

ユーザのための 要件定義ガイド

第2版

要件定義を成功に導く128の勘どころ

ユーザのための要件定義ガイド

第2版

要件定義を成功に導く 128 の勘どころ

独立行政法人情報処理推進機構（IPA）
社会基盤センター（IKC）



目次

ガイド第2版の発行に際して	1
ユーザのための要件定義ガイド 第2版の発行に際して	2
はじめに	3
上流工程の強化に向けた IPA の取り組み	4
デジタル時代の本格化	9
改訂にあたって	10
本ガイドの構成と想定読者	11
第1章 背景	15
1.1 システム開発の現状	16
1.1.1 IT システムの変遷	16
1.1.2 既存システムの現状	19
1.1.3 システム開発の現状	21
コラム1 日本の IT が抱える課題	25
1.1.4 DX に見える今後求められるシステム開発	27
コラム2 DX 関連技術	32
1.2 要件定義をめぐる環境の変化	35
1.2.1 要件定義に求められる3つのこと	35
1.2.2 パラダイムシフト	37
1.2.3 重要度が上がる非機能要求	38
1.2.4 ますます難しくなっている要件定義	40
第2章 要件定義の問題認識	41
2.1 各ステークホルダが意識すべきこと	42
2.1.1 経営層が意識すべきこと	42
コラム3 システム化に際して経営者の果たすべき役割	47
2.1.2 業務部門が意識すべきこと	49
2.1.3 システム部門が意識すべきこと	52
2.1.4 ベンダ企業が意識すべきこと	54
2.2 要件定義の問題	57
2.2.1 問題の抽出と整理	57
2.2.2 要件定義の問題と解決のための勘どころ	60
(1) ビジネス要求定義の問題と解決のための勘どころ	60
(2) システム化要求定義における問題と解決のための勘どころ	64
(3) 要件定義マネジメントにおける問題と解決のための勘どころ	65

第3章 要件定義の全体像	73
3.1 要件定義の位置付け	74
3.2 要件定義プロセス	75
コラム4 ビジネス要求定義、システム化要求定義の主要タスク例	80
3.3 要件定義ドキュメント	82
3.4 用語定義	85
第4章 ビジネス要求定義(BR)における問題と解決の勘どころ	89
4.1 ビジネス要求の獲得(BR.1)	91
4.1.1 現状の把握(BR.1.1)	91
(1) 業務・システムの状況をさまざまな方法を活用して把握する	91
事例1 現行業務に関わる業務分掌規程などの有効利用	95
(2) 現状業務・システムを理解・共有する	100
4.1.2 問題・課題の抽出(BR.1.2)	101
(1) 適切な問題・課題の抽出・分析を行うために、ステークホルダの特性を理解する	101
(2) 問題の原因を分析し解決テーマ(課題)を抽出する	104
4.1.3 ゴールの抽出(BR.1.3)	107
(1) 経営レベルの目的・目標を明確にして共通認識する	107
(2) 業務レベルの目的・目標を抽出する	110
4.1.4 手段の抽出(BR.1.4)	113
(1) 目的・目標を意識して手段を抽出する	113
コラム5 アイデア発想の方法	116
4.2 ビジネス要求の分析(BR.2)	118
4.2.1 要求の体系化(BR.2.1)	118
(1) 要求全体の整合性を分析し評価する	118
事例2 ビジネスに貢献する要求を見極め、ビジネス成果に直結するIT施策を明らかにする	121
(2) 具体的な測定尺度を検討する	125
4.2.2 要求の具体化(BR.2.2)	127
(1) 情報構造の観点から業務を可視化し、新しい業務として具体化する	128
(2) ビジネスプロセスの観点から業務を可視化し、新しい業務として具体化する	135
(3) 情報とプロセスの両方の観点から業務を可視化し整合性を確認する	140
4.2.3 優先順位付け(BR.2.3)	142
(1) 優先順位を明確にする	142
コラム6 要求の優先順位付けと合意形成	145

(2) 要求を棄却する.....	148
4.2.4 要求の交渉(BR.2.4)	149
(1) 経営層や業務部門への説明を行う.....	149
(2) 要求の絞込み交渉と合意形成は、ポイントを意識して行う.....	152
4.3 ビジネス要求の文書化(BR.3).....	155
4.3.1 ビジネス要求の文書化(BR.3.1)	155
(1) ドキュメントの全体像、関連を明確にする.....	155
第5章 システム化要求定義(SR)における問題と解決の勘どころ	157
5.1 システム化要求の仕様化(SR.1)	159
5.1.1 仕様化(SR.1.1).....	159
(1) ドキュメントの全体像、関連を明確にする.....	159
(2) 文章のあいまいさを排除し要求を正しく伝達する.....	161
事例3 図表を使った記述による要件定義文章の品質向上	165
5.2 システム化要求の確認・評価(SR.2).....	169
5.2.1 検証(SR.2.1)	169
(1) 組織的なレビューをする.....	169
事例4 要件定義全体を俯瞰したインスペクションによる手戻りの防止	174
事例5 定量的に捉えた手戻りコストとレビュープロセス 成熟度向上による手戻り改善	177
5.2.2 妥当性確認(SR.2.2)	185
(1) 業務部門が分かるように成果物の作成およびレビューを工夫する.....	185
第6章 要件定義マネジメント(RM)における問題と解決の勘どころ	189
6.1 立上げ(RM.1)	191
6.1.1 構想・企画の確認(RM.1.1)	191
(1) 企画内容を確認するとともに、企画の目的とプロジェクト目標の目的意識を醸成する	191
事例6 要件定義につながるシステム化構想・企画の策定	195
(2) 構想・企画の実現には、どの業務のビジネス要求が必要かを見定める... ..	198
6.1.2 ステークホルダの特定(RM.1.2)	200
(1) 必要なステークホルダを特定・分析し、各ステークホルダへの対応方針を立案する	200
コラム7 要件の種類と要求者	203
事例7 漏れのないステークホルダの特定と分析	205
6.1.3 オーナーの選定(RM1.3).....	209
(1) システムオーナーの選定により、業務部門の役割と責任を明確にする... ..	209
事例8 アプリケーションオーナー制度創設による 要件定義へのユーザ部門参加	

の強化	212
6.2 計画立案(RM.2)	216
6.2.1 要件定義プロセス計画(RM.2.1)	216
(1) 要件定義で必要なプロセスと成果物を見極め、要件定義の計画を立案する	216
事例 9 要件定義のプロセス・ドキュメントの全体像策定	221
コラム 8 成果物に記載する範囲を決める	224
6.2.2 スケジュール作成(RM2.2).....	225
(1) 所要期間の裏付けをできるだけ作り、スケジュール上のクリティカルパスに は、その旨を明示しておく	225
6.2.3 費用見積り(RM.2.3).....	229
(1) 要求の規模は件数ではなく実現にかかるコストで表現する.....	229
(2) 要件定義にかかるコストをそれまでの実績に基づき見直す.....	233
事例 10 要求の定量化と見積り前提条件の見える化による 要件定義工程の見積り	235
6.2.4 体制・チームビルディング(RM.2.4)	246
(1) プロジェクト目標の達成に向けて、体制を構築しチームビルディングを行う	246
事例 11 キーマンの時間が十分に確保できない	250
6.2.5 品質計画(RM.2.5)	254
(1) 成果物の品質は作成方法の指導と定期的な状況確認で保つ.....	254
6.2.6 コミュニケーション計画(RM.2.6).....	257
(1) ステークホルダとのコミュニケーションを計画し、合意形成を図る.....	257
事例 12 ステークホルダとの会議体設定による 確実なコミュニケーション ..	260
事例 13 要件定義を1カ月で実施した事例	262
事例 14 多様化するステークホルダとの合意形成	264
6.2.7 リスク認識(RM.2.7)	270
(1) リスクの特定と対応計画の立案により、プロジェクトの失敗を防ぐ.....	270
事例 15 「プロジェクト危ないことチェックシート」で 要件定義開始前にリスク 特定	273
6.2.8 調達計画と契約(RM.2.8)	276
(1) 要件定義は準委任契約を基本とし、ユーザ企業主体で進める.....	276
6.2.9 プロジェクト管理計画の作成(RM2.9).....	279
(1) プロジェクト管理計画(管理の仕組み)を事前に作成し、経営層や上位マネジ メント層を含む関係者と合意する	279
6.3 監視・コントロール(RM.3).....	281

6.3.1 要件量・スコープ(RM.3.1).....	281
(1) 膨らむ要求を適正量に導く.....	281
事例 16 要件量を可視化する.....	286
事例 17 要件定義工程生産物の出来高管理による膨らむ要求の制御.....	289
(2) 実現する要求は関係者が納得できる指標をもとに選択する.....	292
(3) 決定済の事項を大きく変更することが必要になったら、プロジェクト全体での意思決定により採否を決定する.....	294
6.3.2 スケジュールコントロール (RM3.2).....	297
(1) 細かいチェックポイントの設定と、事前に遅延対策を関係者に周知しておく.....	297
事例 18 To-Be の業務改善案の作成が遅れる (1).....	300
事例 19 To-Be の業務改善案の作成が遅れる (2).....	305
6.3.3 品質(RM.3.3).....	308
(1) 抜け漏れを抑えた要件定義成果物の作成とプロセスを監視する.....	308
事例 20 要件定義ドキュメントの形式品質.....	311
6.3.4 コミュニケーション(RM.3.4).....	314
(1) 仕様がなかなか決まらないときは実態を確認して対処する.....	314
6.3.5 リスク(RM.3.5).....	317
(1) リスクはプロジェクト終結までフォローする.....	317
事例 21 「リスク管理リスト」によるリスク管理.....	319
6.3.6 トレーサビリティ(RM.3.6).....	321
(1) ビジネス要求とシステム化要求の関係は後で追跡できるように台帳で管理する.....	321
コラム 9 モデリング言語の適用によるトレーサビリティの確保.....	323
事例 22 要件定義・システム開発・保守管理リポジトリによる 要件定義ドキュメントの相互関連性の管理.....	324
6.3.7 見積り作成・投資効果分析(RM3.7).....	327
(1) 投資側には保守運用費を入れ、投資効果側は定性的な効果の客観性がポイント.....	327
事例 23 経営者に投資効果を分かりやすく説明する.....	329
6.4 終結(RM.4).....	335
6.4.1 要求評価(RM.4.1).....	335
(1) 決まらない要件は影響を評価し、対応策を検討する.....	335
(2) 内容に自信が持てないならそこには何らかの問題がある、思い切ってやめることも時には必要.....	338
6.4.2 完了判断(RM.4.2).....	340

(1) 実施すべき作業がすべて計画どおり実施されたことを確認する.....	340
事例 24 経営者の投資判断を支援する.....	342
事例 25 要件定義の完了判断が難しい.....	345
(2) 要件未決のまま次プロセスに進む必要があるなら相応の対策を立てておく	348
第7章 要件定義の主要ドキュメント作成(DD)の勘どころ.....	351
7.1 ビジネスコンセプトを確認する(DD.1).....	353
7.1.1 ビジネスコンセプト確認ドキュメント(DD.1.1).....	353
7.2 ステークホルダを漏れなく、正確に記述する(DD.2).....	357
7.2.1 ステークホルダ関連図(DD.2.1).....	357
7.2.2 ステークホルダー一覧(DD.2.2).....	359
7.2.3 リッチピクチャ(DD.2.3).....	361
7.3 重要な要求を漏らさず、正確に記述する(DD.7.3).....	363
7.3.1 問題・ニーズ・課題一覧(DD.3.1).....	363
7.3.2 問題原因分析図(DD.3.2).....	365
7.3.3 要求構造図(DD.3.3).....	367
7.3.4 要求一覧(DD.3.4).....	369
7.4 データ構造を漏れなく、正確に記述する(DD.4).....	372
7.4.1 管理対象分類図(DD.4.1).....	372
7.4.2 概念データモデル(ER図)(DD.4.2).....	374
7.5 ビジネスプロセスを漏れなく、正確に記述する(DD.5).....	379
コラム 10 フロー系成果物のプロセスの粒度を決める.....	380
7.5.1 ビジネスプロセス関連図(DD.5.1).....	382
7.5.2 業務機能構成表(DD.5.2).....	385
7.5.3 ビジネスプロセスフロー(業務フロー)(DD.5.3).....	387
7.5.4 ビジネスプロセスフロー(システム化業務フロー)(DD.5.4).....	391
7.5.5 業務処理定義(DD.5.5).....	395
コラム 11 要求とその仕様の記述 USDM.....	398
7.6 相互作用のモデルを漏れなく、正確に記述する(DD.6).....	399
7.6.1 状態遷移図(DD.6.1).....	399
7.7 コミュニケーション資料を作成する(DD.7).....	401
7.7.1 業務用語定義(DD.7.1).....	401
7.7.2 ビフォーアフター図(DD.7.2).....	403
7.8 一覧系成果物で要求の抜け漏れを防ぐ(DD.8).....	405
7.8.1 システム化業務一覧(DD.8.1).....	405
7.8.2 画面一覧(DD.8.2).....	408

7.8.3 帳票一覧(DD.8.3)	411
7.8.4 外部インターフェース一覧(DD.8.4)	412
7.8.5 エンティティ一覧(DD.8.5).....	414
7.9 インターフェースの確認を中心に要求の妥当性、実現性を確認する(DD.9)	416
7.9.1 システム化要求仕様(DD.9.1)	416
7.9.2 UI 標準(DD.9.2)	419
7.9.3 画面遷移図(DD.9.3).....	425
7.9.4 画面レイアウト(DD.9.4).....	428
7.9.5 帳票レイアウト(DD.9.5).....	432
7.10 データの洗い出しだけでなく、標準化や管理方法も含めて仕様化する(DD.10).....	435
7.10.1 エンティティ定義書／データ項目定義書(DD.10.1)	435
7.10.2 ドメイン定義書(DD.10.2)	438
7.10.3 コード体系定義書／コード内容定義書(DD.10.3)	440
7.11 機能とデータとの間の整合性を検証する(DD.11)	442
7.11.1 CRUD 図(DD.11.1).....	443
コラム 12 要件定義における CRUD 図の役割	445
7.12 十分で妥当な非機能要求を定義する(DD.12)	447
7.12.1 非機能要件書(DD.12.1)	447
事例 26 各ステークホルダが協力しながら進めていく 非機能要件定義の進め方	459
7.13 運用や移行、総合テストの要求も定義する(DD.13)	463
7.13.1 運用要件書(DD.13.1).....	463
7.13.2 全体移行計画書(DD.13.2)	465
7.13.3 総合テスト計画書(DD.13.3)	468
おわりに	471
おわりに	472
付録	473
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	474
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」	477
付録3 CMMI 1.3 と本ガイドの記載の関係	480
参考文献	483

本書の内容に関して

- 本書を発行するにあたって、内容に誤りのないようできる限りの注意を払いましたが、本書の内容を適用した結果生じたこと、また、適用できなかった結果について、著者、発行人は一切の責任を負いませんので、ご了承ください。
- 本書の一部あるいは全部について、著者、発行人の許諾を得ずに無断で転載、複写複製、電子データ化することは禁じられています。
- 本書に記載した情報に関する正誤や追加情報がある場合は、IPA/社会基盤センターのウェブサイトに掲載します。下記の URL をご参照ください。

独立行政法人情報処理推進機構（IPA）

社会基盤センター

<https://www.ipa.go.jp/ikc/index.html>

商標

- 本書に記載する会社名、製品名などは、それぞれの会社の商標もしくは登録商標です。本書の文中においては、これらの表記において商標登録表示、その他の商標表示を省略しています。

ガイド第 2 版の発行に際して

ユーザのための要件定義ガイド 第 2 版の発行に際して

本書はシステム構築の要件定義で必要な活動を起こりがちな問題と対応付けながら解説する。問題を意識しつつ活動を行うことにより、構築するシステムが必要とする品質要求を満たし、対象システムにより実現されるビジネスやサービスによる高い価値を得ることができる。結果として、開発プロジェクトの失敗を減らすことにつながる。

本書の導入として、読者各位が本書をよりよくご理解、ご活用いただくための情報を示したい。本書はシステム構築の上流工程において必要となる活動を解説する。他の開発活動と同じように要件定義を円滑に進めるためには、活動がどのような内容かを知ることと同じくらいその活動がなぜ必要かという理由の理解も重要である。各活動の目的や効果が分からないまま進めると、活動の実施が不十分となることがあるためである。各活動が必要になる理由は、ご自身の現場での実践により理解いただくことが最も効果が高いが、その補足となるように本書では執筆陣が実際に直面した問題を具体的に紹介する。さらに、問題、対策、活動を少し抽象化して読み進めていただくことにより、活動や対策の適用範囲が広がることを期待している。これにより、ご自身の現場で起こる問題との対応が取りやすくなったり打てる対策の幅が広がったりする。

要件定義とその実現をユーザとベンダで分担して進める場合がある。本書では、システム開発のステークホルダを前提として、ユーザ側の活動を中心に紹介し、なるべく多くの組織に通用するよう努めたが、ご自身の組織体制や開発の進め方と一致しない部分があるかもしれない。ご自身の開発において、開発を分担しなかったり、分担する組織体制が違ったりする場合には、それに合わせて適切に解釈いただければ幸いである。その際、先述の活動が必要な理由を意識いただくとより適切な解釈につながる。

スピードや IT 技術の活用がビジネスの成功に影響する割合は年々増している。ビジネスや業務を IT で支援する時代から、IT がビジネス価値を生み出す時代に変遷していることが指摘されて久しい。こうした時代に競争優位を維持し続けるためには、ステークホルダが要件定義の活動の役割を理解し、その効果が得られるように取り組むことが欠かせない。そのために、本書が役立てば幸いである。

2019 年 12 月

システム構築上流工程強化部会 システム化要求 WG
委員一同

はじめに

はじめに

上流工程の強化に向けた IPA の取り組み

独立行政法人情報処理推進機構 社会基盤センター（以下、IPA）は、2006年に「経営者が参画する要求品質の確保 ～超上流から攻める IT 化の勘どころ～」(第2版) [1]を発行し、IT システムの役割が現場中心からビジネス中心へ、個別最適から全体最適へ、と変化する時代に、経営層が IT システム開発の上流に深く関わることの重要性を発信した。それから10年以上が経過したが、上流工程の不備が原因での開発プロジェクトの失敗や運用後のシステムトラブルが減少してきたとは言い難い。ビジネスの要求が多様化して複雑に深化を続け、IT システムがそれまで予想もしなかった相手とつながるようになった今日では、IT システムに問題が発生した際の影響はより広範囲になり、社会に与えるインパクトが以前とは比較にならないほど甚大になっている。そのようなビジネス環境でこそ、「ステークホルダから要求(What)を漏れなく抽出し、内容を見極め、新たなビジネスやシステムを創造する要件として定義する」という、要件定義の取り組みの重要性がますます増加してくる。

IT システムの変遷に伴う課題について少し考察してみたい。

IT システムは、ビジネスの現場で実施されている業務を支援することから始まった。その後、IT システムは、技術の進展と時代の要請から企業の基幹事業や社会インフラなど、その担う役割が質、量ともに進化してきた。書類や口頭でのやり取りを置き換える導入初期や、その後の社内事務を効率化する本格的な活用時期には、提供する機能の品質が重視されていた。IT システムが新たなビジネスを創出し、企業価値を向上する経営戦略上の武器と考えられるようになり、自社の売上に直結するビジネス戦略上の要として利用されるようになった昨今では、それに加えて競争優位性を確保するために稼働までのスピードが重視されるシステムも増えてきた。

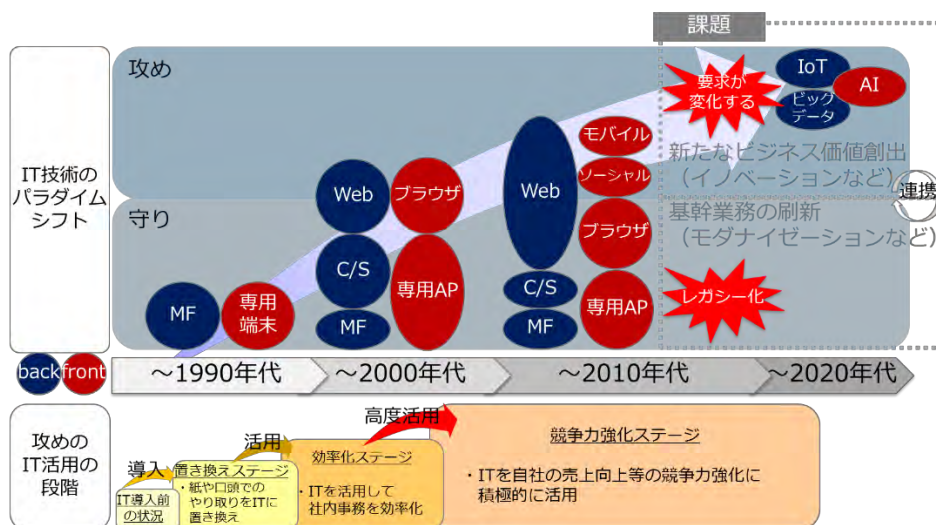


図 始.1 IT システムに関わる IT 技術とユーザ企業の現在

今日、ITシステムは、図 始.1 が示すように基幹業務に必要なデータの転送・蓄積・処理を中心とする従来の「守りの分野」と、顧客とのつながりによる協働を重視し、IoTやビッグデータ、AIなどの比較的新しい技術を活用する「攻めの分野」に分けて表現されるようになってきた。IT技術に関する進歩は目覚ましく、今後さらに加速することは明らかである。「守りの分野」と「攻めの分野」のそれぞれを強化すること、および両分野の連携が経営に直結する新たな課題になった。

攻めの分野では、図 始.2 が示すとおり、産業の垣根を越えて異なる分野の機器やシステムが連携し、目的に応じた「最適な組み合わせ」を導き出して新たなサービスを提供することが求められる。システム化における上流工程の重要性はさらに高まり、これまでのようにユーザ企業が責任を持って要件を定義する、という明確な線引きを行うだけではなく、ベンダ企業とともに協力して、多様なデータを収集、蓄積、解析してサービス化するサイクルを廻すことが重要である。

しかし、攻めの分野においては要求のすべてが開発段階で明らかになっていないことが多く、ITシステムにはサービス開始してから徐々に明らかになっていく要求への対応が常に求められる。使ってみて初めて分かる本来の要求を要件としての確にシステム化し、迅速にサービスとして市場へ提供し続けることが課題になる。

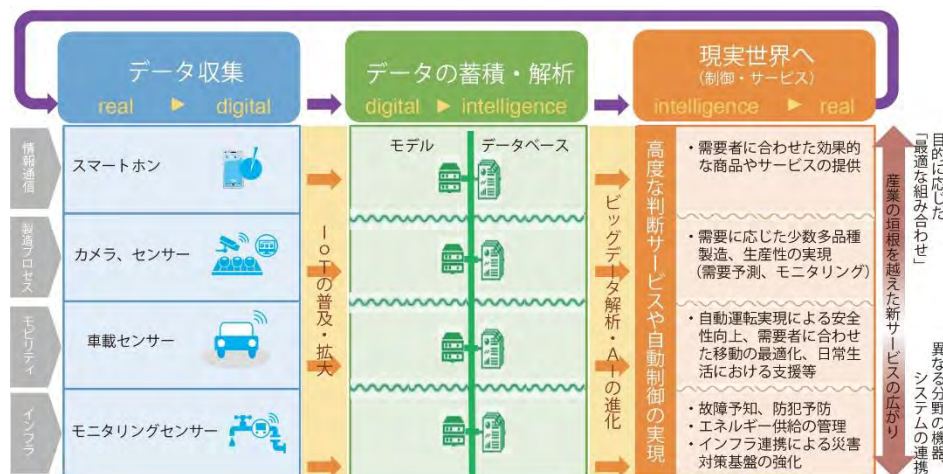


図 始.2 攻めの分野におけるシステム開発ライフサイクル (例)

(出典) 産業構造審議会 商務流通情報分科会 情報経済小委員会「中間取りまとめ～CPSによるデータ駆動型社会の到来を見据えた変革～」を基にIPAが作成

一方で、守りの分野である基幹系システムなどでは、開発完了後に長期間を経過し、システムを構築する際に使用したハード・ソフトや技術が時代遅れになっていった（レガシー化した）システムが増加している。守りの分野における課題であるレガシー化の解消には、図 始.3 が示すような、開発完了後の長年の保守で蓄積された「業務仕様の理解不足」などによる再構築の難しさという問題がある。下流工程においてコスト超過や稼働延滞などが発

はじめに

生ずる大きなリスクになるにも関わらず、モダナイゼーション（システム再構築）の実施前にそれを予測することは難しい。いかに安全確実にレガシー化を解消するかはユーザ企業、ベンダ企業が協力して取り組むべき優先課題であり、上流工程においてリスクを明らかにして、対策をユーザ企業とベンダ企業が合意し、ユーザ企業の経営層を含めてリスクヘッジすることが重要になる。



図 始.3 守りの分野におけるシステム再構築の事例

なお、攻めの分野に投資を拡大するには、支える側の守りが重要である。仮に業務仕様を変えずに基幹システムを再構築する場合でも、投資判断が難しい側面はあるが、最新の IT 技術を用いることによりセキュリティ強化や性能向上が図れるなど、得られる効果がある。経営層は、よりいっそうシステム開発の上流工程に関わり、「基幹業務システムの仕様は現行とおりで・・・」というようなあいまいな要求ではなく、現行とおりととは何か、どうしたいのかを明確に定義するよう、要件定義をリードすることが経営の競争力強化につながる。

また、上流工程強化の取り組みにおいては、図 始.4 に示すとおり、ソフトウェア・エンジニアリングにおける領域を大きく「基本領域」と「応用領域」の 2 つに分類した。基本領域は、システム開発における攻めの分野にも、守りの分野にも共通する内容として、プロセスやドキュメント、人、体制などを整理した領域である。一方、応用領域は、基本領域を取捨選択し、プロジェクト向けにテーラリングして適用する領域であり、攻めの分野であるイノベーションや、守りの分野であるモダナイゼーションなどである。

基本領域の一つである「要件定義」については、正しく伝えるための「How」を整理するという課題に取り組み、原初の要求からどのように要件を抽出し、文書に表現するか、どのように要件定義の不備（抜け、漏れ、あいまい等）をなくし、成果物（以下、このガイドでは、要件定義で作成するドキュメントのうち、アウトプットとしてシステム開発以降に引き継ぐものを“成果物”と表現する）の品質を上げるかの勘どころをとりまとめた。

また、応用領域である「モダナイゼーション」の領域では、上流工程におけるリスクを把握し、合意形成する手段を整理するという課題に取り組んだ。システム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した上で、要件定義に着手する前の企画・計画段階で、把握したリスクの下流の工程での影響について合意して、リスクを減らすための施策（やるべきこと）を明確化した。

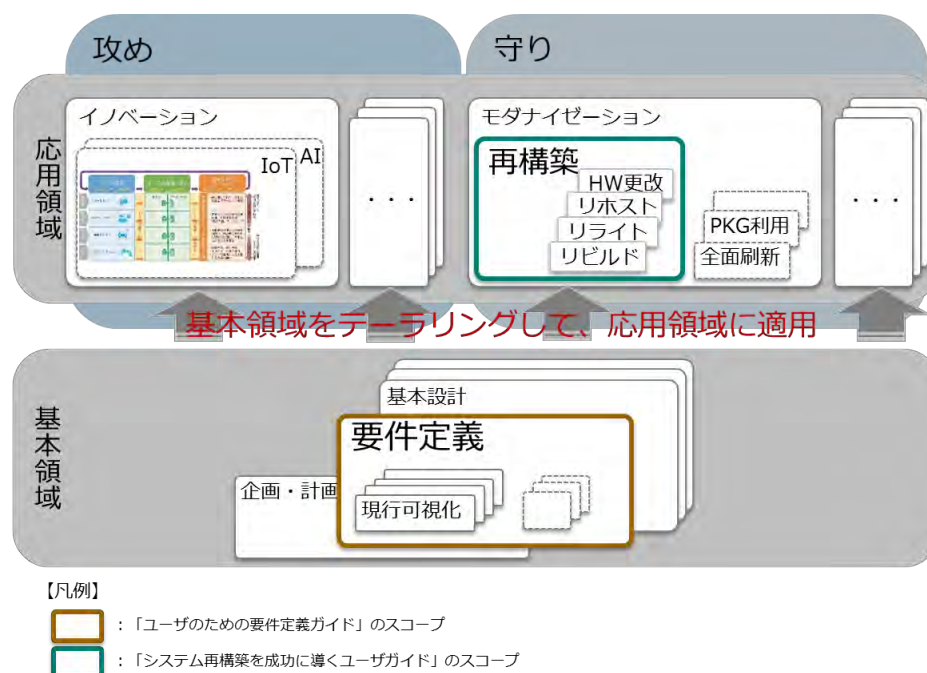


図 始.4 ソフトウェア・エンジニアリングの各領域における取り組み

基本領域では、要件定義をユーザ企業自身が行う際に、「ユーザのための要件定義ガイド 要件定義を成功に導く 128 の勘どころ」（以下、本ガイド）を用いて上流工程に起因する手戻りをなくすことにより、要件定義の品質向上が図れることが期待できる。ユーザ企業は部門間で要求を明確にして、ベンダ企業を含むステークホルダ間で理解し合い、要件定義のインプットとアウトプットの品質を高め、企業の競争力向上につなげる足掛かりとしていただきたい。

応用領域では、再構築において、稼働しているシステムから必要な情報はすべて引き出せるだろうと誤解して上流工程の作業の一部を省略した場合に陥りやすいリスクへの対策が必要になる。「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～」を用いることにより、リスク対策を保守・運用までを含めた計画に反映して、レガシー化した基幹システムの再構築に安心して取り組めることを期待している。そして、将来的にモダナイゼーションで得られる価値を得る足掛かりとしていただきたい。

基本領域と応用領域それぞれの取り組みの背景を図 始.5 に示す。

はじめに

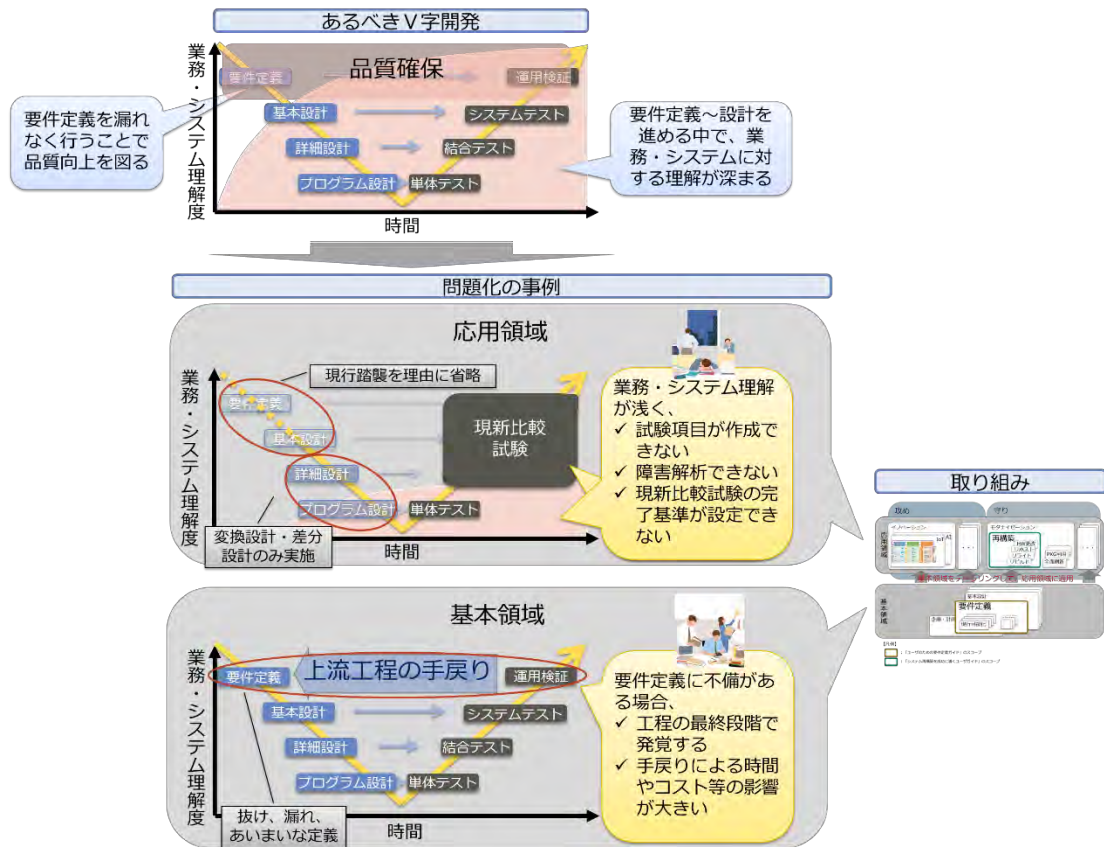


図 始.5 各領域に取り組む背景

なお、攻めの分野に特化した内容は、今回の内容に含まれていない。基本領域では、攻め、守り、いずれの分野のシステム開発にも共通的な内容を解決することから始めた。また、応用領域では、現在のシステムをそのまま再構築することを想定して内容を整理した。

例えば「イノベーション」などに代表される攻めの分野には、従来と異なるシステム構築手法に対応した新たな作業プロセスや勘どころの追加などが考えられる。また、「モダナイゼーション」などの守りの分野でも、全面刷新やパッケージ導入などの内容を整理することが求められる。両ガイドブックを踏まえて、今後の取り組みにつなげていきたい。

はじめに

デジタル時代の本格化

経済産業省が 2018 年に公開した DX(デジタルトランスフォーメーション)レポート [2] (以下、本ガイドでは「DX レポート」と記す) では、「あらゆる産業において、新たなデジタル技術を使ってこれまでにないビジネス・モデルを展開する新規参加者が登場し、ゲームチェンジが起きつつある。こうした中で、各企業は、競争力維持・強化のために、デジタルトランスフォーメーション(DX:Digital Transformation)をスピーディーに進めていくことが求められている。」と述べられている。このレポートでは、多くの経営者が DX の必要性を認識し、実証実験など具体的な取り組みを始めている一方で、その取り組みはまだ実際のビジネス変革にまだつながっていない状況が多く見られる、とも述べられている。

あらゆる人やものがインターネットを通じて膨大なデータでつながる「デジタル時代」を生き延びるためには、「デジタル時代に対応した新たなビジネスモデルの構築と価値創造」と「DX 実現を困難にしている既存 IT システムの再構築」が必要になる。

【デジタル時代に対応した新たなビジネスモデルの構築と価値創造】

経済産業省が同じく 2018 年に公開したデジタルトランスフォーメーションを推進するためのガイドライン [3] (以下、本ガイドでは「DX 推進ガイドライン」と記す) では、DX を「企業がビジネス環境の激しい変化に対応し、データとデジタル技術を活用して、顧客や社会のニーズを基に、製品やサービス、ビジネスモデルを変革するとともに、業務そのものや、組織、プロセス、企業文化・風土を変革し、競争上の優位性を確立すること。」と定義している。デジタル時代の IT システム構築においては、顧客と企業の双方の価値創造を推進するために、要件定義の実施内容を工夫して、このガイドラインに表現されるようなビジネスモデルの変革を実現する必要がある。

【DX 実現を困難にしている既存 IT システムの再構築】

DX レポートでは既存 IT システムについて、「DX を実行していくに当たっては、データを収集・蓄積・処理する IT システムが、環境変化、経営・事業の変化に対し、柔軟に、かつスピーディーに対応できることが必要である。」と述べる一方で、その現状について「老朽化・複雑化・ブラックボックス化した既存システムが、DX 推進のための足かせになっている。」と述べている。このような状態の既存システムを刷新し、DX にシフトしていくためには、自社の情報資産を正確に把握しなおし、最適な再構築手法を選択することが必要になる。

「ユーザーのための要件定義ガイド」および「システム再構築を成功に導くユーザガイド」は DX の 2 つのポイントである「デジタル時代に対応した新たなビジネスモデルの構築」と「DX 実現を困難にしている既存 IT システムの再構築」に強く関連している。DX を推進する際には、両ガイドが役立つことを期待する。

改訂にあたって

初版を公開後、IPA では有識者による上流工程強化のための検討やガイドの普及のためのセミナーを継続して実施してきた。そこで初版のガイドに対して以下の記事の追加をすべきとの意見・要望をいただいた。

- 業務上の要求を適切に分析してビジネス要求として定義するための手法
- 要件定義を推進する際のプロジェクトマネジメント上の問題への対処方法
- 要件定義を実施した際の事例

今回のガイドでは、上記指摘に対応して記載内容の大幅追加を行った。ビジネス要求定義の強化、要件定義マネジメントへのスコープの拡大、事例のさらなる追加を中心に加筆を行い、あわせてシステム化要求定義についても記載内容の見直しを行っている。

【ユーザが主体となって要件定義を実施するためのガイド】

本ガイドはタイトルにあるように「ユーザのための」要件定義ガイドである。

要件定義は、IT システムをビジネスに活用する当事者であるユーザが自ら中心になって実施すべきものである。要件定義は利用部門（ユーザ）の責任、要件定義とは「使える」業務システムを定義することといわれている。昨今は特に、ビジネスプロセスの改革や経営や業務に貢献する IT システムの構築が求められるようになり、ユーザの要件定義への係わり方が重要になってきている。しかし、単にユーザの責任で実施と言っても、ユーザは必ずしも要件定義に慣れているわけではないため、なかなか要件定義がうまくいかないことが多い。そこで本ガイドは、要件定義で何をするか（プロセス）、何を作るか（アウトプット）を明示するだけでなく、どのように考え、どのように作るか、そのときの留意点は何か（HowTo）を提示することを目指した。

【問題とその解決のための勘どころを提示】

本ガイドは、前述したように、HowTo を提示することを狙っている。問題を起点にした解決策の提示、分析などのプロセスを実施するときの考え方、ドキュメントを作成するときの留意事項、さらに解決事例も加えた、要件定義の実践におけるノウハウを集めたものになっている。

記載は、要件定義においてよく遭遇すると思われる問題と、その解決のための勘どころをペアで提示するという形式にした。掲載する問題は、本ガイドの編纂に参加いただいた各委員に、実際に経験した問題、一般的に想定される課題を列挙していただき、そこから選択した。紙面の都合上、重要と思われる事項を優先したため、網羅性については今後の課題としたい。読者の状況にそのまま適合するものばかりではないと思われるが、記載した勘どころをご自身の問題と照らし合わせて、参考になる部分を活用していただきたい。

本ガイドの構成と想定読者

「第1章 背景」

この章には、システムの現状やシステム開発の現状などを記載している。

「昨今要件定義の内容が変わってきている、要件定義の難易度が上がってきている」といった時代の変化、環境の変化などについて説明している。

「第2章 要件定義の問題認識」

この章には、経営層や業務部門、システム部門などのステークホルダが要件定義の問題として意識すべきことと、4章以降で説明する、要件定義を進めるにあたってよく発生する問題とその解決の勘どころの一覧を記載している。

本ガイドにおいてどのような問題が取り扱われているのかを俯瞰する際に利用していただきたい。

「第3章 要件定義の全体像」

この章には、上流工程における要件定義の位置付け、プロセス概要、各プロセスで使うドキュメントなど、要件定義の全体像を記載している。

また、要件定義工程の主要な成果物の関連図や基本的な用語の定義も提示している。本ガイドにおいてどのような成果物が取り扱われているのかを俯瞰する際に利用していただきたい。

「第4章 ビジネス要求定義(BR)における問題と解決の勘どころ」

この章には、ビジネス要求定義における問題、解決のための勘どころを記載している。

企画工程で整理された経営レベルの要求に基づき、利害関係者の要求事項として業務の観点から要求を獲得 (BR.1 ビジネス要求の獲得)、ビジネス要求の項目間の関連の整理・優先度付け・具体化 (BR.2 ビジネス要求の分析)、それらの文書化 (BR.3 ビジネス要求の文書化)のステップに分類して、それらを実施する際に発生する問題と解決の勘どころを提示している。

「第5章 システム化要求定義(SR)における問題と解決の勘所」

この章には、システム化要求定義における問題、解決のための勘どころを記載している。

「システム化要求定義(SR)」を、「BR. ビジネス要求定義」で定義した要求を機能要件・非機能要件の観点で仕様化 (SR.1 システム化要求の仕様化)、その仕様の検証や妥当性確認 (SR.2 システム化要求の確認・評価)のステップに分類して、それらを実施する際に発生する問題と解決の勘どころを提示している。

はじめに

「第6章 要件定義マネジメント(RM)における問題と解決の勘どころ」

この章には、要件定義マネジメントにおける問題、解決のための勘どころを記載している。

要件定義開始にあたっての確認・準備(RM.1 立上げ)、要件定義プロジェクトの計画(RM.2 計画立案)、「BR. ビジネス要求定義」や「SR. システム化要求定義」の活動中における監視・コントロール(RM.3 監視・コントロール)、要件定義工程の実行結果の評価と完了判断を行う終結作業(RM.4 終結)のステップに分類して、それらを実施する際に発生する問題と解決の勘どころを提示している。

「第7章 要件定義の主要ドキュメント作成(DD)の勘どころ」

この章には、要件定義において作成する主要なドキュメントの作成の勘どころを記載している。

ドキュメントの種類に分類し、各ドキュメントの記載項目とサンプルドキュメント、作成時の留意事項などを提示している。問題と解決の勘どころの説明に登場したドキュメントも含めてここに集約して提示している。

本ガイドは、以下のような場面でご活用いただきたい。

- 要件定義を実施する前に、どのような場面でどのような問題が発生するか、など要件定義の全体像を理解するための基礎知識として
- 要件定義を計画する際に、より確かな計画を立案するための参考として
- 問題発生時に、より実効性のある対策を検討する際の参考として

本ガイドの想定読者を図 始.6 に示す。なお、本ガイドはユーザ企業において要件定義に従事される方を主な読者として想定しているが、要件定義を実施する際にはベンダ企業による支援が必要なケースも多いと考えられることから、対象読者に含めている。

要件定義ガイドの内容（何を）		主な想定読者（誰に）				
		経営層 ビジネス 責任者	プロジェクト マネージャー	業務 部門	システム 部門	ITベンダ
背景 (第1章)	要件定義が置かれた現状を理解する					
	要件定義の現状、環境の変化	◎	◎	○	○	○
問題認識 (第2章)	要件定義において発生する問題を認識する	◎	◎	◎	◎	◎
	各ステークホルダが意識すべきこと	◎	◎	◎	◎	◎
全体像 (第3章)	要件定義の全体像を理解する					
	要件定義のプロセス、ドキュメント	○	◎	○	○	○
勘どころ 発生する 問題の解決 (第4-6章)	要件定義の各カテゴリで発生する問題と その解決の勘どころを理解し対処する					
	ビジネス要件定義	○	○	◎	○	○
	システム化要件定義		○	○	◎	○
	要件定義マネジメント	○	◎	○	○	○
文書化 (第7章)	要件定義ドキュメント作成の品質を上げる					
	ビジネス要件定義の文書化		○	◎	○	○
	システム化要件定義の文書化		○	○	◎	○

◎ 主な想定読者 ○ 想定読者

図 始.6 本ガイドの想定読者

はじめに

はじめに

第1章 背景

1.1 システム開発の現状

1.1.1 IT システムの変遷

本書「はじめに」に記載しているように、IT システムは効率化ステージから競争力強化ステージに変遷してきている。図 1.1 は、「IT システムに関わる IT 技術とユーザ企業の現在」の IT 活用の段階を切り取ったものである。本稿では、各ステージについて説明する。

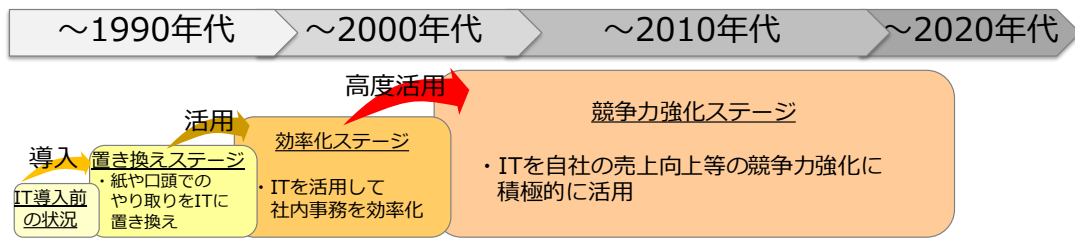


図 1.1 IT 活用の段階

【置き換えステージ】

置き換えステージは、人間の手作業の機械化、すなわち台帳・伝票・帳票を中心とした事務処理の IT 化である。この頃は、ベンダ企業が、ハードウェア、ソフトウェア、アプリケーションを一括して受注する形態が主であり、高価なハードウェアの導入に対しアプリケーションは付随的に対応していた感がある。この頃の要件定義は、関係するステークホルダーも少数の事務担当者であり、ベンダ企業が画面や帳票レイアウトの脇に仕様を書く程度での要件定義でも開発が可能であった。

これらを部門ごと、業務ごとに繰り返して企業の IT 化を進めていき、いわゆる個別最適システムの付け足し開発になり、サイロ化の出发点となった企業も少なくないであろう。

【効率化ステージ】

効率化ステージでは、一通り IT 化がされた業務システムを改善しようとして動き始める。改善の方向性としては大きく 2 つある。1 つは、現状のシステムの機能をさらに強力にしようとするもの。このとき、システム化されていない機能のシステム化だけでなく、より使いやすく、よりシステム化難易度の高い機能の搭載などにより、システムの規模を拡大しロジックの難易度をあげてきた。もう 1 つは、全体最適への動きである。個別サイロ型システムになってしまった企業は、データの多重投入やデータ管理の問題（値の不整合やマスタの同一性の問題など）が発生した。また、似たようで異なる業務プロセスなど、全社としての業務プロセスの連携や標準化、データの連携など個別システムを統合システムとして機能強化していくものである。対応策として ERP パッケージを導入した企業も少なくない。全社最

1.1 システム開発の現状

適化への根本的な対処をせずに既存システムをつなぎ合わせるだけの統合システムになってしまった企業もあり、結果、巨大化、複雑化を増長させた密結合継ぎ接ぎシステムになった。ちょうどこの頃、サイロ化されたシステムの全体最適アプローチとして EA (Enterprise Architecture) が脚光をあびた。

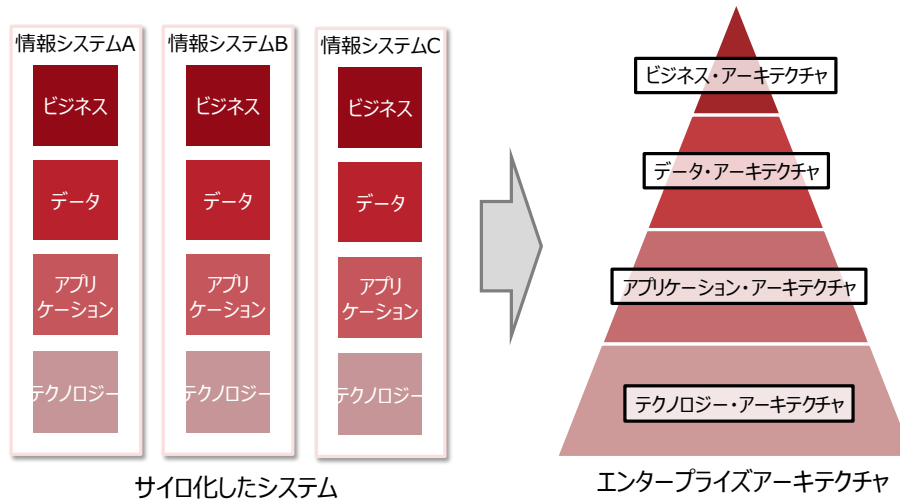


図 1.2 EA の概要

EA では、統合システムを実現していくためには、ビジネスも含めた 4 つの層に分けて進めていくべきだ、という考え方を提示している。

この頃の要件定義は、統合化など全社的視点でのシステム再構築が多かった。ステークホルダも多岐にわたり、既存システムのレベルダウンは許されず、複雑化した現状機能を踏襲せざるをえず、難易度が高かった。ベンダ企業中心の従来の要件定義では困難を極め失敗プロジェクトが多く出た。そして IPA は、要件定義はユーザ責任であることを強く訴えた。
[4]

【競争力強化ステージ】

効率化ステージで苦勞しながらも全社統合のシステムが構築された。統合システムの良し悪しをさておき、経営層は、既に効率化されたシステムに対するさらなる効率化という理由だけの多大な IT 投資を躊躇し、経営や業務に直接貢献する IT システムを求めてきた。売上向上に役立つデータの利活用やビジネスプロセスの改革に直結する IT システムが求められているステージである。

昨今、経済産業省の DX レポートで「あらゆる産業において、新たなデジタル技術を使ってこれまでにないビジネス・モデルを展開する新規参入者が登場し、ゲームチェンジが起きつつある。こうした中で、各企業は、競争力維持・強化のために、デジタルトランスフォーメーション(DX:Digital Transformation)をスピーディに進めていくことが求められている」

1.1 システム開発の現状

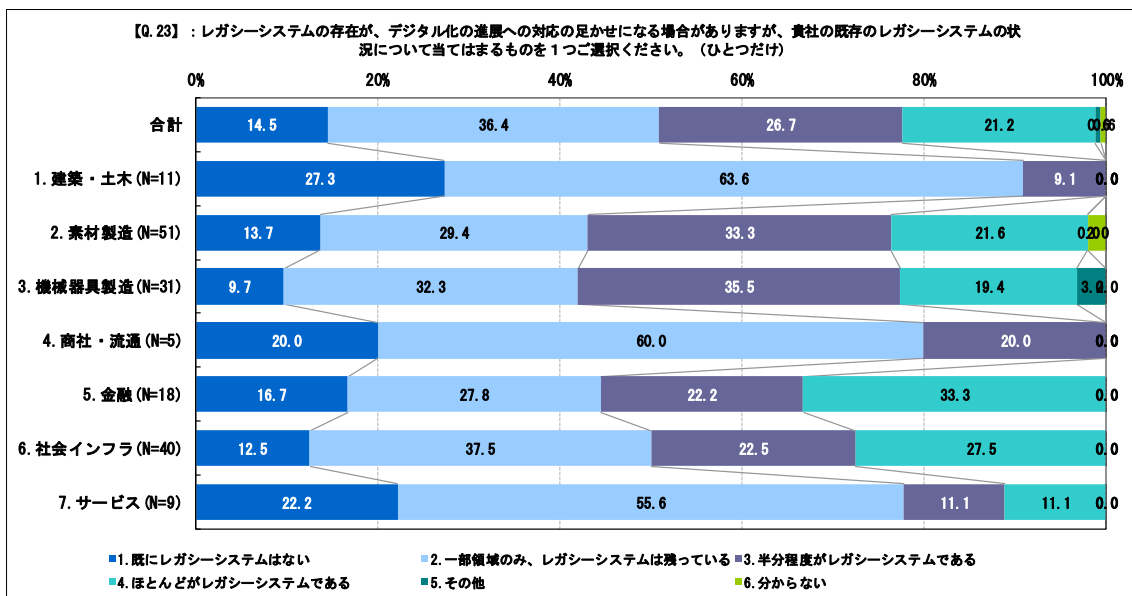
と警鐘を鳴らされた結果、DX の必要性に対する認識は非常に高まってきた。同レポートでは、各企業がこれからも競争力維持・強化していくため、いや生き残っていくためには、「デジタル時代に対応した新たなビジネス・モデルの構築と価値の創造」および「DX 実現を困難にしている既存 IT システムの再構築」を行う必要があると提言している。

1.1.2 既存システムの現状

【老朽化したシステム】

本稿では、(アンケート) 調査結果に基づく既存システムの現状を説明する。

8割強の企業が老朽化したシステムを抱えていると言われている。(図 1.3)



(出典) 一般社団法人日本情報システム・ユーザー協会「デジタル化の取り組みに関する調査 2018」 [5]

図 1.3 老朽化したシステムの現状

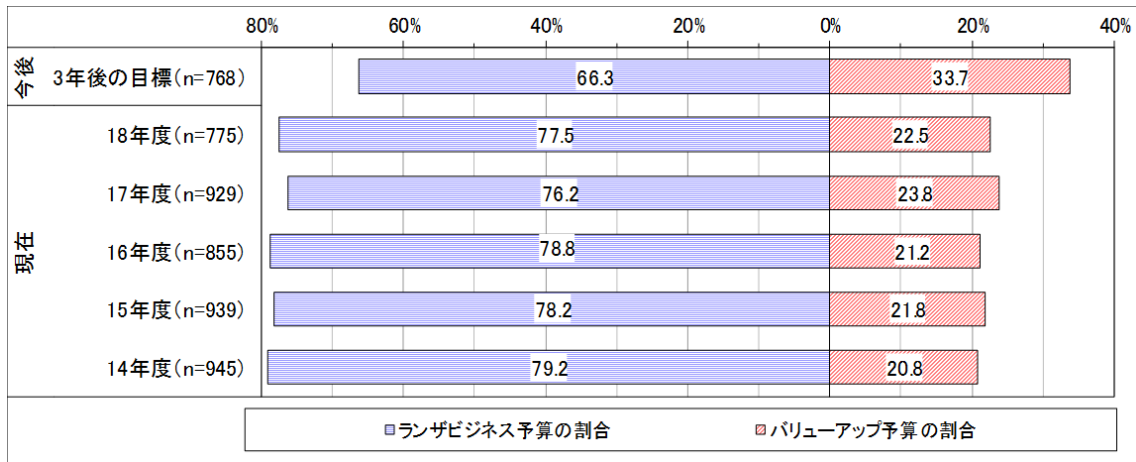
既存システムに改造・拡張などを繰り返し、巨大化、複雑化し理解不能(ブラックボックス)状態になっている企業が多いと想像できる。

環境変化のスピードが速いこの時代では、経営や事業を柔軟にスピーディに変えていかなければならないのにシステムがスピーディに変更できなく企業の足かせになっている企業もある。そうなっている経営者は「既にシステムは存在している。なぜ多大な投資をして再構築しなければならないのか」「売上向上、顧客獲得など、もっと経営や業務に直接貢献する IT システムにならないのか」との疑念を感じている。既存システムの問題を経営問題として捉えられていないことが伺える。

【既存システムの運用・保守に資金も人材も割かれている】

日本情報システム・ユーザー協会(以降、JUAS と表記)の「企業 IT 動向調査報告書 2019」[6]によると、IT 関連費用のうち約 8 割が既存システムの維持管理(ラン・ザ・ビジネス)に割かれており、戦略的な IT 投資(バリューアップ)に資金・人材を振り向けられていない状況が伺える。(図 1.4)

1.1 システム開発の現状



出展：一般社団法人日本情報システム・ユーザー協会「企業 IT 動向調査報告書 2019」 [6]

図 1.4 年度別 IT 予算配分

この報告書では、ラン・ザ・ビジネス予算を「現行ビジネスの維持・運営」、バリュー・アップ予算を「ビジネスの新しい施策展開」と定義されている。調査では、それぞれのキャッシュベースでの IT 予算の配分は 80 対 20 で、その比率は 14 年度から 18 年度までの 5 年間ほとんど変化していない、と報告されている。

1.1 システム開発の現状

1.1.3 システム開発の現状

【納期遅延・品質満足度が低い】

システム開発の実態を見てみると、JUASの「企業IT動向調査報告書2019」[6]によれば、工期遵守状況は図1.5、経営層の品質への満足度は図1.6のようになっている。

500人月以上の大規模システム開発プロジェクトを見てみると、2018年度は「予定より遅延」したプロジェクトが43.9%を越し、23.8%のプロジェクトにおいて経営者がシステムの品質に不満を持っている。規模が小さくなるとこの状況はやや改善されている。

また年々工期、品質は改善されてはいるがまだまだ満足できる値ではない。

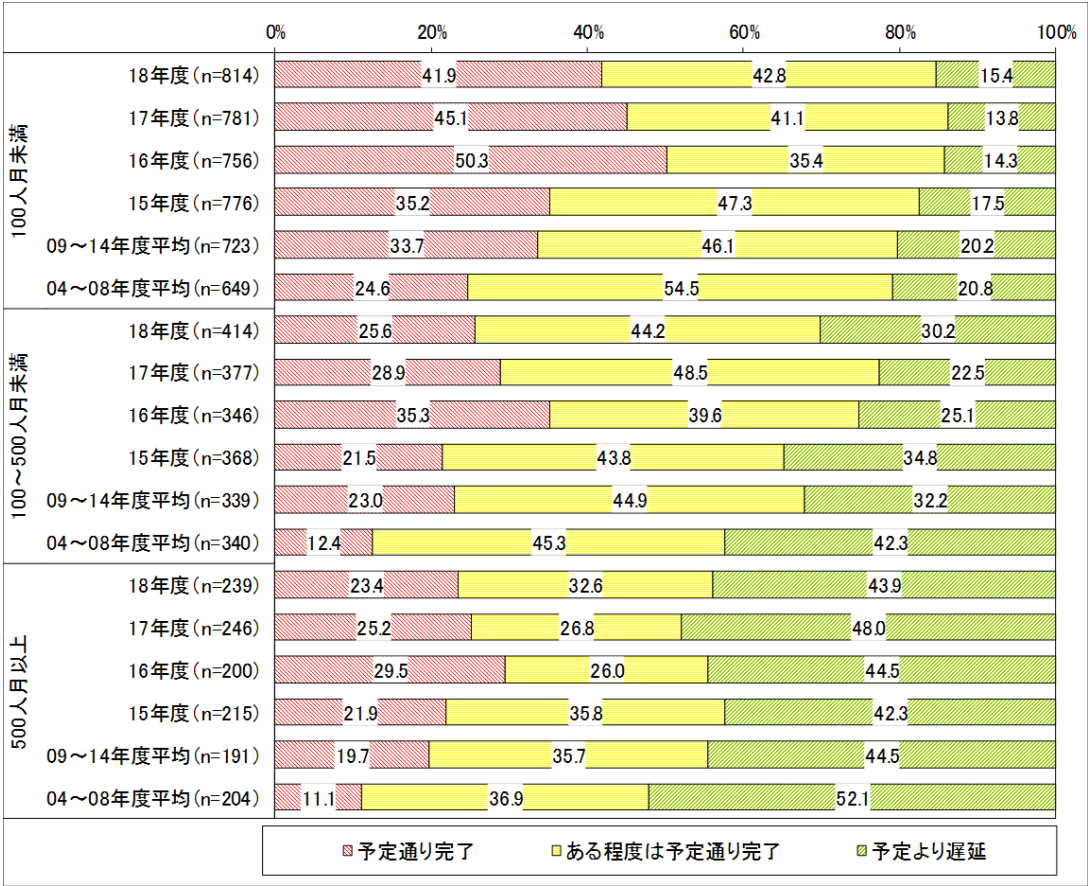


図 1.5 年度別・システム規模別 システム開発の工期遵守状況

(出典) 日本情報システム・ユーザー協会「企業IT動向調査2019」[6]

1.1 システム開発の現状

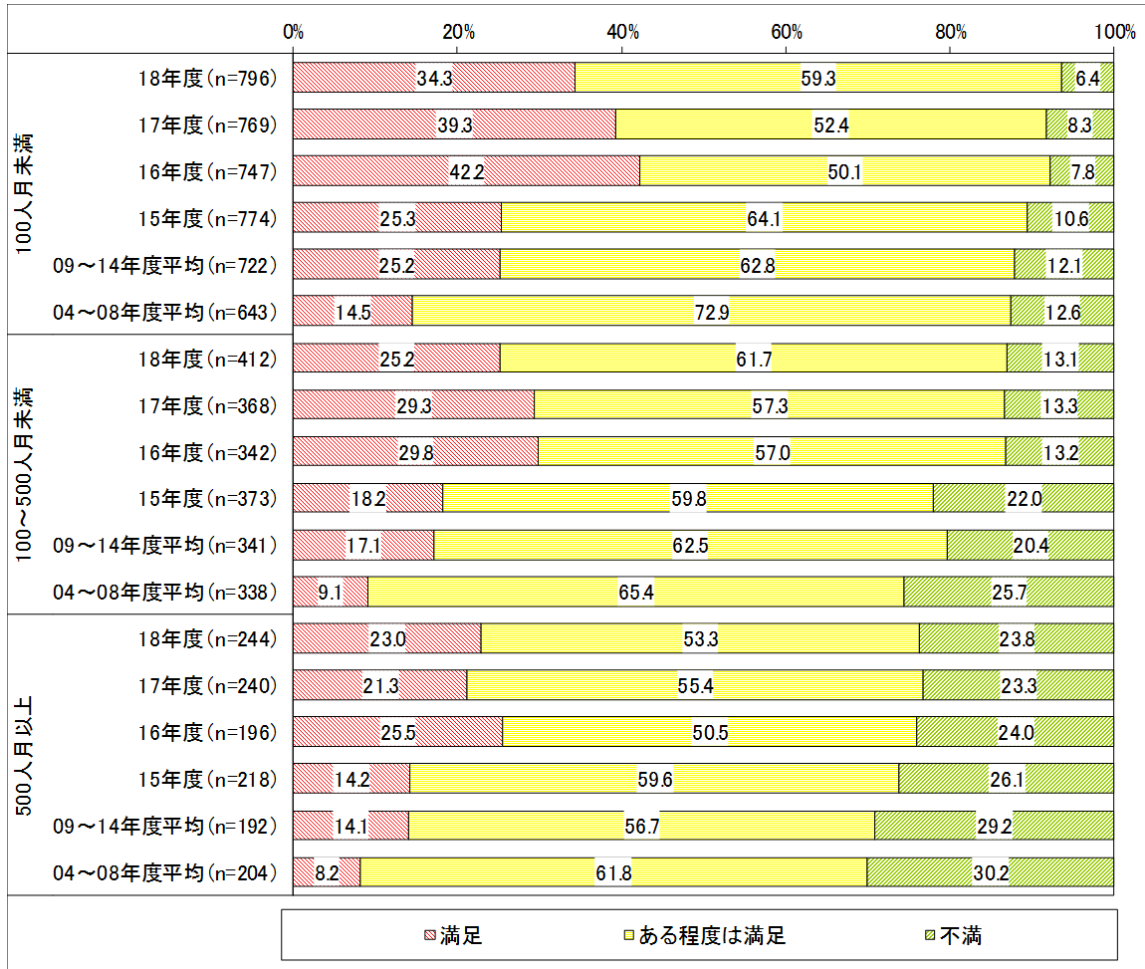


図 1.6 年度別・システム規模別 システム開発の品質満足度

(出典) 日本情報システム・ユーザー協会「企業 IT 動向調査 2019」 [6]を基に加筆

1.1 システム開発の現状

【50%以上が要件定義問題】

工期遅延理由の分析データ（図 1.7）から原因を探ってみる。

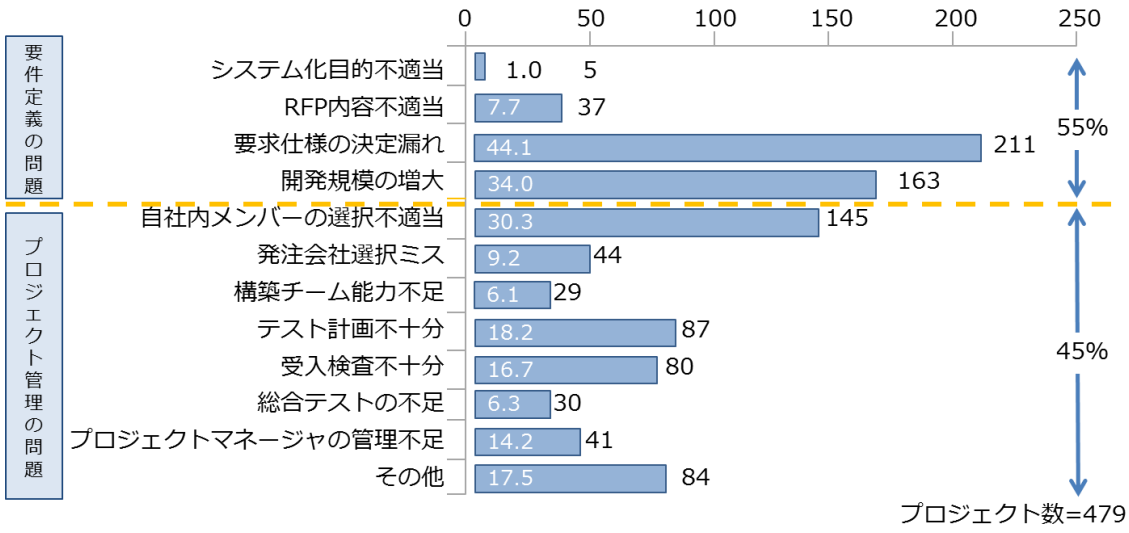


図 1.7 工期遅延理由分析

（出典）日本情報システム・ユーザー協会「ソフトウェアメトリックス調査 2016」 [7]をもとに加筆

図 1.7 は、「JUAS ソフトウェアメトリックス調査 2016」 [7]において 479 プロジェクトを対象に工期遅延の理由を担当者にアンケート調査した結果である。この図から、工期遅延の理由の 50%以上が要件定義の問題にあることが分かる。また、同じ調査の中では、予算オーバーや品質不良においても要件定義の問題が原因の多くを占めることが指摘されている。

【ユーザ企業とベンダ企業の関係】

システム開発は経営要求を含むシステム化構想、要件定義と議論が進められてくるが、ユーザ企業とベンダ企業の責任分担の境目は要件定義である。

IPA の「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～」 [1]では「要件定義は発注者の責任である」と述べている。また、経済産業省の「情報システム・モデル取引・契約書 (2007・4)」 [8]では、要件定義は準委任契約にすることと述べられている。しかし、我が国では要件定義から請負契約を締結するケースも少なくなく、ユーザ企業からベンダ企業への丸投げ状態になり、問題を起こしているプロジェクトが見られる。ベンダがユーザ企業の業務やシステムを理解している、理解することを前提にプロジェクトが進められる。ユーザ企業とベンダ企業の責任や役割分担が不明確のまま進め、紛争・訴訟へと発展するケースも増えている。

【再構築の難しさ】

システムが既に存在していることを前提に考えると、その既存システムの存在が課題を

1.1 システム開発の現状

生んでいる。刷新するにも、改造するにしても既存システム理解が重要である一方、その既存システムが巨大・複雑化しているために、すべての仕様を明確にすることが困難になっているからである。新しい要求は新しい要求を持っている人が存在するので検討し提示すれば良いが、既存システムに変更がない部分は「現行踏襲」という安易な要求で片付けられる。しかし、既存システムのドキュメントが陳腐化していたり、業務知識を持っている人が業務部門にすら居なくなっている。少しでも業務を IT 化しようと拡張と過剰品質を繰り返してきた結果、業務がシステムの中に埋没し業務部門でも分からなくなっている。「システム部門の方が常に保守し続けているからわかっているでしょ」と謂れない責任転嫁がされることもある。さらに、当初の IT システム開発に携わった有識者が定年退職を迎えブラックボックス化に拍車が掛かっている。

【全体最適の難しさ】

個別サイロ型システムを全体統合システムに変更するには、システムの刷新が必要であり、多くの費用と期間が必要となる。ビジネス現場は保守派が多い傾向にある。現場の作業改善がその現場での IT の最大の恩恵である。ビジネスプロセスが大幅に変わることには抵抗し、なれ親しんだ業務を続けたいのである。このような状況では、全体最適化より各事業部の最適化が最優先され、合意形成が困難を極める。経営層の英断とリーダーシップがなければうまくいかない。このような状況では、改修して利用し続けた方が安全であるという判断もあるだろう。

【システム部門の苦悩】

システム部門は、経営層から経営や業務に貢献する IT システム、ビジネス環境の変化に追従できる IT システムが求められていることを理解している。しかし硬直化した既存システムが足かせとなり実現困難になっている企業が少なくない。システム部門は硬直したシステムを柔軟なシステムに刷新したくても、費用、期間も膨大で予算が通らず困っている。

経営層に認められるような新しい試みの導入を理由に予算を確保し、既存システムの変革も同時にしかも少しずつ進めていくなど工夫しているようだが、これもシステム部門にとっては極めて困難な茨の道である。既存システムの現状を経営課題として経営層に認識させる活動が必要である。

1.1 システム開発の現状

コラム 1 日本の IT が抱える課題

日本における IT 投資には、以下に述べるような特徴があり、それぞれに課題がある。

【SE の外部依存度】

SE がユーザ企業、ベンダ企業のどちらに所属しているかの比率を見ると、日本はユーザ企業 25 対ベンダ企業 75 であるのに対して、米国はユーザ企業 75 対ベンダ企業 25 と逆の比率になっている（「グローバル化を支える IT 人材確保・育成施策に関する調査」概要報告書 [9]からの出典）。日本では一般に入札等によってシステム開発ベンダを選択できることから、ユーザ企業にとっては安く高品質のシステムを手入できるプラス面がある。しかし、企業内での IT 新技術の採用や技術面でのイノベーションにおいては、米国のようにユーザ企業内に数多くの SE を抱えている方が有利である。

設備投資全体に占める IT 投資の割合の推移に関する欧米との比較を図 1.8 に示すが、日本は欧米に比べて IT 投資の割合が低く、その差は拡大し続けている。ただし、日本で導入できるハードウェアやシステムの調達価格が安価かつ高品質であることが、売上高に対して IT 投資比が少ないことに結びついているとの指摘もある。

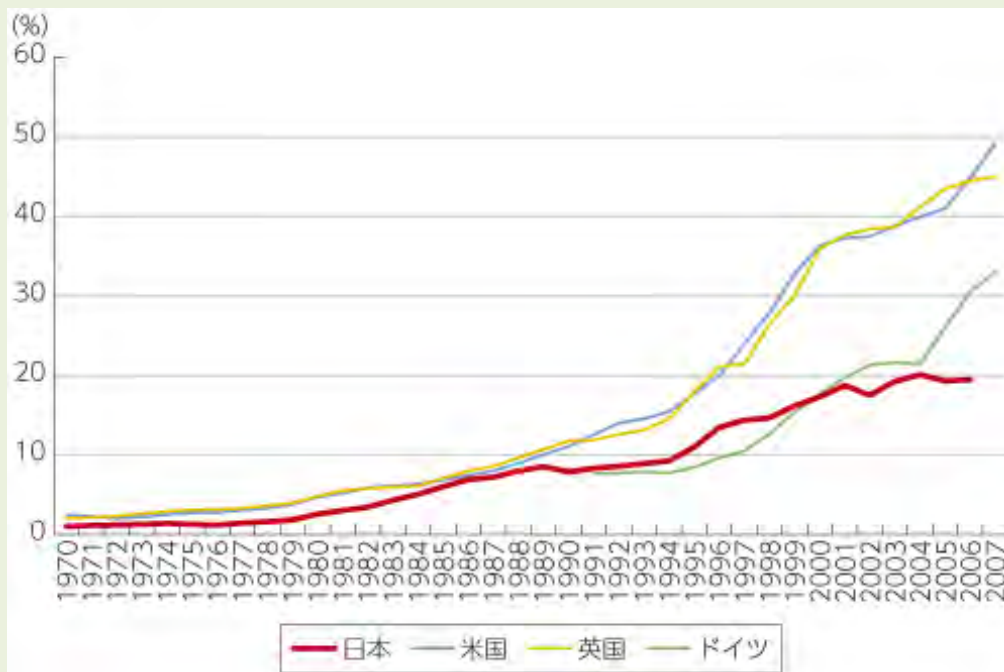


図 1.8 設備投資全体に占める IT 投資の割合

(出典) EUKLEMS より作成、総務省「ICT による経済成長加速に向けた課題と解決方法に関する調査研究」(2014 年) [10]

1.1 システム開発の現状

【IT 予算の割合】

日本企業の営業利益率は、米国企業の営業利益率と比較するとおおよそ半分である。平均 3%程度の営業利益率の日本企業にとっては、対売上高比 0.5~1%の IT 予算は費用削減の対象として注目されがちである。これに対して、ほぼ 2 倍の利益率を達成している米国企業の経営者にとって IT 予算は、費用削減の対象としてよりも、前向きな IT 活用のために必要な投資としての意味合いが大きい。

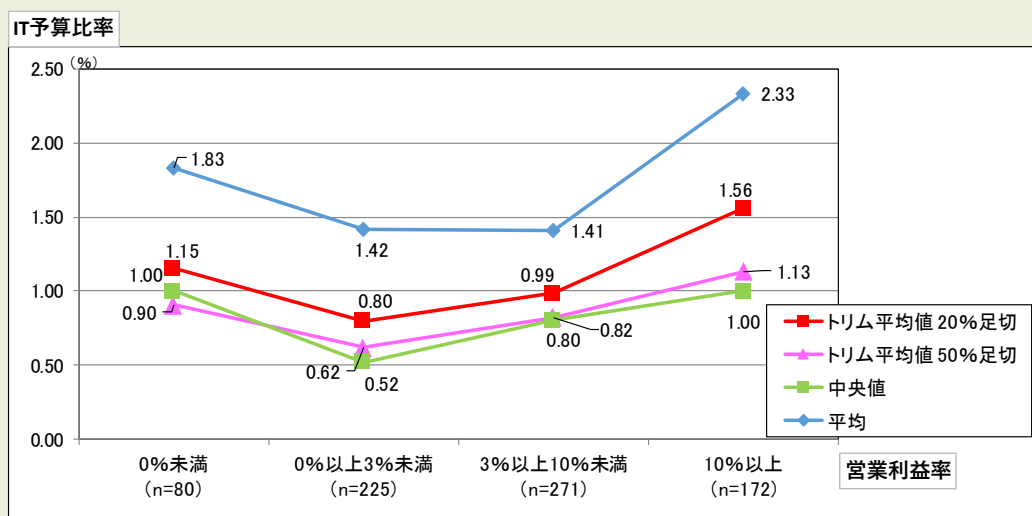


図 1.9 営業利益率別 売上高に占める IT 予算比率

(出典) 日本情報システム・ユーザー協会「企業 IT 動向調査報告書 2019」 [6]

図 1.9 に示すように、営業利益率がプラスの企業の場合、営業利益率と IT 予算比率には関係性がある。営業利益率が高い会社では、IT 予算比率も高くなっている。営業利益率が高い企業は、IT 投資にも積極的であるのに対して、営業利益率が低い企業は IT 投資を削減し、営業利益率を確保する消極的経営戦略を取る傾向にある。

営業利益率が IT 投資予算比率に影響を与えていることは間違いないが、逆に、現在では IT 投資の比率が企業の利益率を押し上げる原動力の一翼を占める存在になっているという認識を持つ必要もあるのではないかと。

1.1 システム開発の現状

1.1.4 DX に見える今後求められるシステム開発

DX とは、AI など最先端の IT 技術（デジタル技術）を使って業務を効率化すれば良いというものではなく、新たな価値を創出し、新しいビジネス・モデルに変えていかないと、これからの時代は競争優位に立てない企業が出る、という警鐘である。これは、今正に第四次産業革命が始まりつつあり、ビジネスモデルレベルでの変化をしないと生き残れなくなる世の中が到来したということの意味している。第一次産業革命は蒸気機関によるもの、第二次産業革命は電力によるもの、第三次産業革命はコンピュータによるものあったのに対し、第四次産業革命は、インターネットの発展によるものである。スマートフォンや IoT(Internet of Things)/IoE(Internet of Everything)など、あらゆるものがインターネットを通じてつながる時代になった（図 1.10）。

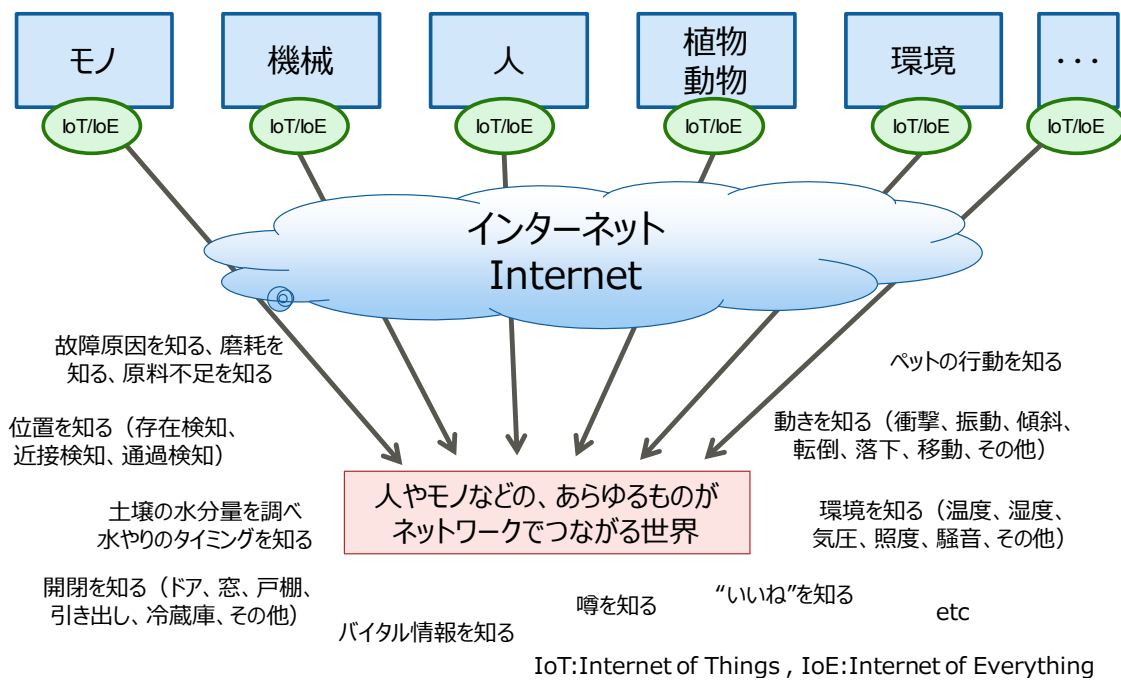


図 1.10 あらゆるものがインターネットでつながる世界

これらが全盛になると、つながる人すなわちステークホルダとして考えなければならない対象が広がり、扱うべきデータも、企業内データから外部のデータ、画像、音声、動画などのデジタルデータへと広がる（図 1.11）。そして、そこで行き来するデータの量も膨大になり、いわゆるビッグデータになる。

1.1 システム開発の現状

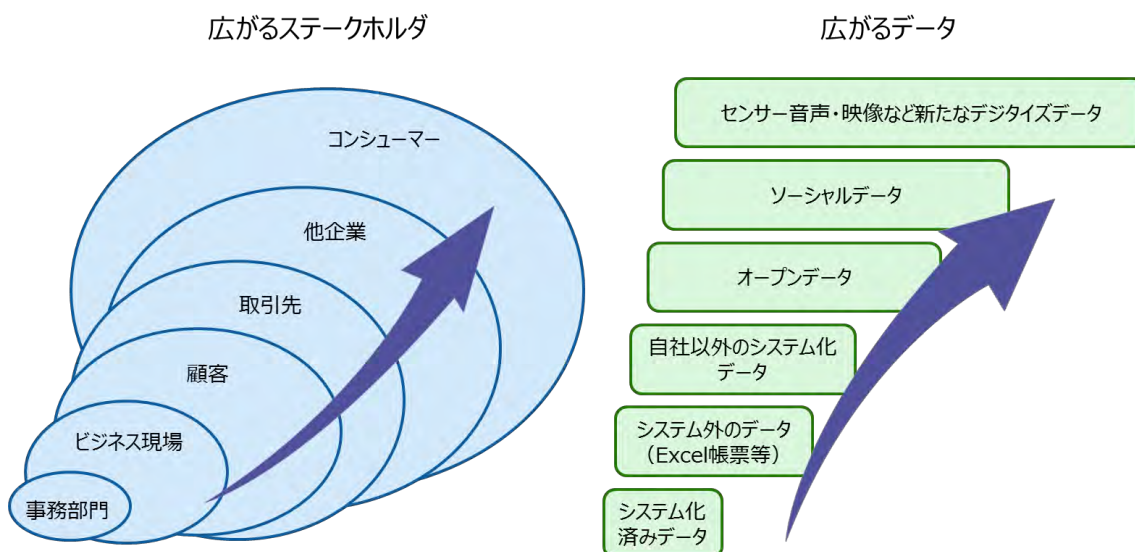


図 1.11 広がるビジネス環境

このように、ステークホルダやデータが広がる「デジタル時代」に競争優位を築くためには、ビジネスモデルチェンジを行わなければならないという意味が DX にはある。

最先端のデジタル技術を使用することが DX ではない、と述べたが、DX を実装するデジタル技術として、AI や AR/VR/MR、ブロックチェーンなどのコア技術となる要素技術から、マイクロサービスのアーキテクチャやそれを格納するコンテナ、動作させるクラウドなどの実装技術が進化し続けており、これらの技術なくしては DX をスピーディに進めていくことができないのも事実である。

1.1.1 でも述べたように、DX では、大きく 2 つのことを行う必要がある。「デジタル時代に対応した新たなビジネス・モデルの構築と価値の創造」と「DX 実現を困難にしている既存 IT システムの再構築」である。昨今のシステム開発は、DX といったイノベーション領域の話と、老朽化・複雑化・ブラックボックス化した既存システムをどのように再構築していくかが重要になっている。いわゆる攻めの IT 投資、守りの IT 投資である。

(1) デジタル時代に対応した新たなビジネス・モデルの構築と価値の創造

あらゆるものがインターネットにつながる時代の例としてスマートフォンの普及が挙げられる。いわゆる個人が利用できるインターネット端末が爆発的に増えていることである。これは、ビジネス・モデルを変えている。今まで、B2B2C ビジネス・モデル企業が B2C ビジネス・モデル企業化している。製造/生産現場→卸→小売店→消費者、サービス提供企業→代理店→消費者といった B2B2C ビジネス・モデルから直接消費者から注文を受け届ける B2C ビジネス・モデルへと変わってきている。

企業が直接消費者につながるにより、本当に消費者が欲しいものは一

1.1 システム開発の現状

体何なのかを突き詰めることからビジネス・モデルを換える企業も出てきている。例えば、ある自動車会社は、消費者が本当に望んでいることは車を買うことではなく、気軽に移動したいということであると考え、製造販売業から移動サービスへとビジネス・モデルを大きく変革しようとしている。

以下に、DX 実現に関係する代表的なアプローチ方法を挙げる。

【人間中心アプローチ デザイン思考/UX デザイン】

ビジネス・モデル変更の背景には、商品を出せば売れる時代ではなくなったことも挙げられる。一昔前は、商品の機能の差別化や品質の差別化で消費者を振り向かせていたが、直ぐにコモディティ化する。贅沢な機能は必要なく基本機能で十分だという。そして、買い換える、いや「モノ」が欲しいという気持ち自体が消費者の中で減っている。「モノ」中心の考え方では消費者は振り向かないのである。人の心をつかまないと、振り向いてもらえない時代である。消費者は、最低限の機能（当たり前）があった上で、色や見栄え、使いやすさ、楽しさ、驚き、感動、心地よさなどで振り向く場合がある。いわゆる人間中心の発想が必要になってきている。ユーザ起点（人間中心）でアイデア発想する「デザイン思考」や、ユーザに「楽しさ」や「驚き」、「心地よさ」を与えようとする「UX (User Experience: ユーザエクスペリエンス) デザイン」などが DX に役立つ技術である。DX レポートでも、DX を「新しい製品やサービス、新しいビジネス・モデルを通じて、ネットとリアルの両面での顧客エクスペリエンスの変革を図ることで価値を創出し、競争上の優位を確立すること」という定義を紹介している。

【IoT/IoE とデータ利活用】

データ利活用も重要になってきている。今までは「情報系システム」と呼ばれ、既存システムのデータを集計し営業や販売、マーケティングといった部門が利用していた。昨今は、IoT(Internet of Things)/IoE(Internet of Everything) すなわち身の回りのあらゆるものがインターネットにつながり、さまざまなものの状態や位置など今まで見えなかったデータ、埋もれていたデータを知ることができるようになってきている。企業内にある情報や SNS などから伺える情報をも含めて、活用できるデータが爆発的に増加している。データの種類も今までの構造化データだけでなく、各種のログや SNS などのテキストデータやセンサなどのストリームデータ、音声や動画、画像のマルチメディアデータなど、各種の非構造化データも利用可能になってきている。

これらを活用することにより、例えば、消費者の行動を解析し最適なサービス検討や消費者を誘導していくために利用、位置情報や画像・音声情報と組み合わせること新しいビジネス・モデルを立案、AI などの技術を活用し、予知・予測の実現など、新たな価値創造、新たなビジネス機会創出が期待されている。

ビジネス・モデルを変革していくには、どのようなデータをどのように活用していくか、

1.1 システム開発の現状

データを活用してどのようにビジネス価値を生み出すか、という発想も重要だ。データの利活用に大きな価値がありそうだと、やみくもにデータを集め蓄積しただけでは“沼”化する（一般的に定型データ、非定型データ、画像データなどのビッグデータ蓄積概念をデータレイクというが、使えないデータレイクを卑下して“沼”と呼んでいる）。データレイクだけ作って価値を発見しようとしてもうまくいかない。どのようなビジネス・モデルにして、どのような価値を作るのかというところがベースにないとデータ利活用もうまくいかないのが現実のようだ。

【リーンスタートアップ、アジャイル開発】

デジタル時代の要件定義で重要なことは、ビジネスを取り巻く環境の変化が激しいので、要件が決められず、決められた要件も頻繁に変更されるので、要件が確定するのを待っていると市場競争に負けてしまうということが起こる。すなわち長期間を要するシステム開発がすぐわなくなってきた。これには、要件定義に対する工夫が必要になる。まずは実用最小限の製品・システム（MVP, Minimum Viable Product）をできるだけ迅速に構築して利用者に提供する。利用者の反応を検証して得られた結果をもとに、当初のアイデアの改良や軌道修正、さらに破棄を図る。いわゆるリーンスタートアップである。そのためには、システム開発も要求の変更に柔軟でハイスピードの開発が必要になる。いわゆるアジャイル開発である。

以上をまとめると、「デジタル時代に対応した新たなビジネス・モデルの構築と価値の創造」では

- ・お客様が本当に欲しい価値は何なのか
- ・お客様のエクスペリエンスとは何なのか
- ・ビジネスにどういう価値を作るのか
- ・ビジネス・モデルを変えるために、データをどのように活用すれば良いのか
- ・デジタル技術を用いて、どのように価値創出に貢献できるのか

などを考えることがベースにある。これはDX以前の従来の要件定義そのものであり、要件定義は普遍であることを示している。

「新たなビジネス・モデルの構築と価値の創造」においても要件定義が重要であることを意味している。

(2) DX 実現を困難にしている既存 IT システムの再構築

DX レポートは「DX を実行していくためには、データを収集・蓄積・処理する IT システムが、環境変化、経営・事業の変化に対し、柔軟に、かつスピーディに対応できることが必要である。しかし、既存システムが複雑化・老朽化・ブラックボックス化する中ではDXの足かせになっている」と述べている。さらに、「複雑化・老朽化・ブラックボックス化した既

1.1 システム開発の現状

存システムが残存した場合、2025年までに予想されるIT人材の引退やサポート終了等によるリスクの高まり等に伴う経済損失は、2025年以降、最大12兆円/年（現在の約3倍）にのぼる可能性がある」とも述べている。

DX実現に向けて、複雑化・老朽化・ブラックボックス化した既存システムの刷新に多くの企業が踏み出せないでいる。この理由としては、コストや時間といったリスクを評価する以前に、自社の情報資産を正確に把握できていないため、どこに課題があり、どのように構築しえいけば良いか判断がつかないことが挙げられる。

これらを踏まえ、既存システムの刷新に必要なプロセスの一部を以下に示す。

- ・ 既存システムの調査・分析
- ・ 情報資産の仕分（機能分割・刷新、機能追加、機能縮小・廃棄、現状維持（塩漬け）など）
- ・ 再構築目的やビジネスリスクの明確化
- ・ 再構築手法の選択、再構築リスクの明確化
- ・ 現行業務知識不足への対応
- ・ 品質保証（業務継続性の担保、テスト）の検討
- ・ 再構築計画と見積り

など

既存ITシステムの再構築においても上流工程が重要であることを意味している。

コラム 2 DX 関連技術

ここでは、DX に関する 4 つの手法を簡単に紹介する。

最初の 2 つはビジネスモデルチェンジに役立つアイデア発想方法であり、後者の 2 つは実装方法/マネジメント方法である。

【デザイン思考】

イノベーションは、マーケットの機会を探ることや技術シーズから起こるケースもあるが、商品やサービスを利用する人のニーズから生まれることもある。デザイン思考とは、人のニーズを知る事を起点にアイデアの創出、ビジネス化の検討を行う、もっともっと人の立場、気持ちに立ったアイデア発想アプローチで HDC (Human-Centered Design) 人間中心設計とも言われている。企業視点でモノを作って終わりという時代でなくなった。「ユーザが本当に悩んでいることは何か」「どのように解決するか」「なぜ必要なのか」「ユーザが価値を感じるポイントはどこか」常にユーザ視点で考える。また、答えを知っているのもユーザである。そのためには、試作（プロトタイピング）して早く利用してもらい、検証していく必要がある。

以前から多く使用されている仮説検証・分析型アプローチ、すなわち、問題を可視化し、原因を分析し、対策立案していくのと比較すると、デザイン思考のアプローチは、ユーザに共感し、問題の本質を探っていく、多様なアイデアの発散を行い、プロトタイピング、検証で収束を計っていくものである。スタンフォード大学 d. school の基本プロセス [11] を図 1.12 に示す。

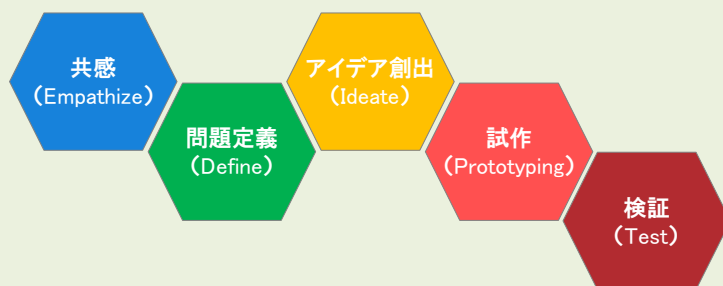


図 1.12 スタンフォード大学 d. school のデザイン思考基本プロセス

【UX デザイン】

UX とは、分かりやすくいうと、ユーザに「楽しさ」や「驚き」、「心地よさ」を与え、「ぜひまた使ってみよう」と思わせるような、アイデア発想アプローチである。

1.1 システム開発の現状

UXとはUser Experienceの略であり、日本語では、ユーザ経験（体験）と訳される。ユーザ経験とは、ユーザが製品やサービスを利用することによって受ける気持ちのことを指す。UXも人間中心設計である。ただ良い商品やサービスを作るということだけでなく、「どんなユーザ経験を提供できるか」を考えることが、ビジネスの成功を左右する重要なファクタになってきている。

UXを分析・設計するための代表的なツールとしてカスタマージャーニーマップがある（図1.13）。製品やサービス、システムの利用中だけでなく、利用前、利用後までを含めた「一連」の人の行動や感情、すなわち「ユーザ経験」を分析する。

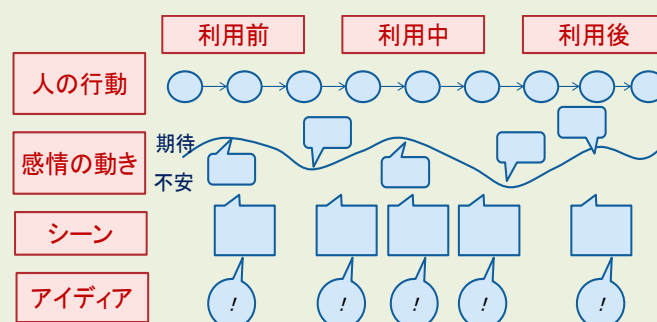


図 1.13 カスタマージャーニーマップのイメージ

【リーンスタートアップ】

リーンスタートアップとは、新しいビジネス・モデルの開発を無駄なく進めるアプローチ、マネジメント方法である。ビジネスを取り巻く環境の変化が激しいので、要件が決められない、要件がまとまるのを待っていると市場競争に負けるということがある。こういう要件のとき、無駄なく早く進める方法である。決してアイデア発想アプローチではない。

まずはアイデア/仮説を立てる。これはアイデア発想アプローチを利用しても良い。その仮説を評価できる最低限実用に足る製品・システム（Minimum Viable Product）で、できるだけ迅速に構築して利用者に提供する。利用者の反応を検証して得られた結果をもとに、当初のアイデアの改良・軌道修正ややめると破棄を図る。これを繰り返す、というアプローチである。

【アジャイル開発】

製品やサービスがリーンスタートアップの進め方を行おうとしたら、それを支えるシステムの開発にもスピードが求められる。アジャイル開発も、俊敏（アジャイル）に開発することで、柔軟に変化に対応し、スピード重視のアプローチ、マネジメント

1.1 システム開発の現状

方法である。アジャイルソフトウェア開発宣言にもあるように、「プロセスやツールよりも個人と対話を、包括的なドキュメントよりも動くソフトウェアを、契約交渉よりも顧客との協調を、計画に従うことよりも変化への対応を、価値とする。」計画やドキュメントに価値は認めるが、それよりも動くプログラムに多くの価値を認め、動くプログラムを中心に、対話と協調で無駄なくスピーディに進めようとするものである。アジャイル開発の代表的な手法としてスクラムとエクストリーム・プログラミング (Extreme Programming) (XP) がある。

スクラムは、チームワークを重要視するのが特徴である。具体的には、開発の計画をメンバで緻密に話し合い、プロジェクトの進行に問題がないかを確認しあいながら進めることで、リリースできないという事態を回避する。

エクストリーム・プログラミングは、途中で仕様の変更にも柔軟に対応することに重きを置いている手法で、初期段階での計画にこだわり過ぎないのが特徴である。

アジャイル開発は、ソフトウェア開発者がユーザとともに使う開発手法である。詳細は他の文献などを参照願いたい。

アジャイル開発のメリットは柔軟に変化に対応できることから、もし不具合が発生してしまった際でも、修正の工数が多くないことである。そして圧倒的なスピード・時間重視という点である。

1.2 要件定義をめぐる環境の変化

1.2.1 要件定義に求められる3つのこと

この度社会経済の変化に対応するために民法が改正された。一部の規定を除き、2020年4月1日から施行される。システム開発契約への影響は、請負契約・準委任契約のそれぞれについて考える必要がある。ポイントは3つある。

ポイント①：【請負】 瑕疵担保責任の改正

ポイント②：【請負】 プロジェクトがとん挫した場合でも報酬請求が可能に

ポイント③：【準委任】 成果完成型の準委任契約が認められた

詳細は省略するが、今回の改正で気を付けるべきは、ポイント①内でうたわれている、成果物の「瑕疵（かし）担保責任」という考え方がなくなり、代わりに「契約の内容に適合しない仕事の目的物」という言葉が用いられていることである。今まで、ソフトウェアの瑕疵といえば、通常はプログラムのバグなど「明確な不具合」を指すことが多い。今後、「経費精算業務の効率化」という契約の「目的」達成を期待できるシステムを作ったかが問題になる。効率化に資することが期待できると「客観的」に思えるものを一定の「品質」を持って納めることが必要になる。すなわち、システム開発を受託する側は、提示されたシステム機能をバグなく作り納品するだけではなく、ビジネスや業務が狙っている目的の達成も求められるようになったことを意味している。（ただし、実際に契約書に記載した内容が優先される）結局、目的の達成に関しても、契約上の両当事者の責任が明確になるようにきちんとした契約を結び、お互い合意を取ってから進め、どういう場合に不適合なのかをジャッジできるようにしておくことが大切と訴えているとともに、要件定義で要求のビジネス貢献度の明確化やその実現方法など契約を結ぶにあたって合意形成ができるような情報を明確にする必要があると訴えているように思う。

これらも踏まえて要件定義では、以下の3つのことを意識して行う必要がある。

【1：経営や業務に貢献する要求を見極める】

基本的には、構想立案やシステム化計画に主要なものが記載されているはずである。しかし、いざ要件定義を終えてみると当初の狙いが達成できていない要件定義になっていることがある。上記民法改正にもあるように、契約の「目的」達成を期待できるシステムを作ったかどうか問題になる。また、経営層や業務部門のすべての要求を実現できるわけではない。経営層や業務部門との合意形成や積極的な経営層、業務部門の参画などが要件定義には求められる。

1.2 要件定義をめぐる環境の変化

【2：要求を実現する新しい業務を作り上げる】

ビジネスや業務のプロセスが変わらないのに、箱（システム）だけ最新のものに置き換えるなんてナンセンスと雑誌で話題になった事がある。リHOSTやリライトなど意図的に業務プロセスを変えたくない開発もあるが、多くの場合、経営者のIT投資の意図は、ビジネスプロセスのチェンジである。新しいビジネスプロセスを明確に定義し業務運用までつなげるのも要件定義の仕事である。

【3：要求仕様を「抜け」「漏れ」「あいまい」なくシステム開発につなげる】

図 1.7 に示したように、システム開発の失敗の原因の多くが要件定義にある。その中でも、要求仕様の決定や漏れが一番多い。要求仕様抜け・漏れがあったり、あいまいであったりしている。システム開発工程で発覚し手戻りを起こしている。手戻り負荷は、発覚が遅ければ遅いほど指数的に大きくなると言われている。

1.2.2 パラダイムシフト

昨今は、クラウドやマイクロサービスなどの普及により、システム構築はサービスを組み合わせて作るものになってきている。従来のパワーによるシステム開発は減っていくはずである。そうなると、もの作りを主体としていたシステム部門やベンダ企業は、新たな分野へパラダイムシフトしなければならない。

システム部門は、今まで以上にビジネス視点を持ち、情報化やシステム活用を考え推進していく必要があり、変わっていかなければならない。開発技術だけではなく、ビジネス指向で経営層や業務部門と対話し推進する組織へと変化する必要がある。ベンダ企業は、お客様のビジネス課題をお客様と一緒に考え、ソリューション、ビジネス価値を提供する企業へと変化する必要がある。

要件定義は、ビジネスと IT システム構築の橋渡しと言われている。橋渡しを行う人を、BABOK(A Guide to the Business Analysis Body of Knowledge)ではビジネスアナリストと呼んでいる。従来の SE 人材をこのビジネスアナリストとして育成、すなわち要求工学（リクアイメントエンジニアリング）を習得し、業務部門やお客様の中に入り込み上流工程を支援していく。

また、IT 技術ありきでのビジネス改革やビジネスモデルチェンジも必要になってきている。IT 知識がビジネス改善や改革のキーになることもある。IT 技術に長けた人が上流工程から入り込むことも必要になってきている。

このような背景からもの作り SE（システムエンジニア）から上流工程人材へパラダイムシフトしようと考えているユーザ企業のシステム部門やベンダ企業が出てきている。

1.2 要件定義をめぐる環境の変化

1.2.3 重要度が上がる非機能要求

昨今、非機能要求の重要さが増してきており、経営や業務の重要な関心事になっている。大きな事件や災害時などの時の事業継続やセキュリティなどが身近なリスクとして重要視されている。企業内の問題ではなく、企業の社会的責任として捉えなければならず経営そのものである。

しかし、利用部門の関心は機能に偏る傾向にある。要件定義のパワーの95%が機能検討にさかれ、非機能の検討は画面レスポンスぐらいという例もあり、非機能の検討が設計工程から検討され、予算やインフラが決まった中で後手に回り苦勞するという悪循環になることもある。

さらに、性能やセキュリティ、信頼性などに加え、使用性（ユーザビリティなど）や満足性（快適性、快楽性など）も求められるようになってきている。これらも非機能要求に位置付けられ重要になってきている。機能による差別化が利用者の関心を獲れなくなってきている。一とおりシステム化された状態では、機能は期待とおりに動いて当たり前になっており、ユーザビリティや利用時の品質が重要視され始めている。例えば、表を画面で表示するときに行の色を変える。これは読みやすさを向上しミスをなくす、疲れにくいなどの効果を期待したものである。また、昨今は、ネット通販のように商品購入時に「他の人はこんな商品を購入しています」、「あなたにこの商品はいかかですか」とリコメンドしてくる。基本機能は買いたい商品を購入できることであるが、予定していた商品よりより良い商品を購入できたとしたら、これは想定以上の価値を提供できた事になり、満足度が高いという評価を得るだろう。

JIS X 25010:2013 (ISO/IEC 25010:2011) [12]でも製品品質に加え利用時の品質も重要な品質であると追加している（図 1.14）。

非機能要求は、重要なビジネス要求、利用者要求として捉え、経営層や企画部門、利用部門などが上流工程から検討を進めないといけなくなっている。

上流工程で、非機能としての要求を明確にすることを目的に IPA は「非機能要求グレード」 [13]を公開している。概要は7.12に記載しているので、そちらを参照していただきたい。非機能要求グレードはインフラを決めるための要求を中心として上流工程でまとめるべきことを列挙してくれている。非機能要求すべてをカバーしているわけではないが、上流工程で決めなさいという示唆は重要であり多く利用されている。

1.2 要件定義をめぐる環境の変化

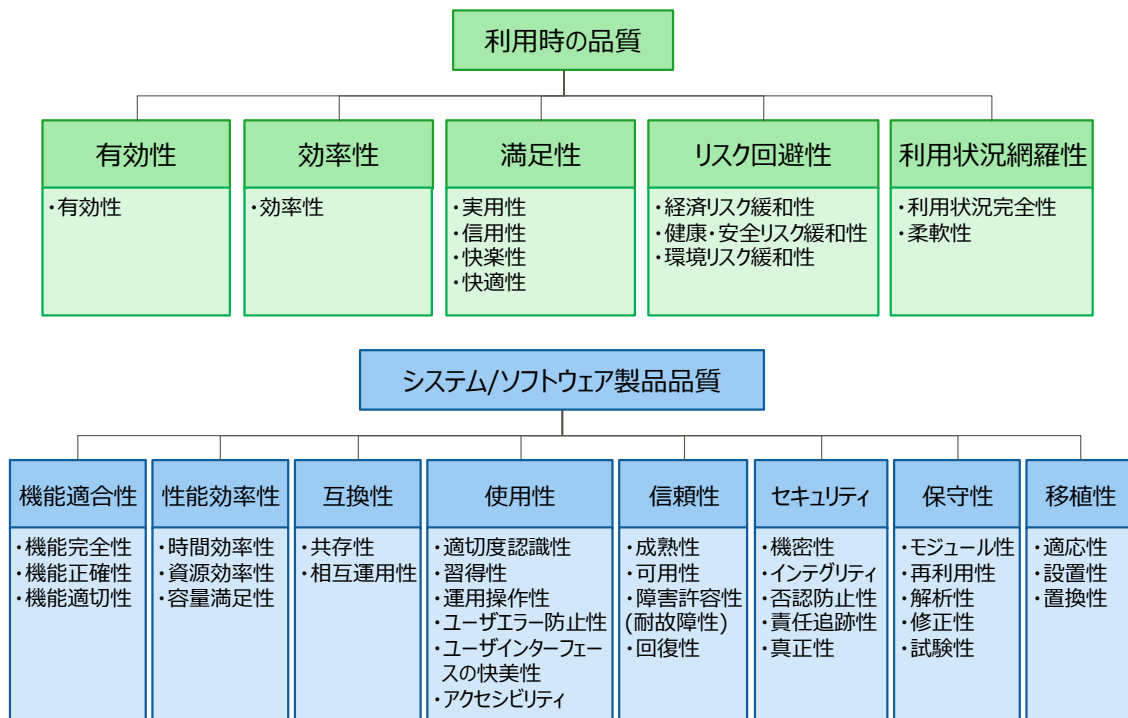


図 1.14 JIS, ISO の品質の考え方 JIS X 25010:2013 (ISO/IEC 25010:2011)

