイント、注意事項	
<b>業務要件の引き継ぎ</b> <hr/> <ul style="list-style-type: none"> <li>➤ 要件定義書の成果物体系上で、ビジネス要件から業務要件、システム要件へのトレーサビリティをどのように可視化しているかを説明する。 それによりシステム要件定義担当者が、業務要件定義から体系的にシステム機能およびシステム要求を把握できる。</li> <li>➤ キーマンの入れ替えがある際は、本サブプロセスを参考に十分な引き継ぎを行う。</li> </ul>	
適用技法	
<ul style="list-style-type: none"> <li>● (特になし)</li> </ul>	



## 参考文献

- [1] JISA REBOK 企画 WG 編（2011）『要求工学知識体系 第1版』株式会社 近代科学社
- [2] Karl Wiegers、Joy Beatty 著 渡部 洋子 翻訳（2014）『ソフトウェア要求 第3版』日経 BP 社
- [3] 神崎 善司 著（2013）『モデルベース要件定義テクニック』 株式会社 秀和システム
- [4] NTT ソフトウェアイノベーションセンタ、NTT データ 著（2015）『ビジネスルールを可視化する要件定義の図解術』日経 BP 社
- [5] 本園 明史 著（2004）『要件定義のチェックポイント427』株式会社 翔泳社