1.2 要件定義をめぐる環境の変化

1.2.4 ますます難しくなっている要件定義

要件定義不要とも思われる話を聞くことがある。それは、環境変化が激しく、先が読めない状況や、本当のお客様のニーズが分からない状況では要件定義が不要、という誤解から来ている。これは、要件定義が不要なわけではなく、今までの要件定義の進め方ではうまくいかないということを意味している。以下に、要件定義の進め方に工夫が必要になる代表的な3つの場面を挙げる。

【再構築時の要件定義】

一とおりシステム化が終わった昨今では、システム開発は既に存在するシステムの見直し、拡張などいわゆる再構築が多い。数十年使ってきた大規模で複雑な業務システムを刷新しようと大量の時間を要件定義に費やした結果まとまらなかった、業務やシステムを理解している人がいなくなり仕様が書けない、現行踏襲というあいまいな要求になる、など再構築時の要件定義が難しい。これは、現状の可視化や理解、変える変えないの判断などを工夫する必要がある。

【本当に役立つ要求が分からない】

将来が見通せない、新しい顧客層でありその顧客要求が分からない、実際に利用してみないと分からないなど、そもそも時間をかけて要件定義を実施しても正しいという確証がないようなものも多くなってきている。これは、重要な要求の見極めやリリースしてから評価するなど、リーンスタートアップやアジャイル開発を前提とした要件定義の進め方が必要になる。

【アイデア創出】

ビジネスモデルチェンジやイノベーションなども求められてきている。デジタル革新や新サービス開発など今までの概念から離れてアイデア自体を創造する。これは、詳細は省略するが、デザイン思考やUX (User Experience) アプローチなどアイデア発想アプローチを取り入れた要件定義を実施するなどの工夫が必要になる。

理想的には、再構築時の要件定義、イノベーション時代の要件定義なども含めて本書に記載していきたいところだが、まず基本領域を対象にガイドとして作成した。はじめにの図始.4 で示しているとおり、要件定義の基本領域、モダナイゼーション領域、イノベーション領域に分けて取り組みを行っている。

本書では対応していないが、このように環境の変化や時代のニーズ変化があることを意識して欲しい。いずれにしる要件定義が重要なポイントの一つである。

第2章 要件定義の問題認識

2.1 各ステークホルダが意識すべきこと

2.1.1 経営層が意識すべきこと

【良き構想なくして良き要件定義はない】

経営や業務に貢献するシステムを作るためには、十分な予算と期間を取ってビジネスの構想やシステムの構想を立案し、それを要件定義につなげていく必要がある。要件定義は構想立案の結果をもとに実施する工程であるが、ダメな構想のもとではいくら要件定義をがんばっても意味がない。単なるシステム置き換えプロジェクトでなければ、業務や組織の仕組みの変更が伴う。ビジネス構想が大前提になっているはずである。要件定義が始まる前に構想立案の確認を行うことが重要であり、不備があったら整備の時間を設けるべきである。

【要件定義プロジェクトの体制がシステム部門に偏っている】

システム開発はシステム部門の仕事という固定概念を捨てる必要がある。経営や業務に貢献するシステムを構築するためには、決してシステム部門中心の体制ではダメというわけではないが、経営層や業務部門の主体的な参画が欠かせない。

【経営層の参画、リーダーシップ】

経営層は、ビジネス構想やシステム構想を要件定義プロジェクトに浸透させる必要がある。業務部門中心での体制を作ったからといって、構想に従ったシステムができあがるとは限らない。業務部門（特に現場）の基本的思考は現状作業のムダ、ムラ、ムリの改善であり、今行っていることの大きな変化を嫌う保守的な傾向を示すこともある。紙の上に構想を書いただけでは、経営層の思ったとおりのビジネスプロセスやシステムにならないことがある。システム化によって、経営として何を目標にするか明確にすることが重要であり、要件定義後に要件定義で実現することが、本来の目標をクリアできるか、確認しなくてはならない。そのレベルの目標にすることが重要であり、数値レベルの具体的な目標あるいは、達成状況が確認できるレベルの目標設定が必要である。この経営目標を達成できないのなら、再度要件定義の見直しが必要になる。経営層は積極的に要件定義プロジェクトに参画し、ビジネス構想、システム構想を伝達し確認しともに考える。また、業務変革の権限を業務部門に委譲するなどビジネス改革の強い意志を明確に示す必要がある。なお、経営層という表現は必ずしも社長や事業担当役員のみを意味するものではない。経営者の意を汲んだ企画担当スタッフなどが代替して参画することもケースによっては可能である。

経営層の参画とリーダーシップが成功のカギを握っている。

2.1 各ステークホルダが意識すべきこと

【非機能要求の重要性が増してきている】

非機能要求とは、システム開発で検討が必要な機能要求（システムに搭載する、業務プロセス順序やその仕様）以外の要求すべてを指す。例えば、性能や拡張性といったものがこれに該当する。これらは、システム技術色が強く、システム部門やベンダ企業に任せる場合が多かった。しかし、昨今は非機能要求の重要性が増してきていて、経営や業務の重要な関心事になっている。大きな事件や災害時の事業継続やセキュリティなどが身近な企業リスクとして重要視されている。これらも非機能要求である。非機能要求は、企業内の問題だけでなく、企業の社会的責任として捉えなければならず、経営そのものの要求も含まれる。

【要件定義工程で十分な工期、工数を割り当てる】

情報システムの開発期間を、大きく要件定義、設計～結合テスト、ユーザ総合テストの3工程に分けて、それぞれに要した期間が全体工期に占める割合をシステム規模（工数）、新規開発と再開発・改修に分類して調査した結果が表 2.1 である。この表を見ると、要求の規模が増えても、要件定義に割いている工期の比率は、どの規模においても20%強である。

表 2.2 には、開発工程別の工数比を示す。新規開発と再開発でやや様子は異なるが、規模が大きくなっているにも関わらず、要件定義への投入工数の割合が少なくなっている。

両表を合わせて見ると、全般に、要件定義の工期割合は、投入工数の割合に比べて大きい。また、新規開発の場合、要件定義を含む各工程の工期割合は開発規模によらずほぼ一定であるが、要件定義の投入工数割合は開発規模が増えるに従って小さくなっている。これは、実装やテストでは規模見合いで要員を投入することにより進捗が進むものの、要件定義では要員を増やすことが進捗に必ずしも効果がないことを表している。

要件定義は、本当に必要な人をアサインし、その人に十分な期間を与えることが成功のポイントである。「一文惜しみの百知らず」。ここでの工期延長、工数増加を惜しんで未決事項を残した結果、設計、実装、テスト期間や負荷が膨張して予算超過になるケースは後を絶たない。

2.1 各ステークホルダが意識すべきこと

表 2.1 開発工程別工期比

全体工数	開発種別	件数	要件定義からユーザー統合テストまでの工期を100%とした工期の割合		
			要件定義	設計～統合（結合）テスト	ユーザー総合テスト
500人月未満	新規	25	21.8%	60.4%	17.8%
	再開発・改修	37	24.3%	61.0%	14.7%
	合計	62	23.2%	60.8%	16.0%
500人月以上	新規	8	21.7%	58.2%	20.0%
	再開発・改修	6	20.1%	59.7%	20.2%
	合計	14	21.1%	58.8%	20.1%
合計	新規	33	21.8%	59.6%	18.6%
	再開発・改修	43	23.2%	60.7%	16.1%
	合計	76	22.5%	60.2%	17.3%

(出典) 日本情報システム・ユーザー協会「ソフトウェアメトリックス調査 2016」 [7]をもとに作成

表 2.2 開発工程別工数比

区分	全体工数	件数	合計を100%とした比率		
			要件定義	設計～統合（結合）テスト	ユーザー総合テスト
新規	10人月未満	0			
	50人月未満	9	16.1%	68.2%	15.8%
	100人月未満	7	13.1%	73.9%	13.0%
	500人月未満	10	16.6%	70.6%	12.8%
	500人月以上	4	10.6%	65.7%	23.7%
	合計	30	11.8%	66.9%	21.4%
再開発	10人月未満	4	19.1%	68.7%	12.2%
	50人月未満	7	19.4%	66.7%	13.9%
	100人月未満	14	13.2%	73.4%	13.4%
	500人月未満	16	12.0%	76.6%	11.4%
	500人月以上	7	4.9%	76.7%	18.4%
	合計	48	7.9%	76.2%	16.0%
合計	10人月未満	4	19.1%	68.7%	12.2%
	50人月未満	16	17.4%	67.5%	15.0%
	100人月未満	21	13.2%	73.5%	13.3%
	500人月未満	26	13.4%	74.8%	11.8%
	500人月以上	11	7.8%	71.1%	21.1%
	合計	78	9.6%	72.0%	18.4%

(出典) 日本情報システム・ユーザー協会「ソフトウェアメトリックス調査 2016」 [7]をもとに作成

2.1 各ステークホルダが意識すべきこと

「まだ要件定義で決めなければならないことが多いが、設計で決めれば何とかなのではないか。実装を行うプログラマーも準備しているので、先を急いで欲しい」などのベンダの要望もあって、要件定義打ち切りとなってしまうことが多い。

しかし、要件定義工程で気がついた修正に要する作業負荷は少ないが、修正の作業負荷は、設計、実装、総合テスト、稼働後と工程が先に進めば進むほど大きくなることに留意し、要件定義に取り組む必要がある。

【既存システムの現状認識】

既存システムのレガシー化が、システム開発の足をすくうことがある。既存システムが巨大化、複雑化し、既存システムのドキュメントも陳腐化、システムの理解者も存在していない、さらには業務部門にも業務精通者が存在しないという「ブラックボックス化」が起こっている。システムをリビルドする（作り直す）には、既存業務やシステムの知識が必要である。大きな業務改革を実現しようとしても既存業務の多くはそのまま残るはずである。新しい仕組みは要求や仕様として意図を持って記載できるが、既存業務と同じという仕様は正確に書けるのかという問題に直面する。この問題に対応するために、経営層は、システム部門が抱えている既存システムの現状を理解する必要がある。現在動いているのだから問題ない、経営や業務に貢献するシステムを早く安く作れという経営層の要求をシステム部門は簡単に実現できないのである。「現行どおり」という要求提示は、要件定義ではない。具体的に「現行機能」を定義できないならば、要件定義は終了できないのである。

【経営者が参画する要求品質の確保 ～超上流から攻める IT 化の勘どころ～】 [1]

独立行政法人情報処理推進機構（IPA）が、システム開発の超上流フェーズを進めるにあたって気を付けるべきポイントを、17 個の原理原則とその基本的な考え方、および行動規範という形でまとめている。「ユーザ企業の積極的な関与がプロジェクトを成功に導く」としているので参考にして欲しい。

- 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件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] システム化方針・狙いの周知徹底が成功の鍵となる
- 原理原則[9] 要件定義は発注者の責任である
- 原理原則[10] 要件定義書はバイブルであり、事あらばここへ立ち返るもの
- 原理原則[11] 優れた要件定義書とはシステム開発を精緻にあらわしたもの

2.1 各ステークホルダが意識すべきこと

- 原理原則[12] 表現されない要件はシステムとして実現されない
- 原理原則[13] 数値化されない要件は人によって基準が異なる
- 原理原則[14] 「今と同じ」という要件定義はあり得ない
- 原理原則[15] 要件定義は「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] 要件定義は説明責任を伴う

コラム 3 システム化に際して経営者の果たすべき役割

経営改革・革新に役立つシステムであること

システム化に際して配慮すべき3階層

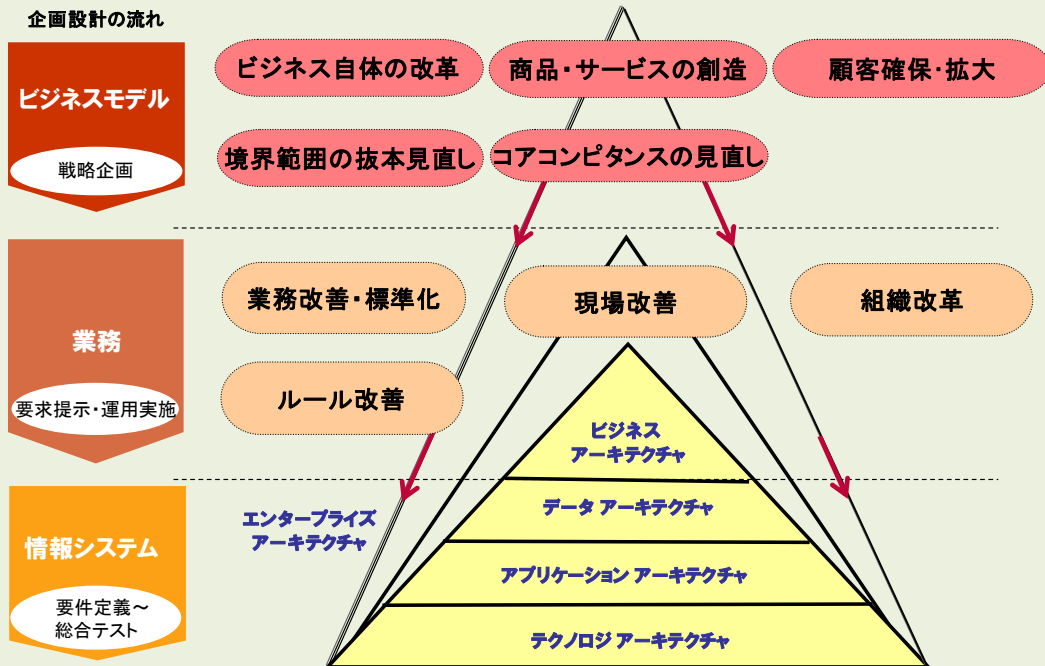


図 2.1 システム化企画における業務の階層構造

図 2.1 はシステム化に際しての 3 種類の業務の存在を示している。

図の最下段は情報システムの作り方の項目であり、エンタープライズアーキテクチャ (EA) のデータアーキテクチャからテクノロジーアーキテクチャに相当する。中段はその情報システムを利用者が業務にどのように活用すべきかを問う項目であり、EA のビジネスアーキテクチャに相当する。ここでは業務説明、業務機能、業務フローが書かれ、これを元にこれらの各種改善をする。最上段は情報システムの作り方や業務の活用以前に、ビジネスをどのように推進していくべきかを問うビジネスモデルの項目であり、ビジネスそのものを考える。

システム化検討の際に経営者に最も実施していただきたいのは、最上段から見た判断である。

- 自社を取り巻く経営環境はどのように変化していくのか、
- それに先手を打って競争力をつけるためには自社の商品サービスをどのように変えていけば良いのか、どこの誰に買ってもらえるのか、

2.1 各ステークホルダが意識すべきこと

- 顧客、販売店、株主、銀行、協力会社、従業員などの期待は何で、どのように対応すべきか

これらの項目を最も正しく理解し展望をもっているのが経営者である。

業務部門、情報システム部門から多額の投資を必要とするシステム化提案が提出されてきた場合には、以下のような考慮ができるのが経営者である。

- 今すぐ実行すべき課題なのか、新しい技術が登場してきて別の方法で解決されることは予想できないか
- 自社がなすべき課題の中で、今回のシステム化提案は最優先すべき課題か
- 提案内容は現状の改善にこだわり過ぎていないか、予算をもっと大きく与えれば、もっと素晴らしい方法が提案できるのではないか
- このシステム化を実施するとして、システム化前に解決しておくべき課題はないか
- できあがったシステムを使いこなすためには業務部門の運用レベルを向上させておく必要がないか
- 期待する効果を出すためには、何をする必要があるのか
- 効果を出す責任者は誰で、このプロジェクトにどの程度参加しているのか

「私には情報システムの問題はよく分からない」と仰る経営者の方がまれにおられるが、そのような方には「システム化企画の内容を理解するために、何についてどのような説明を求めているかを、企画立案者に提示してください」とお願いしたい。

- 当社の本質的問題と関係付けて説明せよ
- 英語の3文字単語は使用しないで説明せよ
- 開発上のリスクは何か
- 開発、運用の最適責任者は誰か、成功させる信念をもっているか

など具体的に指示して、内容を理解した上で企画の採否を行う努力をして欲しい。

以上が経営者に対する期待であるが、最近ではセキュリティや情報システムのトラブルによる業務の中断などによりお客様に迷惑をかける場合も散見されるので、非機能要求についての配慮も欠かせない。

社員からの説明に不安を感じたら外部の有識者の知恵を借りるなどにより、適切な判断ができるようにしていくことが望ましい。

本コラムは、システム構築上流工程強化部会 細川 泰秀委員に寄稿いただいた。

2.1 各ステークホルダが意識すべきこと

2.1.2 業務部門が意識すべきこと

【経営や業務に貢献する】

既存業務のほとんどが既にシステム化されている昨今、経営層は経営や業務に直接貢献する IT システムを求めている。すなわち、システム投資の目的は、システム開発ができたかではなく、経営や業務にどれだけ貢献するかである。これはシステム開発プロジェクトでも、システム部門の問題ではなく、業務部門の問題であることを意味している。

【ビジネスプロセス改革、改善】

ビジネスや業務のプロセスが変わらないのに、システムだけ最新のものに置き換えることには意義がない場合もある。業務部門は、構想立案の方針にしたがい具体的なビジネスプロセス全体を作り上げなければならない。その際には、IT システムを業務に対するサービスであると思って、ビジネスプロセスに組み込むことが必要である。

【業務部門のリーダーシップ】

IT システムとは、業務の一部の機械化もしくは業務に対するサービスの提供であり、業務を遂行するための手段である。すなわちシステムは業務の一部であり、どのようなシステムを作成するか、そしてそのシステムからどのような効果を引き出すかは業務部門の責任である。業務部門のリーダーシップ、要求仕様の提示責任、システムの検証責任などが重要になる。当然、すべての要件定義が業務部門でできるわけではないが、システム開発をシステム部門だけの仕事にせず、業務部門が自分のこととして捉える「態勢」を作ることが大切である。現状の業務の進め方に満足することなく、業務部門として提供すべきサービスを、そのサービスの受益者の立場になって最適化するとともに、業務の効率化や適切な情報提供を通じて経営に貢献するという視点が必要である。詳細は、「6 章 要件定義マネジメント (RM) における課題と解決の勘どころ」「6.1.3 オーナーの選定 (RM1.3)」に記載しているので参照願いたい。

【現行業務、システムの理解】

業務部門は現行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を意識しなければならない。

自部門の業務を IT 化する場合、各作業を人手、機械、システムのうちどれで行うかの組み合わせを考えることになる。特にシステムにどのような機能を乗せるのか、その機能を今後どのように改善すればさらに自部門の効率化が図れるのか、業務部門は理解する必要がある。

一定規模以上の企業では、業務処理に何らかのシステムを利用している。システム開発時点で想定していなかった使い方をされている場合もあるし、使われていない機能もある。シ

2.1 各ステークホルダが意識すべきこと

システムの機能がブラックボックス化して、利用者には理解されていないことも多い。利用者からみてシステム部門やベンダ企業に対する満足度が低い場合もある。

そのため、図 2.2 に示すように、システム開発の前段階で数カ月前から時間をかけて、「次期システムには何を期待するのか」「今のシステムで使用されていない機能はないか」「数年前の開発時と異なり現在の環境にふさわしくない機能はないか」を整理、分析する期間を設けることが望ましい。その過程で「もっと効率的な、もっと顧客に喜んでもらえる機能や仕組みがないか」を、関係者にヒアリングすることも重要だ。

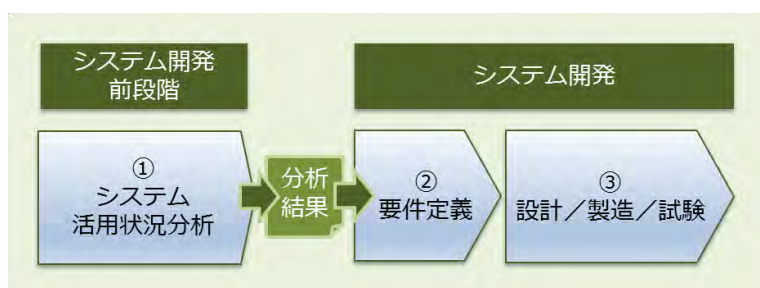


図 2.2 システム開発前段階を含めたプロセス

【合意形成】

昨今のシステム開発は関係するステークホルダが多岐にわたり増えている。システム化の対象となる範囲が広がった結果、システム開発に関係する部門が増えるだけでなく、利用者も他企業やコンシューマにまで広がっている。これらステークホルダの方針、意見、課題などについて、漏れなく綿密に把握し、できることとできないことをユーザ企業、ベンダ企業がともに切り分け、業務要件として取りまとめて合意形成にもっていく責任がある。また、ステークホルダの要求は衝突することがあり合意形成が困難なことがある。利用部門は自部門だけではなく、企業全体の利益を考え合意形成に望む必要がある。

2.1 各ステークホルダが意識すべきこと

【仕様変更が生じた場合を想定した予算確保】

要件定義で確定した要求仕様には、その後も変更が発生しないことが理想であるが、現実には難しい。その場合を考慮して、予算をあらかじめ余分に確保しておくことが大切だ。一度確定した要求仕様を変更することになった場合、それ以降の期間と費用に影響が出る。多くの場合は予算の増額が必要になるが、プロジェクトの開始時点で決定した予算の増額はそう簡単には認められず、その折衝にさらに時間と労力がかかることになる。そこで、プロジェクトの予算を計上する際には、一般にコンティンジェンシー予備やマネジメント予備と呼ばれる予備の予算を確保しておくことが重要になる。

表 2.3 に仕様変更が発生した際の対応予算をあらかじめ組み込んだプロジェクトの比率と仕様変更の発生比率を示す。

表 2.3 仕様変更の見込みと仕様変更費（総開発費に対する割合）

仕様変更をあらかじめ計画 (予算確定) に		仕様変更の発生		合計
		発生した	発生しなかった	
含めた	件数	243	25	268
	割合	90.7%	9.3%	100.0%
	総開発費に対する 割合の平均	9.5%		9.5%
含めなかった	件数	132	73	205
	割合	64.4%	35.6%	100.0%
	総開発費に対する 割合の平均	9.2%		9.2%
合計	件数	375	98	473
	割合	79.3%	20.7%	100.0%
	総開発費に対する 割合の平均	9.4%		9.4%

(出典) 日本情報システム・ユーザー協会「ソフトウェアメトリックス調査2019」 [14]をもとに作成

なお、2016年版のこの調査では、仕様変更予算を10%程度持つておくと、仕様変更が発生した場合の対応が円滑になるとも報告されている。

2.1 各ステークホルダが意識すべきこと

2.1.3 システム部門が意識すべきこと

【既存システムの状況の明確化】

システム部門では優先順位の高い経営課題を IT で解決したいと考えているものの、既存システムの巨大化、複雑化などのレガシー化が足かせになり思うようなシステム開発が行われないというジレンマがある。システム部門からは、「経営層の IT 化に対する理解度が低い」という声もよく聞く。経営層は、システムが動いている限り、既存システムがレガシー化している問題を経営課題として認識していないことがある。また、開発をベンダ企業に依頼するとき既存システムがレガシー化されていることを明示しないで、開始後に発覚することがある。レガシー化したシステムの再構築の起案は、現状業務が大きく変えずにきれいにするだけでは、経営層から見れば価値がよく分からない。今のシステムで良いのではとなる。

システム部門は、既存システムの状況を明確にし、経営課題として経営層に認識させる必要がある。

【ベンダ企業の選定】

要件定義に参加する IT ベンダ企業は、一般にシステム部門が選定するが、システム部門のベンダ選定基準があいまいなことが多い。例えば、「有名なベンダ企業に開発を依頼したから大丈夫」、「一番安いから」、「ベンダが提案したのだからやらせてみよう」など、無責任な理由でベンダを選定することがある。現実問題として要件定義に IT ベンダの協力は欠かせず、そのベンダを適切に選定する責任はユーザ企業側にある。ユーザ企業は、本当に、自企業の課題解決や要件定義に協力してくれる価値あるベンダ企業を選ぶための明確な基準を設定しておく必要がある。

【システム部門のパラダイムシフト】

1.2.2 で示したように、システム部門は今まで以上にビジネス視点を持ち、情報化やシステム活用を考え推進していく必要があり、変わっていく必要がある。開発技術だけではなくビジネス指向で経営層や業務部門と対話し推進する組織へと変化する必要がある。システム部門がビジネスや業務を理解し、業務部門に対して価値あるシステムを提案していく。昨今は、業務部門の要求を鵜呑みにして、言われたとおりのシステムを作る「御用聞き」では経営や業務に貢献するシステムにはならない。

要件定義は、業務部門（オーナー部門）責任が基本だが、システム部門が触媒となり業務部門を引っ張っていき、IT 技術活用を提案していく姿が理想である。それはベンダ企業側にも言えることだ。

2.1 各ステークホルダが意識すべきこと

【プロジェクトマネジメント】

要件定義でもマネジメントは重要である。大まかには、「計画立案」「実行、監視・コントロール」「評価、完了判断」を確実に行うことである。

要件定義では、要件定義に特有なマネジメントが必要になる。例えば「膨らむ要求をいかにコントロールするか」「要件定義の完了をどう判断するか」「要件定義の品質をどう担保するか」などである。本改版ではマネジメント部分を強化しているので本編を参照願う。

2.1 各ステークホルダが意識すべきこと

2.1.4 ベンダ企業が意識すべきこと

【要件定義の契約基本形は準委任契約である】

要件定義は準委任契約を基本とし、ユーザ企業主体で進める。ソフトウェア開発は全工程一括請負が大半であった。その流れで要件定義も一括請負で行うことがあり、問題が発生している。しかし、要件定義の主体をユーザ企業にし、準委任契約で行うことにより問題が減少している。

詳細は、6.2.8 調達計画と契約(RM. 2.8)で述べているので参照願う。

【ユーザ企業課題解決貢献】

ビジネスを変革するのに IT 技術を使うというビジネス変革先行型もあるが、IT 技術があるから何かができるのではという IT 技術先行型もある。経営や業務に貢献するためには IT 技術が欠かせないことは同じだ。1.2.2 や「システム部門のパラダイムシフト」で示したのと同様、ベンダ企業も変わっていく必要がある。お客様のビジネス課題をお客様と一緒に考え、ソリューション、ビジネス価値を提供する企業へと変化する必要がある。そのためには、要件定義に入り、業務部門を引っ張りながら、IT 技術活用を提案していく姿が理想になる。

【見積り】

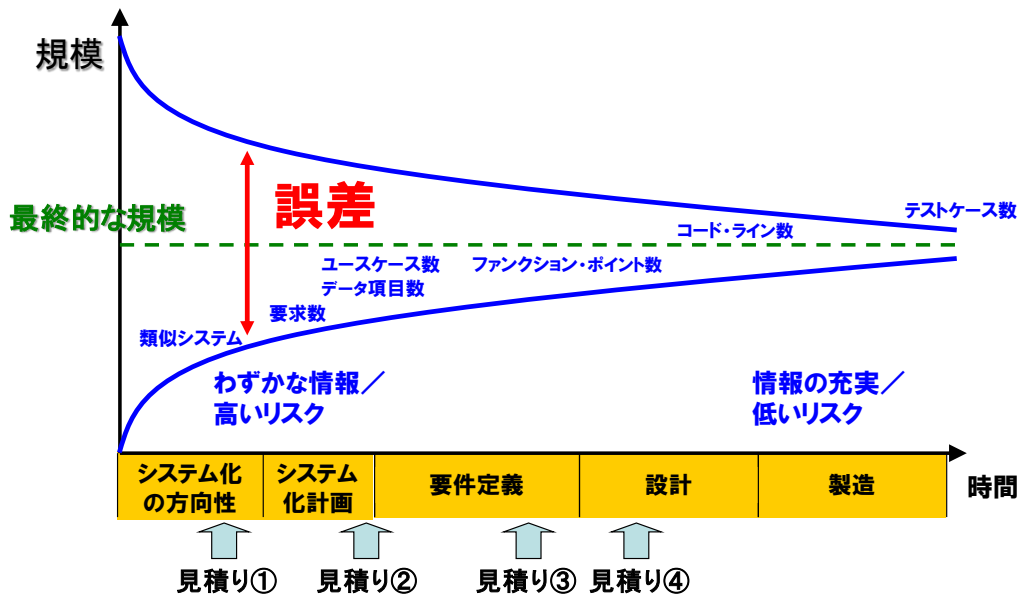
プロジェクトに仕損を出す要因として見積りの誤りがある。原因は以下である。

- ・ 楽観的な見積り：見積り根拠があいまいで、必要な要素をきちんと見積もっていない。
- ・ 早期の見積り：作るものが決まっていない状況で見積りを行う。
- ・ 目標と見積りの混同：営業、マーケティング担当者や顧客の事情で決定される。
- ・ 一度決まったら修正されない見積り
- ・ 時間の余裕がないことが多い（十分に取組めない）。

(*) 出典：ロバート・グラス「ソフトウェア開発 55 の真実と 10 のウソ」より

ソフトウェア開発見積りの課題と解決方法 石谷 靖

IPA SDM 公開講座「現代ソフトウェアエンジニアリングの俯瞰図構」2012 [15]



(出典) Barry Boehm “Software Engineering Economics” (Prentice-Hall 社) [16]の図を元に作成

図 2.3 見積り時期とリスク

見積り精度は図 2.3 に示すとおり、時間を経るにつれて精度が上がるため、都度再見積りをするべきである。しかし、現実には歪んだ見積りになっているケースがある。例えば、概算見積りと言いつつ一度決まったら予算額として変えることができないことやその予算額に合わせるように見積もる（機能を根拠なく削る、部品化すれば減る、流用素材を持ち込むなど）ことが挙げられる。特にシステム開発商談時のベンダ企業は、安くすること（目標）に力を注ぎ本来あるべき見積りを見失う可能性がある。

【要求仕様の凍結】

開発を受託するベンダ企業にとって、要件定義のアウトプットである要求仕様は重要なインプットになる。ところが、要求仕様には、仕様の細部が決まらないとか、仕様が変化し続けるといった仕様未凍結の問題があり、その度合いによってはプロジェクトに仕損を出すことが多々ある。

そうならないようにするために、要件定義の完了時点で要求仕様を凍結できるようにユーザ企業、ベンダ企業ともに努力することが重要である。

決まらない要求仕様を統制するためには、スコープの広がりや要求量、変更量を把握し、ベースラインを定めて適切な要求量、変更量の管理を実施する必要がある。妥協した要求仕様があるかもしれないが、それらは稼働後等に追加する、改良するということも考慮する必要がある。ベンダ企業は、変更は仕様変更として費用をいただく、期間の延長もあり得る等の契約を結ぶのが一般的である。リスクはあるが予定以上の収入を得られるため、仕様変更があっても良いという緩みが出てしまう。ベンダ企業は、要件定義に積極的に参画し要件を

2.1 各ステークホルダが意識すべきこと

確実に決めるべく、リスク、コスト・期限・資源などの制約事項、暫定策・代替策などの現実解等を意識し、スコープと変更を統制して、スケジュールと予算と要件のバランスを取るようコントロールし続ける必要がある。

2.2 要件定義の問題

2.2.1 問題の抽出と整理

本書は問題に対する解決策の一助として勘どころを提示することを目的としている。初版では問題や勘どころを絞り込んだため多くを提示できなかった。第2版作成にあたって、少しでも多くの問題と解決のための勘どころを掲載するため、当WGのユーザ企業およびベンダ企業の双方の委員に要件定義の問題をあらためて提示してもらった。

提示された主な問題は以下のように分類された。

- ・「現行業務やシステムが分からない」といった『現状把握の問題』
- ・「真の問題・課題が抽出できていない」といった『問題課題抽出の問題』
- ・「経営要求が不明確」「経営に貢献する要求にならない」といった『ゴール明確化の問題』
- ・「要求間の整合性がとれていない」といった『要求の体系化の問題』
- ・「要求に基づき新たな業務としてうまく具体化できない」といった『要求の具体化の問題』
- ・「要求が膨らみ、捨てる判断が難しい」といった『優先順位付けの問題』
- ・「各ステークホルダに対し要求の合意形成が難しい」といった『要求の交渉の問題』
- ・「成果物に抜け・漏れ・あいまいが存在する」といった『要求の仕様化の問題』
- ・「要件定義の不具合を抽出しきれない」といった『要求の検証の問題』
- ・「システム化仕様が経営や業務にどう貢献するか分からない」といった『妥当性確認の問題』
- ・「要件定義前の構想や企画が不十分」「開始前に必要なステークホルダが洗い出されていない」といった『立ち上げ時の問題』
- ・「要件定義で何をどこまで実施すれば良いのか分からない」といった『要件定義プロセス計画の問題』
- ・「そもそも開発スコープがあいまいである」といった『スコープ定義の問題』
- ・「要求件数では開発規模の膨らみが分からない」といった『見積りの問題』
- ・「要件定義ドキュメントの品質の確保方法が分からない」といった『品質計画の問題』
- ・「要件定義の体制が不十分である」といった『体制・チームビルディングの問題』
- ・「要件定義の契約形態がバラバラである」といった『契約の問題』
- ・「コミュニケーションが不十分である」といった『コミュニケーション計画の問題』
- ・「リスクを把握できていない」といった『リスク認識の問題』
- ・「要求調整・規模調整のコントロールができない」といった『要求・スコープコント

2.2 要件定義の問題

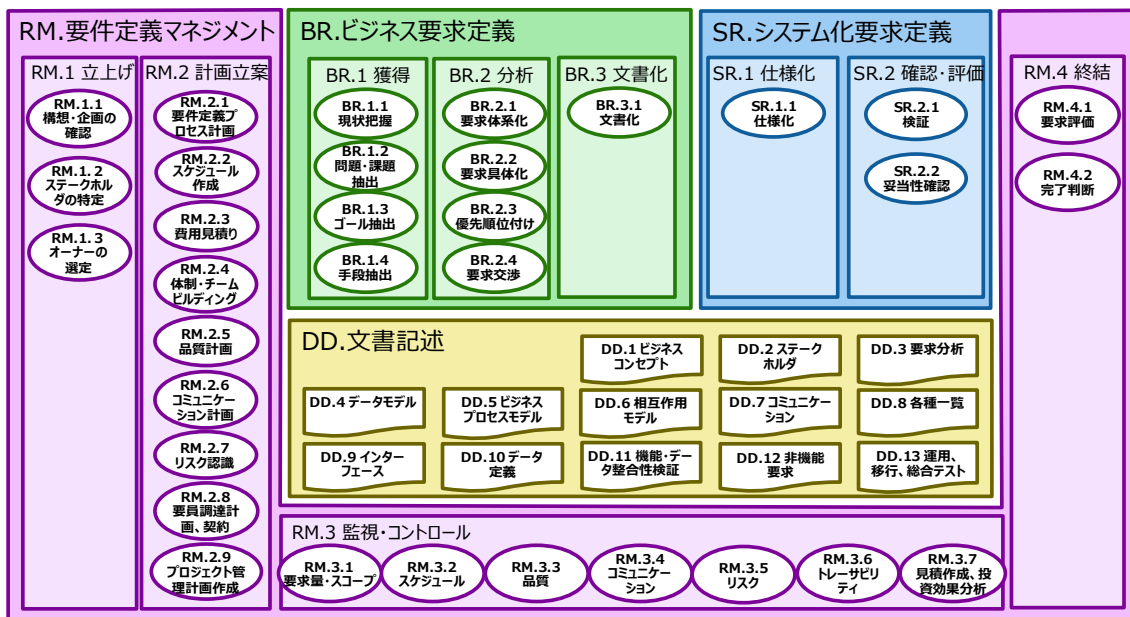
『ルールの問題』

- ・「要件定義の品質を監視できない」といった『品質の監視の問題』
- ・「コミュニケーションがうまくいっているか分からない」といった『コミュニケーション監視の問題』
- ・「課題やリスクが放置される」といった『リスクの監視の問題』
- ・「ドキュメント間の整合性がいつの間にか失われる」といった『トレーサビリティ管理の問題』
- ・「要件定義完了時期に未決定要求が残る」といった『要求評価の問題』
- ・「要件定義の完了を判断できない」といった『完了判断の問題』
- ・「要件定義ドキュメントのサンプルや書き方のコツが欲しい」といった『要件定義ドキュメント記載上の問題』

などが抽出された。詳細は次章を参照願う。

初版では対象外としたマネジメント系の問題が多く抽出されている。

問題・解決のための勘どころを体系だって説明していくために、本WGで抽出した問題を分類、整理し、以下の問題カテゴリマップを作成した(図 2.4)。



BR:Business Requirements SR:System Requirements DD:Documents Description RM:Requirements Management

図 2.4 要件定義問題カテゴリマップ

「ビジネス要求定義 (BR: Business Requirements)」「システム化要求定義 (SR: System Requirements)」「要件定義マネジメント (RM:Requirements Management)」「文書記述 (DD:Documents Description)」の4つの大カテゴリに分けている。

2.2 要件定義の問題

「BR. ビジネス要求定義」では企画工程で整理された経営レベルの利害関係者要求事項を業務の観点で獲得 (BR. 1 ビジネス要求の獲得) し、ビジネス要求の各項目間の関連の整理、優先度付け (BR. 2 ビジネス要求の分析)、文書化 (BR. 3 ビジネス要求の文書化) を行う。「SR. システム化要求定義」では「BR. ビジネス要求定義」で定義した要求を機能要件、非機能要件の観点で詳細化、文書化 (SR. 1 システム化要求の仕様化) し、検証や妥当性確認 (SR. 2 システム化要求の確認・評価) をして要件として定義する。

「RM. 要求マネジメント」は要件定義工程遂行プロジェクトの立上げ (RM. 1 立上げ) と計画 (RM. 2 計画立案)、「BR. ビジネス要求定義」や「SR. システム化要求定義」の活動中における監視・コントロール (RM. 3 監視・コントロール) と要件定義工程の実行結果の評価と完了判断を行う終結作業 (RM. 4 終結) に分類している。

2.2 要件定義の問題

2.2.2 要件定義の問題と解決のための勘どころ

前記の問題カテゴリマップにもとづき、要件定義において発生する問題と解決のための勘どころの対応を以下に示す。

(1) ビジネス要求定義の問題と解決のための勘どころ

ビジネス要求定義プロセスにおいて発生する問題とその解決のための勘どころおよびその説明記載箇所を一覧で示す。

表 2.4 ビジネス要求定義の問題と解決のための勘どころ

問題カテゴリ (記載箇所)		問題	解決のための勘どころ
BR1 ビジネス 要求の獲得 (4.1)	BR1.1 現状の把握 (4.1.1)	<ul style="list-style-type: none"> ■ 現行業務やシステムを理解するのが困難で、正確な現状把握ができていない ・ 既存ドキュメントが陳腐化しておりそのまま利用できない ・ 業務を理解している人材が減っている ・ 対象業務やシステムが複雑、広範囲で理解し難い 	(1) 業務・システムの状況をさまざまな方法を活用して把握する ① 現行システムから把握する ② 全体像を作成して把握する ③ 現行業務を再学習して把握する ④ 埋もれている既存資料を活用して把握する ⑤ 実態と照らし合わせることでより把握する
		<ul style="list-style-type: none"> ■ 現状業務やシステムの分析をしたが、分析した当人以外、誰も内容を把握できておらず、分析結果や全体整合の結果が正しいかがプロジェクトとして判断できない ・ 自分が担当した部分を他の誰もわかっていない ・ 全体を把握している人がいない ・ 業務運用担当者が理解していない 	(2) 現状業務・システムを理解・共有する ① プロジェクト全員の共通認識プロセスを実施する ② 稼動後人材の育成を要件定義から始める

問題カテゴリ (記載箇所)	問題	解決のための勘どころ
BR. 1.2 問題・課題の抽出 (4.1.2)	<ul style="list-style-type: none"> ■ ステークホルダを見誤り、必要な要求が抽出されない ・ 詳細な問題・課題がたくさん出てきて收拾がつかない。 ・ 真の問題・課題が出てこない ・ 問題・課題の抽出に漏れがある 	(1) 適切な問題・課題の抽出・分析を行うために、ステークホルダの特性を理解する ①ステークホルダを漏らさない ②各ステークホルダの問題・課題認識の違いを認識する ③ステークホルダ間の対立・関連を見極める
	<ul style="list-style-type: none"> ■ 真の問題・課題の抽出ができていない ・ 効果的な要求（対策）にならない 	(2) 問題の原因を分析し解決テーマ（課題）を抽出する ①問題と課題の違いを意識する ②要求の源泉を分析する ③なぜなぜ分析を行い、真の原因を見極める ④解決テーマ（課題）を適切に設定する
BR. 1.3 ゴールの抽出 (4.1.3)	<ul style="list-style-type: none"> ■ 経営に貢献する要求、経営方針に合った要求が抽出されない ・ 経営レベルの要求を収集していない ・ 経営レベルの要求を経営層と確認していない 	(1) 経営レベルの目的・目標を明確にして共通認識する ①経営レベルの目的・目標、経営施策を明確にする ②経営レベルの目的・目標を見極める ③経営施策（手段）を見極める
	<ul style="list-style-type: none"> ■ 経営に貢献するためのサブゴールである業務レベルの目的・目標が明確でない ・ 業務として目指すものが不明確 ・ システム化要求が経営レベルの要求にどのように貢献するかが不明確 ・ システムの操作性要求が多くなる 	(2) 業務レベルの目的・目標を抽出する ①目的と手段の違いを意識する ②目的と目標の違いを意識する ③目的・目標にはレベルがあることを意識する ④業務レベルの目的・目標を抽出する
BR. 1.4 手段の抽出 (4.1.4)	<ul style="list-style-type: none"> ■ 効果的な手段が抽出できない ・ 経営や業務に目的・目標に効果的な手段が抽出できない ・ システム化の手段やシステム操作性の手段が多くなる 	(1) 目的・目標を意識して手段を抽出する ①手段のレベルを意識する ②手段の十分性から他の手段がないか検討する ③業務手段・システム手段の双方を抽出する

2.2 要件定義の問題

問題カテゴリ (記載箇所)		問題	解決のための勘どころ
BR. 2 ビジネス 要求の分 析 (4. 2)	BR. 2. 1 要求の体 系化 (4. 2. 1)	<ul style="list-style-type: none"> ■ 抽出した要求が全体として効果的かつ必要十分なものになっていない ・ 目的に合致しない要求が多くある ・ もっと効果的な手段がある ・ 膨らむ要求を捨てられない 	(1) 要求全体の整合性を分析し評価する ① 要求を構造化する ② 妥当性の観点から真の目的を見極める ③ 充分性の観点からより効果的な手段がないか検討する
		<ul style="list-style-type: none"> ■ 要求に具体性がない ・ 具体的な数値目標がない ・ 具体的な対策になっていない 	(2) 具体的な測定尺度を検討する ① 目標に評価指標を設定する
	BR. 2. 2 要求の具 体化 (4. 2. 2)	<ul style="list-style-type: none"> ■ 要求を反映した業務の情報構造にならない ・ 業務（情報）の整理ができていない ・ 業務の複雑さを低減できない ・ 業務用語を正しく理解していない ・ 業務の構造を把握していない ・ データモデルは業務部門には難しい 	(1) 情報構造の観点から業務を可視化し、新しい業務として具体化する ① 管理対象のバリエーションを整理する ② 管理対象のバリエーションを削減する ③ 業務ルールと照らし合わせて概念データモデルを描く ④ 工夫して業務部門と確認する
		<ul style="list-style-type: none"> ■ 要求を反映した業務プロセスにならない ・ ビジネスプロセスが変わらない ・ 業務プロセスを標準化できない ・ 変革ポイントが分からない ・ 要求が反映されているか分からない 	(2) ビジネスプロセスの観点から業務を可視化し、新しい業務として具体化する ① 問題・課題、要求と照らし合わせてプロセスモデルを描く ② 要求との紐付けをする ③ 業務プロセスのバリエーションを整理する ④ 業務パッケージの適用を検討する
	<ul style="list-style-type: none"> ■ 情報と業務プロセスの関係が複雑で不整合が発生している ・ 情報と業務プロセスの整合性がとれていない ・ 業務パターンが多い 	(3) 情報とプロセスの両方の観点から業務を可視化し整合性を確認する ① 相互作用のモデルを描き、情報、業務プロセスの両面から業務の実現性を確認する ② 業務パターンを削減する	

問題カテゴリ (記載箇所)	問題	解決のための勘どころ	
BR. 2.3 優先順位 付け (4.2.3)	<ul style="list-style-type: none"> ■ 限られた工期やコストの中で要求を絞り切れない ・ 優先順位付けができていない ・ 絞り込み時ステークホルダの合意がとれない ・ すべての要求を正確に判断するのに時間を要する 	(1) 優先順位を明確にする ① 客観的な判断基準を明確に定義する ② 大きな判断と詳細な判断を使い分ける	
	<ul style="list-style-type: none"> ■ 優先順位を付けても要求を絞り切れない ・ 要求を捨てられない 	(2) 要求を棄却する ① 要求を捨てる ② 要求を先送りにする	
BR. 2.4 要求の交渉 (4.2.4)	<ul style="list-style-type: none"> ■ 経営層、事業部門長などうまく説明ができない ・ 成果物をそのまま見せてもわかってもらえない ・ 説明資料作成のタスクもなく、担当者もアサインされていない 	(1) 経営層や業務部門への説明を行う ① 説明相手の視点に合った資料を準備する ② 説明用資料の作成の計画を立てる	
	<ul style="list-style-type: none"> ■ 要求の絞り込み交渉がうまくいかない ・ 第三者からの説明に納得感が得られない ・ 声の大きい人に負ける ・ 衝突する要求を捌けない 	(2) 要求の絞り込み交渉と合意形成は、ポイントを意識して行う ① 当事者意識を持たせる ② 声の大きい人に負けてはならない ③ セレモニーでないレビューを行う ④ ステアリングコミッティを用意する	
BR. 3 ビジネス 要求の文 書化 (4.3)	BR. 3.1 文書化 (4.3.1)	<ul style="list-style-type: none"> ■ 要件定義で作成するドキュメントの全体像を理解していない ・ ドキュメント間での不整合が発生している 	(1) ドキュメントの全体像、関連を明確にする ① 成果物関連図を解説する

2.2 要件定義の問題

(2) システム化要求定義における問題と解決のための勘どころ

システム化要求定義プロセスにおいて発生する問題とその解決のための勘どころおよびその説明記載箇所を一覧で示す。

表 2.5 システム化要求定義の問題と解決のための勘どころ

問題カテゴリ (記載箇所)		問題	解決のための勘どころ
SR.1 システム 化要求の 仕様化 (5.1)	SR.1.1 仕様化 (5.1.1)	<ul style="list-style-type: none"> ■ 要件定義で作成するドキュメントの全体像を理解していない ・ ドキュメント間での不整合が発生している 	(1) ドキュメントの全体像、関連を明確にする ① 成果物関連図を解説する
		<ul style="list-style-type: none"> ■ 文章に抜け漏れあいまいがある ・ 後続の設計品質が低下する ・ ユーザテストで要求レベルの指摘が多発する ・ 要求の実装漏れ、仕様誤りが本稼働後に発生する 	(2) 文章のあいまいさを排除し要求を正しく伝達する ① 正確な日本語表現をする
SR.2 システム 化要求の 確認・評 価 (5.2)	SR.2.1 検証 (5.2.1)	<ul style="list-style-type: none"> ■ 記述内容の誤りを摘出しきれしていない ・ 後工程で要件定義書修正に伴う手戻りが発生する 	(1) 組織的なレビューをする ① レビューの準備(レビュー観点の整理)、レビュー実施体制、やり方を定義する
	SR.2.2 妥当性確 認 (5.2.2)	<ul style="list-style-type: none"> ■ システム化要求の妥当性を業務部門が確認するのが難しい ・ システム部門やベンダが作成したドキュメントを業務部門が分からない、レビューできない ・ 実現したいビジネス要求がシステム化仕様として定義できていない 	(2) 業務部門が分かるように成果物の作成およびレビューを工夫する ① ビジネス要求がどのようにシステム化要求に反映されているかが、相手に分かりやすく伝わるようにレビュー対象物を作成する

(3) 要件定義マネジメントにおける問題と解決のための勘どころ

要求定義マネジメントプロセスにおいて発生する問題とその解決のための勘どころおよびその説明記載箇所を一覧で示す。

表 2.6 要件定義マネジメントの問題と解決のための勘どころ

問題カテゴリ (記載箇所)		問題	解決のための勘どころ
RM. 1 立上げ (6.1)	RM. 1.1 構想・企画の 確認 (6.1.1)	<ul style="list-style-type: none"> ■ システム化の目的があいまいで、何をゴールに要件定義すれば良いのか目標がはっきりしない ・ システム化の目的やプロジェクト目標が達成できないシステムができあがる 	(1) 企画内容を確認するとともに、企画の目的とプロジェクト目標の目的意識を醸成する ① 企画内容を評価し、要件定義が開始できることを確認する ② 企画の目的とプロジェクト目標を共有し、目的意識を醸成する
		<ul style="list-style-type: none"> ■ 構想・企画立案で決めたスコープがあいまいで、要件定義計画立案時に不備が発生する ・ 要件定義が終わらない ・ 後工程で手戻りが発生する 	(2) 構想・企画の実現にはどの業務のビジネス要求分析が必要かを見定める ① プロジェクトの目標を実現するための要求事項を具体的に定義する ② 全体像を俯瞰し、システム化範囲が妥当であるかを確認する
	RM. 1.2 ステークホルダの 特定 (6.1.2)	<ul style="list-style-type: none"> ■ ステークホルダに漏れがあり、必要な要求を抽出できない ・ 後工程で手戻りが発生する ・ ステークホルダが使えない役に立たないシステムができあがる 	(1) 必要なステークホルダを特定・分析し、各ステークホルダへの対応方針を立案する ① 必要なステークホルダを漏れなく抽出する ② ステークホルダを分析し、対応方針を立案する
RM. 1.3 オーナーの 選定 (6.1.3)	<ul style="list-style-type: none"> ■ 要件定義は自分たちが責任を持って推進すべきものだとして業務部門が認識していない ・ 利用頻度に関わらず要求が出る ・ 部門ごとにシステム化に対して温度差がある ・ 実務で多忙なので IT システム部門にお任せ ・ システム開発費用の確保は IT システム部門の役目 	(1) システムオーナーの選定により、業務部門の役割と責任を明確にする ① システム構築投資を回収するのは業務部門の役目であることを明示する ② 業務部門に要件定義の結果や開発されたシステムの仕様確認に責任を持つと認識させる	

2.2 要件定義の問題

問題カテゴリ (記載箇所)		問題	解決のための勘どころ
RM. 2 計画立案 (6.2)	RM. 2.1 要件定義 プロセス 計画 (6.2.1)	<ul style="list-style-type: none"> ■ 本来必要なプロセス・成果物を省いたため、要件定義起因の手戻りが発生する 	(1) 要件定義で必要なプロセスと成果物を見極め、要件定義の計画を立案する ① 前後のプロセスを確認し、要件定義のプロセスと必要な成果物、記載レベルを定義する ② 合意形成プロセスを意識して計画を立案する ③ 成果物作成の担当を決める
	RM. 2.2 スケジュール作成 (6.2.2)	<ul style="list-style-type: none"> ■ 要件定義工程は、実行可能性の裏付けのあるスケジュールを作るのが難しい 	(1) 所要期間の裏付けをできるだけ作り、スケジュール上のクリティカルパスには、その旨を明示しておく ① 作業工数、所要期間が見積もれるものは、できるだけそれを活かす ② 所要期間に大きな影響を与えるレビュー、部門間調整、承認に要する期間をあらかじめ織り込んでおく ③ 手戻りや大きなスケジュール遅れに対する遅延対策を、あらかじめ計画しておく ④ 必ず実施すべきこと、省略できることに分けて、すべきことは期間を変更してでも漏らさずやるよう計画する
	RM. 2.3 費用見積り (6.2.3)	<ul style="list-style-type: none"> ■ 要件定義しているシステムの規模が、許容範囲内であるかどうか分からない ・ 今どこまで要求が積み上がっているのか不明 ・ 規模を示す数値が画面などの件数しかなく実態が分からない ・ 要件定義完了まで規模が不明ではスコープ調整の手間が大きい 	(1) 要求の規模は件数ではなく実現にかかるコストで表現する ① 要件定義の途中でもシステム構築費用が概算見積りできるように計画しておく ② 要求の規模を物件の数や難易度から容易に概算できるようにしておく ③ 有識者を投入して規模概算の精度を上げる ④ 概算結果に対して過度の精度を要求しない ⑤ 実績を蓄積して次回に備える
		<ul style="list-style-type: none"> ■ 要件定義にどれ位期間と労力がかかるのか定量的に予測できない 	(2) 要件定義にかかる工数をそれまでの実績に基づき見直す ① 過去実績から定量的見積りを行う ② 途中で実績に基づいて補正する

問題カテゴリ (記載箇所)	問題	解決のための勘どころ
RM. 2.4 体制・チームビルディング (6.2.4)	<ul style="list-style-type: none"> ■ プロジェクトに必要な体制が構築できなかつたり、チームワークが発揮できない ・ 業務知識を持つキーマンの時間が十分に確保できない ・ プロジェクトメンバや関係者が責任を取らない ・ 要件定義に必要な分野の IT 技術をもった人材が確保できない 	(1) プロジェクト目標の達成に向けて、体制を確立しチームビルディングを行う ① プロジェクトに必要な体制を構築する ② チームの発達段階を理解し、メンバに働きかける ③ チームとしての一体感を醸成する
RM. 2.5 品質計画 (6.2.5)	<ul style="list-style-type: none"> ■ 成果物を作成するときに、記載内容に個人差が出て品質にバラつきが出る 	(1) 成果物の品質は作成方法の指導と定期的な状況確認で保つ ① 成果物の作成指針を作成し、要件定義ドキュメント作成の関係者と合意する ② 成果物の作成指針の書き方について説明会を行う ③ 成果物の作成指針に沿ってドキュメントが作成されているかを確認するチェックリストを作成する ④ 執筆初期の段階で重点的なレビューを行う ⑤ 定期的にドキュメントのレビューを行い、ドキュメントの品質水準を確認する ⑥ レビューの記録を残し、問題点などが分析できるようにしておく
RM. 2.6 コミュニケーション計画 (6.2.6)	<ul style="list-style-type: none"> ■ ステークホルダとのコミュニケーションが不十分なため、要件定義に不備が発生する ・ しかるべき人への報告・共有がされず、手戻りが発生する ・ 合意形成ができなかつたり、時間がかかる 	(1) ステークホルダとのコミュニケーションを計画し、合意形成を図る ① コミュニケーション計画を立案する ② コミュニケーションの質を上げ、合意形成を図る
RM. 2.7 リスク認識 (6.2.7)	<ul style="list-style-type: none"> ■ リスクが顕在化し、プロジェクトが止まる 	(1) リスクの特定と対応計画の立案により、プロジェクトの失敗を防ぐ ① リスクを特定し、リスク登録簿に記載する ② リスクを分析する ③ リスク対応計画を立てて、スケジュールに反映する

2.2 要件定義の問題

問題カテゴリ (記載箇所)	問題	解決のための勘どころ
RM. 2.8 調達計画 と契約 (6.2.8)	<ul style="list-style-type: none"> ■ 要件定義の契約形態が実施内容と合っていないため、トラブルが発生する ・ 漠然とした要求にも関わらず、請負契約で要件定義をベンダに委託してしまう 	(1) 要件定義は準委任契約を基本とし、ユーザ企業主体で進める ① 要件定義の契約基本形は準委任契約とする
RM. 2.9 プロジェクト管理 計画の作成 (6.2.9)	<ul style="list-style-type: none"> ■ 要件定義が始まってから、後付けでプロジェクト管理のルールを整備しがちである 	(1) プロジェクト管理計画(管理の仕組み)を事前に作成し、経営層や上位マネジメント層を含む関係者と合意する ① 要件定義の開始時にプロジェクト管理計画を作成し、プロジェクトの関係者に周知する ② プロジェクトオーナーから、経営層、上位マネジメント層にもプロジェクト管理計画を周知してもらう
RM. 3 監視・コントロール (6.3)	RM. 3.1 要件・スコープ (6.3.1) <ul style="list-style-type: none"> ■ 得てして膨らんでしまう要求をコントロールできない ■ 現場からは、「あれもこれも必要で、ないと要求は実現できない」と言ってくる ■ 要件定義がある程度進んだ段階でも平気で大幅な修正を伴う変更を要求してくる <ul style="list-style-type: none"> ・ 要求の変更が歯止めが利かない ・ 変更対応に手間がかかる ・ 変更対応が漏れて後工程でリカバリが必要になる 	(1) 膨らむ要求を適正量に導く 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る ② 定量化した要件量の中で要求を調整する (2) 実現する要求は関係者が納得できる指標をもとに選択する ① 要求事項を客観的な指標で評価し、実現する要求を選択する (3) 決定済の事項を大きく変更することが必要になったら、プロジェクト全体での意思決定により採否を決定する ① 変更を当事者間で安易に受け入れさせない ② 変更するかどうかは、変更の影響と得られる効果から判断する ③ 変更する場合には、抜け・漏れが発生しないように確実に実施する

問題カテゴリ (記載箇所)	問題	解決のための勘どころ
RM. 3. 2 スケジュール (6. 3. 2)	■要件定義は進捗遅れの程度が分かりにくく、かつプロジェクト関係者が多いため遅延対策が難しい	(1) 細かいチェックポイントの設定と、事前に遅延対策を関係者に周知しておく ① 細かいチェックポイントを設けて、進捗遅れを早期発見、早期対処する ② キーマンや上位マネジメントの関与不足による遅れ対策を、あらかじめ考えておき、実施する ③ 遅延が大きくなった場合の対応策を、あらかじめ決めておきタイミングよく実施する ④ 計画時のレビューも代行できる仕組みを用意して実施する
RM. 3. 3 品質 (6. 3. 3)	■要件定義工程での、実用的かつ有効な品質測定指標がない	(1) 抜け漏れを抑えた要件定義成果物の作成とプロセスを監視する ① 要件の抜け漏れをなくすために作ると決めた成果物を作れているか、チェックする ② 要件の抜け漏れをなくすためにやると決めた要件定義プロセスをやれているか、チェックする
RM. 3. 4 コミュニケーション (6. 3. 4)	■部門間、個人間で意見が合わず、仕様の確定に時間がかかり、要件定義が収束しない	(1) 仕様がなかなか決まらないときは実態を確認して対処する ① 仕様を決められる適任者をプロジェクトに参画させる ② プロジェクトメンバー間の信頼関係を構築する ③ 業務担当者側は当事者意識を持ち、会議に参画する ④ うまくいかない場合は、エスカレーションし、問題の解決を図る
RM. 3. 5 リスク (6. 3. 5)	■リスクが監視されずに放置された場合、顕在化した時に影響を及ぼす場合がある	(1) リスクは対応完了までフォローする ① 定期的にリスクの状況を棚卸しする ② リスクの状況に変化があれば、取り扱いを判断する ③ 影響の大きいリスクは、ステアリングコミティで共有する

2.2 要件定義の問題

問題カテゴリ (記載箇所)	問題	解決のための勘どころ	
RM. 3.6 トレーサ ビリティ (6.3.6)	■ビジネス要求がシステム化要求 にどう反映されているのか分か らない	(1) ビジネス要求とシステム化要 求の関係は後で追跡できるよ うに台帳で管理する ① ビジネス要求、システム化要 求それぞれの要求事項を台帳 管理する ② システム化要求の漏れ・抜け を確認する ③ ビジネス要求が変更になっ たら、システム化要求も台帳に 基づいて漏れなく修正する	
RM. 3.7 見積作成 投資効果 分析 (6.3.7)	■定性的な投資効果に対する客観 的な評価を求められて対応に苦 慮する	(1) 投資側には保守運用費を入 れ、投資効果側は定性的な効 果の客観性がポイント ① 投資額の見積りには、コンテ インジェンシーや保守運用費 用を入れて、算定する ② 定性的な投資効果をできるだ け客観化する仕組みを考える	
RM. 4 終結 (6.4)	RM. 4.1 要求評価 (6.4.1)	■予定の期間で仕様が確定せず、 後続工程に影響を与える	(1) 決まらない要件は影響を評価 し、対応策を検討する ① 「決まらない要件」を識別す る ② 「決まらない要件」について システム化しない場合の影響 を評価する ③ 「決まらない要件」について の取り扱いを決定する
	■要件定義が完了したが、どうも 内容に自信が持てない ・まだ検討が十分ではないと漠 然と感じるが、どこに問題が あるのか分からない ・このまま進めても大した効果 が上りそうな気がしない ・もう時間切れで次に進まざる を得ない	(2) 内容に自信が持てないならそ こには何らかの問題がある、 思い切ってやめることも時 には必要 ① 関係者を集めて要件定義の結 果に問題がないことを確認す る ② いったん、施策の実施を棚上 げにする	

問題カテゴリ (記載箇所)	問題	解決のための勘どころ
RM. 4.2 完了判断 (6.4.2)	<ul style="list-style-type: none"> ■ 要件定義の各作業が完了したが、これですべて完了として良いかが判断できない 	(1) 実施すべき作業がすべて計画どおり実施されたことを確認する ① 要求(成果物)の妥当性を確認する ② 未決事項と対応計画を確認する ③ 完了判断のための材料を用意する ④ 経営者を交えた要件定義の完了と後工程の開始判断を実施する
	<ul style="list-style-type: none"> ■ 要件未決のまま要件定義を終わらせて次工程に進む判断ができない 	(2) 要件未決のまま次プロセスに進む必要があるなら相応の対策を立てておく ① 未決事項の先送り基準を決めておく ② 未決事項を台帳管理し、ステークホルダと合意する ③ 未決事項の内容が決定しない場合のリスクを洗い出し、監視する

2.2 要件定義の問題

第3章 要件定義の全体像

3.1 要件定義の位置付け

本ガイドは要件定義を進める上での課題と勘どころをガイドしている。本節では、その記載内容を読者により理解していただきやすくするために、本ガイドにおける要件定義の位置付け、目的について解説する。

要件定義は、システム開発の初期に実施される工程であり、前後に企画、基本設計（外部設計）という工程がある。一般的にはこれらの工程を総合して上流工程と表現する。

企画工程では、一般的に経営上のニーズ、課題を実現、解決するための新たな業務の全体像と実現に向けたシステム構想を立案する。この工程では、システム化構想を具現化するために、運用や効果等の実現性を考慮したシステム化計画、プロジェクト計画を具現化し、ステークホルダの合意を得る。

要件定義工程では、企画工程で立案したシステム化計画をインプットに、ステークホルダのニーズ、要望、課題を分析し、利用者や他のステークホルダが必要とするサービスを提供するシステムに対する要求を定義し、ステークホルダと合意して、要件とする。

基本設計（外部設計）では、要件定義で合意した要件をシステムの技術的要件やソフトウェアの構成要素の要件に詳細化する。

図 3.1 に企画工程、要件定義工程、基本設計（外部設計）工程の主な目的を示す。

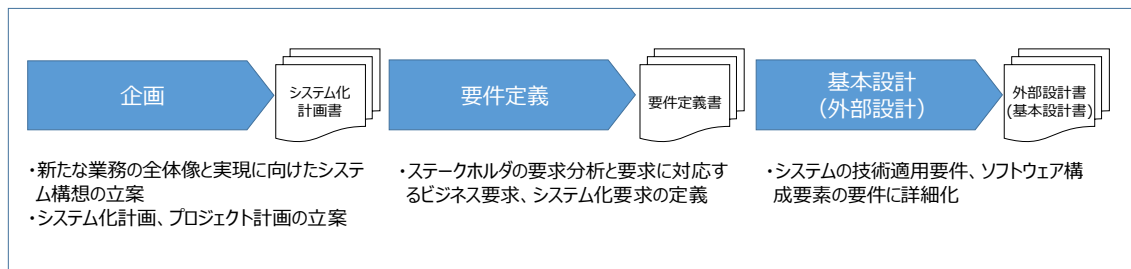


図 3.1 企画工程、要件定義工程、基本設計（外部設計）工程の主な目的

本ガイドでは、要件定義工程の終了時に、設計から移行・運用準備までのコストを概算レベルで見積もることを想定している。

3.2 要件定義プロセス

第2章で述べたとおり本ガイドでは、要件定義工程における問題を分類し、図3.2に示す要件定義問題カテゴリマップに対応付けている。

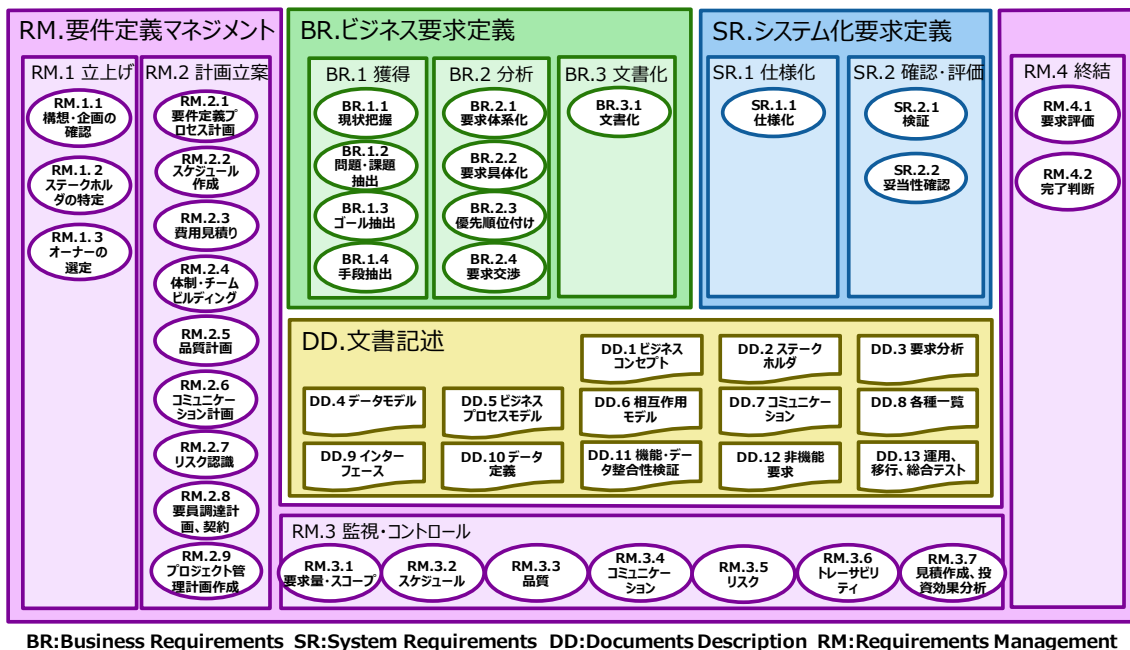


図 3.2 要件定義問題カテゴリマップ (再掲)

以下に、問題カテゴリマップの分類を説明する。

「BR. ビジネス要求定義」では、企画工程で整理された経営レベルの要求に基づき、利害関係者の要求事項として業務の観点から要求を獲得 (BR. 1 ビジネス要求の獲得)、ビジネス要求の項目間の関連の整理・優先度付け・具体化 (BR. 2 ビジネス要求の分析)、それらの文書化 (BR. 3 ビジネス要求の文書化) のステップに分類して、それらを実施する際に発生する問題と解決の勘どころを提示している。

「SR. システム化要求定義」では、「BR. ビジネス要求定義」で定義した要求を機能要件・非機能要件の観点で仕様化 (SR. 1 システム化要求の仕様化)、その仕様の検証や妥当性確認 (SR. 2 システム化要求の確認・評価) のステップに分類して、それらを実施する際に発生する問題と解決の勘どころを提示している。

3.2 要件定義プロセス

「RM. 要求定義マネジメント」では、要件定義開始にあたっての確認・準備 (RM. 1 立上げ)、要件定義プロジェクトの計画 (RM. 2 計画立案)、「BR. ビジネス要求定義」や「SR. システム化要求定義」の活動中における監視・コントロール (RM. 3 監視・コントロール)、要件定義工程の実行結果の評価と完了判断を行う終結作業 (RM. 4 終結)のステップに分類して、それらを実施する際に発生する問題と解決の勘どころを提示している。

「DD. 文書記述」では、ドキュメントの種類に分類し、各ドキュメントの記載項目とサンプルドキュメント、作成時の留意事項などを提示している。問題と解決の勘どころの説明に登場したドキュメントも含めてここに集約して提示している。

問題カテゴリマップは、あくまでも問題を分類したマップであり、マップに示す作業分類以外にも必要に応じて作業を追加する必要があることを留意いただきたい。問題カテゴリマップの各項目で想定する主な実施項目と対応する記載箇所を表 3.1 で示す。

表 3.1 カテゴリマップの各項目における主な実施事項と記載箇所

大分類 (記載箇所)	中分類 (記載箇所)	小分類 (記載箇所)	主な実施事項
BR. ビジネス要求定義 (第4章)	BR. 1 ビジネス要求の獲得 (4.1)	BR. 1.1 現状の把握 (4.1.1)	正確な分析をする上で、ビジネスアナリストが現状のシステムを理解する
		BR. 1.2 問題・課題の抽出 (4.1.2)	すべてのステークホルダが各自の現状の課題を認識するとともに、他の人や組織が抱える課題と、システム全体の課題を抽出する
		BR. 1.3 ゴールの抽出 (4.1.3)	ステークホルダの現状の課題に基づいて、人、または組織が達成すべきゴール、および維持すべきゴールをステークホルダとビジネスアナリストとが共有している状態にする
		BR. 1.4 手段の抽出 (4.1.4)	ゴールを達成する手段と合理的な根拠を明らかにする
	BR. 2 ビジネス要求の分析 (4.2)	BR. 2.1 要求の体系化 (4.2.1)	抽出した要求(問題・課題・ゴール・手段)を分類、整理、構造化し、要求が重複、矛盾、抜け漏れがないことを明らかにする

3.2 要件定義プロセス

大分類 (記載箇所)	中分類 (記載箇所)	小分類 (記載箇所)	主な実施事項
		BR. 2.2 要求の具体化 (4. 2. 2)	整理した要求を実現する手段を明確にする
		BR. 2.3 優先順位付け (4. 2. 3)	限られたリソース（期間、コスト、人など）内で競合する要求をステークホルダの期待に沿う形で適切に優先順位を決定する
		BR. 2.4 要求の交渉 (4. 2. 4)	優先順位をつけた要求の中で、実現すべき妥当な要求をステークホルダと合意する
	BR. 3 ビジネス要求の文書化 (4. 3)	BR. 3.1 文書化 (4. 3. 1)	すべての要求を正確にかつ漏れなく記述することで、経営者からエンドユーザまでのすべてのステークホルダに理解できる文書を作成する
SR. システム化要求定義 (第5章)	SR. 1 仕様化 (5. 1)	SR. 1.1 仕様化 (5. 1. 1)	情報システムに関与するステークホルダが、実現手段の外観を理解し、導入する情報システムや製品が技術面、運用面で実現可能か、利用面で問題ないかの確認できる文書を作成する
	SR. 2 確認・評価 (5. 2)	SR. 2.1 検証 (5. 2. 1)	要件が構造的、意味的な正しさを確認することで要求の内容を明確にし、要求の妥当性確認ができるようにする
		SR. 2.2 妥当性確認 (5. 2. 2)	ビジネス要求に照らし合わせて、ステークホルダが期待している初期の要求を満たしているかを確認する
RRM. 要求マネジメント (第6章)	RM. 1 立上げ (6. 1)	RM. 1.1 構想・企画の確認 (6. 1. 1)	適切なプロジェクト計画を立案するためのプロジェクトコンテキストを理解する
		RM. 1.2 ステークホルダの特定 (6. 1. 2)	的確な要求分析、ならびに適切なプロジェクト遂行に向け、それらに影響を及ぼす可能性があるステークホルダを特定する
		RM. 1.3 オーナーの選定 (6. 1. 3)	プロジェクトの成果に対する結果責任の所在を明らかにする

3.2 要件定義プロセス

大分類 (記載箇所)	中分類 (記載箇所)	小分類 (記載箇所)	主な実施事項
	RM. 2 計画立案 (6. 2)	RM. 2. 1 要件定義プロセス 計画 (6. 2. 1)	プロジェクトを遂行するにあたり、プロセスの内容と管理活動を作成し、文書化し、ステークホルダ間で共通認識を合意する
		RM. 2. 2 スケジュール作成 (6. 2. 2)	要件定義のスコープに基づき、実現可能なスケジュールを作成する。
		RM. 2. 3 費用見積り (6. 2. 3)	要件定義を進める上でのコストを算出する
		RM. 2. 4 体制・チームビルディング (6. 2. 4)	プロジェクト目標を達成するために、チームを形成、動機付けし、高い成果を上げられるチームワークを育む環境を確立する
		RM. 2. 5 品質計画 (6. 2. 5)	プロジェクト、およびその成果物の品質要求事項、品質標準を特定し、それらを遵守するための方法を文書化、ステークホルダと合意する
		RM. 2. 6 コミュニケーション計画 (6. 2. 6)	定義したステークホルダの特性を分析し、最良のコミュニケーションアプローチを定義することで、ステークホルダとの効果的な業務上の関係を築いて維持する
		RM. 2. 7 リスク認識 (6. 2. 7)	プロジェクト全体のリスクをステークホルダと共通認識する
		RM. 2. 8 調達計画と契約 (6. 2. 8)	ベンダ等との契約など、要件定義を進める上での必要リソースの調達をする
		RM. 2. 9 プロジェクト管理 計画の作成 (6. 2. 9)	進捗、課題、変更、リスクなどの管理方法を検討し、文書化とメンバーへの周知を行う
RM. 3 監視・コントロール (6. 3)		RM. 3. 1 要求量・スコープ (6. 3. 1)	スコープの状況を監視し、適切なスコープコントロールを実施する
		RM. 3. 2 スケジュール (6. 3. 2)	進捗状況を監視し、遅延の予防やリカバリなどのコントロールを実施する
		RM. 3. 3 品質 (6. 3. 3)	プロジェクトパフォーマンスをモニタリングし、プロジェクトの成果物が完全かつ正確で、ステークホルダの期待を満たしていることを保証する

3.2 要件定義プロセス

大分類 (記載箇所)	中分類 (記載箇所)	小分類 (記載箇所)	主な実施事項
		RM. 3.4 コミュニケーション (6.3.4)	プロジェクトとステークホルダとのコミュニケーションが望ましい効果を発揮しているかを明確にする
		RM. 3.5 リスク (6.3.5)	特定したリスクの追跡、ならびに新しいリスクを特定・分析し、適切な対応が実施されていることを維持する
		RM. 3.6 トレーサビリティ (6.3.6)	個々の要求項目の発生から実装までを記録することで要求分析、システム化要求定義にわたる一貫性を確保するため。また、要求変更発生時の影響分析をする
		RM. 3.7 見積作成・投資効果分析 (6.3.7)	「ビジネス要求 (BR)」「システム化要求 (SR)」の内容に基づき投資額の見積りと効果分析をする
	RM. 4 終結 (6.4)	RM. 4.1 要求評価 (6.4.1)	文書化されたビジネス要求、ならびに文書化されたシステム化要求の品質評価結果を確認し、適切に要件定義が遂行されたことを確認するため。また、未定義の要件を確認し、対応策についてステークホルダと合意する
		RM. 4.2 完了判断 (6.4.2)	要求評価結果、ならびに要求マネジメント結果をもとに、要件定義工程の完了をオーナーと合意する

3.2 要件定義プロセス

コラム 4 ビジネス要求定義、システム化要求定義の主要タスク例

要件定義の手順はプロジェクトの内容によって大きく変わるので、こうでなければならないというものはないが、ここでは基幹業務システムのスクラッチからの再構築プロジェクトの場合を参考までに示す。(なお、この例は4カ月で要件定義を行う例を示している。)

① 1カ月目

最初の1カ月はビジネス戦略からのビジネス要求や現行業務の改善要求、ならびに現行システムからの改善要求などを集めて、その要求の要因や原因を調べたりすることに主眼を置いている。また、新システム向けに採用する最新の技術や製品の採用可否を検討したりすることを行っている。

② 2カ月目～3カ月目前半

2カ月目に入って、要求から要件への絞り込みを行って、To-BeのモデルやTo-Beの業務フロー、To-Beの業務処理定義を主にユーザ主体に作成している。それと同時に、システム側では、新システムの機能要件や非機能要件を検討して、新技術や新製品の技術検証を行っている。

③ 3カ月目後半～4カ月目

3カ月目半ばに入ってから、これまでに検討されたTo-Beの業務フローやTo-Be業務処理定義書、ならびに新システムの機能要件、非機能要件の検討結果をもとにした新システムの機能一覧や新システム機能定義書の作成を実施している。また、データモデルや稼働環境の実現可能性の検証を行い、新システムの実現可能性を担保するようにしている。他には運用要件や、データなどの移行に関する要件や課題を整理し、また設計の標準化を徹底するように準備している。

その上で、開発・運用費用を概算で見積もり、投資効果を分析し、現新比較表をもとに、経営側への説明の準備を行っている。

このWBSのすべての作業を実施しなければならないわけでもなく、実施するとしても概要レベルの内容にとどめても問題ない場合もあるので、表を参考にテーラリングして、自プロジェクトのWBSを作成されたい。

要件定義(ビジネス要件定義とシステム要件定義)の主要なWBS例 (プロジェクト管理関係を除く)			
期間	1か月	1.5か月	1.5か月
ビジネス要件 関連WBS	ビジネス戦略とビジネス目標、ゴールの確認	プロジェクトのKGI、KPIの作成、確認	投資効果分析資料の作成
	ビジネス戦略からのビジネス要求の洗い出し	ビジネス戦略からのビジネス要求に対する解決策の検討	ビジネスモデルや業務に関する現新比較表(Before/After)の作成
	As-Is業務フローの作成	ビジネス要求、業務改善要求の対応対象の絞り込み	
	As-Is業務の問題分析と業務改善要求の洗い出し	To-Be業務機能関連図の作成	
	To-Be業務関係成果物の標準の作成(粒度指針も含む)	To-Be業務フロー(To-Beシステム化業務フロー)の作成	
	業務用語集の作成	To-Be業務処理定義書の作成	例外業務のシステム化対応範囲の確定 業務移行要件の検討
システム要件 関連WBS	現行システムの機能の詳細調査(必要に応じて)	現行システムの課題解決策の作成と対応対象の絞り込み	新システム全体図の作成
	現行システムの課題の洗い出しと要因分析(必要に応じて)	新システムの機能要件の検討	新システム機能一覧表(画面、帳表、IF、バッチ処理)の作成
		新システム非機能要件の検討	新システム機能定義(画面、帳票、IF、バッチ処理)の作成
		新システムの運用要件の検討	新システム非機能要件の作成
		新システムの設計標準(UI画面遷移など)、成果物標準の作成(粒度指針を含む)	新システムの運用要件の作成
			総合テスト方針の検討
			システム機能、非機能に関する現新比較表(Before/After)の作成
	現行データベースの調査、課題の抽出(データボリュームも含む)	新システム概念データモデル作成	新システム概念データベースの作成
	現行主要マスタの調査、課題の抽出	新マスターの運用要件の検討	新システムの主要マスタの定義作成(運用要件も含む)
	現行コード体系の調査、課題の抽出	新システムのコード体系の検討	新システムのコード体系の作成
			主要データ項目定義の作成
			主要エンティティのCRUD図作成
			データ移行要件の検討(データ、マスタ)
	新システムの稼働環境の調査	新技術や新プロダクトの機能調査、技術検証の実施(POC)	新システムの稼働環境定義の作成
	新システムの稼働環境条件の検討	アプリケーション処理方式や共通フレームワークなどの標準設計	
	将来の拡張性の検討	各種設計標準の作成	
		システム環境移行要件の検討	

図 3.3 要件定義における WBS の例

3.3 要件定義ドキュメント

本ガイドでは、「BR. ビジネス要求定義」、「SR. システム化要求定義」を定義し、後続する設計、実装、テスト工程につなぐ主要な要件定義ドキュメントとして、表 3.2 および図 3.4 に示すドキュメントの目的や作成における勘どころ、サンプルを記載している。

表 3.2 主要な要件定義ドキュメントと記載箇所

大分類	中分類	ドキュメント名	記載箇所
DD. 文書記述	DD.1 ビジネスコンセプト	ビジネスコンセプト確認ドキュメント (SWOT、バランススコアカード、ビジネスモデルキャンパス)	7.1.1
	DD.2 ステークホルダ	ステークホルダ関連図	7.2.1
		ステークホルダー一覧	7.2.2
		リッチピクチャ	7.2.3
	DD.3 要求分析	問題・ニーズ・課題一覧	7.3.1
		問題原因分析図	7.3.2
		要求構造図	7.3.3
		要求一覧	7.3.4
	DD.4 データモデル	管理対象分類図	7.4.1
		概念データモデル (ER 図)	7.4.2
	DD.5 ビジネスプロセスモデル	ビジネスプロセス関連図	7.5.1
		業務機能構成表	7.5.2
		ビジネスプロセスフロー (業務フロー)	7.5.3
		ビジネスプロセスフロー (システム化業務フロー)	7.5.4
		業務処理定義書	7.5.5
	DD.6 相互作用モデル	状態遷移図	7.6.1
	DD.7 コミュニケーション	業務用語定義	7.7.1
		ビフォーアフター図	7.7.2

3.3 要件定義ドキュメント

大分類	中分類	ドキュメント名	記載箇所
	DD. 8 各種一覧	システム化業務一覧	7.8.1
		画面一覧	7.8.2
		帳票一覧	7.8.3
		外部インターフェース一覧	7.8.4
		エンティティ一覧	7.8.5
	DD. 9 インターフェース	システム化要求仕様	7.9.1
		UI 標準	7.9.2
		画面遷移図	7.9.3
		画面／帳票レイアウト	7.9.4
	DD. 10 データ定義	エンティティ定義書／データ項目定義書	7.10.1
		ドメイン定義書	7.10.2
		コード体系定義書／コード内容定義書	7.10.3
	DD. 11 機能・データ整合性検証	CRUD 図	7.11.1
	DD. 12 非機能要求	非機能要件書	7.12.1
	DD. 13 運用、移行、総合テスト	運用要件書	7.13.1
全体移行計画書		7.13.2	
総合テスト計画書		7.13.3	

3.3 要件定義ドキュメント

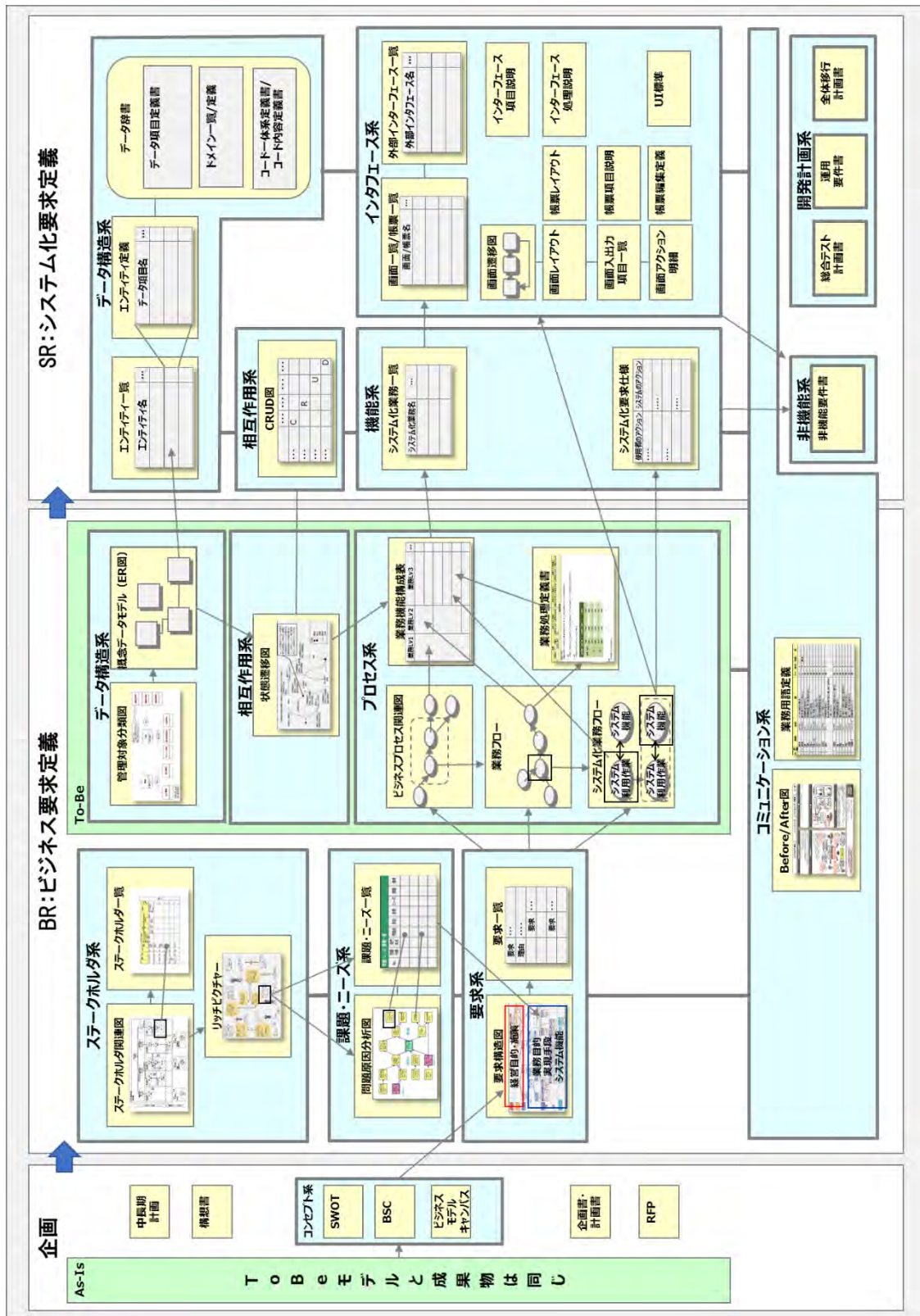


図 3.4 成果物フロー

3.4 用語定義

要件定義に使用される用語は人により解釈が異なることが多い。例えば、「要求」にしてもビジネス戦略上求められる要求を指して使用されているのか、あるいはそれを実現するシステムが具備すべき機能に対する要求を指すのかといった意味解釈や粒度が人それぞれに異なる。用語の定義や粒度をプロジェクトのステークホルダ同士で合わせた上でプロジェクトを開始しなければ、認識齟齬による思わぬ遅れや手戻りを引き起こしかねない。

一方で、要件定義には、それに関する国際標準や日本工業規格が存在する。要件定義は、「ISO/IEC/IEEE 29148 Systems and software engineering - Life cycle processes - Requirements engineering」（以下、ISO/IEC/IEEE 29148 と表記する）[17]として国際標準化されており、そこには、要件定義を含めた開発プロセスの上流工程における用語や概念、プロセスの定義が記載されている。この ISO/IEC/IEEE 29148 をもとに、「JIS X 0166 システム及びソフトウェア技術-ライフサイクルプロセス」（以下、JIS X 0166 と表記する）[18]が日本工業規格化されている。

そこで、本節では、解釈が論議される以下の3項目について、上記の標準、規格を参考に、本ガイドとしての定義を示す。

- 要求の分類
- 要求と要件の違い
- 検証と妥当性確認

あくまで、本ガイドの定義であり、読者のプロジェクトにおける認識合わせのベースとして活用していただきたい。

【要求の分類】

JIS X 0166 では、要求事項は「ニーズとそれに付随する制約・条件とを変換した又は表現する文」と定義されている。要求事項にはいくつかのレベルがあり、「利害関係者要求事項」、「システム要求事項」、「ソフトウェア要求事項」に分類される。それぞれの定義を表 3.3 に示す。

3.4 用語定義

表 3.3 要求の分類

用語	定義
利害関係者要求事項 (Stakeholder Requirement)	利用者及び他の利害関係者によって必要とされるサービスを提供するためのシステムに対する要求事項。利害関係者要求事項の典型的なタイプとして、組織要求事項、ビジネス要求事項、及び利用者要求事項がある。
システム要求事項 (System Requirement)	システム及びその運用環境並びに外部インターフェースに対する要求事項（機能・性能・設計制約・属性）の構造化された集合。システム要求事項はシステムの構成要素ごとにシステム要素要求事項として再帰的に詳細化される。
ソフトウェア要求事項 (Software Requirement)	ソフトウェア及びその外部インターフェースへの要求事項（機能・性能・設計制約・属性）の構造化された集合。

利害関係者要求事項は、利害関係者要求仕様(Stakeholder Requirement Specification: StRS)書として文書化される。利害関係者要求仕様には、なぜシステムを開発または変更するのかという組織レベルの動機とシステムの利用者、運用者、または保守者のニーズを含めたシステムの利用プロセス、利用方針、利用ルールが記述される。

システム要求事項は、利害関係者要求仕様をもとにシステムに対する技術的仕様および想定される人間システム相互作用に対する使用性を識別し、システム要求仕様(System Requirement Specification: SyRS)書として文書化される。また、ソフトウェア要求事項は、システム要求仕様をもとに、機能を実行するソフトウェア製品、プログラム、またはプログラム集合に対する仕様を識別し、ソフトウェア要求仕様(Software Requirement Specification: SRS)書として文書化される。

これらの関係性を図 3.5 に示す。なお、本ガイドでは、利害関係者要求事項を「ビジネス要求」、システム要求仕様を「システム化要求定義」と定義している。

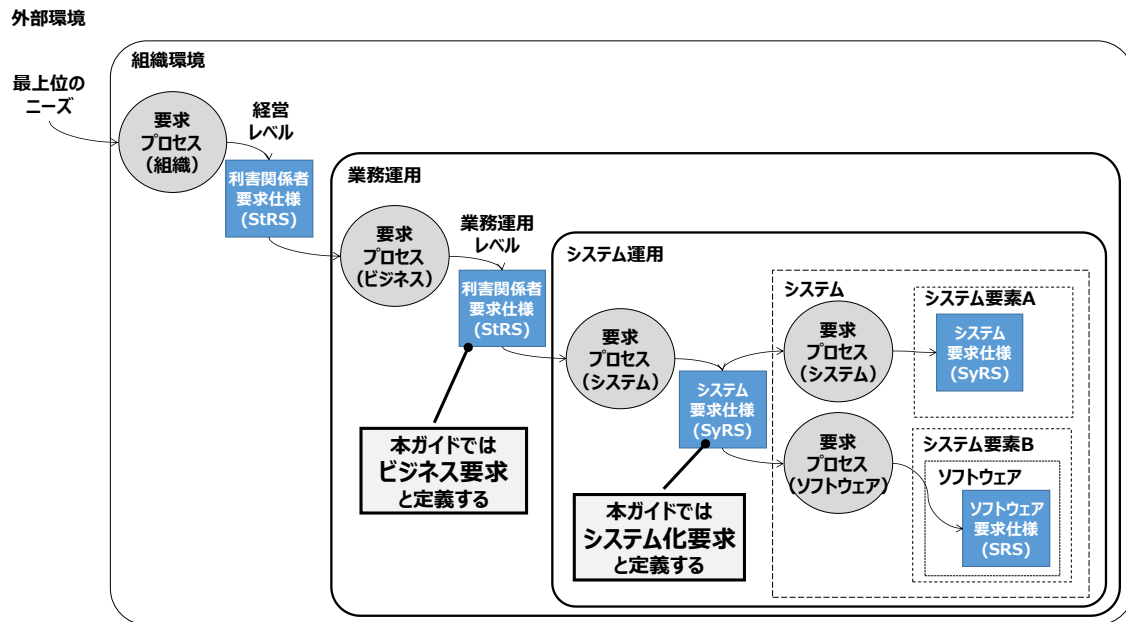


図 3.5 要求分類間の関係性

(出典) JIS X 0166 [18]の図を基に加筆

経営レベルの利害関係者要求仕様、ビジネス要求（業務レベルの利害関係者要求仕様）、システム化要求（システム要求仕様）の例として、部品メーカーの在庫管理におけるそれぞれの具体的なイメージを以下に示す。

- **経営レベルの利害関係者要求仕様**
 - ・ グローバルな視点で製品在庫を最適化し、
 - ・ 過剰在庫の在庫コストを削減
 - ・ 在庫切れをなくし、受注機会損失を削減
 - ・ 競合他社の在庫切れに互換製品を提案し、
 - ・ 売上を伸ばす
- **ビジネス要求**
 - ・ グローバルに在庫の状態がリアルタイムに把握できる
 - ・ 製造中の製品は予定在庫日、予定在庫数が把握できる。
 - ・ 需要予測が実用的な精度でできる
 - ・ 在庫切れが予測された場合、余裕のある倉庫の在庫を移動する。
 - ・ 前提：在庫切れが発生する製品は全体の1割以下で、事前に想定できる。
- **システム化要求**
 - ・ 世界地図の上で倉庫を表示し、指定された品種の在庫がビジュアルに表示できる。
 - ・ 各倉庫の入庫・出庫情報をリアルタイムに新システムに伝送する。
 - ・ 在庫切れを監視する製品を事前に登録できる
 - ・ 上記製品の3か月後先までの需要予測を実施し、自動設定された安全在庫量を下回った場合、メールで通知する。
 - ・ 安全在庫量を下回る製品の一覧を表示し、製造指示が出せる。

図 3.6 ビジネス要求、システム化要求のイメージ

3.4 用語定義

【要求と要件の違い】

前述の JIS X 0166 において、要求事項(requirement)の注記に「requirement は、通常要求又は要件と訳される」と記載されているとおり、要求と要件の標準的な定義は存在しない。本ガイドでは、「要求」は JIS X 0166 における要求事項と同義としている。一方で、要求を文書化、仕様化し、ステークホルダと合意したものを「要件」と定義する。要件は次工程の入力となるため、第三者へ伝達可能（例えば、業務部門から IT 部門へ、IT 部門から開発を担うベンダへ）な形に形式化されている必要がある。

【検証と妥当性確認】

記述された文書の不具合をレビューを通して摘出することが多い。一概にレビューと言っても、目的が異なるとレビュー実施者やレビュー観点が異なる。そこで、本ガイドでは、JIS X 0166 を参考に検証と妥当性確認の定義を、表 3.4 に示すように分けている。検証と妥当性確認はビジネス要求、ならびにシステム化要求のいずれに対しても行われる。

表 3.4 検証と妥当性確認

用語	定義
検証	記述された要件が要求を抜け漏れなく、誤りなく満たしているかを確認すること。
妥当性確認	定義された要求自体が、達成したい目的を実現できるものであるか、手段として妥当であるかを確認すること。

第4章 ビジネス要求定義(BR)における 問題と解決の勘どころ

「BR. ビジネス要求定義」では、大きく 3 つのサブプロセスに分けて作業を行う。まず、「BR1. ビジネス要求の獲得」において、現状業務、システムの把握をした上で、それらの問題、ならびに解決すべき課題を抽出し、目指すべきゴールとその手段を抽出する。続いて、「BR2. ビジネス要求の分析」において、目指すべきゴールを実現するために必要な要求の体系化や実現手段の検討をする。また、体系化した要求、ならびに実現手段をもとに要求の優先順位付けを行い、対象とする要求の範囲、ならびに優先順位をステークホルダで合意する。最後に、「BR3. ビジネス要求の文書化」において、合意した範囲の要求についてその体系と内容、実現手段を文書化する。

「BR. ビジネス要求定義」における課題カテゴリマップを図 4.1 に示す。

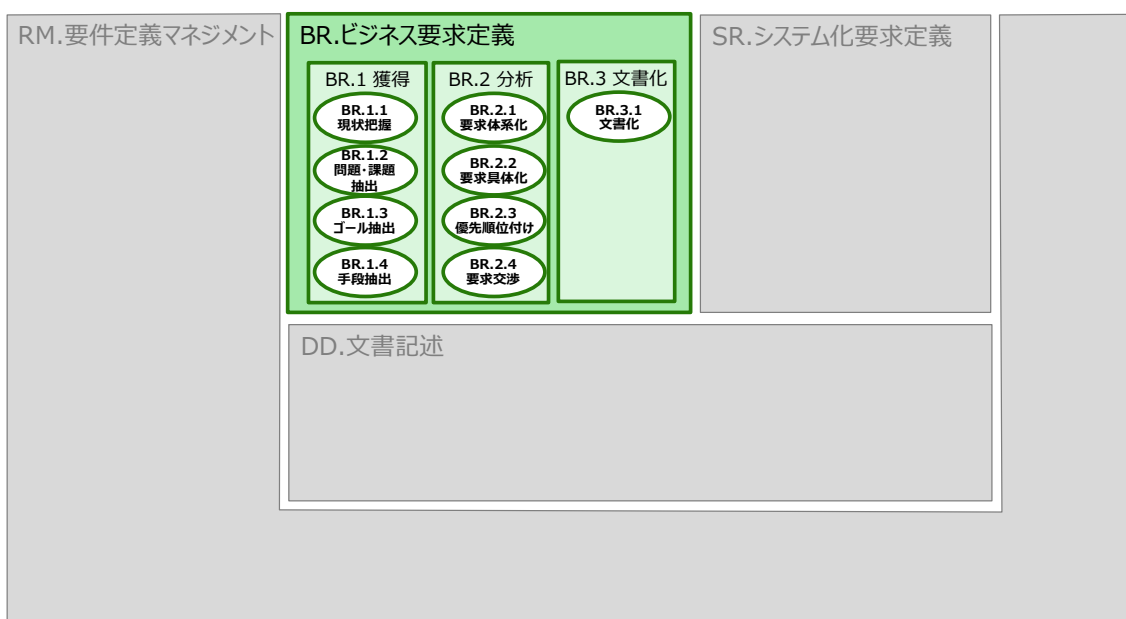


図 4.1 「BR. ビジネス要求定義」の課題カテゴリマップ

4.1 ビジネス要求の獲得(BR.1)

4.1.1 現状の把握(BR.1.1)

昨今のシステム開発には、稼動しているシステムをベースにしたリホスト、リライト、リプレースや、既存システムを捨てて一から作り直すスクラップアンドビルドなど、いわゆる再構築が多く見られる。たとえ選択した手法がスクラップアンドビルドであったとしても、システム再構築の際に既存システムで実施している業務の理解が重要であることに変わりはない。しかし、業務をシステム化してから時間が経過するとともに、システムで実現している業務がなぜそのような形態になっているのかを理解する人が業務部門にも少なくなってくる。一方で、システム化されているからといって、システム部門には業務の精通者が残っているというわけでもない。そのような背景から、システムや業務の理解が浅いにもかかわらず「現行踏襲」という本来受け入れてはいけない要求を受け入れてしまい、業務仕様や機能仕様が不明瞭なままプロジェクトを進め、あとで問題になるようなことも起きている。

現状を理解するために必要な情報源、例えばドキュメントや人などがそろっていない、存在しないことも多い。そのようなときには、今現在、業務で稼働しているシステムを道標にするしか方法がないことがある。何故そうなっているかは分からなくても、どういう動作をしているかは、現行システムにリバースツールを用いて可視化することもできる。また、人が業務を観察して業務フローやモデルを作成する方法もある。いずれにせよ、要件定義の入り口として複数の人が業務を共通理解し、他の人に伝達していけるようにするためには、現行業務や現行システムを可視化し、把握することから始める必要がある。

(1) 業務・システムの状況をさまざまな方法を活用して把握する

【解決したい問題】

- 現行業務やシステムを理解するのが困難で、正確な現状把握ができていない
 - ・ 既存ドキュメントが陳腐化しておりそのまま利用できない
 - ・ 業務を理解している人材が減っている
 - ・ 対象業務やシステムが複雑、広範囲で理解し難い

システム開発ときに作成した業務フローや設計書などが、度重なる業務改善やシステム保守時に適正に修正されず、現行業務やシステムと乖離してしまっていることがよく

4.1 ビジネス要求の獲得(BR.1)

ある。この場合、それらのドキュメントを現状把握のインプットにするためには、まずドキュメントと現状の差を埋める必要がある。

また、業務やシステムを人から教わろうとすることも困難になってきている。業務がシステムに埋没し、業務部門に業務の精通者が減っているという企業もある。開発当時のことを記憶しているシステム精通者がいなくなり、すでにあるシステムの保守だけを担当した人しか残っていない企業もある。

さらに再構築対象の業務やシステムが保守の繰り返しで巨大で複雑になってしまったこともあり、現行の仕様を紐解くのに困難が伴うこともある。

勘どころ① 現行システムから可視化する

システムは業務の写像である。現行業務を理解する手がかりがない場合、もしくは現存するドキュメントが全く最新化されていない場合は、実際に動いているシステムからどういう動作をしているかを可視化することが現行分析の第一歩となる。以下に可視化できる情報の例を挙げる。

- データモデル（データベースから）
- モジュール構成・クラス構成（プログラムコードから）
- 処理ロジック（プログラムコードから）
- ジョブスケジュール（ジョブ管理ツールの設定から）
- OS、ミドルウェア、ソフトウェア設定（設定ファイルから）
- 接続先システム情報（データ連携ツールの設定から）

これらには、リバースツールなどが用意されているものもあるので、利用すると良い。

勘どころ② 全体像を可視化する

可視化の目的は、現行業務、システムの理解である。したがって、精通者でない担当者でも理解できるよう、概略化し全体像を文書化する必要がある。あるべき業務像を検討するために、単に寄せ集めるのではなく、広い視野を持って俯瞰的に整理するよう留意していただきたい。また、再構築対象が複雑で広範囲な場合には、全体像を可視化することが有効である。以下に、全体像の概略の可視化例を示す。

- システム間関連図
接続先システム情報、ジョブスケジュール、データモデルなどをもとに、どのシステムからどのシステムへ、いつ、なにが、どのように連携しているかを可視化する

- システム構成図
OS、ミドルウェア、ソフトウェア設定などからシステムの構成、サーバー間の関連性を可視化する
- 概念データモデル
どのような情報が保持され、どう関連しているかを可視化する
- 画面遷移図
処理ロジックなどから画面遷移を可視化する
- 業務スケジュール
ジョブスケジュールなどから業務スケジュールを可視化する

勘どころ③ 現行業務を再学習する

現行システムの復元だけでは、システムでは対応していない領域は可視化できない。その中には、現行業務担当者の頭の中にだけあり、形式化されていない情報もある。したがって、稼働中のシステム、その運用ドキュメントや現行業務担当者本人からの再学習（リラーン）を通じて、現行業務、システムの可視化結果の補完を行うと良い。再学習とは、理解するために現行システムを調査したり、操作したり、利用されている現場を観察したりして、システムや業務の理解を促進する手法である。

現行業務を再学習することで、要件定義メンバが単に業務の流れ、システムの使い方を知るだけでなく、潜在する問題、ニーズ、課題を一人称で、あたかも自分事のように考えられるようになる。そのような状況で導かれた課題やその課題を解決する要求は、現行業務の担当者に納得感のあるものとして受け入れられやすくなる。

勘どころ④ 埋もれている既存資料を発掘する

社内には多数の資料があるはずである。まずは、対象の業務、システムや周辺業務、システムに関する資料を探すことから始めると良い。以下にドキュメント例を挙げる。

- 現行業務、システム稼働時の説明資料
- 業務分掌規程など業務に関わる規程、基準
- 新人研修用ドキュメント
- 運用マニュアル
- SLA
- 契約書

勘どころ⑤ 実態と照らし合わせて確認する

4.1 ビジネス要求の獲得(BR.1)

既存ドキュメントを使って、関係者全員で理解を深めるための現行の業務シナリオを作成する。ただし、業務シナリオを作成して終了ではない。業務シナリオの作成による可視化はあくまで現行理解への第一歩である。真の理解を得るためには、作成した業務シナリオを、実際の業務担当者に弟子入りする位の気持ちで、実態と照らし合わせて確認すると良い。実態に照らし合わせた確認方法の例を以下に挙げる。

- 現行システム操作

まずは、現行システムを実際に操作し、作成したシナリオを確認する。試験環境や擬似本番環境があればそれを使うほうが望ましい。操作をしている中でさまざまなケースでどのようにしているのか疑問に思う点があるはずなので、記録し業務担当者に確認すると良い。

- 業務担当者へのインタビュー

作成した業務シナリオを使って、業務担当者にインタビューして確認してもらえると良い。あわせて、前述のシステム操作で出てきた疑問点を確認する。インタビューを通して、さまざまなシナリオが抽出できるし、現行業務、システムへの不満、ニーズなども抽出できる。

- 実作業の観察（エスノグラフィー）

次に、業務担当者の作業環境に同行、陪席し、実作業を観察しながら作成した業務シナリオを確認すると良い。業務担当者は日々の作業で当たり前を感じている部分が第三者の目からすると問題に感じられたり、こうしたほうがよくなるのではないかといったアイデアを発想できたりする。

- 業務作業のシャドーイング

さらに、業務担当者と一緒に、もしくは業務担当者に代わって業務を遂行する方法もある。実際に経験することにより、現行業務を一人称で捉えられるようになるだけでなく、「なぜこのような作業が必要なのか？」といった、ドキュメントでは到底表現できない背景や文脈を理解する足がかりを掴むことができる。

事例 1 現行業務に関わる業務分掌規程などの有効利用 (株式会社 ジャステック)

【取り組みの背景】

システム開発の対象となる現行業務や現行システムを把握するために、弊社では各種規程を有効に利用している。弊社では、大分類「会社基本規程、組織規程、業務分掌規程」の中に、現在、186の小分類規程が存在する。例えば、営業部門の業務は74業務に分類され、営業部門が主管する規程には7つの規程が存在する。さらに、営業部門と関連する他部門の規程は97（計104）の規程に及んでいる。

【課題】

現行業務の現状把握は、ヒアリング以外に現行業務処理などを記載したドキュメントが利用される。しかし、一般的に現行業務処理などを示すドキュメントが存在しないか、存在しても現行と一致しない場合があり、現行業務の現状把握が非効率になる場合や開発に入り現状把握ミスが露呈されている。

【解決策】

今回、紹介する営業部門での販売業務管理システムの再構築開発において、営業部門の業務分掌規程から2つの規程（「営業業務決裁申請運用要領」と「販売管理プロセス内部統制基準」）に関して有効性を得た事例を紹介する。

なお、紹介する2つの規程は「規程管理規程」に基づき、営業部門が規程を改訂するごとに改訂履歴を記載するとともに、最終的に取締役会決裁を通し運用されている。そのため、常に最新の規程に基づく業務運用が執行されており、構成管理上、保証された現状把握が可能になっている。

(1) 営業業務決裁申請運用要領の有効利用

「営業業務分掌規程」に基づく営業業務決裁申請に係る手続きを定め、もって業務の円滑な処理を図る目的で作成し運用している。

営業部門が主管するすべての決裁申請の種別、決裁申請手続き、決裁手続き、決裁後の手続きおよび附則などが明記されており、台帳管理含め35種類の決裁関係帳票も定められている。

4.1 ビジネス要求の獲得(BR.1)

(2) 販売管理プロセス内部統制基準の有効利用

金融商品取引法に定められた「財務報告に係る内部統制の評価および監査に関する実施基準」に対応し作成され、内部監査室・コンプライアンス室などの提言および組織規程・業務分掌規程・職務権限規程の改訂にともない改訂されるが、都度、「規程管理規程」に基づき更新し運用している。

販売管理プロセスの内部統制基本方針、改訂手続の契機などとともに、“業務フロー”、“RCM（リスク・コントロール・マトリクス）”から構成しており、概略な業務プロセスの流れは、「取引開始」、「受注～納入・検収」、「請求書発行～売上計上」、「その他‘売上連携’など」となっている。

【効果】

(1) 決裁業務での「現状把握」に関わる有効性

- ① 決裁業務プロセスのバリエーションがすべて網羅されており、決裁業務に関わる全体を俯瞰的に捉えることができ、要件定義工程での特定の生産物というより、全体に及ぶ生産物の作成に有効である。
- ② 35種類の紙帳票のすべてを電子化するとき、最新状態での各項目の把握など、効率的かつ間違いなく捉えるのに有効である。

(2) 販売管理業務プロセスでの「現状把握」に関わる有効性

- ① 当該規程は、業務部門である営業本部（営業管理課）が主管し、常に最新の現行業務フローとRCMを定めている。このため、再構築開発において新たな業務フローとシステム化業務フローの作成に有効であった。また、RCMの「証跡」と「統制活動の内容」は、システム化における対象機能の優先判断や機能要件の詳細化に役立つなど有効性を発揮した。

図 4.2に販売管理プロセス内部統制基準に定められた業務フローの事例を、図 4.5に同じくRCMの事例を紹介する。

なお、参考として販売管理プロセス内部統制基準に定められた業務フローから、新たに作成した新販売管理業務システムの新業務フローを図 4.3に、システム化業務フローを図 4.4に紹介する。

4.1 ビジネス要求の獲得(BR.1)

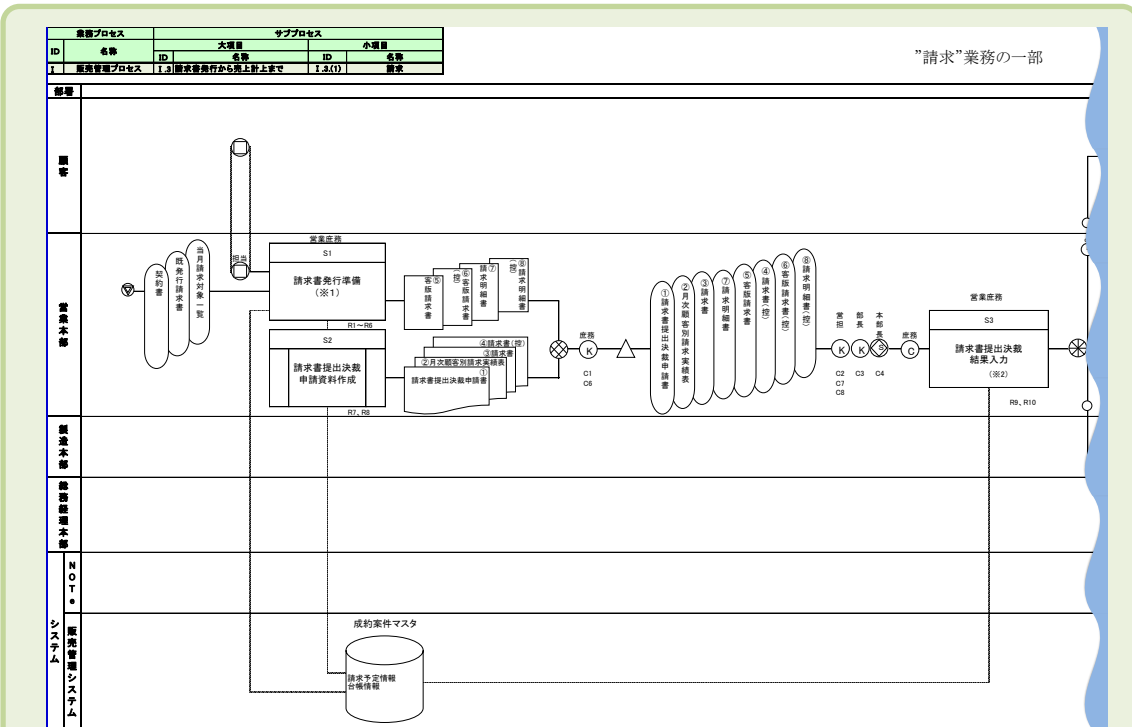


図 4.2 販売管理プロセス内部統制基準に定められた業務フローの事例

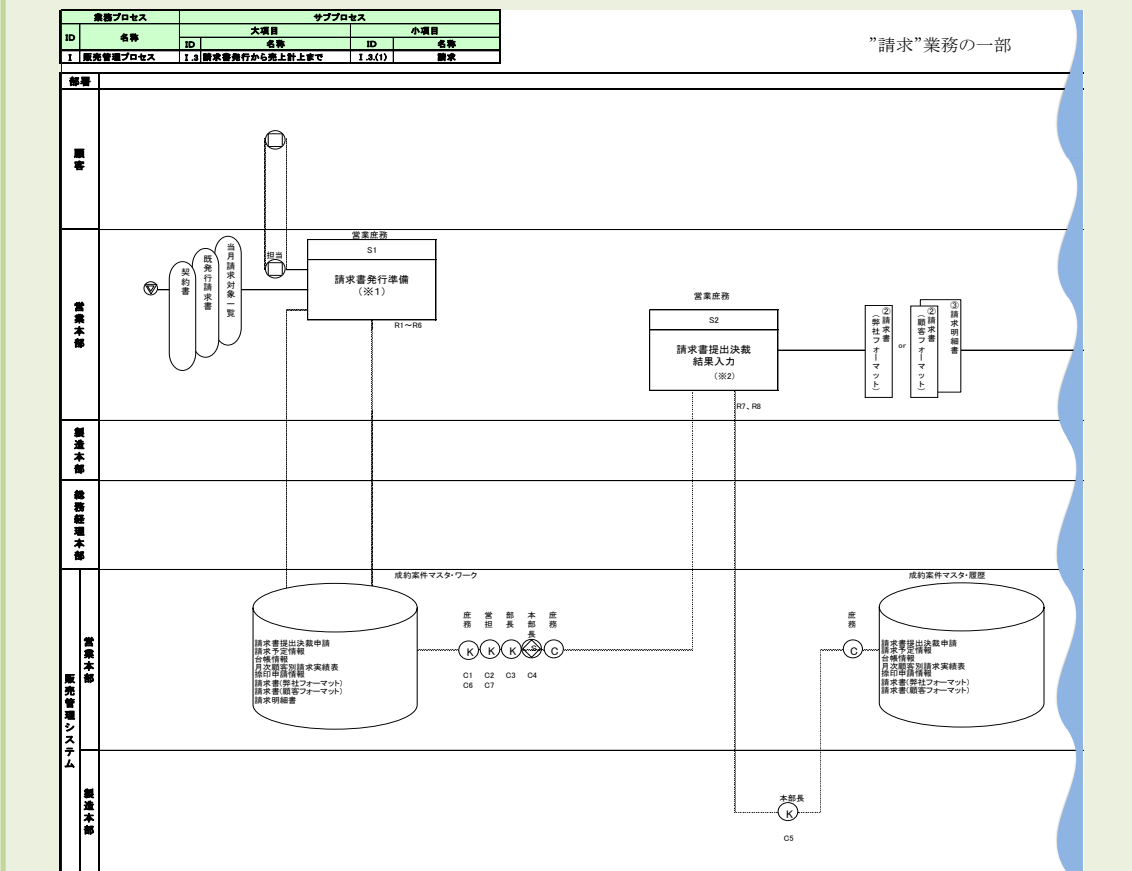


図 4.3 販売管理プロセス内部統制基準の業務フローに基づき作成した新業務フロー

4.1 ビジネス要求の獲得(BR.1)

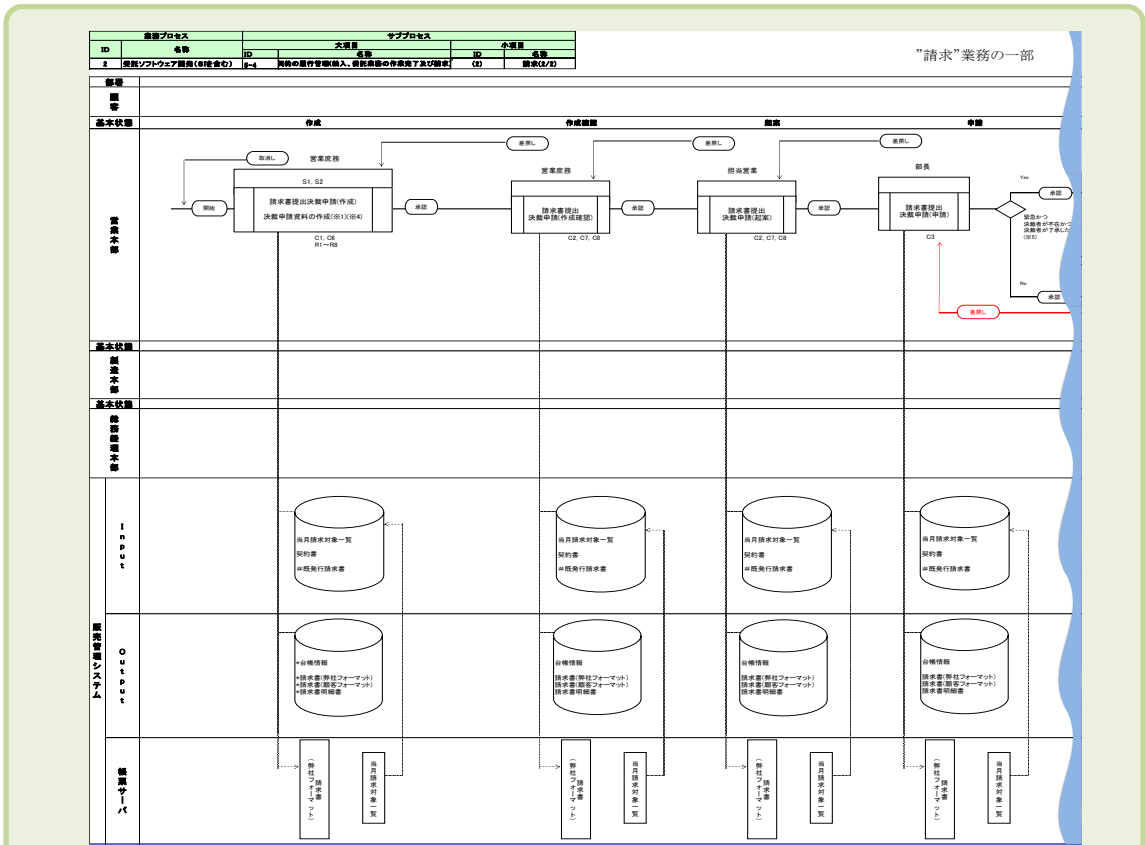


図 4.4 新業務フローに基づき作成したシステム化業務フロー

業務プロセス		サブプロセス												
ID	名称	大項目	小項目											
1	販売管理プロセス	1.1	請求書発行から売上計上まで											
S1	請求書発行準備(注1)	R1	O	承認	振込のない請求が高される。	C1	当月の最初の「売上計上処理」に作成する毎月請求書対象一覧について請求書発行し、かつ請求書を作成する場合は販売管理システムに登録されている作成、変更、金額等の妥当性と照しであることを確認し、実在性、網羅性、及び正確性を確認する。	営業本部	営業部長	-	決裁申請書(実行印)	販売管理システム		
			O	重複	同一取引が重複して請求される。	C2	C1が実施されていることを確認し、決裁申請内容を精査し、顧客様式の請求書を作成する場合は、顧客様が作成済みと重複請求の請求書と販売管理システムが作成した当社様式の請求書と請求金額等を突き合わせることを確認し、実在性、網羅性及び正確性を確認する。	営業本部	担当営業	-	決裁申請書(記帳音印)	-		
			O	漏れ	決裁申請に必要な資料が揃っていないため正しい決裁が行われず請求書等を提出。	C3	G1、G2、D8~D9が実施されていることを確認し、決裁申請内容を確認し、実在性、網羅性及び正確性を確認する。	営業本部	営業部長	「営業業務規程」「営業業務決裁申請運用規程」	決裁申請書(申請音印)	-		
			O	漏れ	請求が漏れる。	C4	C1~C3が実施されていることを確認し承認する。	営業本部	営業部長	「営業業務規程」「営業業務決裁申請運用規程」	決裁申請書(決裁音印)	-		
			O	変更	決裁申請データが事実と異なる内容に差し替えるため請求書等を提出。	C5	C4の決裁内容を精査し、実在性、網羅性及び正確性に問題がないことを確認する。	製造本部	製造本部長	「営業業務決裁申請運用規程」	決裁申請書(確認音印)	-		
		R2	O	重複	同一取引が重複して請求される。	C1~C5	参照	-	-	-	-	-		
			O	漏れ	決裁申請に必要な資料が揃っていないため正しい決裁が行われず請求書等を提出。	C1、C2	参照	-	-	-	-	-		
			O	漏れ	請求が漏れる。	C1、C2	参照	-	-	-	-	-		
		R3	O	漏れ	決裁申請データに事実と異なる内容が記載される請求書等を提出。	C1~C5	参照	-	-	-	-	-		
			O	変更	決裁申請データが事実と異なる内容に差し替えるため請求書等を提出。	C1~C5	参照	-	-	-	-	-		
		S2	請求書提出決裁申請資料作成(注1)	R7	O	漏れ	決裁申請に必要な資料が揃っていないため正しい決裁が行われず請求書等を提出。	C8	決裁に先立ち営業部長がセットアップした決裁申請書一式について必要な書類が揃っていることを確認する。	営業本部	担当営業	-	決裁申請書(記帳音印)	-
					O	漏れ	請求が漏れる。	C7	参照	-	-	-	-	-
S3	請求書提出決裁結果入力(注1)	R8	O	承認	成約案件マスタの内容が決裁書類と異なる内容に修正された旨の請求が高される。	-	-	-	-	-	-			
			O	変更	成約案件マスタの内容が決裁書類と異なる内容に差し替えるため請求書等を提出。	-	-	-	-	-	-			

図 4.5 販売管理プロセス内部統制基準に定められたRCMの事例

4.1 ビジネス要求の獲得(BR.1)

なお、新RCMは、紙での決裁書類に対応する関係者（受付、起案、申請、決裁、確認等）の承認‘印’や紙帳票での確認作業などを電子決裁に変更したことによる改訂にとどまっている。

4.1 ビジネス要求の獲得(BR.1)

(2) 現状業務・システムを理解・共有する

【解決したい問題】

- 現状業務やシステムの分析をしたが、分析した当人以外、誰も内容を把握できておらず、分析結果や全体整合の結果が正しいかがプロジェクトとして判断できない
 - ・ 自分が担当した部分を他の誰もわかっていない
 - ・ 全体を把握している人がいない
 - ・ 業務運用担当者が理解していない

現行業務やシステムを可視化するとき、分担して作業を進めることがある。そのとき可視化を担当した人しかその内容を理解していないという状況がまま発生する。そうになると、ドキュメント作成が手段から目的に変化してしまい、作成が終われば可視化による現状の把握もその時点で達成したという勘違いにつながる。また、業務部門でも自身の業務の全体が把握できなくなっていて作業結果の妥当性を確認できない場合もある。そうなった場合には、この機会に双方で積極的に業務の理解を進める必要がある。

勘どころ① プロジェクト全員での共通認識プロセスを実施する

プロジェクトのメンバー一人がドキュメントを作成するだけでは不十分である。現状に対する共通認識をプロジェクト全員で形成しなければ、同じ方向性を持って問題、ニーズ、課題の分析や目的、手段の検討もできない。

そのため、作成した業務シナリオのプロジェクト全員での読み合わせやウォークスルーを通じて関係者間での認識違いを浮き彫りにするなどをして、お互いの認識を再確認することが必要である。

勘どころ② 業務運用時の人材の育成を要件定義から始める

業務シナリオについての共通認識を形成するだけでなく、現状の業務遂行体制や人材についての共通認識を形成することも重要である。システムライフサイクルからすると開発期間はほんの一部であり、大半は稼働後の運用期間である。したがって、スムーズな稼働だけでなく、安定的に業務を運用するために業務を運用する体制をどのように構築するか、その人材をいかに育成していくかを、要件定義の段階から検討し始める必要がある。そのために、現状の体制、人材についての共通認識を構築することも必要である。

4.1.2 問題・課題の抽出(BR.1.2)

ビジネス要求定義は、各ステークホルダから問題、ニーズ、課題が混在して提示され、そこから解決すべき課題を見極め、施策（ビジネスプロセス改善要求やシステム化要求）を検討するというのが大きな流れである。提示されてくる問題や課題認識は要件定義の出発点になる。しかし、ステークホルダから問題、課題が大量に提示されてきて收拾がつかないとか、本当の問題や課題が出てこないといった問題がこの段階で発生する。ここでは、それらを解決するための勘どころを説明する。

(1) 適切な問題・課題の抽出・分析を行うために、ステークホルダの特性を理解する

【解決したい問題】

- ステークホルダを見誤り、必要な要求が抽出されない
 - ・ 詳細な問題・課題がたくさん出てきて收拾がつかない
 - ・ 真の問題・課題が出てこない
 - ・ 問題・課題の抽出に漏れがある

問題、課題が適切に抽出されないことがある。例えばシステムの利用者から問題・課題を収集したら操作性改善の要求が大量に出てきて收拾がつかなくなった、などである。機能改善が目的であればまだしも、経営への貢献や業務プロセスの改革などがテーマの場合、本当の問題、課題が出てきていないことになる。ステークホルダの特性を理解し、問題・課題を抽出するのに適切なステークホルダを見極めることが重要である。

問題・課題の違いは、(2)を参照していただきたい。

勘どころ① ステークホルダを漏らさない

ステークホルダが漏れていては要求が引き出せない。昨今は、ステークホルダがこれまでよりも多方面に渡ってきている。まずそのステークホルダを漏れなく洗い出せていなければならない。

ステークホルダの洗い出しについては、6.1.2 ステークホルダの特定(RM1.3)を参照していただきたい。

4.1 ビジネス要求の獲得(BR.1)

勘どころ② 各ステークホルダの問題・課題認識の違いを認識する

ステークホルダが明確になったら、問題や課題を抽出するのに適切なステークホルダを見極める必要がある。ステークホルダはその立場によって問題・課題認識が異なる。

各ステークホルダの関心事を以下に挙げる

- 経営層・事業部門長など
理念、長期ビジョン、提言
対社会的な問題意識
他社比較
戦略、利益拡大 など
- マネージャ・リーダーなど
全体俯瞰の視野
非定型業務に関心
リスクの回避、防止
判断のタイミング など
- 実務スタッフなど
個人起点の発想
定型業務
ムダ・ムラ・ムリの改善
既存業務やシステムが大きく変わって欲しくない など

大きな業務プロセス改革をしようとした時に、現場のスタッフ系の人々をたくさん集めて問題・課題抽出を行うと、小さな改善要求やシステムの操作性向上要求などが大量に噴出し整理や対応に苦慮することもある。決して現場のスタッフ系を集めるのが悪いと言っているわけではないが、ステークホルダの特徴を意識して望むことが大切である。

勘どころ③ ステークホルダ間の対立・関連を見極める

ステークホルダを抽出して課題を分析していくと、各ステークホルダが提示する課題に関連性が発生することがある。そのような状況をそのままにして個々の課題や解決策の定義を行っても、最優先で解決すべき課題を見落としてしまい、ステークホルダが納得して合意できなくなる場合もある。

そのような場合、ステークホルダの課題をリッチピクチャ [19]として図や文字、絵で表現し、対立する課題や類似する課題を可視化して整理すると良い。リッチピクチャは要求に関連するステークホルダ間の状況を一枚絵で表現する技法であり、自然言語では表現しづらい関係性の表現、新しい発見の気づき、一目瞭然で分かる理解容易性の向上とい

4.1 ビジネス要求の獲得(BR.1)

った効果がある。リッチピクチャの例を図 4.6 に示す。この例はお見合いパーティ運営会社のイベント企画業務改革を行う上で、ステークホルダ（運営会社社員、顧客）のニーズ、課題、問題を表現し、要求間の関係性を可視化した例である。

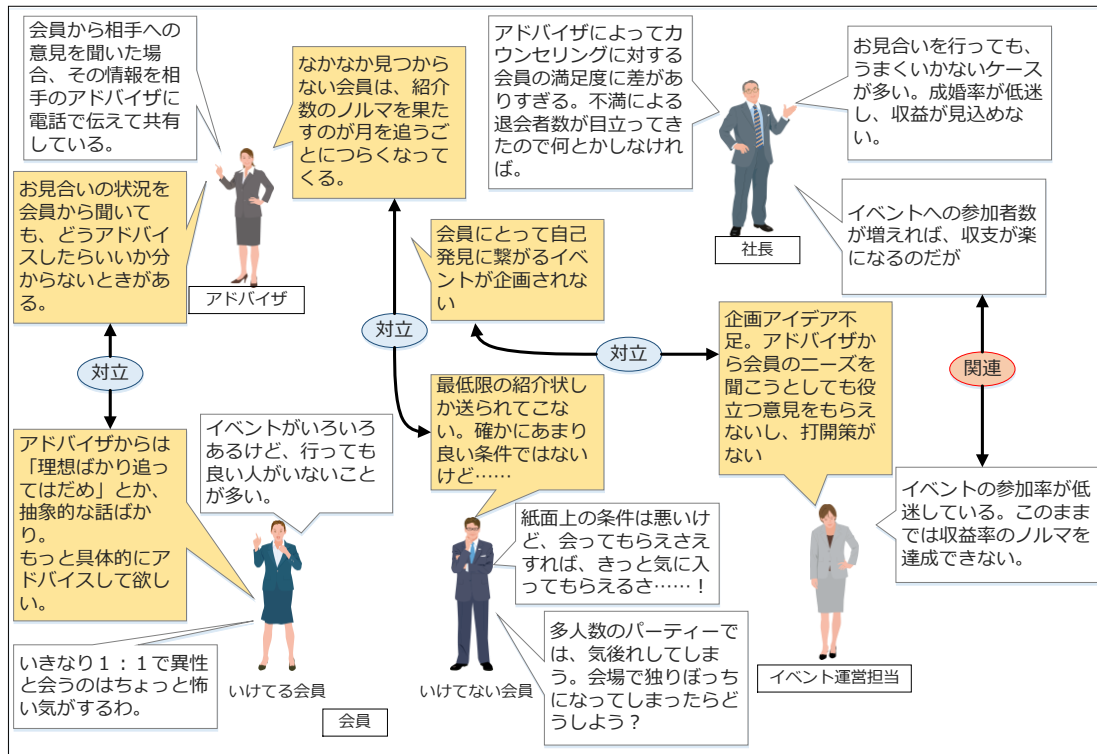


図 4.6 リッチピクチャを用いた課題整理例

対立が多い、提示者が多い課題は解決策がもたらす影響範囲が広く、合意形成が困難である。そのような課題から、目的に照らし合わせた重要課題の深掘り、解決策の導出を行うことでステークホルダとの合意形成、調整をスムーズに進めることができる。

4.1 ビジネス要求の獲得(BR.1)

(2) 問題の原因を分析し解決テーマ（課題）を抽出する

【解決したい問題】

- 真の問題・課題の抽出ができていない
 - ・ 効果的な要求（対策）にならない

要求すなわち目的/目標の設定や手段（対策）の抽出をしたとき、これが真の要求なのか、他にもっと効果的な対策を講じられる要求はないのか不安になることがある。そんなとき、例えば、現場は本当は何に困っているのか、本当の問題は他にあるのではないか、などと深掘してみると、問題・課題の捉え方が不十分だったということがある。

効果的な要求すなわち目的/目標の設定や手段（対策）を見極めるためには、その要求が出てきた理由に当たる問題や課題などの源泉を的確に捉えることが一つの解決方法になる。

勘どころ① 問題と課題の違いを意識する

問題と課題の違いについて辞書を引いてみると、問題とは「研究・論議して解決すべき事柄」、課題とは「課せられた問題」などとある。分かるようで分からないといったところであろう。このガイドでは、問題と課題を以下のように定義する。

問題：「あるべき姿と現状との負のギャップ」であり「解決を要する事実」

課題：「問題を解決するために何をすべきか」「解決すべきテーマ」

例えば、問題とは「コストの計画値が 100 の予定であったのに対し、実績値が 150 であり、予定と結果が異なった。」という事実であり、「課題」とは、この例でいうと「生産性が低いので生産性を高める必要がある」とか「生産性は妥当だがコストの見積り精度が悪いので見積り精度を向上させる必要がある」という事になる。すなわち、解決すべきテーマとして設定したものが「課題」である。

用語の定義をこうするべきだと述べているわけではなく、この違いを意識して要求の抽出をして欲しいということが趣旨である。

勘どころ② 要求の源泉を分析する

要求の分析の出発点は図 4.7 に示すように、問題、ニーズ、課題である。問題とは、現状の業務などで起こっている事実である。ニーズとは市場などから求められている要望である。問題から出発するか、ニーズから出発するかは、都度異なって良い。「ニーズ

4.1 ビジネス要求の獲得(BR.1)

を叶えたいが、このような問題がある」「このような問題があるのでニーズに対応できない」というように、裏腹の関係にあるからだ。昨今では、新たなビジネスやサービスを検討するときは、ニーズだけから発想することもある。問題、ニーズを明確にするということは、要求を定義した理由、すなわち「なぜ(原因)」を明確にすることである。「なぜ」がないと納得感が低く、課題や解決手段に対する合意形成がしにくい。

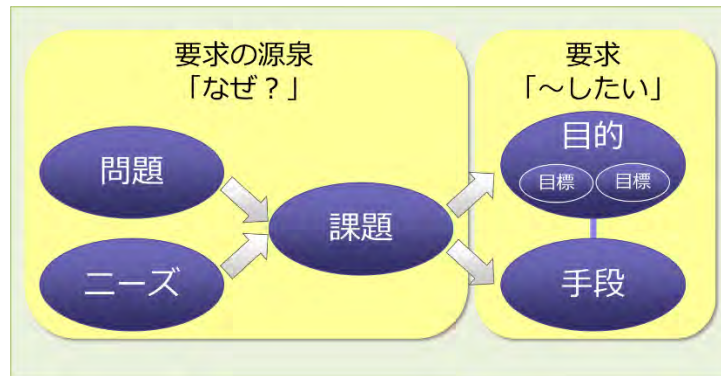


図 4.7 要求分析の基本要素の関係

ニーズや問題が明確になったら課題を設定する。課題とは解決すべきテーマのことである。

この課題（解決テーマ）が設定できれば、目的を明確にして、手段を検討して、現実在即した本質的な解決方法が導かれる。この課題（解決テーマ）をいかに「適切」に設定できるかがポイントとなる。

勘どころ③ なぜなぜ分析を行い、真の原因を見極める

真の原因を見極めるためには「なぜなぜ分析」が有効である。

図 4.8 は、「情報系システムが利用されていない」という問題の原因分析の例である。

図に示した「探す場所が複数ある」という問題に対処しようとするれば、「サーバー統合」などが考えられる。「情報が多過ぎる」に対処しようとするれば、「いかに情報を有用なものだけに絞り込むか」を考えなければならない。両方かもしれない。原因によって対処方法が異なるということが分かる。どの問題を解決すべきかを検討することが大切である。

4.1 ビジネス要求の獲得(BR.1)

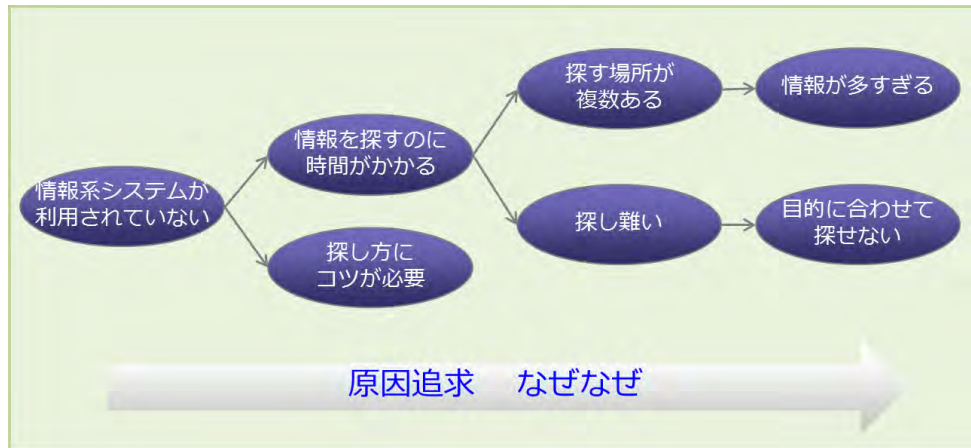


図 4.8 なぜなぜ分析のイメージ

勘どころ④ 解決テーマ（課題）を適切に設定する

前記のような原因分析を実施することにより、本当に解決すべき課題を見つけることができる。

すべての原因を解決しようとして要求が膨らむことがないようにすることが「適切」に設定するポイントの一つである。プロジェクトで本当に解決しなければならない「真の原因（問題）」を見極め、何を「課題」として設定するかを分析、検討する。その際には、ボトルネックになっている問題、効果の大きな問題などを見極める必要がある。

4.1 ビジネス要求の獲得(BR.1)

4.1.3 ゴールの抽出(BR.1.3)

多額の IT 投資を実施して新たなシステムを構築したが、ビジネスプロセスは変わらず、箱（ハード）やユーザインターフェースが変わっただけで、投資に見合わなかったという経営層の声をよく聞く。今日の IT システムには、今実施されている手作業の機械化に代表される事務作業の効率化よりも、経営やビジネスへの直接的な貢献が求められる。そこでは、提示された要求の経営的、業務的な価値も要求の品質として捉える必要がある。

一般的に、経営方針やシステム化方針は、構想立案や計画立案で打ち出される。要件定義工程では、この経営方針やシステム化方針を達成できるように業務要件やシステム化要求を定義しなければならない。しかし実際のプロジェクトを見ると、要件定義の内容が手作業の機械化や操作性向上の検討に終始し、経営方針やシステム化方針が置き去りにになっていることがある。

なぜこのようなことが発生するのか、いくつか理由が考えられるが、一つは、開発する側（情報システム部門やベンダ企業）の担当者が変わることが挙げられる。要件定義工程では、開発する側も検討に加わる。開発する側にとっての要件定義工程での最大命題は、作って欲しいもの（仕様）を明確にすることである。したがって、どれくらい企業や業務をよくするか、という漠然としたものよりも、今ある業務やシステムの延長上で自分たちがどんな機能を作らなければならないかという明確に議論できるものに重きが置かれやすい。

もう一つの理由は、ステークホルダの価値観が異なることである。ステークホルダには、経営者、業務部門の長、業務のリーダー、業務の担当者、システムのオペレータ、IT システムの利用者などさまざまな存在する。これらのステークホルダは各自が重要視するポイントが異なる。例えば、IT システムの利用者であれば、個人視点での作業の改善（ムダ、ムラ、ムリ）に関心が高い。収集した要求も「こんな機能を作って欲しい」「今の機能をこう改善して欲しい」など操作性に関するものが多くなる。これらの要求と、経営方針やシステム化方針には乖離がある。さまざまな価値観を持ったステークホルダから抽出した要求は、整合性を維持しながら整理し、最終的に経営方針やシステム化方針を実現できるよう取捨選択していかなければならない。

まずは、ゴールである経営レベルの目的・目標やそのサブゴールである業務レベルの目的・目標などを明確にし、共通認識して、そこに向かって要求分析を進めていく必要がある。

目的・目標の違いについては、(2)を参照していただきたい。

(1) 経営レベルの目的・目標を明確にして共通認識する

【解決したい問題】

- 経営に貢献する要求、経営方針に合った要求が抽出されない

4.1 ビジネス要求の獲得(BR.1)

- ・ 経営レベルの要求を収集していない
- ・ 経営レベルの要求を経営層と確認していない

経営やビジネスに貢献する要求を抽出するためには、経営レベルの目的・目標や施策が明確でなければならない。本来は、経営レベルの要求は、構想立案書やプロジェクト計画書など事前の工程の資料に掲載されているか企業方針などで明確になっているはずである。しかし、これらが不明確のままプロジェクトが開始されることもある。その場合には、経営レベルの目的・目標や施策を確認し、必要があれば整理して経営層に確認を求めることが必要になる。

勘どころ① 経営レベルの目的・目標、経営施策を明確にする

経営レベルの目的・目標や施策は、システム開発プロジェクトが開始される前の構想立案時やシステム化企画時に提示されているのが本来の姿であり、要件定義工程でそれを検討するのは誤りである。要件定義工程では、提示された経営レベルの目的・目標や施策を確認することが必要である。構想立案やシステム企画が省略されている場合でも、中長期計画や年次方針などで提示されていることがあるのでそれらの情報を整理して経営層に確認し、向かうべき方向性を合意する必要がある。

勘どころ② 経営レベルの目的・目標を見極める

目的・目標にはレベルがある。経営レベル、事業部門長、リーダー、現場の実務者などそれぞれに異なるレベルの目的・目標が存在する。そのため、検討する目的・目標を誤ると、要件定義の結果が経営の意図とは異なるシステム化要求になっている事がある。例えば、システムの要件として現場の利用者から要求を吸い上げると、経営レベルの目的・目標とは異なり、便利で操作性の良いシステムにするための要求（手段としての要求）ばかりがたかさん上がってくるなどである。これを防止するために、経営レベルの目的・目標を明確にし、そこに向かった要求を抽出、取捨選択する必要がある。

経営レベルの目的・目標の種類を以下に挙げる。

- 売上を上げる
- コストを下げる
- 品質を上げる
- 納期を守る
- 顧客満足度を上げる
- 安全性を上げる
- リスクを下げる

4.1 ビジネス要求の獲得(BR.1)

- モチベーションを上げる
- コンプライアンスを守る
- 環境貢献、社会貢献

これらの経営レベルの目的・目標をもとに、業務レベルや現場レベルの目的・目標に展開していく。手作業の機械化の時代の IT システムは、効率化＝コストダウンが主たる目的だった。しかし、効率化すべき業務がほぼシステム化された現在、経営層は、コストダウン以外の価値も求めてきている。

まず、このような経営レベルの目的・目標を明確にする必要がある。

勘どころ③ 経営施策（手段）を見極める

次に経営施策すなわち経営層が打つ施策（手段）を明確にする。経営施策とは、例えば、新たな営業拠点を作る、新たな物流倉庫を作る、組織を変える、などである。これらも、構想立案などで明示されているので洗い出して明確にする。これら経営施策は、これから実施する要件定義の与件となる。

4.1 ビジネス要求の獲得(BR.1)

(2) 業務レベルの目的・目標を抽出する

【解決したい問題】

- 経営に貢献するためのサブゴールである業務レベルの目的・目標が明確でない
 - ・ 業務として目指すものが不明確
 - ・ システム化要求が経営レベルの要求にどのように貢献するかが不明確
 - ・ システムの操作性要求が多くなる

経営やビジネスに貢献する要求が抽出できない原因として業務レベルの目的・目標が不明確であることがある。経営レベルの目的・目標を最終ゴールとすれば業務レベルの目的・目標はサブゴールに当たる。サブゴールを設定せずに手段、施策を検討するとややもすると経営を意識し過ぎて業務に貢献しない要求になったり、逆に意識が薄れて経営に貢献しない要求になったりする。経営レベルの目的・目標と業務レベルの目的・目標の双方を整合性を持って設定することも一つの解決方法である。

勘どころ① 目的と手段の違いを意識する

目的と手段は異なるものであるが、それらを正しく使い分けられていないことがしばしば見受けられる。このガイドでは、「目的」と「手段」を以下のように定義する。

目的：状態の変化、「～を改善したい」「価値を上げたい」といった経営や業務で何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、達成したい結果
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施すべき対策を実行可能な内容で定義したもの

要求は「～したい」と表現できる。「自動集計する機能を作成したい」は要求である。これを実現して何を達成したいのか。「工数を削減したい」これも要求である。

要求には、「～を実施したい」と「～を達成したい」の2種類が存在する。

前者は「手段」や「実行・実施」に対する要求であるのに対して、後者は「目的」や「価値の変化」に対する要求である。まずはこの2つ、“手段”を表す要求と“目的”を表す要求の違いを意識することが重要である。

上記は、用語の定義をこうするべきだと述べているわけではなく、この違いを意識して要求の抽出をして欲しいということが主旨である。

勘どころ② 目的と目標の違いを意識する

目的と目標を本書では以下のように定義する。目的は目指すべき方向でありゴールである。目標とはその目的を達成するためのマイルストーンである。目標は、今はこのくらい（現状値）だが、いつまでに（達成時期）、どのくらいの効果（目標値）を目指すのかを示す具体的な時期と指標を持っている。このように定義すると整理しやすい。目的に対して目標を設定することが重要である。目的と目標の関係を図 4.9 に示す。

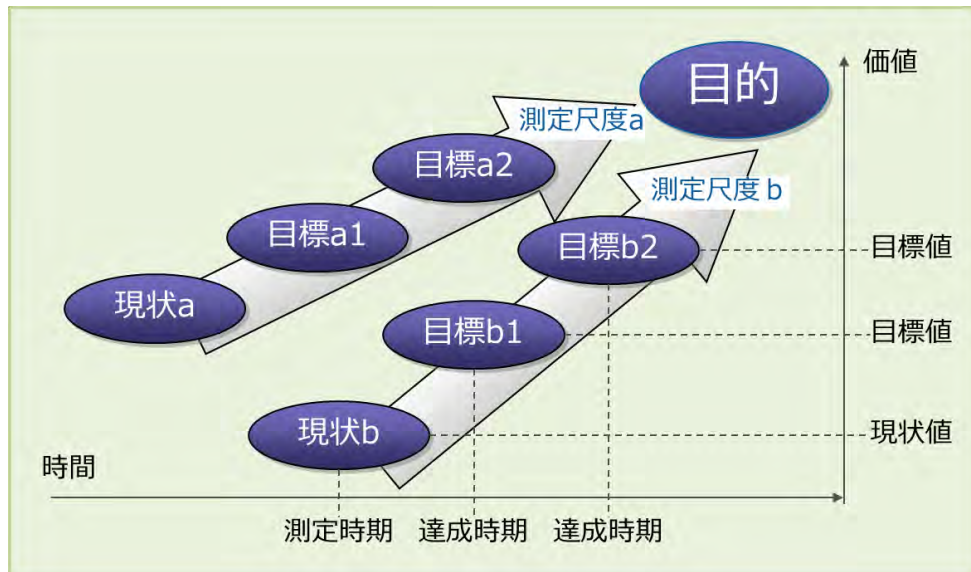


図 4.9 目的と目標の関係

ここも用語の定義をこうするべきだと述べているわけではなく、この違いを意識して要求の抽出をして欲しいということが主旨である。

勘どころ③ 目的・目標にはレベルがあることを意識する

要件定義には、経営者、業務部門の長、業務のリーダー、業務の担当者、システムのオペレータ、IT システムの利用者など価値観の異なるステークホルダが多数存在する。各ステークホルダにはそれぞれの目的・目標がある。経営やビジネスに貢献する要求を見極めるためには、ステークホルダそれぞれの異なる目的・要求を図 4.10 のように関連付ける必要がある。

4.1 ビジネス要求の獲得(BR.1)

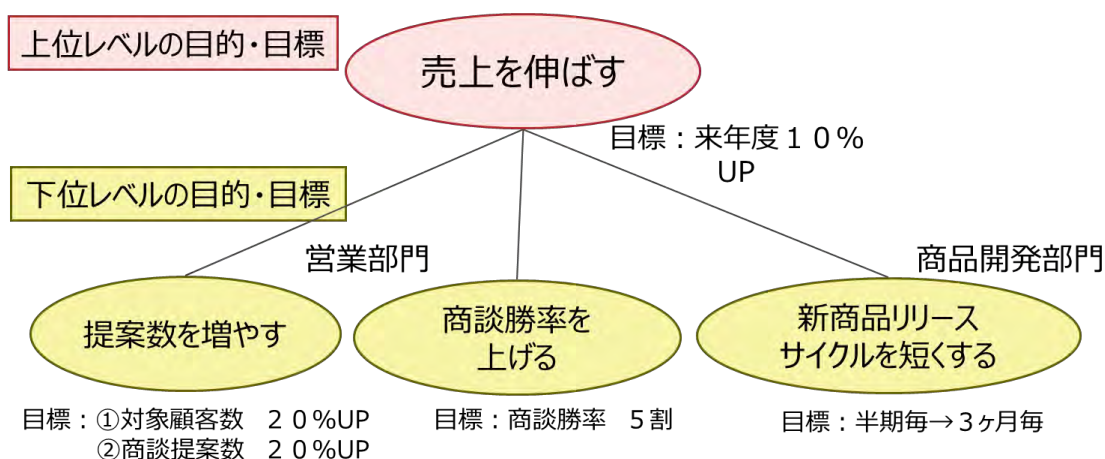


図 4.10 目的・目標の上位・下位関係

図 4.10 に示すように、「売上を伸ばす」という目的は、各ステークホルダの活動を関連付けた結果であり、より上位レベルの目的となる。仮に売上が未達成だったときに、営業部門は商品が悪いから、商品開発部門は売り方が悪いからと目標未達の原因が他のステークホルダの影響を受けることになってしまうので、「売上を伸ばす」という目的は、営業部門や商品開発部門の目的より、上位レベルの目的になる。

勘どころ④ 業務レベルの目的・目標を抽出する

経営レベルの目的・目標や施策は、抽出というより要件定義への与件の確認であった。経営や業務に貢献する IT システム化要求を抽出するには、経営レベルの目的・目標を目指した下位（業務レベルの目的）の目的・目標に相当する要求を獲得しなければならない。

業務レベルの目的・目標の抽出には2つのアプローチがある。経営レベルの目的・目標に沿って部門長などが決めるトップダウンアプローチと、現場の意見を聞いて現場の課題や要求を実現するために設定するボトムアップアプローチである。経営レベルと現場の双方の要求を満たしたものにしたいという思いから、両方のアプローチを併用して抽出することも多い。

経営レベルの目的とシステム機能レベルの要求（手段）だけがあり、業務レベルの目的がない中抜けの要件定義を見かけることがある。この場合、完成したシステムは経営層の意図とは異なるものや、業務プロセスの変わらないものになりがちである。

経営層の意識とシステム開発を有機的につなげるためにも業務レベルの目的の抽出・設定が重要になる。

4.1.4 手段の抽出(BR.1.4)

目的が明確になったら次は手段の抽出である。手段とは目的を達成するために「～を実施したい」という内容の要求である。経営レベルの手段すなわち経営施策は前ステップで明確にしたので、ここでは業務レベル以下の手段である業務手段や IT 手段を抽出する。

(1) 目的・目標を意識して手段を抽出する

【解決したい問題】

- 効果的な手段が抽出できない
 - ・ 経営や業務に目的・目標に効果的な手段が抽出できない
 - ・ システム化の手段やシステム操作性の手段が多くなる

手段にも経営レベルの施策もあれば業務レベルの施策もある。システム化も手段である。システム化は必要とせず業務運用を変えるだけの手段もある。これらの使い分けが十分にできていないと効率的な手段にならない。また、目的を意識することも重要である。問題から直接解決策を検討すると視野が狭くなりがちである。目的にいったん展開してから手段を検討した方が新たな手段が見つかることもある。

勘どころ① 手段のレベルを意識する

手段にも前述した経営レベルの手段（経営施策）から実業務上の手段（人が行う手段など）、システム化するという手段、などがある。これらの手段と目的が有機的に関係づいている必要がある。

目的、手段の全体のイメージを図 4.11 に示す。

4.1 ビジネス要求の獲得(BR.1)

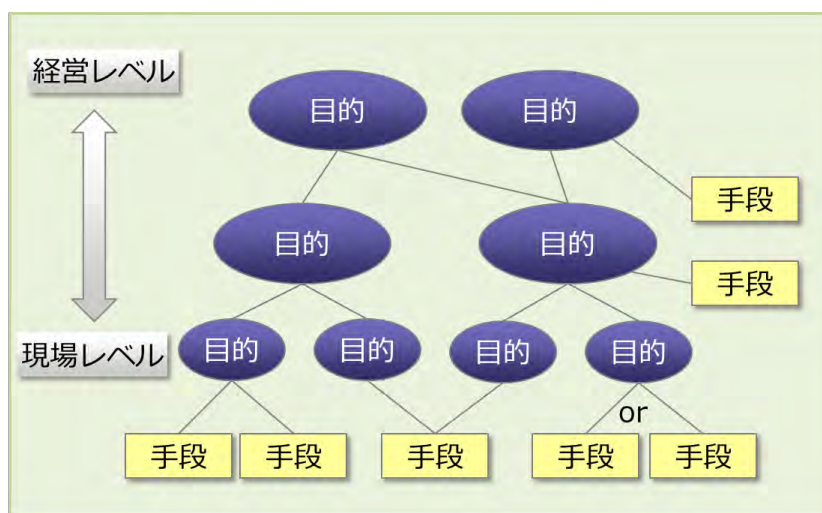


図 4.11 関連付けられた目的、手段のイメージ

手段の表記として、上位の目的を達成するためには、下位の手段すべてが必要であるといった「and 関係」と、候補が複数ありどれか一つで良いといった「or 関係」がある。各手段がどちらであるか分かるようにしておく。

勘どころ② 手段の十分性から他の手段がないか検討する

手段は複数考えられる。より効果的な手段がないかを探ること（新たな要求の抽出）と、複数の案の中からどの手段が目的を達成するために最適な方法かを検討することが重要である。

図 4.12 に示すように、「現状の問題」を仮に「手作業で集計していること」とする。問題を解決する方法として、まず「自動集計するシステムを構築する」が思い浮かぶ。ここで、一度目的に立ち返り、問題を解決することでどんな目的を達成したいのかを検討する。仮に「情報提供スピードの向上」が目的であったなら、この目的の達成のためにどんな手段が考えられるかを検討することができる。「集計前のデータでも良いから早くデータを利用できるようにしては？」「BI(Business Intelligence)ツールを導入し、現場が必要とする形で自由に集計できるようにしては？」といったアイデアが出る。大切なのは、目的を達成することである。すなわち、目的を達成するためにこの手段で十分かどうかを検討することでより効果的な新たな要求が抽出できるかもしれない。

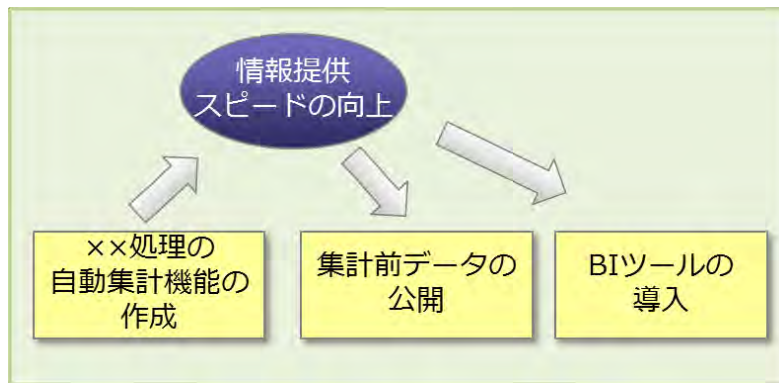


図 4.12 十分性の検討イメージ

勘どころ③ 業務手段・システム手段の双方を抽出する

ITシステム化して提供して欲しいという要求は「手段」に対する要求である。ITシステムからのサービスを利用して業務を実施する業務側からの要求であり、システム開発が前提の場合、システム部門やベンダ企業はこれを重要視しがちである。しかし、忘れてはならないのは、業務としての「手段」要求の抽出である。すべてをITシステムの機能という「手段」だけではなく、業務自体を見直す、業務の運用で補うなど、業務手段である要求の抽出も忘れてはならない。業務の改革・改善が前提の場合は特に重要である。そのイメージを図 4.13 に示す。

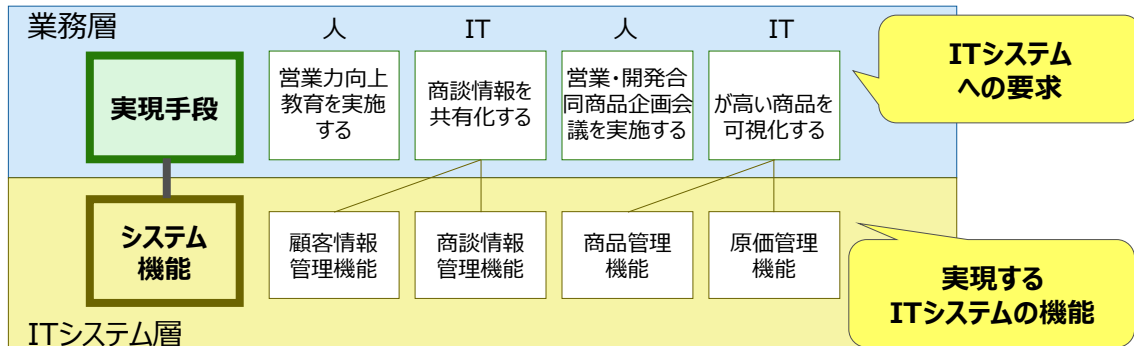


図 4.13 業務手段とシステム化要求

コラム 5 アイデア発想の方法

【悩み】

「目的」を達成するための「手段」は一種のアイデアである。昨今、新ビジネスや新商品などイノベーション時代を迎えるに際して多様なアイデア発想の手法やアプローチが着目されている。本書はイノベーション時代の要件定義がテーマではないが、基本的な要件定義でも参考になるのでここで数点簡単に紹介する。詳細は文献等で調べて欲しい。

【解決案】

(1) ブレインストーミング

否定、批判しないことで意見を出しやすくし、数多くのアイデアを発想するワークショップ。集団思考、集団発想法で量を重視した方法。以下の4つの原則がポイント。

- ・ 否定、批判をしない
- ・ 自由奔放
- ・ 質より量
- ・ 連想と結合

(2) ゼロベース思考 [20]

既存の枠組みでは、過去の事例やさまざまな規制などが思考の幅を狭くし、目的への最適な方法への到達を難しくなるため、既存の枠組みにとらわれず、目的に対して白紙の段階から考えようとする考え方の姿勢のこと。ポイントは以下である。

- ・ 「顧客の立場から考える」
顧客の立場から考えてみると、自分達の前提は意味がないことが分かる。
- ・ 「ポジティブな考え方をする」
「できない」「無理だ」の考えを外して、「どうしたらできるか」という考え方で進める。

(3) 共創

多様な立場の人たちと対話しながら、新しい要求、価値を「共」に「創」り上げていくこと。今までに関わりのない頭脳を持った人と人との組み合わせでアイデア発想をする。既存の枠組みにとらわれない素朴な発想や、想像できなかったものとの組み合わせで新たなアイデアが出ることを狙う。

(4) SCAMPER 法 [21]

以下のチェックリストに当てはめて強制的にアイデアを発想させていく方法。

- Substitute (別のものに置き換えるとどうか?)
- Combine (2つ以上のものを組み合わせるとどうか?)
- Adapt (応用したらどうなるか?)
- Modify (修正するとどうなるか?)
- Put other purposes (別の用途に使うとどうなるか?)
- Eliminate (何かを取り除くとどうなるか?)
- Rearrange/Reverse (並び替えるとどうなるか?逆にするとどうなるか?)

(5) PoC(Proof of Concept)

概念実証のこと。これは新たな概念やアイデアの実現可能性を示すために、簡単かつ不完全な状態でもいいので作ってみる。作ってみるとそのアイデアが具体化されたり、効果がないなどが見えてくる。

(6) デザイン思考/人間中心設計

HCD(Human-Centered Design)/UX(User Experience)

もっともっと人の立場、気持ちに立ったアイデア発想アプローチ。これらの方法には正確には違いがあるが、基本となる考え方は同じである。高機能なモノ、品質の良いモノなどモノ中心のアプローチではなく、人に役立つモノ、人の気持ちに立ったモノ、人に喜ばれるものなど人中心でのアイデア発想アプローチ手法。

他にも色々あると思うので探して欲しい。

4.2 ビジネス要求の分析(BR.2)

「ビジネス要求の分析」では、「ビジネス要求の獲得」で抽出された要求と要求間の関連性や一貫性などを吟味して要求の品質を上げるとともに、要求の詳細化・具体化を行い業務的な検証を行う。

4.2.1 要求の体系化(BR.2.1)

要求の体系化は、「ビジネス要求の獲得」「ビジネス要求の分析」「ビジネス要求の文書化」の3つのカテゴリを横断して実施される。「ビジネス要求の獲得」では、前述したように要求の体系を意識しながら要求の抽出を目的に実施される。「ビジネス要求の分析」では、「ビジネス要求の獲得」で抽出した要求をあらためて体系的に統合して見直すことにより実施される。見直しによって新たな要求が抽出されたり、不必要な要求を取り消したりする。

(1) 要求全体の整合性を分析し評価する

【解決したい問題】

- 抽出した要求が全体として効果的かつ必要十分なものになっていない
 - ・ 目的に合致しない要求が多くある
 - ・ もっと効果的な手段がある。
 - ・ 膨らむ要求を捨てられない

要求の抽出結果には、各ステークホルダの思いが反映されることがある。それらは、経営者や業務部門長の思いと異なったり、目的を達成するための効果的な要求になっていなかったりすることがある。上位目的を意識したサブゴール（下位目的）の抽出や手段の抽出を行うようにと、前述のビジネス要求の獲得時に述べている。基本的な考え方は同じであるが、ここでは、抽出した要求全体を体系的に整理し全体としての整合性、十分性を再確認することを提示している。

4.2 ビジネス要求の分析(BR.2)

開発側「自動集計するとどれだけコストダウンするのですか」
利用者「毎日一人が3時間かけて実施していた作業が減ります」
開発側「一人のためですか？目的は他にありませんか」
利用者「集計データをいち早く顧客に提供できます」
開発側「早く集計データを提供すると、どんなメリットがあるのですか」
利用者「(顧客の)機会損失が低減します。ひいては顧客満足度が向上します」

この検討結果を表現したものを図 4.15 に示す。

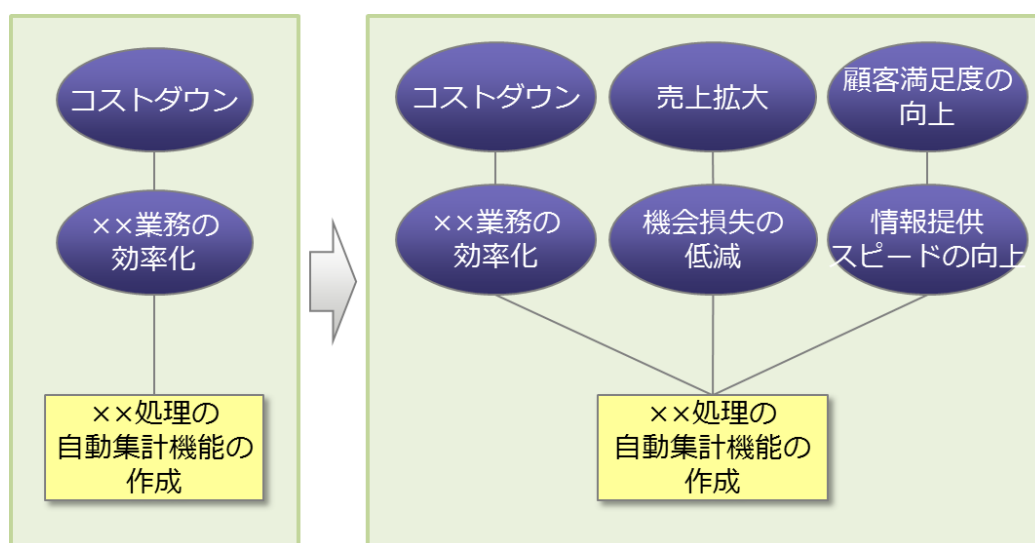


図 4.15 目的を精査した結果のイメージ

勘どころ③ 十分性の観点からより効果的な手段がないか検討する

「十分性」とは、図 4.14 にあるように、構造を上から下に検証することである。上位の「目的」に対し下位の「手段」や「サブ目的」が必要十分であるかということである。抽出された「手段」や「サブ目的」だけで「目的」を達成することができるか、「目的」を達成するために必要な「手段」や「サブ目的」がまだ不足しているのではないかを検証することである。

目的から手段を抽出することは、前述「4.1.4 手段の抽出 (1) 目的・目標を意識して手段を抽出する」の「勘どころ② 手段の十分性から他の手段がないか検討する」で述べられているので、そちらを参照していただきたい。

事例 2 ビジネスに貢献する要求を見極め、ビジネス成果に直結する IT 施策を明らかにする (サントリーシステムテクノロジー株式会社)

【取り組み背景と課題】

システム開発スタート時のコンセプト検討段階で狙ったビジネス成果があったにもかかわらず、いざ要件定義を終えてみると、当初のコンセプトとは異なる要件・施策になっていることがままある。それでは、開発したシステムをいくら利活用してもビジネス成果を獲得できない。この課題を解決すべく、サントリーグループでは、リザルトチェーンを描いて、ビジネス成果に直結するIT施策を明らかにしている。

【解決策（リザルトチェーン）】

リザルトチェーンとは、獲得したいビジネス成果と、それに必要なIT施策との関連を明らかにする整理ドキュメントである。このドキュメントを作成することで、次の効果を獲得できる。

- ・ 自ら定義した最終ビジネス成果（売上増大、業務コスト削減、企業価値向上、リスクヘッジなど）と、その成果に直結する IT 施策を見える化できる
- ・ その最終ビジネス成果と直結する IT 施策をステークホルダで合意できる

図 4.16は、サントリーグループのSCMに関するリザルトチェーンである。一番右側に「業務コストを〇〇億円削減する」など計3つの最終ビジネス成果を配置している。一方、一番左側に「需要計画を取り込む」「計画修正の比較ができるための情報を入力する」などのIT施策を配置し、その施策が「計画修正検討・比較ができる」という機能を実現できる。次にその機能が「在庫計画・生産配分の調整時間を短縮できる」という中間成果を生み、「業務コストを〇〇億円削減する」という最終ビジネス成果に直結する。

このリザルトチェーンの凡例は、図 4.17のとおりである。リザルトチェーンは基本的には「成果」「施策」「前提・制約条件」の連鎖で表される。また、それぞれの因果関係は「貢献」と言う。

「成果」は、リザルトチェーンの左から右へ、①構築する機能、②その機能が業務に与える好影響、③その影響による活用（中間成果）、④その活用による恩恵（最終ビジネス成果）に分類される。「成果」の中には、業務へマイナスの影響を与えるものがあり、それらに対してはマイナスを打消す施策が必要である。

「施策」は、成果をもたらす活動であり、ITによる施策だけではなく、BTOPP（ビジネス戦略、技術（IT）、組織・文化、業務プロセス、人）の観点で施策を

4.2 ビジネス要求の分析(BR.2)

抽出し、記載する。

「前提・制約条件」は、施策がうまく機能するための条件である。

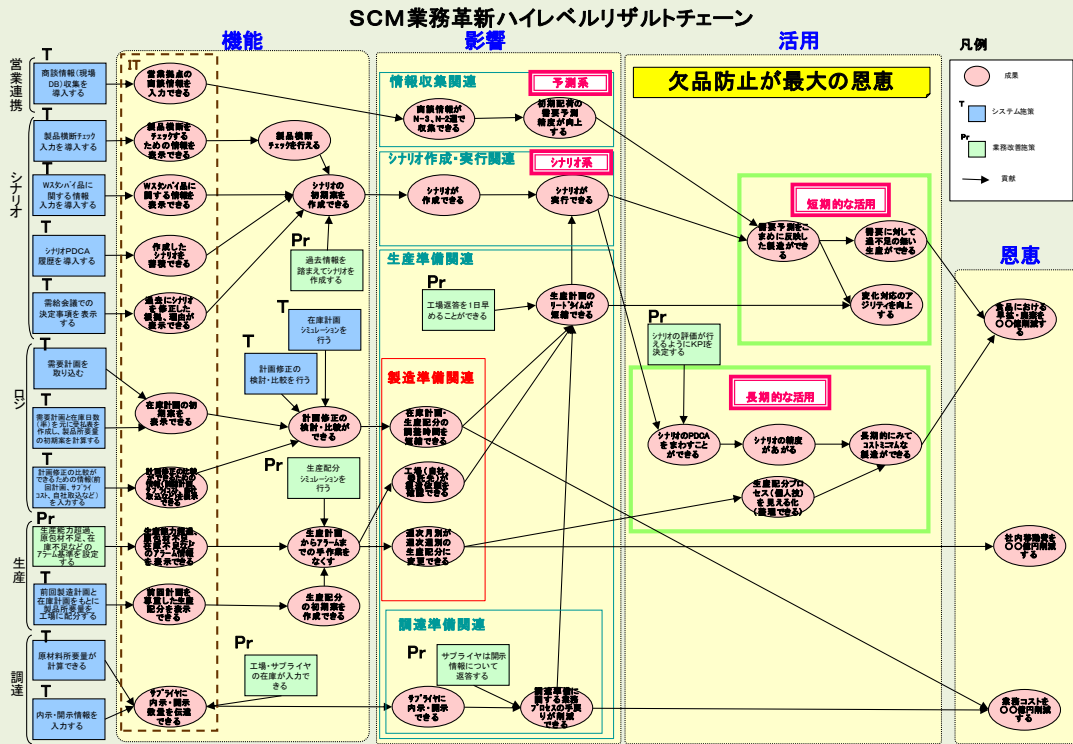


図 4.16 リザルトチェーンの事例

記号	概要
	成果 において実現すべき成果 (最終および中間成果から構成)
	施策 1つ以上の成果をもたらす活動 (BTOPP全てを考慮して定義)
	前提・制約条件 施策が上手く機能するための条件 (リスク)
	貢献 因果関係 (上位施策に対する下位施策の貢献度)

BTOPP とは
 B : Business (ヒ ス戦略)
 T : Technology (情報技術)
 O : Organization (組織・文化)
 P : Process (業務フ ス)
 P : People (人)

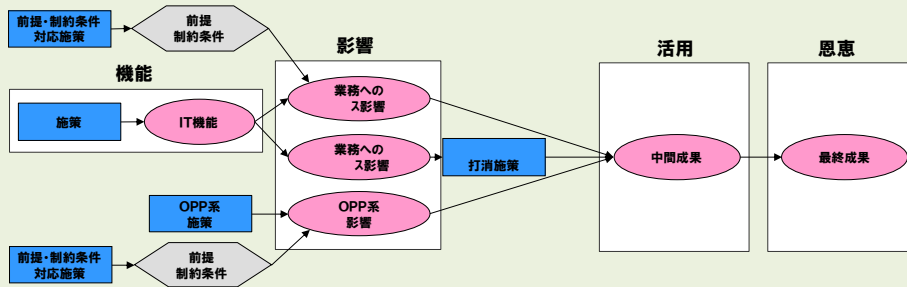


図 4.17 リザルトチェーンの凡例

これらを踏まえて、リザルトチェーンの作り方を、以下図 4.18、図 4.19 (STEP1～5) に示す。このリザルトチェーンを作成することで、捉えた真の問題を元に、あいまいな目的や手段を具体化し、経営方針やビジネス目的とIT施策との関係を明確にしている。

なお、リザルトチェーンの楕円形で表される成果はKGI、KPIになるため、定量化でき、測定できるものでなければならない。そのため、成果はMEDICのどれかで表現できるように設定している。

- Maintained 維持
- Eliminated 廃止
- Decreased 削減
- Increased 増加
- Created 創出

上記に対して、「強化」や「改善」は、定量化しにくい成果表現であり、使用してはならない表現の代表例になる。

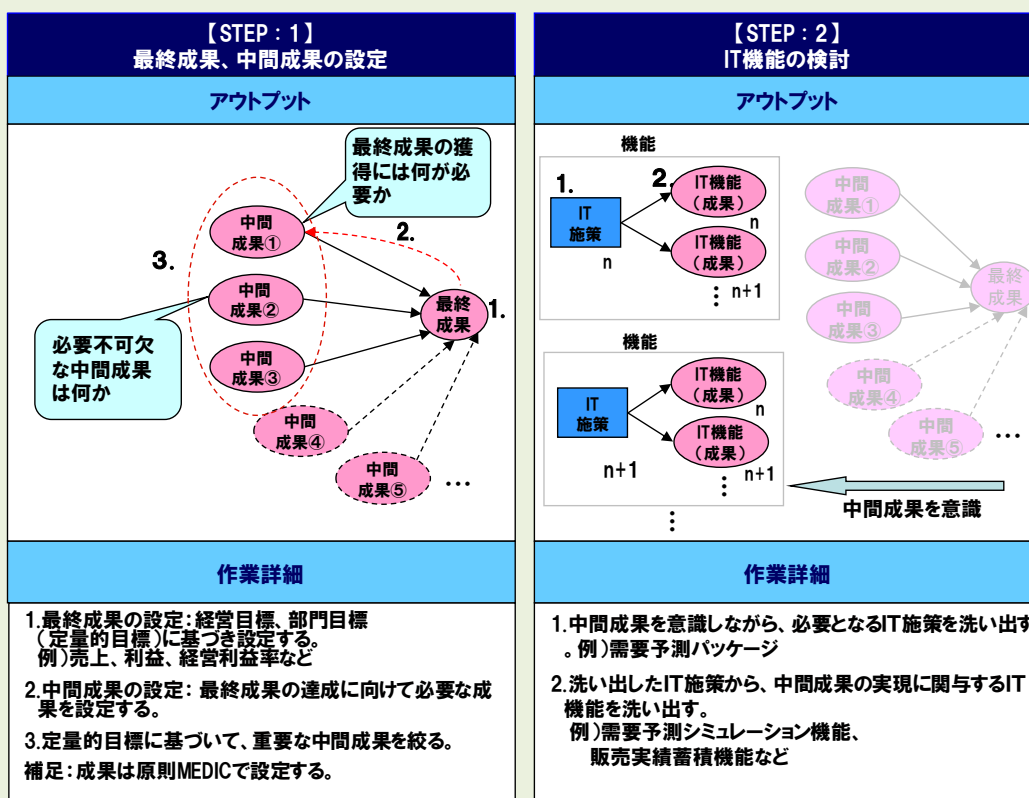


図 4.18 リザルトチェーンの作り方 (1/2)

4.2 ビジネス要求の分析(BR.2)

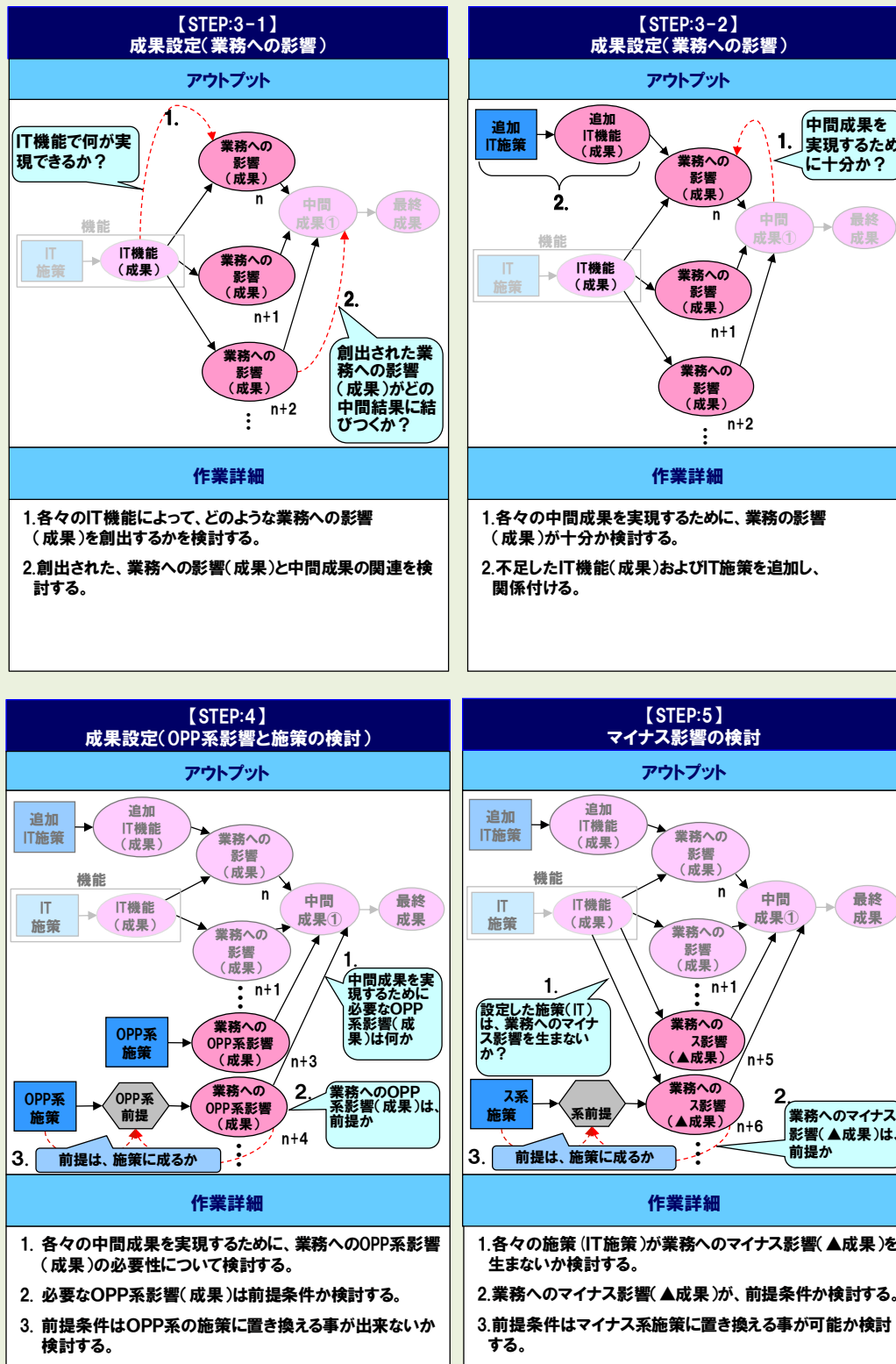


図 4.19 リザルトチェーンの作り方 (2/2)

(2) 具体的な測定尺度を検討する

【解決したい問題】

- 要求に具体性がない
 - ・ 具体的な数値目標がない
 - ・ 具体的な対策になっていない

「～を向上したい」といった要求（目的）は、それだけでは具体性がない。そのためには、いつまでに、どれぐらい向上したいのかまでを明確にする必要がある。

勘どころ① 目標に評価指標を設定する

獲得時に目的・目標も抽出されている前提で、ここでは、もう一度目標を見つめ直して要求の是正につなげることを主題として述べる。目標設定の詳細もここで記載しているので、獲得時にも参考にして欲しい。

現状値や目標値を測る物差しを「測定尺度」と呼ぶ。大切なのは何を測定尺度として設定するかである。例えば、図 4.20 に示すように、顧客満足度の向上を目的にしたとき、顧客満足度をどのように測定するかが問題になる。顧客にアンケートを出して自ら集計するのも良いだろう。しかし、タイムリーには状況が分からない。そこで、間接的にでも良いので、タイムリーに測定できる尺度がないかを検討してみる。「お客様に少しでも早く情報を提供できれば、お客様の満足度は上がるはずである」、「お客様からのクレーム数が減少すれば、顧客満足度は高くなっているはずである」と考えることが重要である。漠然としていた「顧客満足度の向上」という「経営レベルの目的」に対して、具体的な「業務目標」が見えるようになるとともに、具体的な対策（手段）の検討、是正、すなわち、より効果のある手段の見極めにもつながる。

測定尺度は、間接的にでも良いので、定量的に自らが測定可能なものを設定することが重要である。測定尺度を設定したら、達成時期と具体的な達成値を設定する。

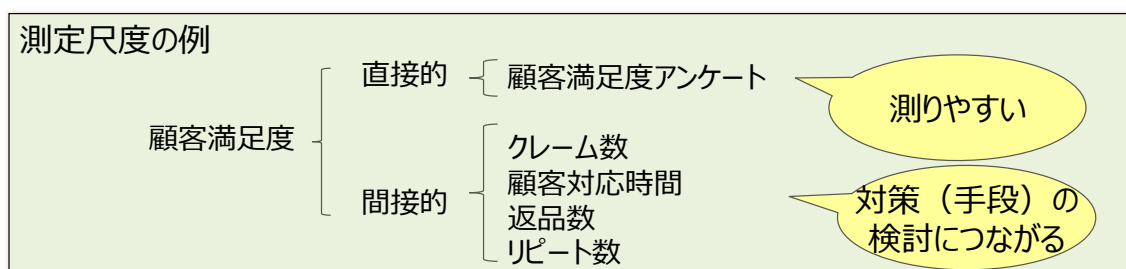


図 4.20 測定尺度の設定

4.2 ビジネス要求の分析(BR.2)

一般的によく使用されている測定尺度として KGI, KPI がある。

- KGI : Key Goal Indicator

「重要目標達成指標」と一般的には呼ばれている。本書での目的、目標の定義に合わせて目的に対する測定尺度に位置付く。

- KPI : Key Performance Indicator

「重要業績評価指標」と一般的には呼ばれているが、最近では KGI との違いを明確にするために「活動評価指標」と呼ばれることもあり、目的を達成するための具体的な活動が期待どおりに行われているかを測定するための評価指標として利用されている。本書では手段に対する測定尺度に位置付く。

4.2.2 要求の具体化(BR.2.2)

「～したい」という要求（業務レベルの目的や業務手段、IT 手段）を具体化・現実化するための新しい業務の姿を描くことを「要求の具体化」と表現することにする。業務の姿を描くためには業務モデルを利用する。業務モデルは大きく、データ構造のモデル、プロセスのモデル、相互作用のモデルの 3 つがある。この 3 つのモデルを使って要求を反映した業務を描いていくことを本項では述べる。

本項は、要求が抽出されていることを前提として記載しているが、業務を描きながら、新たな課題や要求が発生したりするので、進め方はスパイラル的なものになる。

近年、システム企画や構想立案時に、「柔軟で変化にスピーディに対応できるシステムを目指す」というシステム化方針が挙げられているのをよく見る。しかし、要件定義書になると、柔軟なシステムに対する施策として、「SOA、BPM、BRMS、クラウド、マイクロサービスなどの導入を行う」という仕掛けだけしか書かれていないものがある。確かにこれらは、システムの変更を容易にできる仕掛けとして有効である。しかし、柔軟性が欠けている根本原因は業務そのものが複雑であることが多い。それにもかかわらず業務の複雑さを積極的に減らす施策が書かれていないことがある。業務自体の複雑さを見直すのも要件定義工程の重要なタスク（要求）であるのでここで述べる。

- そもそも業務自体が複雑である
商品の種類や契約の仕方などの業務処理のパターンがたくさんある。その中には、めったに発生しないものも少なくない。一般的な企業では、上位 2 割のパターンで業務量の 8 割をカバーできているという。似て非なる業務も多く存在している。もっと業務自体を標準化し、シンプルにして、システムの大規模化、複雑化を根本的に削減することも考えなければならない。
- 昔からのしがらみで、直せない
例えば、EDI 区分が昔からあり、今の業務では使用していないが、削除するとシステムがエラーを起こすことがあるかもしれないので残しておこうと決めてしまう。システムが停止してしまうのは問題だか、このようなゴミと思われるものは削除するという英断も必要だ。
- 贅沢システムが故の硬直化
例えば、入力した項目によって、以降の入力項目が規制されたり変化したりする。入力後のチェックでも問題は起こらないのでは。他にも人間だからできるような処理、例えば、リスク判断を行うため、ノウハウを可視化し、それをロジック化しアラートを出させる、といったような機能をプログラムロジックとして実装する

4.2 ビジネス要求の分析(BR.2)

など。後者は昨今の AI などを実現できるようになっている。これらは、素晴らしいシステムかもしれないが、システムが大規模、複雑になり、修正も容易でなくなるかもしれない。もしかすると人手で対応した方がビジネスの変化にスピーディに対応できるかもしれない。

このように、システムのアーキテクチャ以前に業務自体を見直すことも重要であり、To-Be 業務を創り上げるのは、要件定義工程の重要なミッションの一つである。

(1) 情報構造の観点から業務を可視化し、新しい業務として具体化する

【解決したい問題】

- 要求を反映した業務の情報構造にならない
 - ・ 業務（情報）の整理ができてない
 - ・ 業務の複雑さを低減できない
 - ・ 業務用語を正しく理解していない
 - ・ 業務の構造を把握していない
 - ・ データモデルは業務部門には難しい

データモデルも業務を表現する業務モデルである。データモデルを作成することで、業務の見直しや新しい業務の共通認識につなげなければならない。また、データモデルで表現できる To-Be の業務ルール、すなわちシステムの仕様を明確にして設計につなげなければならない。

昨今は、情報が大量になり複雑さも増している。データモデルを使って、データすなわち情報の整理を実施することも重要である。

データモデルは、業務部門の人員では作成が難しいと思っている企業は少なくない。データモデルをシステム部門などが作成したときに、それをどのようにして「業務部門に確認するかも課題となっている。

勘どころ① 管理対象のバリエーションを整理する

業務は、管理対象の種類とその組み合わせが増えることにより複雑になる。

管理対象とは「顧客」「商品」「オーダー」などである。管理対象の種類とは、例えば「顧客」には「一般顧客」と「得意先」があり、商品には「在庫品」「オーダーメイド品」「サービス商品」があり、オーダーには「買い取り」「レンタル」があるといった具合である。そして業務ルールはこれらバリエーションの組み合わせで決まっていることが多い。「一

4.2 ビジネス要求の分析(BR.2)

般客にはオーダーメイド品は注文できない」し、「オーダーメイド品は必ず買い取りでないといけない」といったルールである。

ここでは、このような複雑さの元になっている管理対象の種類を組み合わせを可視化し、整理して、バリエーションを低減するための勘どころを紹介したい。

- 管理対象の種類の種類

整理の基本は、「同じか」「違うか」「部分か」である。

図 4.21 の a の例では、「ユーザ」と「顧客」は別であり、「ユーザ」は「利用者」と同じであり、「販売先」は「顧客」と言われるが「仕入先」は「顧客」とは言わない。「販売先」には、契約の仕方によって「得意先」と相手を特定する固有の表現をもたない他のグループの「販売先」があることを表現している。

図 4.21 の b は別企業の例である。同じ管理対象（用語）でも意味が異なることを示している。「仕入先」も「販売先」も取引のある企業はすべて「顧客」と呼称していることを表現している。図 4.21 の a と図 4.21 の b は異なる定義になっているが、どちらか一方が正解でもう一方は誤りというわけではない。これらは、各企業の独自の文化による。

よって、このような分類図を使って企業の管理対象（用語）を整理し共通認識を図れるようにすることが有効である。

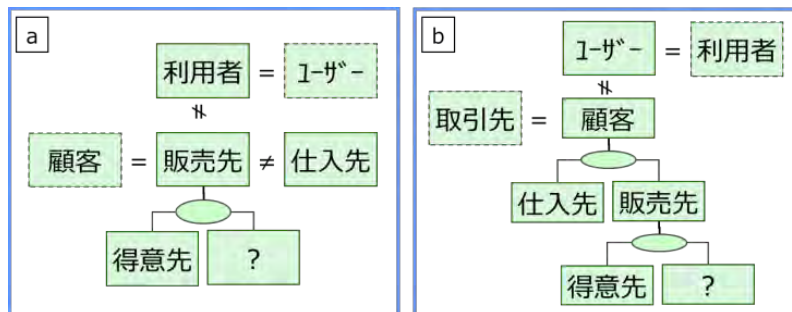


図 4.21 管理対象の分類図の例

- 漏れなくダブリなく整理せよ

ミッシー (MECE : Mutually Exclusive and Collectively Exhaustive) になるように分類する。ミッシーに分類するとは、相互に重なりがなく、全部集めて漏れがない状態にすることをいう。図 4.22 の a の図では、「取引先」を「大口」と「小口」に分類したが、実際には他の分類もあった。図 4.22 の b の図では「取引先」を「仕入先」と「販売先」と「大口」に分類したが、「大口」は「取引先」にも「仕入先」にも両方含まれていた。このような状態はミッシーになっていない。これらを正しく描くと図 4.22 の c の図のようになる。ミッシーに分類することは、正しく業務を写し取っているかであり、システムで言うと意味の違うものが同じコードに混

4.2 ビジネス要求の分析(BR.2)

在している事になり、影響範囲が広がるとか特定できないなどの柔軟性の欠如につながる。

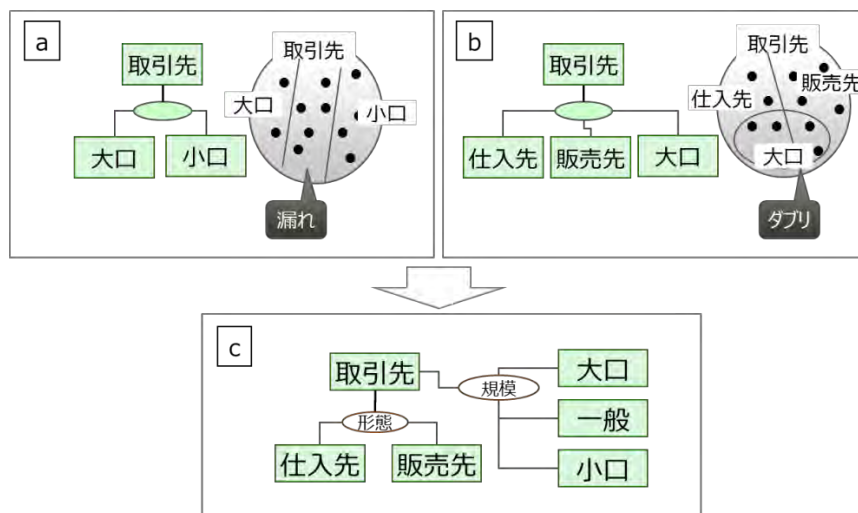


図 4.22 ミッションに分類するイメージ

- 名前のない分類に名前を付けよ

図 4.21 の a の例は前述したとおり、「販売先」には契約の仕方によって「得意先」以外に「得意先」でない固有の表現を持たないグループがあることを表している。この固有の表現を持たないグループに明確に名前をつけることが重要である。例えば「一般販売先」というように名前を付ける。こうすることにより、明確に使い分けて、仕様書を記述できるようになる。

- 用語の定義として共通認識せよ

このような整理は、用語の定義としても有効である。特に、同じ意味のものはどちらかを仕様書上で使うかを明確にしておくことと良い。図 4.21 の a の例でいうと、「顧客」という用語を使わず「販売先」という用語で統一する、同様に「ユーザ」という用語を使わず「利用者」という用語で統一する、といった具合である。使わないと決めた「顧客」や「ユーザ」は禁止用語とし、仕様書の記述に使わないよう徹底することも、仕様書のあいまいさを低減する上で有効である。

用語定義の詳細については、7.7.1 業務用語定義 (DD. 7.1) を参照していただきたい。

- 企業文化に依存するので注意せよ

図 4.21 の a, b の例のように、管理対象の整理は企業文化に依存するので、結果は企業ごとに異なる。各自の思い込みで整理しないように、確認することが重要である。また整理した結果は全員で共有する必要がある。

このような可視化、整理ができれば、次にバリエーションを削減できないかを検討する。

勘どころ② 管理対象のバリエーションを削減する

- 分類の廃止を検討せよ

図 4.23 のように、現在使用されていないバリエーション、今後使用しないバリエーションを洗い出し、削除を検討する。

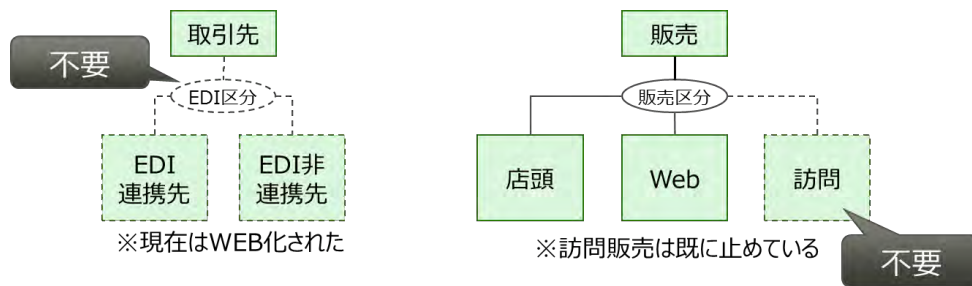


図 4.23 不要な分類を廃止する

- 分類の統合を検討せよ

他の部署や、他のシステムと異なる分類を使っており統一されていない場合がある。分類の統合が可能か検討し統合する (図 4.24)。

分類の統合を突き詰めていくと、最終的には全社統合マスタの実現に行き着くが、その実現には大変な労力を要する場合がある。そこまでできないときは、範囲を限定して統合を検討する。

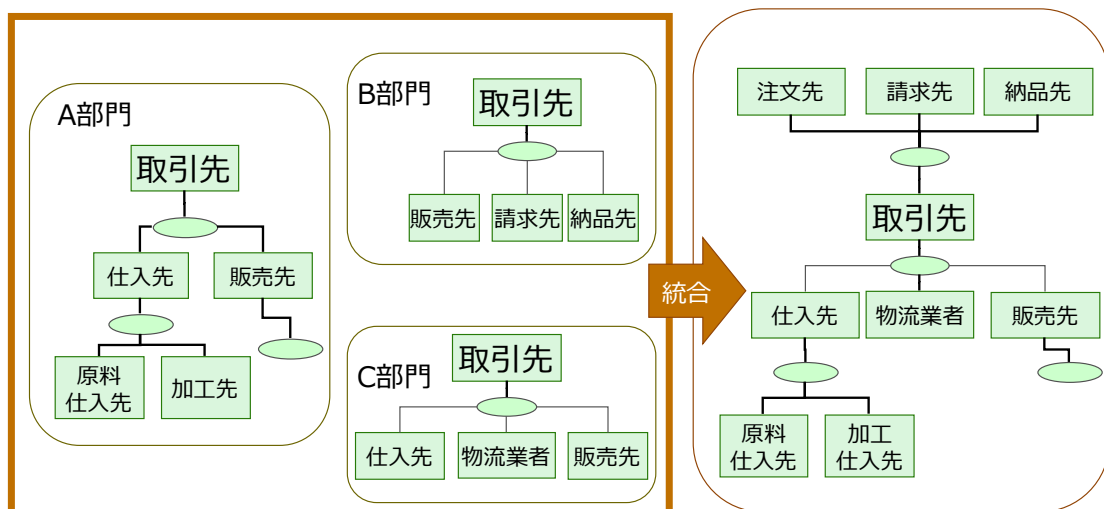


図 4.24 分類の統合の例

4.2 ビジネス要求の分析(BR.2)

業務のバリエーションを減らすための整理として記載したが、このような整理は以下のような効果も生む。

- 業務用語の理解促進、誤解低減
- 業務仕様の漏れ防止（バリエーションが明確になっている）
- バリエーションの低減による業務やシステムの標準化、シンプル化、柔軟性向上

勘どころ③ 業務ルールと照らし合わせて概念データモデルを描く

概念データモデル（ER 図）は業務を表現するモデルである。業務は、業務の管理対象を扱うものであり、業務で扱う管理対象であるデータやそのデータに関する業務ルールを調査し、エンティティとその関連として構造化した概念データモデルは、実業務の管理過程をよく表している。

概念データモデルは、受注、顧客、商品といった業務を構成する「もの」や「こと」である管理対象を表すエンティティ（Entity）、エンティティ同士の関連（Relationship）、およびエンティティを構成するデータ項目（Attribute）を図示したものであり、エンティティの意味やエンティティ間の関連（多重度など）を業務ルールとして読み取ることができる。詳細な概念データモデルの書き方は「4.3 ビジネス要求の文書化」を参照していただきたい。

従来の考え方では、ER 図はデータベース設計のための手法として基本設計以降に登場するが、要件定義段階で概念レベルの ER 図を導入して業務の実態を把握しておくこと、基本設計段階で要件定義への手戻り発生を抑止することに効果がある。

ER 図を作成することで、業務システム全体を鳥瞰でき、自然言語で要件を記述する場合に比べ、あいまいな記述や漏れを少なくし、要件定義の精度を向上させることができる。

業務ルールと概念データモデルの関係を図 4.25 に示す

4.2 ビジネス要求の分析(BR.2)

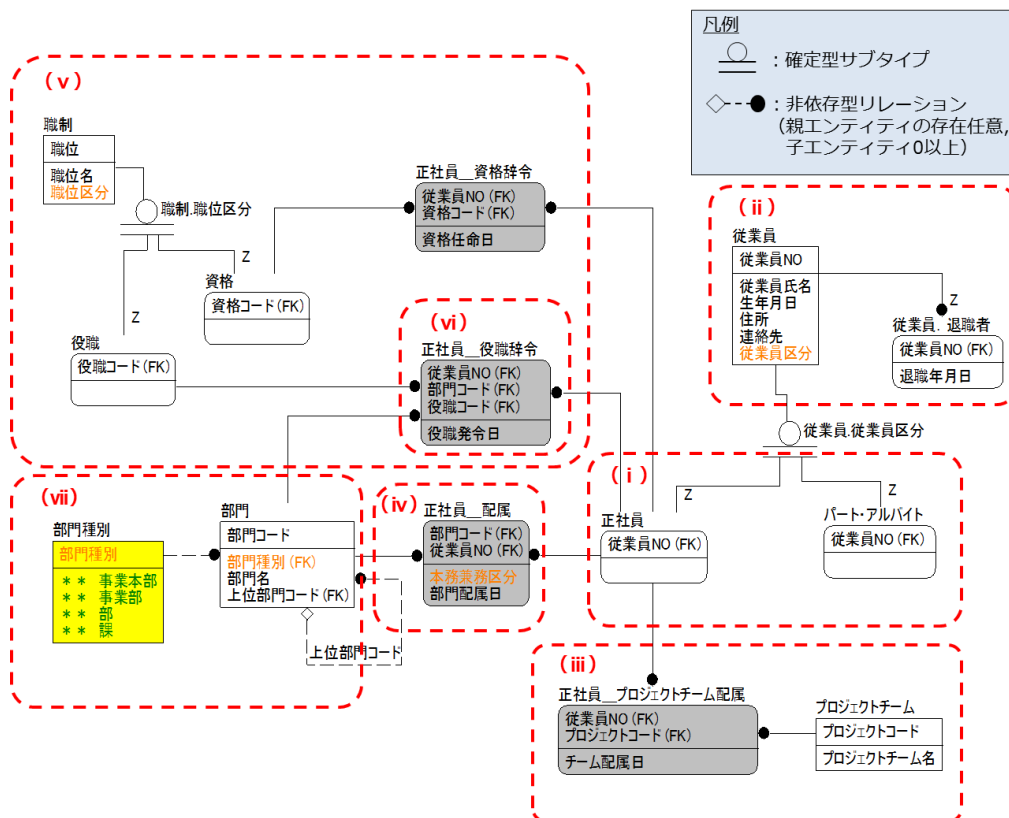


図 4.25 概念データモデルの例

- (i) 従業員には正社員とパート・アルバイトがある。正社員は職位の付与、部門への配属、プロジェクトへの配属があるが、パート・アルバイトはない。
- (ii) 従業員は現在在籍している従業員に加えて退職した従業員も範囲に含めて把握したい。
- (iii) 正社員は同時に複数のプロジェクトに従事できる。プロジェクトチームにはプロジェクトコードが付与され、プロジェクトコードでプロジェクトチームが管理されている
- (iv) 従業員は本務に加えて他部門への兼務がある。
- (v) 職位には資格と役職があり、それぞれ辞令発令がある。役職とは、部長、課長、係長等を表し、資格とは技術スタッフ、1級、2級などを意味し、基本給のベースとなる。
- (vi) ある部門では部長だが、別の部門では課長というように部長兼課長のような兼任がある。
- (vii) 社内組織 (部門) は事業部の下に部、部の下に課等の階層関係が存在する。

このようにして、整理された管理対象を用いて全体の概念データモデルを描き、抽出された要求の実現性や具体化、さらには業務的な検証を行う。

4.2 ビジネス要求の分析(BR.2)

勘どころ④ 工夫して業務部門と確認する

概念データモデルは、業務部門には作成が難しいと言われている。システム部門やベンダ企業が業務ヒアリングやレビューを行って作成し、それを業務部門に確認していく進め方がよく用いられる。ただし、作成した概念データモデルをそのままレビュー依頼しても大抵うまくはいかない。5.2.2 妥当性確認(SR.2.2)に示すような工夫をして業務部門と確認して欲しい。また、課題や要求と関連付けて確認していくのが良い。

(2) ビジネスプロセスの観点から業務を可視化し、新しい業務として具体化する

【解決したい問題】

- 要求を反映した業務プロセスにならない
 - ・ ビジネスプロセスが変わらない
 - ・ 業務プロセスを標準化できない
 - ・ 変革ポイントが分からない
 - ・ 要求が反映されているか分からない

業務を表現するモデルとして業務フローなどのプロセスフロー系のモデルがよく使われる。業務部門やシステム部門にとっても理解しやすいモデルであるが作成量が多くなる傾向にある。現状業務を中心に業務フローを作成したが、作成が終わったときには現状とそう変わりのないフローになっていたというプロジェクトも少なくない。要求を実現するためのビジネスプロセスの見直しの検討に時間が確保できていないプロジェクトもある。昨今は手作業の機械化だけではないので、ビジネスプロセスの見直し、To-Be 業務の検討に注意を払って業務フローを作成していくことが重要である。

勘どころ① 問題・課題、要求と照らし合わせてプロセスモデルを描く

企業におけるビジネスの実態を一番意識しやすいのは、業務フローなどのプロセス系ドキュメントである。しかし、フロー系成果物は精度を上げて具体的に記載すればするほど量が多くなり、作成には時間もかかるため、対象をすべて記載するところまでで精一杯になり、内容に対する議論が疎かになる傾向にある。そうならないように、作成者が経営的視点でビジネスプロセスを改革、改善するという意識を持って、業務フロー図などのプロセス系成果物の作成に臨むことが重要である。

フロー系のドキュメントの検討において使用される 3 階層のフロー図（ビジネスプロセス関連図、業務フロー、システム化業務フロー）（図 4.26）のうち、ビジネスプロセス関連図では、より経営レベルに近いビジネスプロセスの改革（要求）が表現でき、業務フロー、システム化業務フローでは、より現場レベルに近い細かいビジネスプロセスの改善（要求）が表現できる。プロセス改革の検討においては、ビジネスプロセス関連図レベルの検討から始めるのが良い。

4.2 ビジネス要求の分析(BR.2)

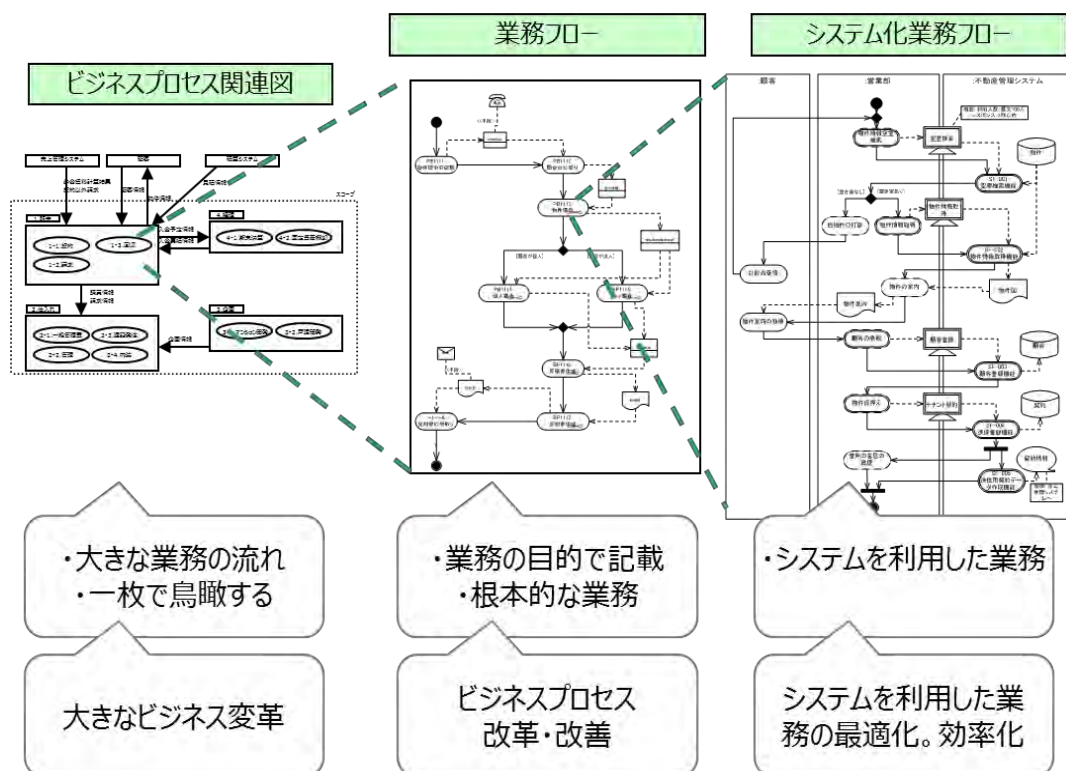


図 4.26 フロー系のドキュメントのイメージ

システム部門（またはそこから委託されたベンダ企業）がフロー図を作成すると、現行システムをベースに新たな要件への対応を追加するというアプローチで検討を行った結果、ビジネスプロセスの変革が期待したほどでなかったという結果を招くことになりがちである。それを防止するためにも、フロー系成果物は、利用部門が主体となって検討を行うことが望ましい。

勘どころ② 要求との紐付けをする

要件定義において業務フロー図を作成するプロジェクトは多数あるが、業務フロー図上に問題点・課題や要求を記載しているプロジェクトはそう多くはない。

業務フロー図上に問題点・課題や要求のマッピングを行い、現状の課題の共通認識や新しい業務の検討に業務フロー図を用いる。それとともに、前出している要求との照合を行うことが重要である。

改革・改善を意識させるための簡単かつ有効な方法は、フロー図に要求・課題の欄を設け、それらとマッピングさせながらフローの検討を進めることである（図 4.27）。

- As-Is フローを作成する
- 経営レベルの課題を As-Is フローにマッピングする

4.2 ビジネス要求の分析(BR.2)

- 現場レベルの問題点を As-Is フローにマッピングする
- 新たな問題点・課題がないか検討し、分析を補完する
- As-Is フロー図に基づいて問題点・課題の共通認識、合意形成を行う
- 経営レベルの目的・施策を To-Be フローにマッピングする
- 現場レベルの目的・手段を To-Be フローにマッピングする
- マッピングした内容をもとに To-Be フロー図を作成する
- To-Be フロー図をもとに施策、手段の共通認識、合意形成を行う

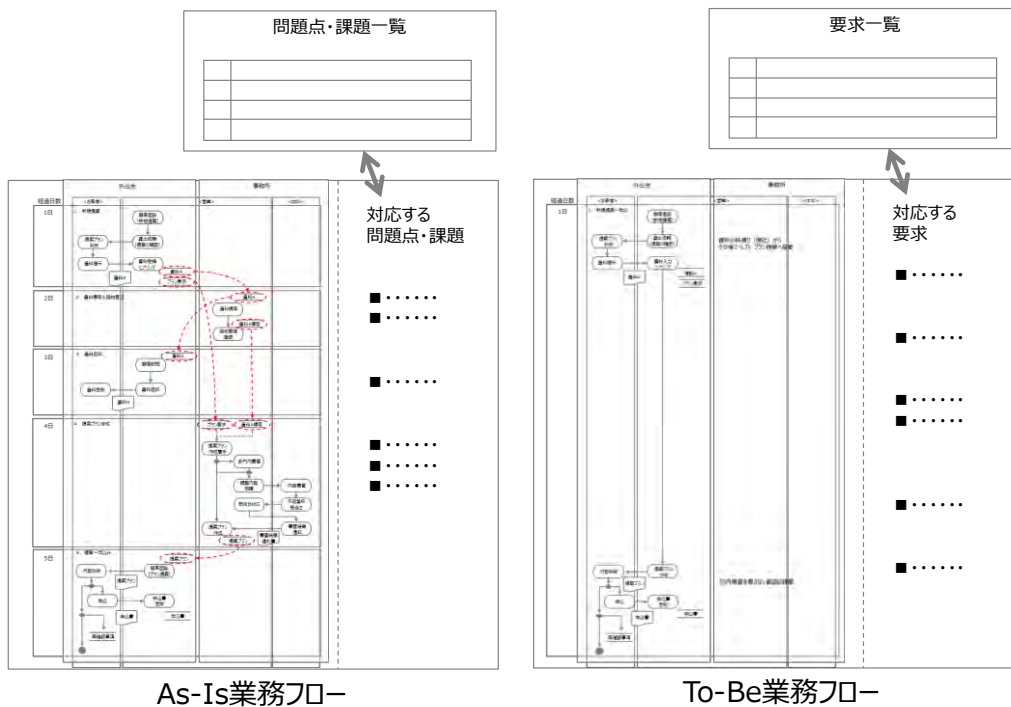


図 4.27 問題点・課題一覧、要求一覧とフロー図とのマッピングイメージ

勘どころ③ 業務プロセスのバリエーションを整理する

ここでは、業務プロセスや業務パターンを整理し、業務プロセスの観点から業務の複雑さを低減するための勘どころを述べる。業務プロセスを見直すときに常に意識しておくが良い。

- 業務プロセスの廃止を検討せよ (図 4.28 の a)
業務をシンプルにするためには、不要なプロセスをなくすことが効果的である。
「これは本当に必要なのか？要らないのでは？」という観点で検討する。システム

4.2 ビジネス要求の分析(BR.2)

化されているプロセスを人手による運用に戻すということも検討する。

- 業務プロセスの集約を検討せよ (図 4.28 の b)
似て非なる業務プロセスを共通化したり片寄せしたりして1つにする。「同じルールにできないのか？」という観点で検討する。
- 業務プロセスの連結を検討せよ (図 4.28 の c)
2つのプロセスを1つにする。「複数人で実施しているプロセスを一人に集約して実施できないのか？」という観点で検討する。プロセスが分かれているということは、各プロセスを別の人が実行しても良いということである。複数人になれば、そこにインターフェース(業務間インターフェース)が存在し、インターフェースが存在するという事は、効率を落としたり、コミュニケーションギャップを生み出したりする原因になることを意味する。これをなくそうという観点で検討するものである。
- 業務プロセスの並行化を検討せよ (図 4.28 の d)
順次行われていた業務プロセスを並行で実施できないか検討する。業務プロセスの順序は環境の変化などによって変わることがある。順序が変わっても柔軟に対応するためには、順序の依存関係がないように見直しておくことも必要である。

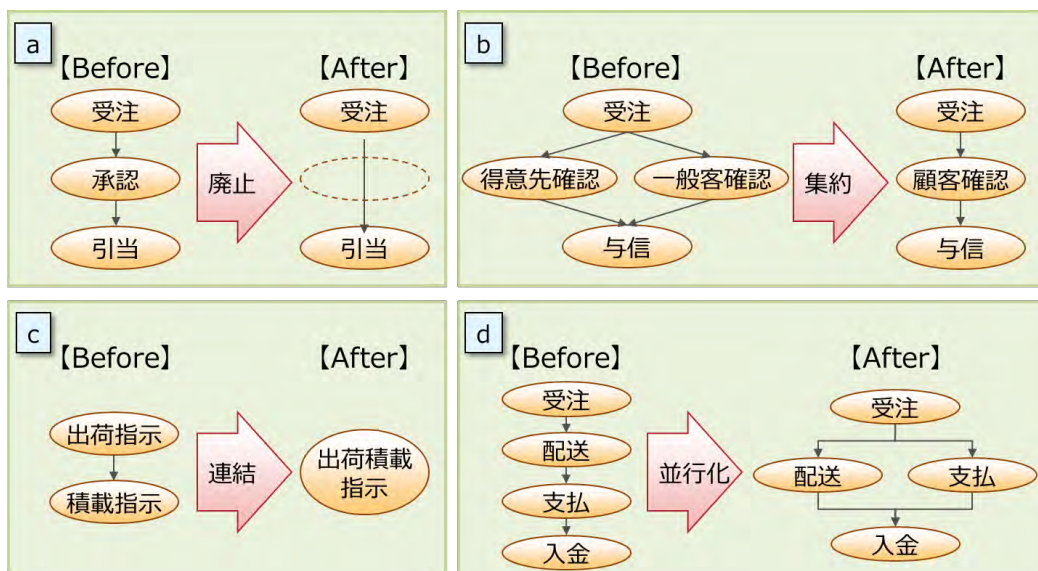


図 4.28 業務プロセスの見直しの観点

勘どころ④ 業務パッケージの適用を検討する

業務パッケージの導入は、業務の見直しや標準化を手早く実現する方法であるが、安易に考えると失敗する場合もあり、導入には注意が必要である。

欧米ではビジネスプロセス大幅見直しのためのパッケージの導入は当たり前になっている。現場からの業務改善よりもトップダウンで業務を変えていく場合が多く、その方法としてパッケージ導入が採用される。一方、日本では現場を巻き込んだ改善が多い。ここに注意が必要である。以下に注意点を記す。

- カスタマイズ量に注意せよ
失敗の原因はパッケージのカスタマイズ量が多くなることである。カスタマイズが 20%以上あると新規開発と変わらないと言われている。
- 業務をパッケージに合わせるという意識を強く持て
業務をパッケージに合わせると明確に宣言できているかが重要である。パッケージの導入目的が単なる IT システム開発の効率化のためのソース流用であると、カスタマイズ量が増え、プロジェクトは失敗する。業務をパッケージに合わせるという意識をプロジェクトメンバに周知させておかないと失敗する。
- 業務パッケージの導入に対する現場の反対をコントロールせよ
現場は業務をパッケージに合わせることに反対する。慣れ親しんだ業務が変わる、退化するかもしれない、大幅に変えると調整などが大変になるなどの思いからである。その反動から今まであった機能がなくなる事に反対し、あれも必要、これも必要となり、カスタマイズ量が増え失敗する。
- 経営トップが参画せよ
業務をパッケージに合わせるという目的での業務パッケージの導入は、今までの業務を変えるという意志表示でもある。新しいやり方に業務を変えるわけなので、経営トップが積極的に参画するか、プロジェクト責任者に業務変更権限を明確に与えないと成功しない。
- パッケージ導入の是非を再度判断せよ
パッケージ導入検討の際に Fit&Gap 分析を行い適合性を判断していても、要件定義が進むにつれて、上記のような理由で、カスタマイズ量が増える。しかし、パッケージの導入が決定しているので無理にカスタマイズするプロジェクトも多い。要件定義工程で新たな要求が明確になった時点で、もう一度 Fit&Gap 分析を行い、パッケージ導入が適切であるかを判断し直すことが重要である。

注意は必要だが、業務を大幅に見直したい、標準化したいというときには、業務パッケージの導入は有効な手段である。

4.2 ビジネス要求の分析(BR.2)

(3) 情報とプロセスの両方の観点から業務を可視化し整合性を確認する

【解決したい問題】

- 情報と業務プロセスの関係が複雑で不整合が発生している
 - ・ 情報と業務プロセスの整合性がとれていない
 - ・ 業務パターンが多い

概念データモデルはデータ（情報）とその関係から業務を表現すのに対して、業務フローなどはプロセスとその流れの観点から業務を表現する。各々が業務を異なる角度から写像している。これらのモデルを両方の観点からモデル化するのが相互作用のモデルである。概念データモデルと業務フローを別々に作成した場合にはその整合性の確認が不十分になることがある。そんなときに、相互作用のモデルを作成して検証することができる。また、相互作用のモデルの方が業務を分かりやすく表現できることもある。

勘どころ① 相互作用のモデルを描き、情報、業務プロセスの両面から業務の実現性を確認する

振る舞いのモデルとは、状態遷移図や CRUD 図に代表されるモデルである。管理対象（エンティティ）のライフサイクルを表現したもので、業務上のイベント（プロセス）により、エンティティがある状態から別の状態に変化することを明確にしたものである。このモデルの作成によりエンティティと業務プロセスの関係が明確になり、双方に矛盾なく業務が遂行されるかどうかを検証できる。図 4.29 に状態遷移図の例を示す。

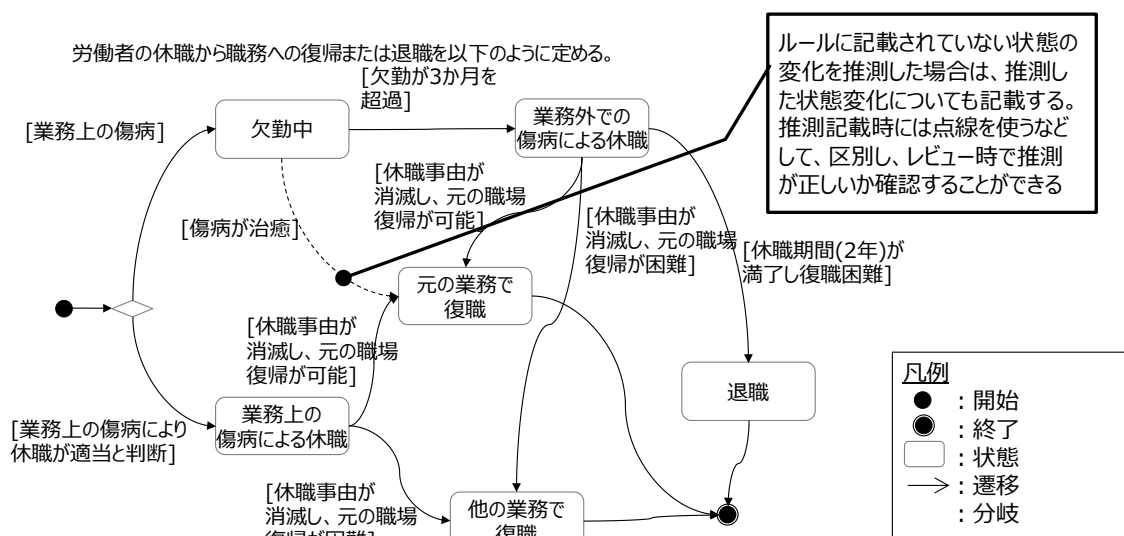


図 4.29 状態遷移図のイメージ

4.2 ビジネス要求の分析(BR.2)

状態遷移図では、どの状態がどのような条件で変化するかが読み取りやすく、状態が変化する条件を確認すれば、条件分岐の誤りや漏れを発見できる。

勘どころ② 業務パターンを削減する

似て非なる業務を標準化したい、幹となる業務と枝葉である業務を切り分けたいといったニーズも多くある。これは、業務がシンプルになるだけでなく、業務プロセスの重複の低減や効率の向上にもつながる。前述の管理対象の種類組み合わせ等により、多くの業務パターンが存在することになる。このパターンの上位 20%で実際に発生する業務の 80%をカバーできるといわれている。しかし、システム開発では利用する頻度が少ないものでも同等に開発しなければならない。上位 20%のパターンよりも、利用頻度の少ない残りの 80%のパターンの方が、開発難易度が高く品質確保が困難な業務であることも多い。したがって、業務プロセスを標準化したりバリエーションを減らしたりすることは、システムのシンプル化、規模削減に大いに効果がある。

概念データモデルや業務フローを作成した後に、どのような経路を辿るパターンがあるのかを明確にすることは、業務を検証する上でも、運用テストのシナリオを作成する上でも有効である。この一連の業務処理のパターンを削減できないか検討する。図 4.30 のような表で、どのような時にどのプロセスを辿るのかをトレースしていく。そうすると、似たようなパターンが見えてきて、前述した観点で業務プロセスの見直しを行い、パターンを削減していくことができるか検討しやすくなる。各パターンの業務上の発生頻度や重要度などからパターンそのものを止められないかを検討する。滅多に発生しないパターンの統合や、業務運用によるカバーを検討する。この分析は業務全体を鳥瞰し、業務の大きな流れを一つのパターンとして整理するものであり、パターンの統合や削減による業務の標準化やシンプル化の効果が高い。

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	
...

図 4.30 一連の業務処理のパターンの可視化

4.2 ビジネス要求の分析(BR.2)

4.2.3 優先順位付け(BR.2.3)

要件定義において抽出されたすべての要求(手段)を実現することができるプロジェクトは稀である。膨らむ要求を限られた工期やコストの範囲に抑え込むためにも、要求には優先順位をつけて、効果の薄い要求や贅沢な要求は捨てる選択をしなければならない。また、昨今では、リーンスタート的な発想で、最低限必要な機能でまずリリースして使用してもらい、実績を評価して徐々に成長させていく(もしくは止める)手法のシステム開発を選択する場合もある。いずれにしても、どの要求は先送りできるのかを判断できるようにしておく必要がある。

(1) 優先順位を明確にする

【解決したい問題】

- 限られた工期やコストの中で要求を絞り切れない
 - ・ 優先順位付けができていない
 - ・ 絞り込み時ステークホルダの合意がとれない
 - ・ すべての要求を正確に判断するのに時間を要する

要求は取捨選択しなければならない。要求が実現できないことの説明や、ステークホルダ間での調整に苦労しているプロジェクトも少なくない。要求の選択は、できる限り客観的な評価指標を用いて実施したことを説明できるようにしておくことが望ましい。

勘どころ① 客観的な判断基準を明確に定義する

一般的には、優先順位をつけるためには以下のような判断基準が用いられる。優先度はその総合評価により決定される。

有効性：目的、目標にどれだけ貢献するか(達成効果)

必要性：法制度対応、内部統制、社会的責任、将来性などの観点で必要か

緊急性：急を要するかどうか

費用：実現するのにどれくらい費用がかかるか

実現性：使用する技術や人材で本当に実現できるのか(技術実現性、人的実現性)

新たな問題：この手段を実現したときに発生する新たな問題はないか

- 有効性
 - 目的に対する効果は重要な指標である。定量的効果、定性的効果の観点でそれぞれ

4.2 ビジネス要求の分析(BR.2)

の要求を評価すると良い。定量的効果については、目標の数値だけでなく、その数値の算出方法、算出根拠も必要である。

- 必要性
目的外であったり、効果が低くても実施しなければならないものもある。法律が変わるので対応しなければならない、セキュリティ事故が起こったので対策を講じなければならない、社会的責任や将来のことを考えたら実施した方が良い、などを考慮する必要がある。
- 緊急性
上記の「必要性」があるもの、ビジネス上の施策の納期が明確で逼迫しているもの。例えば、4月1日に法律が変わるのでそれまでに対応が必要、物流倉庫の運用が4月1日に決まっているなど、その達成期限により優先度が高くなる。
- 費用
費用対効果の観点で評価するために、「有効性」だけでなく実現にかかる費用、運用にかかる費用などを評価する必要がある。
- 実現性
技術的实现性と人的実現性の2つの実現性で評価する。技術的实现性とは、例えば実践事例も少ないような先進技術の適用などのリスクがある場合や、本当にこの技術を適用して期待していることが実現できるのかというリスクに対する評価である。人的実現性とは、実行できる人材が育っていない、集められないといったリスクに対する評価である。先進技術の適用時であっても、すでに確立された技術の適用時であっても、このリスクは技術的实现性とは別に評価する。
外部動向、内部動向などを踏まえて要求の実現性を評価し、リスクが高ければスコープから外すという選択もプロジェクトを成功に導くための重要なリスク管理である。また、ベンダ企業の知見などを活用し、技術的な観点で実現性を評価することも重要である。
- 新たな問題
新たな手段を実現する際には、その手段を実現したことにより現状の業務には存在しない新たな問題が発生することがある。これは、実現検討の際にあわせて検討すべき事項である。新たな問題の発生を検討し、発生するならその対処方法も検討する。それらを含めて要求実現の費用や実現性を検討する必要がある。

勘どころ② 大きな判断と詳細な判断を使い分ける

多くの要求の優先度を前述の判断基準に基づいて調査し、優先度を決定するには多大な時間を要する。

4.2 ビジネス要求の分析(BR.2)

効率的な進め方の一つの方法を示す。例えば、有効性という指標に対し、「効果が大きい／効果あり／効果が少ない」の3段階で判断する。すなわち、本当の効果を費用想定で見積もるのではなく、ある程度 of 感覚や複数人の合意で大まかに決める。このようにして要求全体の優先度を仮決めしてみても、この段階で要求の取捨選択を行う。そして、取捨選択の境界線上にあって判断に迷うものは、詳細に調査・分析を実施し正確なデータをもとに判断するという方法である。

要求の取捨選択のための全体評価には、MoSCoW 分析が参考になる。優先順位の判断基準を参考に、取捨選択を以下の4つで付けるものである。前述判断基準をベースに、総合得点を計算し客観的、機械的に採点するか、関係者で集まって主観的に評価するかなど、実施方法を検討する必要もある。

- M(Must) : 必須 (これが実現しなければ目的を達成できない要求)
- S(Should) : 推奨 (目的を達成するために必須ではないが重要な要求である)
- C(Could) : できれば (あれば良いレベルの要求)
- W(Won't) : 不要

コラム 6 要求の優先順位付けと合意形成

【取り組みの背景】

ステークホルダの多様化により提示される要求が膨らみ、システム開発に支障をきたすようになった。要求を絞り込むためには客観的な評価に基づく合意形成が必要であることから、個々の要求を評価するための客観的かつ総合的な指標を設定し、評価の結果に基づいて優先順位を付けることが求められた。

【課題】

複数のステークホルダの要求を絞り込むための合意を得るのが難しい。

- 声の大きい人の意見に流される。
- 要求の評価観点や基準が明確でない。
- ステークホルダによって重要視するポイントが異なる。
- 個々の要求を客観的に（数値で）評価する指標がない。

【解決策】

客観的かつ総合的に、要求に優先順位を付けた取り組み例を以下に示す。

(1) 総合的な評価観点の設定

昨今のシステムは人間の手作業の機械化（効率化）だけではなく、経営や業務への貢献度やコスト、環境・社会への貢献度など多様な観点からの評価が求められている。

図 4.31 に評価観点の例を示す。

必要性	有効性	実現性	コスト
①経営施策 部門施策の実現	①売上向上	①技術実現性	①開発費用
②老朽化リスク・ 障害リスクへの対応	②顧客満足度向上	②人的実現性	②設備費用 など
③法制度 内部統制 社会的責任 など	③情報共有・ 情報精度の向上		
	④コスト削減/効率化		
④他案件との関連度	⑤品質向上		
⑤代替手段の有無	⑥安全性/リスク回避		
⑥将来性	⑦人材育成		
	⑧環境・社会貢献		

図 4.31 評価観点の例

4.2 ビジネス要求の分析(BR.2)

(2) 評価観点に得点を設定する

各々の評価観点に対して評価目安とデジタルな指標（得点）を設定する。

「有効性」の「①売上向上」の例を図 4.32 に示す。

有効性		
①売上向上	売上向上が定量的に示されており直接的な効果が高い	5
	売上向上がある程度見込まれる。または間接的（収益力強化に向けた分析・検討など）に効果が見込まれる	3
	明確な売上向上効果は薄い	0

図 4.32 評価観点に目安と指標を設定した例

この例のポイントを以下に示す。

- 5点満点である
- 得点が高いほど良い（優先順位が高くなる）
よって、コストは低いほど得点は高くなるように設定する
- 目安を3つ用意している
- 1点、2点、4点などの中間をつけても良い

中間の得点にあえて目安を用意しないことにより、目安と多少異なる状況を主観的に判断してもらう。正確性は低下するが、指標の設定負荷軽減と指標どおりでないことによる問い合わせ等が減る。すべての観点を5点満点で設定しているが、例えば、重要項目は10点満点とするなど、重み付けされた観点があっても面白いかもしれない。

(3) 客観的な評価

客観的な評価を行うために、要求一覧に得点をマッピングしている（図 4.33）。

一覧にマッピングすることで、横並びで評価できるとともに、デジタルな得点をつけていることにより客観的な評価ができる。

【要求一覧 優先順位の判断】 更新日: 20nn年nn月nn日

要求ID	要求名	要求概要			優先順位	(小計) 必要性	(小計) 有効性	(小計) 実現性	(小計) コスト	必要性の評価							有効性の評価							実現性		コスト							
		経営目的	業務目的	実現手段						採否	経営施策部門の実現	老朽化リスク・障害リスクへの対応	法制度内部統制社会的責任など	他案件との関連性	代替手段の有無	将来性	売上向上	顧客満足度向上	コスト削減/効率化	品質向上	安全性/リスク回避	環境・社会貢献	情報共有・精度の向上	人材育成	技術実現性	人的実現性	開発費用	設備費用など					
001																																	
002																																	
003																																	
004																																	
005																																	
006																																	

図 4.33 要求一覧に評価をマッピングした例

この例のポイントを以下に示す。

- すべての評価観点と得点を記載している
- カテゴリ別の小計を記載している
- 要求の優先度と採否区分を設けている

特筆すべきことは、優先度だけでなく、採否区分を設けていることである。採否区分とは、今回取り下げた要求にマークを付けるために使用するものである。採否区分を使用して、優先度の高低だけでなく、実現しない、あるいは次期に見送るという判断をさせることも要求の評価においては必要になる。

また、評価をするときは、合計得点だけでなく、カテゴリごとの小計を参考にして判断するなど柔軟に対応する。

(4) 合意形成

ステークホルダ、システム開発関係者や第三者などを集めて、上記の一覧を使用した要求の優先順位付けおよび取捨選択結果のレビュー・合意形成を行う。個々の要求の評価は、第三者が要求の内容を詳細に把握することが極めて困難であることから、要求の提示元が実施する。

4.2 ビジネス要求の分析(BR.2)

(2) 要求を棄却する

【解決したい問題】

- 優先順位を付けても要求を絞り切れない
 - ・ 要求を捨てられない

膨らんでしまった要求のすべてに対して優先順位を付けて棄却を判断していくのには時間を要する。優先順位を付ける前に、要求の体系化などの時点で判断できるものもある。要求が膨らみがちになったとき、常に要求を捨てられないか、先送りにすべきでないかを意識して総量を抑え込む必要がある。

勘どころ① 要求を捨てる

要求が膨らむことでシステム開発が失敗することが多々ある。膨らむ要求は抑え込む必要がある。その際には、単に優先順位をつけるだけで終わらせず、要求を捨てる努力をする必要がある。

一昔前の情報システムに対する評価は、少しでも多くの機能をシステム化することであつたり、システムでは難しいとされる人間の判断やノウハウ機能などもシステムの機能として実現することであつたりしたが、昨今は変化してきている。情報システムは環境変化に俊敏に対応できることが望まれていて、どちらかというシンプルでスリムなものが良いという評価に変わってきている。難しいものは逆に人が運用で対応する方がコストも低減するし、対応力も上がるという発想である。

これらを考慮し、前述した優先順位をもとに要求を捨てる努力をする必要がある。「断捨離」の気持ちで要求を捨てる英断をしていただきたい。

勘どころ② 要求を先送りにする

リーンスタートアップに見られるように、初めからすべての要求を実現した 100 点のシステムを作成せずに、60 点の評価が得られる程度の必要最低限の機能だけでシステムをリリースし、実際に使用してもらってから徐々に機能を成長させていく方が良いものもある。要求を捨てるのではなく、先送りして、運用後に対応しようという発想も必要である。

4.2.4 要求の交渉(BR.2.4)

要求内容について合意形成する、要求を取捨選択するための合意形成を行う、新しい業務の合意形成を行うなどさまざまな場面で合意形成のシーンがある。要件定義は合意形成プロセスと言われるだけあり、難しく時間もかかる。この要求の交渉に対するプロセスの重要性を意識してプロジェクトを進める必要がある。

(1) 経営層や業務部門への説明を行う

【解決したい問題】

- 経営層、事業部門長などにうまく説明ができない
 - ・ プロジェクト成果物をそのまま見せてもわかってもらえない
 - ・ 説明資料作成のタスクもなく、担当者もアサインされていない

経営層と合意形成を行うときは、経営層用の資料を作成しなければならない。それらの資料はシステム開発につなげる資料とは異なるものになることが多い。例えば、作成した業務フローを使って新しく変わる業務を説明しようとしても、量が多い、ポイントが分からないなどと言われ新たな説明資料を作成することになる、などである。業務を表現するドキュメントやシステム開発につながるドキュメントの作成には担当者の割り振りやスケジュール化がなされているが、経営層に説明する資料の作成にはこれらがなされていないというプロジェクトがある。経営層への説明資料作成も要件定義の重要な仕事である。

勘どころ① 説明相手の視点にあった資料を準備する

- 経営者向け

経営層に説明するとき、A4 もしくは A3 用紙 1 枚で説明資料を作成した経験のある人は少なくないはずである。経営層は、今までと何が違って、何が良くなるのか、そしてそれに対する投資対効果を、端的に知りたいというのが一般的である。

これらを表現したドキュメントの例を図 4.34 に示す。これは、「ビフォーアフター図 (B/A 図)」と言い、現行とこれからの違いを中心に 1 枚の紙で描いたものである。

4.2 ビジネス要求の分析(BR.2)

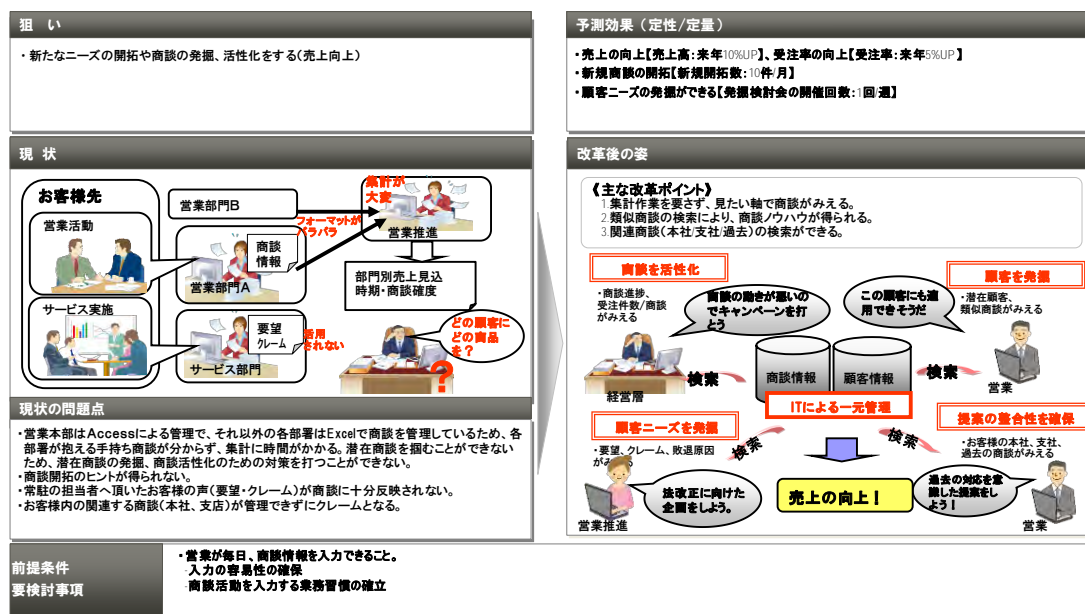


図 4.34 ビフォーアフター図の例

この例の B/A 図はテーマごとに描かれているが、すべてのテーマを説明しなくても良い。3 つぐらいの主要なテーマを選んで説明すると経営層には効果的である。そうすると、経営者に今回のプロジェクトの価値を端的・的確に説明することができる。

一方、投資対効果の説明、すなわち経営者による投資の判断は正確に実施すべきである。「6.3.7 見積り作成・投資効果分析」を参照していただきたい。要件定義の最大の目的は、このままシステム開発に着手させるかまたは中止させるかを正しく判断できる材料を経営者に提供することである。経営者説明用資料の作成を検討し、要件定義のタスクとして明確にして実施するべきである。

● 業務部門長向け

業務部門長は要件定義に主体として参加していない業務部門の人に対しても新しい業務を説明し、それらの人々と合意形成を行う必要がある。業務部門の最大の関心事は、新しい業務プロセスがどのようになるのか、システムがどのようにサポートしてくれるのかであり、業務フローで説明するのが理解を得やすい。だからと行って作成した業務フローをすべて説明するのは非効率である。業務フローは意外と量が多い。業務部門も経営層と同様に、関心事は何が変わって、何が良くなるのかである。作成した業務フローを用いて、As-Is 業務フローであれば問題や課題のありかを中心に、To-Be 業務フローであれば要求によって変わるところを中心に説明する。そのためにも業務フローには、問題点・課題や要求の紐付けをしておくべきである。

勘どころ② 説明資料作成の計画を立てる

設計工程につながるドキュメントは、担当者を決めてスケジュールを作成して計画的に作成されることが必要である。次工程の担当者への説明資料は、作成者やスケジュールなどが不明確になっているプロジェクトが多い。担当者を明確にしてスケジュール化しておく必要がある。要件定義は合意形成が重要なので合意形成のためのプロセスも含めて計画する必要がある。

4.2 ビジネス要求の分析(BR.2)

(2) 要求の絞込み交渉と合意形成は、ポイントを意識して行う

【解決したい問題】

- 要求の絞込み交渉がうまくいかない
 - ・ 第三者からの説明に納得感が得られない
 - ・ 声の大きい人に負ける
 - ・ 衝突する要求を捌けない

要求の絞込みに対する交渉や合意形成は意外と難しい。交渉や合意形成先であるステークホルダーがプロジェクトのメンバであるとは限らない。要件定義においては、要件定義の担当者が課題の抽出やヒアリングの時から現場部門と同じ当事者意識を持ち、プロジェクトに対する理解を深め、合意形成の場に臨むことが重要である。

勘どころ① 当事者意識を持たせる

要求の源泉として、問題、ニーズ、課題を分析するべきであると前述した。プロジェクトメンバ全員が我が事として要求を認識し、納得し、自分の言葉でプロジェクト外の人に説明する責任を持っている。プロジェクト参画者はこれから新しい業務やシステムをプロジェクト外の人に展開していく中枢になるはずである。展開の際には、新システムを利用する業務担当者から「どうしてこのような業務の流れになっているのか?」「どうしてこのような機能が必要なのか」質問を受けるであろう。その際、「××さんが必要だと言ったから」と答えたら、業務担当者は新システムに不満を持ち続けるだろう。

そうならないようにするため、プロジェクト参画者がそれぞれ考えを出し合い、質問し合い、議論し合うことにより、相互理解を深めて共通認識を形成するプロセスを経て、一人称で問題、ニーズ、課題を扱えるようになる必要がある。

一定の時間を確保してワークショップを開催し、以下のようなことを行うことにより、相互理解や共通認識構築をしやすくする。

- 問題、ニーズのブレーンストーミング
- 模造紙と付箋を使った問題、ニーズの深掘り、課題の抽出
- 模造紙と付箋を使った目的、手段の構造化、詳細化

ワークショップは、関係者が1つの場所に集まって議論を展開する場であり、ここに参加すること、参加させることがプロジェクトへの当事者意識、責任意識を醸成させる第一歩になる。

勘どころ② 声の大きい人に負けてはならない

4.2 ビジネス要求の分析(BR.2)

要求の取捨選択をする際、声の大きい人の意見に流される傾向がある。また当然だが、企業全体より自組織のことを優先しがちである。そのために前述の優先順位付けで述べた判断基準を作成している。この基準は客観的な指標であり、この指標を利用して大きな声に負けないようにする。

また、議論の進行役であるファシリテータの役割も重要である。ファシリテータは各メンバーの発言時間をコントロールしたり、全員に均等に意見を聞いたりして議論をコントロールする役目を担う。ステークホルダの洗い出し時に各ステークホルダの特徴を事前に把握しておくことも必要である。ファシリテータは客観的な立場であり、それを駆使して議論をコントロールしなければならない。

勘どころ③ セレモニーでないレビューを行う

成果物を表面的にレビューして合意書に署名するなどの形式的なレビューには意味がない。レビューは、要求が自分たちの共通認識に適正に反映されているかどうかを関係者全員で確認する最終イベントである。

例えば業務フローのレビューでは、関係者全員が集まってそれぞれが業務担当者となつて新しい業務フローに沿った業務をシミュレーションすることにより、問題が解決できるのか、ニーズを満たすことができるのか、または、例外的な業務もこのフローで対応できるのかなどを確認する必要がある。

レビューは要求にエラーがあるかどうかを確認する場であり、エラーの解決策を検討する場所ではないため、できるだけエラー摘出に集中することが重要である。

勘どころ④ ステアリングコミッティを用意する

上位層の判断を仰ぐ必要がある事項や、当事者間の意見の衝突が発生して合意形成に時間がかかるケースが往々にしてある。そのような場合には、上位層に意思決定を委ねて、その結果をプロジェクトで共有したほうが早い。その準備として、プロジェクト計画時に、意思決定を行う上位層の責任者の任命と意思決定者へのエスカレーションパス、決定機関（ステアリングコミッティ）の設置と関係者への事前合意を行っておくことが望ましい。

意志決定機関への参加者の例を以下に示す。

- プロジェクトオーナー
- プロジェクトスポンサ
- 業務部門の責任者
- システム部門の責任者
- ベンダの責任者

4.2 ビジネス要求の分析(BR.2)

意志決定機関は、定期開催に加えて重大な課題が発生した場合や緊急を要する場合に
関係する責任者のみを招集して実施する臨時開催もできるようにしておくが良い。

4.3 ビジネス要求の文書化(BR.3)

4.3.1 ビジネス要求の文書化(BR.3.1)

ビジネス要求を表現するドキュメントや課題や問題を分析するドキュメント、その要求を実現するための業務モデルなど、ビジネス要求定義で使用するドキュメントを作成することが必要になる。これらのドキュメントはビジネス要求の獲得やビジネス要求の分析で作成し、使用していく。その際、部分的に記載していたもの、重点を絞って議論されていたものがあるなら、最後に、正確かつ漏れなく不足部分を記載し、ドキュメントを完成させるプロセスが必要になる。

ここでは、ビジネス要求定義のドキュメントの全体像を示す。

ドキュメントをどのような記述フォーマットで何に注意して記載していけばいいのかなど、個々のドキュメントの作成についての勘どころは、システム化要求定義も含めて7章にまとめて提示する。

(1) ドキュメントの全体像、関連を明確にする

【解決したい問題】

- 要件定義で作成するドキュメントの全体像を理解していない
 - ・ ドキュメント間での不整合が発生している

要件定義に使用されるドキュメントの全体像やドキュメント相互の関係性が明確でないため、ドキュメント間の不整合が発生している。このドキュメントのこの項目が次のこの項目と一致して、詳細はこちらのドキュメントに記載するなど、ドキュメントの全体像や関係を理解していないと不整合や標準化が徹底できない。

勘どころ① 成果物相互の関連性を明確にする

「BR. ビジネス要求定義」で作成する主要ドキュメントとその関係を図 4.35 に示す。

なお、プロジェクトにおいて、「BR. ビジネス要求定義」で作成するドキュメントを計画する際には、「RM. 2.1 プロセス計画」に記載された勘どころを参照し、作成ドキュメントの取舍選択、追加、ならびに記述粒度を定義することが必要になる。

4.3 ビジネス要求の文書化(BR.3)

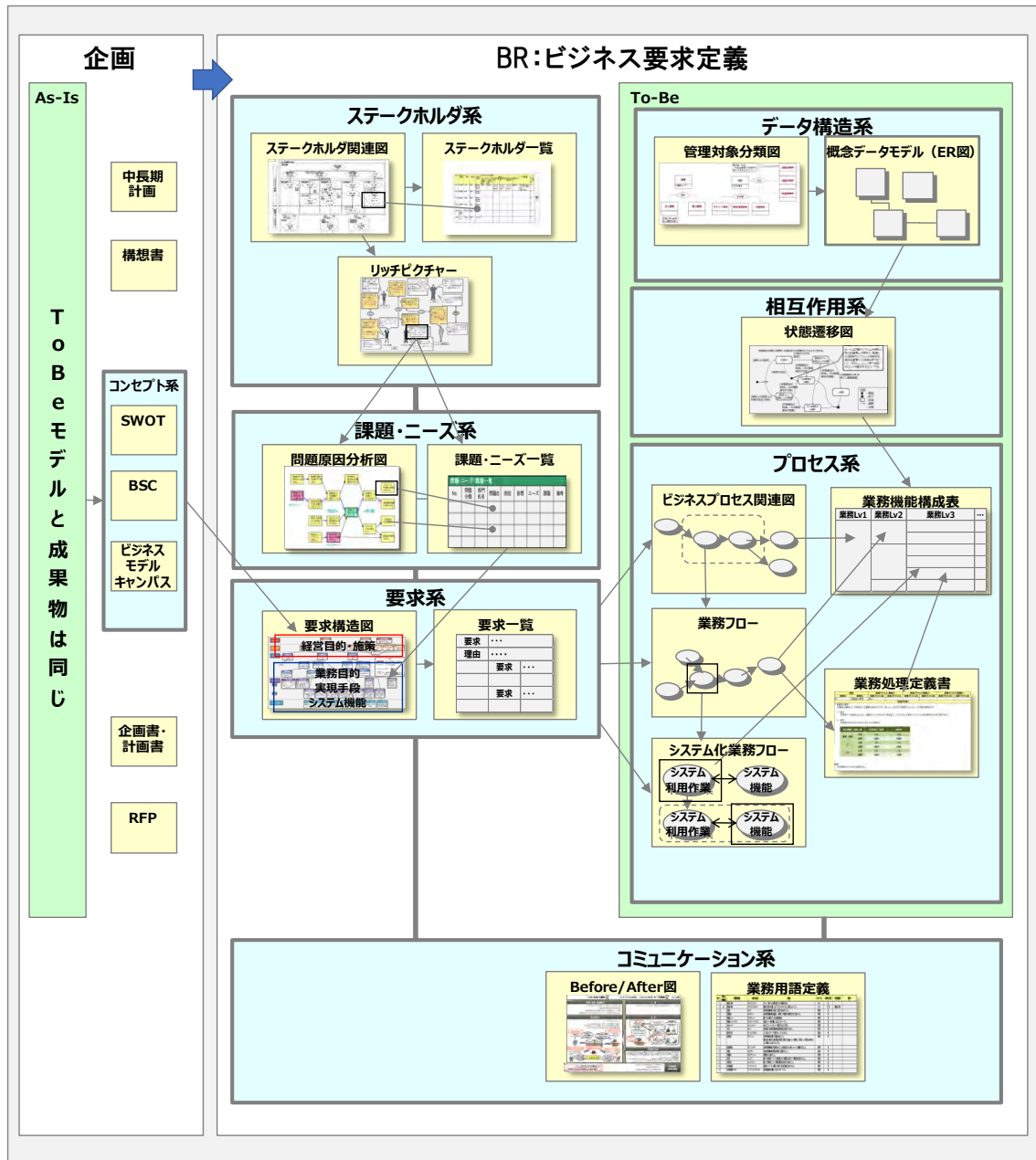


図 4.35 「BR. ビジネス要求定義」の主なドキュメントの関連

第5章 システム化要求定義(SR)における問題と解決の勘どころ

「SR. システム化要求定義」では、大きく2つのサブプロセスに分けて作業を行う。まず、「SR1. 仕様化」において、「BR. ビジネス要求定義」で文書化した要求を機能面、非機能面の両面で分析し、文書にまとめて仕様化する。続く「SR. 2 確認・評価」では、作成した文書の記述内容が構造的、意味的に正しいかどうかをレビューを通じて検証する。また、ビジネス要求に照らし合わせて、ステークホルダが期待している初期の要求を満たしているかどうかをステークホルダと確認する

「SR. システム化要求定義」における課題のカテゴリマップを図 5.1 に示す。

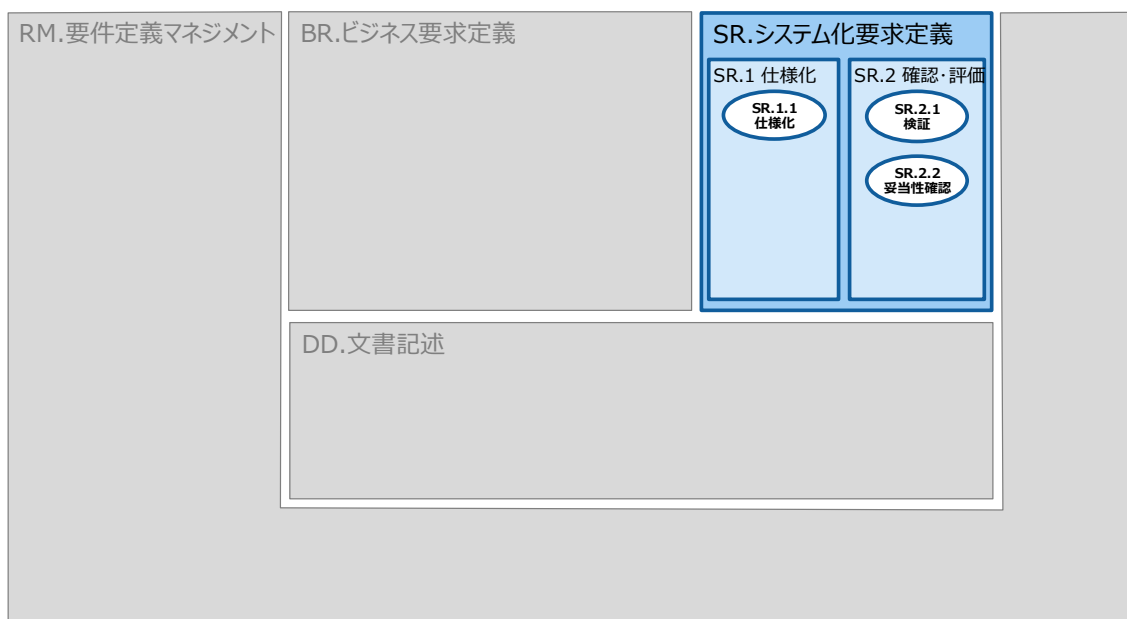


図 5.1 「SR. システム化要求定義」の課題カテゴリマップ

5.1 システム化要求の仕様化(SR.1)

5.1.1 仕様化(SR.1.1)

情報システムに関与するステークホルダが、実現手段の外観を理解し、導入する情報システムや製品が技術面、運用面で実現可能か、利用面で問題ないかを確認できる文書を作成する。

(1) ドキュメントの全体像、関連を明確にする

【解決したい問題】

- 要件定義で作成するドキュメントの全体像を理解していない
 - ・ ドキュメント間での不整合が発生している

「SR. システム化要求定義」においても「BR. ビジネス要求」同様、ドキュメントの全体像や関係性が明確でないためにドキュメント間の不整合が起こっている。ドキュメントの項目間の関係性を理解していないと不整合や標準化が徹底できない。

勘どころ① 成果物関連図を解説する

「SR. システム化要求定義」で作成する主要ドキュメントとその関係を図 5.2 に示す。なお、プロジェクトにおいて、「SR. システム化要求定義」の成果物定義を行う際には、「RM. 2.1 プロセス計画」の勘どころを参照して、作成する成果物の取捨選択、追加、ならびに記述粒度の定義を行うことが必要になる。

5.1 システム化要求の仕様化(SR.1)

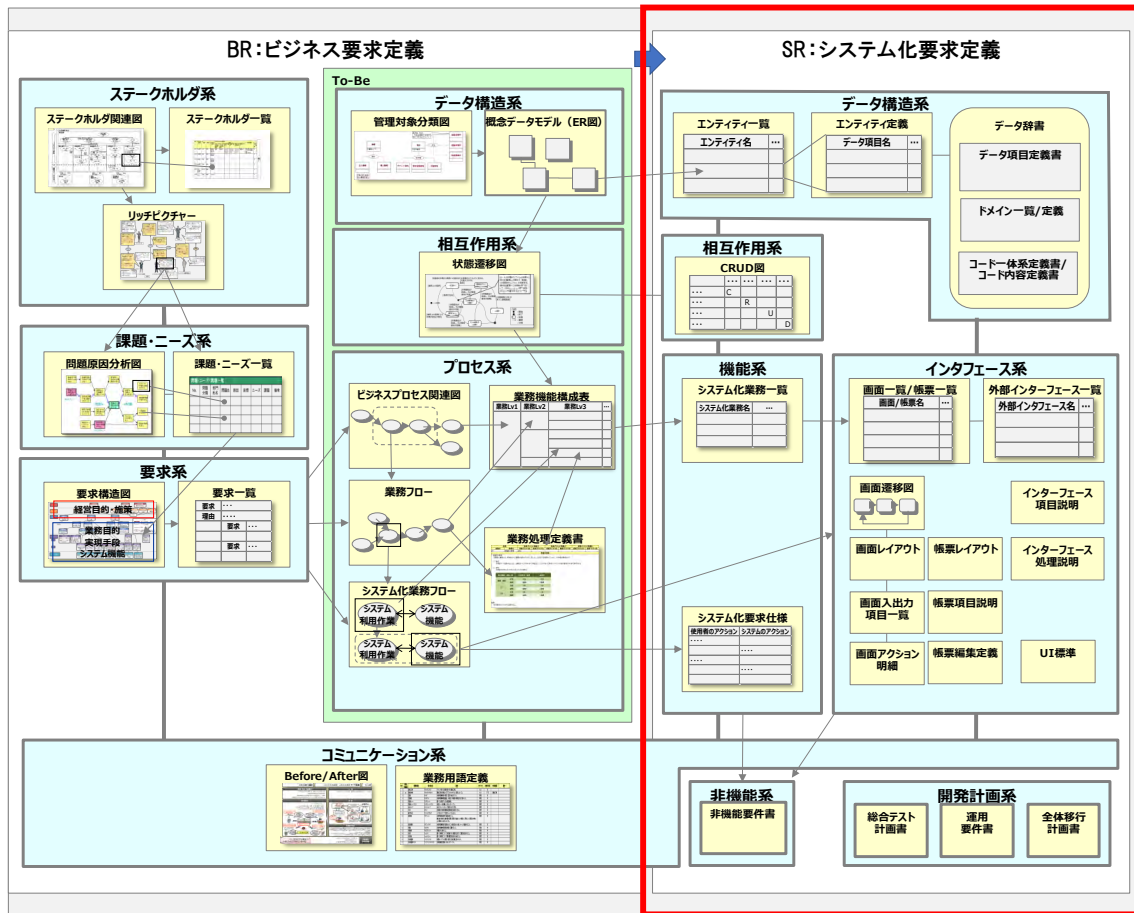


図 5.2 「SR. システム化要求定義」の主なドキュメントの関連

(2) 文章のあいまいさを排除し要求を正しく伝達する

【解決したい問題】

- 文章に抜け漏れあいまいがある
 - ・ 後続の設計品質が低下する
 - ・ ユーザテストで要求レベルの指摘が多発する
 - ・ 要求の実装漏れ、仕様誤りが本稼働後に発生する

要件定義においては、定義した事項は文書にして関係者で内容を確認し、記録として保管することにより、後工程に伝達する。そのため、要件定義した内容を記録した文書があいまいであると、定義した結果が正しく伝わらず、結果として開発されたシステムに仕様漏れや仕様誤りが入り込むことになる。そうすると、以下のような問題が発生する。

- 設計工程以降の成果物のレビュー時に指摘が多発し、設計期間が延びる
- 完成したシステムを利用部門が参加してテストした際に指摘が多発し、要件定義にまでさかのぼる大幅な手戻りになる
- 本稼働後に利用部門から機能の実装漏れや仕様誤りの指摘が多発し、業務運用に支障をきたす

上記のような機能不足への対応のためのシステム開発遅延、費用増加などの発生により、システム導入時に期待した投資対効果が得られないことになる。

勘どころ① 正確な日本語表現をする

要件定義の結果を正しく後工程に伝達するためには、作成される文書からあいまい性（内容が不明確、複数の解釈ができる、など）を排除する必要がある。そのためには、文章を作成する際に、自分が作成した文章に伝えたいことが正確に反映され、読み手に誤解なく伝えられるようになっているかどうか注意して、書いた文章を必ず数回推敲することが必要である。

以下に、要件定義文章の書き方についての注意事項を、全般的な注意点、助詞の使用法、口頭言語と書記言語の違いに分けて示す。内容は「SE を極める 仕事に役立つ文章作成術」（日本情報システム・ユーザー協会編、福田修著） [22]を参考にした。

要件定義文章の記述規約をあらかじめ決めておき、周知した上で要件定義書を作成すると、文書のあいまい性を排除するのに効果がある。あいまい性を排除した適切な要件定義文章の書き方についての全般的な注意点を以下に示す。

5.1 システム化要求の仕様化(SR.1)

表 5.1 要件定義文章（ソフトウェア文章）を記載する際の注意点

No.	注意事項
1	抽象的な単語は定義を明確にして使用する 特に「x x 管理」は何をすることか、明確に定義する 辞書を引いても出てこない業界の専門用語、企業固有の専門用語、また事業部門固有の言葉、ならびに辞書等と異なった意味で使われている用語をまとめておき、関係者間での齟齬をなくす
2	短文主義、一文一義で書く（長過ぎる文は読みにくい文章となる） 一文 40 文字平均が読みやすいと言われている 長い場合でも 1 文 200 文字以内、3 行以上の文は書かない
3	文には単文と重文それに複文の 3 種類がある 単文：A は B である。A が B する。 重文：A は B であり、C は D である。 複文：A が B であるため、C は D である。 技術文章は単文主義が原則であり、重文・複文は文を分ける
4	冗長、回りくどい、あいまい、嘲笑的な表現を排除する
5	技術文章では、原則「である調」を用いる 「である調」、「ですます調」、「体言止め」を混在させない
6	書けるものは、箇条書きで書く
7	表にできるものは表にする
8	視覚化する 図やイラストで、視覚化して伝える
9	数式化する 以上、より大きい、以下、未満などは日本語が母国語でないメンバにはあいまいとなりがちであるため、等号、不等号を用いて表現する
10	かなと漢字の比率は、2：1 が読みやすい 漢字の比率が 5 割を超えると読みにくくなる
11	形容詞や副詞を使用した場合は、補足説明が必要である [形容詞] 大量のハードディスク容量 → ハードディスク容量 \geq 100GB [副詞] 高速で動作する → 入力後 1 秒以内に応答がある

文書を正確に記述するためには、助詞を適切に使用することが必要になる。一般に、助詞を連続させると読み辛くなる。助詞は述部が表している「ありよう」や「振舞い」を整えるために用いるため、これを連続させると助詞の「ありよう」や「振舞い」が不明瞭になるからである。以下に助詞を使用する際の注意点を示す。

表 5.2 文書作成時に助詞を使用する際の注意点

No.	注意事項
1	「は」を連続させない (誤) 今日はテストは楽だった。 (正) 今日のテストは楽だった。
2	「が」を連続させない (誤) 進捗会議が時間が来るまで続いた。 (正) 進捗会議は時間が来るまで続いた。
3	<p>接続助詞「が」を用いない</p> <p>接続助詞「が」には以下の4つの機能があり、これを用いると読み手に4つの内どの意味なのかの選択を求め負担をかける。別の言い方をすると接続助詞「が」は書く立場からは便利な助詞であるために読者に負担をかける</p> <p>(i) 2つの事柄について接続詞の役割をする。共存か時間的推移を表す。 (誤) 重要な結果が報告されたが、現実には大きな矛盾が萌芽してきた。 (正) 重要な結果が報告された。しかし、現実には大きな矛盾が萌芽してきた。</p> <p>(ii) 題目・場面などを持ち出し、それらについて叙述する。 (誤) 開発の件ですが、5年前とは状況が変化しています。 (正) 開発の件については、5年前とは状況が変化しています。</p> <p>(iii) 補充的説明の添加 (誤) 機能障害はたびたび起きるものだが、多くはバグによるものである。 (正) 機能障害はたびたび起きる。その多くはバグによるものである。</p> <p>(iv) 内容を対比させる。前件に拘束されず後件が存在することを表す。 (誤) 文章には自信のある私であるが、できるだけ正確に書くよう心がけている。 (正) 文章には自信がある私であっても、できるだけ正確に書くよう心がけている。</p>
4	「の」を連続させない 格助詞「の」を連続して使うと文の意味が複雑になり、分かりにくくなる。 (誤) 演算命令の再度の実行の場合は、データの処理の時間の確保のための方法… (正) 演算命令を再度実行した場合は、データ処理時間を確保するための方法…
5	「を」を連続させない (誤) PCを保守などをする安全装置と呼ばれるソフトウェア (正) PCの保守などをする安全装置と呼ばれるソフトウェア
6	「に」を連続させない (誤) 明日までに移行作業に支障がでないよう、… (正) 移行作業に支障がでないよう、明日までに…
7	「まで」を連続させない (誤) 夜9時までテストが終わるまで待った。 (正) 夜9時までテストが終わるのを待った。

5.1 システム化要求の仕様化(SR.1)

また、文書作成においては口頭言語を用いない。話し言葉を口頭言語と言い、書き言葉を書記言語と言う。口頭言語と書記言語とは本来異質なものである。口頭言語は目の前の相手に対して当事者的関係においてする伝達の言語であるのに対して、書記言語は思考や感覚を自己の内面において客観化する認識の言語である。

以下に、口頭言語と書記言語の相違を示す。

表 5.3 口頭言語と書記言語

分類	口頭言語	書記言語
音便・省略	じゃない	ではない
	話している	話をしている
	そんな	そのような
	やっぱり	やはり
	みんな	みな
	話すなんて事は	話すなどということは
	いろんな	いろいろな
	人間なんである	人間なのである
自立語	お父さん、お母さん、僕	父、母、私
	すごく	とても、非常に
	ちゃんと	きちんと、はっきりと、正しく
	一発で	一度で、一回で
接続詞	でも	しかし
	なので	だから、このため
用言語尾	知らなく	知らず
	見れる	見られる
	移らさせて	移らせて
	書かして	書かせて
	みたい	ようだ
	というのは	というものは
助詞	おもしろいなと思った	興味深く感じた
	開くけど、開くけれど	開くが
	たら	なら

事例3 図表を使った記述による要件定義文章の品質向上 (株式会社エヌ・ティ・ティ・データ)

【取り組みの背景】

要件定義工程で定義する機能要求はその記述内容に応じて、標準的な技法（ER 図、UML、BPMN など）を使って記載する。しかし、ビジネスルールに関しては、標準的な技法が定着している状況ではない。このため、ビジネスルールをフリーフォーマットで記述してしまう傾向にある。

【課題】

ビジネスルールを自然言語で記述する場合、以下のような問題がある。

- あいまいな記述のため、抜けや矛盾を含みやすい
- 作成者によって記述粒度、品質のバラつきが生じやすく、可読性が低くなる、など

【解決策】

NTT データでは、要件定義書、設計書の作成時に、ビジネスルールをその内容に適した記法で記述できるよう、記法の使い分けと記述のコツを体系化し、グループ内のみならず、一般書籍「NTT ソフトウェアイノベーションセンタ、NTT データ：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、2015」 [23]として展開している。

ここでは、その書籍の内容を一部引用し、ビジネスルールの記述内容に対する記法の使い分けと、一部記法の概要を説明する。

(1) ビジネスルールの記述内容に合わせた記法の使い分け

ビジネスルールの記述に内容に合わせた記法の使い分けを下表に示す。

表 5.4 記法を選択

ビジネスルール分類	記述内容	記法
『事実』	データモデル	ER 図
	用語の定義	表（用語定義）
	項目の列挙	箇条書き
	分類の指定	表（分類）
	項目が存在する条件の指定	表（項目の存在条件）

5.1 システム化要求の仕様化(SR.1)

ビジネス ルール分類	記述内容	記法
	複数の値に対する関係の指定	計算式
	文字（記号、符号）列のフォーマットの定義	図（フォーマット）
		表（フォーマットの適用項目）
	値の有効範囲の指定	数直線グラフ
		表（値のセット）
	条件による事物などの定義	表（事物の成立条件）
	事物などの関係（関連と多重度）の定義	階層図、クラス図
行為に対する責任の割当て	責任分担表	
『契機』	アクターごとのアクティビティの定義	アクティビティ図、BPMN
『制約』	事物などに対するアクターごとの操作の制限	CRUD 図
	条件による行為の制限	デシジョンテーブル
		ステートマシン図
		表（状態ごとの行為の可否） 表（対象ごとの行為の可否）
	事前に実施する行為の指定	アクティビティ図、BPMN
	期間や期限の指定	期限チャート
値の増減に関する制限	折れ線グラフ	
空間的な制約の指定	空間配置図	
『推論』	事物などの状態変化の指定	ステートマシン図
	時間帯の指定	時間帯グラフ
	条件の組み合わせによる出力パターンの指定	例示
デシジョンテーブル		
『計算』	計算式の実行順序の指定	アクティビティ図
	四則演算、切り上げ切り捨て	計算式
	入力値によって変化する計算式の指定	階段グラフ
		計算式
		デシジョンテーブル

(2) 「期限チャート」を使用した「期間や期限の指定」の記述例

ある事由に対する実施内容とその期限が指定されている場合、事由と実施内容と期限の時系列関係を図示することで、条件に対して、実施する行為の期限が、時間の長さとして視覚的に理解しやすくなる。

さらに、具体的日付を入れた例も合わせて記述することによって、条件や実施する行為の時間的な制約として具体的な始点と終点を把握しやすくなる。

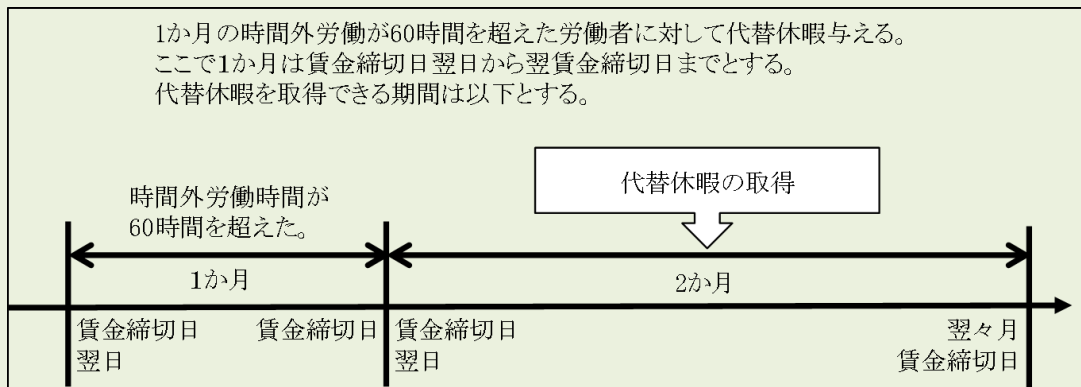


図 5.3 「期限チャート」をつかった「期間や期限の指定」の記述例

(3) 「デシジョンテーブル」を使用した「条件の組み合わせによる出力パターンの指定」の記述例

条件の組み合わせと判定結果（動作）を表形式で記述することにより、可読性の低下や、条件の組み合わせの漏れや誤りを見つけやすくなる。

①受験者の確認

試験会場（研修室）入り口で受験者持参の「写真付き証明書」で本人確認をする。

- ・「写真付き証明書」がない ⇒ 社員証、健康保険証等で本人確認ができれば例外的に受験可能。
- ・本人でない ⇒ 受験不可。
- ・受験日が違った ⇒ 受験不可。
- ・「受験票」がない ⇒ 受験票のコピーと照らし合わせ本人確認ができれば受験可能。

②受験者への説明

受験者に受験票を返し、オリエンテーション開始までに必ず着席するよう伝える。また、試験中はトイレ休憩がないため、事前に行くようあわせて伝えること。

③遅刻者の対応

試験受付時間内に受付をしない場合、受験することができない。但し、交通機関の遅延に限り、遅延証明書があり試験開始時刻を過ぎていない場合のみ受験を認める。

- ・交通機関の遅延（遅延証明書がない） ⇒ 受験不可
- ・その他の理由による遅刻（体調不良等） ⇒ 受験不可
- ・交通機関の遅延（試験開始時刻を過ぎた） ⇒ 受験不可
- ・交通機関の遅延（遅延証明書がある、試験開始時刻を過ぎていない） ⇒ 受験可能

図 5.4 ビジネスルールを自然言語で表現した例

5.1 システム化要求の仕様化(SR.1)

条件	写真付き証明書の有無	あり	Y	-	Y	-	-	-	-	-
		なし	-	Y	-	Y	-	-	Y	-
	写真付き証明書で本人確認	本人である	Y	-	Y	-	-	-	-	-
		本人でない	-	-	-	-	Y	-	-	-
	受験日	合っている	Y	Y	Y	Y	-	-	-	-
		間違っている	-	-	-	-	-	Y	-	-
	受験票の有無	あり	Y	Y	-	-	-	-	-	-
		なし	-	-	Y	Y	-	-	-	Y
	社員証、健康保険証等で本人確認	一致	-	Y	-	Y	-	-	-	-
		不一致	-	-	-	-	-	-	Y	-
	受験票のコピーと照らし合わせ本人確認	一致	-	-	Y	Y	-	-	-	-
		不一致	-	-	-	-	-	-	-	Y
動作	受験可能	X	X	X	X	-	-	-	-	
	受験不可	-	-	-	-	X	X	X	X	

凡例

【条件】 Y : 条件が真 - : 条件に無関係
 【動作】 X : 動作が生じる - : 動作が生じない

図 5.5 ビジネスルールをデシジョンテーブルで表現した例

5.2 システム化要求の確認・評価(SR.2)

5.2.1 検証(SR.2.1)

要件の構造的、意味的な正しさを確認することにより要求の内容を明確にし、要求の妥当性確認ができるようにする。

(1) 組織的なレビューをする

【解決したい問題】

- 記述内容の誤りを摘出しきれていない
 - ・ 後工程で要件定義書修正に伴う手戻りが発生する

勘どころ① レビューの準備(レビュー観点の整理)、レビュー実施体制、やり方を定義する

要件定義で定義すべき内容は多岐にわたるため、抜け漏れを完全に防止することは難しい。しかし、過去の案件や類似システムでの実績などから、不具合が入り込む観点を整理し、レビュー観点としてまとめた上でレビューすることにより、抜け漏れを抑止することは可能である。また、レビューの役割分担、準備を入念に行うことで、不具合の摘出を効率的にできる。

成果物のレビューとひと口に言っても、レビュー手法には非公式、公式を含め、多くの種類が存在する。代表的なレビュー手法と期待される主な効果を表 5.5 に示す。組織の成熟度やメンバの経験度に合わせて適切なレビュー手法を選択していただきたい。繰り返しレビューを行った結果にもとづくレビュープロセスの改善や、プロジェクトメンバへのトレーニングなどにより、組織的にレビュープロセス成熟度の向上を図ることが望ましい(事例5参照)。

5.2 システム化要求の確認・評価(SR.2)

表 5.5 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	クリティカルまたは高リスクのコンポーネントの品質の実証	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	確認 変更またはバグ修正の正しさの	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	●	●	●	●	●	●	●	●						

(出典) SECBOKS「高信頼化ソフトウェアのための開発手法ガイドブック」 [24]

プロジェクトの特性にもよるが、特にソフトウェアの高信頼化に関連する要求および要件定義成果物のレビュー方法を検討する場合、インスペクションは重要な手法になる。よって、本稿ではインスペクションに焦点を当て説明する。なお、インスペクションは、「SR2.1 レビュー」だけではなく、「SR2.2 妥当性確認」にも有用であるため、「SR2.2 妥当性確認」を行う際の勘どころとしても参照いただきたい。

インスペクションは ANSI/IEEE 標準において体系が定義されている最も厳格で公式なレビュー手法である。インスペクションは他の手法に比べ、実施コストは高くなるが、欠陥の発見に最も効果が高い手法と言われており、その目的を次のように定めている。

- 成果物（ソフトウェア要素）が基本仕様、指定された品質属性、顧客要求を満たしているかどうかの検証
- 成果物（ソフトウェア要素）が関連する基準、規制、規則、計画、手順に適合しているかどうかの検証
- 発見された欠陥とインスペクションに使用した時間に関する評価指標の供給
- 表現方法の良し悪しや文体に対する検査は行わない

また、インスペクションでは手順も定められており、一般的に図 5.6 に示すように7つの段階を踏む。

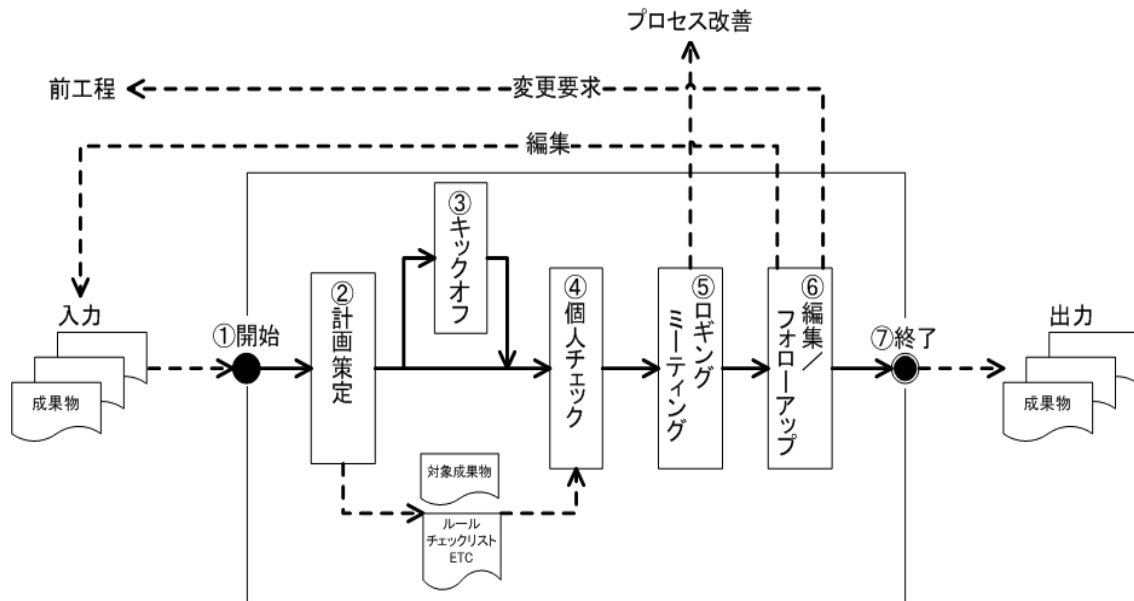


図 5.6 インспекションのプロセス

(出典) SECBOOKS 「高信頼化ソフトウェアのための開発手法ガイドブック」 [24]

以下に、インスペクションを進める上での勘どころ、留意事項を記載する。

- ビジネス目的・施策をインスペクション関係者間で共有する
要件定義成果物を対象にしたインスペクションの導入は、ユーザ要求への適合性検証が主眼に置かれるため、ビジネス目的・施策を各役割の担当者間で共有しておく必要がある。さらに、品質を重視するチーム文化の構築、各種スキル(知識含む)の共有・移植促進およびステークホルダ間の合意形成などの工夫努力により、チーム全体の結束を高める効果が副産物として生まれることを認識すること。
- チェックリストを準備する
品質の観点から要件定義成果物に対するチェックリストを準備することが必要である。インスペクションでのロギングミーティングの開始前に、誤字脱字や形式の不整合など一見して分かる欠陥は、チェックリストの配布などにより単独チェックによって検出しておくこと。さらに、ISO/IEC 25010「品質モデル」 [12]など、何らかの品質基準を基にチェックリストなどを作成し、観点のバラつきをなくす工夫も必要である。
- 役割を明確にして効率的に欠陥を取り除く
インスペクションの特徴の一つに、役割をはっきり分けることがあげられる。インスペクションには、作成者、進行・まとめ役、記録役、説明役、レビューの5つの役割がある。特に注意しなければいけないことは、作成者は説明役以外を兼任でき

5.2 システム化要求の確認・評価(SR.2)

ないという点である。インスペクションでは、第三者の目を見た客観性、論理性、具体性、実現性などについての指摘などの評価コメントを受け、これに対して回答することで、作成者の思い込みを排し、可能な限り多くの欠陥を取り除くことを目指している。指摘やコメントは記録役によって迅速に収集される必要があることから、インスペクションの対象にもよるが、インスペクションでは記録役、進行・まとめ役の負担もかなり高いため、これら2つの役も兼任を避けるべきである。

表 5.6 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>

(出典) SECBOOKS「高信頼化ソフトウェアのための開発手法ガイドブック」 [24]

- 進行中の成果物は事前に出来栄を確認する
進行中の成果物をレビューする場合は、インスペクション開始時に成果物の作成者に、未完成部分と完成部分、さらには完成部分でも不安要素が残る部分があるなどといった成果物全体の出来栄状況をヒアリングしておく。ヒアリングしておくことで、事前にレビューポイントを押さえておくことができる。
- 検出と除去(対策)は分離して行う
インスペクションの目的はあくまで欠陥の検出であり、対策の検討、議論は別の場

で実施する。

- レビュー・メトリクスによる予実評価と傾向分析を行う
インスペクション結果から得られる成果の一つに、レビュー・メトリクスの予実評価による変動幅の小さいベースラインの確保がある。変動幅の小さいベースラインの確保とは、手順、手法および規約を整備し、有効なレビュー実績データを蓄積・評価してレビュー指標の変動幅を小さくし、組織標準にすることである。メトリクスにはレビュー密度、欠陥指摘密度、レビュー指摘効率および手戻り量などがある。さらに、機能分類ごとや、重要機能および重大欠陥などでの欠陥指摘傾向を考慮することで、欠陥の偏在性向などを掴むことができ、是正や予防につながる。なお、傾向分析においては欠陥を作り込んだ当事者に対する人的評価は行わないことが重要である。
- 指摘事項を整理して横展開をする
インスペクションで得た指摘事例等は整理して実事例集（症例ガイドンス等）として横展開すること。横展開の内容に基づきチェックリストを改善するなど組織的なレビュープロセス改善を行うことができる。

事例 4 要件定義全体を俯瞰したインスペクションによる手戻りの防止 (株式会社 ジャステック)

【取り組みの背景】

一般的に個々の業務および要件定義成果物ごとにレビューすることで、検証・妥当性確認を終了させてしまうことがある。しかし、全体を俯瞰した検証・妥当性確認ができていないため、個々の業務や要件定義成果物間の不整合/不統一、仕様漏れなどが開発工程で検出され、手戻りが発生するケースがよく見受けられる。

【課題】

弊社での販売管理業務システムの再構築開発において、こうした問題を解決すべく以下の2つの課題を設定した。

- 要件定義全体を俯瞰した検証・妥当性確認を可能とする生産物の課題
- 本来要求(目的・施策)に不適合な仕様欠陥を摘発するためのレビュー方法の課題

【解決策】

◆要件定義中間生産物による要件定義全体を俯瞰したインスペクション

弊社の「インスペクション導入ガイド」には、要件定義工程のインスペクションは鉄則として詳細から検証するのではなく全体を俯瞰した検証から実施すること、および本来要求(目的・施策)に不適合な仕様欠陥の摘発に留意した妥当性確認のガイドがある。また、要件定義全体を俯瞰したインスペクションとして中間生産物の作成を促している。以下に当該再構築開発で作成した3つの中間生産物(システムテスト仕様の作成にも有効)の事例を紹介する。

5.2 システム化要求の確認・評価(SR.2)

表 5.7 CRUD 表 (テーブル編) による機能の抜け・漏れ・不整合の確認

【新販売管理業務システム】CRUD表 (テーブル編)		「O(登録)、R(閲覧)、U(更新)、D(削除)」																																	
機能	DB名称	案件ワーク					決裁情報管理				決裁履歴				請求書決裁管理				定数テーブル			保守・運用													
	テーブル名称	案件ワークDB・案件情報	プロジェクト情報	プロジェクト月次情報	契約情報	契約明細情報	請求予定情報	決裁情報管理	決裁申請管理	決裁情報管理	決裁履歴管理	プロジェクト月次情報	プロジェクト月次情報	プロジェクト月次情報	契約明細情報	契約明細情報	請求予定情報	請求書集約情報	請求書集約情報	請求書集約情報	地域区分	地域区分	地域区分	定数テーブル	定数テーブル	定数テーブル	売上集約連携ログ	販売管理システム	ログインID						
	テーブル使用数 (機能No01. ~No. 11)	5	1	7	5	5	6	1	5	4	7	4	2	5	1	6	5	1	6	8	1	7	1	1	7	3	1	1	1	1	1	3	5	2	1
01. メインメニュー	1.0.1 システムメニュー																																		
02. 顧客契約承認	1.1.1 案件情報	R	U	R	U	R	U	R	U	R	U	C	C	C	C	C																			
	1.1.2 プロジェクト情報																																		
	1.1.3 プロジェクト月次情報																																		
11. 保守・運用	1.10.1 パスワード管理																																		
	1.10.2 占有使用開始終了																																		
	1.10.3 ログインユーザ照会																																		
	1.10.4 主管事業部、主管部署一括変換	R	U	R	U	R	U	R	U	R	U																								
	1.10.5 売上用顧客割付																																		

表 5.8 新旧画面(論理)対比表による旧画面に対する新画面の対応方法の確認

新画面	画面名	画面詳細	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)				プロジェクト情報																									
			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	売請用主管部署選択																				
1.1	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																																			

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

表 5.9 業務とワークフローステータス対比表による相互の関連を確認

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V2747開発-①取引開始承認	●			●	●	●	●	●	●	●	●
② (1)受注V2747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	●									●	●

【効果】

表 5.7 の CRUD 表は機能とエンティティに抜けや漏れがないかを分析・検証するのに有効である。ただし、業務部門を巻き込む場合、業務部門に分かりやすい表現にするなどの工夫をすることも必要である。

表 5.8 は旧画面の各項目が新画面のどの項目に紐付いているかを分析・検証するのに有効である。

表 5.9 はインスペクションを含め、特にシステムテスト計画における業務シナリオ作成時においても、対応するワークフローに抜けや漏れがないかを分析・検証するのに有効である。

さらに、全体を俯瞰したインスペクションを実施することで検証効率の向上が可能になること、また、レビュー時間が足りなくなった場合に影響度が大きい欠陥を先に検出できているため、詳細はサンプリングでも足りうる場合もある。

事例 5 定量的に捉えた手戻りコストとレビュープロセス 成熟度向上による手戻り改善 (株式会社 ジャステック)

【取り組みの背景】

要件定義工程での成果物の誤り、漏れ、不整合などが、それ以降の開発工程で摘発されると、当該摘発工程により手戻りコストが増減する。特に、開発工程ではシステムテストでの摘発が最大になるが、具体的な数値で捉えられているだろうか。数値を感覚的に捉えている場合や、あるいは仕様変更や欠陥修正はどの開発工程で行っても変わらないといったような手戻りコストを意識していない場合もある。その結果、開発の後工程で手戻りコストなどが増加して、QCD が悪化するケースが見受けられる。

【課題】

こうした問題を解決すべく、弊社では手戻りコストに関わる見える化と図る化を推進すべく、以下の3つの課題に取り組んでいる。

- 当該工程独自欠陥について、欠陥摘発工程の違いによる手戻りコストを定量的に捉える課題
- 手戻りコストを削減するためのレビュープロセス成熟度向上の課題
- 手戻りコストの改善率を取り入れたチームの目標管理課題

【解決策】

これら3つの課題に対し、要件定義工程で混入した欠陥を当該工程のレビューで摘発した場合と以降の開発工程で摘発した場合の手戻りコストに関わるROI (Return On Investment) を算出するとともに、レビュープロセス成熟度と手戻りコストに関わるROIを使用し、要件定義工程のレビュープロセス成熟度向上による手戻りコストの改善を図っている。

ソフトウェア製品の検証^{*1}および妥当性確認^{*2}は、各々、弊社のレビューおよびテストの両方に該当する。当該事例では、特に要件定義工程のレビューを中心に据え、弊社のレビューに関する取り組み事例を紹介する。

*1: 「仕様どおりであることの確認」を行うこと。

*2: 「ユーザの仕様意図が、十分に実現されていることの確認」を行うこと。

5.2 システム化要求の確認・評価(SR.2)

(1) 欠陥摘発工程の違いによる手戻りコストから捉えた ROI

一般的に、混入した工程独自欠陥の手戻りコストは、経験的にレビューとテストでの摘発工程の違いにより増減することがわかっている。しかし、どのくらいの増減値になるかを定量的に捉えた文献資料は極めてまれである。

当該事例では要件定義工程独自欠陥を要件定義工程レビューで摘発した場合と、それ以降の開発工程（基本設計～システムテスト）で摘発した場合の手戻りコストの増減値を定量的に捉えることで、費用対効果を算出している。具体的には手戻り量の算出式と各工程間生産物量変換係数（各工程生産物量単価を含む）を使用し、手戻りコストを求めるとともに、各工程の実績収集データと試算値とを突き合せ照合し、精度を高めている。なお、本事例「定量的に捉えた手戻りコスト」は弊社の「仕様変更見積り方式」に基づき応用した事例である。

◆手戻り量の捉え方

a) 欠陥混入工程と欠陥摘発工程を鑑みた手戻り量

ある工程 i の成果物に混入した工程独自欠陥による手戻り量と開発量との関係は図 5.7 のとおりである。

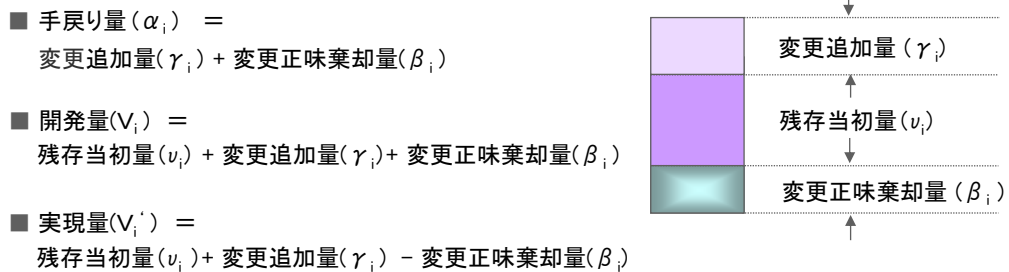


図 5.7 工程独自欠陥による手戻り量と開発量との関係

しかし、ある工程 i に混入した工程独自欠陥の手戻り量は、必ずしも工程 i の成果物対応だけにとどまらない。つまり、手戻り量は欠陥混入工程と欠陥摘発工程との関係で増減する。その欠陥混入工程（ここでは要件定義工程）と欠陥摘発工程との関係を考慮した手戻り量は、以下の算出式で求めることができる。理解を深めるために表 5.10、表 5.11 に事例を記載している。

$$\text{手戻り量} = \sum (n_i \times \gamma_i) + \sum (n_k \times \beta_i)$$

n は要件定義工程を基準 '1' ($n_1=1$) とした場合の各工程生産物量変換係数
 i は開発工程を指し、欠陥混入工程からシステムテスト工程までを対象
 k は開発工程を指すが、欠陥混入工程から欠陥摘発工程までを対象

表 5.10 要件定義工程独自欠陥を要件定義成果物のレビューで欠陥摘出したケース

開発工程 量	要件定義 (i = 1)	基本設計 (i = 2)	パッケージ設計 (i = 3)	プログラム設計 (i = 4)	プログラミング (i = 5)	統合テスト (i = 6)	システムテスト (i = 7)
変更追加量 (γ_i)	γ_1	$n_2 \times \gamma_1$	$n_3 \times \gamma_1$	$n_4 \times \gamma_1$	$n_5 \times \gamma_1$	$n_6 \times \gamma_1$	$n_7 \times \gamma_1$
残存当初量 (ν_i)	ν_1	$n_2 \times \nu_1$	$n_3 \times \nu_1$	$n_4 \times \nu_1$	$n_5 \times \nu_1$	$n_6 \times \nu_1$	$n_7 \times \nu_1$
変更正味棄却量 (β_i)	β_1	—	—	—	—	—	—

留意) n_i は生産物量変換係数で、ここでは $n_1=1$ である。手戻り量 = $\sum (n_i \times \gamma_i) + \beta_1$ 本事例は $K=1$ ($n_1=1$)

表 5.11 要件定義工程独自欠陥をシステムテストで欠陥摘出したケース

開発工程 量	要件定義 (i = 1)	基本設計 (i = 2)	パッケージ設計 (i = 3)	プログラム設計 (i = 4)	プログラミング (i = 5)	統合テスト (i = 6)	システムテスト (i = 7)
変更追加量 (γ_i)	γ_1	$n_2 \times \gamma_1$	$n_3 \times \gamma_1$	$n_4 \times \gamma_1$	$n_5 \times \gamma_1$	$n_6 \times \gamma_1$	$n_7 \times \gamma_1$
残存当初量 (ν_i)	ν_1	$n_2 \times \nu_1$	$n_3 \times \nu_1$	$n_4 \times \nu_1$	$n_5 \times \nu_1$	$n_6 \times \nu_1$	$n_7 \times \nu_1$
変更正味棄却量 (β_i)	β_1	$n_2 \times \beta_1$	$n_3 \times \beta_1$	$n_4 \times \beta_1$	$n_5 \times \beta_1$	$n_6 \times \beta_1$	$n_7 \times \beta_1$

留意) n_i は生産物量変換係数で、ここでは $n_1=1$ である。手戻り量 = $\sum (n_i \times \gamma_i) + \sum (n_k \times \beta_i)$ 本事例は $K=1$

手戻り量は欠陥混入工程と欠陥摘発工程との関係で増減するが、前事例(表 5.10、表 5.11)からも分かるように、その増減する値は、既に作成された成果物への変更追加量 (γ) と変更正味棄却量 (β) の多寡に依存する。しかし、表 5.10 と表 5.11 の差分値は、' $\sum n_k \times \beta_1$ ($k=2\sim 7$)' となり変更正味棄却量 (β) に依存している。よって、レビューとテストでの手戻り量(コスト)に関わる ROI を考察する場合には、変更追加量 (γ) の多寡は考慮する必要がない。

b) デグレードを考慮した手戻り量

欠陥摘発工程がテスト工程である場合は、デグレード(改造型開発のテスト巻き込み)を考慮する必要がある。例えば、デグレード率 (d) によるテスト追加量を取り入れた手戻り量は、以下の算出式になる。

5.2 システム化要求の確認・評価(SR.2)

$$\begin{aligned} \text{デグレードを考慮した手戻り量} &= \sum (n_i \times \gamma_i) + \sum (n_k \times \beta_i) \\ &+ \sum (d_j \times n_j \times \nu_i) \end{aligned}$$

〔 $\sum (d_j \times n_j \times \nu_i)$ の‘j’は単体テスト（プログラミング）j=5から欠陥摘発テスト工程までを対象〕

(2) 欠陥混入工程と欠陥摘発工程の違いによる増減する手戻りコストの比較事例

手戻りコストは、生産物量変換係数（n）*3を考慮した手戻り量に工程別生産物量単価（m）を乗算することで算出することができる。よって、欠陥混入工程（要件定義工程）と欠陥摘発工程との違いにより増減する手戻りコストは以下の算出式になる。なお、 m_j の‘j’は上記「 $\sum (d_j \times n_j \times \nu_i)$ 」の‘j’に準拠する。

*3：各工程間の生産物量変換係数は4つの基本パターン（基盤と処理形態の組み合わせ）が存在する。

$$\text{増減する手戻りコスト} = \sum ((n_k \times \beta_i) \times m_k) + \sum ((d_j \times n_j \times \nu_i) \times m_j)$$

表 5.12 に生産物量変換係数「基盤‘Server系’、処理形態‘Online’」のパターンについて、要件定義書‘1’とした場合の生産物量変換係数を紹介する。

表 5.12 生産物量変換係数

開発工程生産物 パターン	要件定義書	基本設計書	パッケージ 設計書	プログラム 設計書	プログラミング	統合テスト 項目	システムテスト 項目
基盤 : Server 処理形態 : Online	1	1.44	7.6	9.53	0.93	65.55	18.69

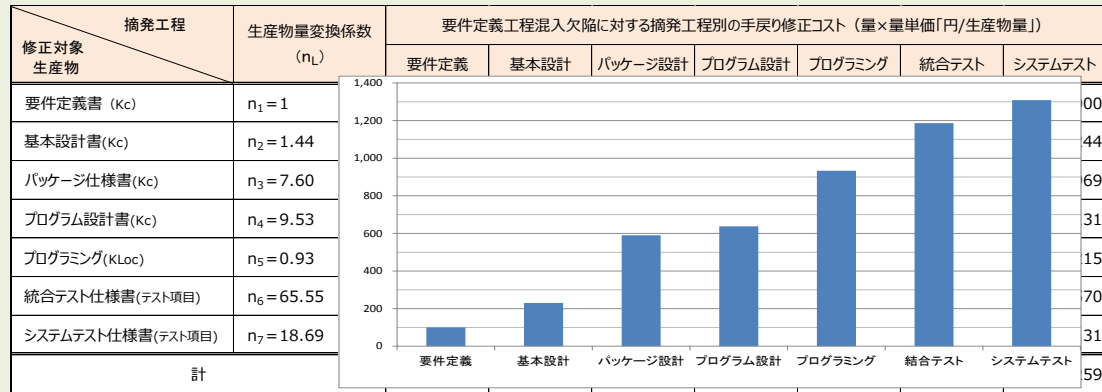
注)テスト工程生産物量は組数(ケースなど)を使用しているが、ここでは一般的に使用されているテスト項目数で表記している。

増減する手戻りコスト算出式と表 5.12 の生産物量変換係数から、要件定義工程独自欠陥を要件定義工程で欠陥摘発した手戻りコストを‘1’とした場合の、要件定義工程以降の開発工程での各欠陥摘発工程の違いによる手戻りコストのコスト倍率事例を表 5.13 に示す。

表 5.13 基盤‘Server系’、処理形態‘Online’のケース

要件定義	基本設計	パッケージ設計	プログラム設計	プログラミング	統合テスト	システムテスト
1.00	2.30	5.90	6.38	9.33	11.86	13.09

注) デグレード率は25%(各テスト工程)



なお、表 5.13 は基盤‘Server系’、処理形態‘Online’のケースであるが、基盤‘Host系’、処理形態‘Online’のケースでは、要件定義工程独自欠陥を要件定義工程で欠陥摘発した手戻りコストを‘1’とした場合、生産物量変換係数の違いによりシステムテストでは‘13.70’になる。

(3) レビュープロセス成熟度と品質指標

① レビュープロセス成熟度とは

ソフトウェア成果物を早期にかつ効率的に欠陥を取り除くには、組織のレビューに関するプロセス能力を高めることが必要である。当該事例ではレビューに関わるプロセス能力を、組織の振る舞いの観点からレビュープロセス成熟度（レベル1～レベル4）として定義する。

その概要を図 5.8 に示す。

5.2 システム化要求の確認・評価(SR.2)

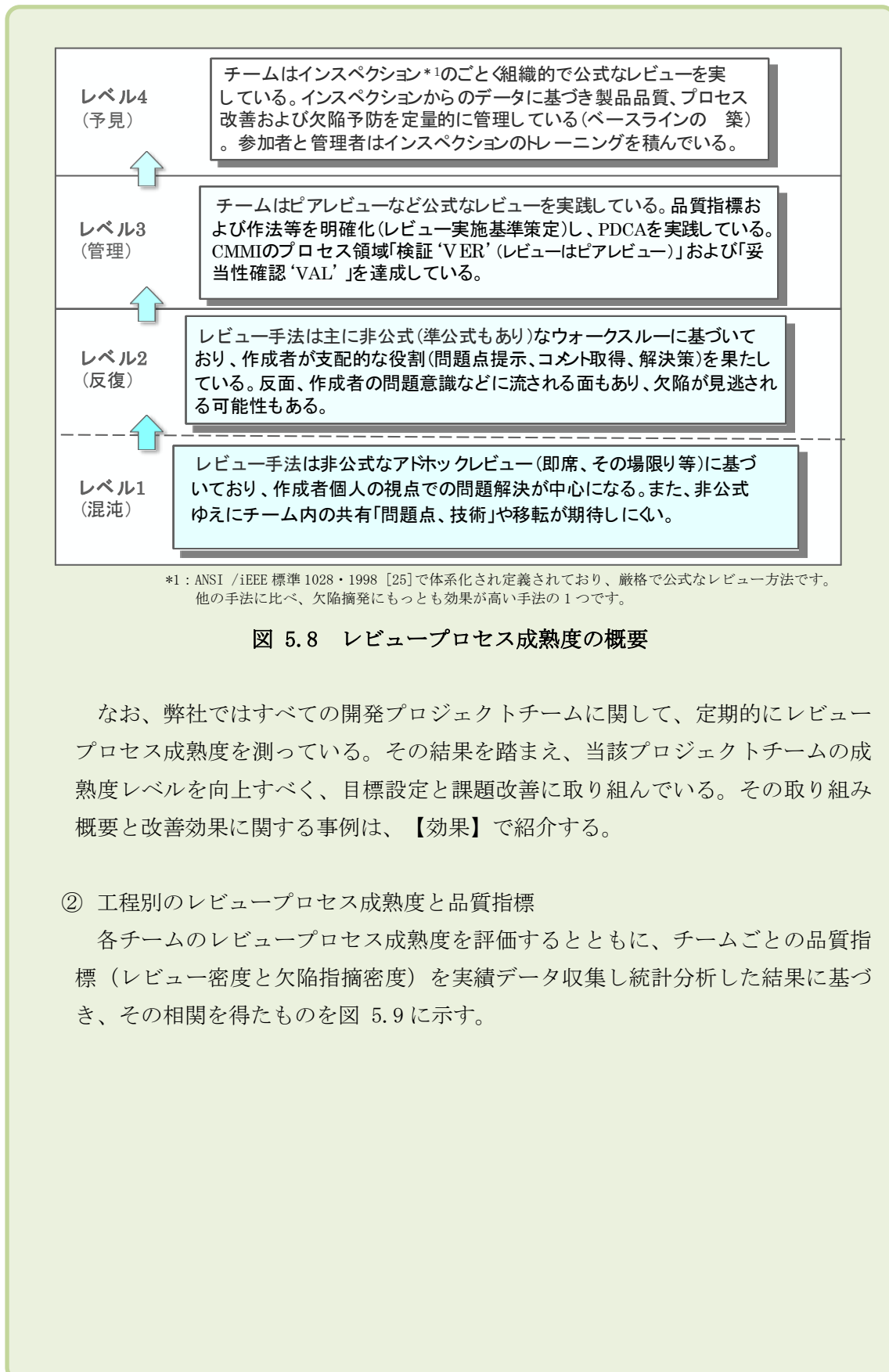


図 5.8 レビュープロセス成熟度の概要

なお、弊社ではすべての開発プロジェクトチームに関して、定期的にレビュープロセス成熟度を測っている。その結果を踏まえ、当該プロジェクトチームの成熟度レベルを向上すべく、目標設定と課題改善に取り組んでいる。その取り組み概要と改善効果に関する事例は、【効果】で紹介する。

② 工程別のレビュープロセス成熟度と品質指標

各チームのレビュープロセス成熟度を評価するとともに、チームごとの品質指標(レビュー密度と欠陥指摘密度)を実績データ収集し統計分析した結果に基づき、その相関を得たものを図 5.9 に示す。

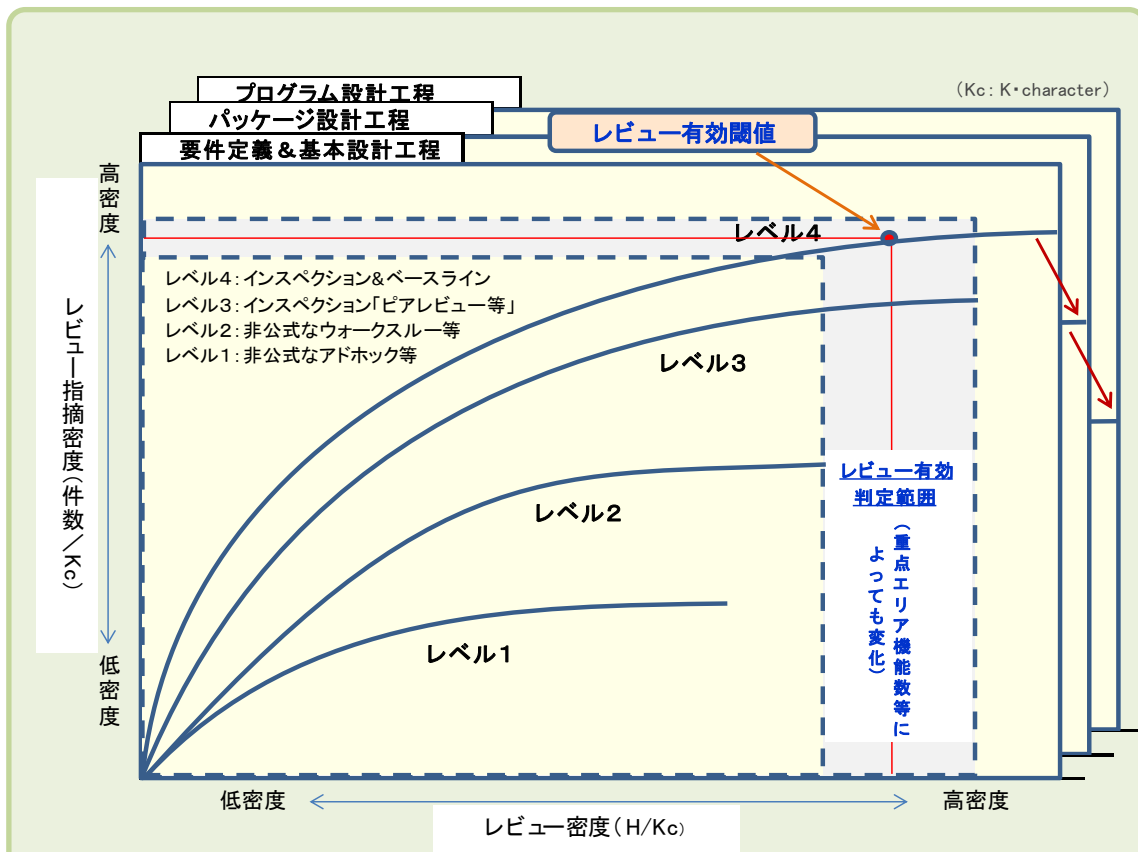


図 5.9 工程別のレビュープロセス成熟度と品質指標

相関関係は「要件定義&基本設計工程」～「プログラム設計工程」までを調査している。図 5.9 には「要件定義&基本設計工程」の事例を表記しているが、実績データ収集から得られた相関関係は、「パッケージ設計工程」、「プログラム設計工程」と工程が進むにつれて、レベル1～レベル4の回帰曲線は図に示す(赤色の矢印)ように寝てきており、かつ、レベルの差は縮む傾向にある。

つまり、上流工程「要件定義&基本設計工程」ほど、レビュープロセス成熟度の高レベル効果が現れることになる。よって、弊社では、特に上流工程での高レベル化を図っている。

(4) レビュープロセス成熟度の向上による手戻りコスト改善事例

◆ レビュープロセス成熟度の違いによる手戻りコスト差異

要件定義に混入した要件定義工程独自欠陥が、どの工程でどのくらいの割合(%)で欠陥摘発されるかを、チーム(レビュープロセス成熟度)ごとに実態を調査している。以下にレビュープロセス成熟度レベルごとの欠陥摘発割合分布(平均値)を示す。

5.2 システム化要求の確認・評価(SR.2)

表 5.14 レビュープロセス成熟度レベルごとの欠陥摘発割合分布(平均値)

レビュープロセス成熟度 \ 開発工程	要件定義 摘発比率%	基本設計 摘発比率%	パッケージ設計 摘発比率%	プログラム設計 摘発比率%	プログラミング 摘発比率%	統合テスト 摘発比率%	システムテスト 摘発比率%	計 摘発比率%
レベル4	94.2	2.3	1	0	0	0.9	1.6	100
レベル3	88.5	4.9	1.6	0	0	1.7	3.3	100
レベル2	73.4	9.2	3.8	0.8	0.1	4.4	8.3	100

参考1) 純粋な欠陥(誤り、漏れ等)以外に、後工程での積み残し分(仮置き含む)の対応も含まれる。

参考2) JUASソフトウェアメトリクス調査2016では工期遅延の50%が要件定義工程が不十分との調査結果もある。

ここで、上記表での工程別の欠陥摘発割合および表 5.13 での手戻りコストのコスト倍率を使用し、レビュープロセス成熟度レベルの違いによる手戻りコスト差異を試算している。その結果、手戻りコスト差異はレベル4を‘1’とすると、レベル3は1.3倍に、レベル2は2.1倍のコスト増になるという結果を得た。

【効果】

◆レビュープロセス成熟度の向上による手戻りコスト改善効果事例

1つ目は表 5.14 のレビュープロセス成熟度レベルの違いによる手戻りコスト差異の試算事例から、レビュープロセス成熟度レベル向上による手戻りコストの改善効果が得られる。

以下にレベル向上の3パターンを想定した改善効果を示す。

- レベル2からレベル3への向上による手戻りコストの削減効果は39%
- レベル3からレベル4への向上による手戻りコストの削減効果は21%
- レベル2からレベル4への向上による手戻りコストの削減効果は52%

2つ目は要件定義工程に特化した事例ではない。先に、弊社の全開発プロジェクトチームに関して、定期的にレビュープロセス成熟度を測っていることを述べた。ある年度(改善前)のプロジェクトチーム別のレビュープロセス成熟度レベルの実績から、チーム別に成熟度レベルの向上目標を宣言(改善後)し、向上による組織全体のコスト改善効果を得る事例である。ここでは、紙面の関係で詳細は割愛するが、直近の年度予算において、改善前と改善後のコスト改善比率8.1%を目標としている。

ただし、成熟度レベル向上に関わる費用(トレーニングや資料整備などの費用、管理費用など)を除いている。

5.2.2 妥当性確認(SR.2.2)

(1) 業務部門が分かるように成果物の作成およびレビューを工夫する

【解決したい問題】

- システム化要求の妥当性を業務部門が確認するのが難しい
 - ・ システム部門やベンダが作成したドキュメントを業務部門が分からない、レビューできない
 - ・ 実現したいビジネス要求がシステム化仕様として定義できていない

勘どころ① ビジネス要求がどのようにシステム化要求に反映されているかが、相手に正確に伝わるようにレビュー対象物を作成する

定義したビジネス要求がシステム化要求に適正に反映できているかを確認せずに進めた場合、稼働後、想定していたほどのビジネス効果を得られなくなってしまう。したがって、業務部門とのシステム化要求レビューにおいて、想定しているビジネス要求が実現できているかどうかを確認、合意することは重要である。

しかし、IT に馴染みが薄い業務部門担当者の場合、専門的な技法で記述されたドキュメントを見て妥当性を確認することは難しい。そのような場合は、実業務と照らし合わせた図表を補助資料として作成し、その補助資料をベースにレビューを行うと良い。

ここでは、概念データモデルを題材に、業務部門との妥当性確認の勘どころを説明する。

要件定義で作成する概念データモデルはデータベースの設計図ではなく、実世界（業務そのもの）を写像したモデルである。本来、業務部門が主体となって業務をモデル化し、概念データモデルとして定義することが理想であるが、概念データモデルの作成は学習と実践を繰り返さないとなかなかうまくできるようにならないのが現実であり、多くのプロジェクトでは、システム部門の有識者か外部の有識者が作成者にアサインされている。

そこで発生することは、作成する側は概念データモデルを作成する技術を保有しているが、概念データモデル化する対象の業務には精通していないことであり、逆に、業務部門側は業務を理解しているが、作成された概念データモデルが業務を正しく写像しているかどうかを確認することができないという状況である。このようなギャップを埋めるためには以下に配慮すると良い。

- 業務部門にヒアリングしながら概念データモデルを作成する
 - 業務部門から業務内容をヒアリングしながら、理解した内容をフローや概念デー

5.2 システム化要求の確認・評価(SR.2)

タモデルとしてホワイトボード等書き示していくことにより、業務部門の人にも概念データモデルが表現している内容を理解してもらいながら作成を進める。

- 概念データモデルそのものではなく補助説明資料を用意して確認する
概念データの構造を絵で表現して説明し、内容理解の妥当性を確認するイメージを図 5.10 に示す。

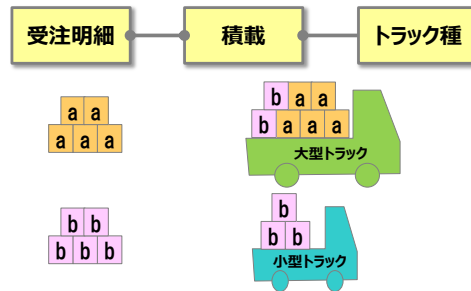


図 5.10 概念データモデル補足説明（絵）

この例は、一回の注文で受注した商品を複数のトラックに小分けして積載・配送する処理があるかどうか、そして、量が少ない場合には運送手段を経費の安い小型トラックに変更するような運用が認められているかなどを、絵と概念データモデルで表現したものである。

次に、構造をインスタンス（値）で表現して確認するイメージを図 5.11 に示す。

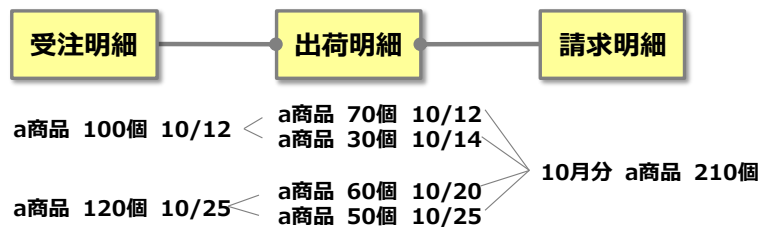


図 5.11 概念データモデルの補足説明（インスタンス）

この例では「一回の受注で受けた a 商品が分割されて出荷され、出荷実績は月別に集計して請求する。しかし 120 個注文しながら当月には 110 個しか出荷しないパターンがあり、そのパターンでは月別の受注と出荷、請求の数量は一致しない」という業務ルールを、インスタンス（実数値例）で表現するシーンを表している。

- 抽象化（最適化）を急がない
概念データモデルの理解を難しくしている原因の一つに、抽象化によるモデルの最適化という作業がある。実ビジネスを概念データモデルに正確に描き出すため

5.2 システム化要求の確認・評価(SR.2)

には避けて通れない作業であるが、その検討は難しく、できあがったものも業務部門には直感的に理解されにくいものになる。例を図 5.12 に示す。

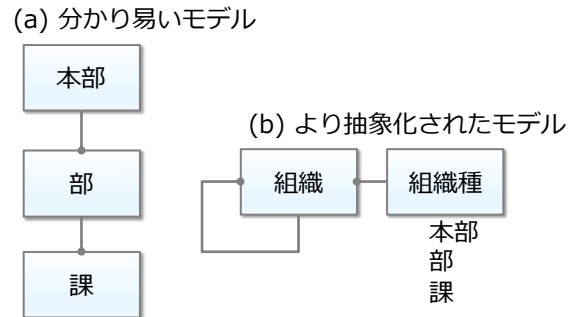


図 5.12 概念データモデルの抽象化レベルの違い

この例では、「本部」「部」「課」で階層化されている組織をモデル化している。(a) は組織の構造を実体ごとの縦の関係で表現しているのに対して、(b) は組織の構造を繰り返して表現し、各組織が本部・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ており、(b) の方が組織変更に強いモデルになっている。概念データモデルの検討においては、(a) を (b) にするよりも先に、(a) の状態で業務部門とモデルを確認してから抽象化を実施すると良い。

5.2 システム化要求の確認・評価(SR.2)

第6章 要件定義マネジメント(RM)における問題と解決の勘どころ

「RM.要件定義マネジメント」では、大きく4つのサブプロセスに分けて作業を行う。まず、「RM1.立ち上げ」において、要件定義工程前に策定した構想・企画を確認の上、当該プロジェクトを進める上でのステークホルダを特定する。そして、当該プロジェクトの成果に対する責任の所在を明確にすべく、成果のオーナーを選定し、オーナーより要件定義工程を開始する認可を得る。続いて、「RM2.計画立案」において、プロジェクトの目標達成に向け、プロジェクトの範囲を定義し、プロジェクト計画を精緻化した上で、体制の構築、立ち上げを行う。また、「RM3.監視・コントロール」において、プロジェクトの進捗やパフォーマンスを監視し、計画の変更が必要な事柄を特定し、適切に変更を行う。最後に、「RM4. 終結」において、要件定義工程の成果を評価し、当該工程の完了を判断する。

「RM.要件定義マネジメント」における課題カテゴリマップを図6.1に示す。



図 6.1 「RM.要件定義マネジメント」の課題カテゴリマップ

6.1 立上げ(RM.1)

6.1.1 構想・企画の確認(RM.1.1)

要件定義はゼロから始まるわけではない。要件定義を開始する際には、そのインプットとしてシステム化の企画段階で構想された内容（目的、期待効果など）を確認することから始まる。ここでは、要件定義の開始に際してそのインプットから、何を目的にシステム化するのか、正しく理解するためには何に注意しなければならないか、内容を理解した結果どうなることが望ましいのかについて述べる。

(1) 企画内容を確認するとともに、企画の目的とプロジェクト目標の目的意識を醸成する

【解決したい問題】

- システム化の目的があいまいで、何をゴールに要件定義すれば良いのか目標がはっきりしない
 - ・ システム化の目的やプロジェクト目標が達成できないシステムができあがる

要件定義の開始にあたり、最初にシステム企画の内容を確認する。しかし、例えば企画の目的が「サプライチェーン業務の見直し」といったようなあいまいなものだと、何のためにサプライチェーン業務を見直す必要があるのかが不明確となる。その結果、ステークホルダによって企画の目的やプロジェクト目標の捉え方に認識の違いが生じたり、プロジェクトマネージャの目的意識が希薄なまま要件定義を進めた場合、手戻りが発生するリスクが懸念される。そこで、企画内容を確認する際に、企画の目的やプロジェクト目標を捉えるポイントや、目的意識の醸成について説明する。

勘どころ① 企画内容を評価し、要件定義を開始できることを確認する

システム企画とは、経営課題の達成や業務改善に貢献する新規システムの構築や既存システムの改善を起案し、投資判断する役員や上級管理職から承認を得る一連の活動であり、システム構築の最上流工程にあたる。

企画の主な項目と確認内容を、表 6.1 に示す。

6.1 立上げ(RM.1)

表 6.1 企画内容の確認

項目	確認内容
(1) システム化の背景・目的	システム化の背景(システムがないことで発生している(発生する可能性のある)好ましくない状況)と目的(システム化によって実現したい状態)
(2) 問題・課題の内容	現行業務・システムの解決すべき問題や達成すべき課題
(3) 問題解決・課題達成のポイント	問題解決や課題達成のための業務・システムの改善内容や進め方のポイント
(4) 方策の立案と期待効果	方策となる新しい業務の仕組みやシステム化内容の立案とそれらの期待効果
(5) プロジェクト計画	作業タスク、推進体制、スケジュール、費用、人的リソースなどの方策の具体的な実行計画

システム企画のこれらの項目は、要件定義のインプット情報になることから、要件定義の開始にあたり関係者全員の共通認識となるよう読み合せをするとともに、認識に違いがある部分や疑問点があれば要件定義を始める前までに明確化する。

要件定義の開始判断は、通常、前工程(システム化企画)の最終段階で実施されていることから、このガイドでは触れないが、開始と決まった企画であっても、その内容確認は必須である。企画内容を評価した結果、不十分であると判断される場合は、無理に要件定義を開始せずに、構想・企画に差し戻し、再検討を依頼する。これは、不十分な企画内容のまま進めた場合、要件定義以降での手戻りや新たなリスク要因になるためである。

勘どころ② 企画の目的とプロジェクト目標を共有し、目的意識を醸成する

企画内容を確認する際、システム化の内容だけではなく、なぜこの企画に取り組む必要があるのか、プロジェクトのゴールは何かを理解・納得することが重要である。それにより、プロジェクトマネージャをはじめ関係者の目的意識が醸成され、取り組みの動機付けとなる。

企画の目的は、経営、顧客、業務部門、システム部門のそれぞれの立場・視点によって異なるため、例えば、図 6.2 に示すようなバランスト・スコアカード(BSC 戦略マップ)等を使って整理する。

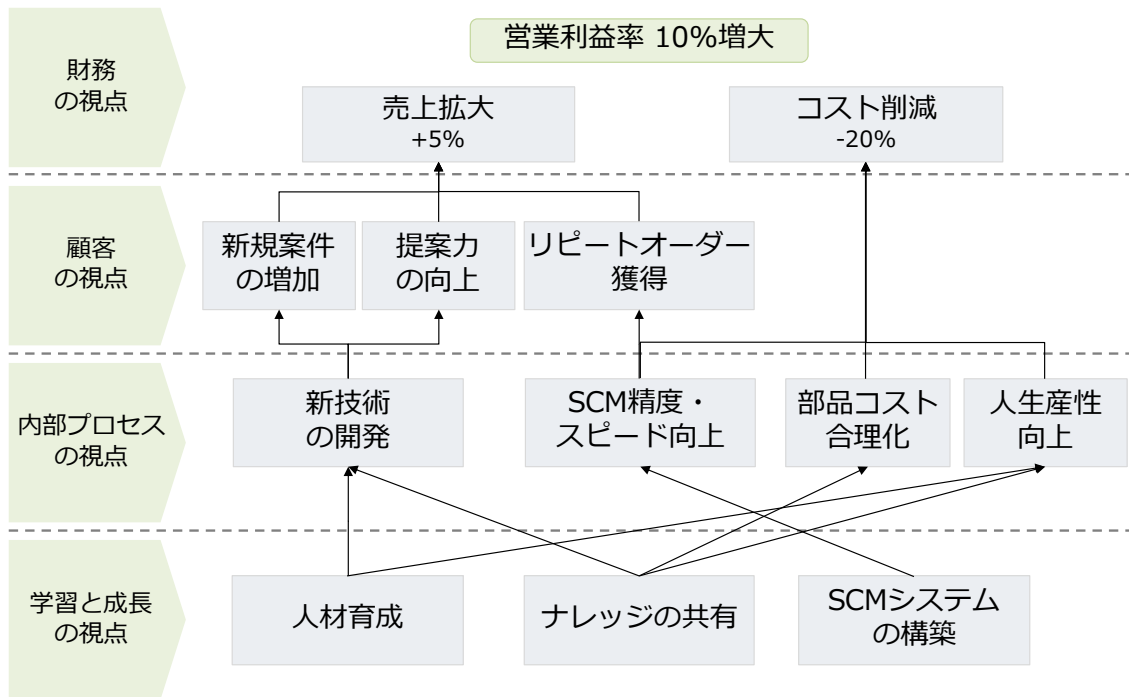


図 6.2 バランスド・スコアカードによる企画目的の整理

バランスド・スコアカードでは、「財務の視点」「顧客の視点」「内部プロセスの視点」「学習と成長の視点」の4つの視点で整理する。表 6.2 に、各視点の内容について説明する。

表 6.2 バランスド・スコアカードの4つの視点

視点	説明
財務の視点	組織目標に掲げられている財務的目標数値に直結する戦略であり、顧客、内部プロセス、学習と成長の3つの成果として最終的に繋がってくる。 利益性（収益向上）、成長性（新しい収益の源泉）、生産性（コスト削減、投入リソースの有効活用）等で整理することが多い。
顧客の視点	顧客からの評価（満足度、ロイヤリティ、サービス）、顧客へ提供する価値（価格、製品、品質、機能）等で整理することが多い。
内部プロセスの視点	革新的な新製品・新サービス、顧客価値の向上、業務品質向上等で整理することが多い。
学習と成長の視点	コンピタンス（人材・スキル、知識の共有）、技術（ビジネスモデル、情報技術）、組織風土（社風、動機付け、モラル）等で整理することが多い。

6.1 立上げ(RM.1)

このように、目的を4つの階層で確認する狙いは2つある。

1つは、「財務の視点」や「顧客の視点」を理解することで、システム化の重要性や緊急性を理解することである。それによって、システム化に取り組む必要があるのかどうか、取り組む必要があるならどの程度のコストや期間をかけるべきかを適切に判断できるようになる。

もう1つの狙いは、利用部門の考える解決策が「財務の視点」や「顧客の視点」に対して効果的で実現性の高いものかどうかを理解するためである。

また、上記のような企画の目的に加えて、表 6.1 の(5)解決策のプロジェクト計画に記載されたプロジェクト目標（スケジュール、費用、人的リソース）についてもステークホルダで共有し、目的意識を醸成する。その際には、企画書をただ配布するだけでなく、関係者が全員参加のキックオフ会議を開催し、システムオーナーやプロジェクトマネージャ等の責任者から直接説明する機会を設け、企画内容に込められた想いを直接ステークホルダへ伝えることが有効である。

事例 6 要件定義につながるシステム化構想・企画の策定 (セイコーエプソン株式会社)

【取り組みの背景】

システム化構想・企画では、業務・システムに関する経営課題や目指す姿などに加え、対象範囲、スケジュール、費用などのプロジェクト計画が策定され、「システム企画書」として取りまとめる。取りまとめたシステム企画書は、役員や上級管理職から必要な承認を得るとともに、プロジェクトとしてQCD（品質・コスト・納期）に対するコミットとなる。

しかし、後続の要件定義における対象範囲の拡大、要求の抜け・漏れ・誤りによる手戻りや想定外の工数増加により、納期遅延や費用超過が発生し、その結果、当初のプロジェクト計画を達成できない場合がある。

そのような問題を未然に防止するために、要件定義につながるシステム化構想・企画の品質向上を図るとともに、要件定義を開始して良いかどうかの判定ゲートを新たに設定することとした。

【課題】

本事例では、解決すべき課題を以下のとおり設定した。

- システム化構想・企画の内容が不明確なため、当初のプロジェクト計画が達成できない
- システム化構想・企画がどこまでできていれば要件定義を開始できるかの前提条件が不明確なため、内容が不十分なまま要件定義が開始される

【解決策】

(1) システム化構想・企画時の実施内容の明確化

システム化構想・企画に必要な実施項目を以下に6項目挙げる。

- ① システム化の背景・目的の明示
- ② 目的を実現するための課題設定
- ③ 業務改革・システム化による打ち手の立案
- ④ システムの実現方式の提案
- ⑤ プロジェクト計画（作業項目、推進体制、費用、スケジュール、課題・リスク等）の策定
- ⑥ 費用対効果の算出

上記の実施項目のうち、①～③は「何のために何をシステム化するか（目的と対

6.1 立上げ(RM.1)

象)」、④～⑤は「それをどうやってシステム化するか(実現手段)」、⑥は「その結果、どのように企業価値向上に寄与するのか(費用対効果)」について検討するための実施項目である。各実施項目の内容は以下のとおりである。

- ① システム化の背景・目的の明示
ステークホルダの要望やニーズをとりまとめた上で、なぜシステム化が必要なのかその理由と、システム化によって実現したい状態を明確にする。
- ② 目的を実現するための課題設定
システム化の目的の実現を阻害する現状業務の解決すべき問題や、よりレベルの高い業務を達成するための課題を対象範囲とともに明確にする。そのために、現状業務とあるべき業務との品質・スピード・生産性などのギャップを捉えるとともに、その要因となっている業務プロセスや制度といった業務の仕組みに関する課題も明確にする。
- ③ 業務改革・システム化による打ち手の立案
新業務の仕組みと新システムの有効性や実現性を明確にする。システム化にあたっては、まず業務の標準化・最適化などの業務改革に取り組む。その際には、業務内容の変化点と改善点を明確にするとともに、業務の中でのシステムの担当する対象範囲と役割を明記する。
- ④ システムの実現方式の提案
システムを具体化するためのソリューションを検討する。システムに求められる機能要件・非機能要件に合わせて、スクラッチ開発・パッケージ導入・クラウド活用等の適切なソリューションを選定する。
- ⑤ プロジェクト計画の策定
打ち手の立案とシステムの実現方式の提案を実行するために必要な作業項目や推進体制、費用、スケジュール、推進上の課題やリスク等のプロジェクト計画を明確化する。
- ⑥ 費用対効果
プロジェクト計画を実行した場合に期待できる ROI を試算する。売上拡大や費用削減、工数削減といった期待効果に対して、プロジェクト計画の費用をもとに ROI を算出し、3年以内に投資回収できるかといった指標で評価する。ま

た、遵法対応やリスク、組織力強化といった数値化しづらい定性効果を盛り込む場合もある。

システム化構想・企画では、これらの6項目を盛り込んだ「システム企画書」を作成し、業務部門、システム部門、システムオーナー、ステアリングコミッティ等のステークホルダに対して提案し、提案内容の承認を得る。

(2) 要件定義の開始条件明確化とレビューポイントの追加

要件定義の開始にあたり、システム化構想・企画の成果物である「システム企画書」に対し、以下のような観点で評価を行う。

- ① 経営課題や業務部門の問題解決・課題達成に対して、システム化の目的が合っているか（合目的性）
- ② 課題設定やその打ち手が漏れなく検討されており、合理的、かつ十分であるか（網羅性・十分性）
- ③ システムの実現方式やプロジェクト計画が実現可能な内容か（実現可能性）
- ④ 投入する費用に対して、十分な効果が得られるか（費用対効果）

上記は、全社最適化や経営視点に立った見地であるため、必要に応じて経営層の確認も求められる。システム企画は現場の困りごとの解決だけではなく、経営課題が達成できるかどうかの視点が重要である。その視点が欠けていると、後続の要件定義をいくら緻密に実施しても、「間違っただけを正しく作る」こととなり、プロジェクト後半に大きな手戻りとなりかねない。

また、対象範囲があいまいな場合や、実現不可能なプロジェクト計画の場合、納期遅延や費用超過に直結し、プロジェクトのQCDが達成できない。そのようなことを未然に防止するためにも、システム化構想・企画の慎重な評価が求められる。

そのため、要件定義の開始にあたっては、社内規定を通してシステム企画書が評価・承認されるようデザインレビューによるレビューポイントを新たに追加し、不十分な企画のまま要件定義が開始されることを未然に防止するようにした。

【効果】

本事例では、システム化構想・企画の実施内容の明確化と、要件定義の開始条件明確化とレビューポイントの追加により、要件定義につながるシステム化構想・企画の品質を向上することができた。その結果、納期遅延や費用超過等を抑制し、プロジェクトのQCDを確保することで、当初の目的・計画に沿ったプロジェクトの遂行ができるようになった。

6.1 立上げ(RM.1)

(2) 構想・企画の実現には、どの業務のビジネス要求が必要かを見定める

【解決したい問題】

- 構想・企画で決めたスコープがあいまいで、要件定義計画立案時に不備が発生する
 - ・ 要件定義が終わらない
 - ・ 後工程で手戻りが発生する

プロジェクトの失敗の大きな要因の一つが要件定義であるといわれている。要件定義において何を実現すべきかを定義せずに、システム化要求を決めようとする、要求事項が膨れ上がったりするため、いつまでたっても要件定義が終わらない、要件定義が不十分なまま、時間の制約により、設計・開発工程に入ってしまうなどのことが発生する。そのため、結果的に追加開発や手戻りなどが発生し、納期やコストに影響が出てしまう場合があり、プロジェクトが失敗してしまう。

プロジェクトを成功させるために、まずは、要件定義工程の立ち上げの段階において、しっかりと要件定義のスコープを確定させ、システム化する箇所を明確にしたい。

要件定義を実施する際には、まず対象の業務の範囲（スコープ）を確認する。スコープ定義とは、これから要件定義を実施する対象となる活動の実施範囲を示す定義であり、要件定義における諸活動の対象は、決定されたスコープ内に限定される。スコープの決定は、以降の活動のスケジュールや活動に投入する資源（ヒト・モノ・カネ）に影響を及ぼすものであることから、スコープを変更する際には相応の慎重さを求められる。

この項では、これから要件定義を行うにあたり、構想・企画の段階で決定されたスコープ（ビジネス要求スコープ、システム化要求スコープの両方）を確認する際の問題と対策について説明する。

勘どころ① プロジェクトの目標を実現するための要求事項を具体的に定義する

プロジェクトの目標を実現するための対象業務を確認する。本来、構想・企画の段階で決められるべき事項ではあるが、あいまいなまま要件定義が開始される場合がよくある。したがって、要件定義の立ち上げのタイミングで、プロジェクトの目標を実現するために必要な要求事項に対する認識をステークホルダと共有することが大事である。

要求事項については、一覧表などにまとめ、プロジェクトの目標との関連が分かるように記載すると良い。要求事項を整理するために、大項目、中項目、小項目など要求の内容を階層的に詳細化し、一覧化すると、実際に要件定義工程を進めていくうえで、システム化するかどうか判断しやすくなる。一覧表には、優先順位付けができるような項目も入れ

とくと良い。以下のフォーマットでは「優先順位」で表している。

さらに可能であれば、要件定義にかかる工数の見積りをすると良い。これは、この後計画を立案するうえで、工数・期間を算出していくうえでのインプット情報となる。

表 6.3 要求事項一覧フォーマット

		ユーザー記入				ベンダ記入		
#		要求事項			理由	優先順位	要求元	見積り
		大項目	中項目	小項目				
1	0	0						
1	1	0						
1	1	1				大		
1	2	0						
1	2	1				中		

要求事項をまとめたら、ユーザに漏れなく抽出できているのか確認を取る。ユーザは、要求がすべて洗い出されているのか確認する。ポイントは、システム化するかどうかは別として、要求事項がすべてあげられているのかを確認することである。必要であれば、システム化の対象となる業務現行の業務フローが最新性を保っているなら、それらと照らし合わせるなどして、要求事項が洗い出されているのか、関係者と確認を取るべきである。

勘どころ② 全体像を俯瞰し、要件定義の範囲が妥当であるかを確認する

最後に、要件定義の範囲が決まったら、プロジェクトで実現する要件定義の範囲が現実的であるのか、検証する。俯瞰的にみて本当にシステム化の検討の必要があるのかを確認する。プロジェクトの全体像を確認することで、システム化しない方が総合的にみて現実解になっている場合もあり得る。また、要件定義の範囲を決めるにあたり、前提条件なども決めておく。

これらの作業は、プロジェクトの目標をよく理解したキーマンを参画させて、合意形成しながら進めると良い。限定されたメンバだけでシステム化の範囲を決めてしまうと、判断を誤り、範囲に過不足が発生する可能性が出てくる。また、プロジェクト全体で合意形成しているという形も取れるので、キーマンの参画は不可欠である。

6.1 立上げ(RM.1)

6.1.2 ステークホルダの特定(RM.1.2)

要件定義の最初の段階で実施すべきことに、ステークホルダ（利害関係者）の洗い出しがある。ステークホルダとは、企業の経営活動に関わる利害関係者のことで、具体的には消費者（顧客）、従業員、株主、取引先、地域社会などである。昨今のシステム開発ではステークホルダが多様化している。要求を抽出するためだけでなく、合意を形成するためにも、新しいビジネスを創出するためにも、まずステークホルダを漏れなく洗い出さなければならない。

ステークホルダの洗い出しが不十分だった場合、そのステークホルダから本来引き出せたはずの要求が漏れ、最悪、システムの稼働後にそれが判明して、システムの利用価値を大きく損なうような事態にもなりうる。

(1) 必要なステークホルダを特定・分析し、各ステークホルダへの対応方針を立案する

【解決したい問題】

- ステークホルダに漏れがあり、必要な要求を抽出できない
 - ・ 後工程で手戻りが発生する
 - ・ ステークホルダが使えない役に立たないシステムができあがる

要件定義後の開発工程において、要件変更はプロジェクトの QCD に大きな影響を与える。

そのような要件変更は、要件定義段階で識別していなかったステークホルダから発生することもある。これは、プロジェクトの大規模化・複雑化にともない、プロジェクトのステークホルダも多様化し、かつ数が多いことに起因する。

したがって、関係するステークホルダを洗い出すことが、成功への第一歩となる。

また、ステークホルダを洗い出せたとしても、ステークホルダの問題認識に対し、適切な課題設定や解決策を提示できないために合意形成に時間がかかったり、ステークホルダ間での調整に時間がかかったりして、要件定義の期間を守ることができないケースも多い。したがって、必要十分なステークホルダを洗い出した上で、それぞれのステークホルダ自体とステークホルダ間の関連を分析し、プロジェクトへの影響度、関心度といった各ステークホルダの特性に合わせた対応方針を立案する必要がある。

上記のアプローチを取ることにより、ステークホルダそれぞれが腹に落ちた要求として定義でき、スムーズに要求の合意形成を進められるだけでなく、開発工程における要件変更リスクを低減することができる。

勘どころ① 必要なステークホルダを漏れなく抽出する

ステークホルダを洗い出す際、目の前にある対象とする業務やシステムに関係する直接的なステークホルダのみに関心が向き、視野が狭くなってしまうことが多い。端緒としては良いが、対象業務やシステムとの間で間接的に影響を受ける／影響を与える関係者もステークホルダとして抽出する必要がある。

また、現場の業務担当者だけでは組織としてのゴールや経営課題が見えていない可能性があり、表面的な意見や個人的な意見が混じる可能性がある。そのため、経営層もステークホルダとして識別する必要がある。

以下に抽出の観点を示す。

- 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

上記はあくまで一例であるため、プロジェクト関係者へのインタビューやブレインストーミングなどを使用して、漏れなくステークホルダを抽出する必要がある。

勘どころ② ステークホルダを分析し、対応方針を立案する

漏れなく抽出をしても各ステークホルダに合った対応をしなければ、限りある時間を無駄に浪費してしまう可能性がある。例えば、プロジェクトへの影響度も高くなく、関心度も低いステークホルダに熱心に対応したとしても、影響度、関心度ともに高いステークホルダの一声で決定事項が覆ってしまう事態も発生しかねない。

したがって、プロジェクトへの影響度と関心度の程度を各ステークホルダに対して評価し、その特性に応じた対応を取る必要がある。図 6.3 が、影響度、関心度による対応例である。

6.1 立上げ(RM.1)

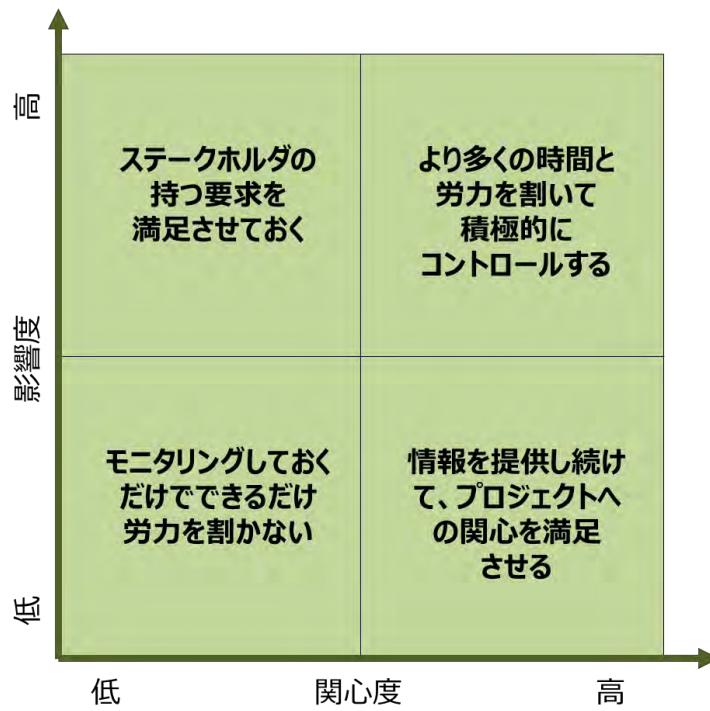


図 6.3 ステークホルダ分析

コラム 7 要件の種類と要求者

要件定義においては、ビジネス要求やシステム化要求を適切に抽出することが最も重要である。そのためには、抽出すべき要求を正しく認識している要求者（ステークホルダ）を認識し、それらの要求者にプロジェクトへの参加を求めることが不可欠である。

ビジネス要求やシステム化要求において定義する各種の要件と、それをどの要求者から引き出すかを組み合わせた例を以下に示す(表 6.4)。

要件定義においては、引き出したい要求が、どの要求者からなら最も正確に抽出できるかを的確に判断し、必要な人員を漏れなくプロジェクトに巻き込むことが、成功のカギになる。

表 6.4 要求の種類と要求者

No.	要件の種類	主な要求者	説明
1	経営要件	経営者・部長	ビジネス（経営）の観点での要件 <ul style="list-style-type: none"> ・長期計画や戦略 ・売り上げ、利益、能率等の KPI に関わるもの ・法律等の制約 ・コンプライアンスに関わる事項 ・稼働時期、投資予算 など
2	業務要件	課長・リーダー	業務遂行上の要件 <ul style="list-style-type: none"> ・業務手順 ・改善された業務フロー ・出張先から交通費の精算ができる ・1時間で60枚のFAX注文を入力できる など
3	UI 要件	利用者	画面・帳票等のユーザインターフェースに関する要件 <ul style="list-style-type: none"> ・操作性、理解性、習得性、業務効率に関するもの ・画面デザインに関するもの ・操作環境（主張先、工場環境、ラベルプリンタの設置場所） ・スマートデバイスの利用 ・多国語対応 など
4	外部 I/F 要件	関連システム	システム間連携等のやり取りに関する要件 <ul style="list-style-type: none"> ・通信手段（ツール、プロトコル）、通信頻度 など

6.1 立上げ(RM.1)

No.	要件の種類	主な要求者	説明
5	システム要件	全参加者	業務要件を満たすために、完成したシステムが満たすべき条件 ※他の要件の要求者から、欲しい機能を直接要求されることがあるが、それをそのまま受け入れずに、業務上それがなぜ必要なのか、他にもっと優れた解決策はないのかを確認することが重要
6	プロジェクト要件	上位層	プロジェクトを進めるための制約等 ・開発予算、納期 ・開発体制、要員 ・開発環境（オフィス、サーバー、ツール等） ・報告
7	運用要件	運用組織	運用組織がシステムを受け入れる条件 ・バックアップ取得の容易性 ・保守の容易性 ・リリースの容易性 ・運用ドキュメントの充実度

事例 7 漏れのないステークホルダの特定と分析 (セイコーエプソン株式会社)

【取り組みの背景】

セイコーエプソンでは、事業環境の変化やメガトレンドなどを踏まえ、長期ビジョン「Epson 25」を制定した。この「Epson 25」実現に向け、IT 基盤をグローバルに統一・強化するミッションに取り組んでいる。

その要となる情報システムについては、業務オペレーションの全体最適化に向け、社内 IT 環境と業務システムの計画的な企画・設計・導入に取り組んでいる。本取り組みでは、これまで事業ごとに使用してきた異なるシステムを、業務領域を横断して共通化・標準化する必要がある。そのため、既存システムのような単一事業に関わるステークホルダのみならず、各事業間や関係会社、経営層、取引先も含めた広範囲なステークホルダと合意形成を図っていく必要がある。

本事例は、そのような大規模化・複雑化する IT システムに関係するステークホルダを漏れなく特定・分析するかについて、弊社の取り組みを紹介する。

【課題】

本事例では、解決すべき課題を以下のとおり設定した。

- ステークホルダを漏れなく特定できない

【解決策】

(1) ステークホルダの特定

多様化するステークホルダと合意形成を図るために、まず、ステークホルダの特定をする。ステークホルダとは、開発対象のシステムから直接、または間接的に便益を得る組織や人であり、プロジェクト発足時（企画段階）の活動の中で特定する。ステークホルダの特定に当たっては、対象のシステム開発に関係するステークホルダを漏れなく特定（識別）することが重要である。プロジェクトを取り巻くステークホルダの一例を以下に示す（図 6.4）。

ステークホルダの特定や分析が不十分な場合、以下のような問題が発生する。

- キーパーソンでないステークホルダへのヒアリング実施による変更要求の手戻り発生
- ステークホルダの特定漏れによる要求（機能）漏れの発生

上記のような問題を未然に防止するためには、要求の源泉となるステークホルダを特定・分析し、合意されたステークホルダを中心に要件定義の実施、ならびに合

6.1 立上げ(RM.1)

意形成を図ることが重要である。ステークホルダの特定では、対象システムがどのステークホルダに対して価値・サービスを提供するのかに着目して、漏れのないようにステークホルダの特定を行う。その際には、対象のシステムによって、価値・サービスを受ける直接的な受益者だけではなく、間接的に影響を受ける関係者についても、漏れなく特定する。

どのステークホルダが、価値・サービスを直接的に受ける受益者なのか、または間接的に影響を受ける関係者なのかは、対象システムによって異なるが、その一例を以下に示す（表 6.5）。

例えば、会計システムの場合、経営判断に必要な会計情報をマネジメントコックピットとして提供する。その場合、価値・サービスを直接的に受ける受益者は経営層であり、間接的に影響を受ける関係者は各事業・関係会社や事業管理部門となる。また、会計システム構築に関わるプロジェクト関係者として、システム部門やベンダを抽出する。

さらに、場合によっては、事業管理部門と連携して業務を行うステークホルダ（生産管理部門など）や、事業管理部門に情報提供などの支援をするステークホルダ（経理部門など）もステークホルダとして特定した方が良い場合もある。

最初は、ステークホルダの漏れを防ぐ観点で広めにステークホルダを特定し、プロジェクト体制の整備や要求を明確化する中で、実際にどの範囲のステークホルダまでをヒアリングや合意形成の対象とする必要があるかを判断すると良い。

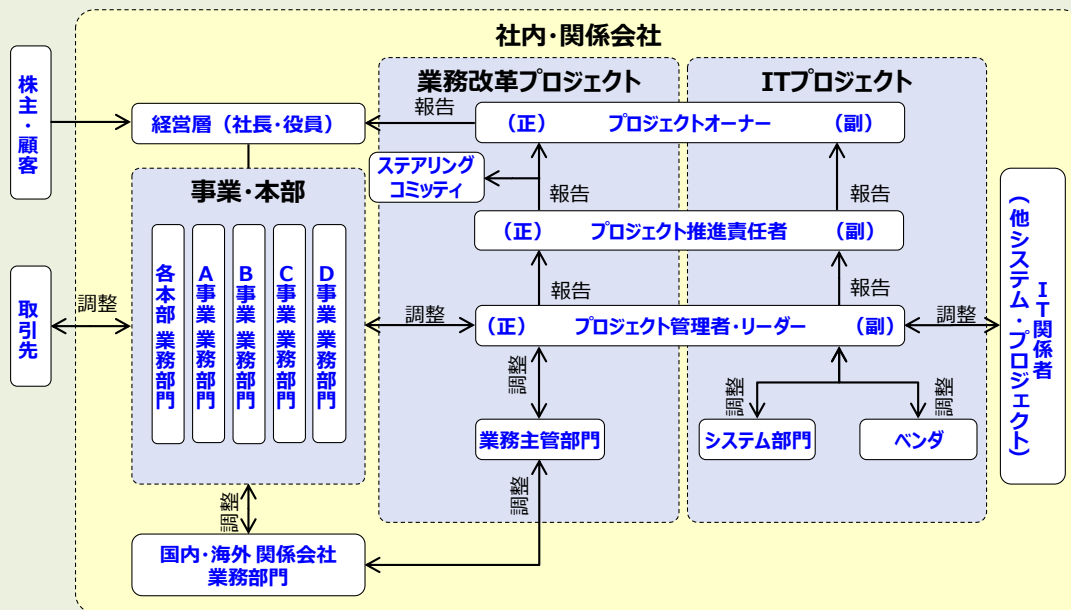


図 6.4 プロジェクトを取り巻くステークホルダ

表 6.5 ステークホルダの特定

対象システム ステークホルダ	会計 システム	生産管理 システム	販売 Web システム	本社・アドミ 系システム	文書管理 システム
顧客	—	—	◎	—	—
経営層	◎	△	△	△	◎
各事業・関係会社	△	△	△	△	◎
業務部門	△	◎	◎	◎	◎
利用部門	—	△	—	◎	◎
システム部門	□	□	□	□	□
ベンダ ※委託ありの場合	□	□	□	□	□
取引先	—	◎	△	△	△

(凡例) ◎：価値・サービスを直接的に受ける受益者

△：間接的に影響を受けるステークホルダ

□：プロジェクト関係者

—：概ね関係ないステークホルダ

(2) ステークホルダの分析

ステークホルダを特定する中で、「ステークホルダリスト」を作成し、ステークホルダを分析する（表 6.6）。ステークホルダリストでは、組織／個人の視点で、次のような事項を明確にする。

- 組織の視点
- 会社／事業部／部門：社内外のステークホルダの組織名
- ステークホルダ：ステークホルダの種類（顧客、経営層、各事業・関係会社、業務部門、利用部門、システム部門、ベンダ、取引先など）
- 個人の視点
- 氏名：個人名（キーパーソンや役職者中心にヒアリングや合意形成の対象者を絞り込む）
- 役職：役職名（役員、部長、課長など）
- 役割：プロジェクトでの役割（業務部門担当者、システム部門担当者、プロジェクト管理者、プロジェクト推進責任者、ステアリングコミッティ、システムオーナーなど）
- 影響度：プロジェクト活動における影響度合い
- 対立関係：ステークホルダ間のコンフリクト

6.1 立上げ(RM.1)

表 6.6 ステークホルダリスト

組織		個人				
会社／事業部／部門	ステークホルダ	氏名	役職	役割	影響度	対立関係

前記事項のうち、「役割」、「影響度」、「対立関係」は特に重要な事項である。

役割については、実行担当者、意思決定者、相談者、報告者に分類し、合意形成を図る手順に活用する。

例えば、大きな業務変革を伴う場合、まず、相談者である社内外の有識者に意見を聞いた上で、意思決定者であるプロジェクト推進責任者やステアリングコミッティに説明を行い、方向性について同意を取り付ける。次に、実行担当者である業務部門担当者に説明し、報告者であるシステムオーナーに進捗状況を報告することで、円滑な合意形成を図ることができる。なお、これらの役割は、ステークホルダの特定・分析の段階ではまだ決まっていないことも多いが、プロジェクト体制を整理する中で明確にする。

また、影響度については、ステークホルダの発言や行動によって影響を受ける可能性で判断し（プロジェクトが円滑に進む良い影響の場合と、悪い影響の場合がある）、その度合いによって、プロジェクトへの巻き込み方やステークホルダへの接し方（事前確認を行うなど）を変えるようにする。

【効果】

本事例では、この施策の実施により、ステークホルダの漏れを未然に防止し、ステークホルダとのコミュニケーション計画の策定に効果を得ることができた。

6.1.3 オーナーの選定(RM1.3)

情報システムは誰の持ち物か。ここで言う持ち物とは、それを使用して、そのシステムから期待どおりの業務上の効果を引き出す責任は誰にあるのか、ということであり、システム開発の費用を確保するのは誰か、という意味ではない。業務システムの開発・運用にかかるコストは、それを利用することによって発生する利益によって償却される。情報システムは、その利用者の要求を最大限実現する機能を提供する必要があるのと同時に、利用者は、自らの要求にしたがって構築された情報システムを使用して、想定どおり、またはそれ以上の利益を回収する責任を追う。

(1) システムオーナーの選定により、業務部門の役割と責任を明確にする

【解決したい問題】

- 要件定義は自分たちが責任を持って推進すべきものだと業務部門が認識していない
 - ・ 利用頻度に関わらず要求が出る
 - ・ 部門ごとにシステム化に対して温度差がある
 - ・ 実務で多忙なので IT システム部門にお任せ
 - ・ システム開発費用の確保は IT システム部門の役目

要件定義は、業務部門（構築されたシステムを利用する部門）からの要求に基づき実施される。業務部門からの業務機能に対する要求をもとにシステム化のための要求を定義し、システムを構築する流れになる。その際には、以下のような問題が往々にして発生する。

- 業務部門からの要求が大きくなり、それを実現するための費用が当初計画を上回る
- 利用頻度がそれほど多くない、または緊急性がそれほど高くないと思われる業務に対してもシステム化要求が止まらない
- 要件定義の現場に業務部門からの意見がなかなか集まらず、システム化の効果が高い要求が抽出できない
- システムを利用する部門も必ずしも一枚岩ではない。システムが完成したら得られる営業実績データを活用する商品企画部門の意見はシステム化に積極的、営業部門は及び腰、データを日々入力することになる営業事務部門は被害者意識のみ、などシステム化に対する各業務部門の思惑が分かれることにより要求が偏る。
- システム構築のテスト段階で業務部門の参加度合いが低く、業務部門が実際に必

6.1 立上げ(RM.1)

要とする機能が備わっていないことを、システムが本稼働する前に発見できない。

上記のような問題が出た場合、システム化を要求した利用部門において当初想定した利用価値を生み出せず、システム化投資がいつまでたっても回収できない、という結果を生み出してしまい、そのシステム開発は失敗という評価を背負うことになる。

このようなことを防止するためには、利用部門とシステム構築部門が以下のような事項について合意しておくことが必要になる。

勘どころ① システム構築投資を回収するのは業務部門の役目であることを明示する

システム開発は業務部門からの要求をもとに実施されるものである。したがって、業務部門からのシステム化要求が実現したら、それを活用して開発投資以上の成果を上げる責任は業務部門にある。要件定義を開始する際には、開発するシステムの仕様とシステム開発の投資対効果に最も責任をもつのはシステム部門ではなく、業務部門である。要求を提示した業務部門が、これから要件定義を行い、その結果開発されるシステムのオーナー（持ち主）であることを関係者に明示し、業務部門はそれを自覚した上で、自らの要求の投資対効果を自ら判断しなければならない。これに対して、システム開発部門は、要求された事項を過不足なく実現できるシステムを業務部門に約束した費用で開発することに責任を持つ。

勘どころ② 業務部門に要件定義の結果や開発されたシステムの仕様確認に責任を持つと認識させる

業務部門の要求にしたがって要件を定義し、それに基づいてシステム開発を行うのであるから、業務部門は要件定義の結果に責任を持つ。業務部門は、自らが利用するシステムが想定した効果を生み出せるよう、自ら積極的に要件定義に関わる責任がある。また、開発されたシステムは要求を形にしたものであることから、総合テスト段階においては、テスト対象のシステムが想定した効果を生み出せるかどうかを、自らの目で検証する責任がある。これらのことをプロジェクト発足時の役割分担等に明示する。

一般に、要件定義プロセスの役目は、要件を定義してシステム構築にその成果をつなぐことであり、それができたら要件定義の役目は終わりと考えがちである。しかし、要件定義の段階で想定した投資対効果がそのとおりに実現できたかどうかの評価は、開発したシステムが本稼働して業務部門がその利用を開始してからが始まりである。その点において、要件定義の終わりは、システム構築の終わりよりも後であり、当初描いていた効果

6.1 立上げ(RM.1)

がシステム本稼働後に発揮されたかどうかの評価を待つ必要がある、と言える。それを正しく評価できるのはシステム部門ではなく、当初の要求を提示した業務部門である。要件定義の結果をシステム本稼働後に評価し、以後のシステム化投資に反映するのは、業務部門の重要な責務である。

事例 8 アプリケーションオーナー制度創設による
要件定義へのユーザ部門参加の強化
(東京海上日動火災保険株式会社)

【取り組みの背景】

Windows95 の登場、PC の廉価化、インターネットの普及等をきっかけに、90 年代後半から企業の活動に IT が不可欠となるだけでなく、競争の激化にともない、商品の複雑化が進み、すべての商品がシステム開発を伴う形で出されることとなり、従来とはレベルの違う量のシステム開発が行われるようになってきた。

【課題】

しかしながら、急激に増大したシステム開発を成功させるためのポイントも十分掴めないまま実施したことにより、システムトラブルが多発し、企業の本来活動に支障をきたし、経営問題となった。

【原因】

コンサルタントも入れた分析の結果、システムトラブルの 8 割は、上流の要件定義局面でのコミュニケーション・ギャップから問題が生じていることが判明した。コンサルタントからは、以下のような問題事象も指摘された。

- ① ビジネス側の要件の確定が遅い（期日までに決めなくてはいけないというマインドが乏しい）。
- ② 要件の変更が多い。
- ③ 要件を最終的に文書で確認していない。
- ④ その要件が他システムにどのような影響を与えるかの分析が甘い。
- ⑤ 要件が設計に正しく反映されたかを複数の眼でチェックしていない。

④、⑤については、情シス部門内での仕事の進め方に関する問題であり、これらについても抜本的な改善が必要であるが、①～③については、事業部門が「システム開発を情シス部門の仕事」として任せきりにするのではなく、事業部門もシステム開発において一定の役割と責任を果たすようにしないと、本質的に解決できない問題であると考えた。

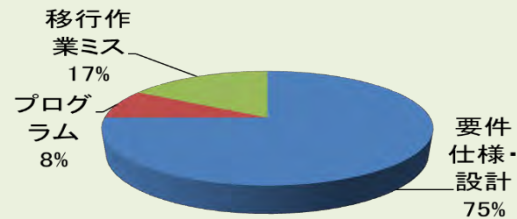


図 6.5 システムトラブルの原因
東京海上日動火災保険株式会社の例、1999 年

【対策】

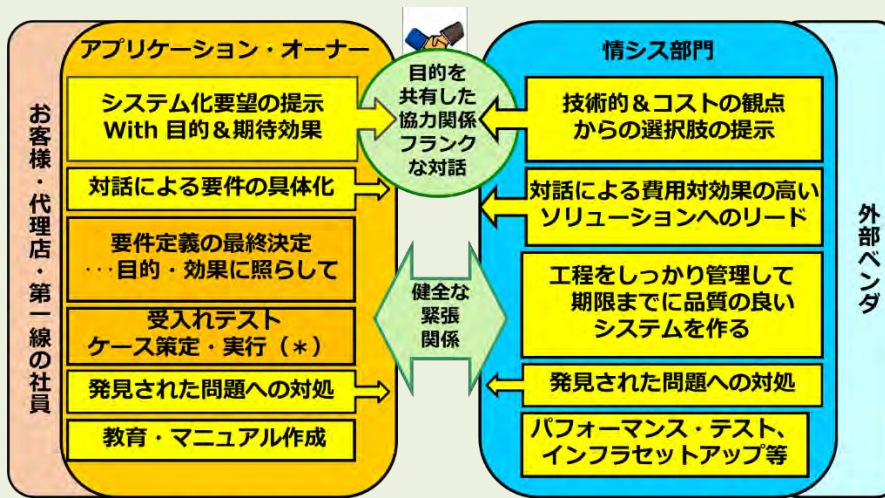
この問題を解決するために、システム開発におけるビジネスサイドの役割と責任を明確化し、コミュニケーションの質を高めることとした。

そのためには、お互いの理解を補正し合い、「言いたいこと」と「聞いて理解したこと」が同じになるまで、対等な立場でオープンなダイアログを繰り返すことができる「態勢」作りが必要である。この態勢を「アプリケーション・オーナー制度」と呼ぶ。アプリケーション・オーナー制度のポイントは以下のとおり。

- ① システム開発は、情シス部門に任せきりにすべき仕事ではなく、自分の考えた商品や施策を具体化するために行う自分自身の仕事であるという「オーナーシップ」の考え方を持たせる。
- ② 事業部門に、要件の詳細が固まるまで、情シス部門と対話を繰り返す責任を持たせ、要件定義の最終責任を負わせる。
- ③ 事業部門に、要件定義どおりにシステムができたかどうか受入れテストを実施する責任を負わせる。

「事業部門は要件定義に責任を持つこと」とするだけでは、表面的な責任にとどまり、要件の揺り戻し等の問題は解消しないので、要件を定義した以上、その要件どおりシステムができたかどうかを確認する UAT (User Acceptance Test、ユーザ受入れテスト) のテストケースを作ってテストすることが重要である。実際に手を動かす(ハンズオンの)責任にしない限り、本当に責任を取ることにならない。

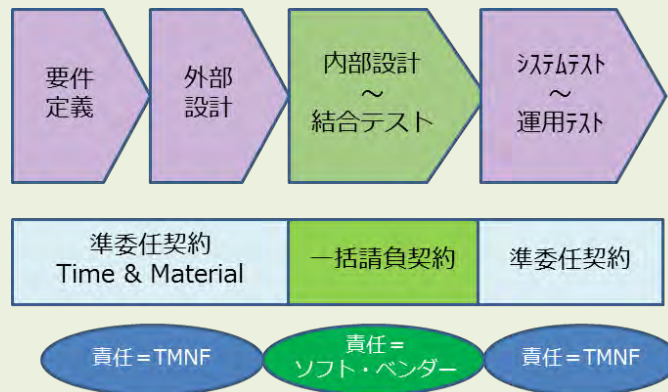
6.1 立上げ(RM.1)



* 要件定義に責任を持つ以上、要件通りできたかの受入れテストも実施することが重要。このように手を動かす責任にしない限り、表面的なものになる。

図 6.6 アプリケーション・オーナー制度：責任と役割分担
東京海上日動火災保険株式会社の例

ちなみに、この制度に基づく、東京海上日動火災保険株式会社の情シス部門とソフトベンダとの契約関係は以下のとおりである。この責任態勢は、アプリケーション・オーナーが要件定義～外部設計の工程、およびシステムテスト～運用テストの工程に責任をもつことで裏打ちされている。



※図中の TMNF (Tokio Marine Nichido Fire) とは、東京海上日動火災保険株式会社のこと

図 6.7 情シス部門とソフトベンダの契約関係

【効果】

2000年からアプリケーション・オーナー制度を導入したことにより、東京海上日動火災保険株式会社では2000年度から2001年度にかけてシステムトラブルが8割削減され、以降10数年、同じ水準を保っている（東京海上日動火災保険株式会社の例）。

■ 教訓

システム開発を情シス部門だけの仕事にせず、各事業部門が自分のこととして捉える「態勢」をつくるのが大切。因みに、「態勢」とは以下のとおり。

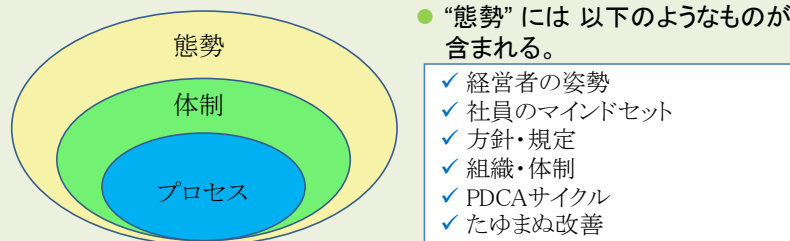


図 6.8 態勢・体制・プロセスの関係

東京海上日動火災保険株式会社から提供

この事例は、情報処理システム高信頼化教訓集 IT サービス編(IPA 発行) [26]に収録されている教訓(G01)「システム開発を情シス部門だけの仕事にせず、各事業部門が自分のこととして捉える「態勢」をつくるのが大切」から転載した。

6.2 計画立案(RM.2)

6.2.1 要件定義プロセス計画(RM.2.1)

要件定義には、後述するさまざまな作業プロセスが存在する。利用者からのさまざまな要求を正確に引き出すプロセス、引き出した結果をシステム上で実現するためにはどのような機能が必要とされるかを定義するプロセス、そしてそれらを目標とする期間内で予定した人員で円滑に完了させるコントロールプロセスが複雑に交差する。要件定義を開始する際には、まずこれらのプロセスをどのように関係付けるか、各プロセスの成果物をどの程度の粒度で記載し、後続のプロセスの成果物にどのように結果を反映していくかを計画しておく必要がある。プロセス計画は、要件定義の流れを決定付ける重要な作業であり、その精度は、その後の成果物の品質、作成の進捗に大きな影響を与える。

(1) 要件定義で必要なプロセスと成果物を見極め、要件定義の計画を立案する

【解決したい問題】

- 本来必要なプロセス・成果物を省いたため、要件定義起因の手戻りが発生する

システム化の企画がまとまり、問題点や課題・ニーズがおおよそ明らかになり、システム開発目的やスコープが提示され、主要な要求やビジネスプロセスの関連、業務フローが見えてきた段階で、要件定義プロセスが開始される。ところが、要件定義において、どの成果物をどのレベルまで作成するかを決定することは意外に難しい。例えば、業務フローは企画プロセスで作成すべきか要件定義プロセスで作成すべきか、システム化業務フローは要件定義プロセスで作成すべきか設計プロセスで作成すべきか、概念データモデルはどのプロセスで作成すべきか、など成果物によってはプロセスの線引きが難しいものがある。

これらの問題を解決する勘どころとして、以下の対策を実施する必要がある。

勘どころ① 前後のプロセスを確認し、要件定義のプロセスと必要な成果物、記載レベルを定義する

要件定義の成果物をプロセスにマッピングした例を表 6.7 に示す。同表は、要件定義プロセスで作成する成果物と上流工程の各プロセスが、どのように関係付けられている

6.2 計画立案(RM.2)

かを示している。ここでは構想立案で使用される SWOT 図や要件定義で使用されるステークホルダ関連図など、作成プロセスが明確なものは省略している。プロジェクトの状況や目的によって成果物の作成プロセスは異なるので、これは一つの例として参照いただきたい。

なお、表 6.7 に示すプロセスは、「共通フレーム 2013」 [27] が規定するものを用いている。共通フレームが規定する要件定義関連のプロセスと工程とのマッピングについては、付録 1 を参照願いたい。また、システム要件定義の成果物については、「機能要件の合意形成ガイド」 [28] にもそのコツがまとめられている。その一覧を付録 2 に示す。

表 6.7 プロセスへの成果物のマッピング例

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
構想立案	システム化計画立案		(外部設計: ※合意形成ガイド)
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
外部システム関連図	外部システム関連図	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	外部インターフェース一覧	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
	帳票一覧	帳票一覧	帳票一覧
		帳票レイアウト	帳票レイアウト
			帳票項目定義
		帳票概要	帳票概要
			帳票編集定義
		データ項目定義書	データ項目定義書
		ドメイン一覧/定義	ドメイン一覧/定義
		コード一覧/定義	コード一覧/定義

6.2 計画立案(RM.2)

表 6.7 は、要件定義の各成果物は複数のプロセスにおいて、それぞれ異なる目的のために作成される可能性があることを示している。使用の目的が異なれば、どのプロセスで、どのレベル（範囲、深さなど）で作成するかが変化する。成果物の記載サンプルを活用して、記載の範囲・深さを合意することがよく行われる。成果物フォーマットは合意していても、「どこまで書くのか」後になってもめることが多いからである。要件定義を実施する際には、前後のプロセスの目的を踏まえて成果物を決定することが重要である。個々の成果物のプロセス別目的は、7章に記載しているため参照されたい。また、要件定義工程を終結できるよう「〇〇の成果物を完成しておく」などの完了基準を設け、完了判断できるようにしておく（詳細は、「6.4.2 完了判断」参照のこと）。

勘どころ② 合意形成プロセスを意識して計画を立案する

要件定義中に作成する成果物を決めたら、どのステークホルダの誰と合意したらその成果物は完成と言えるのか、あらかじめ確認して決めておくべきである。合意する相手は必ずしも一人とは限らない。むしろ複数の利用部門の合意を取るケースが昨今増えてきている。合意する相手を洗い出し、合意の順番や段取りを計画するべきである。

具体的な計画立案方法は、次項 6.2.2 スケジュール作成の「勘どころ②所要時間に大きな影響を与えるレビュー、部門間調整、承認に要する期間をあらかじめ織り込んでおく」や、6.2.6 コミュニケーション計画の「勘どころ②コミュニケーションの質を上げ、合意形成を図る」を参照されたい。

勘どころ③ 成果物作成の担当を決める

成果物の作成にあたっては、どの部門が主体となって作成するかを明確にすることが重要である。要件定義はステークホルダの要求仕様を要件として定義する作業であり、成果物に記載された内容に対する責任は、ユーザ企業の利用部門（オーナー部門）などのステークホルダが負うのが原則であることから、作成もそれらの部門が主体的に実施することが理想である。

しかしながら、利用部門が要件定義をすべて実施することは、技術的にも時間的にも困難である。また、昨今の業務は IT 抜きには改革できないことから、利用部門、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あって成果物を作成することが現実的である。

主要な要件定義成果物と作成時の役割分担の案を例示しておく（表 6.8、表 6.9）。どの成果物を利用部門やシステム部門が主体で作成するかは、プロジェクトの状況に応じて異なるので、要件定義プロセス計画時に明確にしておく。

表 6.8 「ビジネス要求定義 (BR)」と作成時の役割分担

項番	成果物	役割分担				補足
		ユーザ			ベンダ	
		業務部門	開発部門	運用部門		
7.1.1	ビジネスコンセプト確認ドキュメント	○	△	△	△	
7.2.1	ステークホルダ関連図	○	△	△	△	
7.2.2	ステークホルダー一覧	○	△	△	△	
7.2.3	リッチピクチャ	○	△	△	△	
7.3.1	問題・ニーズ・課題一覧	○	△	△	△	
7.3.2	問題原因分析図	○	△	△	△	
7.3.3	要求構造図	○	△	△	△	
7.3.4	要求一覧	○	△	△	△	
7.4.1	管理対象分類図	○	△	△	△	
7.4.2	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 (5.2.2 妥当性確認を参照)を確認する。
7.5.1	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。
7.5.3	ビジネスプロセスフロー (業務フロー)	○	△	△	△	
7.5.4	ビジネスプロセスフロー (システム化業務フロー)	○	△	△	△	
7.5.2	業務機能構成表	○	△	△	△	
7.5.5	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。
7.6.1	状態遷移図	△	○	△	△	
7.7.1	業務用語定義書	○	△	△	△	
7.7.2	ビフォーアフター図	○	△	△	△	

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)

6.2 計画立案(RM.2)

表 6.9 「システム化要求定義 (SR)」と作成時の役割分担

項番	成果物	役割分担				補足
		ユーザ			ベンダ	
		業務部門	開発部門	運用部門		
7.8.1	システム化業務一覧	△	○	△	△	
7.8.2	画面一覧	△	○	△	△	
7.8.3	帳票一覧	△	○	△	△	
7.8.4	外部インターフェース一覧	△	○	△	△	
7.8.5	エンティティ一覧	△	○	△	△	
7.9.1	システム化要求仕様	△	○	△	△	
7.9.2	UI標準	△	○	△	△	
7.9.3	画面遷移図	△	○	△	△	
7.9.4	画面レイアウト	△	○	△	△	
7.9.5	帳票レイアウト	△	○	△	△	
7.10.1	エンティティ定義書/ データ項目定義書	△	○	△	△	
7.10.2	ドメイン定義書	△	○	△	△	
7.10.3	コード体系定義書/ コード内容定義書	△	○	△	△	
7.11.1	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。
7.12.1	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。
7.13.1	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。
7.13.2	全体移行計画書	○	○	○	△	
7.13.3	運用要件書	○	○	○	△	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。

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）

事例 9 要件定義のプロセス・ドキュメントの全体像策定 (セイコーエプソン株式会社)

【取り組みの背景】

システム構築において、品質確保、納期・予算遵守の観点から上流工程の取り組みが重要であり、その要である要件定義の品質向上が欠かせない。しかし、要件定義は一般的に確立されたプロセスや成果物があるわけではなく、実施にあたっては要件定義者の経験や裁量に委ねられることが多い。その結果、要件定義者の力量の違いによる品質のバラつきが生じやすいという問題があった。そのような属人性の排除と、要件定義の品質向上を目的に、要件定義プロセスとドキュメントの全体像を策定することとした。

【課題】

本事例では、解決すべき課題を以下のとおり設定した。

- ドキュメントの全体像、関連が明確でない

【解決策】

(1) 要件定義のプロセス・ドキュメントの全体像

要件定義のプロセス・ドキュメントの全体像を以下に示す(図 6.9)。業務部門とシステム部門との間で、要望・要求を獲得・分析し要件にまとめ、妥当性確認と合意形成を図るか、プロセスとドキュメントの両面から明確化した。

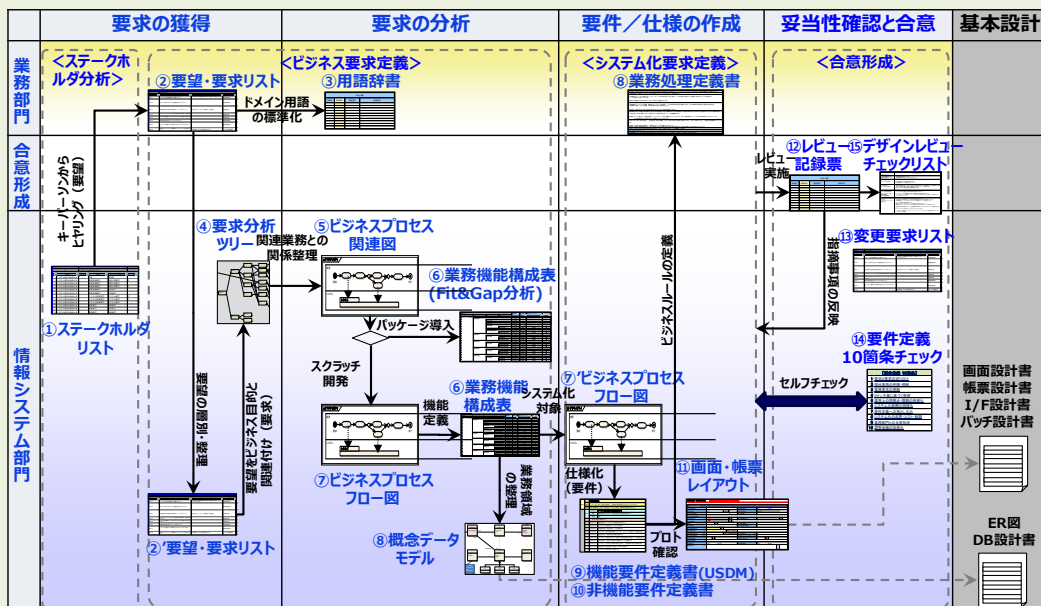


図 6.9 要件定義プロセス・ドキュメントの全体像

6.2 計画立案(RM.2)

プロセスは大きく「要求の獲得」「要求の分析」「要件／仕様の作成」「妥当性確認と合意」の4プロセスから成り立っており、「ステークホルダ分析」「ビジネス要求定義」「システム化要求定義」「合意形成」の手順を踏む。以下に具体的に説明する。

<ステークホルダ分析>

- ・ 「ステークホルダリスト」を作成し、ステークホルダの特定と分析を行う。

<ビジネス要求定義>

- ・ 「①ステークホルダリスト」に基づき、キーパーソンから初期要求をヒアリングする。そのヒアリング結果を「②要望・要求リスト」にまとめる。
- ・ 業務ドメイン固有の用語や、関係者で認識の齟齬が生ずる可能性のある用語については、「③用語辞書」に用語を定義する。
- ・ 「②要望・要求リスト」は、理由・目的、業務機能分類、優先度、実現可能性などの分析結果を付加し、「②' 要望・要求リスト」として更新する。
- ・ ビジネス目的と手段の整合性を確認するために、「②' 要望・要求リスト」を元に「④要求分析ツリー」を作成する。その結果、必要に応じて「②' 要望・要求リスト」の見直しを行う。
- ・ 今回の対象業務とその関連業務との関係を「⑤ビジネスプロセス関連図」に整理し、ビジネスプロセスと情報の繋がり整合性を確認する。
- ・ パッケージ導入の場合は、「⑥業務機能構成表」を元にFit&Gap分析（パッケージやクラウドに業務を合わせるFit to Standardという手法もある）を行う。
- ・ スクラッチ開発の場合は、対象業務を詳細化した「⑦ビジネスプロセスフロー図」と、「⑥業務機能構成表」「⑧概念データモデル」を整理する。

<システム化要求定義>

- ・ 「⑥業務機能構成表」を元に、システム化対象業務を明確化し、その業務についてはシステム化を想定し、「⑦' ビジネスプロセスフロー図」に更新する。
- ・ システム化対象業務については、ビジネスルールを「⑧業務処理定義書」に、システム機能を「⑨機能要件定義書」および「⑩非機能要件定義書」にそれぞれ定義する。
- ・ システム機能については「⑪画面・帳票レイアウト」によるプロトタイプの確認も合わせて行う。

<合意形成>

- ・ システム要件定義で作成したドキュメントを中心に、要件定義のレビューを実施

し、その結果を「⑫レビュー記録票」に記入する。指摘事項については必要に応じて各種ドキュメントの見直しを行う。

- ・ 変更要求については、「⑬変更要求リスト」による変更の実施可否判断を行う。
- ・ 要件定義の抜け・漏れ・誤りによる手戻り防止を目的とした「⑭要件定義 10 箇条」によるセルフチェックを実施後、システム構築プロセス全体の品質向上と関係者との合意形成を目的とした「⑮デザインレビューチェックリスト」によりプロジェクトとしての公式レビューを実施する。

(2) ドキュメントの視点と粒度

ドキュメント作成にあたっては、各ドキュメントの視点の整理と粒度の明確化が重要である。ドキュメントの視点と粒度を以下に示す（図 6.10）。

特に、粒度が荒いまま進めてしまった場合、後工程で詳細検討した際に「総論賛成、各論反対」となりかねないので注意が必要である。そのような手戻りを防止するため、「業務機能構成表」をベースに粒度のレベル感を次のように定義した。

対象業務全体をレベル 1、その対象業務の主要機能をレベル 2、主要機能の詳細をレベル 3、業務手順・システム機能をレベル 4 として定義し、機能要件定義書によって手順レベルまで要件定義と合意形成が図れるよう工夫した。

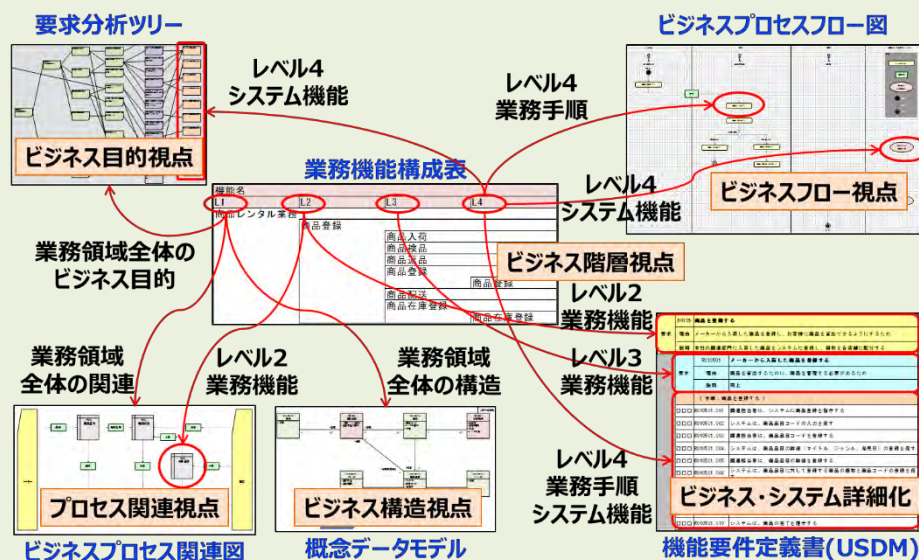


図 6.10 ドキュメントの視点と粒度

【効果】

本事例では、要件定義プロセスとドキュメントの全体像の策定とドキュメントの関連や視点、粒度の整理により、要件定義者の力量の違いによる品質のバラつきを最小限に抑えるとともに、要件定義の手戻りを防止し品質向上を図ることができた。

コラム 8 成果物に記載する範囲を決める

【悩み】

成果物は漏れなく正確に書くことが大前提である。しかし、成果物の記載ルールに「主要な〇〇を記載する」と書かれていることがよくある。「主要」だけでは漏れにつながるのではと心配になってしまう。

【解決案】

「主要」で良い成果物と、「すべて」が求められる成果物を明確にしておくことが重要である。

(1) 一覧系成果物の扱いを明示する

一覧系成果物は、漏れのチェックや管理に適している。そのため、一覧系成果物には「すべて」が求められる。何を一覧表で管理するのかを明確にし、他のどの成果物と関係付けられているのかを明示する。

一覧系成果物と、他の成果物の関係の例を以下に示す。

- 「問題点一覧（すべての問題点を記載）」
⇨ 「問題原因分析図（図で分析が必要な複雑な問題）」
- 「要求一覧（すべての要求を記載）」
⇨ 「要求構造図（すべての要求を記載）」など
- 「画面一覧/帳票一覧（処理単位で分割されたすべての画面を記載）」
⇨ 「システム化業務フローの画面（抽象化された画面名）」

(2) どのプロセスで作成するかによって記載範囲が変わる成果物を明示する

概念データモデルなどは、表 6.7 に示されているように、どのプロセスでも作成される可能性がある成果物である。このような成果物は、どのプロセスで、どの範囲で記載するのかを明示しよう。概念データモデルの記載範囲がプロセスごとにどう異なるかの例を表 6.10 に示す。記載サンプルを作って記載の詳細度を合意することがよく行われる。

表 6.10 各プロセスでの概念データモデルの記載範囲の例

	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
エンティティ	主要エンティティ	全エンティティ	全エンティティ
データ項目	主要データ項目	主要データ項目	全データ項目

(3) 後続の成果物で記載を補完する場合のドキュメント関係を明示する

「業務フロー」は業務改革レベルの変更を行う業務だけを対象として作成するが、「システム化業務フロー」はシステム化するすべての業務について記載するという、後続ドキュメントですべてを追求するという判断もある。

6.2.2 スケジュール作成(RM2.2)

要件定義の作業に関する工数と所要期間を見積もり、スケジュールを作成する。要件定義に必要な作業や成果物は、プロジェクトによって大きく異なるので、前の工程で作成、承認されたプロジェクトの企画書やシステム化計画書をベースにして、必要な作業（WBS）を洗い出して、漏れのないスケジュール計画を立てる必要がある。

なお、主要な WBS に関しては第 3 章 コラム 4 に参考例があるので、参考にされたい。

- (1) 所要期間の裏付けをできるだけ作り、スケジュール上のクリティカルパスには、その旨を明示しておく

【解決したい問題】

- 要件定義工程は、実行可能性の裏付けのあるスケジュールを作るのが難しい

要件定義工程の特徴として、主要な WBS の作業量と所要期間との関連性が弱いタスクが多いことである（すなわち、作業工数や投入人員を増やしても所要期間が短くなりにくいタスクが多い）。そのためどうしても、ここまでにこの WBS は終えておく必要がある、あるいは、ここまでに終わっているべきという観点で、スケジュールを引きがちである。しかし、次のようなことは、最低限取り組んでみる必要がある。

勘どころ① 作業工数、所要期間が見積れるものは、できるだけそれを活かす

作業量を見積もる際は、成果物単位ごとの見積基準（業務フロー 1 枚書くのに、どれくらいの時間を要するのかを過去の経験知、無ければトライアルで実測、または推定値）を設定して、作成する数量から作業工数を見積もり、それに安全係数を加味して必要工数を見積もる。

次に所要期間であるが、作業を担当するユーザ担当者の場合は、100%アサインでないケースも多く、月に 0.2 人月などの関与割合の方も居られるので、その点も考慮のうえ、投入可能工数とその方の作業予定期間も考慮のうえ、所要期間を見積もらなければならない。（作業効率も考慮して、所要期間にコンティンジェンシーも加味した方が良い）

勘どころ② 所要期間に大きな影響を与えるレビュー、部門間調整、承認に要する期間をあらかじめ織り込んでおく

6.2 計画立案(RM.2)

所要期間を見積もるにあたっては、ステークホルダ間での検討会議や、レビュー会議、承認会議などを要するものは、その開催頻度やタイミングも考慮する必要がある。なぜならば、作業遂行途中での関係者のレビューや、上位者の承認などが所要期間を決める大きな要素になるからである。一般的に、さまざまな検討を行う準備期間は別にして、レビュー・承認に要する期間（レビュー結果の修正対応も含む）は、大方の目安として、担当者が成果物作成に要する期間と同等かそれ以上の期間を要すると考えたほうが良い。

また、作業ボリュームは大きくはないが、複雑な課題の検討や、各種調整、根回しが特に必要なタスクについては、見積りの期間と実際とが大きく乖離しやすいので、スケジュール計画にその特性を織り込んでおく。ただし、長めに取れば取るほど、さらに長くかかる場合も多いため（すなわち、日程遅延が顕著にならないと結論がまとまらない傾向があるため）、あまり実態に合わせた計画が良いとは言えない点にも、留意することが必要。

上記のような、要素を考慮してスケジュールの実行可能性をできるだけ裏付けておくことが望ましい。ただし、実行可能性を高めれば高める程、要件定義の期間が長くなるため、予定期間に収めるための調整が欠かせない。そのために次のことを行う。

- 一部のタスクは「ここまでで終えなければならない」などの条件や事情から完了予定日を決めることがある。その場合は注意書きなどで、そのことが関係者に分かるようにしておく。また、完了できなかった場合に備えて、スケジュール立案段階で思いつく対策も、同様に分かるようにしておく。
- タスク間の関連性（先行タスクが終了しないと開始できない関係性）を考慮しつつも、並行作業を当初から相当程度、考慮する。（実際、相互の関連性もあるので、並行作業にしたほうが、結果として早く終わる関係のタスクも少なくない）
- リソースの制約を加味しながらも、作業着手を可能な限り前倒しするようにする。（着手することで、作業上の課題が明確になる利点がある）

勘どころ③ 手戻りや大きなスケジュール遅れに対する遅延対策を、あらかじめ計画しておく

要件定義の成果物は相互の関連性が強く、前に作ったものの修正が入ることや詳細を詰めてみて初めて分かる先行作成物の誤りや不十分性の発覚があるために、遡っての見直しが発生することがよくある。したがって、いったん完了となっているものへの修正が入ることを、あらかじめ見込んでおく必要がある。しかし、個別成果物単位に上記を織り込むことは難しいので、要件定義の後半にまとめて2週間ほどの見直し、修正期間を設けておくことが望ましい。

一方遅延対策として、大きな遅延が発生した場合の優先度の考えをまとめておき、思い切った成果物の削減や粒度の調整などができるようにしておく。例えば、システム要件関連の成果物の詳細度（完了基準）に関しては、一部を緩めて概要レベルにするとか、一部

の成果物を次工程にまわすことなども遅延対策として、あらかじめ関係者の了解を取っておくと良い。大きな遅延が起きた場合でも、関係者の合意が取りやすく、迅速に対応できようになるためである。

勘どころ④ 必ず実施すべきこと、省略できることに分けて、すべきことは期間を変更してでも漏らさずやるよう計画する

要件定義工程のどの成果物をどこまで記載するか決めたら、「3.2 要件定義プロセス」と、コラム 4 を参考にスケジュールを組み立てていく。その中で特にプロジェクトマネジメントに関する要素は次のとおりである。

1. 前プロセスで決めたプロジェクトの目的・目標・スコープ・問題課題をあらためて確認する
2. 各成果物一覧と、その作成担当・レビュー担当を割り当てて、承認者を決める
3. 進捗管理の方法を決める
4. 体制（レポートライン・責任者・プロジェクトオーナー）とチームビルディングの方法を決める
5. コミュニケーション方法（進捗報告、要件定義会議、レビューなど）、プロジェクトメンバの教育方法を決める
6. 次工程で必要な見積り方法を決める
7. 品質の担保方法を決める
8. スコープ・要件変更時の段取りを決める
9. 要件定義プロセスで使用できる予算とその管理方法を決める
10. リスクとその対策を決める
11. 契約方法を決める

これら 11 要素は、1 から順に組み立てていくわけではなく、プロジェクト制約の強い要素から検討を進めると良い。例えば、体制が前プロセスから既に決まっていて、かつ変更しようがない場合、その体制前提でコミュニケーション方法や品質担保方法を決めていけば良い。その際、その前提ではプロジェクトが前に進みづらいとリスク認識したら、上司やプロジェクトオーナーにエスカレーションして、前提の変更や改善を求めていくべきである。

また、これら 11 要素は、短期間に同時並行で関係者とすり合わせていく必要がある。よって要件定義プロセスに初めて関わる人材が要素を漏らさず同時にすり合わせることは大変困難であり、要件定義プロセスの経験者が実施すべきと考えている。ここでいう経験者とは、リーダーの指導下で、要件定義プロセスを経験した者のことである。経験者が

6.2 計画立案(RM.2)

過去の実績や標準的な計画書をベースに、11 要素の観点からプロジェクトの特殊性を加味して、プロジェクトで予測される事態を想定し、WBS 作成して実際に使えるスケジュールを組み立てていく必要がある。

昨今のシステム開発の短納期化にともない、要件定義の工期を短くせざるを得ず、予定工期とおりに要件定義を終えられてないケースをしばしば耳にする。その場合は、要件定義プロセス立案時、

- 対象業務の知見を持ったシステム部門メンバをアサインし、要件獲得時間を短縮する
- 合意形成の段取りをあらかじめ決めておいてロス時間を極小化する
- リスク認識と対策の検討を実施し、リスク顕在化したときのロス時間を極小化する

ということを計画したい。

最後に、これら 11 要素のうち、1. 4. 5. 6. 7. 10. 11. のスケジュール作成に必要な知見は、本節「6.2 計画立案」の他の項で述べるので参照されたい。それ以外の要素からスケジュールを組み立てるためのチェックポイントは次のとおりである。

- 成果物は過去プロジェクトの知見が入っている標準ドキュメントのフォーマットを使用して作成すること
- 機能要件だけでなく非機能要件も抽出する計画をすること。(忘れがちな非機能要件抽出プロセスは、その組織内の標準プロセスに組み込み、皆が参照するようにしておくこと)
- 非機能要件のリスク回避のためのタスクをスケジュールに組み込むこと。例えば画面応答時間にリスクがあれば、早期に検証すること。
- 業務部門やエンドユーザのタスクをスケジュールに組み込み、十分な時間を確保すること

6.2.3 費用見積り(RM.2.3)

一般に開発ボリュームの見積りは要件定義が終わってから実施するものと思われがちであるが、システム開発においてはそもそも投入できる総予算が最初から決定していることがほとんどである。要件定義においても、すべてが完了してから予算超過に気がつくという進め方をすることはない。要件定義の途中で何回か、そこまでに積み上がった要求に基づいてシステムを構築するとどの程度の規模になるのかを定量的にチェックできるように作業プロセスを計画すること、計画にしたがって途中でチェックをすること、そして、その結果から総量の規制を行うことにより、開発できる規模をはるかに超えた無駄な要件定義を行う愚を避ける必要がある。

また、途中段階での費用の見積りは、システム構築だけでなく、実施中の要件定義プロセスを対象としても実施する必要がある。要件定義プロセスの最大の課題は「膨らむ要求のコントロール」であるが、要求が膨らめば、その対応の検討にかかる作業量は、絞り込みにかかる作業量も加えれば、それ以上に増加する。システム開発の規模と同様に、要件定義プロセスの作業量も計画の範囲内に収めるべくコントロールする必要がある。

(1) 要求の規模は件数ではなく実現にかかるコストで表現する

【解決したい問題】

- 要件定義しているシステムの規模が、許容範囲内であるかが分からない
 - ・ 今どこまで要求が積み上がっているのか不明
 - ・ 規模を示す数値が画面などの件数しかなく実態が分からない
 - ・ 要件定義完了まで規模が不明ではスコープ調整の手間が大きい

要件定義は、一般に利用部門からの要求を積み上げることから始める。そのため、要求は基本的に膨らみ、実現のためのコストもそれに合わせて増加する。ところが、要求検討や検討後のシステム化にかけられる人的リソースや資金は有限であるため、その中から優先度の高い要求を選択してシステムを構築する必要がある。

したがって、要件定義においては、途中の段階で、要求やそれを実現するのにかかるリソース（人員や資金）がどこまで積み上がっているのかを把握し、要求の総量を投入可能なリソースの範囲に抑えることが必要になる。しかし、要件定義の途中、特にビジネス要求の検討段階では、積み上がった要求の総量を把握することが難しい。それらは、要件定義の現場では、以下のような問題として浮かび上がってくることが多い。

- どこまで業務部門の要求を受け入れたら良いのか限度が分からない

6.2 計画立案(RM.2)

- 業務部門からシステム部門への要求の提示が止まらない
- 業務部門の要求をすべて検討すると要件定義が終わらない
- 業務部門がコスト、期間、効果を考えないで言いたいことを言う
- 業務部門からの要求の実現にどれだけのコストがかかるのか見当がつかない
- 個々の要求を実現するための開発規模をシステム部門で定量的に評価できない

上記のような問題が要件定義の途中で発生しないようにするためには、以下のような対策を取ることが必要になる。

勘どころ① 要件定義の途中でもシステム構築費用が概算見積りできるように計画しておく

システム開発の規模は、本来は、要件定義の完了後、システム開発に着手して基本設計を完了した時点で確定するものであり、システムの開発費用を要件定義の途中で予測することは難しい。また、システムに適用するソフトウェアや開発・運用インフラの選定も、要件定義を進めていく途中で当初の予定から変更になることがある。しかしながら、要件定義の途中でも、以下のような理由により、システム構築・運用にかかる費用を概算することが求められることが実際には多い。

- 開発対象のシステム全体や個々の機能に対する投資額を算出し、投資対効果を評価したい
- ビジネス上の効果から決定した予算総額を開発費用が超えてないか、途中の段階でこまめに確認したい
- 要件定義の途中で要件量が膨らんだため、要求ごとに開発費用を概算し、各要求元と実現する要求の範囲・内容を交渉したい
- プロジェクトの投資額の見込みを経営者に報告したい

ここで言う要件量とは、要件定義工程が終わるまでに獲得する要件の数量のことで、機能数や画面数のように、客観的な情報に基づき、定量的に評価できる尺度のことである。要件定義の途中で、現時点で要求（とその実現コスト）がどれだけ積み上がっていて、規制をかけないと行けない段階に来ているのかどうかを確認することは、要件定義を成功裏に完了させるための必須条件であり、必ずこのような作業を計画段階から何回か、あるいは節目のタイミングで実施することをスケジュールに組み込んでおく。

要件定義の途中段階でもシステム開発の規模（費用）を概算できるようにするためには、以下のような方法を用いて、要件定義のそれぞれの結果をそれぞれ定量的に評価しながら、要件定義を進める必要がある。その際には、その時点で定量的に把握できる物量と、

6.2 計画立案(RM.2)

これまでの類似案件での経験から設定された生産性の積算により費用を算出する。規模を把握する際には、例えば、概算見積り時点で計測できる以下のような数値を用いる。

- ユースケース数
- 要求を実現するために必要と予測した画面、帳票、バッチ機能などの数量
- 要求の複雑度、実現の難易度などにより重み付けした上記の数量
- システム改造案件の場合、システムの機能のうち修正対象になるものの規模

また、概算見積りを実施する際には、以下の点に留意する必要がある。

- 要件定義工程の、どのタイミングで、どういう意図で概算見積りを実施して、どのような意思決定に用いるかを明確にし、その目的に応じた概算見積りを実施する
- あくまで途中段階での推定であり、まだ変動するという前提に立ち、前提条件を明確にする
- 設定した前提が、以降の要件定義工程での検討の結果により覆る場合があるので、概算見積り結果を報告する際には、見積りの前提を明記する
- 概算に用いる生産性指標は、プロジェクト完了時点で評価し、常に最新化しておく
- システム開発費用だけでなく、開発時や運用時に使用するソフトウェア、インフラ機器などの購入、開発・運用環境の構築費用、ならびにシステム運用開始後にこのシステムの運用を想定する期間の運用コストを加算する

勘どころ② 要求の規模を物件の数や難易度から容易に概算できるようにしておく

個々の要求の内容具体化の段階で、その実現に必要なとなる画面や帳票、バッチ他の機能のおよその数、開発難易度を概算で見積もる。種類別に見積もった規模にそれぞれの生産性を掛けてコストを算出し、それらを積算する方法によりコストを算出する。これを必要時にすぐに実施できるように、画面や帳票の数や複雑度を、必要性が明確になった都度、一覧表などに蓄積しておくことをルール化することにより、途中段階での概算見積りに備える。

勘どころ③ 有識者を投入して規模概算の精度を上げる

要求の具体化の段階で開発規模を正しく見積もるためには、要件定義対象の業務に対する知見だけでなく、それを実現するシステムの開発に関する知見を必要とする。より正確に見積りを実施するためには、このような知見を持った要員を要件定義に参加させることが望ましい。要件定義の体制構築時には、そのような知見をもった開発担当者・ベンダの協力を得られるようにしておく。

6.2 計画立案(RM.2)

勘どころ④ 概算結果に対して過度の精度を追求しない

要件定義途中での規模概算を行う際には、社内やベンダの見積り結果に対して過度の精度を追求しないことが必要である。概算ではあっても「見積り」を依頼すると、特にベンダには公式な見積書との印象を与え、拘束力を持ちそうに思わせる可能性がある。そうになると、概算見積りであっても結果が出るまでに時間を必要とするし、余分なリスクが積まれることにも繋がりがねない。概算見積りはあくまで「目安」でありスピードを優先すること、その代わり精度が粗くなるのはプロジェクトマネジメント側のリスクと割り切る、後で「目安」に対して言質を問わない、などの前提をつけておくことが、場合によっては、逆に見積り精度を向上させる効果をもたらす。

勘どころ⑤ 実績を蓄積して次回に備える

要件定義において各要求を実現するために要したコストの実績は、次回の要件定義に向けた貴重な経験になりうる。各プロジェクトで実施した要件定義と開発規模の実績や、規模当たりの開発生産性の実績を蓄積する、そのような活動を継続することにより、それ以降の要件定義において規模の概算見積り精度を向上させる効果が期待できる。

(2) 要件定義にかかるコストをそれまでの実績に基づき見直す

【解決したい問題】

■ 要件定義にどれ位期間と労力がかかるのか定量的に予測できない

システム開発のコスト（工数）は、これまでの長期にわたる実績の蓄積から、生産量×各生産物の単位生産性の総和により、実績と大きな乖離が発生しない精度で求めることができるようになった。これに対して要件定義は、そもそもが思考の繰り返しによって作業が進められる性質をもったプロセスであり、労力の投入量当たりの成果物の作成量が一定の範囲に収まらないことが多く、単位当たりの生産性を数値化することが困難であることから、投入予定人数×期間によって概算見積りされることが多い。

そのため、要件定義開始時の計画が、要求の膨張により遅延し始めても、その後どれ位の人員・期間を投入すれば終わるのが定量的に予測できず、投入人員の増加と期間の延長を何回も繰り返す、という事態に陥りやすい。

勘どころ① 過去実績から定量的見積りを行う

システム開発費用と同様に、要件定義にかかるコスト（工数）を生産量×生産性により見積もる。そのためには、過去の実績を蓄積しておくことが必要である。

生産量 システム数、機能数、画面数、帳票数など数値化できる指標を用いる

生産性 過去の実績から要件定義において作成される成果物のそれぞれの作成に要する標準的なコストを設定する。

補正 プロジェクトの特性や投入人員のスキルレベルなどにより生産性に差が出やすい成果物は、標準的なコストに上限値、下限値を設定し、適用するプロジェクトの特性に応じた値を選択して補正する

過去の実績がない成果物の作成を計画する場合には、なるべく類似した成果物の生産性を代用するか、サンプル作成により生産性を測定するなどして、設定値の精度を上げる努力をすることが望ましい。

勘どころ② 途中で実績に基づいて補正する

要件定義を実施する際には、生産性を設定した成果物の単位で実績を収集し、計画値との乖離が発生していないかどうかを点検するプロセスを作業計画に組み込む。点検の結果、計画値との乖離が大きく、作業計画（人員投入計画、スケジュールなど）の見直しが必要と認められる場合には、直ちに計画の修正を行う。なお、予定よりも実績の方が生産性が高い場合であっても、乖離がある場合には同様に見直しを行うことが必要である。それ

6.2 計画立案(RM.2)

により、要求が計画よりも膨らんだ場合への許容量を実態に合わせて補正することができる。

また、要件定義のコストを算出する際には、作業効率化のためのハード・ソフト費用のほかに、POC(Proof of Concept)用のハード・ソフト費用など、人件費以外の付帯費用の計上が必要になる。これについても随時見直しを図り、コストとして加算することを忘れないようにする。

これらの施策により、プロジェクトにおける要件定義の各成果物の実態の生産性を把握することができ、要求が膨らんだ場合に、それ以降に必要な労力をより高い精度で予測することができるようになる。

次ページで、要件定義プロセスのコストを、要件量の数値化と、見積り実施の前提条件の明確化に基づく生産性基準値の設定により、システム開発と同様の高い精度で見積もっている事例を紹介する。要件定義においても、実績を蓄積して、それを分析することにより要件量の数値化と生産性設定が可能になり、それを用いてこのプロジェクトにおける要件定義に対してシステム開発と同様の見積りを行うことが実際に行われている事例として参考にして欲しい。

事例 10 要求の定量化と見積り前提条件の見える化による
要件定義工程の見積り
(株式会社 ジャステック)

【取り組みの背景】

一般的に要件定義工程の見積りは、要件未確定を理由に、「要件定義工程の成果物を作成できる要員が何人ほど必要で、いつまでに作るから何人月です」といった感覚的な見積りが見受けられる。その結果、期限切れによる要件積み残しの発生および要件積み残しのまま無理に後工程を執行することによる要件の取り違い、さらには当該工程での成果物の誤り、漏れが発覚し手戻りコストの増加などにつながる原因の一つになっている。

【課題】

弊社での販売管理業務システムの再構築開発において、こうした問題を解決すべく、以下の3つの課題を設定した。

- 当該工程に定量的プロジェクト管理の導入を前提とした要求の定量化に関する課題
- 当該工程の見積りを行うための前提条件課題
- 当該工程見積りと整合させた開発全体工程の見積り課題

【解決策】

当該再構築開発において要求の定量化を行うために、ソフトウェア開発での独自の生産管理方式 (ACTUM) に基づき、要件定義工程の成果物量 (生産物量) と生産性を代用指標とした定量的プロジェクト管理を導入し、課題の対策と改善を通して問題の解決を図った事例を紹介する。

(1) 要件定義工程の見積りを行うための前提条件

要件定義工程のインプットとしてシステム企画書 (目的・施策、ROI、優先度、予算等) および組織標準 (開発プロセス、生産物と記述水準、各種規約、体制、役割分担、責任等) などが存在し、ステークホルダ間で合意していることが前提である。

当該事例では要件定義工程の見積りを行うにあたり、特に、前提条件として弊社が留意「確認と合意」している事例を以下に紹介する。

① 業務要件とシステム要件との整合

一般的に要求 (ビジネスの目的・施策) は階層構造をしているが、最上位の要求

6.2 計画立案(RM.2)

から見積りを可能とする方式は弊社でも課題となっている。しかし、ビジネス目的を反映した業務要件の As Is(現状)→To Be(こうしたい)から改善課題を析出し、析出した当該課題対策の中に IT 施策としてのシステム要件が存在する。当該事例の要件定義工程での見積り方式を適用するにあたり、少なくとも業務要件とシステム要件の統合がドキュメント化され合意が得られていることが前提である。図 6.11 にドキュメント事例を紹介する。

業務要件					対策(システム要件)			備考	
大分類	現状(As Is)	こうしたい(To Be)	番号	小分類(改善課題)	課題区分(※1)	番号	内容		影響度(※2)
1	決裁業務の効率化(時間短縮) 例による運用であるが、本社に「G/T」は半ば普及していない(緊急の顧客対応にのみ用いられる場合がある)	本社以外の法人・セパラルPC等での決裁業務を可能にしたい。		決裁業務の時間短縮(外出先からも決裁業務可能な仕組みの構築) 電子決裁および電子捺印保管を前提とした業務フローおよび決裁業務要件記述書の見直しを行う。	B	1-1-1	電子決裁(ワークフロー)システムを構築する。	新規	業務要件を踏まえ、パソコンソフトの導入を検討。注「御提案」で電子決裁(ワークフロー)のワークフローの構築。決裁システムで電子捺印を構築し、構築後運用を円滑にする。 業務要件に基づいて、以下の検討も必要。 ・決裁以外の印品表での捺印監視の可否(セキュリティ)。 ・捺印監視の要件。 ・顧客と構築後で対応する業務の受渡し方法。 ・顧客と構築後で対応した業務の受渡方法。
						1-1-2	現在の組織による運用から電子捺印での運用に変更する。	改良	

業務要件					対策(システム要件)			備考	
大分類	現状(As Is)	こうしたい(To Be)	番号	小分類(改善課題)	課題区分(※1)	番号	内容		影響度(※2)
NO	フロントデスク業務は決裁以外の決裁業務を管理(申請書や管理番号の連携)が後の運用となる。バックアップは必要(申請)を要する。	決裁業務を電子化することで、管理番号の連携効率向上、スピード向上、エラー防止を期待したい。		決裁業務を管理の電子化による本人の業務の廃止を行う。	C	1-12-1	全ての決裁業務の管理番号システム化する。	新規	管理番号の付与を前提とする。 -申請番号 -管理番号
						1-12-2	管理番号のシステム化によって、決裁業務を自動化する。	改良	

図 6.11 業務要件とシステム要件とを統合させたドキュメント事例

② 要件定義工程の生産物および記述水準

弊社の見積りモデルは生産物量と生産性が基準指標となるため、要件定義工程(各開発工程含め)の生産物にどのようなドキュメントを適用し、どのような記述水準にするかを合意することが前提である。以下に生産物ごとの第1、第2アクティビティおよび生産物の作成要領を定義している標準化事例の一部(業務フロー)を図 6.12 に紹介する。

③ 非機能要求の要求レベル

弊社の非機能要件定義は見積り方式の‘環境変数’の一部に相当し、環境変数には‘品質特性’と‘環境特性’があり、両特性の因子ごとに要求レベルを5段階評価している。

環境変数は‘(2)②’および‘当該事例末項’に記載した‘添付別紙1’に示す。

開発フェーズ名称	要件定義	10	生産物ID	030	生産物名称	業務フロー
030	業務フロー	8	第1アクティビティ	03001	対象業務の決定	業務を分解し、フロー展開すべき対象業務(粒度)を定める。
			第2アクティビティ		アウトプット	生産物を作成するための要領
10	関係者の特定と権限の整理		03002		①業務フローに展開する業務および順序を特定する。	
10	業務スケジュールの整理		03003		①フロー展開する業務(粒度)毎に、業務実施者、関係者(組織)、利用者	
10	INPUT, OUTPUTの特定		03004		①考慮すべき業務の執行タイミングをすべて洗い出す。	
10	インターフェースの確認		03005		①抽出した業務毎に、実施時のINPUT, OUTPUTを特定する。	
10	業務フロー・特殊事項の確認		03006		①業務間のインターフェースおよび業務内での関連組織(内部/外部、	
10	業務の効率化検討		03007		①業務上、起こり得る異例処理の有無を確認する。 ②異例処理が存在する場合は、以下事項を明らかにすること。	
10	業務フローの作成		03008		①組織別の業務についても効率化(コスト含)を目的に、統合を検討す	
10	抽出結果(業務、内部組織、権限と承認、外部組織)、確認結果(処理タイムズ、入出力、インターフェース、異例処				①抽出結果、確認結果をもとに、業務フローを作成する。 ②関連組織およびシステム(内部/外部)と対象業務は、同一ページ内で表現できるようにすること。 ③組織間のインターフェースはその媒体、伝達手段を明記すること。 ④フロー毎に処理タイムズ(日次、月次等)を明記すること。処理タイムズが複数存在する場合は、フロー図を分けて作成すること。 ⑤異例処理に該当するフローは「異例処理」であることを明確にすること。 ⑥「特殊事項」は「業務フロー」に「特殊事項」を記載すること。 ⑦システムを利用する業務が、利用しない業務と明確にすること。 ⑧フローに表現できない作業等は特記、説明として記載すること。	
10	検証および承認				①関係部署による業務フローレビューを実施し、承認する。 ②特に利用ユーザー部門の承認を得ること。 ③複数システムの連携で実現される業務に関しては、各システム運用担当者が参加したワークスローレビューを開催すると良い。	

図 6.12 標準化事例の一部(業務フロー)

見積りの前提としては、環境変数のインプットになる非機能要求確認書(非機能要件設定)を作成している。その事例を図 6.13 に事例を示す。

(3) 非機能要件設定

(3)-1. 非機能要件設定(環境変数「共通版」新規開発、改造型開発、再構築開発」)

非機能要件に関しては、当社の環境変数ごとに要件を設定する。
当数要件は非機能要件仕様及び見積りの環境変数(生産物、生産物)の入力とする。
(非機能要件仕様は2.(3)、見積りも「4.(3)費用見積り」参照)

環境変数		要件(内容)	
機能性	品質特性	合目的性(含む環境配慮機能)	① 利用者の立場(中継業務本部の一部が利用することになり、影響は軽微) ② ステークホルダーの増加(現行と同じコンテナエンジン)に対する機能要求(現行と同じ)
	正確性	① システム標準「検索」「レビュー時間」「テスト項目」に準拠する。	
	接続性	① 他システム(「会計システム」「損益管理システム」とのインターフェースは、現行と同じ) ② ワークフロー(前掲「スケジュール」)とのインターフェース、並びにロード及びフォーマット変換仕様も、非機能要件仕様にて定義する。	
	セキュリティ	① システム標準「情報セキュリティ/開発・保守基準」に準拠する。	
	成熟性	① 現行システムと同水準とする。	
	信頼性	障害許容性	① システム標準「開発・保守に関する信頼性基準」に準拠する。
		回復性	① システム標準「開発・保守に関する信頼性基準」に準拠する。
		理解性	① 新たに、営業本部でマニュアル(標準業務)を作成するが、開発局員と別冊で作成する。
	使用性	習得性	① 同上
		操作性	① 現行システムと同水準とする。
効率性	実行効率性	① 実行開発(「ワークフロー」)向け開発環境「業務フロー」先は標準機能等)にて既済確認済みのパフォーマンスが優れる案のフェーズに最適化する。	
	資源効率性	① 最大同時利用人数を配分し、CPJ仕様情報システム部の協力を得て決定する。	
保守性	解析性	① システム標準「開発・保守に関する信頼性基準」に準拠する。	
	変更作業性	① 新たな保守用システムを作成しない(継続開発生産物として、変更作業を行う)。	
開発環境特性	環境特性	要件(内容)	
	業務特性	① 現行業務を担当する営業員(庶務員含む)が参加する。旧システムを特化した営業員が一部参加する。 ② ①、②を踏まえ、開発・設計(一部ワークフロー設計含む)、テスト実施を本部で行う。	
	HW特性	① 当社で使用実績のあるハードウェア(OS含む)を利用する。	
	SW特性	安定度/信頼性/使用実績	① 市販パッケージは、当社で使用実績がほとんどないものを採用することになる。開発・設計(一部ワークフロー設計含む)、テスト実施を本部で行う。(後述の「開発手法」参照)。
		顧客窓口特性	① 開発(製造工程)は情報システム部にお届けするが、上位工程作成者(営業本部)窓口とする。
	コミュニケーション特性	工期の厳しき	① 標準工期は2ヶ月(21日)以内。開発・設計(一部ワークフロー設計含む)は、開発・設計(一部ワークフロー設計含む)と並行して実施する。
		コミュニケーション/基盤	① 開発(製造工程)は情報システム部にお届けするが、上位工程作成者(営業本部)が全面的に支援する。
		レビュー体制	① 上位工程(企画、要件定義、基本設計(一部ワークフロー設計含む))及びテストは、システムと業務部が連携して実施する。 ② 開発工程(製造工程)は情報システム部にお届けするが、システムと業務部が連携して実施する。
	業務/他システム資料規約/標準化	① 現行業務管理システムの設計書については、一部不足していることと、開発・設計(一部ワークフロー設計含む)と並行して実施する。 ② 一方、設計においては、既済資料、設計書を採用し、活用する。	
	開発手法/開発環境	① 2024年7月以降により、以下の生産性向上を図り、SW特性での生産性を確保する。 ② 後工程の手戻り抑制(先)問題点を顕在化する。」「習熟度を高める。	
テスト要領書水準	① システム標準「テスト」規格項目、「認定水準」に準拠して作成する。		

図 6.13 非機能要件の設定

6.2 計画立案(RM.2)

④ 現行（現行業務と既存システム）の可視化

要件定義工程での生産物量と生産性の見積りを行う上で、現行を調査し把握することは重要である。弊社の環境変数「再構築特性」には、現行の可視化「業務の可視化度合、システムの可視化度合「データ含む」、トレーサビリティ」の3つの副特性が存在し、因子ごとに具備状態を5段階評価している。見積りの前提としては、環境変数のインプットになる上記の3つの副特性以外の副特性を含めて確認書を作成している。その環境変数は「(2)③」および「当該事例末項」に記載した「添付別紙1(2)」に示す。

なお、5段階評価結果でのレベルに応じた生産物の復元量は、開発時の生産性との兼ね合いで合意を得る必要がある。この場合の復元作業は要件定義工程での付帯作業として見積もっている。

a) 開發生産性との相互関係を勘案した現行の可視化

現行の可視化度合いによる現行生産物の復元量は、現行の可視化に関する生産性環境変数（現行ドキュメントの品質「解析性」、既存システムの錬度「業務、システム」、ツール具備「調査」）の評価レベルを勘案する。また、他の生産物量環境変数と生産性環境変数との相関関係（例：棚卸しと移植性など）を含め、最適なアルゴリズムが存在する。ここでは紙面の都合上、アルゴリズムの紹介は省略する。

■ 現行の可視化に関する生産物量環境変数*1

主特性	副特性	基準生産物量からの変動率(%)			
		要件定義	設計	製作	テスト
現行(業務&システム)の可視化	業務の可視化度合	0 ~ 35	-	-	-
	システムの可視化度合 (データ含む)	0 ~ 30	0 ~ 38	0 ~ 15	0 ~ 10
	トレーサビリティ	0 ~ 5	0 ~ 10	0 ~ 15	-

*1: 付帯作業としての復元生産物量を対象

■ 現行の可視化に関する生産性環境変数*2

主特性	副特性	基準生産性からの変動率(%)			
		要件定義	設計	製作	テスト
現行ドキュメント等の品質	解析性	0 ~ 80	0 ~ 80	0 ~ 30	-
現行の錬度	既存業務	0 ~ 150	0 ~ 60	-	0 ~ 70
	既存システム (旧新アーキテクチャ差異含む)	0 ~ 125	0 ~ 170	0 ~ 130	0 ~ 100
ツールの具備度合	既存システムの調査ツール	0 ~ 15	0 ~ 20	0 ~ 20	-

*2: 付帯作業(復元生産性)と当該開発工程作業(開發生産性)を対象

図 6.14 開發生産性との相互関係を勘案した現行の可視化

b) 見積りを行う上での必要な現行（要件定義工程の生産物）の可視化

現行のすべての生産物が可視化されていることが望ましいが、実態はすべての生産物が品質の保証含め可視化されていることはまれである。そこで、弊社では見積りを行う上で、以下に記載した3つの現行の生産物を、現行の可視化に係る生産性環境変数との兼ね合いを勘案し、可視化（復元）することの合意を得るようにしている。

- ・ 業務フロー（現行業務の流れ‘組織、手段、手順’の把握）
- ・ 概念レベルのER図（現行の‘データ、業務ルールなど’の管理過程の把握）
- ・ 個別業務処理定義書（現行各業務機能と現行システム機能との対応付け）

なお、‘(1)①’「新業務要件と新システム要件の整合表」が存在していること、また、実態上において、画面/帳票レイアウト、データ項目（棚卸しを留意）は存在するので利用可能である。よって、可視化対象から外している。

(2) 生産物量と生産性に基づく見積りモデル

① 要件定義工程における見積りの基本アルゴリズム（新規開発見積り方式）

要件定義工程(i=1)での、あるアクティビティ（生産物）’K’のコスト C_k は生産物量 V_k 、生産性を P_k 、基準生産物量を V_k^B および基準生産性を P_k^B で表現すると、次式で求まる。

$$C_k = V_k \times P_k = V_k^B (1+a_i) \times P_k^B (1+b_i)$$

$$a_i: (\sum \alpha_{ij}) \quad b_i: (\sum \beta_{ij})$$

添字 (i, j, k)
i: 工程
j: 環境変数因子
k: 工程生産物

(注) KC : K . Character H : 時間 C_k : 円 or 工数 V_k : KC P_k : 円/KC or H/KC KC は Page 変換有

要件定義工程(i=1)のコストは当該工程で作成する各生産物に対応するコストの総和である。よって、コスト C_1 は次式で求まる。

$$C_1 = \sum C_k$$

表 6.11 に要件定義工程の生産物量と生産性を見積り事例を紹介する。

6.2 計画立案(RM.2)

② 非機能要求を定量化した環境変数

- a_i 、 b_i は V^B_i および P^B_i に対して品質要求の多寡や開発環境の違いによる変動を吸収する‘環境変数’と呼ぶパラメータである。(a_i は生産物量環境変数、 b_i は生産性環境変数)
- a_i 、 b_i は‘品質特性’と‘環境特性’から影響される独立した変動要素 (α_{ij} 、 β_{ij}) から構成している。 $a_i(\sum \alpha_{ij})$ と $b_i(\sum \beta_{ij})$ は‘添付別紙 1(1)’の共通編(新規開発、改造型開発、再構築開発)を参照のこと。なお、改造型開発と再構築開発は共通編に加え‘改造型特性’と‘再構築特性(新規開発と改造型開発の一部を共有)’がある。‘添付別紙 1(2)’の固有編を参照のこと。

③ その他の見積り方式について

a) 改造型開発見積り方式

CP(Cost of Positional Analysis)、**CN**(Cost of Net)、**CR**(Cost of Regression)

$$C_i = CP_i + CN_i + CR_i$$

$$CP_i = VP_i \times PP^B_i (1 + b_i + bP_i)$$

$$CN_i = VN^B_i (1 + a_i) \times P^B_i (1 + b_i + bN_i)$$

$$CR_i = VR^B_i (1 + a_i) \times P^B_i (1 + b_i + bR_i)$$

改造型固有の環境変数「 $bP_i: (\sum \beta P_{ij})$ $bN_i: (\sum \beta N_{ij})$ $bR_i: (\sum \beta R_{ij})$ 」

b) 再構築開発見積り方式

CD(Cost of Differential Analysis)、**CRe**(Cost of Re-build)

$$C_i = CD_i + CRe_i$$

$$CD_i = VD_i \times PD^B_i (1 + b_i + bD_i)$$

$$CRe_i = VRe^B_i (1 + a_i + aRe_i) \times P^B_i (1 + b_i + bRe_i)$$

再構築固有の環境変数「 $bD_i: (\sum \beta D_{ij})$ $aRe_i: (\sum \alpha Re_{ij})$ $bRe_i: (\sum \beta Re_{ij})$ 」

表 6.11 要件定義工程の生産物量と生産性の見積りの事例

工程	生産物	分担	生産物量 (KC)							境変数 (率) bi
			妥当量	基準生産物量 V	生産物量環境変数		分担率 RB	許容要求 変更量 (V*(1+a ₁))*ω	許容要求 変更率 ω	
					(率)	a ₁				
要件定義	ビジネス関連図	営業本部	16,459 KC	14,250 KC	0.713	5.0%	95.0%	1,496 KC	10.0%	5.0%
		システム部	0,867 KC	0,751 KC	0.038	5.0%	5.0%	0,079 KC	10.0%	5.0%
		小計	17,326 KC	15,000 KC	0.750	5.0%		1,575 KC	10.0%	5.0%
	業務一覧	営業本部	28,678 KC	23,750 KC	1,188	5.0%	95.0%	3,741 KC	15.0%	8.0%
		システム部	1,509 KC	1,250 KC	0,063	5.0%	5.0%	0,197 KC	15.0%	8.0%
		小計	30,188 KC	25,000 KC	1,250	5.0%		3,938 KC	15.0%	8.0%
	業務フロー	営業本部	45,885 KC	38,000 KC	1,900	5.0%	95.0%	5,985 KC	15.0%	8.0%
		システム部	2,415 KC	2,000 KC	0,100	5.0%	5.0%	0,315 KC	15.0%	8.0%
		小計	48,300 KC	40,000 KC	2,000	5.0%		6,300 KC	15.0%	8.0%
	概念レベルのER図	営業本部	54,079 KC	42,750 KC	4,275	10.0%	95.0%	7,054 KC	15.0%	8.0%
		システム部	2,846 KC	2,250 KC	0,225	10.0%	5.0%	0,371 KC	15.0%	8.0%
		小計	56,925 KC	45,000 KC	4,500	10.0%		7,425 KC	15.0%	8.0%
	個別業務処理定義書	営業本部	53,130 KC	44,000 KC	2,200	5.0%	80.0%	6,930 KC	15.0%	5.0%
		システム部	13,283 KC	11,000 KC	0,550	5.0%	20.0%	1,733 KC	15.0%	5.0%
		小計	66,413 KC	55,000 KC	2,750	5.0%		8,663 KC	15.0%	0.0%
	画面/帳票レイアウト (画面変遷を含む)	営業本部	66,853 KC	55,250 KC	5,525	10.0%	65.0%	6,078 KC	10.0%	0.0%
		システム部	35,998 KC	29,750 KC	2,975	10.0%	35.0%	3,273 KC	10.0%	0.0%
		小計	102,850 KC	85,000 KC	8,500	10.0%		9,350 KC	10.0%	0.0%
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	7.2%
	要件定義工程計	営業本部	370,916 KC	301,000 KC	24,080	8.0%	86.0%	45,836 KC	14.1%	5.8%
		システム部	56,991 KC	49,000 KC	3,430	7.0%	14.0%	4,561 KC	8.7%	6.5%
		合計	427,908 KC	350,000 KC	27,510	7.9%		50,398 KC	13.4%	

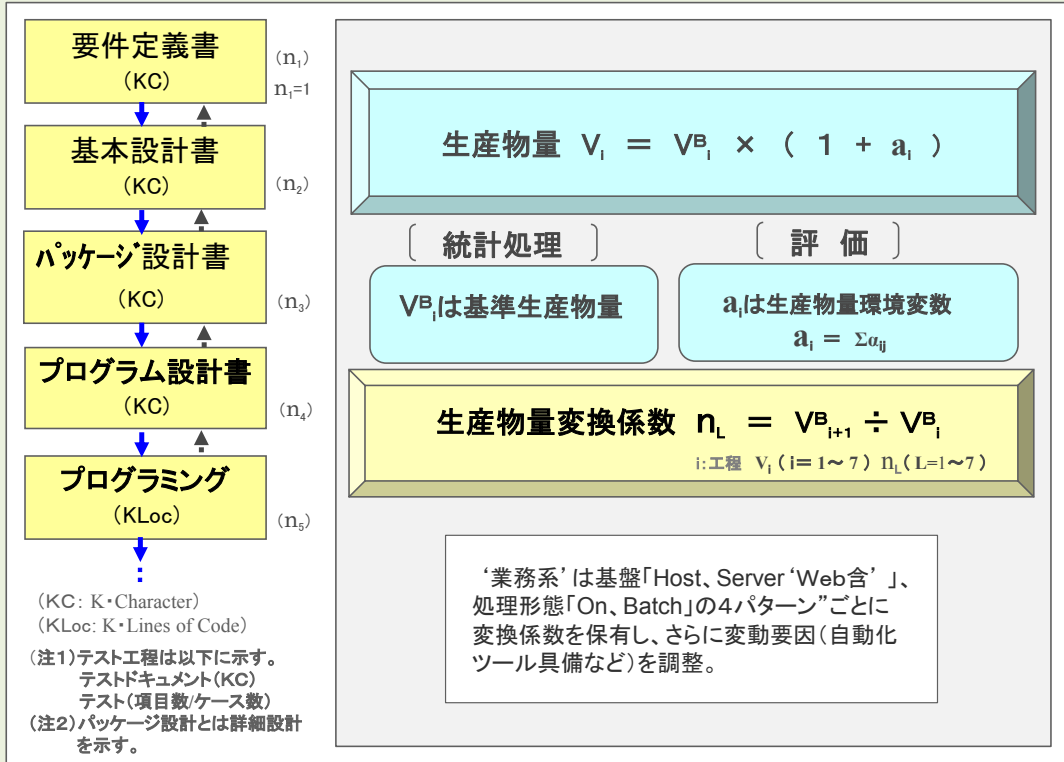
注) 妥当量、妥当生産性および許容要求変更量‘率’は後述の事例 17 を参照

なお、見積りをお客様と合意するために、お客様の標準見積り指標 (FP、機能数等) に準じる場合がある。その場合は弊社見積り基準指標値 (生産物、生産性) から変換する方式を用意している。

(3) 生産物量変換係数による開発全体の概算見積り

要件定義工程の生産物量から各工程間の生産物量変換係数を使用して、開発全体の概算見積りを算出している。なお、弊社では要件定義工程が完了していない見積りを概算見積りと呼んでいる。以下に各工程間の生産物量変換係数の概要を図 6.15 に示す。さらに、生産物量変換係数を使用して、各工程の概算見積りを算出した事例を表 6.12 に示す。

6.2 計画立案(RM.2)



注) 変換係数に使用する要件定義書はシステム要件定義書を対象

図 6.15 生産物量変換係数の概要説明

表 6.12 生産物量変換係数を使用した開発全体の概算見積り事例

工程生産物	生産物量変換係数 (n _L)	(量×量単価[円/生産物量]) 当該事例は要件定義書'1KC'とした場合						
		要件定義	基本設計	パッケージ設計	プログラム設計	プログラミング	統合テスト	システムテスト
要件定義書 (Kc)	n ₁ = 1	¥?0,000	¥?0,000	¥?0,000	¥?0,000	¥?0,000	¥?0,000	¥?0,000
基本設計書(Kc)	n ₂ = 1.44	¥0	¥?04,244	¥?04,244	¥?04,244	¥?04,244	¥?04,244	¥?04,244
パッケージ仕様書(Kc)	n ₃ = 7.60	¥0	¥0	¥?87,969	¥?87,969	¥?87,969	¥?87,969	¥?87,969
プログラム設計書(Kc)	n ₄ = 9.53	¥0	¥0	¥0	¥?8,131	¥?8,131	¥?8,131	¥?8,131
プログラミング(KLoc)	n ₅ = 0.93	¥0	¥0	¥0	¥0	¥?36,215	¥?36,215	¥?36,215
統合テスト仕様書(テスト項目)	n ₆ = 65.55	¥0	¥0	¥0	¥0	¥0	¥?72,184	¥?72,184
システムテスト仕様書(テスト項目)	n ₇ = 18.69	¥0	¥0	¥0	¥0	¥0	¥0	¥?3,411
合計		¥?0,000	¥?84,244	¥?72,213	¥?10,344	¥?46,559	¥?18,743	¥?002,154

注) 基盤 ‘Server’ で処理形態 ‘Online’ のパターンを適用 (見積り値の一部をマスク)

【効果】

当該再構築開発では生産物量と生産性に基づく要件定義工程の見積り、生産物量変換係数を使った全体工程の見積りなどを課題を定めて実践した。その結果、弊社標準に比べ基本設計以降の見積り精度向上や手戻りコストが抑えられたこと、さらに見積り前提条件を整備し明確化したことで、以下に示す要件定義工程のいくつかの課題に対して改善することができた。

- 要求獲得—現状把握 “現行業務要件とシステム要件との整合”
- 要求分析—要求の体系化 “As-Is To-Be による要求の不明確さの是正”
- 要求の仕様化—仕様化 “要件定義成果物「記載項目&記述水準」の遵守・欠陥防止”
- 非機能要求—環境変数 “環境変数因子ごとの調査分析による漏れ防止や要求水準把握”
- 現行生産物可視化—対鍊度アルゴリズム等 “復元の効率化、規程の有効利用”
- 計画立案—要件管理計画 “要求量の可視化、見積り精度向上と合意形成 “

さらに、当該事例の見積りモデルと事例 17 で紹介している出来高予実管理方式は、基準指標値（生産物量、生産性）が前提になっている。繰り返しになるが、見積りは見積りのためだけでなく、見積りを‘予’とした予実管理を可能にするためでもある。定量的プロジェクト管理の真髄である各指標値を精錬させ、変動幅の小さいベースラインを確立させることが重要である。

ただし、弊社のモデルと方式は他に比べ面倒であるといった側面もある。しかし、日本人は製品への緻密な取り組みや既成概念に捉われない品質へのこだわりが、世界に伍した製品を生み出してきた。グローバル化が進展する状況下、ソフトウェア業界もグローバル化されていくのは自然であるが、一方で、その国民性を活かした取り組み姿勢を忘れてはならない。

そのような思いとともに、当該事例を通して経営者、ユーザ部門、システム部門（情報関連会社含む）およびベンダが、見積り前提条件を整備し明確化するなどを含め、要件定義工程での見積りについての新たな‘気づき’が得られればといった期待効果がある。

6.2 計画立案(RM.2)

添付別紙 1 環境変数の概要説明表

(1) 共通編

① 生産物量環境変数（新規開発、改造型開発、再構築開発）

対象見積りモデル：新規開発(C)、改造型開発(CP、CN、CR)、再構築開発(CD、CRe)

特性タイプ	主特性 (対象コストモデル)	副特性	評価の観点(概略内容)	基準生産物量からの変動率(%)						
				要件定義	設計	製作	テスト			
品質特性	使用性 (C、CN、CRe)	理解性	対象見積りモデル：新規開発(C)、改造型開発(CP、CN、CR)、再構築開発(CD、CRe)	機能性	合目的性	0 ~ 50	0 ~ 50	0 ~ 50	0 ~ 50	
					正確性			0 ~ 20	0 ~ 50	
	保守性 (C、CN、CRe)	操作性	接続性	他システムとの接続によるコード変換、フォーマット変換数	0 ~ 5	0 ~ 20	0 ~ 20	0 ~ 20		
				セキュリティ	0 ~ 20	0 ~ 20	0 ~ 20	0 ~ 20		
	信頼性 (C、CN、CRe)	解析性	標準適合性	環境基準「法規制・条約など」がシステムに適用する要求で、地球環境配慮設計・実装に必要なとする付加機能数	0 ~ 5	0 ~ 10	0 ~ 10	0 ~ 10		
				成熟性	故障低減に必要な実現機能数	0 ~ 5	0 ~ 10	0 ~ 10	0 ~ 10	
					障害許容性	異常検知に必要な機能数	0 ~ 5	0 ~ 10	0 ~ 10	0 ~ 10
						回復性	再開処理に必要な実現機能数	0 ~ 5	0 ~ 10	0 ~ 10

② 生産性環境変数（新規開発、改造型開発、再構築開発）

対象見積りモデル：新規開発(C)、改造型開発(CP、CN、CR)、再構築開発(CD、CRe)

特性タイプ	主特性 (対象コストモデル)	副特性	評価の観点(概略内容)	基準生産性からの変動率(%)						
				要件定義	設計	製作	テスト			
環境特性	業務特性 (C、CP、CN、CR、CD、CRe)	安定使用	対象見積りモデル：新規開発(C)、改造型開発(CP、CN、CR)、再構築開発(CD、CRe)	機能性	合目的性(要求仕様の網羅性)	0 ~ 100	0 ~ 30	0 ~ 10		
					正確性	0 ~ 5	0 ~ 5	0 ~ 3	0 ~ 5	
	ハードウェア特性 (C、CP、CN、CR、CD、CRe)	安定使用	接続性	基準単位(100KS)に対する社内/社外システムとのインターフェース先の数	0 ~ 20	0 ~ 10	0 ~ 10			
				整合性	0 ~ 15	0 ~ 15	0 ~ 8	0 ~ 8		
	ソフトウェア特性 (C、CP、CN、CR、CD、CRe)	顧客	効率性	実行効率性	0 ~ 5	0 ~ 10	0 ~ 5	0 ~ 10		
				資源効率性	0 ~ 5	0 ~ 10	0 ~ 5	0 ~ 10		
	コミュニケーション特性 (C、CP、CN、CR、CD、CRe)	工期	品質特性	保守性	解析性			0 ~ 5		
					安定性	0 ~ 8	0 ~ 10	0 ~ 8		
					移植性	環境運用性	0 ~ 28	0 ~ 28	0 ~ 12	0 ~ 25
						移植作業性				

(注) 移行、教育、保守、運用作業は生産性環境変数の適用対象から除いている

(2) 固有編

① 生産物量環境変数（再構築開発）

対象見積りモデル：再構築(再構築開発「CD、CRe」)

特性タイプ	主特性 (対象コストモデル：再構築特性は主特性に)	副特性	評価の観点(概略内容)	基準生産物量からの変動率(%)			
				要件定義	設計	製作	テスト
再 特 性	現行(業務&システム)の可視化*1 (CD, CRe, CP)	業務の可視化度合*1	・現行業務の目的・施策および業務内容を含めた全体の現行業務生産物(業務フロー等)の具備状態を評価	0 ~ 35	-	-	-
		システムの可視化度合*1 (データ含む)	・現行システムのハードウェア、ソフトウェア、主要DB(ファイル)、ネットワークの各構成および関連を明確にした、現行システム全体俯瞰資料の具備状態を評価、 ・現行システムおよびアプリケーションの構造を明確にした、シーケンス図、モジュール構成図、クラス図などの具備状態を評価、 ・データの目的およびデータ間の関連を明確にした、業務データ関連図、概念データモデルなどの具備状態を評価	0 ~ 30	0 ~ 38	0 ~ 15	0 ~ 10
		トレーサビリティ*1	・業務要件とシステム要件、システム要件と基本設計(外部設計生産物)、外部設計生産物とパッケージ設計生産物、パッケージ設計生産物とプログラムの紐付けに関する整備度合いを評価	0 ~ 5	0 ~ 10	0 ~ 5	-
	移 行 (CRe)	移行ツール	・移行ツール機能('プログラムand/orデータ'、検証用コンペア等)の具備要求度合いを評価	0 ~ 5	0 ~ 10	0 ~ 20	0 ~ 15
		棚卸し*1	・移行元リソースの品質向上を鑑みた既存システムの棚卸し(プログラムのスリム化、データクレンジングなど)作業の要求度合いを評価	0 ~ 2	0 ~ 5	0 ~ 20	0 ~ 10
		移行パターン	・旧新(新旧)とリソース「データ、プログラム、環境」の組合わせは、6パターン(論理上'8')存在する。当該再構築がどのパターンに該当するかを判断し、移行にあたり「データ、環境」がプログラムに影響(修正対応)する量を評価。	0 ~ 10	0 ~ 25	0 ~ 25	0 ~ 25
		追い付き 差分取り込み	・再構築開発と並行保守開発している現行システムの並行保守開発期間(凍結タイミング)を鑑みた、差分取り込みの多寡を評価	0 ~ 5	0 ~ 10	0 ~ 20	0 ~ 20

*1: '現行の可視化'と'棚卸し'は、付帯作業の生産物量である。よって、新規開発'C1'と改造型開発'CP'も対象になる。

② 生産性環境変数（改造型開発、再構築開発）

対象見積りモデル：改造開発型(母体調査分析「CP」、正味規模改造「CN」、テスト巻き込み「CR」)

特性タイプ	主特性 (対象コストモデル)	副特性	評価の観点(概略内容)	基準生産性からの変動率(%)						
				要件定義	設計	製作	テスト			
改 造 型 特 性	対象見積りモデル：再構築(差異調査分析「CD」、再構築開発「CRe」)			基準生産性からの変動率(%)						
	(CP)	特性タイプ	主特性	副特性	評価の観点(概略内容)	要件定義	設計	製作	テスト	
	母体録	(対象コストモデル)	現行ドキュメント等の品質*2	移植性*2	・移行元全リソース(データ等)の不備や不整合度合いなどを評価	0 ~ 5	0 ~ 20	0 ~ 80	0 ~ 50	
	既存母品質			解析性*2	・既存業務&システム全体のドキュメント、ソースコード等、成果物の解析容易性を評価(保守状態「可視化度合も含む」)	0 ~ 80	0 ~ 80	0 ~ 30	-	
	(CP, CN)			環境適応性*2	・多様なハード、ソフト、運用環境への旧新適用度を評価	0 ~ 15	0 ~ 20	0 ~ 20	0 ~ 30	
	既存テ	(CD, CRe, C1)	再 特 性	現行の練度 (既存システム運用含)	現行業務 ²	・既存業務(システム運用含む)有識者の関与度合「工程毎の有識者関与割合」標準との対比」を評価	0 ~ 150	0 ~ 60	-	0 ~ 70
	(CN, CR)				既存システム (旧新アーキテクチャの差異を含む)	・調査分析経験(回数)と調査分析対象規模(既存システム全体との割合)を評価 ・既存システム開発者は影響度'O'で評価(但し、既存システム全体との割合を考慮) ・アーキテクチャ使用and/or開発経験(回数)を評価「旧新差異の取り込み時の生産性への影響」	0 ~ 100	0 ~ 170	0 ~ 130	0 ~ 100
(注)改造型開発	ツールの具備度合				既存システムの調査ツール	・調査ツール機能(絞込み、モニタリング、リバース等)の数で評価	0 ~ 15	0 ~ 20	0 ~ 20	-
(CD, CR)	移行ツール				・移行ツール機能('プログラム言語、データタイプ、コンペアツールなど)の数で評価	-	0 ~ 5	0 ~ 25	0 ~ 40	

2: '現行業務に関係するドキュメントの品質'と'現行業務練度'は、新規開発'C1'も対象になる。

6.2.4 体制・チームビルディング(RM.2.4)

要件定義は、一般に複数のステークホルダがチームを組んで実施する。システムを利用する側、システムを構築する側から、対象業務システムの構築に必要な知見を有する人員をバランスよく集めてプロジェクトに投入することが成功の秘訣であるが、最初からそうはうまく行かないのが世の常である。ここでは、チームを構築する際の問題点、集めたメンバーでどうやってチームワークを発揮するかについて述べる。

(1) プロジェクト目標の達成に向けて、体制を構築しチームビルディングを行う

【解決したい問題】

- プロジェクトに必要な体制が構築できなかつたり、チームワークが発揮できない
 - ・ 業務知識を持つキーマンの時間が十分に確保できない
 - ・ プロジェクトメンバーや関係者が責任を取らない
 - ・ 要件定義に必要な分野の IT 技術をもった人材が確保できない

システム開発は、多くのステークホルダとプロジェクト目標の達成に向けて体制を組み、チームとして活動する。しかし、体制構築やチームビルディングができていないと上記のような問題が発生する。

その場合、プロジェクト目標が達成できない、トラブルが多発する、納期が間に合わないといったことが起こり、メンバーが閉塞感を感じたり、思うような成果を出すことができず、モチベーションの低下にもつながる。

このようなことを防止するためには、プロジェクト目標の達成に向けたチームワークが欠かせない。しかし、チームとは、ただ人が集まれば自然にできるものではなく、リーダーが意識をして強靱なチームを作り上げていく必要がある。

勘どころ① プロジェクトに必要な体制を構築する

「体制構築でプロジェクトの成否の大部分は決まる」とよく言われる。プロジェクトの立ち上げにあたって、業務部門とシステム部門でチームを編成する。チーム編成では、チームごとの役割分担が重要である。各チームが実施する作業に漏れが出ないようにプロジェクトのタスクとチームの役割を対応付けるようにする。そのために、体制図の作成と合わせて、各自の役割分担（プロジェクトオーナー、プロジェクト責任者、プロジェクトマネージャ、担当リーダーなど）や責任範囲、権限を明確にした役割分担表（表 6.13 役割

分担表)を作成するようにする。

その際には、システム部門だけではなく、業務部門の体制も明確にし、各役割のカウンターパートが分かるようにする。また、業務部門の体制、特にキーマンの影響は大きいですが、優秀なメンバは既に別のプロジェクトにアサインされていることも多く、キーマンの時間が十分に確保できないため、要件定義が失敗に終わることも多い。こちらについては、事例 11「キーマンの時間が十分に確保できない」を参照されたい。

また、本来、システムはビジネスの競争力を高め、差別化するために活用するものである。しかし、ビジネスとシステムの橋渡しができないばかりに、経営に資するシステム活用とならないケースも多い。システムを経営に活かすために、「ビジネスアナリスト」を含めた体制構築が重要である。

表 6.13 役割分担表の例

役割	役割の説明
プロジェクトオーナー	・会社代表として、担当プロジェクトにおける最高意思決定者
プロジェクト責任者	・プロジェクト計画書の承認 ・ステアリングコミッティの開催 等
プロジェクトマネージャ	・プロジェクト計画の実行、コントロール ・進捗会議の開催 等
ビジネスアナリスト	・業務部門とシステム部門の橋渡しによるビジネス強化の主導 担当領域を成功裏に実行する責任を果たす
業務部門責任者	・業務要求や業務要件の取りまとめ ・業務チーム定例会議の開催 等
業務部門メンバ	・ビジネス要求の定義 ・業務側テストの計画・実施 等
システム部門責任者	・システム化要求やシステム要件の取りまとめ ・要件定義マネジメントの計画、監視・コントロール ・システムチーム定例会議の開催 等
システムアーキテクト	・システム方式の設計／開発の責任者
システム部門メンバ	・システム化要求の定義 ・システム側テストの計画・実施 等

勘どころ② チームの発達段階を理解し、メンバに働きかける

チームビルディングを知る上では、心理学者のタックマンが提唱した「タックマンモデル」が参考になる。タックマンモデルによると、チームが形成されていく過程を4段階で

6.2 計画立案(RM.2)

表し、最後に解散する時期のチーム状態を加えた以下の5段階で示している。

- 成立期(Forming)
プロジェクトが立ち上がり、チームメンバが収集された初期段階である。メンバが集まったばかりのこの段階では、メンバ同士がお互いのことを知らず、プロジェクト目標も分からず模索している状態である。この状態は、様子見といった具合でリーダーからの指示を待っている状態である。
この時期のリーダーは、メンバ全員にプロジェクト目標を共有し、明確な指示を出して仕事を進める。
- 動乱期(Storming)
目的・目標に対する意見の違いや、人間関係、作業の進め方についての対立が発生する状態である。
この時期のリーダーは、チーム力を生み出すための準備段階として重要な「お互いを知ること」が目的となる。お互いの仕事や人間性(特技や強み、弱み)を理解し、プロジェクト目標を達成するために誰の強みを活かし、弱みをどう補い合うかを共有する段階である。
- 安定期(Norming)
目的や進め方、メンバの役割が共有されている状態である。チーム内の自分の立場が安定することによって、他のメンバやチーム全体のことに気を配れるようになり、チームワークが生まれる。
この時期のリーダーは、相互に助け合えるような関係性を構築できるようメンバ間のコミュニケーションを促進し、チームとしての成功体験を積み重ねるよう促すことである。
- 遂行期(Performing)
結束力が生まれメンバが互いに協力できるようになる段階である。メンバ間でお互いの立場やスキル、人間性を理解し、強固なチームワークが生まれる。また、自立的な活動も活発に行われる。
この時期のリーダーは、メンバに権限を委譲し細かな指示は避けるようにする。リーダーだけがリーダーシップを発揮するのではなく、メンバ全員がリーダーシップを発揮する。各メンバが周りを巻き込み、協力し合うことで、チームとしての成果を最大化する。
- 解散期(Adjourning)

プロジェクトが終了し、メンバが移動し関係が終了する段階である。
この時期のリーダーは、メンバに感謝しながら最後の締めくくりを行う。

このように、チームはメンバが収集された初期段階から始まり、成長によってメンバ同士の関係性が強まり、さらに成長することでチームとして最大限の成果を獲得できるようになる。

勘どころ③ チームとしての一体感を醸成する

システム開発を行う上で、個人やチームごとに与えられたタスクをただこなすというやり方ではプロジェクトを成功させることは難しい。規模の大きな開発では、チームごとに機能や役割を分担し、それらを合わせて 1 つのシステムを作り上げる方法を取ることが多いが、つなげたときにチーム間の情報共有の不足に起因する不具合も多い。

そのような際には、そのチームだけに責任を負わせるのではなく、プロジェクトメンバ全員で問題解決を目指す姿勢が大事である。そのためには、チームとしての一体感を醸成することが求められる。

リーダーはメンバを従わせるのではなく、ファシリテータとなることである。その際には、適切な「質問」と「傾聴」を繰り返すうちに、メンバは自分で気づき、行動するようになる。とはいえ、メンバの知識や経験不足により進むべき道が導き出せない場合もある。そのような場合には、メンバのニーズを捉えた上で「助言」し、メンバの決意に対しては「応援」する。また、成果が芳しくない場合には、チームの力で「支援」をするようにする。

このような一体感を醸成したチームビルディングにより、メンバー一人ひとりにプロジェクトへのオーナーシップを持たせながら信頼関係を構築することで、チームとして高い成果を創出することが、プロジェクト目標を達成することにつながる。

事例 11 キーマンの時間が十分に確保できない (キーマンが役割、責務を果たすプロジェクト環境が整えられていない)

【取り組みの背景】

0社では、前回のプロジェクトで、ユーザ部門からアサインされたユーザ代表メンバ（以下、「キーマン」と称する）の時間が確保できず、ユーザ課題の解決遅れや、ユーザ側のレビュー遅れが頻繁に発生したため、要件定義工程が大きく遅延したことがあった。

今回の「業務改革に基づく基幹システムの再構築プロジェクト（ERP 導入プロジェクト）」は、稼働時期を厳守することを求められており、何らかの対策が必要となっていた。（なお、本プロジェクトの総工数は当初 860 人月の予定であった。）

【課題】

前回のプロジェクトの反省会では、次のような課題があげられていた。

- ①キーマンは現場の業務を優先しがちで、プロジェクトの作業を先送りするため、常に期限が守られていなかった。
- ②ユーザ課題には、部門間をまたがる調整および合意を要するものが多く、キーマンの負担が非常に大きかったことも、期限が守られなかった大きな要因であった。
- ③また、ユーザ部門の上位マネジメントにも、キーマンの負担を下げ、課題解決期限や、レビュー完了期限を守るようにするマネジメントがほとんど見られなかった。

など、改善の余地があることがわかった。

【対応策】

前回の経験を踏まえ、ユーザ側プロジェクトマネージャ（以下、「PM」と称する）はプロジェクトオーナー（経営企画担当常務取締役、以下、「オーナー」と称する）とも協議の結果、次の対応策を取ることの了承を得、オーナーにも積極的に動いていただき、次の対応を取った。

- (1) キーマンを専任化して、兼務者も含めて人事発令を行った
 - ・今回は PM からオーナーをお願いをして、オーナーからユーザ部門の上位マネジメントに、キーマンの専任化と、兼務メンバのアサインの依頼を行っていただいた。

- ・また、プロジェクトは会社内の委員会組織と同じ扱いとして、プロジェクトメンバを委嘱する人事発令を行い、専任化されたキーマンについては専任であることを明示してもらった。(兼務者については必要関与工数をアサインメント依頼書に明記して、オーナーから現場マネジメントに依頼し、合意を得た。)
 - ・所属組織における人事評価においても、プロジェクトにおける評価を反映してもらう運用を取り決めて、キーマンや兼務者の責任感とモチベーションの確保を図った。
 - ・一方、専任者については、プロジェクト専任に伴う不利益を絶対に生じさせない事、プロジェクト終了後は現職復帰を前提に、本人の希望を最優先とすることを、所属組織担当役員からのコミットメントとして、ご本人に提示していただいた。
 - ・また、専任者の業務については、部門内の人事異動によるカバーと、他部門からの人事異動によるカバーで、プロジェクト期間中は乗り切ることが、全社で決まって実行された。
- (2) 分科会を置いて、部門間調整を迅速化した
- ・部門間をまたがる業務仕様、課題はテーマ別の分科会を設置し、分科会長には、部門間調整力を有する組織内で認められた人材をアサインしてもらった。
 - ・分科会長には、意思決定の権限と責任を与えた。
 - ・キーマンも分科会のメンバになり、プロジェクトで生じているユーザ課題の正確な連携を期した。
 - ・分科会とプロジェクト組織との関係は、下図の「プロジェクト体制図」のとおりである。

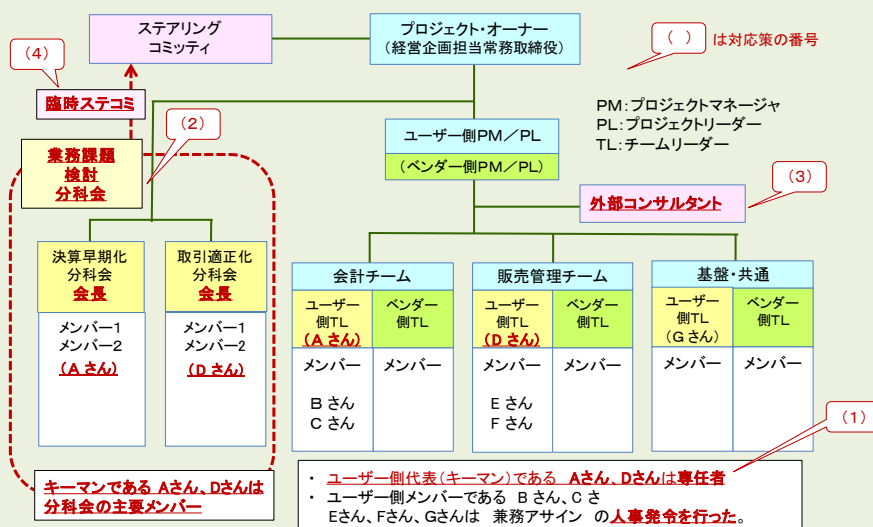


図 6.16 プロジェクト体制図

6.2 計画立案(RM.2)

- (3) 外部コンサルタントを活用した
 - ・要件定義の進め方や課題解決のコンセンサスを得やすくするために、要件定義の経験の深い外部コンサルタント1名を確保して、PMの支援を行ってもらった。
 - ・特に、外部のコンサルタント中立性を活かして、プロジェクトチーム内におけるプロジェクト推進上の課題解決、進捗遅れへの対応に関して、メンバー間の合意をファシリテートしてもらった。(今回は、プロジェクトの中身ではなく、進め方や進捗管理面で支援をいただき、PMやキーマンの負担を下げてもらった)
- (4) 分科会でもまとまらない課題は随時、臨時ステアリングコミッティを開催し、解決促進を図った
 - ・定例の月一回のステアリングコミッティ(以下「ステコミ」と称する)では、スケジュールが間に合わない場合は、臨時のステコミを開催することを、プロジェクト発足時にオーナーからステコミボードメンバーにはお願いして、承諾をいただいていた。
 - ・臨時のステコミは、電話会議も駆使して、それでも出席がどうしても難しいボードメンバーには事前説明、ないしはオーナーへの一任をいただいて、迅速に意思決定を行った。

【効果】

- (1) キーマンの専任化と兼任者を含む人事発令
 - ・専任化と人事発令により、キーマンの時間確保が担保され、他の兼務メンバーも関与工数目途が組織的に合意されていたので、現業業務との調整に上位マネジメントの協力が得られ、キーマンの時間不足による遅延は解消した。
- (2) 分科会設置で部門間調整を迅速化
 - ・分科会を早期に設置したこと、分科会長に決定権限が与えられていたこと、それでも意思決定が難しいケースに対して、臨時ステコミ開催を事前合意できていたので、ユーザ課題の解決が促進され、途中での解決期限超過はあったものの、要件定義工程内で重要課題のすべての解決を図ることができた。
- (3) 外部コンサルタントの活用
 - ・外部コンサルタントの中立性を活かして、プロジェクト推進上の課題解決をファシリテートしてもらい、プロジェクトチーム内のコンフリクトをうまく解決し、結果として要件定義期限内に、必要な作業を終えることができた。

(4) 随時ステアリングコミッティの随時開催

- ・ 重要課題は随時ステコミ決裁を仰ぎ、効率的に課題解決を図ることができた。

これらの施策はキーマン自身への支援策にもなっており、キーマンの責任感とやる気を引き出した面も大きく、課題であったキーマンの関与時間不足による遅延を防止できた。

6.2.5 品質計画(RM.2.5)

システム開発の品質は何で評価するか。一般的には、レビューやテスト、本稼働後に発生した障害の指摘件数や指摘内容から評価することになる。システム開発は、要件定義の結果をもとに要求どおりの仕様を具備しているか、仕様どおりの動作をするかをチェックすることによりバグを抽出することができる。これに対して要件定義には、レビューでチェックできる成果物の見た目の品質（誤字脱字、文書の正確さなど）以外に、ユーザの隠れた要求を適切に抽出できているかどうかという答えのない品質が求められる。ここでは、主に前者の要件定義の見た目の品質の確保について言及する。前者の品質は保てないが、後者の品質は保てるということは、実態としてあり得ないことであり、見た目の品質を確保することから始めることが重要である。後者については、第4章を参照願いたい。

(1) 成果物の品質は作成方法の指導と定期的な状況確認で保つ

【解決したい問題】

■ 成果物を作成するときに、記載内容に個人差が出て品質にバラつきが出る

要件定義工程において、各種ドキュメントを作成するにあたり、まずは、そのドキュメントの作成目的、成果物の作成指針などを決める。さらに作成指針に即したサンプルなどを準備して、ドキュメントの品質レベルをできるだけ均一にしようとするが、実際には、作業を開始すると成果物の品質は個人差が発生し、バラつきが出てしまう。

記載粒度などもどこまで詳細に記載すべきなのかは個人差が出やすい。また、成果物の作成指針があったとしても、作成指針と違う書き方をしてしまう担当者がある場合もある。まとめると以下のような課題が発生することがある。

- 要件定義成果物の品質が管理できない。
- レビューのたびに大量の指摘事項が発生する。
- 担当者によって成果物の品質が揃わない。

上記課題に対して、ドキュメントの品質のバラつきが出ないようにするための対策を記載する。

勘どころ① 成果物の作成指針を作成し、要件定義ドキュメント作成の関係者と合意する

6.2 計画立案(RM.2)

やはりドキュメントを作成するうえで、成果物の作成指針は必要である。ポイントとしては、成果物の作成指針を作成したうえで、要件定義ドキュメント作成の関係者と合意することである。これは、執筆側だけではなく、レビューする側を含めた関係者との合意である。そうすることで、成果物の標準化につながり、後述の記載内容や記載粒度について、統一化させる一助になる。

記載内容についてもなるべく具体的にシンプルにまとめることが大事である。例えば、記載事項をシンプルに書いて欲しい時に、「1文 100文字程度までに抑える」よりは、「主語と述語の関係がしっかり分かるように書く」とした方が、成果物の作成指針としてチェックすべき観点がはっきりする。一見 100文字という具体的な指示のように見えるが、100文字以下だからといって、分かりやすい文章になっていない場合もある。文章構造の書き方を成果物の作成指針に書くべきであり、その点からも後者の方が、成果物の作成指針としては、適切であると思われる。ドキュメントのサンプルを作成することによって、担当者に作成イメージを持ってもらうとより効果的である。

さらには、記載粒度についても成果物の作成指針に記載する必要がある。要件定義書において、どのレベルまで記載すべきなのかを成果物の作成指針によって決めておく。注意すべき点は、ユーザに理解できること、次工程のインプット資料として適切な記載レベルとなっていること、など意識して決めると良い。ユーザに理解してもらうためには、言葉だけではなく、図などを利用して視覚的に理解できるような配慮があると、認識のずれなどは軽減できる。

文章のあいまいさを排除して要件を正しく伝達することも大事なことであるが、そのことについては5章に記載している。合わせて確認して欲しい。

勘どころ② 成果物の作成指針の書き方について説明会を行う

成果物の作成指針を作成し、関係者と合意したら、ドキュメント作成者を集めて、書き方についての説明会を行ったほうが良い。成果物の作成指針を作成者がそれぞれ読み込んで作成することになると、やはりそこは記載レベルや内容に個人差が出てしまう可能性がある。どのような意図で、成果物の作成指針を作成したのかなど、経緯を含めて説明会を行うことで、個人差が発生しないようにしておいた方が良い。また、説明会を行うことで、成果物の作成指針の改善点が見つかる場合も想定される。ドキュメント作成を本格的に開始する前に、成果物の作成指針を共有し、ドキュメント作成の品質につなげていく活動が、遠回りのように見えるかもしれないが、期限を守ることにもつながっていくことを認識すべきである。

勘どころ③ 成果物の作成指針に沿ってドキュメントが作成されているかを確認するチェックリストを作成する

6.2 計画立案(RM.2)

成果物の作成指針だけでは、ドキュメント作成内容に個人差が出やすいので、各自チェックができるようにチェックシートを作成し、作成した成果物が作成指針に沿って作成できているかチェックできるようにしておく。各レビューの前に作成者が内容を自己チェックしておくことによって、品質が良くなる。

勘どころ④ 執筆初期の段階で重点的なレビューを行う

ドキュメントが作成され始めた初期の段階で、成果物の作成指針に沿ってドキュメントが記載されているかを確認するために、レビューを実施する。成果物の作成指針に合わせて記載ができているかの確認を、ドキュメント作成の初期の段階で実施することで、手戻りリスクを軽減できる。また、このタイミングでレビューを実施することで、成果物の作成指針自体のブラッシュアップにもつなげられ、より高品質なドキュメントを作成することが期待できる。

勘どころ⑤ 定期的にドキュメントのレビューを行い、ドキュメントの品質水準を確認する

①から④までで担当者が記載方法について理解できるようになるが、要件定義作業を進めていくなかで、定期的にレビューを行う必要がある。内容はもとより、記載方法の品質が担保されている状態にあるか確認した方が良い。これを怠ると、ドキュメントの品質レベルが下がる可能性がある。

勘どころ⑥ レビューの記録を残し、問題点などが分析できるようにしておく

レビューをすることが重要であると記載したが、レビューを実施したら必ずレビュー記録表を作成し、指摘内容などを記録しておく。これは、もちろん指摘事項がドキュメントに反映されているかを確認するのも当然であるが、指摘傾向、指摘件数などの情報を収集することで、今後のプロジェクト活動へのフィードバックにつなげ、プロジェクト全体の品質向上につなげていく狙いもある。

6.2.6 コミュニケーション計画(RM.2.6)

要件定義にはさまざまな職種のメンバが参加する。また、それぞれが担当する業務もさまざまであり、内容についての共通理解を確保するのは難しい。ここでは、要件定義における関係者間のコミュニケーションを進める上での問題と対策について述べる。なお、担当者相互のスムーズな意思疎通、認識の統一は物事を進める上での成功の鍵であり、ここで述べることは、要件定義に限らず、すべてのプロジェクトでのコミュニケーション強化に関連する話である。

(1) ステークホルダとのコミュニケーションを計画し、合意形成を図る

【解決したい問題】

- ステークホルダとのコミュニケーションが不十分なため、要件定義に不備が発生する
 - ・ しかるべき人への報告・共有がされず、手戻りが発生する
 - ・ 合意形成ができなかったり、時間がかかる

システム開発は、多くのステークホルダとプロジェクト体制を組んで進めていくものなので、ステークホルダとどのようにコミュニケーションを取るかが重要となる。

ステークホルダとのコミュニケーションが不十分な場合、上記のような問題が発生する。

このようなことを防止するためには、ステークホルダに対して、情報を伝える方法を意識して効果的に組み立てるようコミュニケーション計画を立案する。

勘どころ① コミュニケーション計画を立案する

コミュニケーション計画の際には、「伝達相手」「伝達内容」「媒体」「頻度」を適切に決めることがポイントとなる。以下にそのポイントを記載する。

- 伝達相手
ステークホルダの特定・分析によって明確となったステークホルダより、コミュニケーションが必要となる伝達相手を特定する。
- 伝達内容
相手の置かれた背景や立場、環境がそれぞれ異なるため、相手の状況を理解した上でのコミュニケーションが必要である。その際、相手の状況を理解することと、相手の主張に同意することは別であることを意識した上で、少しずつ意識のずれを

6.2 計画立案(RM.2)

埋めていく工夫が求められる。また、相手が求めているメッセージになっているか、分かりやすさ、簡潔さ、詳細さなどは伝達相手によって変える必要がある。

● 媒体

伝達相手と伝達内容に合わせて、適切なオンライン・オフラインツールを選択する必要がある。オンラインツールには、電子メール、電話、TV 会議等の各種コミュニケーションツールが挙げられる。また、会議などはオフラインツールである。相手に伝わる情報量では、電子メール→電話→TV 会議→会議の順で増える。

例えば、電子メールでは、相手は文章以外の行間を読むことまでは難しいが、電話では声の調子によってニュアンスまで伝えることができる。しかし、TV 会議のように相手の表情や場の空気を読み取ることは難しく、対面の会議にはかなわない。その一方、電子メールでは大人数に一度に情報を伝えることができるが、会議では場所や時間に制約がある。そのため、複数の媒体を組み合わせ、コミュニケーションツールの長所と短所を組み合わせることが有効である。

● 頻度

一般に頻度が多いほど相手との信頼関係も向上し、コミュニケーションの質も向上しやすい。しかし、討議が必要な場なのか通常の状況報告なのかなど、何を目的としているのかによっても、コミュニケーションに必要な頻度は変わる。例えば、プロジェクト内部の進捗会議であれば毎週、場合によっては毎日の方が良い場合もあるし、一方でシステムオーナーへの報告であれば月 1 回が適切な場合もある。伝達相手と伝達内容から最も効果的な頻度を設定することが望まれる。

勘どころ② コミュニケーションの質を上げ、合意形成を図る

上記のようなコミュニケーション計画の立案と合わせて、コミュニケーションの質を高めることも重要である。特にステークホルダが多岐にわたる場合は、各ステークホルダとの納得感を生む合意形成のルールが必要となる。

そのためには、ステークホルダとの議論内容や合意事項はなるべく詳細に議事録等に記録・配布し、共通認識を積み重ねていくことが重要である。さらに、プロジェクトの目的や目標を明確化し、業務部門からの要求をどのように優先順位付けするかといった手順を定めることが効果的である。

また、ステークホルダを一カ所に集めてコミュニケーションするのではなく、ステークホルダのもとに何度も足を運んで説明を繰り返し理解を求めたり、各ステークホルダが抱える個別の問題についても一緒に考えることにより、納得感を得られやすくなる。

ステークホルダが多いほど、納得してもらえらるルールを作り、直接説明する機会を増やすことがコミュニケーションの質を高めることに繋がり、合意形成を図る上では有効である。

6.2 計画立案(RM.2)

コミュニケーションの質向上に向けた会議体設定や、上位の意思決定の設定などは、事例 12「ステークホルダとの会議体設定による確実なコミュニケーション」を参照されたい。

事例 12 ステークホルダとの会議体設定による 確実なコミュニケーション

(セイコーエプソン株式会社)

【取り組みの背景】

プロジェクトを円滑に進めるためには、関係するステークホルダとの報告・共有をタイムリーに実施することが重要である。そのためには、ステークホルダと役割を明確にした上で会議体を設定し、定期的にコミュニケーションが図ることが有効である。

【課題】

本事例では、解決すべき課題を以下のとおり設定した。

- しかるべき人への報告・共有がされず、手戻りが起こる
- ステークホルダ間の連携が悪い。自部門のことしか考えてない

【解決策】

(1) ステークホルダとの会議体設定

ステークホルダとの会議体設定にあたっては、ステークホルダの立場や階層によって関心事や意思決定のレベルが異なるため、目的ごとに明確に会議体を設定する。その際には適切な頻度で定例会議として設定することが望ましい。ステークホルダとの報告・共有・合意形成に必要な会議体の一例を以下に示す（表 6.14）。

表 6.14 ステークホルダとの会議体設定

会議体	目的・内容	出席者	頻度
ステアリング コミッティ	・オーナーへの進捗状況の報告 ・ハイレベルなプロジェクト課題の意思決定	システムオーナー プロジェクト推進責任者 プロジェクト管理者	月次
プロジェクト 進捗確認会議	・プロジェクト進捗状況の報告 ・プロジェクト課題への意思決定	プロジェクト推進責任者 プロジェクト管理者	週次
開発 進捗確認会議	・開発進捗状況の報告 ・開発課題への意思決定	プロジェクト管理者 システム部門リーダー ベンダ責任者	日次
仕様検討会議	・要求・仕様の検討	プロジェクト管理者 システム部門リーダー 業務部門リーダー 利用部門リーダー	随時

(2) ステアリングコミッティの設置

プロジェクト当事者では解決できない事項やステークホルダ間のコンフリクトにより意思決定に時間がかかる場合がある。そのような場合は、プロジェクト内の最上位の意思決定機関として、ステアリングコミッティの設置を行う（図 6.17）。

ステアリングコミッティには、プロジェクト推進の主体者である業務部門責任者、システム部門責任者、プロジェクト推進責任者のみならず、システムオーナー、経営層といったプロジェクト上位関係者も交えた体制とし、有事の際のエスカレーションパスを明確にする。

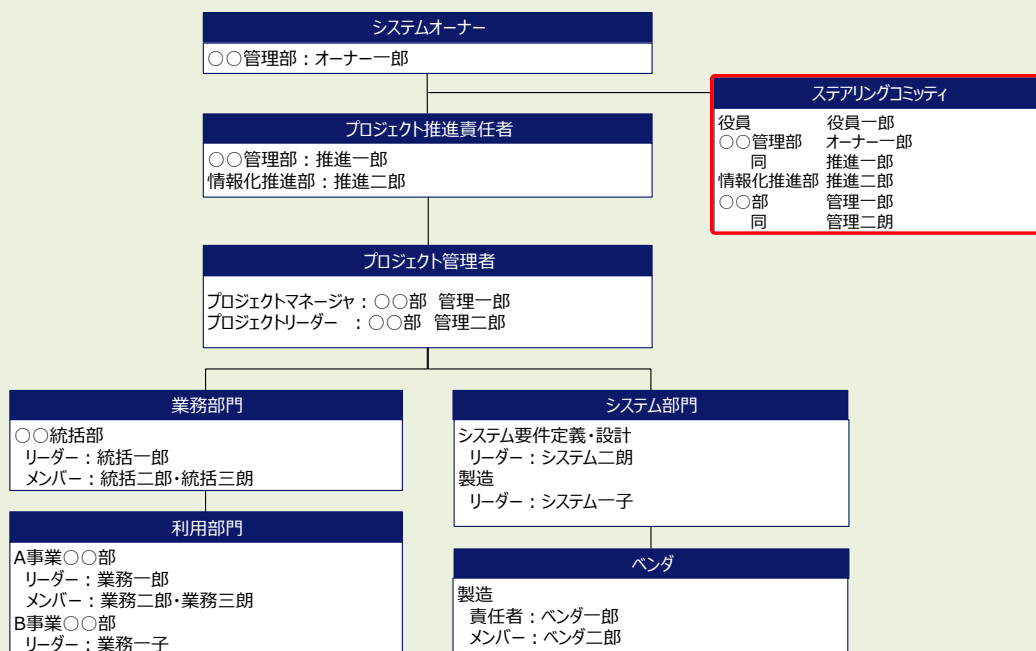


図 6.17 ステアリングコミッティの設置

【効果】

本事例では、ステークホルダとの報告・共有をタイムリーに実施することで確実にコミュニケーションを図る。その結果、プロジェクトを円滑に進め、手戻りを抑制する効果を得ることができた。

事例 13 要件定義を1カ月で実施した事例 (住友電工情報システム株式会社)

【取り組みの背景】

住友電工情報システム株式会社では住友電工グループ各社の基幹システムを構築している。ここでは、短納期で高品質が求められたシステムの要件定義の事例を紹介する。今回、管理方式の異なる2つの部署が統合することになり、生産管理システムの1サブシステムを再構築する必要があった。システム規模は約1100FP、90機能であった。稼働は年度替わりのタイミングに合わせるため、開発期間は6カ月しかない状況であった。納期を守るためには要件定義を1カ月で完了させる必要があり、また、開発中に要件変更が発生しない状態にする必要があった。

【課題】

本事例では、1カ月で質の高い要件定義を実施する必要があり、そのために以下の課題を解決する必要があった。

- 要求合意のペースを見極める必要がある。
- 参加者が17名と多く、会議日程をうまく調整する必要がある。
- 参加できない場合でも議論の流れが分かるようにする。
- 複雑な要件を合意する。
- 納期を守るために要件漏れをなくす。

【解決策】

(1) 要件ベースラインの設定

システム開発のために決定すべき要件が多く、利用部門で内部調整が終わっていない状況であった。一方、開発チームは旧システムの保守を担当していたため、決定すべき要件の優先順位、問題点・改善点、対策をリスト化することができた。要件の量が開発期間に収まる量ではなかったため、ビジネス上必須なものに絞り込み、納期までに開発できる範囲の要件を提案した。この要件がベースラインとなり、要件の入れ替えはあったものの総量はあまり変化しなかった。要件の詳細化と決定は、あいまい・未決定と思われる項目を優先的に取り上げることにした。

(2) 会議の計画

プロジェクトの関係者が17名と多いことから、上述の要件一覧をベースに週4回、1カ月間の会議をスケジュールし、それぞれの会議でどの要件を決めるか計画した。その計画を利用部門と緊迫感をもって合意することで要件合意のペースを決め、納期までに合意できないリスクを軽減した。また、誰がどの会議に出席すべきか判断できるようになり、キーマンが参加していないリスクも軽減させた。

(3) 説明資料と決定事項の一元管理

対象業務は複雑なロジックが必要であったため、開発部門の理解を図示し、コミュニケーション・ミスの防止に努めるとともに会議の進行を効率化した。会議中に変更があれば、その場で変更し、合意事項が残るようにした。また、会議で使用した説明資料と決定事項は1つのプレゼン形式の資料にまとめることで、会議に参加できなかった人も経緯が把握できるようにし、議論が円滑に進むようにした。この取り組みによりしかるべき人が報告を受けていないために手戻りが発生するリスクを低減した。

【効果】

以上の取り組みにより計画とおり、要件定義を1カ月で完了させることができた。さらに、重要な要件漏れはなかった。また、外部設計以降も納期短縮の工夫を重ねることで納期とおりシステムを稼働させることができた。

事例 14 多様化するステークホルダとの合意形成 (セイコーエプソン株式会社)

【取り組みの背景】

システムの大規模化・複雑化にともない、プロジェクトに関係するステークホルダも広範囲に広がり、かつ多様化している。そのような多様化するステークホルダと合意形成を図るには、事例7「ステークホルダの特定と分析」を元に、ステークホルダに合意を取り付ける際の方法（シナリオ）を事前に決めておくことが重要である。

【課題】

本事例では、解決すべき課題を以下のとおり設定した。

- 合意形成ができない。合意形成に時間がかかる。

【解決策】

(1) コンフリクト（対立関係）の解消

プロジェクト活動では、異なる立場のメンバが同時に活動するため、業務部門、システム部門、ベンダなどの各ステークホルダの立場によって関心事や優先事項が異なり、コンフリクトが発生する。コンフリクトの存在は、合意形成の障壁となることが多い。

コンフリクトが起こりやすいパターンを以下に示す（表 6.15）。

表 6.15 コンフリクトが起こりやすいパターン

ステークホルダ 合意内容	業務部門	システム部門
業務要求	他部門よりも自部門の要求を優先したい	シンプルかつ標準化したい
システム要件	手厚い仕様にしたい	最小限の仕様にしたい
システム変更要求	要求はいつでも変更したい	極力、変更は受けたくない
システムの QCD (品質/コスト/納期)	高品質は当然だが、早く安く稼働したい	計画時の QCD は必達したい

業務部門、システム部門、ベンダとの間で、上記のような複雑なコンフリクトが存在するため、ステークホルダ分析の中でコンフリクト解消に向けた対策を打つ必要がある。

コンフリクトを解消するためには、相互理解を深めることにより一致点と相違点を層別し、対立の原因（論点）を明確化する。また、対立の原因（論点）について、プロジェクトの目的に立ち返り、より高い視点やより広い視野から話し合い、互いに協力しながら第3の案を創造し、Win-Winの関係になるよう努力する必要がある。

しかし、必ずしもいつも第3の案が創造できるとは限らない。また、システム要件やシステム変更要求の場合、コストやスケジュールが超過しないように、要求を抑え込もうとするシステム部門のプロジェクトマネージャと業務部門との対立はよく発生する。そのような場合、強引に結論を急がず、何を優先するかについて、参加者全員が納得する評価基準をあらかじめ決めておくが良い。

どの要求を優先するかは、対象システムが何を重要視するかによって異なるが、例えば、法令順守、リスク回避、プロジェクト目標達成への寄与、効率化といった観点で要求の優先度付けをすることで、ステークホルダの納得感を高め、コンフリクトを解消するようにする。

(2) ステークホルダに合わせた合意形成の方法

ステークホルダリスト（表 6.6）で、ステークホルダとその役割、影響度、対立関係の分析を行った結果と、合意を取り付ける内容を基に、合意形成の方法をあらかじめ決めておき、確実に合意形成を図る。以下にその事例を示す（表 6.16）。

表 6.16 ステークホルダに合わせた合意形成方法

ステークホルダ	合意を取り付ける内容	合意形成の方法
業務部門	業務要求・システム要件	(a) 要件定義 10 箇条 および、要件定義レビュー
	システム変更要求	(b) 変更要求管理
	システムの QCD	(c) システム DR（デザインレビュー）
各事業・関係会社	各事業・関係会社への影響	事業連絡会、グローバルミーティング
経営層	経営方針との整合	経営会議

例えば、業務部門に対して、システム変更要求について合意を取り付ける場合には、変更要求管理の合意プロセスを経ることによって、確実に合意形成を図る。

ステークホルダに合わせた合意形成（表 6.16）のうち、以下の合意形成方法に

6.2 計画立案(RM.2)

ついて詳細を示す。

- (a) 要件定義 10 箇条
- (b) 変更要求管理
- (c) システム DR (デザインレビュー)

なお、要件定義レビューについては、事例 5 「定量的に捉えた手戻りコストとレビュープロセス成熟度向上による手戻り改善」を参照されたい。

表 6.17 要件定義 10 箇条

項目	分類	ポイント
1. 要望と要求の切り分け	業務要求	<ul style="list-style-type: none"> ・ 要望と要求を区別し、不必要な要望は検討の対象から外す ・ 要求では、実現すべき目的を明確にする
2. 現状業務の把握・理解		<ul style="list-style-type: none"> ・ 業務要求の対象となる業務プロセスの現在の姿を理解する ・ 具体的には、業務の目的、業務フロー、使用する帳票、現在使用しているシステム、法令などの制約
3. 業務要求の理解		<ul style="list-style-type: none"> ・ 業務要求を漏れなく洗い出すための To-Be 業務の詳細化と、経営上の問題／課題の解決に寄与するかの確認をする ・ 業務要求一覧と業務フローを作成する
4. 5 ゲン主義に基づく整理		<ul style="list-style-type: none"> ・ 5 ゲン主義 (3 現主義「現場」「現物」「現実」に「原理」「原則」を加えた考え方) で業務要求を整理し、俯瞰的な視点で要件定義を実施する
5. 業務上の問題・課題の明確化		<ul style="list-style-type: none"> ・ 業務上の問題・課題を確認できる一覧を作成する ・ 対応の有無、実現手段 (システム or 業務変更) を決定する
6. システム化範囲の明確化	システム要件	<ul style="list-style-type: none"> ・ システム構成図を作成し、システム化範囲を決定する ・ As-Is と To-Be の両面から明確化する
7. 要件定義への落とし込み		<ul style="list-style-type: none"> ・ システム要件定義書を作成する ・ 要求、目的／理由、システム仕様、制約などを明記する
8. システム化の品質・コスト・納期		<ul style="list-style-type: none"> ・ システムの QCD のバランスに配慮し検証する ・ 無理な納期設定による品質低下、実現する要求に対する費用対効果、過剰品質による納期設定を検証する
9. 業務部門との合意形成	合意形成	<ul style="list-style-type: none"> ・ システム化範囲、システム要件、システムの QCD などを業務部門と合意する
10. 調整会議の効率化		<ul style="list-style-type: none"> ・ ヒアリングを効率的に行うために、会議の目的、出席者、会議の進め方、ファシリテーション方法、会議サイクルなどを事前に会議設計する

(a) 要件定義 10 箇条

要件定義の抜け・漏れ・誤りにより手戻り防止と、業務部門との確実な合意形成を目的に、要件定義者が要件定義で何をすべきかをまとめた「要件定義 10 箇条」を作成した（表 6.17）。

業務要求、およびシステム要件は、本 10 箇条に沿った活動と要件定義レビューにより、業務部門と合意形成を図る。各項目のポイントを表 6.17 に例示する。

(b) 変更要求管理

要件定義の終盤、もしくは要件定義後の要求の変更は、手戻りによるシステムの QCD に与える影響が大きいため、事前にルール化しておくことが重要である。既に承認された要求・要件については、ベースラインを設定し、以降の変更は変更要求管理の手順を通してのみ変更できるルールとすることで、変更要求に関して、業務部門とシステム部門との間で合意形成を図る。

変更要求においては、第 4 章の問題で述べているような要件定義の不具合に起因する後工程からの手戻りと同様に、開発工数の増大や開発工期の遅延につながるが多いため、慎重な判断が求められる。その際には、事前にルールを決めておき、できるだけ機械的に判断できるようにしておく、業務部門の納得性が高く、合意形成を図りやすい。

具体的には、以下のような変更要求管理フローとなる（図 6.18）。

- ① 課題リストの中で、変更要求につながる項目を「変更要求リスト」に記載し、一覧管理する（表 6.18）。
- ② 変更要求の実施可否を評価する。その際、以下のような点がポイントとなる。
 - ・ 合目的性：システム構築の目的と合致しているか
 - ・ 効果への貢献度：効果創出への貢献度はどれ位か
 - ・ システムの QCD への影響：プロジェクト全体の QCD などの制約事項への影響度合いはどれ位か。変更による既存システムへの影響範囲の特定とその見積りを行う
 - ・ 実施タイミング：どのタイミングで実施するのが適切か
- ③ 上記評価内容をプロジェクト管理者（PM）以上の責任者が総合的に判断し、各変更要求の実施可否を承認する。
- ④ 実施可として承認された要求については、関連ドキュメントを更新するとともに関係者に周知し、ベースラインに取り込む。

6.2 計画立案(RM.2)

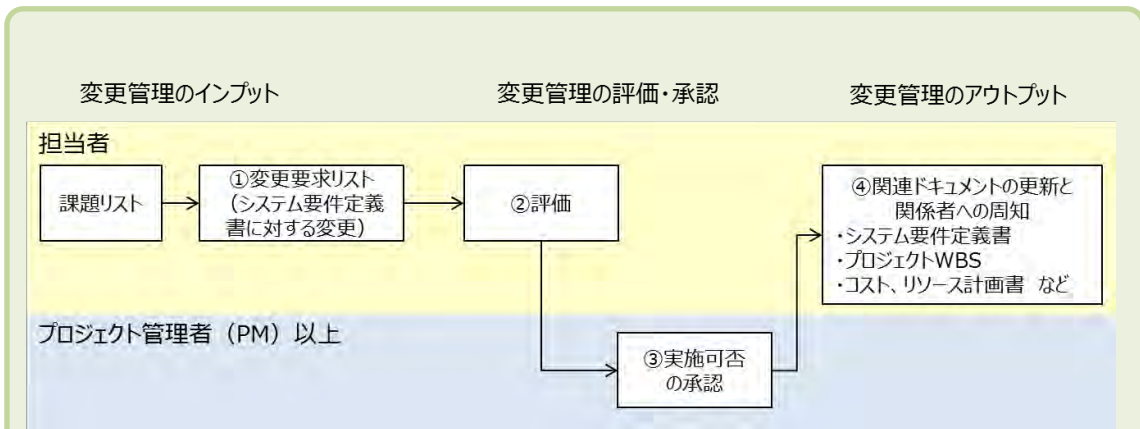


図 6.18 変更要求管理フロー図

表 6.18 変更要求リスト

ID	起票日	起票者	変更内容	変更理由	影響	対応可否	承認者	完了日

(c) システム DR (デザインレビュー)

システム DR (デザインレビュー) 基準を制定し、必須の実施事項としている。システム DR は、システム構築における各工程での成果物を、複数のステークホルダで確認することで、システム構築におけるシステムの QCD 確保とステークホルダとの合意形成を目的としている。具体的には、システム化構想から本番稼働後の効果確認までの開発ライフサイクルの中で、節目となる地点で、チェックリストのレビュー項目を基に、各チェックポイントに従った確認を行う (図 6.19)。

デザインレビューの参加者が、成果物について指摘し合うことにより、対象工程での問題・課題を業務部門とシステム部門の双方で共有し、合意形成を図る。その際には、各ステークホルダが当事者意識を持って活発な議論を行い、単なるセレモニーとしないようにすることが重要である。

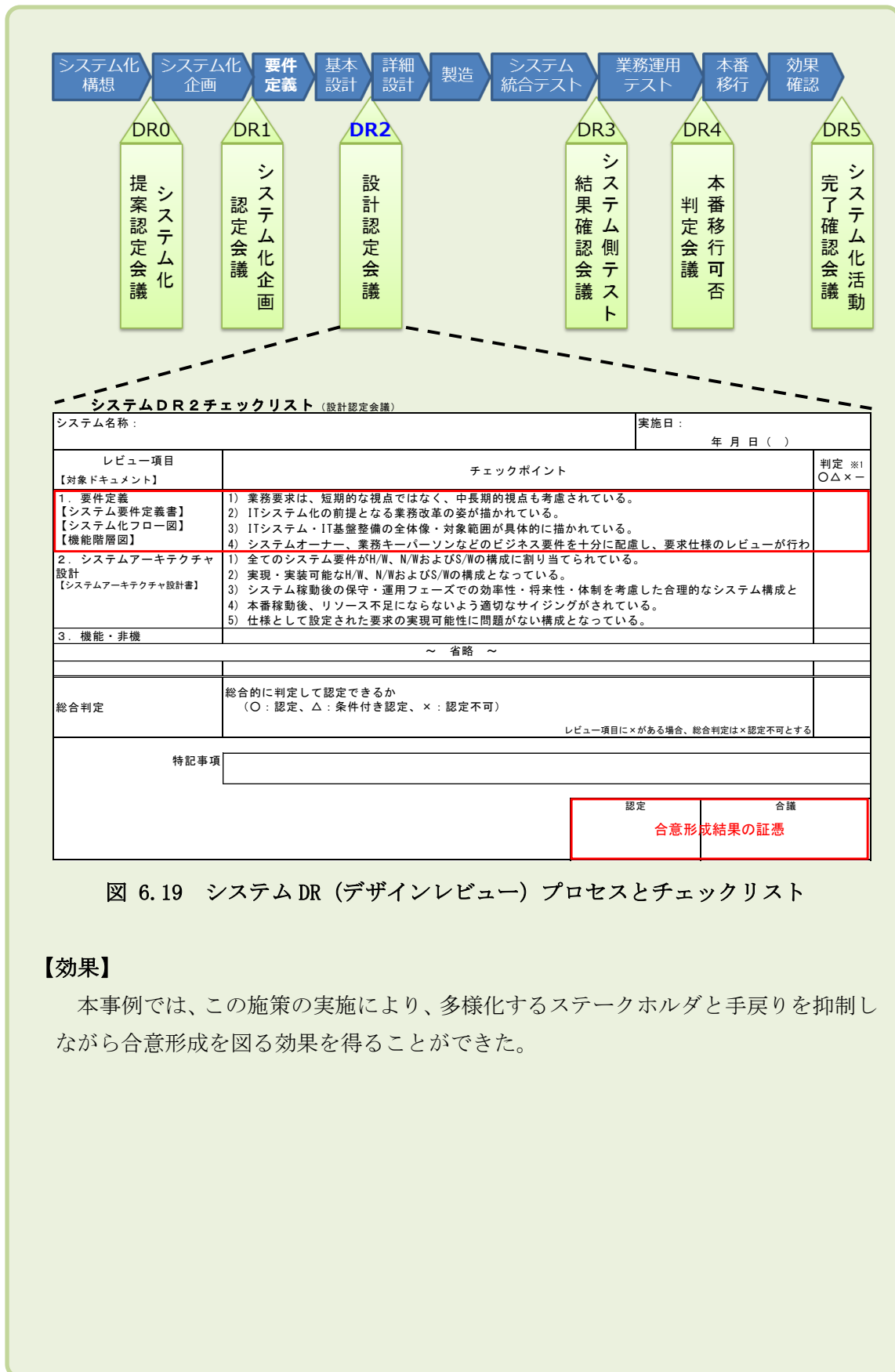


図 6.19 システム DR (デザインレビュー) プロセスとチェックリスト

【効果】

本事例では、この施策の実施により、多様化するステークホルダと手戻りを抑制しながら合意形成を図る効果を得ることができた。

6.2.7 リスク認識(RM.2.7)

リスクとは、現時点で発生していないが発生したらプロジェクトの遂行に大きな影響をもたらすことが予想される事象のことである。そのため、プロジェクトの遂行時には事前にリスクを認識し、対策を立てておく。要件定義においてもリスクの識別と事前の対策は重要な意味を持つ。ここでは、要件定義におけるリスクの顕在化に対して、どのような対策を事前に計画するかについて説明する。リスク認識においては、過去にどのような事象が実際に発生したかを蓄積し、そこから各プロジェクトでどのリスクに対して対策を取るかを選択するアプローチも有効である。そのような対策を事例で紹介する。

(1) リスクの特定と対応計画の立案により、プロジェクトの失敗を防ぐ

【解決したい問題】

■ リスクが顕在化し、プロジェクトが止まる

こんな苦い経験はないだろうか。

- 要件を決めるのに話を通すべき部署が複数存在する中、要件定義終盤でなかなか要件がまとまらずに、要件定義工程のスケジュールが遅延した
- 他の業務が忙しくなり、本開発案件に十分に時間が割けず、要件定義を予定とおりに進められなかった
- 個人情報や開示範囲指定情報を取り扱う要件があるにもかかわらず、取り扱い方を要件定義工程で決めてなかったために、開発プロジェクト終盤、大きな設計変更を余儀なくされ、開発コストが大幅に超過した

これらの問題は、要件定義開始時に具体的に起こっておらず、それが故にプロジェクト上の課題として認識されない。しかし、起こってから対策を検討しているのは被害が広がり、スコープ、スケジュール、コスト、品質に悪影響を与えてしまい、プロジェクトの失敗に直結することすらある。よって、これらプロジェクト目標に影響を与える不確実な事象や状態を、あらかじめ「リスク」と捉えて、要件定義前の計画段階で、回避策や緩和策を検討しておく必要がある。

以降、リスクの特定から対応計画を立てるまで（勘どころ①～③）を解説する。

勘どころ① リスクを特定し、リスク登録簿に記載する

発生可能性のあるリスクを洗い出すために、要件定義前のコンセプト検討資料や、その時点の課題、体制、スケジュールをレビューする。また、他案件でしばしば起こる QCD 悪化原因が本案件でも発生しないか確認する。

リスクは次の 2 種類あり、両方とも洗い出す必要がある。

- プロジェクトリスク（開発チーム観点）
品質低下・コスト超過・納期遅延など、開発プロジェクトの失敗原因となるリスク
- プロダクトリスク（ユーザ観点）
取引先や得意先への支払漏れ・個人情報漏洩など、重要要件を満たせず大規模障害となるリスク

次に、洗い出したリスクを、一覧形式のリスク登録簿に記載する。記述する項目例は、リスク事象、リスク原因、発生確率、影響度、対応優先度、対応策、対応担当者、対応日、対応ステータスである。

勘どころ② リスクを分析する

①で洗い出したリスクの発生確率と影響度を分析し、リスク対応の優先度を定める。発生確率は、「高」「中」「低」、影響度は、「大」「中」「小」で表す。その掛け合わせから対応優先度「高」「中」「低」を決める。例えば発生確率が高く、影響度が大きいリスクは、最優先で対応すべきである。

勘どころ③ リスク対応計画を立てて、スケジュールに反映する

②の対応優先度の順に、対応策を検討する。対応策には、次の 4 種類がある。

表 6.19 リスク対応策

対応策	対応内容
回避	リスクを完全に排除する
転嫁	リスクの影響を責任とともに第三者に移管する
緩和	リスクが顕在化した被害を受容可能な限界値まで抑える
受容	リスクを受け入れ、事象が発生したときに対処する

対応策は当たり前だが実施されなければ、リスクが顕在化したときに案件への影響を

6.2 計画立案(RM.2)

最小限にとどめられない。対応策を立てるだけで終わらずに、対応担当者、対応日も決めて計画し、リスクを未然に防いでいく活動に反映していくべきである。

特に、プロダクトリスクの対応は、要件定義工程で入念に要件を詰めていきつつ、設計工程での妥当性確認をしたり、設計レビュー・テスト工程でしっかり欠陥検出したり、サービスイン後に早期発見すべく稼働検証したりする必要がある。

特定したリスクを、危機意識としてプロジェクトチーム内やステークホルダに共有するとともに、リスクが顕在化してないか監視しつつ、決めた日までに対応策を実施していく。なお、ステークホルダにすべてのリスクを共有しづらい場合もあるので留意したい。

「オーナーが優柔不断でジャッジや調整ができず、決定遅延や手戻りの原因となる」といったリスクはよくあり、この場合はオーナー所属会社以外のプロジェクトチーム内で共有すると良い。

事例 15 「プロジェクト危ないことチェックシート」で
要件定義開始前にリスク特定
(サントリーシステムテクノロジー株式会社)

【取り組みの背景・課題】

リスクの特定と対策がプロジェクトマネージャ個人の能力に依存し、顕在化していないリスクへの備えが甘いプロジェクトがしばしば見受けられ、結果、納期遅延や品質低下を招いていた。また、そのプロジェクトのリスクと備えを第三者が把握しづらく、適切なアドバイスや意思決定を妨げていた。

【解決策】

サントリーシステムテクノロジーでは、サントリーのシステム部門内で標準化した「プロジェクト危ないことチェックシート」を使用しており、例えば、システム担当者が一定金額以上の規模の開発に携わった経験、プロジェクトのスコープ対象業務の開発経験、そのプロジェクトの専任度、など、51 のチェック項目を要件定義開始前にチェックしている。

この「プロジェクト危ないことチェックシート」を使用することで、次の効果を得られる。

- システム担当者の能力に依存せず、危ないことを特定することができる
- デザインレビュー時、チェック内容を第三者に伝え、そのプロジェクトの危なさを共有し、アドバイスをもらえる

51 のチェックそれぞれに A、B、C のランクが設定されており、A は QCD への悪影響が大きい選択肢であり、C は小さい選択肢である。例えば、プロジェクトリーダーが一定金額以上の規模の開発に携わった経験がなければ A ランク、主担当ではないが携わった経験があれば B ランク、主担当として携わった経験があれば C ランクという具合である。

51 のチェックの観点は次のとおりである。

- 人・体制面のチェック（システム部門・ベンダ・業務部門）
- スケジュール面のチェック（工期の厳しさ）
- 費用面のチェック（目標コストの厳しさ）
- 法的なチェック（契約・著作権など）
- インフラ（新規のハード／ソフト・開発環境の整備）
- システム特性（処理方式の新規性など機能性・性能・再構築特性など）

6.2 計画立案(RM.2)

- ビジネスリスク（請求機能など得意先に影響する機能実現）

次に、Aランクのチェック内容をベースにリスクの特定をする。例えば、システム担当者が一定金額以上の規模の開発に携わった経験がない場合（=Aランク）は、要件定義の進め方が分からず、要件定義工程の遅延や、コスト増がリスクとなる。そして特定したリスクの影響度や起きうる可能性を整理し、検討した対策を「プロジェクト危ないことチェックシート」に記入する。その後、その対策の担当者や実施予定日を決めてスケジュール化していく。

以下に、「プロジェクト危ないことチェックシート」のフォーマット（一部）を記載する。

システム	ランク別項目を「開発計画書」に記載		リスク	説明	リスクのレベル(1=High)	要件リスク
担当者			A	200万円以上の開発に携わった経験がない	1	High
			B	500万円以上の開発に携わった経験がある	2	Middle
			C	1000万円以上の開発に携わった経験がある	3	Low
			総数	5項目中3項目		

大分類	中分類	チェック項目	ランク	適合	リスク	担当者	起票日	具体的な対策	対応担当者	対応予定日	ステータス		
人・体制	システム担当者	システム開発経験	A	200万円以上の開発に携わった経験がない									
			B	500万円以上の開発に携わった経験がある									
			C	1000万円以上の開発に携わった経験がある									
		業務精通度	A	当該業務(1)の20%以上の経験がない									
			B	当該業務(1)の40%以上の経験がある									
			C	当該業務(1)の60%以上の経験がある									
		PM経験	A	PM経験が1年以上(1PM経験のみ)									
			B	PM経験が2年以上(2PM経験以上)									
			C	PM経験が3年以上(3PM経験以上)									
		スキル	A	必要スキルを100%以上保有している(100%以上のスキル)									
			B	必要スキルを80%以上保有している(80%以上のスキル)									
			C	必要スキルを60%以上保有している(60%以上のスキル)									
		専任度	A	20%以下(他業務)の専任(100%専任)									
			B	30%以下(他業務)の専任(70%専任)									
			C	40%以下(他業務)の専任(60%専任)									
システム規模に対する担当者の人数	A	1担当者が1000万円以上の開発に携わっている											
	B	1担当者が500万円以上の開発に携わっている											
	C	1担当者が200万円以上の開発に携わっている											
協力会社	システム開発経験 (PMおよびSE)	A	協力会社担当者が1年以上の開発に携わっている										
		B	協力会社担当者が2年以上の開発に携わっている										
		C	協力会社担当者が3年以上の開発に携わっている										
	業務精通度 (PMおよびSE)	A	当該業務(1)の20%以上の経験がない										
		B	当該業務(1)の40%以上の経験がある										
		C	当該業務(1)の60%以上の経験がある										
	専任度 (PMおよびSE)	A	20%以下(他業務)の専任(100%専任)										
		B	30%以下(他業務)の専任(70%専任)										
		C	40%以下(他業務)の専任(60%専任)										
	技術面からの体制 (SE、PM、PG)	A	100%専任(100%専任)										
		B	80%専任(80%専任)										
		C	60%専任(60%専任)										
	開発体制の構築	A	当該業務(1)の20%以上の経験がない										
		B	当該業務(1)の40%以上の経験がある										
		C	当該業務(1)の60%以上の経験がある										
主管部署	業務精通度	A	当該業務(1)の20%以上の経験がない										
		B	当該業務(1)の40%以上の経験がある										
		C	当該業務(1)の60%以上の経験がある										
	レスポンス、約束期限の遵守状況、決まりを守る度合い	A	遅延が頻りに発生している										
		B	遅延が時々発生している										
		C	遅延がほとんど発生していない										
	関係部署の多さ(仕様調整の複雑さ)	A	仕様調整が頻りに発生している										
		B	仕様調整が時々発生している										
		C	仕様調整がほとんど発生していない										
	システム開発への参加度合い	A	当該業務(1)の20%以上の経験がない										
		B	当該業務(1)の40%以上の経験がある										
		C	当該業務(1)の60%以上の経験がある										
	システム構築・開発・導入の経験	A	システム構築・導入の経験がない										
		B	システム構築・導入の経験がある										
		C	システム構築・導入の経験がある										
全般	関係者等に対する配慮	A	当該業務(1)の20%以上の経験がない										
		B	当該業務(1)の40%以上の経験がある										
		C	当該業務(1)の60%以上の経験がある										

図 6.20 プロジェクト危ないことチェックシート

特定するリスクと対策の例は次のとおりである。

表 6.20 リスクと対策例

チェック項目	リスク	対策
PM/メンバの 掛け持ち	システム担当の A さんが別の一定金額以上の他案件と掛け持ちしていて、このプロジェクトに工数を予定とおりに割けない	<リスクの低減> A さんが担当している他案件を別のメンバに引き継ぎ、掛け持ちを解消して要件定義に集中させる。
技術面からの 体制(SE、アーキテクト、PG)	オフショア先のメンバはサントリーのシステム改修が初めてなので、知見不足でQCD悪化	<リスクの低減> 要件定義工程は日本メンバがすべて実施。日本メンバが標準に準拠した設計書をすべて準備してオフショア先で解説を行う。
関係部署の多 さ(仕様調整 の難易度)	複数の責任部署の要件が異なり、要件の合意形成が困難	<リスクの低減> 業務部門を代表しプロジェクトオーナーを決める。 要件の合意形成の段取りをあらかじめ計画する。

6.2.8 調達計画と契約(RM.2.8)

要件定義を自組織内のメンバだけで実施することは少ない。一般に、要件定義の実施に必要な業務・技術スキルを保有する人員を外部から調達することがほとんどである。ここでは、外部から人員を調達する際の契約形態が、要件定義の成果物の完成度に影響することを述べる。要件定義は一般に、新たな発想を絞り出すプロセスの連続であり、定められた成果物を定められた期日までに一定の品質で完成させるという業務とは本質的に異なる。要件定義におけるこのような特性と適合した契約形態で人員を外部調達することが必要である。

(1) 要件定義は準委任契約を基本とし、ユーザ企業主体で進める

【解決したい問題】

- 要件定義の契約形態が実施内容と合っていないため、トラブルが発生する
 - ・ 漠然とした要求にも関わらず、作成請負で要件定義をベンダに委託してしまう

勘どころ① 要件定義の契約基本形は準委任契約とする

要件定義を IT ベンダに委託する際には契約形態を準委任にすることが、経済産業省の「情報システム・モデル取引・契約書(2007・4)」[8]で推奨されている(図 6.21 参照)。そこでは、その後の基本設計(図 6.21 ではシステム設計と記載)は請負または準委任契約を推奨すると記載されている。図 6.21 に示す契約モデルが発行される前は、ソフトウェア開発は全工程一括請負が大半であったが、問題が多発したことから、この契約方式に切り替え、開発に関するトラブルは減少した。

システム開発は仕様が明確に決まっていれば、後はそんなに大きな問題が発生することは多くないが、もし基本設計以降の作業管理者が「要件定義仕様決定が不十分なので、基本設計に入れたい」と感じた場合には、基本設計も準委任契約とした上で、要件定義での仕様決定が十分であることを確認する必要がある。

また、再構築の場合には、現行仕様の理解不足である場合が多い。そのため、要件定義に入る前に現行仕様の調査し分析する工程を設けるなど、要件定義を円滑に進めるための考慮が必要だ。

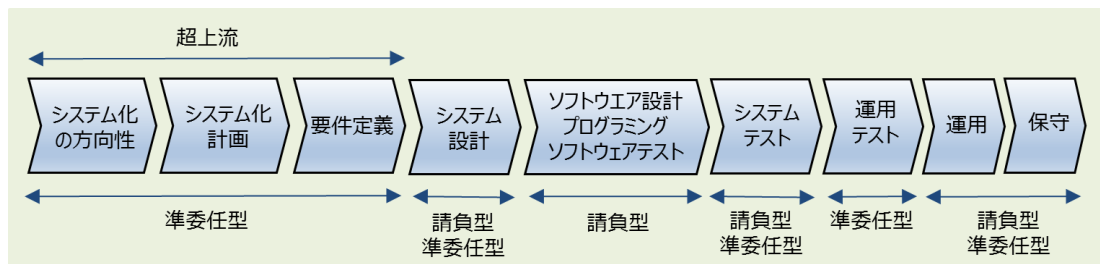


図 6.21 信頼性向上・取引可視化のための標準モデル

(出典) 経済産業省：情報システム・モデル取引・契約書（2007・4） [8]を基に作成

全工程を請負開発で行うことのリスクを示したデータとして、契約方式と発生欠陥数の関係の分析結果を表 6.21 に示す。全工程を自社開発で行った場合と全工程を請負開発で行った場合を比較すると、換算欠陥率の差が特に大きいことが分かる。「すべてお任せ」の請負契約は、最初の要件定義工程からユーザの参加割合が低くなりがちで、最後の総合テスト時点になって「これでは使えない」とユーザが変更を申し出るパターンが多く、その結果プログラム修正に追われて、品質低下に陥るケースが多い。要件定義を自社開発または準委任契約とすることにより、ユーザに責任の自覚と覚悟が高まり、結果としてプロダクトの品質が高くなる。

表 6.21 契約方式と発生欠陥数の実態

フェーズごとの契約形態			換算欠陥率（注）	
要件定義	設計	実装	件数	平均値
委任	委任	委任	29	0.38
委任	委任	請負	11	0.36
委任	請負	請負	57	0.27
請負	請負	請負	92	0.62
自社開発	自社開発	自社開発	39	0.24

(注)換算欠陥率：欠陥を大中小に分類し、重み付けをした欠陥数／全体工数

(出典) 日本情報システム・ユーザー協会「ソフトウェアメトリクス調査 2012」 [29]を基に作成

要件定義の各成果物の作成において習得した対象業務に関する知識や、各ステークホルダとの間で構築した信頼関係をそのまま活用できることから、要件定義を依頼したベンダに、それ以降の工程の作業を引き続いて依頼することは多い。そのような場合でも、例えば「もし要件定義に未決事項が何件かあり、その解決にかかる負荷が本来の要件定義作業工数の 10%を超える見込みである場合には、基本設計の契約形態は引き続き準委任

6.2 計画立案(RM.2)

とする。」などの取り決めを何らかの形でユーザとベンダの間で合意しておくことにより、ユーザ側の自覚を促しシステム開発でのトラブル発生を避けることも考えておきたい。

6.2.9 プロジェクト管理計画の作成(RM2.9)

要件定義の作業が開始される前に、プロジェクト管理の仕組み（進捗管理、課題管理、変更管理、品質管理、リスク管理のプロセスやルールおよび会議体など）を検討して文書化して、プロジェクトメンバへの周知徹底のみならず、経営層や上位マネジメント層にも説明して協力の約束を得ておく。こうした管理のルールは最初に決めて周知しておくことが大事で、後からでは協力を得られないことも多い。

要件定義が計画とおりに進められるか否かはプロジェクト管理計画次第と言っても良い面があるので、しっかり準備したい。

(1) プロジェクト管理計画(管理の仕組み)を事前に作成し、経営層や上位マネジメント層を含む関係者と合意する

【解決したい問題】

- 要件定義が始まってから、後付けでプロジェクト管理のルールを整備しがちである

要件定義は関与するメンバが多岐にわたるにも関わらず、プロジェクト全体に占める工数割合が10-15%程度で、次工程以降に比して参画するメンバの数も限られているため、プロジェクト管理計画(管理の仕組み)をしっかりと整えないまま、要件定義が行われることも多い。その結果として、スケジュールの大幅な超過や品質低下、大量の残課題を残したままの要件定義終了といった事態も起きている。そうした事態を少しでも減らすために、要件定義のスタート時点で、プロジェクト関係者(上位マネジメントや利用部門の方も含む)には、プロジェクト管理計画(仕組み)をよく理解してもらい、それを遵守する合意を得ておくことが大事である。

勘どころ① 要件定義の開始時にプロジェクト管理計画を作成し、プロジェクトの関係者に周知する

プロジェクト管理計画とは、要件定義の予定作業を予定期間内に予定した品質で終了させるための管理の仕組みを定めたものである。特に進捗管理、課題管理、変更管理、品質管理、リスク管理に関する管理ルールを取り決めておくことが大事である。

進捗管理に関しては進捗度合いを測る基準や進捗会議の運営、進捗会議で報告する内容や、作業遅延が大きくなった際の対応方針(何を優先して何を先送りし、次工程に回してもやむを得ないとするのか)などを明文化しておく。

6.2 計画立案(RM.2)

また、課題管理も重要である。課題に挙げたまま、なかなか解決策が固まらないことが非常に多いが、その背景には、利害関係者が多いことや、決める責任者が不明確なことなどがある。そのため、解決期限を超過した課題に対しては、課題区分ごとに誰が決めるか、どの組織、会議体にエスカレートして決めるかなどの一定のルールを決めて、関係者間で認識を合わせておく必要がある。また、変更管理についても、システム企画やシステム化計画段階におけるスコープからの変更や、要件定義内において確定している要件などへの安易な変更を抑える仕組みを、用意しておく必要がある。

品質管理も非常に重要である。要件定義における品質目標としては、プロジェクトの目的と整合性が取れていることや、ビジネス要求、業務要件、システム要件に抜け、漏れ、間違いがないことである。要件定義の成果物の品質を確保するために、有識者によるレビュー・承認が必須で、成果物を誰が見て、誰が承認するのかのルールのほかに、レビューにどのくらいの時間をかけるべきなのかも有効な管理指標となる。リスク管理も当初のスコープ範囲やスケジュールから外れる要素の発生リスクを管理して、予防していく仕組みを考えておかなければならない。

これらの仕組みを文書にまとめて、プロジェクト関係者間でレビューを行い、承認を得ておく必要がある。要件定義工程を円滑に進めるためには、こうしたプロジェクト管理の仕組みについて、スタート時に作成の上、関係者全員が理解し、合意している状態を作っておかなければならない。

勘どころ② プロジェクトオーナーから、経営層、上位マネジメント層にもプロジェクト管理計画を周知してもらう

また、要件定義工程では、経営層や上位マネジメント層に意思決定の依頼や承認をお願いする場合も多い。しかし忙しい方が多いために、日程を守ってもらうことが難しい。そのため、プロジェクトのオーナーから経営層や上位マネジメント層に、プロジェクト管理計画の骨子と、守っていただきたいことを、よく話ししておいてもらうと効果がある。何事も最初が肝心と心得て、上位層のプロジェクトへの協力の約束を取り付けて、要件定義が円滑に進められるベースを作っておくことをお勧めしたい。

6.3 監視・コントロール(RM.3)

6.3.1 要件量・スコープ(RM.3.1)

要件定義の初期段階においては、ユーザの業務を刷新する上で真に求められるシステム機能を抽出するために、あえて好きなだけ要求を抽出してもらうプロセスを設けることがある。業務の刷新には新たな発想が必要であり、要求は増え続ける。一方で実現できる要求には限りがあり、どこかで制限をすることが必要になる。要件量・スコープのコントロールは要件定義において最も必要な事項であり、多くの要求から最も価値あるものをいかにして選択して制約の範囲内に収めることができるかが、要件定義の成否を握っている。ここが要件定義マネジメントにおいて最も重要な要素である。

(1) 膨らむ要求を適正量に導く

【解決したい問題】

■ 得てして膨らんでしまう要求をコントロールできない

要件定義において要求は膨らむ。昨今さまざまな仕組みが IT 化され、身の回りにあるシステムから想起されるアイデアが溢れんばかりにある状況で、プロジェクトが開始する。また、実際に要件定義が進むにつれて、現行業務、現行システムに潜在する問題や課題が明確になったり、新業務、新システムのイメージが明確にされたりする。そうになると、さらにさまざまな要求が出てくるのだ。

一方で工期やコストが有限であることも事実である。そうした中で、必要不可欠な要求を捉えた上で、優先度の低い要求をスコープ外にするなどの調整が不可欠となる。

加えて、単にシステム部門が業務部門に「優先度が低いので▲▲機能はスコープ外にしました」や「要求が多過ぎるので、要求一覧から〇〇件をスコープ外としました」と一方的に伝えたとしても納得性もない。最終的には、多くの要求を実現したい業務部門と要求を絞りたいシステム部門がお互いにわだかまりを抱えたまま譲歩して、要求を落ち着かせることになる。正に「2:4:2:3 の法則」を地で行く結果となってしまう。

本節では、要求をいかに漏れなく抽出しながら、かつ納得性をもって絞込んでいくかの具体的方法の例を示す。

6.3 監視・コントロール(RM.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る

要件定義には2つのフェーズが存在する。それは、「要求を発散させるフェーズ」と「要求を収束させるフェーズ」である。

要件定義開始直後は、業務部門もこれからの業務や次のシステムをどうすべきかが明確でないことが多い。その際、要求が膨らむことを恐れて要件定義前に挙がっている課題や要求に絞って収束させてしまうと、後から後から要求が出続けて要件定義が一向に終わらなかつたり、要件定義を終えたとしてもその後の開発で仕様変更がいくつも起こってしまつたり、稼働後にユーザ満足度が低下してしまつたりする恐れがある。よって、要件定義の前半に、発散フェーズを設けて、ユーザへの要件ヒアリング時に、否定をせずに要件を聴いたり、他にどんな要件がありますか？と漏れのないよう聴いたり、なぜその要件が必要なのですか？とその要件の必要性や要件の背後にある真の要件を聴いたりしていく。

ただ、いくら発散フェーズだからと言って、ユーザが望むコストと工期を無視して夢物語の要求を検討することは、ユーザにとっても開発者にとっても得策ではなく、要件定義工程の手戻りコストを生んでしまいかねない。そこで、例えばコスト制約の大きい案件であれば、要件定義での各検討単位(=業務・要件・課題などその案件で検討しやすい単位)で、最初の1時間のヒアリングの場のみ発散フェーズにすることをユーザと合意する。また、工期制約の大きい案件であれば、要件定義の最初の五分の一の工期のみを発散フェーズにすることをユーザと合意する。こうして要件定義工程をコストや工期面でコントロールされた状態になるよう計画する。

一方で要件定義の後半では、業務やシステムが具体化されていく中で要求も膨れがちになる。そのため、何らかの基準を用いた要求管理ルールに則って、優先度を考慮して要求の採否を判断する必要がある。要求管理ルールも厳格にし、要求として収束に向かわせなければならない。その際、要求の絞り込みを業務部門のみに任せるのではなく、システム部門やベンダが知見や経験を活かし、ファシリテーションをすると良い。

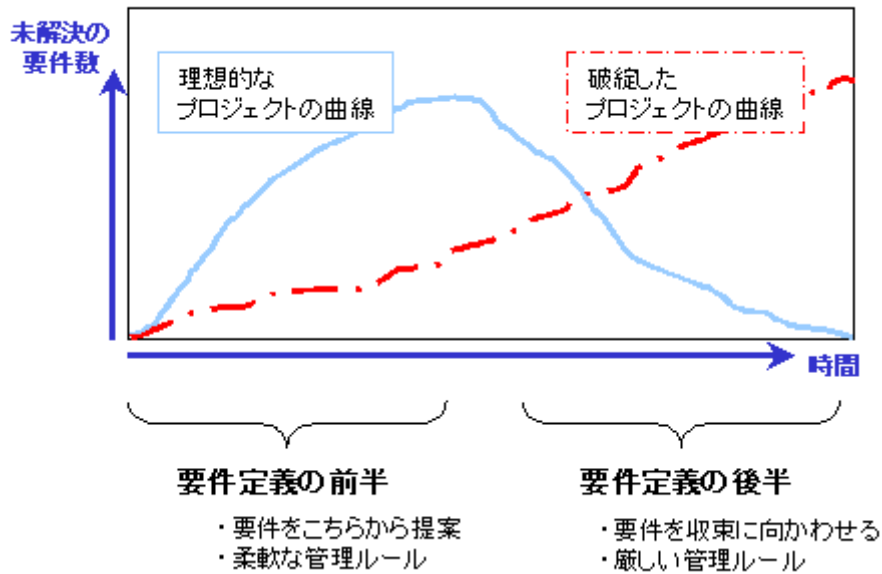


図 6.22 要件定義のフェーズを意識した要求コントロール例

勘どころ② 定量化した要件量の中で要件を調整する

要件に優先順位をつけて、どこまでも要件を捨てて良いわけではない。関係者間で納得したうえで要件を調整するためには、客観的な情報に基づき、かつ定量的に評価した方が良い。要件を定量化し、プロジェクトに割り当てられたコストの範囲で実現可能な要件量を明確にすることが大切だ。

定量的に要件を評価する際に利用できる尺度の例を以下の表 6.22 に挙げる。尺度それぞれメリット・デメリットがあり、プロジェクトごとに適切な尺度を活用して欲しい。

6.3 監視・コントロール(RM.3)

表 6.22 要求の定量化尺度の例とメリット・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"> ・ 確立したルールに基づいた算出値のため納得性のある数値となる ・ 粒度にバラつきがない ・ 生産性指標の参考として外部書籍（ソフトウェアメトリクス調査やソフトウェア開発データ白書など）を活用できる 	<ul style="list-style-type: none"> ・ 計測できる要員を調達、育成する必要がある
機能数	<ul style="list-style-type: none"> ・ 簡易に計測できる ・ IT にうとくても理解しやすい 	<ul style="list-style-type: none"> ・ 粒度にバラつきがあり、精度が低い
画面・帳票数	<ul style="list-style-type: none"> ・ 簡易に計測できる ・ IT にうとくても理解しやすい 	<ul style="list-style-type: none"> ・ 同じ要求でも実現方式により数が異なる

たとえ、要件を定量化したとしても、定量化した要件量を使った調整のタイミングも重要であり、考慮が必要である。もし、あなたが業務部門だとして、要件定義の最終局面でシステム部門から「本プロジェクトのコストからすると〇〇FPしか実現できないため、〇〇FPまで減らしてください」と言われたとしたら、素直に「分かりました。減らします」と言えるだろうか？後出しジャンケンで、システム部門の担当者に対し、懐疑的になってしまうのではないだろうか。

これを避けるためには要件定義着手前に本開発で開発する要求量のベースラインをあらかじめ合意し、そのベースライン要件量の中でスコープに入れる要求を取捨選択すると良い。その際、プロジェクトのコストと、過去のプロジェクト（もしくは類似プロジェクト）の生産性や外部の生産性を参考値として利用しながら、ベースラインとなる要求量を説明すれば、関係者で納得性をもって合意できるはずだ。

「6.2.3 費用見積り」では、要件量をベースに概算コストまで算出することを言及している。要件量なのか概算コストなのか、業務部門がより納得する方を採用して、要件の絞り込みをすれば良い。

要件定義をしている中で、図 6.23 のように定期的にその時点における要件量を計測すると現時点での状況やその推移から要件量を予測できるようになる。

これにより、業務部門やプロジェクト外の関係者に対しても、プロジェクト状況を可視化することができる。

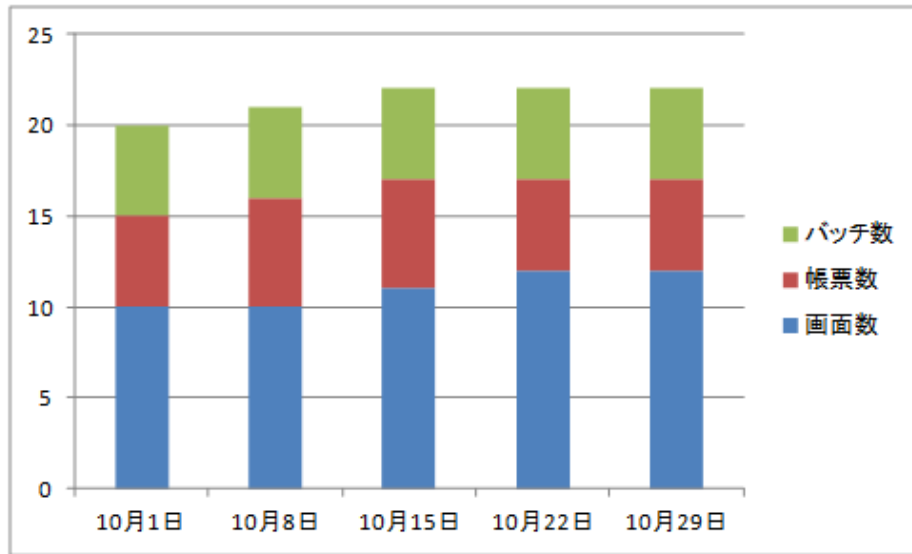


図 6.23 種類ごとの要件量推移の可視化例

一点留意しておきたいことは、要件定義工程の初期段階で機能数やFPの算出が難しいので、その時点で機能数やFPを要件量のベースラインにはできないことである。代わりに要件定義工程の初期段階では、企画工程で算出した予算額をベースラインにし、機能数やFPが算出できるタイミングで機能数やFPで要件量を算出してコントロールすることが現実的である。

事例 16 要件量を可視化する (株式会社エヌ・ティ・ティ・データ)

【取り組みの背景】

近年のシステム開発プロジェクトでは、ビジネス環境、技術環境の変化が早く、システム要件の変化も早い。しかし、スコープ（要件量）が増大していることに気づかないまま要件定義を行ってしまい、問題化するプロジェクトが散見される。

【課題】

上記のようなプロジェクトではスコープ増加についてユーザと認識の共有ができておらず、すべての要求を受け入れてしまったり、全要件量を把握しておらず、適切なスコープ調整ができていなかったりする。

【解決策】

このような背景のもと、NTT データでは上流工程の早い段階でスコープを定量化し、プロジェクトライフサイクルを通じて、スコープ増減を監視、管理するマネジメント手法をグループ内で展開している [30]。

要件量とは、システム化対象とした要件を特定の規模尺度（例として、ファンクションポイントや機能数など）で定量化したものである。本手法では機能要件を対象として、要件量を用いて定量的にスコープの増減を管理する。

(1) 要件量管理手法の概要

本手法の全体フローを図 6.24 に示す。本手法は以下に示す、2 つの特徴がある。

- 要件定義着手前に見積もった要件量による基準（ベースライン）の設定
予算に基づく開発中に受け入れ可能な要件量（許容ライン）やスコープ増減の基準を定める。ベースラインを初期にユーザと合意することで、その後のユーザ調整に役立てる。
- スコープに対する変更要求と検討状況の監視
ベースラインを設定した後、要件量の変動を管理する。これにより、要件全体をコントロール可能であり、上流工程における要件量の増大を防ぐことができる。

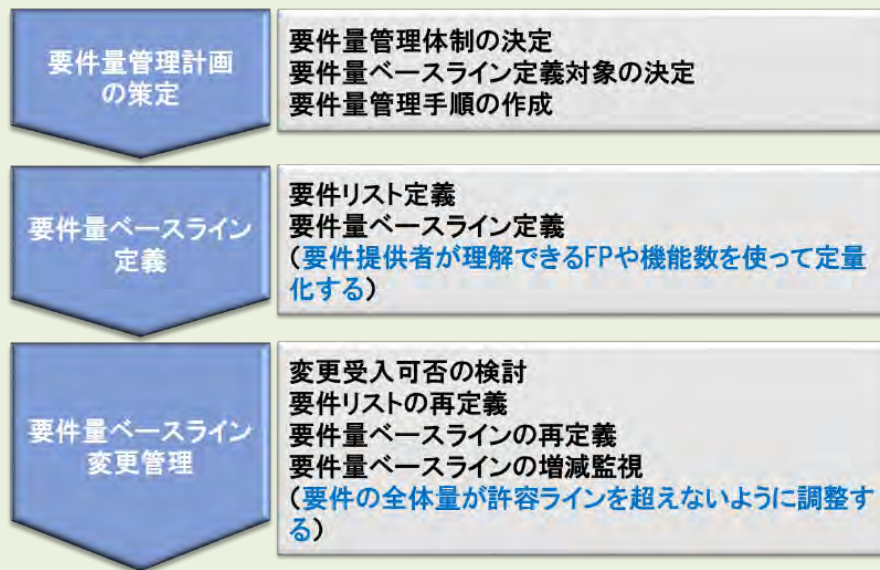


図 6.24 要件量管理手法の全体フロー

(2) 要件量管理の適用例

本手法の適用例を示す。

● 要件量定義

ユーザからのシステム化要求に基づき、機能要件の一覧を作成する。一覧に記載した機能要件を事前にユーザと合意した規模尺度を用いて定量化し、要件量を算出する。また、各機能要件の検討状況を記載する。定期的（週次、日次）にその時点での要件量を把握できるよう工夫する。

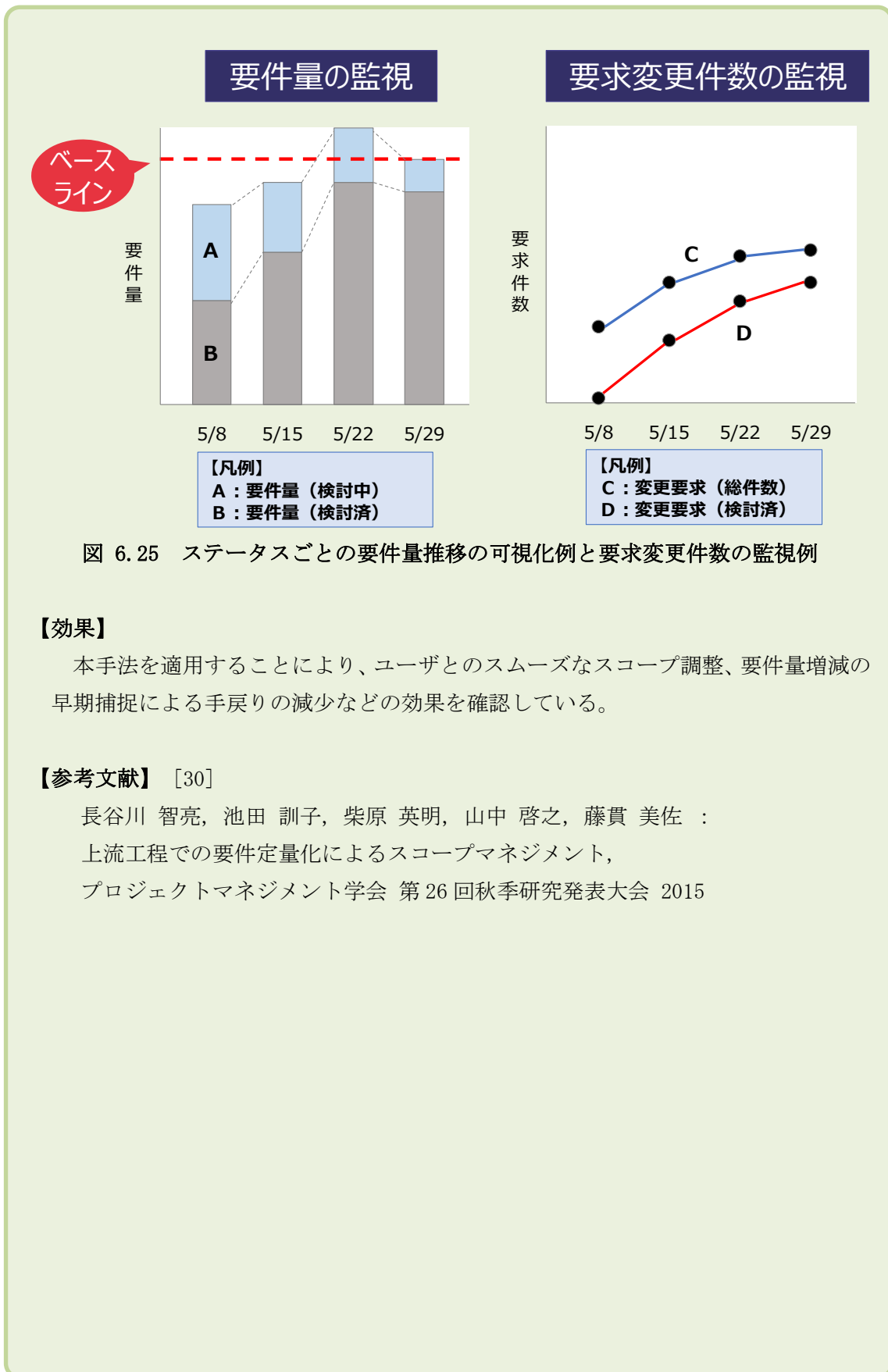
● 要件量監視

上記の機能要件一覧を用いて、要件量を監視する例を図 6.25 に示す。

図の左側は、全体の要件量の増減と要件の検討状況を可視化しており、Aは検討中、Bは検討済みの要件を表している。

一方、図の右側は要件に対する変更要求の件数を示しており、C は変更要求の総件数、D は検討済みの変更要求件数である。

このように変動を要件量と要求件数の両面で可視化することで、スコープの変動を直観的に把握することが可能となる。



事例 17 要件定義工程生産物の出来高管理による膨らむ要求の制御 (株式会社 ジャステック)

【取り組みの背景】

事例 10 の「取り組みの背景」に記載したように、要件定義工程の見積りは感覚的な見積り（人月）に成りがちである。さらに、見積りと計画、計画と実績（残予測を含む）がカウンタブルな指標値として連動していないなどにより、要求の量を単位とした出来高予実管理ができにくいため、膨らむ要求の制御が後手に回るケースが見受けられる。

【課題】

弊社での販売管理業務システムの再構築開発において、こうした問題を解決すべく、以下の 2 つの課題を設定した。

- 当該工程での見積りと出来高予実管理に使用する指標値の連動課題
- 当該工程における膨らむ要求を制御する出来高予実管理方法の課題

なお、当該工程の見積りに関する課題に関しては、事例 10 を参照願う。

【解決策】

販売業務管理システムの再構築開発では、ソフトウェア開発での独自の生産管理方式（ACTUM）に基づき、見積り（計画）、実績、残予測の 3 つの指標値が連動した出来高予実管理を実践している。さらに、妥当値と成行値を使用した出来高の予実差異分析による膨らむ要求の制御を行い問題の解決を図った事例を紹介する。

(1) 要件定義工程の妥当値と成行値による出来高予実管理

進捗度（出来高把握）は原価比例法や EVM (Earned Value Management) を使用した方法が一般的である。当社では見積りと出来高（実績+残予測）を連動させ、精度を高めるために、生産物量（生産性含む）の妥当値*1 と成行値*2 の考え方を取り入れて実践している。なお、EVM を使用する場合は、PV (Planned Value) と EV (Earned Value) は妥当値を、AC (Actual Cost) は成行値を適用する。

- *1: 妥当量（基準生産物量、生産物量環境変数、分担率、許容要求変更量‘率’）と妥当生産性（基準生産性、生産性環境変数）から構成している。お客様への見積りは妥当値を適用する。妥当生産性の基準生産性は最速な生産性を指す。

6.3 監視・コントロール(RM.3)

*2: 成行量 (妥当量に超過要求変更量‘率’と自責変更量‘率’を加算)と成行生産性 (妥当生産性に自責生産性‘適/不適’)を加算)から構成している。

なお、弊社の標準では「仕様変更量‘率’ = 許容要求変更量‘率’ + 超過要求変更量‘率」として、妥当量に仕様変更量‘率’を組み入れている。販売管理業務システムの再構築開発では、要件定義工程における膨らむ要求を制御するために、許容要求変更量‘率’ (妥当量)と超過要求変更量‘率’ (成行量)とに分けて設定している。

図 6.26 は生産物量 (妥当量、成行量)と生産性 (妥当生産性、成行生産性)の項目および当該時点 t= ‘u’ 「実績、残予測、実績+残予測」での生産物量の出来高管理表を示す。

図 6.26 要件定義工程での生産物ごとの妥当値と成行値

(2) 出来高の予実差異分析による膨らむ要求の制御事例御

妥当量の許容要求変更量は、要件定義工程 (生産物量) の完了時に許容できる要求の変更量であり、成行量の超過要求変更量は許容要求変更量を超過した変更量である。

図 6.26 にある (当該時点 t= ‘u’ 「実績、残予測、実績+残予測」) は生産物量を対象にした事例である (出来高をコストとする場合は生産物量×生産性「円/量」に

になる)。

図 6.27 は当該時点 t= 'u' での出来高の予実差異分析 (妥当値、成行値) から超過要求変更量を検知し、代替案を含む改善提案と調整により過剰な要求量を制御した事例である。

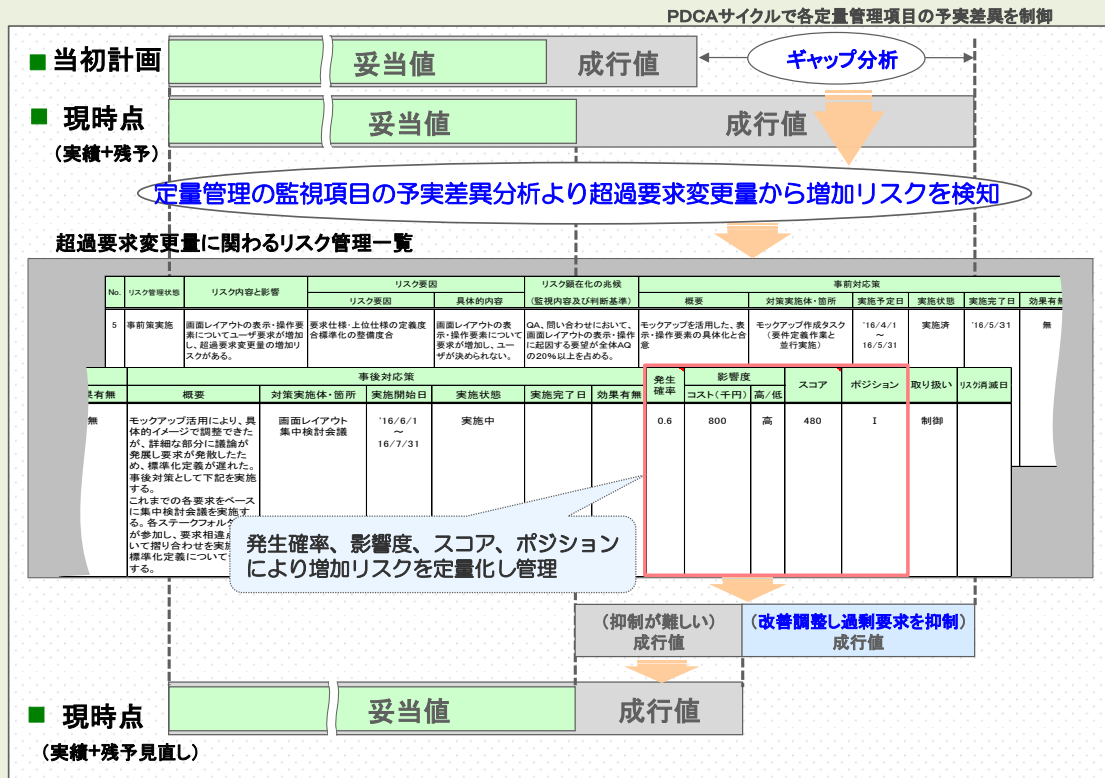


図 6.27 出来高予実分析による膨らむ要求の制御事例

【効果】

一般的に要件定義工程に定量的プロジェクト管理を適用している事例はまれであるが、当該事例は弊社の販売管理業務システムの再構築開発において、要件定義工程に定量的プロジェクト管理を適用した事例を紹介している。

当該再構築開発では要件定義工程進行中に許容要求変更量と超過要求変更量とに分けて膨らむ要求を制御したことにより成行量の増加リスクを抑えることができた。

また、EVMは現時点「t= 'u」の計画「PV(妥当値)」、実績「EV(妥当値)とAC(成行値)」は存在するが、一般的に現時点の残予測 (妥当値、成行値) があいまいになっている。

弊社では、常に現時点の残予測見積りを行うことで、要件定義工程の完了時点を見越した膨らむ要求の制御を可能としたことの効果が大きい。

6.3 監視・コントロール(RM.3)

(2) 実現する要求は関係者が納得できる指標をもとに選択する

【解決したい問題】

- 現場からは、「あれもこれも必要で、ないと要求は実現できない」と言う

実現する要求は、プロジェクトの目標を達成するために、プロジェクトオーナー側でまとめていく必要がある。しかし、プロジェクトオーナー側だけでは、その要求をまとめるのは難しく、一般的には利用者となるユーザ側の現場に対してもヒアリング等を行い、要求事項をまとめていくことになる。ユーザ側の現場からは、従来の業務を変えたくないという思いが強くあり、そのことを前提としてさまざまな要求が発生する。そのため、要求事項が膨らみやすい。課題としては以下のようなことが発生する。

- 要求が絞り込めない
- 各業務部門が要求事項を譲らない
- すべての要求事項を実現するには、予算と期間が足りない
- どの要件が優先なのか絞り込めない
- 要求事項を不公平感なく、現場が納得するように切り分けできない

上記のような課題に対して、対策としてそれぞれの要求事項を以下の観点で評価する。

勘どころ① 要求事項を客観的な指標で評価し、実現する要求を選択する

最初に、プロジェクトの目標を実現するためにシステム化が本当に必要な要求なのかをプロジェクトの目標との適合度で評価する。数値評価が可能であれば、数値評価することを推奨するが、判断が難しいのであれば、「高」「中」「低」などの評価でも良い。本当に必要な要求事項なのかをはっきりとさせると良い。

また、あわせて予算を決定すべきである。期待するビジネス上の効果を想定し、それに見合う予算額を算出する。実現するシステムの予算を前提条件として設定することで、要求の肥大化の抑制にもつながる。

これは、ユーザ側で判断すべきものである。

次に、現場業務への影響度を測定する。その要求事項がシステム化されないことによる影響度(生産性、品質など)を評価する。影響については、メリットとデメリットの双方を評価すると良い。できれば、これも数値評価が望ましいが、難しいのであれば、「高」「中」「低」などの評価でも良い。この評価については、ユーザ側の現場が評価するが、ユーザ

6.3 監視・コントロール(RM.3)

側のプロジェクトオーナーと評価を精査する必要がある。

また、パッケージ開発の場合、「アドオン判定基準」などを合意しておくことも有効である。以下のようなルール・基準を決めておくが良い。

- 見栄えや 1 件 1 分相当程度に操作性をよくするためだけのアドオンは作らない
- 現行実現できていることでも、年間 2 人日程度の代替策で解決する程度の要求は採用しない
- 年間 n 件以下の特殊レアケースで、ハンド事務で救済できるものはシステム化しない
- 業務フローの見直しで解決することはシステム化しない。その評価のために、現状にこだわらず代替事務フローは必ず検討する

さらに、要求事項の要件量を測定する。費用でなくでも、物量など具体的な数値で測定すると良い。この要件量の測定は、実際に開発する際の見積りにもつながってくるので、ベンダ側が行った方が良い。

要求事項を上記 3 つの観点で評価を行ったら、関係者とともいどの要求を選択するのか検討を行う。注意すべき点は、プロジェクトの目標を見失わないこと、常にシステム全体を意識し、偏ったシステムにならないようにすること、などである。また、要求事項の取捨の経緯や理由については必ず記録を残しておく。経緯や理由の記録がないと、蒸し返されることが発生し、プロジェクトの進行の妨げとなるリスクが発生する。さらには、要求事項を見直す際には検討材料にもなる。

表 6.23 要求事項の評価の例

要求事項	適合度	影響度	要求量
要求事項A	高	高	10ユースケース
要求事項B	中	低	13ユースケース

6.3 監視・コントロール(RM.3)

(3) 決定済の事項を大きく変更することが必要になったら、プロジェクト全体での意思決定により採否を決定する

【解決したい問題】

- 要件定義がある程度進んだ段階でも平気で大幅な修正を伴う変更を要求してくる
 - ・ 要求の変更に歯止めが利かない
 - ・ 変更対応に手間がかかる
 - ・ 変更対応が漏れて後工程でリカバリが必要になる

ここでは、要件定義の途中でも、それまでの検討で決定された事項を変更する必要がある際、特に、その変更を受け入れることによりこれまでに実施してきた作業の結果に大きく影響することが予想されるような場合における対応の勘どころを記載する。

要件定義は、一般に試行錯誤の連続であり、以前にこれが最善と思って決定した内容でも、検討が進み、内容がより具体化していくにつれて修正が必要と思い、相当議論が進んだ段階でも、要件変更が必要と主張してくることが、ある程度の頻度で発生する。要求実現の優先度や相反する要求の選択に揺らぎがある場合にこのようなことが発生する。要件定義の現場においてそのようなことが発生した場合には、これらは以下のような問題として顕在化してくる。

- 会議のたびに違うことを提案してくる
- 現場部門が一度決定したことを次の打合せで平気で覆すことを繰り返す
- 要求の変更が止まらない
- その変更が受け入れられないとシステム導入の効果がないと言って聞かない
- 決定済の事項（作成済の成果物）に対する多数の修正への対応に時間がかかる

各部門が抱えるビジネス課題をもとに要求を漏れなく集める要件定義の初期段階ならまだしも、ある程度要件定義が進んでまとめに入ったような段階でこのようなことを繰り返すと、作業に余分な時間を要し、全体の進捗に影響を与えることが多々発生する。そのようなことにならないように、要件定義の現場で上記のような問題が発生してきたときには、以下のような対策を取ることが必要になる。

勘どころ① 変更を当事者間で安易に受け入れさせない

決定したことを変更する場合、特に、変更を受け入れた場合に、それまでに作成したものの修正が大規模になり、以降のスケジュールに大きな影響が発生することが予想される、スコープ変更のような場合には、実施を担当者間だけで決定せず、開発工程以降にお

6.3 監視・コントロール(RM.3)

ける変更管理委員会（CCB：Change control Board）のようなプロジェクト全体での意思決定プロセスを用意して、そのプロセスで採否を決定する。

なお、大きな影響があるかどうかを現場の判断に任せた場合、判断ミスなどにより、実際には多大な影響のある変更であったものが誤って実施されるリスクがあるため、上記のようなプロセスで採否を決定する必要がある変更かどうかの規準をプロジェクト内で設定して、要件定義の開始時に、関係者全員に周知しておくことが望ましい。

勘どころ② 変更を実施するかどうかは、変更の影響と得られる効果から判断する

変更を受け入れるかどうかの規準は、変更を受け入れることによって得られる効果（以降の開発への好影響や完成したシステムの適用効果の向上など）が、変更を実施することによる要件定義やそれ以降のシステム開発への影響を上回るかどうかを、プロジェクトのQCDに責任を持ち、方針決定を行えるレベルの参加者により採否を決定する。

本項では、要件定義プロセスにおける変更管理について説明したが、変更管理は本来、要件定義にとどまらず、開発プロセスにおいても、開発後の運用プロセスにおいても活動を継続するべきものである。以降のどのプロセスにおいても、要件や機能に変更要求が発生したら、発生した要求に適切に対応し、変更を受け入れた項目については要件定義の成果物にまで遡って修正をすべきものであることを付記する。

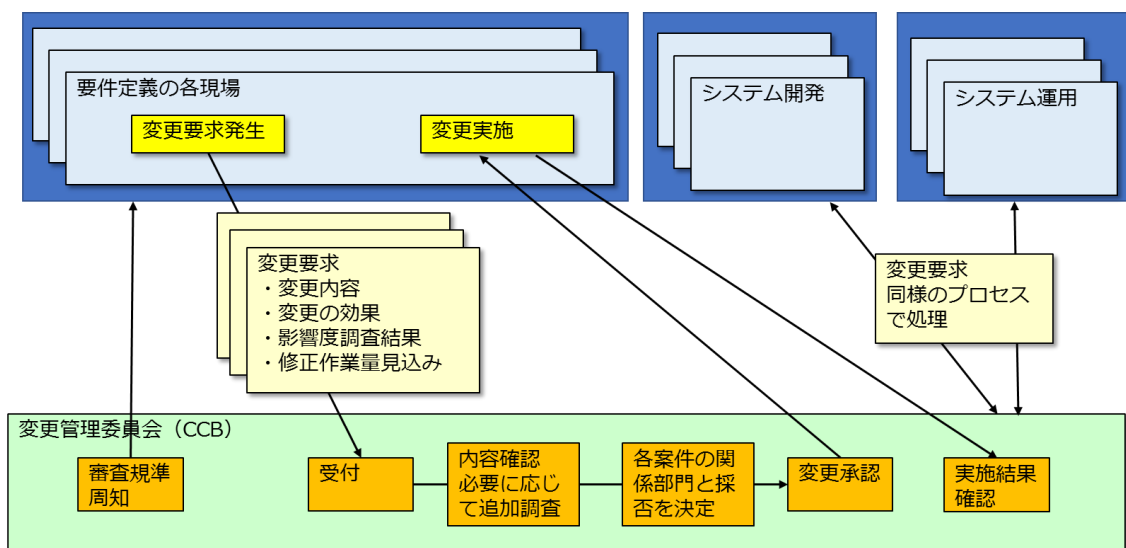


図 6.28 変更管理委員会の活動イメージ

6.3 監視・コントロール(RM.3)

勘どころ③ 変更する場合には、抜け・漏れが発生しないように確実に実施する

変更するとなった場合には、それまでに作成した成果物を抜け・漏れなくすべて正確に修正することが必要になる。修正に抜け・漏れがあると、それは高い確率で後工程での障害の混入につながる。

修正を正確に実施するためには、「6.3.6. トレーサビリティ」で説明するトレーサビリティ管理の手法などを用いて、ドキュメント間の連携を維持できるような作業の標準化を採用しておくことが望ましい。

6.3.2 スケジュールコントロール (RM3.2)

要件定義のスケジュールに従って、決められたタスク（作業）を実行し、スケジュールとおりにすべてのタスクが完了するようにコントロールを行う。しかし、スケジュールとおりに進めるには、要件定義特有の難しい面がある。その要因の一つは、タスクの多くが、調査、分析、解決策の立案、ステークホルダ間の調整、組織的なオーソライズを要するものが多いこと、また関係者が多いためにコントロールが難しいことである。もう一つの要因は、作業ベースに落ちていないタスクが多く、出来高ベースで進捗具合を把握できないことである。そのため、計画とおりに要件定義を終えられず、次工程に一部作業を回すことや、条件付きの完了とするケースも多い。このように難易度が高いスケジュールコントロールであるが、次のような対応は考えておくべき点である。

(1) 細かいチェックポイントの設定と、事前に遅延対策を関係者に周知しておく

【解決したい問題】

- 要件定義は進捗遅れの程度が分かりにくく、かつプロジェクト関係者が多いため遅延対策が難しい

要件定義の作業は、調査、検討、複数人によるレビュー、調整、承認を伴うものが多く、基本設計以降のように進捗基準に基づく「出来高」（10人日の作業で50%の進捗度であれば5人日分の作業が完了）による進捗判断が難しい。そのため、要件定義全体の遅れ具合の把握が難しく、大きな進捗遅れに気づかないリスクがある。

また、遅延の原因には、経営層も含めた上位のプロジェクト関係者に起因する遅れも多く、有効な遅延対策を実行することが難しい面がある。

勘どころ① 細かいチェックポイントを設けて、進捗遅れを早期発見、早期対処する

タスクの担当者からは、この先に要する工数と期限内に完了できるか否か、できないとすれば完了見込み日はいつになるのか、そのための条件は何かなどを、タスクごとに報告してもらうことになるが、担当者の主観的な判断に依存する部分も多い。また進捗遅れの要因には、作業工数不足以外にも、担当者が作業に入れない状況とか、関係者間の調整に想定以上の期間を要しているとか、上位者の承認待ちなどによる遅延が多くみられる。そのため、要件定義全体として、遅れ具合がどれ位かの判断が難しく、要件定義全体の進捗具合や、要件定義の完了見通しが見えないまま進むことも珍しくない。

6.3 監視・コントロール(RM.3)

この状態を完全に防止することは難しいが、きめ細かいチェックポイント（全体としてのマイルストーン以外に、まとまったタスク群ごとに小さいマイルストーンを設定する）を置いて、それを関係者が認識している状態を作っておく。もしマイルストーンを守れない時には、次のマイルストーンを守るにはどうすれば良いのかを上位者も交えて協議し、対応策を決める。（関係者を絞って実質討議ができる場を設けておくことが望ましい。）

勘どころ② キーマンや上位マネジメントの関与不足による遅れ対策を、あらかじめ考えておき、実施する

要件定義の遅延理由には作業工数不足以外の要素も多い。特にユーザ側のキーマンの関与不足や、上位マネジメントの関与不足の影響が大きい。したがって、計画段階で有効な遅延防止策をしっかり立てておかなければならない。例えば、ユーザ側担当者の業務優先度の調整責任の明確化や、部門を横断する課題検討分科会や、プロジェクトオーナーを頂点とした意思決定承認機構を整備しておくことなどである。また、要件や課題に関して、決められる人材の関与が進捗に大きく影響するので、そうしたメンバの会議体への参画を確保しておかなければならない。なお、ユーザ側のキーマンの関与不足による遅延や、ステークホルダ間の調整などの問題への対処として、事例18、事例19も参考にされたい。こうした対応策の準備と実施にあたっては、プロジェクトオーナーの役割が非常に重要で、オーナーとの認識合わせや事前の理解の共有が欠かせない。

勘どころ③ 遅延が大きくなった場合の対応策を、あらかじめ決めておきタイミングよく実施する

リカバリ対策の一つとして、要件定義の進捗遅れの度合いによっては、ビジネス要求や業務要件を固めることを優先して、システム要件関係を先送りさせるのも一案である。システム要件に関する成果物の粒度を見直して、詳細検討を次工程にまわすなどの処置により、システム側の担当者の工数を、ビジネス要求側に振り向けることも検討する。また、課題の大きな遅れに関しては、決定権を持つ上位の方に出席してもらって臨時の拡大課題検討会議（仮称）を開いて、一気に課題解決を図ることも有効な対応である。もちろんこれらの対策実施には、オーナーをはじめとする上位層の承認を要するが、遅延問題が大きくなってから対応策を検討してはタイミングを逃すリスクもあるので、あらかじめ対応策を決めておき、遅延が大きくなった場合は、速やかにこれらの対応策を実施する。（「6.2.2 スケジュール作成」でも述べたが、遅延対策の迅速な実施にあたっては、あらかじめプロジェクト関係者間で遅延対策を共有しておくことをお勧めしたい。）

勘どころ④ 計画時のレビューも代行できる仕組みを用意して実施する

6.3 監視・コントロール(RM.3)

また、レビュー・承認遅れに対しては、レビュー・承認者を代行できる体制を整えておき、当初のレビューの関与度が計画とおりでない場合や、レビューの諸事情によりレビュー・承認が滞る事態が発生した場合は、レビュー・承認者を切り替えて進められるように、あらかじめ決めておくことも有効である。レビュー・承認遅れが顕著になった場合は、躊躇なくオーナーなどを交えて協議の上、代行者によるレビュー・承認に切り替えて、スケジュールを守るようにする。

事例 18 To-Be の業務改善案の作成が遅れる (1)

【取り組みの背景と課題】

中堅素材メーカーである Y 社では、要件定義の中心メンバ(各部門を代表するキーマン)が現業を抱えているため専任化が難しく、業務要件の提示、確定に遅れが出ることが予想された。

【対応策】

- (1) キーマンの負荷を下げるために、部門の複数の現場担当者にプロジェクトメンバになって貰ってチームで対応することとした。複数人で作業を分担するため、業務フローや業務改善策の粒度などにバラつきが出るのが想定されたので、業務プロセスを可視化して分析し、管理するための BPM (Business Process Management : ビジネスプロセス管理) ツールを導入して、標準化の推進と業務フロー作成作業の効率化を図った。
- (2) 業務改善の観点(後述)を決めて、メンバ全員に教育を行い、それに基づく業務課題の洗い出しと改善策の立案を行った。
- (3) 業務改善策の検討にも、BPM ツールを活用し、解決策を BPM ツールに登録してもらい、改善策の議論がしやすい環境を整えた。

具体的には

(1) BPM ツールの導入と標準化の推進

BPM ツールとして、日本で販売している“iGrafx BPR+”を導入して、BPM ツールによる標準化を推進した。まず標準化ルールを作成し、メンバ全員に教育を行い、最初の成果物をチェックして、ルールの徹底と成果物の品質向上を図った。標準化ルールと教育は次のとおりである。

<標準化ルール項目>

- ・ 標準書式としての業務フローテンプレートの作成
標準化ルールとして、命名規則、使用する図形の統一、作業の分割単位(入力、確認、承認、出力、送付、判断、保管など)の統一、条件分岐ルール、例外処理の取り扱い、部門間会議の取り扱い、用語の統一など
- ・ サンプル業務フロー図で粒度指針を提示

<教育>

- ・ メンバには BPM ツールを使って、半日 2 回のツールトレーニングと標準化教育を実施。なお、標準化ルールの作成から教育までは、経験豊富な BPM ツ

ールベンダのコンサルタントに業務委託した。

BPM ツールを使った業務フロー図の例 (要約版) と業務処理定義の画面例を図 6.29、
図 6.30 に示す。

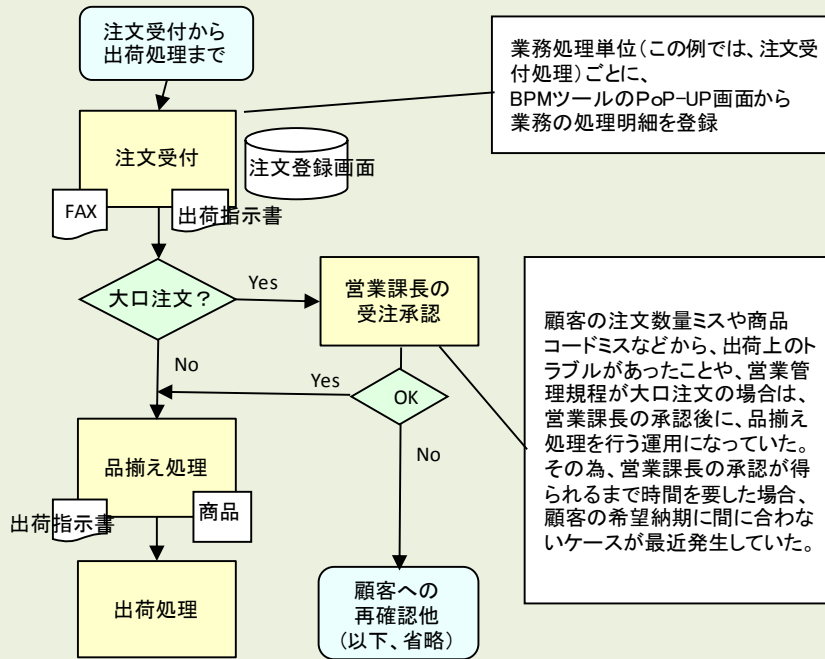
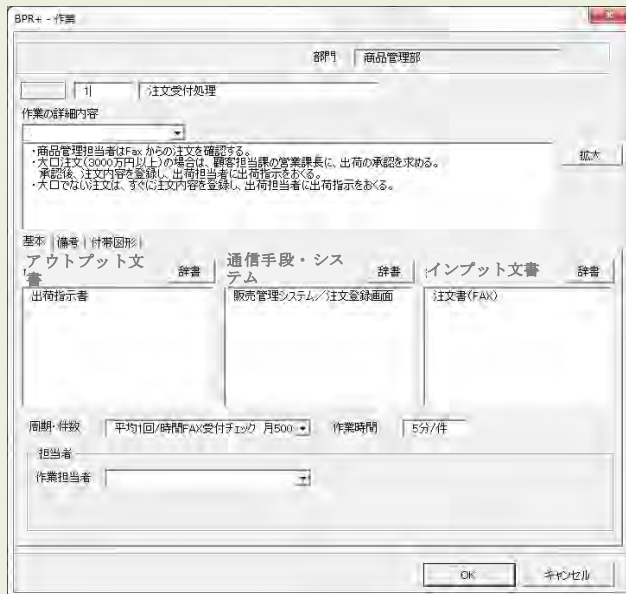


図 6.29 As-Is の注文受付から出荷までの要約業務フロー



- ① 業務処理単位に業務処理内容を記載。通常の例外処理なども詳細に記載。
- ② 入力情報名、出力情報名、システム処理名を記載すると業務フロー図に図で表示される。(図1参照)
- ③ 担当組織や発生件数、平均処理時間などを記載
- ④ 一覧表形式でEXCELにダウンロードでき、全体のチェックが出来る。

図 6.30 As-Is の注文受付の業務処理定義画面

6.3 監視・コントロール(RM.3)

(2) 業務改善の観点の教育

To-Be の業務フローや業務処理を検討するにあたっては、

- ・ 作業時間の短縮や重複作業などの排除、ならびに効率化
- ・ ミスの削減やサービス品質の向上
- ・ サービスのスピードアップ
- ・ トラブルの防止
- ・ 将来の業務増加に備えた対応力の拡大
- ・ 制度変更や関連法規への対応
- ・ 内部統制面への対応

などの業務改善の観点と、その実例を教育した上で、自分が担当する業務プロセスの問題点や課題を洗い出してもらった。

問題点や課題は、直接 BPM ツールに登録して全員で共有した。

(3) 業務改善の議論に BPM ツールを活用

関連部門の責任者や担当者も出席の上、BPM ツールで作成した As-Is と To-Be の業務フロー図と各プロセスの問題と改善策をスクリーンに映して討議を行い、討議結果を、その場で BPM ツール上の「To-Be 業務フロー」や、「業務課題と改善案」に登録、または修正し、その場で確認して検討会議の効率化を図った。

その一例を図 6.31、図 6.32 に示す。

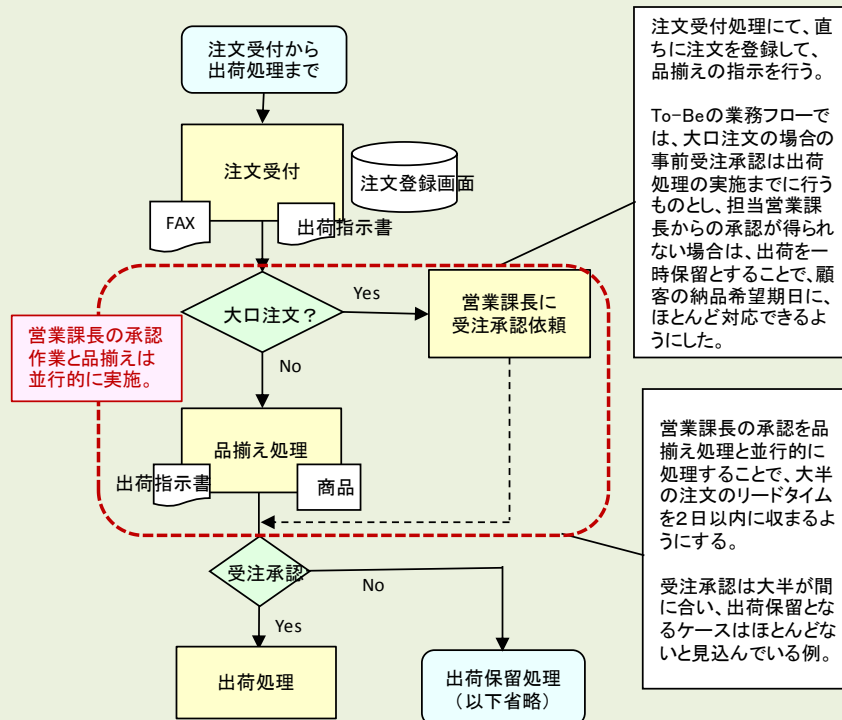


図 6.31 To-Be の注文受付から出荷までの要約業務フロー

作業: 大口注文の受付作業 部門: 商品管理部

業務上の問題: 検討中 注文受付処理

大口注文の場合、担当営業課長の承認後、商品品揃え、出荷準備に入りますが、営業課長が不在などで承認が遅れることが、しばしば発生しており、顧客の納期希望に添えないケースが最近増加している。(月に20件ほど、全体の5%を占める)

起案内容 | 整理 | 備考 |

あるべき姿: 大口注文の出荷リードタイムを2日以内とする。

想定される要因: 担当営業課長の承認が、不在などにより遅れるため。
営業管理規定の制定時に比べて、大口注文が増加しており、営業管理規定が現状と合っていないため。

改善案: A案、大口注文の判断基準を引き上げ、現行3000万円以上を5000万円以上にして、営業課長承認案件の発生件数を現在の3分の1にする。(提案 営業部 山口次郎)
B案、大口注文の場合も、事後承認方式を取り入れて、仮出荷指示を設け、受付日から2営業日以内に、出荷の差し止めがなければ、出荷を可とする。(提案 営業管理部 〇〇三郎)
C案、営業部にWEBのワークフローを導入して、電話確認をなくし、担当営業課長が外出時でも出張時でも承認できるようにする。(提案 情報システム部 △△四郎)
D案 A案とB案を組み合わせる。(商品管理部 〇〇太郎)

部門・影響範囲: 〇〇事業部 重要度・緊急度: 高

提案者: 商品管理部 〇〇太郎 提案日: 2017/11/12

改善案のメンバーの提案内容

- ・A案 大口注文の判断基準を引き上げ、現行3000万円以上を、5000万円以上にして、営業課長承認案件の発生件数を現在の3分の1にする。(提案 営業部 山口次郎)
- ・B案 大口注文の場合も、事後承認方式を取り入れて、仮出荷指示を設け、受付日から2営業日以内に、出荷の差し止めがなければ、出荷を可とする。(提案 営業管理部 〇〇三郎)
- ・C案 営業部にWEBのワークフローを導入して、電話確認をなくし、担当営業課長が外出時でも出張時でも承認できるようにする。(提案 情報システム部 △△四郎)
- ・D案 A案とB案を組み合わせる。(商品管理部 〇〇太郎)

図 6.32 To-Be の注文受付から出荷までの改善案 検討画面

【効果】

- ① BPM ツールを使うことで、効率的に As-Is の業務フローの作成から、各業務プロセスの問題・課題をまとめ、To-Be の業務フローを完成させることができた。
- ② BPM ツールを使うことの具体的なメリットは次のとおりだった。
 - Excel で作成していたのと比較すると、5 倍から 10 倍の生産性があった。
 - 標準化の徹底がしやすく、複数のメンバが作成したことによるバラつきも効率良く修正できた。
 - As-Is の業務プロセスの問題点や課題を直接、業務フロー図に登録することができ、業務フロー全体を見ながら改善策の検討が可能であった。
 - 問題・課題の一覧が Excel にダウンロードできたので、進捗管理が容易だった。
 - 成果物が BPM ツール内で一元管理されるので、成果物管理も容易だった。
 - その結果として、To-Be の業務フロー、業務詳細定義が効率的に作成できた。BPM ツールなしでは、短期間に As-Is 業務フロー/業務処理定義の作成から、To-Be の業務フロー/業務処理定義を固めることは難しかったと思われる。

6.3 監視・コントロール(RM.3)

- ③ BPM ツールの導入と、現場担当者に作業分担したことで、キーマンは成果物のレビューと部門間調整に集中できたため、予定納期内に To-Be の業務フローや業務処理定義書の承認が得られた。
- ④ なお、副産物的であるが、現場担当者(メンバ)が BPM ツールを使って、自ら業務フロー、業務処理改善案を作成してみたことで、普段気づかなかった問題や新しい気づきがあり、また他部門の業務理解が深くなった担当者もいた。

事例 19 To-Be の業務改善案の作成が遅れる (2)

【取り組みの背景】

中堅サービス業の K 社では、前のプロジェクトでキーマンによる業務改善案の立案がうまく行かず、要件定義の遅れが顕著となったためプロジェクトを中断したことがあった。

【課題】

業務改革を伴うシステム開発プロジェクトの要件定義工程は、業務部門の代表メンバ（以降、「キーマン」と称する）の業務分析から業務改善案の立案までの能力に大きく依存しているが、キーマンの業務改善立案能力の確保と、キーマンの改善案立案を支える支援策の強化が課題となっていた。

【対応策】

経営者は以前の結果から上記の課題を認識しており、自社の社員だけでは業務改革の改善案を取り纏めるのは難しいと判断し、外部コンサルタントによる自社の社員の教育と業務改善立案作業の支援をお願いするのが良いと判断した。その結果、取った対応策は次のとおりであった。

(1) キーマンの教育実施（業務改善の進め方）

- ① 外部コンサルタントが選んだ教本（業務改善ハンドブック）を全員に与えて、要件定義開始後、速やかにその本をベースにした業務改善の進め方を教育した。
- ② 教育対象は、業務部門側から参加するプロジェクトメンバ全員とした。その理由はキーマンだけではなく、それを支援する現業の中核メンバ（プロジェクトメンバ）の理解と協力が不可欠だからである。
- ③ 教育で工夫した点は、各部門のキーマンに、テキスト本の章単位（多い章は分割）に要約した説明資料を作ってもらって、その章の講師役になってもらい、より内容の理解を図るようにした。
- ④ 外部コンサルタントは、講師役のキーマンの説明を補う形で、業務改善の進め方の大事なポイントをフォローした。

(2) 外部コンサルタントによる成果物のレビュー

キーマンの作成する成果物に関しては、業務改善の専門家である外部コンサルタントが、改善案の進め方や改善案の論理性、実現可能性などに関してレビューした。

6.3 監視・コントロール(RM.3)

(改善内容そのものよりも、結論に至るプロセスと論理的な整合性を主に見ていただいた)

(3) 部門間をまたがる To-Be 業務改善案の合意に、外部コンサルタントの中立性を活用

- ① キーマン作成の業務改善案のうち、部門間をまたがる改善案については、他のキーマンも含めて、プロジェクトチーム内の「業務改善検討会議」で、To-Be 業務改善案をまとめたが、意見の相違や自部門優先がどうしても出て、結論が出ないことが多かった。
- ② そのため、外部コンサルタントにファシリテータ役になってもらい、部門間で合意できない問題の論点を整理してもらって、合意に至るための筋道をつけていただいた。

(4) 部門責任者の承認にも外部コンサルタントの中立性を活用

- ① プロジェクトチームで作成した業務改善案について、キーマンは所属部門責任者に報告を都度行った。また部門間調整事項については、部門責任者をメンバとした「業務改革推進会議」(月 1 回)に報告して早めの合意を図るようにした。
- ② なお、「業務改革推進会議」には、外部コンサルタントにも出席いただき、会議運営のアドバイス、支援をもらった。
- ③ 最終案の承認を求める会議では、外部コンサルタントの方には、ファシリテータ役を務めてもらい、部門間の利害調整が円滑に進むように支援いただいた。また、最終案承認には定期会議のほかに臨時の会議を開いて、部門間調整済みの業務改善案をまとめた。

【効果】

業務改善案の承認が遅れることによる要件定義の遅れがこれまであったが、今回は予定期間内に終えることができた。

その要因として

- ① 要件定義の開始時点で、キーマンとそれを支援する現場の中核メンバに対して、業務改善の進め方の教育を実施したこと。その結果、業務改善能力が向上したこと。

特に有効だったのは、「業務の見える化手順」、「業務フローの書き方」、「業務改善の視点とやり方」であった。また、現場の中核メンバのモチベーションが非常に高まったことも大きかった。

- ② キーマンも外部コンサルタントのレビューがあるため、納期を守る意識が高くなり、業務の繁忙期や問題発生時にも期限をほとんど守って、業務分析から To-Be 業務改善案の原案が作成されたこと。
 - ③ 業務改善に絡む部門間の調整は、社内だけではなかなか合意に至るのに時間を要するが、今回は外部コンサルタントの専門性と中立性が早期合意に有効であったこと。
 - ④ 部門責任者の改善案承認においても、③の外部コンサルタントの活用が有効であったこと。
- などがあげられる。

6.3.3 品質(RM.3.3)

要件定義においてどのようにして品質を確保する計画を立案するかについては、「6.2.5 品質計画」で述べた。ここでは、プロジェクトの実行段階で計画したことが適正に実施されているかを監視し、計画したとおりの状態を維持するかについて述べる。計画の達成が実行段階での監視と問題の早期摘出、問題がある場合の是正で成り立つのは、要件定義においても同じである。

(1) 抜け漏れを抑えた要件定義成果物の作成とプロセスを監視する

【解決したい問題】

■ 要件定義工程での実用的かつ有効な品質測定指標がない

品質の高い要件定義とは、前工程で決めた開発目的や効果目標、予算、納期、実現方法をベースに、効果が最大となるようユーザと要件を決めることである。そのためにはユーザの要件を抜け漏れなく聴いて整理する必要がある。要件に抜け漏れがないと、以降の工程での手戻りが抑えられたり、リリース後の障害が低減されたりする。

したがって要件定義工程での品質を監視・コントロールしたいが、監視・コントロールに実用的かつ有効な測定指標がない。例えば基本設計工程以降の代表的な測定指標は、レビューページ数やレビュー指摘件数であったり、テスト工程以降はテスト件数や欠陥件数であったりし、その測定指標を使って標準値とのバラつきをチェック・評価して、必要な対応策を実行する。しかし、そのような測定指標を要件定義工程に適用できにくいのが実情である。

そこで、要件定義工程での品質のチェックは、次の2つを実施することを推奨する。

勘どころ① 要件の抜け漏れをなくすために作ると決めた成果物を作れているか、チェックする

要求一覧、業務機能構成表、ビジネスプロセス関連図、業務フロー、画面一覧など、その開発プロジェクトの要件定義工程で必要とする成果物（「6.2.1 要件定義プロセス計画」の表 6.7 参照）が漏れなく作れているかをチェックすることで、要件の抜け漏れを抑えていく。それらの成果物はシステム側組織で規定・標準化された成果物のフォーマットや、準備している成果物サンプルを使用するのが望ましい。なぜなら、要件の抜け漏れは長年の課題であり、それら標準の成果物フォーマットやサンプルに抜け漏れを抑制するノウハウが反映されているからである。

6.3 監視・コントロール(RM.3)

そして、要件一覧、画面一覧など一覧系の成果物をユーザと活用して、抜け漏れを防いでいくように、第4章、第5章の成果物作成プロセスを遵守する。また、「6.2.5 品質計画」にあるとおり、成果物作成要綱の合意や作成初期の重点レビューを通して、成果物作成品質のバラつきを抑えていく。

これらの作ると決めた成果物の完成度合いを、チェックシートを使ってチェックする。そのチェックシートには、作ると決めた成果物の一覧と、誰（チームリーダー、PM、ユーザ担当、ユーザ責任者）がレビューすべきかを記載しており、実際にレビューをしたらチェック・記録していく。もし成果物の作成ができてなかったり完成度が低かったりしたら、原因および改善方法を特定していく。

勘どころ② 要件の抜け漏れをなくすために作ると決めた要件定義プロセスをやれているか、チェックする

勘どころ①の成果物チェックに加えて、その成果物を作るプロセスがきっちりやれているかをチェックする。そのプロセスとは主に、現行業務とシステムの可視化と理解から始まり、問題・課題の抽出、経営や業務に貢献する要求の獲得、効果的な実現手段の抽出と判断、ビジネスプロセスの見直し、要求の絞り込みの交渉・合意形成（以上、詳しくは「4.1 ビジネス要求の獲得」「4.2 ビジネス要求の分析」を参照）、一覧系成果物の活用、インターフェースの確認、非機能要求の定義、移行や運用の要件の定義（以上、詳しくは「5.1 システム化要求の仕様化」を参照）である。こちらもシステム側組織で規定・標準化された要件定義プロセスを適用するのが望ましい。

そして、それらのプロセスを推進し要件の抜け漏れをなくすには、ユーザもしくはシステム開発者に、対象業務の有識者と、システム再構築プロジェクトであれば既存システム仕様の有識者が必要となる。その有識者を重要な推進者かつレビューにして現場現物を見て妥当性確認を進め、要件定義プロセスを推進・レビューできたかチェック・記録していくべきである。

採用する要件定義プロセス自体に、要件の抜け漏れをなくすチェックリストがあれば、それを活用していく。以下代表的なチェックの例である。

- コンセプトと要件の関連は明確か？
- 要件と機能の関連は明確か？
- 業務フロー上、だれがいつ何をするか明確か？
- 人命に関わる業務や社外と金銭が絡む業務など、業務リスクに対する要件は明確か？
- 新旧機能比較表は完成しているか？
- 開発範囲が明確か？

6.3 監視・コントロール(RM.3)

- 機能名と機能が一致しているか？
- 1つで規模が大き過ぎる機能はないか？
- 特別な非機能要件はないか？
- 連携する他システムと役割分担は明確か？

事例 20 要件定義ドキュメントの形式品質

(富士通株式会社)

【取り組みの背景】

昨今、要件定義の抜け漏れが問題視され要件定義の重要性が認識される一方で、稼働を考えると要件定義に割ける時間が少ししかないという矛盾した状態になることが多くある。要件定義で決めなければならない事を本当に決めているのであろうか、要件定義ではそもそも何を決めなければならないのか。

当社では、要件定義に必要な要素を記載しているのか(集合)、記載した内容に矛盾はないのか(整合)のチェックを実施しているので紹介する。これは、記載してある内容の意味が理解できないとチェックできないものではなく、第三者でも機械的にチェック可能なものとして考案・実施している。また、これらのチェックの効率化のためのツールも適用している。

【課題】

解決すべき課題は以下である。

- 要件定義で必要な記述要素を記述しているのか
- 記述した内容に矛盾がないのか
- 当社基本フォーマットを用いていなくともチェックできないか

【解決策】

(1) モデルの設定

まず要件定義のモデル(標準)を規定している。要件要素(要件定義で決めるべき要素)とドキュメント(当社基本フォーマットを記載)との関係を図 6.33 に示している。カテゴリは分類、集計用のものである。

モデルは、9つのカテゴリ、33の要件定義ドキュメントと、その中の487の要素から構成されている。このモデルをもとにプロジェクトの成果物を評価する。

モデルを設定している理由は、要件定義のドキュメントや進め方の決定権はユーザ企業側にあり、指定されることが多いからである。当社基本フォーマットを利用してくれると必要な要素はフォーマットに含まれるので記載すべき要素には問題がないはずであるが、それと異なるフォーマットを利用するとき、どのドキュメントにどの要素が記載されているのか、当社基本フォーマットとの差分は何なのかを明確にするのが狙いである。

6.3 監視・コントロール(RM.3)

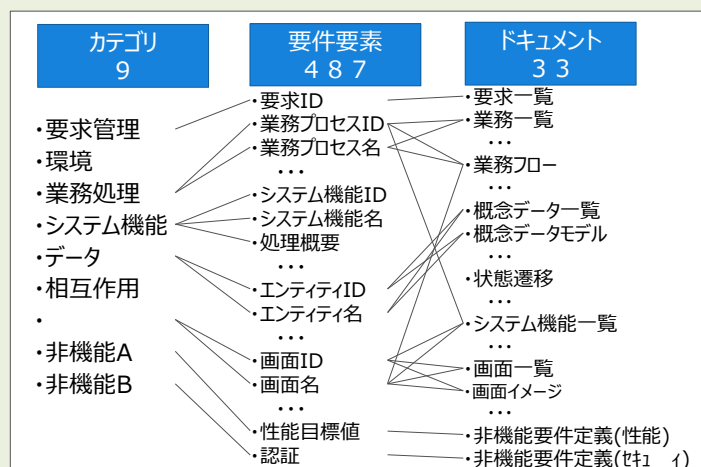


図 6.33 要件定義のモデル

(2) ドキュメントチェック

モデルで規定した要件定義ドキュメント 33 種を作成しているかをチェックする。

〈チェック例〉 概念データモデルが作成されているか

そもそもドキュメント単位で記載を省略することがある。記載していないドキュメントをチェックする。記載していない場合、記載しない明確な理由を説明できなければならない。

(3) 集合チェック

要件定義に必要な要件要素の有無をチェックする。

〈チェック例〉 業務一覧に業務プロセス ID や業務プロセス名が記述されているか

フォーマットが異なっても要素単位でチェックする。

(4) 整合チェック

要件定義ドキュメント間で関連する要件要素に矛盾が存在しないかどうかをチェックする。

・ 従属値一致チェック

ドキュメント間において、キー項目、および従属項目の一致チェック。

〈チェック例〉 画面 ID と画面名の関係が画面一覧でも画面レイアウトでも矛盾しない

・ 重複チェック

同一設計書中で一意にすべき設計値の重複チェック。

〈チェック例〉 画面 ID はもとより画面名も同じ名前が存在していないか

- ・ 参照元存在チェック

参照元の設計書の存在をチェック。

＜チェック例＞ 業務一覧に記述されている業務プロセス名が業務フローに記述されているか、画面一覧にある画面分のレイアウトが存在しているかなど

評価シートのイメージを図 6.34 に示す。



図 6.34 要件定義ドキュメントチェックの評価イメージ

【効果】

上記の各チェックは、第三者チェックサービス、プロジェクト自主チェックなど多くのプロジェクトで実施している。重点監視対象のプロジェクトでは、自己評価の実施を義務付けられることもある。実際、最初から 100 点を取れるプロジェクトは存在せず、評価して是正していくというプロセスになる。是正箇所が特定できるので効果的に是正処置が行われる。要件定義工程の完了時にチェックするだけでなく、(2) (3) のチェックは、要件定義で作成するドキュメントの計画時や各ドキュメントに記述する内容の標準規約を作成した時にも実施すると有効である。

また、プロジェクト期間中に業務別や人別にサンプルを選んで実施するプロジェクトもある。その業務や人の特徴を早期に捉えるという目的である。

6.3.4 コミュニケーション(RM.3.4)

要件定義の各プロセスは基本的に、調査、検討の結果から作成したドキュメントをもとに関係者で会議を実施して成果物の内容を合意するというプロセスを進める。その際に、なかなか内容の合意にいたりず、いたずらに時間が過ぎていくような事態が生ずることがある。ここでは、そのような事態に陥った場合、プロジェクトマネージャはどのような行動により事態を収束させるかについて述べる。会議が膠着した場合に、どのような対策が考えられるか、みなさんも考えてみていただきたい。

(1) 仕様がなかなか決まらないときは実態を確認して対処する

【解決したい問題】

- 部門間、個人間で意見が合わず、仕様の確定に時間がかかり、要件定義が収束しない

ビジネス要求をシステム化要求としてまとめていくときに、その仕様がなかなか決まらないことがある。また、一度決まった仕様が再三訂正されることもある。課題としては以下のことが考えられる。

- 主に業務担当者側の問題
 - 全体像が見えておらず、正しい仕様かどうかの判断ができない。
 - 部門間で意見が対立する
 - 業務担当者部門が要求事項を整理しきれていない
 - (再構築の場合) 現行システムがブラックボックス化しており、正しい仕様を業務担当者が認識できていない
 - レビューのたびに新たな意見が出て収束しない
- 主にシステム担当者側の問題
 - 業務知見が低い
 - 業務担当者側とシステム担当者側とで言葉が通じない(言葉の理解に齟齬がある)
 - やたらと専門用語を使ってしまう(業務担当者側が理解できない)
 - 作成する成果物の品質が悪い

上記の課題がすべて該当するわけではないので、仕様がなかなか決まらず、要件定義工

6.3 監視・コントロール(RM.3)

程が進まないのであれば、プロジェクトマネージャは実態を確認する必要がある。具体的には、進捗報告、成果物、個々の会議の議事録などのドキュメントを確認し、経緯および状況を確認する。加えて、業務担当者側、システム担当者側双方のプロジェクトメンバに対して、なぜ仕様が決まらないのかヒアリングを行う。コミュニケーションが原因とする場合、以下の対応策を検討する。

勘どころ① 仕様を決められる適任者をプロジェクトに参画させる

業務担当者側のメンバでプロジェクトの目標を理解し、業務知見がある適任者をプロジェクトに参画させないと、要件は決まらない。もしくは、正しい要件を抽出できない。業務担当者側として、適任者を選任し、プロジェクトに参画できるようにする。いわゆるキーマンの選任である。キーマンとなる適任者は、多忙な方が多いので、業務担当者側のプロジェクトオーナーは、アサインだけではなく、プロジェクトに参画できるように調整を行う必要がある。適任者がいないのであれば、要件定義はうまくいかない。その場合は、コンサルタントなどをアサインして、業務分析から始めるなどの抜本的な是正対策が必要となってくる。

システム担当者側については、業務に知見のある適任者を参画させるのが一番良いが、難しいのであれば、有識者を外部調達するか、業務担当者側に協力を要請する必要がある。気をつけなければいけないのは、適任者がアサインできず、担当者を短期間で何度も交代させることである。担当者を何度も交代させることは業務担当者側にとって不審に思われてしまうので、信頼関係が構築できなくなる。

勘どころ② プロジェクトメンバ間の信頼関係を構築する

信頼関係とは、要件定義を進めていくうえで、プロジェクトの目標を理解・共有し、プロジェクトを推進していく意識を待ち、プロジェクトメンバをお互いに対等の立場で認知することを示す。

業務担当者対業務担当者、業務担当者対システム担当者、システム担当者対システム担当者とメンバ間の関係にも色々パターンがあるが、いずれの関係についても信頼関係がないと要件定義に限らず、プロジェクトはうまくいかない。当然、当事者間での問題も解決できなくなってくるので、担当者を交代するなどの対応が必要となってくる。

とはいえ、前述のとおり、担当者の交代は、信頼関係を損なう可能性があるため、そもそも信頼関係が構築できるように細心の注意が必要である。

6.3 監視・コントロール(RM.3)

勘どころ③ 業務担当者側は当事者意識を持ち、会議に参画する

要件定義は、ビジネス要求をどうシステムに反映させていくかを決めていくプロセスなので、業務担当者の占める役割は大きい。要件定義の仕様がなかなか決まらない要素に、業務担当者側の当事者意識の欠如がある場合、会議の進め方を含めて見直しを行ったほうが良い。

「どうしてこのような業務の流れになっているのか?」「どうしてこのような機能が必要なのか?」というような質問を担当者同士で行い、議論しあうことで、相互理解を深め、共通認識を構築するプロセスを持つことが大切である。必要に応じて一定時間を確保し、ワークショップを開催し、相互理解や共通認識を持てるようにするのも良い。

場合によっては、部門をまたいで横断的に検討するための「分科会」の設置などを検討するのも要件をまとめるときに有効な手段となる。

勘どころ④ うまくいかない場合は、エスカレーションし、問題の解決を図る

対策を実施しても、コミュニケーション課題が解決できず、当事者同士の意見の衝突が発生し、合意形成ができなかったり、時間がかかったりする場合は、プロジェクト外の上位層で意思決定して、プロジェクトへ指示したほうが早い。したがって、プロジェクト計画時において、プロジェクト外の上位層で意思決定を行う責任者の任命と意思決定者へのエスカレーションパス、決定機関(ステアリングコミッティ)の設置と関係者への事前合意を行うと良い。

決定機関への参加者の例を以下に挙げる。

- プロジェクトオーナー
- 業務部門の責任者
- システム部門の責任者

定期開催に加えて、重大な課題が発生した場合や緊急な意思決定を要する場合に関係する責任者のみを招集して実施する臨時開催もできるようにすると良い。

上記のように、仕様が決まらない場合は、あらゆる手段を講じて、問題の解決を図るべきではあるが、どうしても仕様が決められないこともあり得る。このことについては、「6.4.1 要求評価」に記載している。合わせて参照して欲しい。

6.3.5 リスク(RM.3.5)

リスクの認識と発生時の対策を事前に計画することの重要性は「6.2.7 リスク認識」において述べた。ここでは、リスクが顕在化した際に、解決までどのような対策を取ることが必要であるかについて述べる。よくリスクが顕在化してもそれを放置して、より大きな事態を発生させることがある。発生したリスクの後始末は、要件定義マネジメントにおいても重要な業務である。

(1) リスクはプロジェクト終結までフォローする

【解決したい問題】

- リスクが監視されず、放置され、顕在化した時に影響を及ぼす場合がある

計画段階において、リスク登録簿を作成し、リスクの一覧を抽出している。しかし、いったん抽出したリスクをそのままに放置していることはないだろうか。放置した結果、リスクが顕在化し、プロジェクトに思わぬ影響を及ぼすことがある。これでは、せっかく抽出したにもかかわらず、意味がなくなってしまう。したがって、抽出したリスクについては、監視する必要がある。

勘どころ① 定期的なリスクの状況を棚卸しする

リスクは、プロジェクトが進行すると、状況が変わる。状況を監視するためには、進捗会議や、プロジェクトレビューなどで、リスクの棚卸しをする必要がある。リスク登録簿に状況を入力し、プロジェクトメンバと共有する。そのためにも、個々のリスクに対して、監視者(担当者)を指定するなどして、リスクを放置しないような仕組みを設定するなどの工夫をすると良い。ただし、監視者にすべてを任せるのではなく、他のプロジェクトメンバもリスクの監視をするという意識を持つことは重要である。

また、気をつけなければならないのが、状況に変化の見えないリスクである。本当に状況に変化がないのか、埋もれているだけなのか、注意が必要となってくる。要件定義プロセスは、プロジェクトの停滞が発生しやすいので、停滞しているように見えている状況が、既にリスクが顕在化している結果である可能性がある。特に要件定義プロセスはユーザ主体工程であり、どうしてもユーザに関連するリスクが多く結果的に埋もれやすくなるので、積極的に監視する必要がある。そのためにも、前述のとおり、進捗会議、プロジェクトレビューなどでリスクの棚卸しが重要となってくる。

6.3 監視・コントロール(RM.3)

勘どころ② リスクの状況に変化があれば、取り扱いを判断する

リスク状況に変化が見えたならば、あらかじめ決めておいた対策を適用するのか判断する。これはプロジェクトとして判断すべき事項である。その際、リスク登録簿として、監視を継続する必要があるかどうかをあわせて判断すると良い。

対策を講じた場合に、その効果を確認することも大事である。「対策を講じたから大丈夫！」ということはない。リスク対策を実施したとしてもそこで終わりではなく、対策の効果を評価して追加対策を打ったり、戦略を受容に切り替えたりすることもあり得る。そもそも「回避」策を打ったとしても回避が確実になるわけではないので、監視を継続して「やっぱダメそうだから転嫁も考える」など、監視を継続することも必要である。

また、リスク自体が消滅したことを確認できたのであれば、リスク登録簿のステータスを「消滅」などにし、監視の対象から外す。

勘どころ③ 影響の大きいリスクは、ステアリングコミッティで共有する

プロジェクトに対して影響の大きい重要なリスクについては、ステアリングコミッティなどで共有し、対策を検討することも大事となる。

いずれにせよ、リスクが消滅、解決するまで監視するスキームを確立する必要がある。

事例 21 「リスク管理リスト」によるリスク管理 (セイコーエプソン株式会社)

【取り組みの背景】

システムの大規模化・複雑化、ビジネス環境の急速な変化など、プロジェクト推進上のリスクは年々高まってきている。

特に要件定義は、確立された手法が確立されていないことや、人対人のコミュニケーションを重視した活動が多いため、キーパーソンの時間が取れない、要件定義者が要求を引き出すことができない、プロジェクトの環境変化による要求の変化等、要件定義ならではのリスクが存在する。

そのようなリスクを見落としてしまうと、手当が遅れてトラブルが拡大し、手戻りによる納期遅延、コスト超過などプロジェクト全体の QCD に重大な影響を与えてしまうことが多い。

このようなことを防止するために、リスクを管理し、確実にフォローアップすることで、リスクをコントロールすることが要件定義においても求められる。

【課題】

本事例では、解決すべき課題を以下のとおり設定した。

- 課題やリスクが提示されても、それがそのままの状態では放置され、成果物への反映などの解決に至らないままになってしまう

【解決策】

(1) 「リスク管理リスト」によるリスク管理の立案

事例 6 において、システム化構想・企画に必要な実施項目について述べた。システム化構想・企画では、システム化の背景・目的と、目的を実現するために解決すべき問題・課題とその解決策、プロジェクト計画を明確にする。さらに、解決策の実施やプロジェクトの実行に伴って発生するリスクを想定し、その発生や影響を極小化するための対策を検討することが重要である。

その際、想定されるリスクを「リスク管理リスト」を使って、以下のような事項を検討し管理する（表 6.24）。

- 想定リスク
解決策の実施やプロジェクトの実行に伴って発生するリスクを洗い出す
- 影響度
想定リスクが発生する可能性と、発生したときの影響度を「高」「中」「低」

6.3 監視・コントロール(RM.3)

で評価し、「高」を中心に対策を検討する

- 発生原因

リスクが発生する原因を想定する。表面的な現象ではなく、根本的な原因を分析する

- 対策

リスクが極力発生しないようにする予防対策と、リスクが発生した場合の影響を極小化する発生時の対策の両面から対策を検討する

- 実施時期

リスク予防対策や、発生した際に対策を実施する時期を検討する

- 対応責任者

対策を実施する対応責任者を明確化する

表 6.24 リスク管理リスト

ID	想定リスク	影響度	発生原因	対策	実施時期	対応責任者

(2) リスクの確実なフォローアップ

(1) で立案したリスク管理リストに基づいて、適切な実施時期に対策を実施する必要がある。しかし、プロジェクトの進行に従ってリスクが放置され、対策が実施されないことを防止するために、確実なフォローアップが求められる。

また、リスクをもたらす前提となる事実や不確実性は、プロジェクトを通じて常に変化するため、既存リスクの再評価による影響度や対策の変化や、リスクが消滅したり新たなリスクが発生する場合がある。したがって、リスク管理はプロジェクトの初期段階の計画だけでは不十分であり、要件定義も含めたプロジェクト全体を通じて継続的に実施する必要がある。

プロジェクトの全体を通して、残存リスクの監視と新たなリスクの洗い出し、対策の実施および評価を行い、「リスク管理リスト」に反映させる。

上記のようなリスク管理のフォローアップは、デザインレビュー等の何らかの社内ルールに組み込み、確実にフォローアップされるように仕組み化する。

【効果】

本事例では、「リスク管理リスト」によるリスク管理の策定により、リスクの予防と発生した場合の影響を極小化するようコントロールするとともに、確実なフォローアップにより、リスクが放置されたり解決に至らないままになることを防止し、プロジェクトのQCDを確保する効果を得ることができた。

6.3.6 トレーサビリティ(RM.3.6)

要件定義において作成される各種成果物は相互に関係性をもっている。一般に下位の(後で作成する)成果物は上位の(先に作成される)成果物の内容を受けて作成される。したがって、上位の成果物に変更が発生した場合、下位の成果物それぞれに影響があるかどうかを見極めて、修正が必要な場合には適正に修正する必要がある。下位の成果物に修正が発生した場合も同様に影響の有無は確認しておく必要がある。このような相互関係は、何らかの方法により相互の関係性を管理しておくことが、修正の際の作業漏れを防止する有効な手段になる。

(1) ビジネス要求とシステム化要求の関係は後で追跡できるように台帳で管理する

【解決したい問題】

■ ビジネス要求がシステム化要求にどう反映されているのか分からない

ビジネス要求をシステム化要求に反映する場合、どのビジネス要求がシステム化要求と関係しているのか意識しながらドキュメントを作成するのが一般的である。しかし、システム化要求を整理し、要件定義をまとめている段階においても、ビジネス要求の追加・変更は発生する可能性はある。その場合もドキュメント相互の関係を保つ必要がある。そうしないと、以下のような課題が発生することが想定される。

- ビジネス要求がシステム化要求にどう反映されているのか分からない
- ビジネス要求の変更に合わせたシステム化要求の変更で漏れ・ミスが発生する

ビジネス要求の変更がシステム化要求のどこに影響するのか分からない上記課題に対応するため、以下のような対応策を実施した方が良い。

勘どころ① ビジネス要求、システム化要求それぞれの要求事項を台帳管理する

ビジネス要求、システム化要求それぞれの要求事項を台帳管理する。台帳管理するにあたり、要求ごとの項番(ID)を割り振ることが望ましい。

台帳に整理されたビジネス要求(ビジネス要求台帳)に要求ごとに振られた番号を、台帳に整理されたシステム化要求(システム化要求台帳)のシステム化要求ごとに該当する

6.3 監視・コントロール(RM.3)

ビジネス要求の番号を付与していく。

表 6.25 ビジネス要求とシステム化要求の対比表の例

ビジネス要求		システム化要求	
ID	内容	ID	内容
BR1		SR1	
		SR2	
		SR3	
		SR4	
BR2		SR5	
		SR6	
		SR7	
BR3		SR8	
		SR9	
		SR10	
		SR11	
		SR12	
		SR13	

勘どころ② システム化要求の漏れ・抜けを確認する

システム化要求台帳に記載されたビジネス要求番号が抜け・漏れなく反映されているかを確認する。ビジネス要求台帳には、システム化されない要求については、それが分かるような情報を記載しておく。

トレーサビリティ情報の作成を手動で行おうとすると、どうしても漏れが発生する可能性が高くなるので、なるべく機械的にチェックできるようにすると、精度が高くなるし、時間の短縮にもつながる。

例えば、モデリング言語などを使うと整理がしやすい。図式化されるので、言葉だけで表現されるものより、ユーザ側との認識が合わせやすいので、認識の齟齬は出にくい。

勘どころ③ ビジネス要求が変更になったら、システム化要求も台帳に基づいて漏れなく修正する

システム化要求定義を進めていく間でも、ビジネス要求の変更が発生する可能性はある。ビジネス要求に変更が発生した場合、当然のことながら、システム化要求にも影響が発生する。その場合は、「6.3.1 要件量・スコープ」の(3)に記載したような変更要求管理プロセスに従って処理をすることと並行して、ビジネス要求台帳を最新化し、影響範囲を特定する。システム化要求台帳も合わせて修正を行い、システム化するビジネス要求がすべて反映できていることの管理・監視を行っていく。

コラム9 モデリング言語の適用によるトレーサビリティの確保

システム設計を行う際の手法として、モデリング技法を適用する開発技法がある。モデリングとは、対象を抽象化して単純化（モデル化）することであり、それを表現する手法としてモデリング言語が使用される。モデリング言語の代表的なものにはUML (Unified Modeling Language) やSysML (Systems Modeling Language) などがあり、モデリングを適用した開発では、この言語を使用して要件定義から開発に至る各層の定義をモデルとして記述する。

モデリング技法を適用した際には、モデリング言語の持つトレーサビリティに関する記述や、モデリング言語を取り扱うツールに装備された各層のモデルの相互の関係を表示する機能によって、モデル相互の関連性を管理することができる。モデル変更の際には、それを使用して、他に変更を必要とするモデルがどれであるかを表示することができる。

以下に、モデル相互の関係性をモデル・トレーサビリティマトリクスとして表示した例を示す。この例では、モデリング言語で定義した各層のモデルをキー情報として関係付けした情報をもとにマトリクス化した情報を表示している。

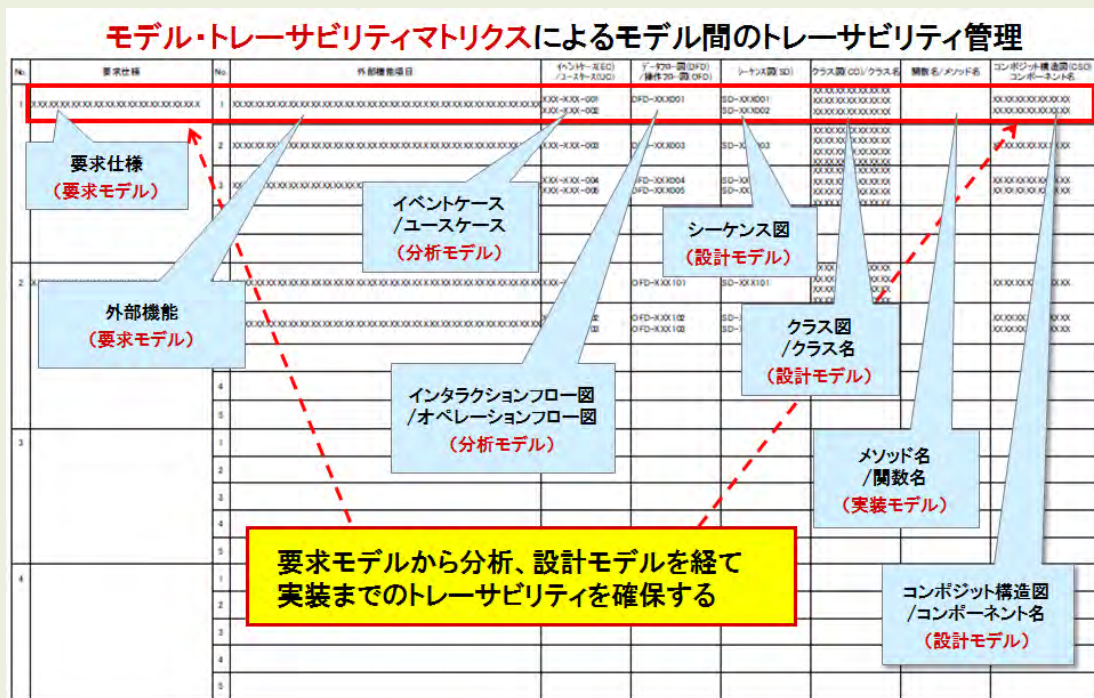


図 6.35 モデル・トレーサビリティマトリクスの作成例

(出典) IPA 社会基盤センター 先進的な設計・検証技術の適用事例報告書 2017 年度版
事例 81 統合モデル技術によるユーザ主体のソフトウェア開発手法
「モデル指向開発 (MOD)」の紹介 [31]

事例 22 要件定義・システム開発・保守管理リポジトリによる 要件定義ドキュメントの相互関連性の管理

【取り組みの背景】

A社では複数の情報システム開発案件を並行で遂行していたが、外部環境の変化に対応するため、各案件の要求仕様の変更が、要件定義・システム開発を経てシステム運用に至る各段階で多数発生し、それらへの対応に優先的に投入が必要になる社内・協力会社のリソースが年々増加してきた。そのため、本来実施すべき新たな案件への対応に手が回らず、新規システム投入の遅れによるビジネス機会の遺失が年々顕著になりつつあるなど、システム開発投資の経営への貢献度の低下が問題視されるようになってきた。

要件変更が多数発生することはビジネス上やむを得ない面があるが、保有する既存システムや、実施中の要件定義、システム開発の規模が大きくなるにつれ、個々の要件変更への対応にかかる費用・時間がこれまでよりも相当大きくなる傾向が顕著になり、これが新規案件へのリソース投入を困難にする要因としてクローズアップされてきた。

【課題】

これまでのシステム開発の実施方法では、「システム開発の目的」「業務要件」「システム化要求」の相互の関係や、これら要件定義の成果物とその後のシステム開発の各種成果物を関係付けて管理できていないため、以下の課題が発生していた。

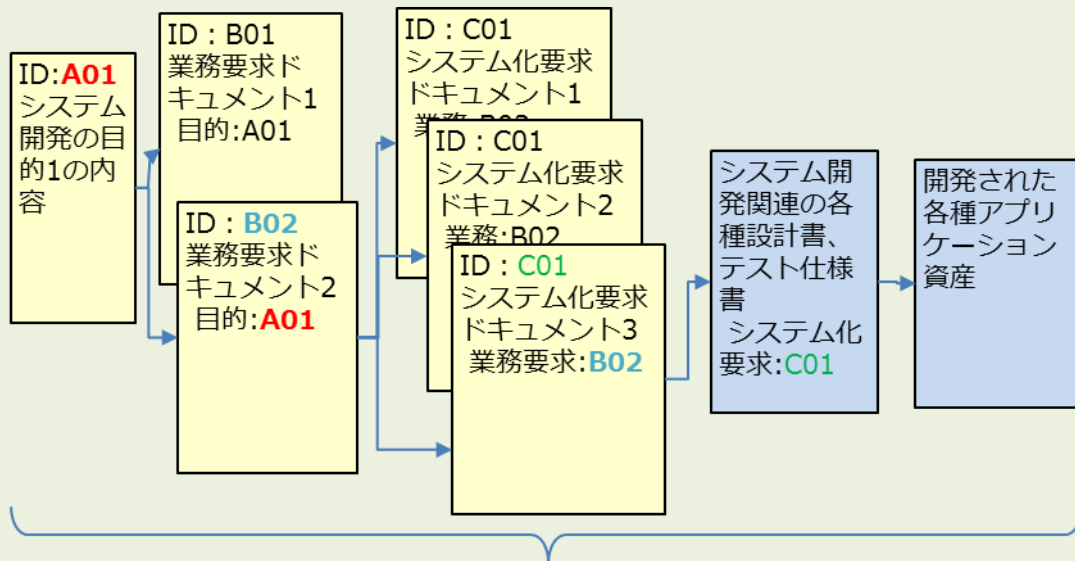
- 要件の変更があったときに、それぞれどのドキュメントのどこを修正すれば良いのか分からないので漏れが生ずる
- 修正の際に関係性の調査から始めないといけないので、修正に手間がかかる
- 修正範囲の調査漏れ・ミスのため、誤った仕様でシステム開発をすることになり、手戻りが発生する
- 要件変更により必要なくなった機能が、それと分からず維持メンテの対象になる

【解決策】

要件定義や、それに続く開発工程の各ドキュメントの関係性（トレーサビリティ）を確保するために、以下に施策を実施した。

- ① 各ドキュメントの内容をもとに後続ドキュメントに展開する際に、各ドキュメントが上位のどのドキュメントから派生したものを、上位ドキュメントに付与した ID を下位ドキュメントに記載することにより紐付けする。

- ② 上記を確実にするため、ドキュメントをリポジトリ管理ツールに読み込ませる際の規約(各 ID をどのドキュメント上のどの位置に記載するか、ID 命名規約など)を決定し、全プロジェクトで標準化する。
- ③ これらの関係性(トレーサビリティ)を要件定義・システム開発管理リポジトリ管理ツールで管理する。



要件定義・システム開発関連成果物管理用リポジトリツールで管理する対象

図 6.36 要件定義ドキュメント相互の ID での紐付けイメージ

【効果】

この事例では、この施策の実施により、要件定義内容やシステム開発仕様が開発や保守において変更になった際には、上記のリポジトリの情報をもとに関係を洗い出し、各種ドキュメントやアプリケーションの変更を行うことができるようになった。これにより、関連ドキュメントの洗い出しの精度向上や効率化を実現した。

また、相互の関係性をより正確に追跡できるようになったことから、以下のような効果を得ることができた。

- ① 各テーマが抜け漏れなくシステム化要求に展開されているかをドキュメントのチェーン関係を辿ることにより確認でき、要件定義した内容の実現漏れを防止できる(フォワードトレーサビリティ管理)。
- ② 要件定義の目的から紐付けされていないシステム化要求や実装された機能を洗い出すことができ、システム化の目的に合致していない機能を不要なものとして開発・維持の対象から除外できる(バックワードトレーサビリティ管理)。

6.3 監視・コントロール(RM.3)

なお、上記のツールを適用してトレーサビリティを管理する際には、作成するドキュメントからキーワードを抽出する部分の書式を定型なものにして、リポジトリツールに登録する際の精度を落とさないことが必要である。

6.3.7 見積り作成・投資効果分析(RM3.7)

要件定義の重要な目的の一つに、要件定義以降のプロジェクトの実施判断を経営層に仰ぎ、承認を得ることがある。要件定義の完了後に、経営判断としてプロジェクトの続行が確定した場合、これまでに投入した投資額や、推進する当事者の責任問題などから、それ以降の工程でのプロジェクトの中断、取止めは、かなり難しくなる傾向にある。

したがって、要件定義の完了時に、あらためてプロジェクト目標の妥当性や実現可能性、投資額の妥当性、投資額と期待効果のバランス等を客観的に示し、この先のシステム投資について経営層が適切に判断できるようにする必要がある

(1) 投資側には保守運用費を入れ、投資効果側は定性的な効果の客観性がポイント

【解決したい問題】

■ 定性的な投資効果に対する客観的な評価を求められて対応に苦慮する

最近のシステム投資は、ビジネスの構造を変えたり、質的にこれまでと違う状態を作り出すものが多くなっており、単純にコスト削減効果を目的にしないものも増えている。したがって、設備投資型の投資のように何年で投資額を回収できるかどうかで投資判断することは難しい。また定性効果を無理やり定量効果に換算して比較することも適切ではなく、定量効果と定性効果を合わせて、総合的に判断せざるを得ない案件が増えているが、その際の定性効果の客観性の確保が重要なテーマになってきている。

勘どころ① 投資額の見積りには、コンティンジェンシーや保守運用費用を入れて、算定する

投資額見積りの中で最大のものは開発人件費であるが、開発人件費は、システムの個別機能(画面、帳票、バッチ処理の個別機能)ごとに、予定した見積り方式(FP法、簡易FP法、SLOC法の概算レベルのものなど)で見積もり、その総和で算出する。一方、それ以外の開発人件費は、過去の類似方式、すなわち過去の要員投入実績をもとに、プロジェクトの特性の差異を考慮して見積もる。投資額の他には、開発に要するハード、ソフト、作業場所などの人件費以外があるが、その多くは、外部ベンダからの見積りを基本に算定する。また、移行や教育に要する費用も見積もる必要がある。

いわゆる規模拡大の法則(「2:4:2:3の法則」)もあるので、要件定義段階では、コンティンジェンシー(不測の要件拡大、条件の変更などへの備え)として、最低でも20%から

6.3 監視・コントロール(RM.3)

30%以上の費用を追加で見込んでおくのが望ましい。

また、投資側の費用には、開発費だけではなく、保守運用費用も見積もる必要がある。保守運用費には、開発・本番環境のハード・ソフトに関わる経常費用（ハード、ソフトの保守メンテナンス費用や減価償却費用など）、ならびに運用保守に関わる人件費などがある。

投資側の費用は、初期の開発費と稼働後3年間とか、5年間の保守運用費用の合計額で、開発費用だけではないことに留意いただきたい。また、投資効果側も3年間とか5年間の累積効果を算出して、投資側と比較することになる。

（比較する期間は、システムのライフサイクルによって変わる。例えば、変化が激しいフロント系のシステムは稼働後1-2年で判断していく必要がある場合もある。）

勘どころ② 定性的な投資効果をできるだけ客観化する仕組みを考える

定量的な投資効果については、システム投資による収益拡大に基づく利益増加効果部分と、効率化などによるコスト削減効果部分の2つで、一定の仮説や条件を置いて、3年間または5年間の累積効果額を見積もる。一方、定性的な投資効果については、評価の客観性を確保するために、定性評価ができかつ評価結果の実現に責務を負うラインマネジメント、ならびに、客観視する立場にある管理部門の有識者からなる評価チームによって、評価項目を5段階、または10段階で評価した結果の平均値を取って、定性効果を算定する。経営側には、図や表などを駆使して、分かりやすく Before/After を説明した資料を用意して、投資効果の内容をよく理解してもらった上で、定量効果と定性効果を合わせて総合的に判断を仰ぐ資料を作成して、経営側に説明を行うのが良い。（定性効果を定量効果に変換することは、恣意性も高くなり、逆に判断をゆがめるリスクもあるので、無理して行うことは避けた方が良い。）

なお、定性効果の客観性のためには、複数の受益部門(ライン)責任者の総合的な投資判断の平均値のほかに、投資効果実現に向けたラインマネジメントのコミットメント(実現責任を持つて経営側に約す)をもとに、経営側に判断してもらうことが現実的と言えよう。これに関連する事例として事例23を参照されたい。

事例 23 経営者に投資効果を分かりやすく説明する (投資判断の精度を確保する)

【取り組みの背景と課題】

中堅サービス業のT社では、基幹系業務のシステム化投資を終えた現在、システムの投資効果を、従前の「業務の手作業をシステム化すること等によるコスト削減効果」で説明することが難しくなっていた。

すなわち、従来のコスト削減効果が主体であった時代から、事業拡大や事業の質を高める活動を支援していくシステム投資案件が増えてきたために、経営側がシステム化投資判断をしやすい情報の提供が求められていた。

しかし、投資効果を定量的に表すことが難しい定性的な評価項目も多くなっており、定性的な評価項目に関して、いかに客観的な評価を行うかが、課題となっていた。

【対応策】

(1) 投資評価項目の事前確認

一定の規模以上となるシステム投資案件については、システム化計画完了後（要件定義工程に入る前に）、経営側に、経営会議において必要とされる「投資効果に関する評価項目」を事前に提示、確認しておくこととした。

(2) 「改善効果説明書」の必須化

経営会議には、要件定義書に記載の「システム投資の目的と達成目標」のほかに、新システムによって、何がどう良くなるのかを具体的、かつ簡潔にまとめた「新システムによる改善効果説明書」を提出し、経営側に新システムのメリットを分かりやすく説明することとした。

(3) 「システム投資効果評価表」の新設

改善効果やメリットは、システム化の効果として得られるものばかりではないことや、定性的な効果が中心となる案件も多いため、(2)の資料だけでは客観的な判断が難しいということで、新システムの受益部門、ならびに関連部門の責任者、およびそれに準じる管理職による複数の眼による投資評価をまとめた「システム投資効果評価表」（新設）を作成して、各部門の投資評価を、経営側に提示することとした。

(4) 受益部門責任者のコミットメント

(3)は新システムを利用する部門の責任者が投資効果の評価を行い、投資効果があると判断した以上、その投資効果を達成する責任が生ずるものとし、部門責任者の「責任ある評価」を求めたものであり、同時に部門責任者の「投資効果達成に関す

るコミットメント」を兼ねるものとなった。

また、評価結果が経営ならびに他の部門責任者に対してオープンになることにより、システム投資に著しく消極的になることを防ぐことも企図したものであった。

(5) 開発途中におけるマイルストーンの設定と評価の実施

基本設計完了時点、詳細設計完了時点、テスト完了時点において、プロジェクトの継続性の判断ポイントを設け、要件定義完了時点（投資決定時点）からの変化に対応した、プロジェクトの縮小や取り止めの選択肢も選択しうるマネジメント（仕組み）を取り入れた。

【効果】

各対応策を実施した結果と効果は、次のとおりであった。

(1) 投資評価項目の事前確認

- ・ 事前に経営側に投資判断に必要な項目を確認することで、プロジェクト側も評価項目に関して、より重点を置いた要件定義を行うことができた。
- ・ 今まではプロジェクト側で設定した項目で投資判断を仰いでいたが、逆に投資判断を難しくしていた部分があったことが明らかになり、経営側とのコミュニケーションの改善にもつながった。

(2) 「改善効果説明書」の必須化

- ・ 従前以上に、新システムの稼働によって、何がどう変わり、どう良くなるのか、メリットは何かを、経営側もより具体的かつ詳細に理解できるようになり、システム投資の目的やメリットがあいまいなまま実施されるプロジェクトはなくなった。

(3) 「システム投資効果評価表」の新設（具体例は後述）

- ・ 定性的な評価項目に関して、受益（利用）部門、関連部門責任者の評価が提示されるので、経営側の判断が容易になった。

(4) 受益部門責任者のコミットメント

- ・ 新システムを活用して評価項目を達成することが、受益部門のマネジメントに強く求められることになったため、システム投資を活かす現場のマネジメントが、従前以上に行われるようになった。その結果、投資効果の達成度合いも高まった。

(5) 開発途中のマイルストーンの設定と評価の実施

- ・ 常に最適な投資判断を行うことが担保されるため、経営側の迅速な判断にも繋がっている。（プロジェクトの途中での取り止めもできるという点で）

【対策と効果の実際例（システム投資効果評価表の具体例）】

カスタマーサービス(CS)部門の CRM システムの機能強化を目的としたシステム開発プロジェクトで、そのうちの一部機能である「CRM システムと販売管理システムとの連携機能」（この機能の開発費は 1,000 万円）の新設例を取り上げる。

<現状の課題と対応> (図 6.37、図 6.38 参照)

- ・ これまでは CS 部門の窓口オペレータは、既顧客からの問い合わせや相談に対応するだけであったが、1 年前から窓口オペレータに既顧客のニーズに対応した、新しいサービスを同時に紹介することを始めていた。しかし、CRM システムから注文を登録することができず、注文データの伝票を起票して、窓口対応の合間を利用して、販売管理システムに注文データを登録していた。
- ・ しかし、オペレータの負担も大きく、一部のオペレータからは不満も出ていたため、連携機能の開発を企画した。
- ・ この機能により CRM システムのデータは、自動的に販売管理システムに取り込まれるだけでなく、商品説明のアシストや受注時のさまざまな入力チェックがなされるため、後で顧客に再確認などの手間がなくなることが期待された。

<コスト削減効果>

- ・ しかし、窓口オペレータの作業時間の削減効果は、1 件当たり平均 2 分、月間の新規契約件数が 100 件ほどのため、月 3 時間 20 分の削減効果しかなく、かつオペレータの人件費削減効果は実際にはゼロである。したがって、コスト削減の投資効果は事実上ゼロであった。(ただし、本件の場合は売上拡大効果がある)(図 6.38 の①参照)

<定性的投資効果の評価実施>

- ・ そこで、投資効果について、評価項目を定め各部門の責任者による定量効果と定性効果評価を集め、その結果に基づいて経営に判断していただくことにした。評価は実施した場合のプラス効果と、実施しない場合のマイナス効果の両方を評価した。(図 6.38 の②参照)
- ・ 定性効果については、主観的な判断になる要素が大なため、関連部門も含めて 5 人以上の責任者からの評価の平均点をもって、客観的な評価とした。(図 6.38 の③参照)
- ・ 定性効果については、先の投資額（開発費 1000 万円）を前提にして、「5. 大いに投資効果が認められる」から「1. 全く認められない」の 5 段階で、受益部門

6.3 監視・コントロール(RM.3)

の責任者の評価点を1.3倍とし、その他の部門の評価を1倍として、その平均点が4点を上回る場合（図 6.38 の④）には投資効果を認めることとし、総合評価コメントを付して経営側に提示することとした。（図 6.38 の⑤参照）

- ・ このことにより、定性効果に関する経営判断上の客観性の確保と、受益部門の投資効果達成のコミットメントを担保した。
- ・ その例を図 6.37、図 6.38 で示す。（図 6.37 は評価者による個別評価、図 6.38 は複数人の評価をまとめた総合評価である。）

システム投資効果評価表 (個別評価)				
1. CRM・販売管理連携機能 (主要機能単位のケース)		なお、法規制への対応や情報システムコストの削減を主目的の案件は除く		
氏名	サービス1部 A部長			
評価項目	評価対象	評価詳細項目	システム投資を行った場合のプラス貢献度評価	
			数値効果 (受益部門は出来るだけ設定)	定性的な効果
PLへの貢献	◎ 売上拡大	CS部門経由の売上20%、年間で2000万円増加	1年後	2年後
	○ コスト削減 その他収益拡大	CS部門の作業時間 年間2400分、20万円削減	1年後	2年後
事業競争力の強化	◎ 他社との差別化、または差の解消	新しいビジネス機会の創出	1年後	2年後
	○ フラントイメージ、ブランド力の向上 マーケティング力の強化	CS部門経由の売上20%、年間で2000万円増加 CS部門の作業時間 年間2400分、20万円削減	1年後	2年後
品質、サービスの向上	◎ 製品、商品の品質向上	顧客対応時間を平均2分短縮	1年後	2年後
	◎ サービス品質の向上 トラブルやクレームの減少	顧客対応時間を平均2分短縮	1年後	2年後
グローバル化への対応	◎ 顧客満足度の向上	定期的な調査の結果で0.3ポイント向上	1年後	2年後
	◎ 事業の一体推進の推進 全体最適の実現 標準化の推進、マネジメントの統一化	定期的な調査の結果で0.3ポイント向上	1年後	2年後
働き方改革	◎ 労働生産性の向上	CS部門のオペレーターへの負担が下がること、顧客への提案型対応(サービスの拡充)への理解が深まり、モチベーションも大いに向上する。	1年後	2年後
	◎ 社員のモチベーション向上 公正な職場の実現	CS部門のオペレーターへの負担が下がること、顧客への提案型対応(サービスの拡充)への理解が深まり、モチベーションも大いに向上する。	1年後	2年後
事業リスクへの備え	◎ セキュリティーの強化	CS部門のオペレーターへの負担が下がること、顧客への提案型対応(サービスの拡充)への理解が深まり、モチベーションも大いに向上する。	1年後	2年後
	◎ 内部統制、IT統制面への貢献 BCP対応力の向上	CS部門のオペレーターへの負担が下がること、顧客への提案型対応(サービスの拡充)への理解が深まり、モチベーションも大いに向上する。	1年後	2年後
その他 (二つまで可)				
総合評価				

評価項目	評価対象	評価詳細項目	数値効果 (受益部門は出来るだけ設定)	定性的な効果	システム投資をしなかった場合のマイナス影響度評価
PLへの貢献	◎ 売上拡大	CS部門経由の売上20%、年間で2000万円増加	1年後	2年後	1年後
	○ コスト削減 その他収益拡大	CS部門の作業時間 年間2400分、20万円削減	1年後	2年後	1年後
事業競争力の強化	◎ 他社との差別化、または差の解消	新しいビジネス機会の創出	1年後	2年後	1年後
	○ フラントイメージ、ブランド力の向上 マーケティング力の強化	CS部門経由の売上20%、年間で2000万円増加 CS部門の作業時間 年間2400分、20万円削減	1年後	2年後	1年後
品質、サービスの向上	◎ 製品、商品の品質向上	顧客対応時間を平均2分短縮	1年後	2年後	1年後
	◎ サービス品質の向上 トラブルやクレームの減少	顧客対応時間を平均2分短縮	1年後	2年後	1年後
グローバル化への対応	◎ 顧客満足度の向上	定期的な調査の結果で0.3ポイント向上	1年後	2年後	1年後
	◎ 事業の一体推進の推進 全体最適の実現 標準化の推進、マネジメントの統一化	定期的な調査の結果で0.3ポイント向上	1年後	2年後	1年後
働き方改革	◎ 労働生産性の向上	CS部門のオペレーターへの負担が下がること、顧客への提案型対応(サービスの拡充)への理解が深まり、モチベーションも大いに向上する。	1年後	2年後	1年後
	◎ 社員のモチベーション向上 公正な職場の実現	CS部門のオペレーターへの負担が下がること、顧客への提案型対応(サービスの拡充)への理解が深まり、モチベーションも大いに向上する。	1年後	2年後	1年後
事業リスクへの備え	◎ セキュリティーの強化	CS部門のオペレーターへの負担が下がること、顧客への提案型対応(サービスの拡充)への理解が深まり、モチベーションも大いに向上する。	1年後	2年後	1年後
	◎ 内部統制、IT統制面への貢献 BCP対応力の向上	CS部門のオペレーターへの負担が下がること、顧客への提案型対応(サービスの拡充)への理解が深まり、モチベーションも大いに向上する。	1年後	2年後	1年後
その他 (二つまで可)					
総合評価					

評価項目	評価対象	評価詳細項目	数値効果 (受益部門は出来るだけ設定)	定性的な効果	システム投資をしなかった場合のマイナス影響度評価
PLへの貢献	◎ 売上拡大	CS部門経由の売上20%、年間で2000万円増加	1年後	2年後	1年後
	○ コスト削減 その他収益拡大	CS部門の作業時間 年間2400分、20万円削減	1年後	2年後	1年後
事業競争力の強化	◎ 他社との差別化、または差の解消	新しいビジネス機会の創出	1年後	2年後	1年後
	○ フラントイメージ、ブランド力の向上 マーケティング力の強化	CS部門経由の売上20%、年間で2000万円増加 CS部門の作業時間 年間2400分、20万円削減	1年後	2年後	1年後
品質、サービスの向上	◎ 製品、商品の品質向上	顧客対応時間を平均2分短縮	1年後	2年後	1年後
	◎ サービス品質の向上 トラブルやクレームの減少	顧客対応時間を平均2分短縮	1年後	2年後	1年後
グローバル化への対応	◎ 顧客満足度の向上	定期的な調査の結果で0.3ポイント向上	1年後	2年後	1年後
	◎ 事業の一体推進の推進 全体最適の実現 標準化の推進、マネジメントの統一化	定期的な調査の結果で0.3ポイント向上	1年後	2年後	1年後
働き方改革	◎ 労働生産性の向上	CS部門のオペレーターへの負担が下がること、顧客への提案型対応(サービスの拡充)への理解が深まり、モチベーションも大いに向上する。	1年後	2年後	1年後
	◎ 社員のモチベーション向上 公正な職場の実現	CS部門のオペレーターへの負担が下がること、顧客への提案型対応(サービスの拡充)への理解が深まり、モチベーションも大いに向上する。	1年後	2年後	1年後
事業リスクへの備え	◎ セキュリティーの強化	CS部門のオペレーターへの負担が下がること、顧客への提案型対応(サービスの拡充)への理解が深まり、モチベーションも大いに向上する。	1年後	2年後	1年後
	◎ 内部統制、IT統制面への貢献 BCP対応力の向上	CS部門のオペレーターへの負担が下がること、顧客への提案型対応(サービスの拡充)への理解が深まり、モチベーションも大いに向上する。	1年後	2年後	1年後
その他 (二つまで可)					
総合評価					

図 6.37 評価者による個別評価例

6.3 監視・コントロール(RM.3)

1. ORM・販売管理連携機能 (主要機能単位別のケース)		システム投資を行った場合のプラス貢献度評価										システム投資をしなかった場合のマイナス影響度評価				
評価項目	評価対象	評価詳細項目	数値効果 (受益部門は出来るだけ設定)				定性的な効果				コメント					
			A部長	B部長	C部長	D部長	E部長	平均点	A部長	B部長	C部長	D部長	E部長	平均点	コメント	
PLへの貢献	◎ 売上拡大	①	CS部門経由の売上20%、年間で2000万円増加					4	4	4	4	3	4	28		・ サービス1部、2部ともCS部門経由売上上げが伸びることは、確かと判断しており、もっと伸びる可能性もあると評価。 ・ CS部門のコストセンターから、専らフロントセンター化については、販売面一貫面である。マーケティング力の強化にもつながるとの評価と考えられる。 ④
	○ コスト削減		CS部門の作業時間 年間2400分、20万円削減					4	3	4	3	3	3.82			
	○ その他収益拡大		CS部門の作業時間 年間2400分、20万円削減					3	4	4	4	3	4.02			
事業競争力の強化	◎	他社との差別化、または差の解消	②					4	4	4	4	3	4.28		・ サービス品質の向上についても、講師効果は期待されるが、その他項目については、期待はやや下がる。 ・ 社員のマナー・マインドの向上については、投資効果が期待されている。 ⑤	
	◎	新しいビジネス機会の創出						4	3	4	3	3	3.82			
	○	マーケティング力の強化						3	4	4	4	3	3.56			
品質、サービスの向上	◎	製品、商品の品質向上						4	3	4	3	3	3.82			
	○	サービスの品質の向上						2	2	4	3	3	3.04			
	○	トラブルやクレームの減少						3	3	4	3	3	3.56			
グローバル化への対応	◎	事業の一体構造的推進														
	◎	全体最適の実現														
	◎	標準化の推進、マネジメントの統一化														
働き方改革	◎	労働生産性の向上														
	◎	社員のモチベーション向上						5	4	5	4	4	4.94			
	◎	公正な職場の実現														
事業リスクへの備え	◎	セキュリティの強化														
	◎	内部統制、IT統制面への貢献														
	◎	BCP対応力の向上														
その他 (二つまで可)																
総合評価																

図 6.38 複数人による総合評価例

6.4 終結(RM.4)

6.4.1 要求評価(RM.4.1)

要件定義を完了させる際に実施すべきことに、システム化の対象になった要求の評価がある。要求の評価とは、要求された事項が実現に値するものなのかどうかを個々に確認することであり、それについては「5.2.2 妥当性確認」で述べている。本項では、要件定義マネジメントの範疇で実施することとして、未決の要求の決着について述べる。未決事項の対処については、先に「6.3.4 コミュニケーション」でも述べているが、ここでは未決になった事項を最終的にどう決着させるかについて述べている。要件定義において結論が出ないという事態は時に発生する、ここに述べた対策は一つの考え方として参照して欲しい。

(1) 決まらない要件は影響を評価し、対応策を検討する

【解決したい問題】

■ 予定の期間で仕様が確定せず、後続工程に影響を与える

ビジネス要求からシステム化要求に反映し、要件定義書を作成することになるが、この要件定義書工程は、予定の期間で完了できないことがよくある。

後続の設計・製造・テスト工程の品質を担保する上でも、要件定義工程においてしっかりと要件を決めることは極めて重要なのは周知の事実であるが、だからと言って要件が決まらないからと期間をずるずると延ばすことは、後続工程の期間が圧迫されることになる。要件が決まらない原因として以下のことが考えられる。

- ユーザ側で要件が絞り込めない
- 部門間で意見が対立する

上記のような状況になると、なかなか要件定義が進まなくなるので、以下のような対策をあらかじめ決めておく。

勘どころ① 「決まらない要件」を識別する

要件定義工程においても進捗管理は行われるはずである。決まらない(決められない)要件は遅延タスクとして顕在化するので、決められない要件に関して、その対応を検討す

6.4 終結(RM.4)

る必要が発生する。まずは、どのビジネス要求がシステム化要求に落とし込めていないのか、はっきりと区別させる。

ただし、ここでは終結の章であるので、決められない要件が数件程度であることを想定している。大量に決められない要件がある状態であるならば、プロジェクト全体の進行に問題があるので、これはまた別の問題である。

勘どころ②「決められない要件」についてシステム化しない場合の影響を評価する

「決められない要件」に対して、以下のことを評価する。

- システム化しないことによる業務への影響
- 要件が決まらないことによる後続工程の影響

この評価を行うことで、対策も変わってくる。

ただし、この評価については、ユーザ自身が行うことはなかなか難しい場合がある。要求として上がっている以上、システム化したいという思いがあることが想定される。プロジェクトマネージャが判断できれば良いが、プロジェクトの当事者が判断するのも難しい場合もある。その場合は、第三者視点を交えると良い。ビジネスアナリスト(BA)のような人をアサインすることも検討すべきである。

勘どころ③「決められない要件」についての取り扱いを決定する

「決められない要件」を評価したのであれば、それぞれの要件について、対策を検討する。

例えば、業務上どうしてもシステム化が必要だという場合、要件定義のスケジュールを延ばしてでも、要件を決定する。というのも判断の一つとなり得る。その場合、後続工程に影響することになるので、その評価を加味したうえで、対策を決めたほうが良い。要件定義が延長した分、開発工程が短くなることは、プロジェクトにとって悪い影響しか与えない。要件定義工程を延長するなら、その分後続工程もそれなりに延長することが望ましい。

また、評価した結果、「決められない要件」について、システム化不要という選択肢もあり得る。該当する要件の利用者が限りなく少ない、業務を見直した結果、手作業で行ってもそれほど負荷にならない、などあり得る話である。

決められないのであれば、いったんスコープから外し、別プロジェクトで採用の可否を含めて再検討することも選択肢とするなど、柔軟な対応がプロジェクトの成功につながる。

いずれにせよ、決定した内容については、プロジェクトオーナーに承認をもらう。

プロジェクトオーナーは、「間違ってるかもしれないけど、もう間に合わないのだからこう決める。後で間違いがわかったら変更扱いとする。追加コスト支出には自分が責任を持つ」と言い切るくらいの決断をすることがプロジェクトを進めていくうえで大事なことになる。もし、そう言えないなら、機能をあきらめるかスケジュールを見直す方向で責任を負う。プロジェクトオーナーはそれだけ権限と責任があることを認識して欲しい。要件が変更になった場合は、変更管理の手順に従い、適切に対応する。

その場合も、プロジェクトなので、予算(C)・期限(D)などの制約事項を洗い出すことを忘れてはならない。これらの制約条件を踏まえたうえで、プロジェクトとして共有し、必要に応じてエスカレーションしてプロジェクト全体として判断をする仕組みを作っておくと良い。

とはいえ、要件未決のまま次工程に進む場合もあり得る。その際に考慮すべきことについては後述する。

6.4 終結(RM.4)

- (2) 内容に自信が持てないならそこには何らかの問題がある、思い切ってやめることも時には必要

【解決したい問題】

- 要件定義が完了したが、どうも内容に自信が持てない
 - ・ まだ検討が十分ではないと漠然とを感じるが、どこに問題があるのか分からない
 - ・ このまま進めても大した効果が上りそうな気がしない
 - ・ もう時間切れで次に進まざるを得ない

要件定義の「完了判断」を実施する際には、一般に要件定義の各作業が完了していることの確認、投資効果を分析した上でのプロジェクトの継続性審査、最終決裁者による最終承認を行うが、その際には以下のような課題が出ることもある。

要件定義プロセスは全体にアイデア創出の連続であり、どこまでいっても検討に終わりが見出せないが、時間や労力には限りがあるため、どこかで検討を打ち切り、そこまでに検討した内容に基づいて成果物を作成せざるを得ないという特性を持つ。そのため、要件定義を実施して一とおりの成果物が出揃い、あとは完了判断を待つだけという状態になっても、以下のような成果物に対する漠然とした不安感を抱くことがある。

- どうも要件定義の内容に自信が持てない
- この内容で本当に当初目的とした効果が発揮できるのか
- 今は浮かばないが他にもっと良いアイデアがまだあるのではないか

要件定義した内容に責任者・リーダーが自信が持てないなら、そこには何らかの問題が潜在している可能性がある。そのような状況を打開せず、そのまま完了判断をすることは決して望ましいことではない。そのような場合には、完了判断をする前に、以下のような対応により要件定義の実施内容の確認を行うことが必要になる。

勘どころ① 関係者を集めて要件定義の結果に問題がないことを確認する

成果物の内容や品質、想定されるリスクの評価を、開発するシステムのユーザ、ビジネスアナリストやプロジェクト管理者を含めたメンバで実施し、内容を修正する必要がないことを確認する。その上で、経営者に状況を素直に報告して、要件定義した内容の採否を、経営者を交えて最終合意する。

勘どころ② いったん、施策の実施を棚上げにする

6.4 終結(RM.4)

実施内容に対する自信がどうしても持てないが、解決策もない、関係者で会議をしてもこれで大丈夫という確証も得られない、というような状態に陥ったら、いったんその施策の実施を棚上げ・中止することを選択肢に加える。

上記のような検討により、要件定義した内容に対する不安感を払拭した上で、要件定義の完了判断を実施し、次の工程に進むことが重要である。

6.4 終結(RM.4)

6.4.2 完了判断(RM.4.2)

要件定義が完了したかどうかは、通常、計画時に実施すると決定した事項が、計画時に定めた品質をもって終了し、残作業、残課題がないかどうかで判断する。その際に、開発工程に進むかどうかの判断があわせて実施されて、投資対効果が十分にあると判断されれば、先にすすむことになる。ここでは、要件定義が完了した際のプロジェクトリーダーの悩みについても述べている。多額の投資を行うかどうかの判断はプロジェクトマネージャだけの責に帰するものではないが、そこに至るまでの葛藤をどう解決するかについての一考察として提示する。

(1) 実施すべき作業がすべて計画どおり実施されたことを確認する

【解決したい問題】

- 要件定義の各作業が完了したが、これですべて完了として良いかが判断できない

要件定義を完了させる際には、以下の各観点での確認が必要になる。

勘どころ① 要求(成果物)の妥当性を確認する

抽出したビジネス要求、業務要件とそれを実現するためのシステム化要求の定義結果が妥当であり、システム化すべきビジネス要求が要件定義書の成果物に反映されていることを確認する。確認対象の各成果物は、作成段階で自己チェックに合格し、関係者間で内容が合意されていることが前提になる。確認の際には、自社で共通して用いられている完了基準に照らし合わせてチェックすることが望ましい。また、確認はプロジェクト内での自己点検だけでなく、類似案件の有識者や組織内の検証機関などによる第三者検証を受けることが望ましい。

ビジネス要求として抽出されているもののシステム化要求から外れたものがある場合には、レビュー記録などの証跡により各ステークホルダと合意していることを確認する。

勘どころ② 未決事項と対応計画を確認する

成果物の品質確認とあわせて、未決事項の有無を確認する。未決事項がある場合には、未決の解消までの行動が具体的に計画されていることを確認する。(詳細は(2)参照)

勘どころ③ 完了判断のための材料を用意する

要件定義した対象をシステム化するかどうかの経営判断を、経営者やプロジェクトオーナーを交えて実施する。その判断ができるようにするために、以下のような材料を用意する。

- 投資
 - システム開発費用
 - システム維持・運用費用
 - 人員投入計画 など
- 回収
 - 収益計画
 - 社会的評価、従業員評価など直接収益に換算しづらいメリット など
- その他
 - 今後予想されるリスクと対応策 など

勘どころ④ 経営者を交えた要件定義の完了と後工程の開始判断を実施する

要件定義書の作成者は、上記①から③の確認結果をまとめて、経営者やプロジェクトオーナーに報告を行い、次工程へ進むことの承認を取る。

判断の際には、投資額や回収額を甘く自己査定していないかどうかをチェックする必要がある。プロジェクトの承認を得たいために意図的に甘く自己査定することはさすがにないとしても、心理的に甘く判断することはあり得る。開発が開始されたら中止判断は難しく、追加投資を認めて継続せざるを得なくなる。投資回収計画についてはプロジェクト担当者もチェックする側も厳しく評価する必要がある。もし疑義があれば双方で内容を精査し、それが解消するまで完了判断を延期するべきである。

また、プロジェクトオーナーは、この判断により、要件定義において実施すべき作業がすべて完了したことを判断するという、これにより要件が確定し、後工程がスタートするという意味と重要性を理解しておかなければならない。

要件定義が完了し、要件が確定したら、それ以降の変更はすべて要件変更案件として管理される。決定した要件を後工程で変更した場合、後になればなるほど対応のための費用と期間は級数的に増加する。

事例 24 経営者の投資判断を支援する (投資判断の精度を確保する)

【取り組みの背景】

Y社では、情報システム投資額の増加に疑問が提起されていたことや、投資効果に疑問のあるプロジェクトがいくつか発生したことから、情報システム部だけに任せておけないのではないかという問題認識が経営者にも生じ、情報システムの投資判断に関して改善が求められていた。

【課題】

情報システム部が作成するシステム化計画書をもとに経営会議で審議が行われていたが、専門的な部分もあり、経営会議で十分な検討が行うことが難しい状況にあった。

そのため、次のような課題があげられていた。

- 経営会議のボードメンバだけでは情報システム部が作成するシステム化計画の情報化投資の適正判断が難しいこと。そのため、厳格な投資判断が実施されていないため、投資効果の低いプロジェクトも実施されたケースがあったこと。
- システム化計画承認後は、経営会議メンバはステアリングコミッティ（以下、ステコミと称する）の評価結果の報告を受けるのみになっており、いったん承認されたプロジェクトの見直しは、ほとんど行われていなかったこと。
- システム稼働後の投資効果の測定、評価を行っておらず、投資効果判断を次の投資に活かすことができていなかったこと。

など、改善余地が大いにあることがわかった。

【解決策】

これまでの結果を踏まえ、情報化投資の意思決定から評価までのプロセスを、大きく変えることとし、次のようなプロセスとした。

- (1) 経営会議に諮る前に、投資委員会にシステム化計画書を諮る。
 - 一定額を超えた情報システム投資については、システム化計画書に記載の投資効果に関して、経営会議に諮る前に投資委員会の評価を受ける仕組みにした。
 - 投資委員会は、もともとは事業投資の投資判断を行うのが主務であったため、評価メンバは経営企画や管理部門を中心とした評価メンバが多かったが、情報化投

資案件の場合は、当該プロジェクトの主要な利用部門とそれ以外の現場管理職メンバを加え、必要に応じて外部専門家の意見を加えて、投資効果評価を行うものとした。

- ・ 投資委員会の評価は投資額算定の適正性のほかに、定量的な投資効果の妥当性と、定性的な投資効果の客観性に関するものであった。なお、評価結果は、投資委員会の意見書としてまとめ、情報システム部と経営会議メンバに対して事前に、または経営会議の場で説明を行うこととした。

(2) 要件定義工程完了時に、投資効果を再評価する。

- ・ システム化計画段階から大きく変わる面もあるため、要件定義工程の完了時に投資効果の再評価を投資委員会が行うこととした。
- ・ 評価に際しては、プロジェクトマネージャやプロジェクトリーダーのほかに主要なチームリーダークラスからのヒアリングも実施し、評価精度を高めることとした。
- ・ また主要利用部門の上位マネジメントの意見を記載することで、当該部門の責任も担保した。また、定性的な評価については、主要利用部門以外のメンバも一定数加えて、10段階評価方式による平均値を使って、客観性を確保することとした。
- ・ 上記結果の要約を投資委員会の意見書としてまとめ、ステコミに提示することとした。
- ・ ステコミでは、プロジェクトチーム作成の次工程のプロジェクト計画書のほかに、投資委員会からの意見書も考慮して、プロジェクトの継続性について審議し、ケースによってはプロジェクトの中止や、大幅な見直しを選択肢に入れて判断することとされた。(中止も選択肢の一つに明文化され、その判断が求められた点は大きな変化であった。)
- ・ その結果をもとに、ステコミのチェアマンであるプロジェクトオーナーが経営会議に、次工程計画の承認または中止判断を諮ることとした。

上記の業務概略フローを表したものが、次図である。

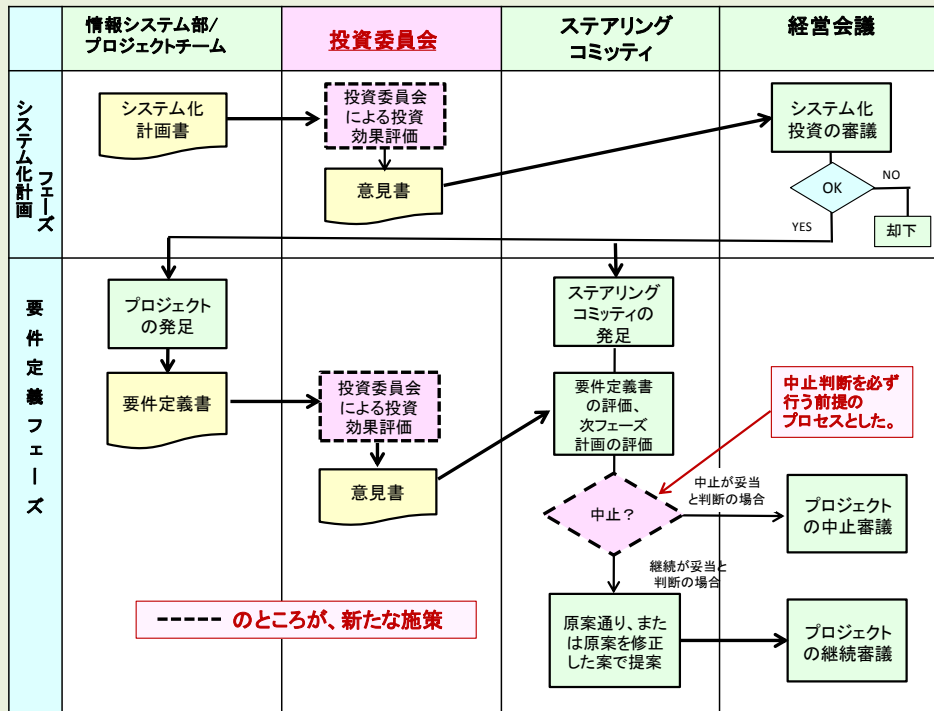


図 6.39 システム投資判断の業務概略フロー

(3) 稼働1年後をめぐり、情報システム投資の投資評価を行う。

- ・ 稼働後の評価については、いまだ実施段階にないが、この仕組みを入れることで、投資効果評価の厳密性を高めることができ、おのずと投資効果の高い情報化投資が行われるものと期待されている。
- ・ なお、投資効果の測定に関しては、投資委員会が要件定義書に記載された達成目標項目に関して、主要利用部門、および関連部門のマネージャクラスから達成度評価（10段階評価）と評価コメントを求め、分析結果を意見書にまとめ、情報システム部、ならびに経営会議に投資効果の報告を行うこととしている。
- ・ また、稼働後の投資効果に責任を持つ利用部門のプロジェクト責任者を再設定し、投資効果の達成責任をより明確にしている。

【効果】

いまだその効果を判断できる稼働後の評価データがないため、この仕組みの最終判断を行う時期にはないが、投資委員会を活用した情報システムの投資判断プロセスについては、経営側にとっても意思決定がしやすくなったとされ、投資判断に関する改善効果を経営者も認めている。

事例 25 要件定義の完了判断が難しい

【取り組みの背景】

中堅製造業の F 社では、前のプロジェクトで要件定義の完了条件が不明確だったため、ステアリングコミティ（以下、「ステコミ」と称する）で意見が分かれ、要件定義未了ということで、追加作業を強いられたことがあった。

【課題】

次のプロジェクトでは、プロジェクトチーム内はもちろんのこと、ステコミのボードメンバ間においても、要件定義の完了判断が異ならないように、要件定義の完了基準を定める必要があった。

【対応策】

取った対応策は次のとおりであった。

- (1) 要件定義の完了基準を定め、それに基づいて(2)以降の対応を行った。
 - ① 要件定義が開始される前に、表 6.26「要件定義の完了基準」を定めて、ステコミの承認を得た後、プロジェクトメンバ、ならびに関係者に周知した。
 - ② 要件定義の完了基準に基づいて、(2)以降の対応を取った。
- (2) 作成する成果物と詳細度指針の事前合意
 - ① 作成する成果物を一覧化して、ステコミボードメンバにも事前説明の上、確認を取った。
 - ② また、成果物（業務要件に関する成果物、ならびに画面、帳票などのシステム機能要件と非機能要件に関する成果物）ごとの詳細度の基準を定め、サンプルを作成して、詳細度不足の指摘を受けないように、メンバへの教育（説明会を開き、QA をしっかり行い、最初の成果物は途中でチェックしてフィードバック）を行った。
 - ③ その後は、成果物の作成過程の途中で、適宜チェックを入れるようにして、詳細度について基準を満たしているかを確認した。満たしていない成果物の作成者には、あらためて教育を行った。
 - ④ 詳細度の指針の基準例として、例えば、業務処理定義書においては、例外系も含めてすべての業務ケース、業務ロジックが網羅され、かつシステム対応要件か否か、ないしはペンディング要件か、が明示されていること。

表 6.26 要件定義の完了基準

要件定義の完了基準	
1. 関係者の合意（関係者とは、要件定義フェーズの計画時に定めた下記条件に関するユーザー部門、システム部門の承認権限者）	
① 業務改革要素に関する変更点に関して、関係者全ての合意が得られていること。 ないしは、不同意の関係者の合意に目途が付いているか、或いは一部の不同意でも実施可能な状態にあること。	
② 投資効果に関する分析評価がなされ、関係者のレビュー・承認が完了し、第三者が検証できる状態にあること。 また、投資額についても同様に見積られ、関係者のレビュー・承認が完了し、第三者が検証できる状態にあること。	
2. 成果物の品質	
① 計画されたすべての成果物が完成し、関係者のレビューが行われ、承認が得られていること。	
② 成果物の詳細度に関して、当初の計画のレベルに達していること、またそれを確認していること。	
③ 上記条件に反する一部の成果物がある場合は、後続工程への影響はほとんど認められず、また、影響が認められる場合であっても、その対応計画が次フェーズのWBSなどのプロジェクト計画に織り込まれていること。	
3. 残課題の影響	
① 業務要件に関する課題はすべて解決して、関係者の合意が取れていること。 ないしは、残課題の後工程への影響は限定的で、解決策を次フェーズで業務要件、システム要件などに反映させる工数は、軽微であること。また、確定合意に要するスケジュールとプロセスが明確になっており、且つ関係者間で合意され、後工程に影響がない程度に短期間であること。	
② システム要件に関する課題はすべて解決し、関係者の合意が得られていること。 あるいは、残課題の後工程への影響は限定的で、解決策を次フェーズでシステム要件、業務要件などに反映させる工数は、軽微であること。また、確定合意に要するスケジュールとプロセスが明確になっており、且つ関係者間で合意され、後工程に影響がない程度に短期間であること。	
③ 上記業務要件、システム要件の残課題の解決に関して課題オーナーの解決期限のコミットメントがあること。 並びに要件定義書、基本設計書への反映について、あらかじめ、次フェーズのWBSなどのプロジェクト計画に織り込まれていること。	

画面定義書であれば、主要な更新系画面を除き、画面レイアウトは不要、画面項目は画面機能を定義することに必要な項目以外は、その他の項目としても良い。入力チェック仕様は不要、画面遷移に関してはすべて網羅すること。等の要件定義設計指針をサンプルで示した。

(3) 成果物のレビュー品質の確保

- ① レビュープロセス基本指針（表 6.27 参照）をもとに、成果物ごとにレビュープロセスを定め、レビューも事前に決めて実施した。また、レビュー計画に基づき、レビューには、あらかじめレビュー工数とスケジュールを計画化しておいてもらい、レビュー時間を確保した。（表 6.27 はシステム機能要件に関する成果物）

表 6.27 レビュープロセス基本指針の例（システム機能要件に関して）

成果物	レビュー種別					備考
	システム側 チームリー ダー レビュー	システム側 PM/PL レビュー	システム側 テクニカル レビュー	ユーザー側 チームリー ダー レビュー	ユーザー側 PM/PL レビュー	
更新画面系	○	△※1	△※4	○	△※1	※1は複雑、または重要な更新処理のみ ※4は技術的な難度が高いもののみ
照会画面系	○	△※2	—	○	—	※2は複雑、または重要な照会画面のみ
帳票系	○	△※3	—	○	△※3	※3は複雑、または重要な帳票のみ
Batch系	○	△※1	△※4	○	△※1	※1は複雑、または重要な更新処理のみ ※4は技術的な難度が高いもののみ
基盤系	○	—	△※5	—	—	※5は特に技術難度の高い基盤設計のみ

② また、レビュー項目に、詳細度基準を満たしているか否かをいれて、詳細度の品質を確保した。

(4) 利害関係者の合意の確保

- ① 業務改革を伴う業務要件については、特に入念に利害関係部門のすべての合意を取るように会議体を設定し、会議の主催者には決定権限を有する方になって貰った
- ② システム要件についても、同様に関係者の合意が得られるための仕様検討会議を設定し、成果物に関して(3)で述べた計画的なレビューを行い、ユーザに最終承認を行ってもらった。

(5) 残課題の工数見積りの実施

- ① 残課題管理を徹底し、要件定義工程の後半からは残課題の解決ならびに、要件定義書への反映に要する工数を見積もり、優先度管理を徹底した。

【効果】

これらの対応策により、ステコミにおいても要件定義の完了に関して問題になることもなく、円滑に完了の判断をいただいた。

なお、要件定義が予定期間内に所定の品質で終わったのは、メンバ全員が最初から要件定義の完了条件を意識した仕事ぶりになっていたことも大きかった。

6.4 終結(RM.4)

(2) 要件未決のまま次プロセスに進む必要があるなら相応の対策を立てておく

【解決したい問題】

■ 要件未決のまま要件定義を終わらせて次工程に進む判断ができない

要件定義プロセスにおいて、ビジネス要求をシステム化要求にまとめていくが、納期を守るため、次プロセスに着手せざるを得ないことが、現実問題として発生することがある。しかし、要件が未決のまま次プロセスに進むことにはさまざまな問題が発生する。

- システム化のスコープ、ベースラインが決まらない
- 未決事項がいつまでも未決のままになる
- 未決事項が忘れられ、テスト工程などの後工程で影響が出る
- 未決事項を検討した結果、決定していた仕様に変更が入る(手戻り)

上記のような課題を発生させないためにも、要件が未決のままであるならば、要件定義の期間を延長してでも、未決要件事項をすべて解決すべきである。これが本来の要件定義プロセスのあるべき姿であることは認識すべきである。

このあるべき姿を逸脱して要件未決事項を残したまま、次プロセスに進むのであれば、相応のリスクを覚悟して進めていく必要がある。そのリスクを軽減するために、要件未決事項については、しっかりマネジメントしていく必要がある。

勘どころ① 未決事項の先送り基準を決めておく

要件定義プロセスにおいて、どうしても未決事項が発生する可能性はないわけではない。ただ、どれもこれも未決事項で先送りすることがよくないことは明らかなので、未決事項として、先送りする基準を決めておくと良い。例えば、以下のような基準である。

- システムの効果に直接関係するものでなく、システムスコープに影響を与えない
- 業務フローが確定している
- 明らかに解決する期日が決まっている

勘どころ② 未決事項を台帳管理し、ステークホルダと合意する

そのためには、台帳管理が重要となってくる。台帳には以下のような項目を管理すると良い。

表 6.28 要件未決事項台帳の例

項番	未決事項(タイトル)	未決事項(詳細)	対応状況	責任者	期限	未決による影響	決定までの見積もり工数	決定のための条件

台帳に登録を行ったら、ステークホルダを集めて、要件未決事項の内容について合意を取る。これは要件未決事項の内容だけではなく、期限など一覧の項目すべてについてである。誰がいつまでに要件未決事項を解決していくのかを合意し、ステークホルダは解決に向けて全面的に協力していくことも含めて合意をする。責任者だけに、解決を押し付けても解決することは難しい。プロジェクト全体で、要件未決事項解決に向けて、フォローアップできるような協力体制を築くことも大事である。

勘どころ③ 未決事項の内容が決定しない場合のリスクを洗い出し、監視する

プロジェクトマネージャは、要件未決事項の内容が決定しない場合のリスクを洗い出しておく。リスクが顕在化した時の影響などを考慮し、対策を検討する。このリスクに関しては、事象はもちろんであるが、顕在化した際の影響なども含めて、プロジェクトとして共有しておく。特にユーザ側に認識してもらおうと良い。

さらにプロジェクトマネージャは、この要件未決事項については定期的に状況を確認し、解決状況を確認していく。もし、状況が芳しくないのであれば、対策を検討する。ただし、未決状態の先送りだけは絶対に避けるべきである。

6.4 終結(RM.4)

第7章 要件定義の主要ドキュメント作成(DD)の勘どころ

「DD 文書記述」では、4 章、5 章で用いた成果物も含め、要件定義で作成する 36 の主要な成果物をここに集約し、各ドキュメントの記載項目とサンプルドキュメント、作成時の留意事項などを提示している。

成果物の課題領域を、「DD.1 ビジネスコンセプト」、「DD.2 ステークホルダ」、「DD.3 要求分析」、「DD.4 データモデル」、「DD.5 ビジネスプロセスモデル」、「DD.6 相互作用モデル」、「DD.7 コミュニケーション」、「DD.8 各種一覧」、「DD.9 インターフェース」、「DD.10 データ定義」、「DD.11 機能・データ整合性検証」、「DD.12 非機能要求」、「DD.13 運用、移行、総合テスト」の 13 に分類し、それぞれの作成の勘どころを提示している。

「DD 文書記述」における課題カテゴリマップを図 7.1 に示す。



図 7.1 「DD.文書記述」の課題カテゴリマップ

7.1 ビジネスコンセプトを確認する(DD.1)

「4.1.3 ゴールの抽出」でも述べているように、経営レベルの目的や施策は、構想立案時やシステム化企画時に検討され、要件定義が開始される時点で提示されるのが通常である。そのため、ここではビジネスコンセプトの確認に使用されるドキュメントの紹介にとどめる。

7.1.1 ビジネスコンセプト確認ドキュメント(DD.1.1)

【目的】

- 中長期計画や構想立案、さらには、それらが反映されているだろう、計画書やRFPなどに記載されているビジネスコンセプトを整理し、経営層に確認し、向かうべき方向性を合意する。

【留意事項】

- 既存ドキュメントで十分表現できている場合は作成する必要はない。
- 整理がされていない、再確認が必要などの場合、以下に紹介するドキュメントなどを利用して簡単にまとめる。
- 構想立案自体を実施しなければならないと判断した場合は、要件定義工程外の作業として別途期間を確保して実施する。
- 以下に紹介するドキュメント以外にもビジネスコンセプトをまとめる多数の手法、ドキュメントが存在する。ここでは、一部の手法を紹介し、このようなまとめ方があるということを理解して欲しい。

SWOT

経営戦略や計画の現状分析を行うための分析手法。自社の内部環境と外部環境の分析を統合的に行い、自社のビジネスの機会を発見するために役立つ手法。

SWOTのイメージを図7.2に示す。

7.1 ビジネスコンセプトを確認する(DD.1)

	プラス要素	マイナス要素
内部環境	Strength : 自社の強み <ul style="list-style-type: none"> ・親会社が多角化している ・強い地域主義 ・認知度が高く、ブランド化している ・販売数、XXX地域で優勢 	Weakness : 自社の弱み <ul style="list-style-type: none"> ・XXXのような国内外への影響力がない。 ・スタッフは自分の役割以外にも多くの負担を強いられている。 ・新しい顧客を引きつけるコンテンツがない
外部環境	Opportunity : チャンスとなる外部要因 <ul style="list-style-type: none"> ・デジタル・メディア、モバイルの利用により新たな顧客を開拓できる可能性がある ・顧客チャネルの多角化 ・新しい提供形態 ・顧客ごとにテーラリングした商品の提供へビジネスモデルの変更 	Threat : 脅威 <ul style="list-style-type: none"> ・業界全体が低迷している ・オンライン・メディアの理解に苦労 ・オンラインでは競争相手が多い ・異業種や個人の新規参入が増えている

図 7.2 SWOT のイメージ

SWOT の「S=Strength」、「W=Weakness」については、自社の企業努力でコントロールできる内部要因。逆に、「O=Opportunity」と「T=Threat」については、政治動向や規制、経済・景気、社会動向、技術動向、業界環境やユーザのニーズの変化など、自社の企業努力だけで変えられない外部要因。

Strength : 自社の強み

技術力の高さや長年の運用経験など、自社が持つ強みについて記述する。ユーザがなぜ自社サービスや商品を利用してくれるのかを考える。

Weakness : 自社の弱み

自社の弱みや苦手なことについて記述する。コストやリソースなどで競合よりも足りていない部分や、情報の打ち出し方など、自社が苦手とする部分を洗いざらい抽出する。

Opportunity : チャンスとなる外部要因

自社にとってビジネスチャンスとなるような環境変化や、変化に対して競合他社がどのような動きをしているのかなどについて記述する。徹底的にデータを収集し、どれほど小さなことでもチャンスとなりうる要因を1つでも多く抽出する。

Threat : 脅威

自社の強みを打ち消してしまう危険性のある環境の変化や、競合他社の動きなどを

7.1 ビジネスコンセプトを確認する(DD.1)

記述する。外部要因なので、自社の企業努力だけでまかなえない部分もあるが、脅威を知ることで新たなビジネスチャンス抽出することもできる。「Opportunity」と同様、ささいなことでも脅威となりうることはすべて抽出する。

バランススコアカード (BSC)

バランススコアカード (BSC) [32]とは、「財務」「顧客」「社内ビジネスプロセス」「学習と成長」の4つの視点で業績管理指標をバランスよく組み合わせ、戦略実行や業績評価を行うための手法。

バランススコアカード (BSC) の例を図 7.3 に示す。

	戦略目標 (KGI)	重要成功要因 (CSF)	業務評価指標 (KPI)	ターゲット数値	アクションプラン
財務の視点	<ul style="list-style-type: none"> 利益性 低コスト 売上拡大 	<ul style="list-style-type: none"> 市場の評価 少ない機種 顧客の拡大 	<ul style="list-style-type: none"> 株価 飛行機のリースコスト 1座席あたり売上高 	<ul style="list-style-type: none"> 30%アップ 20%ダウン 10%アップ 	<ul style="list-style-type: none"> 低コストで売上増強
顧客の視点	<ul style="list-style-type: none"> 定刻離着陸 低価格 	<ul style="list-style-type: none"> スケジュールを守る 常連客の確保 	<ul style="list-style-type: none"> 定刻離着陸 顧客の定着率 	<ul style="list-style-type: none"> 30分以内 30%以上 	<ul style="list-style-type: none"> 業務の品質管理と顧客定着率アップのプログラム開発
業務プロセスの視点	<ul style="list-style-type: none"> 実稼動時間のアップ 	<ul style="list-style-type: none"> 時間の厳守 	<ul style="list-style-type: none"> 定刻着陸率 定刻離陸率 	<ul style="list-style-type: none"> 90%以上 90%以上 	<ul style="list-style-type: none"> サイクルタイムの最適化
学習・成長の視点	<ul style="list-style-type: none"> 地上クルーのチームワーク 	<ul style="list-style-type: none"> 従業員のモチベーション 教育 	<ul style="list-style-type: none"> 地上クルーの持ち株比率 地上クルーの教育訓練度 	<ul style="list-style-type: none"> 1年度 70% 2年度 90% 3年度 100% 年4回 	<ul style="list-style-type: none"> ストックオプション 地上クルーの教育訓練

テクノファNEWS 第44号から抜粋。「テクノファIT コーディネーター・シンポジウム」2002年7月11日収録
<https://www.technofer.co.jp/downloadfiles/pdf14.pdf>

図 7.3 バランススコアカード (BSC) の例

ビジネスモデルキャンパス

ビジネスモデルキャンパスの作成はビジネスモデル全体の把握に役立つ手法である。

図 7.4 に示すように、ビジネスモデルキャンパス [33]は、9つの欄に分かれている。中央が価値提案であり、ビジネスモデルの中核が価値提案であることを表している。価値提案を中心に、右側がマーケティング要素を表現している。左側は、価値提案を実現するための組織体制やマネジメントなどのバックエンドの仕組みである。下側が収益とコスト構造である。この全体で、ビジネスの全体像を表す。

7.1 ビジネスコンセプトを確認する(DD.1)

ビジネスモデルキャンパス

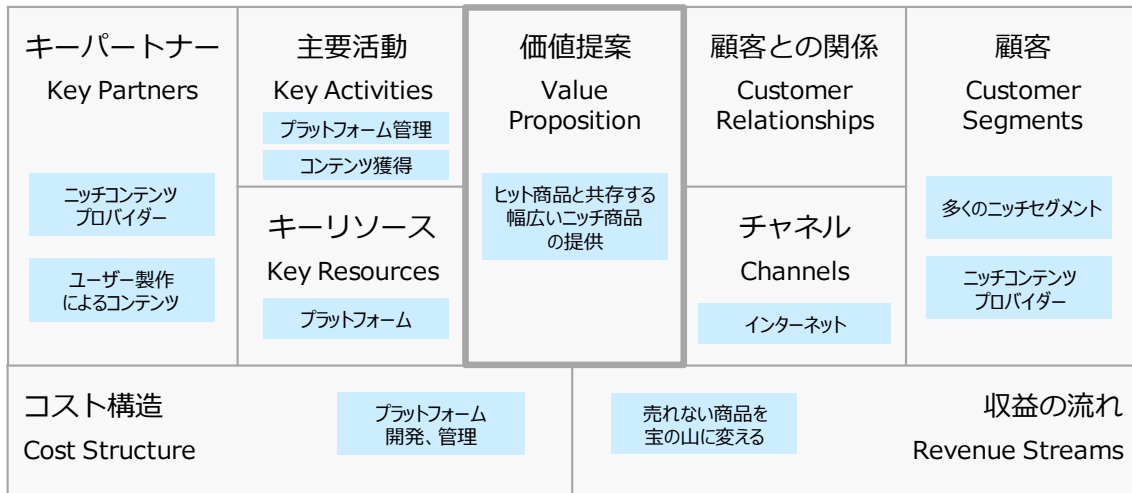


図 7.4 ビジネスモデルキャンパスのイメージ

7.2 ステークホルダを漏れなく、正確に記述する(DD.2)

昨今ステークホルダが多様化しており、関係するステークホルダを漏れなく抽出することが、成功への第一歩となる。

また、適切なステークホルダの抽出ができたとしても、各ステークホルダの問題認識に対して、適切な課題の設定や解決策の提示ができないために合意形成に時間がかかったり、ステークホルダ間での調整に時間がかかったりして、要件定義の期間を超過するケースも多い。したがって、過不足なくステークホルダを抽出した上で、各ステークホルダの特性とステークホルダ相互の関連性を分析し、適切な要件定義アプローチを取る必要がある。

適切なアプローチを取ることで、ステークホルダそれぞれの理解を得られる要求として定義でき、スムーズに合意形成を進められるだけでなく、開発工程における要件変更リスクを低減することができる。

7.2.1 ステークホルダ関連図(DD.2.1)

【目的】

ステークホルダの漏れがない状態にする。

- 問題・課題の抽出先
- 要求抽出先
- レビュー先
- 合意形成先
- 要件定義プロジェクトメンバ

【説明】

- ステークホルダを中心に、その組織／役割などを表した図。

【留意事項】

- ステークホルダ関連図を記載する目的を設定する。
- 状況・目的に応じてフリーフォーマットで工夫する。次に示す成果物のイメージは一例にすぎない。
- キーマン存在先を検討し、そのキーマンがプロジェクトメンバとして参画できるかどうかを検討するのに利用すると良い。

7.2 ステークホルダを漏れなく、正確に記述する(DD.2)

【成果物イメージ】

- ステークホルダ関連図に記載する主な記載項目、イメージを表 7.1、図 7.5 に示す。

表 7.1 ステークホルダ関連図の主な記載項目

分類	項番	記述内容	記述内容の説明
1. 構成要素	1.1	ステークホルダ名	ステークホルダの名称
	1.2	組織、役割など	会社名、組織名、機能名、役割など
	1.3	ステークホルダ間の関係	業務上のやり取りなど

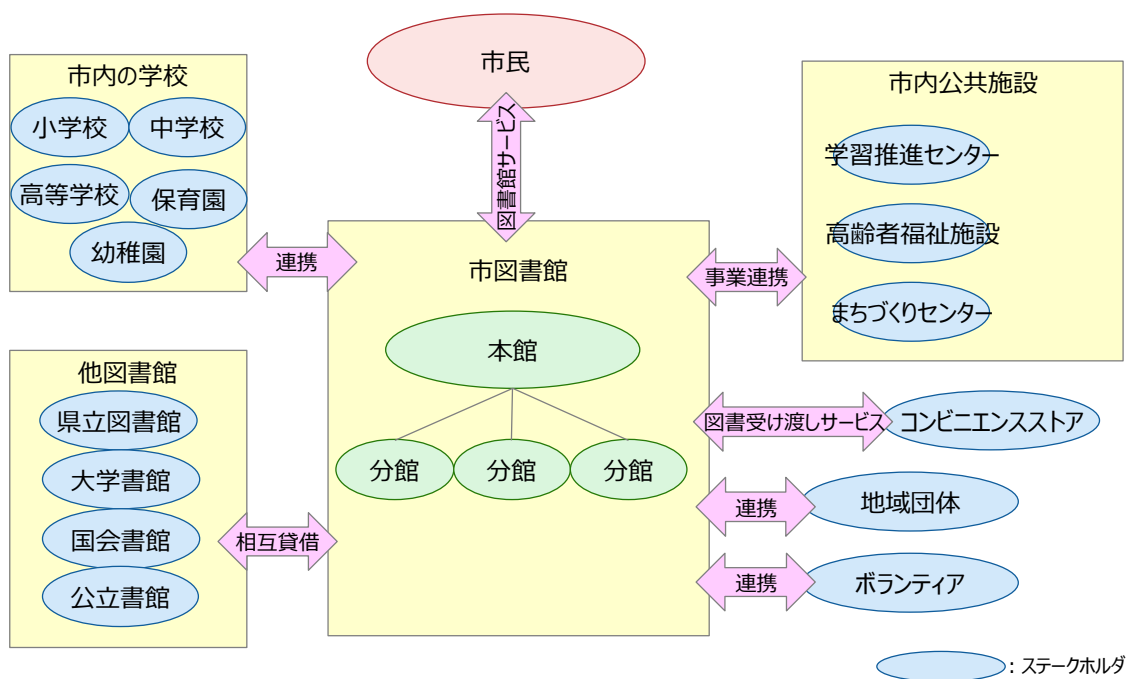


図 7.5 ステークホルダ関連図のイメージ

7.2.2 ステークホルダー一覧(DD.2.2)

【目的】

- ステークホルダとの課題抽出や合意形成などを効率的に実施していく状態にする。
- ステークホルダの漏れがない状態にする。

【説明】

- 洗い出したステークホルダを一覧化し、ステークホルダ各々のプロジェクトへの関与の姿勢や権限、影響度などの属性を記載する。

【留意事項】

- ステークホルダを洗い出し、各ステークホルダのプロフィール（組織、名前、役職、役割・職務など）を整理する。
- 各ステークホルダのプロジェクトへの関与（姿勢、承認権限、影響度、位置付け、重要度）を評価する。
- 適宜、評価の見直しを行う。

【成果物イメージ】

- ステークホルダー一覧に記載する主な記載項目、イメージを表 7.2、図 7.6 に示す。

表 7.2 ステークホルダー一覧の主な記載項目

分類	項番	記述内容	記述内容の説明
1. 構成要素	1.1	プロフィール	組織、名前、役職、役割・職務など
	1.2	プロジェクトへの関与（承認権限）	プロジェクトへの関与姿勢の評価
	1.3	プロジェクトへの関与（影響度）	プロジェクトに対する影響度の評価
	1.4	プロジェクトへの関与（位置付け）	プロジェクトにおける位置付け
	1.5	プロジェクトへの関与（重要度）	プロジェクトに対する重要度の評価

7.2 ステークホルダを漏れなく、正確に記述する(DD.2)

No.	プロフィール				プロジェクトへの関与				位置づけ	重要度	備考
	組織	名前	役職	役割・職務	承認権限	プロジェクトへの影響度	影響度	他のステークホルダーとの関係			
0001	物流部	中村	部長	関東地方の物流総括	要求内容の承認 プロジェクト運営の承認 予算の承認	大	大	営業)新垣Bとの 師弟関係あり	<input checked="" type="checkbox"/> プロジェクトオーナー <input checked="" type="checkbox"/> 意思決定参加者 <input checked="" type="checkbox"/> ヒアリング対象 <input type="checkbox"/> アンケート対象 <input type="checkbox"/> 要求検討WGメンバー <input type="checkbox"/> 要求内容レビューメンバー <input type="checkbox"/> その他	大	
0002	物流部	森田	担当	在庫管理担当		大	小		<input type="checkbox"/> プロジェクトオーナー <input type="checkbox"/> 意思決定参加者 <input type="checkbox"/> ヒアリング対象 <input type="checkbox"/> アンケート対象 <input checked="" type="checkbox"/> 要求検討WGメンバー <input checked="" type="checkbox"/> 要求内容レビューメンバー <input type="checkbox"/> その他	中	
0003	営業部	新垣	部長	関東地方の営業総括		中	大		<input type="checkbox"/> プロジェクトオーナー <input checked="" type="checkbox"/> 意思決定参加者 <input checked="" type="checkbox"/> ヒアリング対象 <input type="checkbox"/> アンケート対象 <input type="checkbox"/> 要求検討WGメンバー <input type="checkbox"/> 要求内容レビューメンバー <input type="checkbox"/> その他	大	
0004	営業部	塚田	担当	営業担当		中	小		<input type="checkbox"/> プロジェクトオーナー <input type="checkbox"/> 意思決定参加者 <input type="checkbox"/> ヒアリング対象 <input type="checkbox"/> アンケート対象 <input checked="" type="checkbox"/> 要求検討WGメンバー <input checked="" type="checkbox"/> 要求内容レビューメンバー <input type="checkbox"/> その他	中	
0005	△△自動車販売	佐々木	課長	顧客		小	中		<input type="checkbox"/> プロジェクトオーナー <input type="checkbox"/> 意思決定参加者 <input checked="" type="checkbox"/> ヒアリング対象 <input checked="" type="checkbox"/> アンケート対象 <input type="checkbox"/> 要求検討WGメンバー <input type="checkbox"/> 要求内容レビューメンバー <input type="checkbox"/> その他	中	
0006	××タイヤ	江波	部長	仕入先		小	小		<input type="checkbox"/> プロジェクトオーナー <input type="checkbox"/> 意思決定参加者 <input checked="" type="checkbox"/> ヒアリング対象 <input checked="" type="checkbox"/> アンケート対象 <input type="checkbox"/> 要求検討WGメンバー <input type="checkbox"/> 要求内容レビューメンバー <input type="checkbox"/> その他	中	
									<input type="checkbox"/> プロジェクトオーナー <input type="checkbox"/> 意思決定参加者 <input type="checkbox"/> ヒアリング対象 <input type="checkbox"/> アンケート対象 <input type="checkbox"/> 要求検討WGメンバー <input type="checkbox"/> 要求内容レビューメンバー <input type="checkbox"/> その他		
									<input type="checkbox"/> プロジェクトオーナー <input type="checkbox"/> 意思決定参加者 <input type="checkbox"/> ヒアリング対象 <input type="checkbox"/> アンケート対象 <input type="checkbox"/> 要求検討WGメンバー <input type="checkbox"/> 要求内容レビューメンバー <input type="checkbox"/> その他		
									<input type="checkbox"/> プロジェクトオーナー <input type="checkbox"/> 意思決定参加者 <input type="checkbox"/> ヒアリング対象 <input type="checkbox"/> アンケート対象 <input type="checkbox"/> 要求検討WGメンバー <input type="checkbox"/> 要求内容レビューメンバー <input type="checkbox"/> その他		

図 7.6 ステークホルダー一覧のイメージ

7.2.3 リッチピクチャ(DD.2.3)

【目的】

- ステークホルダの位置付けや心情などからステークホルダの依存関係や課題などを明確にし、ステークホルダとの対話を効率的に実施していく状態にする。

【説明】

- ステークホルダの主観的な思いや意見を絵と文字によって表す。

【留意事項】

- できるだけ言葉を使わないようにビジュアル的に書く。
- 課題を網羅的に抽出することを目的としない。
- ステークホルダの把握、ステークホルダの課題発見方法として利用する。

【成果物イメージ】

- リッチピクチャに記載する主な記載項目、イメージを表 7.3、図 7.7 に示す。

表 7.3 リッチピクチャの主な記載項目

分類	項番	記述内容	記述内容の説明
1. 書誌情報	1.1	リッチピクチャの ID	リッチピクチャを一意に識別するためのコード
	1.2	リッチピクチャの名称	本リッチピクチャの名称
	1.3	概要	リッチピクチャの範囲、目的を説明する
2. 構成要素	2.1	ステークホルダ	ステークホルダ
	2.2	思いや意見	思いや意見を吹出して記載
	2.3	課題	課題を吹出して記載
	2.4	依存関係	依存関係を線とその説明で記載する

7.2 ステークホルダを漏れなく、正確に記述する(DD.2)

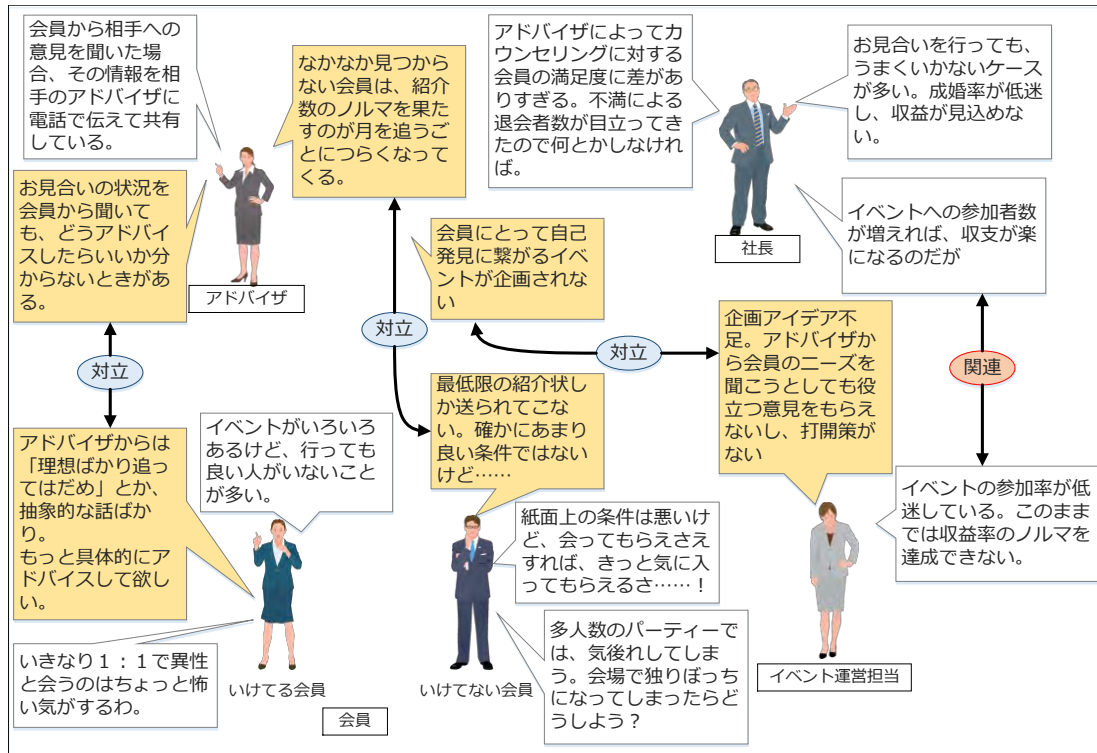


図 7.7 リッチピクチャイメージ (再掲)

7.3 重要な要求を漏らさず、正確に記述する (DD.7.3)

要求の分析は図 7.8 (再掲) に示すように、問題、ニーズ、課題、目的、手段等を明確にすることである。

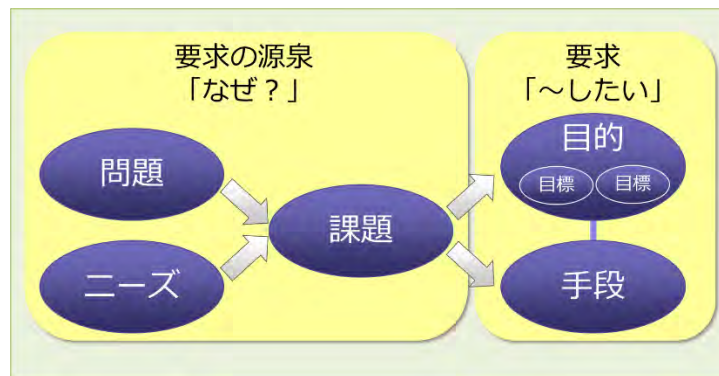


図 7.8 要求分析の基本要素の関係 (再掲)

ニーズや問題が明確になったら課題を設定する。課題とは解決すべきテーマのことである。この課題（解決テーマ）が設定できれば、目的を明確にして、手段を検討して、現実にした本質的な解決方法が導かれる。

ここでは、これらに役立つドキュメントを紹介する。

7.3.1 問題・ニーズ・課題一覧(DD.3.1)

【目的】

- ニーズや問題、解決すべき課題を明確な状態にする。
- 収集した問題点やニーズを一元管理できるようにする。
- 問題点・ニーズの漏れがない状態にする。
- 問題点、ニーズと要求の照合を実施することにより、要求の漏れがない状態にする。

【説明】

- 問題ごとに、抽出先、問題、原因、影響、ニーズ、課題を簡単に記載した一覧

【留意事項】

- 抽出時は、簡単に（1つだけ）原因、影響なども含めて記載してもらう。抽出が目的

7.3 重要な要求を漏らさず、正確に記述する(DD.7.3)

であり因果関係の分析に陥らないようにするため。

- 問題分析等の実施後では、その結果を記載する。
- 要件定義後半では、要求と問題の照合を行い取りこぼしがないか検討する。
- 原因、影響の考え方は、次項、問題原因分析図参照。

【成果物イメージ】

- 問題・ニーズ・課題一覧に記載する主な記載項目、イメージを表 7.4、図 7.9 に示す。

表 7.4 問題・ニーズ・課題一覧の主な記載項目

分類	項番	記述内容	記述内容の説明
1. 構成要素	1.1	No.	問題・ニーズ・課題一覧の通し番号（行番号）
	1.2	問題抽出テーマ	自由記述 問題を抽出時に提示した観点など
	1.3	問題分類	自由記述 期待効果など
	1.4	提起者	問題を抽出先。部署名や氏名などを記載する
	1.5	問題	今起こっている問題を記載する
	1.6	影響	問題の影響を記載する
	1.7	原因	問題の原因を記載する
	1.8	ニーズ	ニーズから出発しているものはここを記載する
	1.9	課題	解決すべきことを記載する
	2.0	備考	自由記述

NO.	問題抽出 テーマ	問題 分類	提起者	問題	影響	原因	ニーズ	課題	備考
001	販売物流 業務の改 善	コストダウン	営業 X氏	営業所において過剰 在庫をかかえている。 (4億円/年)	倉庫管理費が拡大 傾向にある。 (販売コストの 8%)	商品の在庫量は担 当者の判断であり、 安全な在庫を確保 する傾向が強い (判断者：1～2 人/営業)	営業所の過剰在庫 をなくす。	チェーンストアの販売 費に連動した適性 在庫を実現する。	
002									
003									
004									
005									
006									
007									
008									
009									
010									

図 7.9 問題・ニーズ・課題一覧のイメージ

7.3.2 問題原因分析図(DD.3.2)

【目的】

- 真の問題を発見し、より効果の高い、実現性のある課題を設定することにより、実現すべき要求の効果や実現性を高める。

【説明】

- 抽出された問題の原因を繰り返し表現（なぜなぜ分析）し因果関係を明確にしたもの。また、解決すべきポイントを明示したもの。

【留意事項】

- 抽出した問題の影響系（この問題が起こるとさらにどんな問題が起こるのか）を検討し視野を広げる。
- 上記で明確にした問題のなぜなぜ分析を行う。
- なぜなぜは、制約（人の問題、お金の問題など）に行き着いたらやめる。
- 対策を講じたらより効果的な問題（ここを解決すると複数の問題が解決できる）を発見する（解決ポイント）。
 - 解決すると定量効果の大きい問題を選ぶ
 - 悪循環を引き起こしている問題に着目する
 - トップの指摘する問題に着目する
 - 制約条件は選ばない（自分達で解決できるものを選ばないと意味がない）

【成果物イメージ】

- 問題原因分析図に記載する主な記載項目、イメージを表 7.5、図 7.10 に示す。

表 7.5 問題原因分析図の主な記載項目

分類	項番	記述内容	記述内容の説明
1. 書誌情報	1.1	問題原因分析図の ID	問題原因分析図を一意に識別するためのコード
	1.2	問題原因分析図の名称	問題原因分析図の名称
	1.3	概要	問題原因分析図の範囲、目的を説明する
2. 構成要素	2.1	問題	問題を端的に表現した文章
	2.2	因果関係	影響系、原因系の因果関係を紐付ける
	2.3	解決ポイント	解決すると効果的な問題

7.3 重要な要求を漏らさず、正確に記述する(DD.7.3)

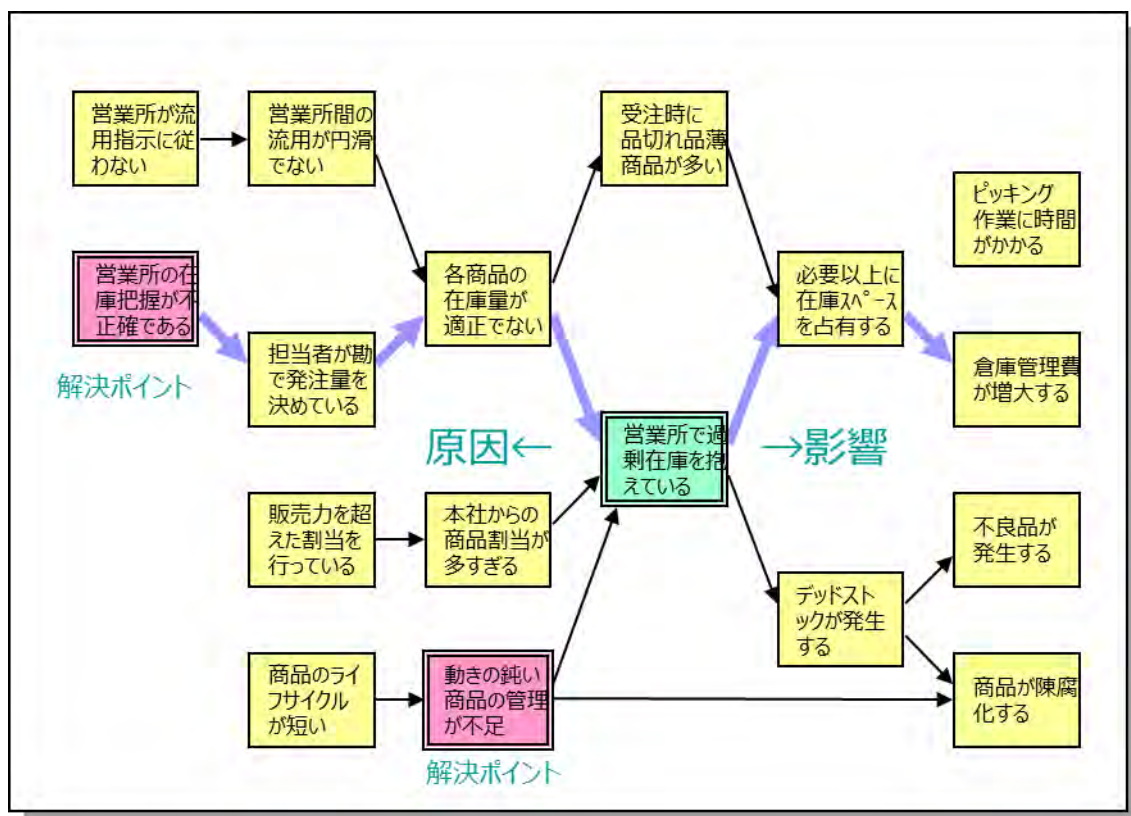


図 7.10 問題原因分析図のイメージ

7.3.3 要求構造図(DD.3.3)

【目的】

- 経営目的・業務目的に必要な業務の実現手段やシステム機能が明確になった状態にする。

【説明】

- 要求（目的/手段）を経営目的、経営施策、業務目的、実現手段、システム機能に区別して表現し、その関係を描く。

【留意事項】

- 階層を分けて表現する。
- AND 要求と OR 要求を区別する。
AND 要求と OR 要求の構造化のルール
 - 下位の要求が「目的」ならば、すべて AND 要求である（すなわち OR 要求はない）。
 - 下位の要求が「手段」ならば、AND 要求と OR 要求のいずれかである。
 - 記述を省略した場合は、AND 要求とする。
- 妥当性と十分性を検討し、要求を是正する。
- 人の手段と IT 手段の両方を検討する。

【成果物イメージ】

- 要求構造図に記載する主な記載項目、イメージを表 7.6、図 7.11 に示す。

表 7.6 要求構造図の主な記載項目

分類	項番	記述内容	記述内容の説明
1. 構成要素	1.1	No.	要求の通し番号
	1.2	要求種類	画面の名称
	1.3	要求	要求（経営目的、経営施策、業務目的、実現手段、システム機能）を端的に書いた文章
	1.4	関連	目的/手段の関係
	1.5	AND/OR 区分	AND 要求/OR 要求の区分
	1.6	備考	補足事項。自由記述

7.3 重要な要求を漏らさず、正確に記述する(DD.7.3)

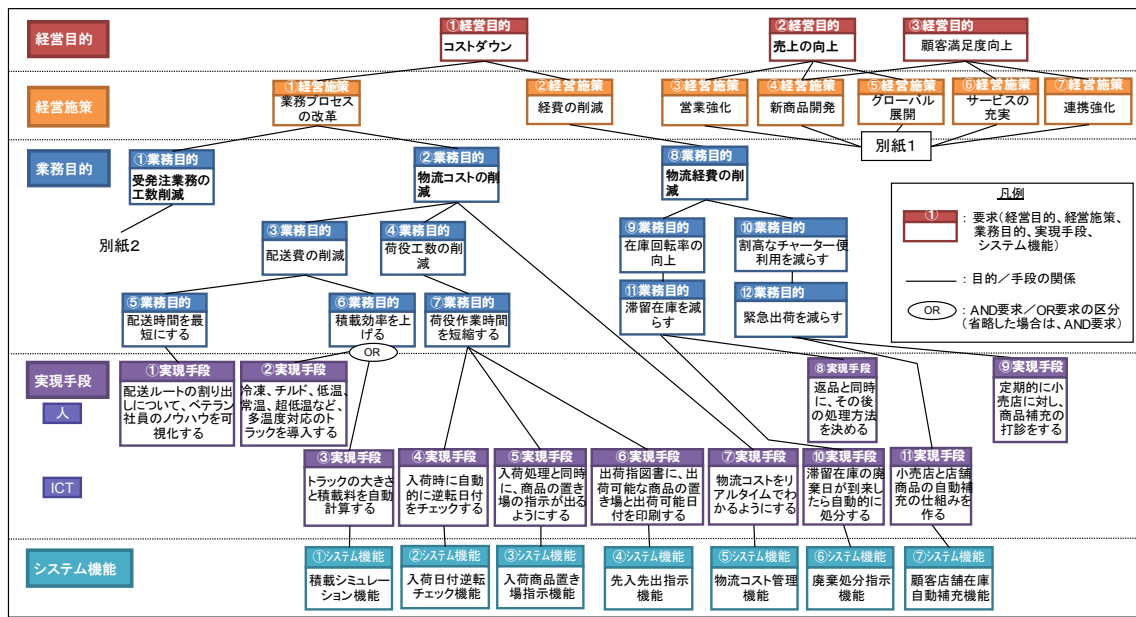


図 7.11 要求構造図のイメージ

7.3.4 要求一覧(DD.3.4)

【目的】

- 経営目的・業務目的を達成する手段を一覧化し管理可能な状態にする。
- 要件の消し込み・変更管理を行うためのベースラインを設定する
- 業務フローなど具体的業務やシステムと要求を照合することにより、業務フローなどに要求の反映漏れがない状態にする。

【説明】

- 要求、すなわち経営目的、経営施策、業務目的、実現手段、システム機能を一覧にしたもの
- 要求構造図で示した内容、および各要求の属性を設定する。

【留意事項】

- 要求すべてを一覧化し管理できるようにする。
- 目的の要求には、測定尺度、現状値、現状値の測定時期、目標値、目標値の達成時期を記載する。
- 手段系の要求には、評価指標（達成効果、費用、実現性）、優先度、新たな問題、予防措置を記載する。
- 膨らむ要求を抑えるために採用しなかった要求を明確にするための採否区分を記載する。
- 一覧の記載する属性をプロジェクトの状況に合わせてカスタマイズする。
- 提案書、RFP、業務フローなど他のドキュメントとの関連が分かるようになっているとより良い。
- 業務目的、実現手段を中心に記載すると書きやすい。

【成果物イメージ】

- 要求一覧に記載する主な記載項目、イメージを表 7.7、図 7.12 に示す。

7.3 重要な要求を漏らさず、正確に記述する(DD.7.3)

表 7.7 要求一覧の主な記載項目

分類	項番	記述内容	記述内容の説明
1. 構成要素	1.1	要求 ID.	経営目的、経営施策、業務目的、実現手段、システム機能別 ID を振る
	1.2	内容	要求を端的に表現した文章
	1.3	経営目的・経営施策の属性	<ul style="list-style-type: none"> ・ 目的：測定尺度、現状値、現状値の測定時期、目標値、目標値の達成時期を記載 ・ 手段：評価指標（達成効果、費用、実現性）を記載 ・ 元資料インデックス、備考を記載
	1.4	業務目的の属性	<ul style="list-style-type: none"> ・ 測定尺度、現状値、現状値の測定時期、目標値、目標値の達成時期、提起部門、責任部門、採否区分を記載 ・ 元資料インデックス、詳細資料インデックス、備考を記載
	1.5	実現手段の属性	<ul style="list-style-type: none"> OR かどうか、評価指標（達成効果、費用、実現性）、優先度、新たな問題、予防措置、担当部門、実現方法分類、進捗ステータス、採否区分を記載 ・ 元資料インデックス、詳細資料インデックス、備考を記載
	1.6	システム機能の属性	OR かどうか、詳細資料インデックス

7.3 重要な要求を漏らさず、正確に記述する(DD.7.3)

経営目的・経営施策				業務目的										実現手段						
要求ID	内容	元資料インデックス	備考	要求ID	内容	測定尺度	現状値	測定時期	目標値	達成時期	提起部門	責任部門	採否区分	元資料インデックス	詳細資料インデックス	備考	OR	ID	内容	
A01	オペレーションコストを削減する 医療事務コスト:2020年10%削減		中期経営計画	C01	フィルム取扱業務(撮影・保管・検索)のオペレーションコストを削減する	撮影・保管・検索業務時間	計80h/日	2010年度	計60h/日	2013年度	内科診療	内科診療	○						D01	画像情報を電子化する(フィルムレス運用)
A03	購買などの費用を削減する 購買費:2020年10%削減		中期経営計画	C04	フィルムの使用・廃棄をなくす	購入費用・廃棄費用	・400万円 ・30万円	2010年度	・0円 ・0円	2013年度	内科診療	業務部	○							
A04	環境貢献		経営ビジョン			廃棄量	2000枚	2010年度	廃棄ゼロ	2013年度										
A03	購買などの費用を削減する		-	C05	保管スペースを削減する	保管費用	60万円	2010年度		2013年度	管理部	管理部								
																			D01-01	医療画像PKGを導入(画像)
																			D01-02	PKGに合わせた業務プロセス(画像)

続く

実現手段														システム機能				
要求ID	内容	評価指標	優先度	新たな問題	予防措置	担当部門	実現方法分類	進捗ステータス	採否区分	元資料インデックス	詳細資料インデックス	備考	OR	システム機能ID	内容	詳細資料インデックス	備考	
		達成効果	費用	実現性														
		20h/日削減																
		430万円/年削減																
		2000枚/年削減																
		60万円/年削減																
D01-01	医療画像情報PKGを導入する(画像)		初期費用3000万円	○	大	企画室	IT(PKG)	○	提案書					MBE05	PKG画像処理	PKG説明書		
														MBE02	PKG画像診断			
														MBE01	PKG撮影			
D01-02	PKGに合わせた業務プロセスとする(画像)		200万円	○	大	内科診療	人	○										

続き

図 7.12 要求一覧のイメージ

7.4 データ構造を漏れなく、正確に記述する (DD.4)

業務を表現するデータ構造系のモデルとして「管理対象分類図」と「概念データモデル」を説明する。概念データモデルはデータベース設計まで続く成長型のドキュメントであるが、ここではその取り掛かりとして業務を表現するモデルとして説明する。データ構造モデルは業務のルールやビジネス構造など実世界を写像したものである。As-Is のデータ系のモデルを作成することで、現状のビジネスや業務の課題が浮き彫りにすることができる。To-Be のデータ系のモデルを作成することで、新しいビジネスや業務の姿を表現することができる。

また、本モデルでデータ構造の側面から、業務の理解や共通認識をはかることができる。機能系のモデルより少ない情報量（論理的に1枚の絵）で業務の全体が理解できる。

7.4.1 管理対象分類図(DD.4.1)

【目的】

- 管理対象のバリエーションを明確にして共通認識をはかる
- 同じ意味の用語を統一し（禁止用語を明確にする）齟齬を低減させる
- 用語が定義されていない管理対象をなくし、齟齬を低減する

【説明】

- 管理対象をミッシー(MECE)になるように分類し定義したもの

【留意事項】

- 概念データモデルのエンティティに相当する用語（管理対象）を抽出する。
- 抽出した用語の集合関係を見極め分類図として表現する。
- 分類はミッシー(MECE)になるようにする。
- 用語が定義されていない管理対象に用語を付ける。
- 禁止用語を明確にする。

【成果物イメージ】

- 管理対象分類図に記載する主な記載項目、イメージを表 7.8、図 7.13 に示す。

表 7.8 管理対象分類図の主な記載項目

分類	項番	記述内容	記述内容の説明
1. 書誌情報	1.1	管理対象分類図の ID	管理対象分類図を一意に識別するためのコード
	1.2	管理対象分類図の名称	管理対象分類図の名称
	1.3	概要	管理対象分類図の範囲、目的を説明する
2. 構成要素	2.1	管理対象名	管理対象の名称 管理対象の識別子
	2.2	分類観点	分類される条件や理由
	2.3	補足コメント	管理対象の説明 同一用語（禁止用語）の明示 不足用語の明示
	2.9	備考	状態、特記事項など自由記述

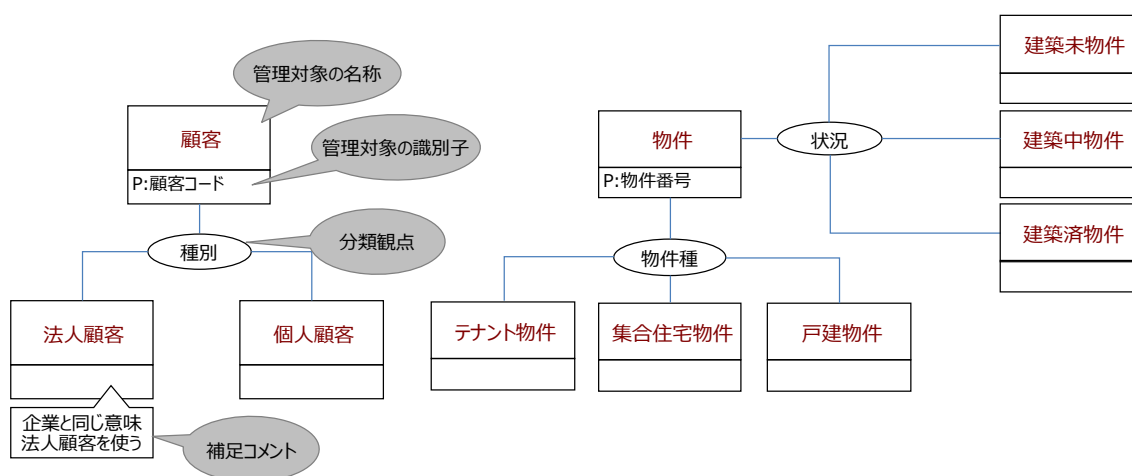


図 7.13 管理対象分類図のイメージ

7.4 データ構造を漏れなく、正確に記述する(DD.4)

7.4.2 概念データモデル (ER 図) (DD.4.2)

【目的】

- As-Is モデル作成の場合、業務をデータ構造の観点から表現し、現状のデータ構造上の問題点を発見するとともに、現状の業務や課題を共通認識できる状態にする。
- To-Be モデル作成の場合、新しい業務をデータ構造の観点から表現し、共通理解できる状態にする。

【説明】

- 概念データモデル (ER 図) は、受注、顧客、商品といった業務を構成する「もの」や「こと」であるエンティティ (Entity)、エンティティ同士の関連 (Relationship)、およびエンティティを構成するデータ項目 (Attribute) を図示したものであり、業務の実態 (業務で扱うデータ構造) を表現したものである。
- 従来の考え方では、ER 図はデータベース設計のための手法として基本設計以降に登場するが、要件定義段階で概念レベルの ER 図を導入して業務の実態を把握しておくこと、基本設計段階で要件定義工程にさかのぼることを少なくすることができる。
- カージナリティ (多重度) の表現 (IE 表記による)
図 7.14 に示すように、カージナリティはエンティティ A という集合とエンティティ B という集合の要素 (インスタンス) 間の関係を示している。

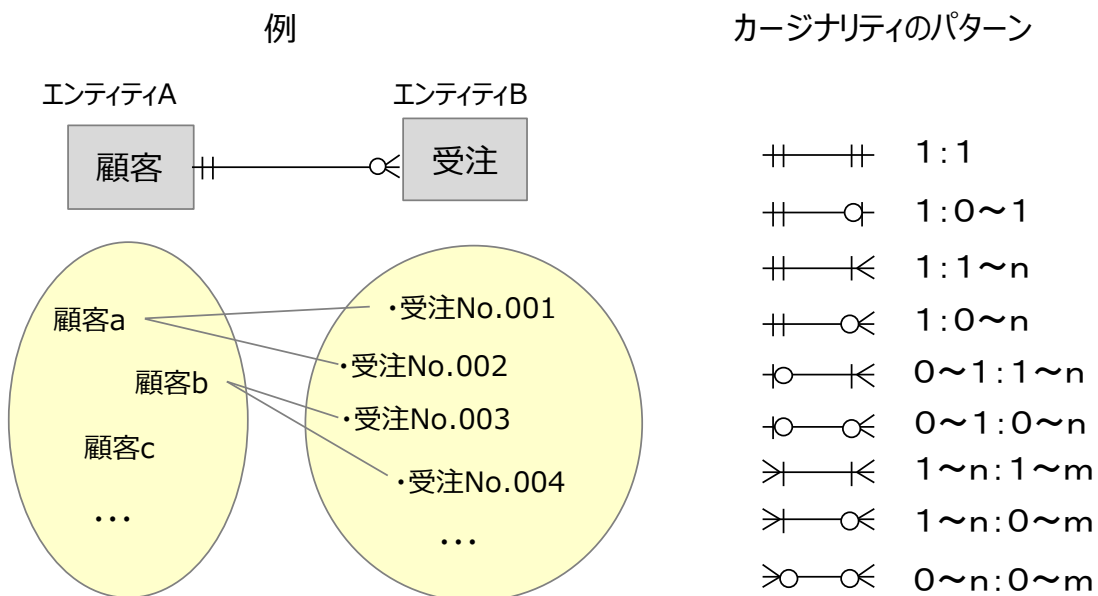


図 7.14 カージナリティ (多重度)

7.4 データ構造を漏れなく、正確に記述する(DD.4)

● 概念データモデルからの業務の読み取り方

- 分類を読み取る

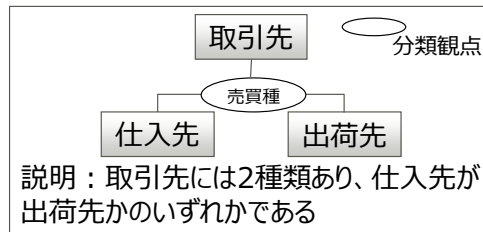


図 7.15 分類からの業務の読み取り方

- カージナリティ (多重度) を読み取る

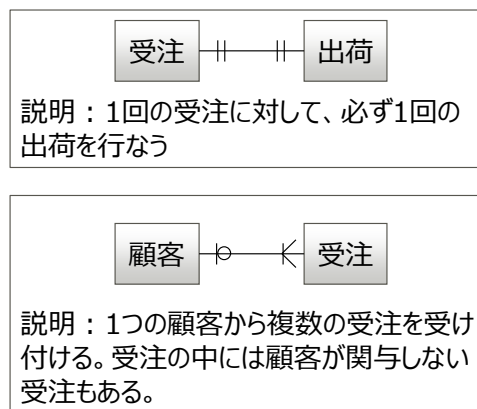


図 7.16 カージナリティ (多重度) からの業務の読み取り方

よって、図 7.17 に示すように業務が異なれば概念データモデルも異なる。

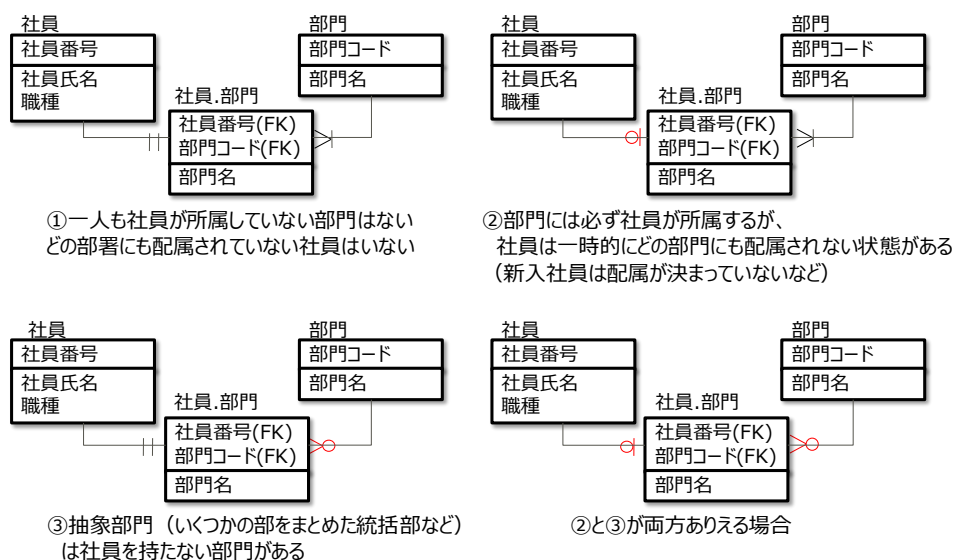


図 7.17 業務の違いは概念データモデルの違いの例

7.4 データ構造を漏れなく、正確に記述する(DD.4)

【留意事項】

- 概念データモデルを描く際には、業務ルールや業務実態のヒアリング、および既存の画面、帳票やドキュメント（業務マニュアル、システム設計書等）の収集を行う。そこから収集した情報をもとに、ER図を描くことで、業務の管理対象や業務ルール等の事実関係を確認していく。
- As-Is の概念データモデル作成の目的は、現状の業務の問題や課題を発見し、モデル上のエンティティ、関連と対応付けることにより業務を理解することである。
- To-Be 概念データモデル作成の目的は、新しいモデルと施策を関係付けることにより、業務プロセス改革内容の合意を図ることである。
- 作成順は As-Is 概念データモデルからである。概念データモデルのエンティティは業務フローとは異なり安定しており、フローほど大きく変わらない特徴がある。したがって、As-Is 概念データモデルを作成した上で、それをもとに To-Be 概念データモデルとして修正する手順が効率的である。また、概念データモデルは、短時間で正確に業務を理解するのに適したモデルであることから、業務を理解して共通認識を得る意味で As-Is 概念データモデルから作成すると良い。
- 現状を表す概念データモデルは業務の実態、事実に忠実に描く。エンティティを描く場合は、実業務でそのデータが管理対象として運営されていることを確認した上で描く。関連についても確認したものだけを描く。過去の経験からこうあるはずだと描いてしまうと、事実とは違う都合の良いものができてしまう。
- 概念データモデルの記載方法は業務部門にも説明し、両方で理解を確認する。概念データモデルの作成は一般的にはシステム部門の担当になるが、内容確認は業務部門でも実施できるよう記載方法を覚えてもらう。
- 現在の業務実態にはなくても To-Be として必要ないかという観点でも検証する。現在は存在しないエンティティとエンティティの関連を検討することにより、業務担当者に気づきを与えることができ、要件提示の漏れの防止につながる。
- 具体的なデータを入れて検証してみる。個々のデータ項目に具体的な値を入れて検証すると、抽象的な図では気づかなかった事項が明らかになることが多い。
- 概念データモデルは、企画プロセスやシステム要件定義プロセスなど複数のプロセスにまたがって作成されることから、どのプロセスでどこまで作成するか、各プロセスでの概念データモデルの完成度を事前に決めておく必要がある。業務モデルとして合意形成するなら、表 7.9 にあるように、属性であるデータ項目は主要なもの、すなわちエンティティを説明するのに必要なものだけでも構わない。

7.4 データ構造を漏れなく、正確に記述する(DD.4)

表 7.9 各プロセスでの概念データモデルの記載範囲の例（再掲）

	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
エンティティ	主要エンティティ	全エンティティ	全エンティティ
データ項目	主要データ項目	主要データ項目	全データ項目

【成果物イメージ】

- 概念データモデルに記載する主な記載項目、イメージを表 7.10、図 7.18 に示す。

表 7.10 概念データモデルの主な記載項目

分類	項番	記述内容	記述内容の説明
1. 書誌情報	1.1	概念データモデルの ID	概念データモデルを一意に識別するためのコード
	1.2	概念データモデルの名称	概念データモデルの名称
	1.3	概要	概念データモデルの範囲、目的を説明する
2. 構成要素	2.1	エンティティ名	エンティティの名称
	2.2	識別子	エンティティを一意に識別するもの
		属性	エンティティに従属するデータ項目
	2.3	リレーション名	リレーションの名称
		カージナリティ	エンティティ集合間の対応関係 (1:1/1:n/n:m)
		オプションリティ	対応する集合量 (0 あり/最低でも 1、平均、最大)
	2.4	分類観点	エンティティに分類関係があるときの分類の条件や理由
	2.6	補足説明	エンティティやリレーションの補足説明 (自由記述)
2.9	備考	特記事項など自由記述	

7.4 データ構造を漏れなく、正確に記述する(DD.4)

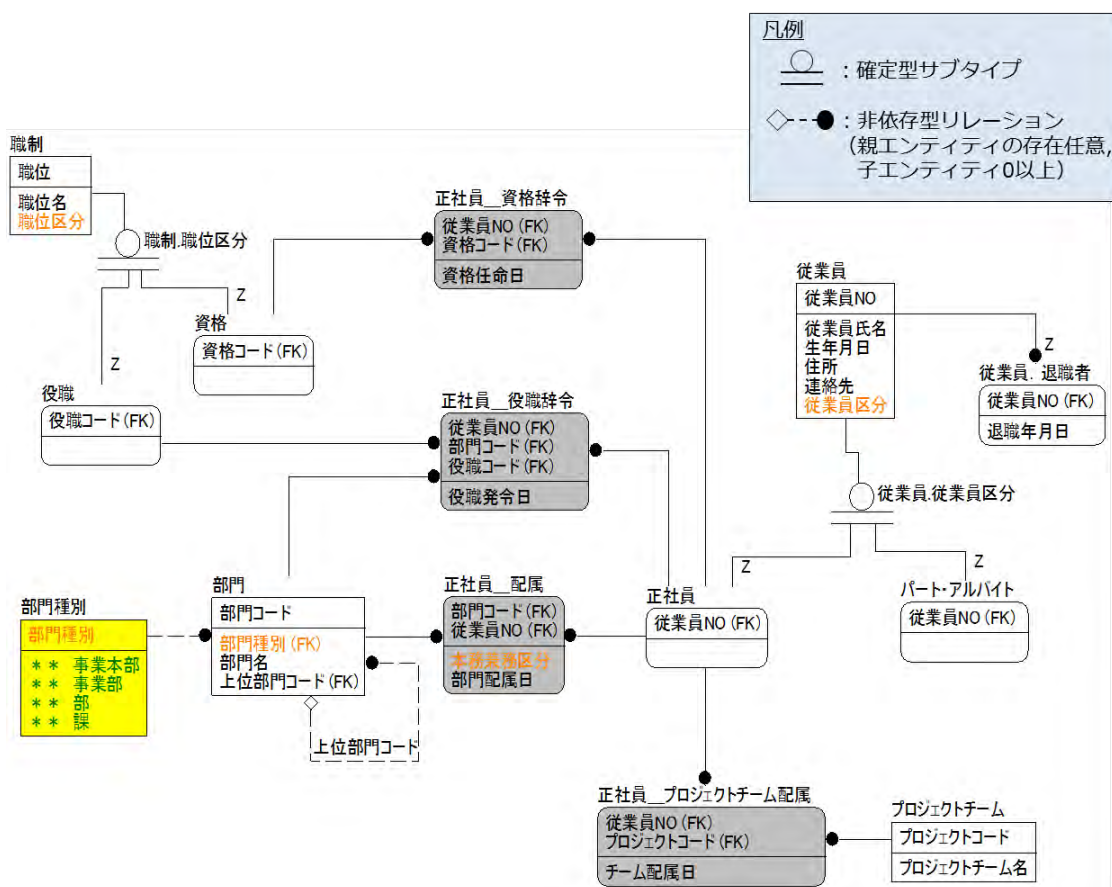


図 7.18 概念データモデルのイメージ

7.5 ビジネスプロセスを漏れなく、正確に記述する(DD.5)

業務を表現するドキュメントを以下に提示する。内容は、フロー系のモデル（フロー図）として「ビジネスプロセス関連図」、「ビジネスプロセスフロー（業務フロー）」、「ビジネスプロセスフロー（システム化業務フロー）」、業務の処理内容を記述するプロセスの詳細を記述する定義書として「業務処理定義」、一覧化し全体像を把握するための定義書として「業務機能構成表」の5件である。これ以外に、プロセスの詳細を記述する定義書として「システム化要求仕様」もあるが、これはシステム化プロセスの詳細であるのでシステム化要求定義（SR）で提示する。各ドキュメント相互の関連を図 7.19 に示す。

As-Is モデルを作成することで、現状のビジネスや業務の課題を浮き彫りにしたり、To-Be モデルを作成することで、新しいビジネスや業務の姿を表現したりすることができる。フロー系のモデルは、ある程度慣れ親しんだモデルであり、業務部門でも業務の理解や共通認識を容易にすることができる。一般的にデータ系のモデルより情報量が多くなるので、作成や理解には時間を要する。

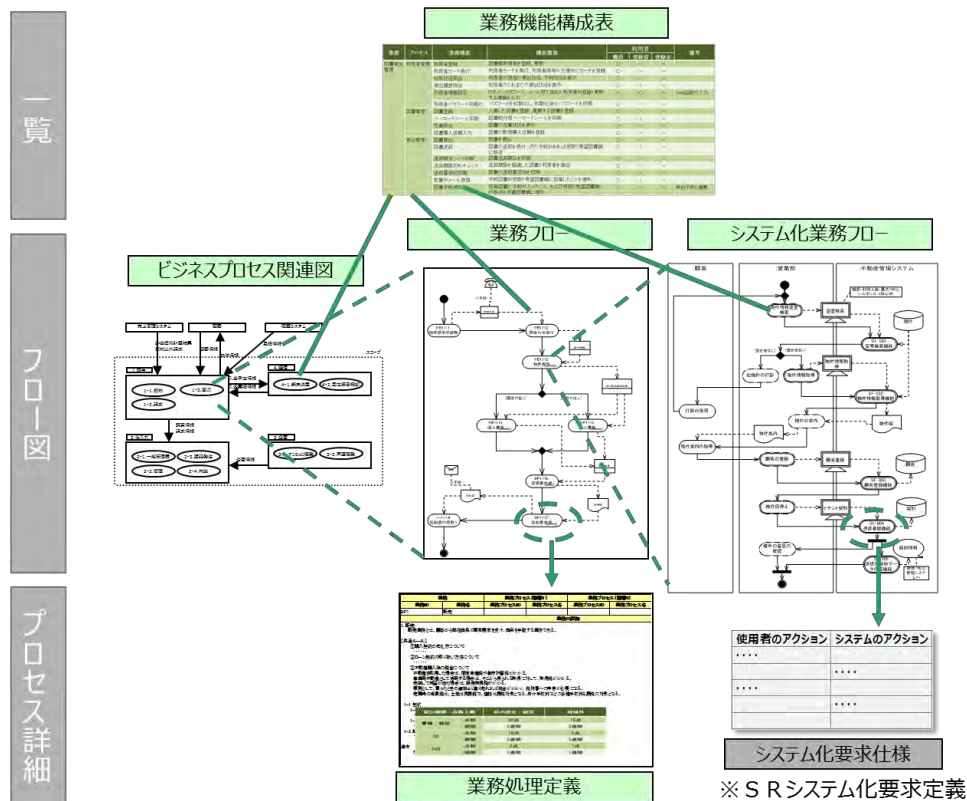


図 7.19 ビジネスプロセス系ドキュメントの関係

7.5 ビジネスプロセスを漏れなく、正確に記述する(DD.5)

コラム 10 フロー系成果物のプロセスの粒度を決める

【悩み】

業務フローを作成する時に、書く人によって、大まかになったり、やたら細かくなったりする。プロセスの粒度を定義しようとする定義自体が難しいし、定義に従うのも難しい。業務フローのプロセスの粒度を揃えることに苦勞する。

【解決案】

粒度を定義し成果物と対応付けて書き方として提示することが重要である。

(1) 業務上のプロセスを定義する

図 7.20 にビジネスプロセスやシステム機能などの定義例を示す。

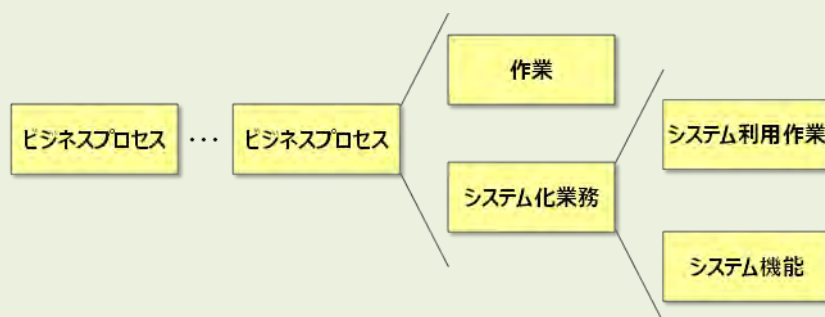


図 7.20 ビジネスプロセスやシステム機能などの粒度の定義例

- 「作業」とは人手によって実施するプロセスを意味する
- 「システム化業務」とは IT システムを利用して実施するプロセスを意味する
- 「システム化業務」は、IT システムを利用する人手の「システム利用作業」とその人に対して IT システムが提供する「システム機能」の2つで構成される
- 「ビジネスプロセス」とは、「作業」や「システム化業務」をまとめた業務そのもののプロセスである。ビジネスプロセスには粒度の大きなプロセスや小さなプロセスがあり、それらが階層化される

(2) どの階層のプロセスをどの成果物に記載するかを決定する

一つの例であるが、業務上のプロセスは「ビジネスプロセス関連図」「業務フロー」「システム化業務フロー」に3階層で記載することに決める。業務の規模や複雑さによっては、さらに業務フローを2階層や4階層に分けても良い。ここでは、3階層での定義例を示す。

この3種類のドキュメントとプロセスの定義の関係を図 7.21 に示す。

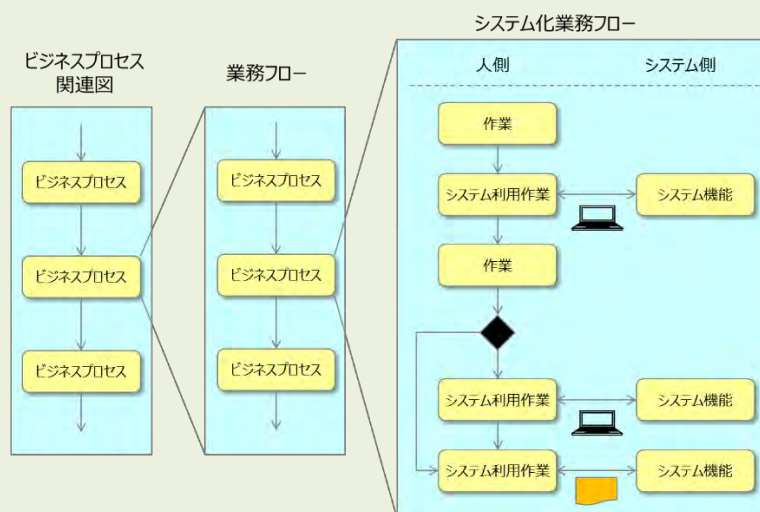


図 7.21 3種類のドキュメントとプロセスの定義の関係

(3) 作成ルールを提示する

「ビジネスプロセス関連図」と「システム化業務フロー」はドキュメント記載の粒度を提示するより、以下のルールを提示した方が、作成者による粒度のバラつきを抑えることができる。

● ビジネスプロセス関連図

ルール：1枚に全体を記載すること、鳥瞰できること、目次になること。

● システム化業務フロー

ルール：「システム機能」を業務に提供するサービスと想定し、ITシステムを利用した業務のフローを「システム機能」が「システム利用作業」と1:1で対応するように記載すること。

記載単位は、業務フローと同じ単位（ビジネスプロセス）で記載しても良い（同じ業務の詳細版）し、業務フローの1ビジネスプロセスのブレイクダウン（構造化表現）として記載しても良い。

(4) 業務フローは、粒度の考え方を提示して浸透させる

● 業務フロー

ルール：業務で何をするのかを記載する（HowではなくWhatを書く）。

「作業」と「システム化業務」を区別せずに記載する。

ビジネスプロセスの改革、改善を意識して記載する。

このように、プロセスを3種類のフロー図で階層化し、各成果物の記載の方法を提示することで、プロセスの粒度に悩まずに作成できるようになる。

7.5 ビジネスプロセスを漏れなく、正確に記述する(DD.5)

7.5.1 ビジネスプロセス関連図(DD.5.1)

【目的】

- 対象とする業務の全体概要を把握できる状態にする。
- プロジェクトの範囲が明確にされた状態にする。
- 関係する外部組織・システムの関連について明確にされた状態にする。
- 大きなビジネスプロセスの変更（改革）内容が把握できる状態にする。

【説明】

- フロー系モデルであり、プロセスとその間での情報のやり取り/順序を中心に描いたものである。
- ビジネスプロセス関連図は、今回の対象範囲（範囲）を明確に示し、範囲内のビジネスプロセス、範囲外との関係が明確になるようにし、全体を鳥瞰できるようにしたフロー図である。
- ビジネスプロセス関連図には、情報・物品・資金の流れの各プロセスで実施されている抽出・集約・加工・移送・記録の結果が、他のプロセスとどう関連しているのかを記載する。ビジネスプロセス関連図に上記を記載し、図上の各プロセスが果たす目的を理解すれば、ビジネスの全体像を把握することができる。
- 三種類のフロー図の中で一番大きなレベルになる。三階層で業務を表現したときの一階層目を中心に、二階層目まで表現したものであり、このレベルで To-Be が変わるのは大きなビジネス改革がある場合である。その場合は、ビジネスプロセス関連図上で変革を確認する。
- このレベルで変更がない場合は、全体像の把握、範囲に含まれる業務プロセス、範囲外とのやり取り（外部インターフェース）を確認するのが主な目的になる。

【留意事項】

- 1枚で書く（全体を1枚で表現することにより全体が把握しやすくなる）。
- ビジネスプロセス関連図を作成したら、それをもとに全体最適を考え、組織全体のシステム化計画との整合性を確保することが必要である。
- 関連する部門や外部組織のどこまでをシステム化の対象とするか、全体最適を視野に入れているかなど範囲の妥当性を確認する。
- 範囲が明確になったら、範囲外との情報の受け渡し（外部インターフェース）があるかを洗い出す。外部インターフェースが制約になることがあるので漏れなく洗い出し、一覧で管理して取り扱いを明確にしていく（SR. システム要件定義）。
- 範囲内に記載するビジネスプロセスの粒度は無理に揃えようとせず、1枚に書ける大きさと記載するぐらいの方が悩まない。詳細は以降の業務フロー上で明確にして

7.5 ビジネスプロセスを漏れなく、正確に記述する(DD.5)

いく。

- ただし、粒度はバラバラでも漏れていてはならない。ここで記載したビジネスプロセスをもとに、個々に詳細化を進めていく。

【成果物イメージ】

- ビジネスプロセス関連図に記載する主な記載項目、イメージを表 7.11、図 7.22 に示す。

表 7.11 ビジネスプロセス関連図の主な記載項目

分類	項番	記述内容	記述内容の説明
1. 書誌情報	1.1	ビジネスプロセス関連図の ID	ビジネスプロセス関連図を一意に識別するためのコード
	1.2	ビジネスプロセス関連図の名称	ビジネスプロセス関連図の名称
	1.3	概要	ビジネスプロセス関連図の範囲、目的を説明する
2. 構成要素	2.1	ビジネスプロセス	三階層で業務を表現したときの一階層目、二階層目
	2.2	処理の流れ	処理の順序を表現する
	2.3	情報の流れ	伝票や帳票、データなどの流れを表現する
	2.4	スコープ	要件定義で対象となるスコープを示した境界線
	2.5	他業務連携	スコープ外との関係を示す。他の業務の「業務」または「プロセス」
	2.6	組織など	組織、他企業、倉庫など自由記述
	2.7	補足説明	フロー図を分かりやすくするために、分岐、コメントなどを自由記述して良い
	2.8	備考	特記事項など自由記述

7.5 ビジネスプロセスを漏れなく、正確に記述する(DD.5)

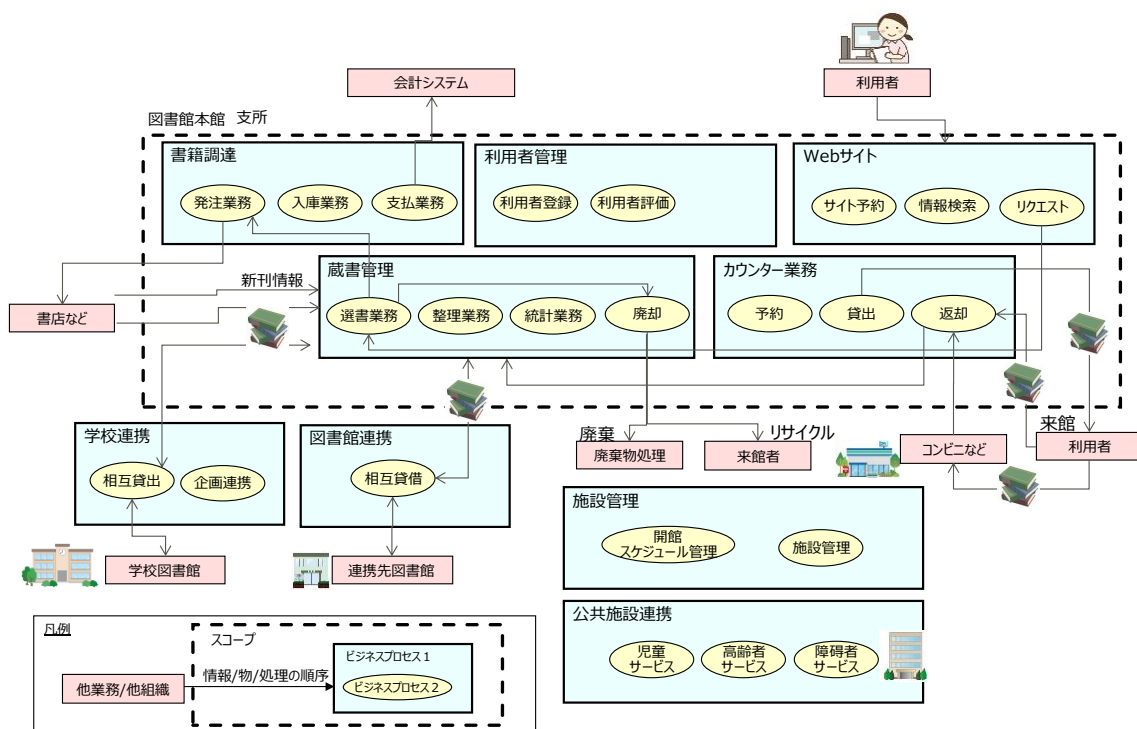


図 7.22 ビジネスプロセス関連図のイメージ

7.5.2 業務機能構成表(DD.5.2)

【目的】

- 業務の機能を一覧化し、全体が把握できるようにするとともに漏れがない状態にする。
- 対象の業務全体をまとまりのあるプロセス単位でグループ化し、プロセスごとにどのような業務機能が存在するのかを明確な状態にする。

【説明】

- 業務機能構成表はシステム化対象の業務機能を作業レベルまで分解し、各業務に含まれる機能を具体的に把握できるようにするための表であり、システムに含まれる業務機能の全体像を明らかにするために作成する。業務機能は、原則として3階層（大分類、中分類、小分類）程度に階層分けして展開するのが分かりやすい。
- 業務機能構成表の例を図 7.23 に示す。この例は、業務に含まれる機能を3段階の階層に分けて、機能ごとに、その業務に含まれる機能を取りまとめた表の例である。この表を参照すれば、システム化対象の業務に含まれる処理の全体像が把握できる。

【留意事項】

- 作成した業務フローやシステム化業務フローだけではすべての業務機能が記載されていない場合は、一覧で漏れなく洗い出す必要がある。
例) 業務フローでは、To-Be として変わる部分しか記載せず、現行と同じ部分の業務フローを省略した、など。
- また、業務機能構成表を先に作成してから業務フローを作成するパターンもある。この場合、まず業務機能構成表上に全体概要を表し、業務フローにより詳細化、検証し正確に作り上げていく流れになる。
- 業務機能の抽出は、システム化対象の業務だけでなく、人手で実施される業務についても漏れなく実施することが重要である。現在、人手で実施されている業務機能についても、システム化後も継続して人手で実施することがここに明示されていないと、システム化後に継続して人手で作業できるようにするための情報の提供が漏れるなど、システムの機能漏れが後で発見される原因になる。また、システム化対象外にした理由も明記しておくこと、システム化対象範囲の妥当性を再確認する際に効率化が図れる。

【成果物イメージ】

- 業務機能構成表に記載する主な記載項目、イメージを表 7.12、図 7.23 に示す。

7.5 ビジネスプロセスを漏れなく、正確に記述する(DD.5)

表 7.12 業務機能構成表の主な記載項目

分類	項番	記述内容	記述内容の説明
1. 書誌情報	1.1	業務機能構成表の ID	業務機能構成表を一意に識別するためのコード
	1.2	業務機能構成表の名称	業務機能構成表の名称
	1.3	概要	業務機能構成表の範囲、目的を説明する
2. 構成要素	2.1	ビジネスプロセス 1	三階層で業務を表現したときの一階層目
	2.2	ビジネスプロセス 2	三階層で業務を表現したときの二階層目
	2.3	作業/システム利用作業	三階層で業務を表現したときの三階層目
		業務機能概要	業務機能の概要を説明する
		利用者	利用者を記載する
	2.9	備考	状態、特記事項など自由記述

ビジネスプロセス 1	ビジネスプロセス 2	作業	システム利用作業	業務機能概要	利用者			備考	
					ID登録済	ID未登録	登録済		
LM11 資料利用管理									
LM113 資料利用者管理	LM1131 資料利用者登録	資料利用者を登録	LM1131- 利用者情報登録 SF-001	資料貸出を希望する利用者を本人確認の上で登録	○				
	LM1132 利用者情報登録	利用者がパスワード、連絡先を登録	LM1132- 利用者登録更新 SF-001	Webのログインパスワードや予約資料の貸出可能メールの送付先などを利用者本人が設定		○		Web公開	
	LM1133 利用者登録更新	利用登録を更新	LM1133- 利用者登録更新 SF-001	住所等連絡先、利用期限を本人確認の上で更新	○				
LM114 資料貸出予約	LM1141 資料照会	資料の有無、貸出状況を検索	LM1141- 資料検索 SF-001	所属する資料を検索	○	○	○	Web公開	
	LM1142 資料貸出予約	資料の貸出を予約	LM1142- 貸出予約 SF-001	資料を貸出予約	○	○		Web公開	
	LM1143 貸出予約取消	貸出予約を取消	LM1143- 貸出予約取消 SF-001	自身の貸出予約を取消	○	○		Web公開	
	LM1144 貸出期間延長	貸出期間を延長申請	LM1144- 貸出期間延長 SF-001	他の利用者に予約がない場合に貸出期間を延長	○	○		Web公開	
LM117 資料貸出	LM1171 資料利用者確認	利用者カードにより本人確認	LM1171- 利用者カード読み込み SF-001	利用者カードにより本人確認あわせて利用期限、貸出中資料点数、返却遅延有無、予約資料の有無など、以降の処理に必要な情報を表示	○				
LM117 資料貸出	LM1171 資料利用者確認	利用者カード以外の方法で本人確認	LM1171- 利用者情報参照 SF-002	利用者カードを携帯していない場合に、自動車運転免許証などの公的証明書と自宅の電話番号で本人確認	○				
LM117 資料貸出	LM1171 資料利用者確認	利用者カードの期限を更新	LM1133- 利用者登録更新 SF-001	上記参照	○			資料利用者管理の機能を使用	
	LM1172 予約資料取り出し	予約資料を取り出し		予約資料がある場合に、予約資料欄から資料を取り出し	○				
	LM1173 資料貸出	予約資料の提示			予約資料がある場合に、確認のために予約資料を利用者に提示	○			
		付帯資料の有無確認			付帯資料がそろっているか確認	○			
		欠落の資料への記録			返却時のトラブル防止のために不足があることを貸出資料に記載	○			
	欠落のシステムへの登録		LM1173- 資料状態登録 SF-001	資料検索時に表示するために内容をシステムに記録	○				
	資料貸出の入力		LM1173- 資料貸出登録 SF-002	資料の貸出をシステムに登録	○				
	貸出入力終了		LM1173- 返却期限印刷 SF-003	資料の返却期限を記載したシートを印刷	○				
	資料の貸出			貸出登録済の資料を返却期限記載シートとともに貸出	○				

図 7.23 業務機能構成表の成果物イメージ

7.5.3 ビジネスプロセスフロー（業務フロー）(DD.5.3)

【目的】

- 新しいビジネスプロセスを創造し、To-Be 業務フローとして表すことにより、新たなビジネスの流れを共通認識できる状態にする。
- 必要に応じて As-Is 業務フローを作成し、現状の業務ビジネスの流れを明示することにより、現状の業務ビジネスフローの問題点や課題を共通認識できる状態にする。

【説明】

- ビジネスプロセスフロー（業務フロー）（以下、業務フロー）はフロー系モデルであり、プロセスとその間での情報のやり取り/順序を中心に描いたものである。
- 業務フローは、三階層で業務を表現したときの二階層目を中心に業務を表現したものである。手段であるシステム利用を表現していないレベルであり、業務の根本的な機能/目的が分かるようにすることを目的として作成する。
- 根本的な業務を表現することができるので、As-Is 業務フローは現状業務の本質的課題の発見に、To-Be 業務フローは新しい業務としてビジネスプロセス自体を見直そうとしたときに役立つ。

【留意事項】

- As-Is と To-Be の両面から業務を分析する。As-Is と To-Be の作成目的が異なるため、それぞれの目的を意識して作成することが望ましい。As-Is 業務フロー作成の目的は、現状の業務の流れをフロー図に書き出すことにより、現状の業務の流れを明確にし、そこに潜む問題点や課題を抽出することである。これに対して、To-Be 業務フロー作成の目的は、新しい業務プロセスとそれを実現するために必要になる業務機能に対応付けることにより、業務プロセス改革内容の整合性を確認し、実施内容の合意を図ることである。
- To-Be 業務フロー作成にあたっては、業務プロセスの改革を行うという意識を持ち、現状追従にならないように注意することが重要である。その際、対象業務の目的、期待する効果を記載しておくこと、それをもとにシステム化の目的を随時再確認することができ、後続工程で議論が後戻りすることを予防できる。
- 5W2H に注意して要件を記載すると、要件がより具体的に表現され、漏れや不整合の防止につながる。確認ポイントを表 7.13 に示す。

7.5 ビジネスプロセスを漏れなく、正確に記述する(DD.5)

表 7.13 業務フロー作成における確認ポイント

5W2H	確認ポイント
What	・商品やサービス(What)による違いはないか
Who	・相手(Who)による違いはないか
When	・時間、時期、タイミング、順序など(When)による違いはないか ・状況や状態、条件による違いはないか
Where	・場所(Where)による違いはないか
Why	・理由(Why)による違いはないか
How	・やり方や方式(How)による違いはないか
How many (How much)	・量(How many)や金額(How Much)による違いはないか

- まず、各業務機能における通常処理（月次の業務量の大半（例えば95%以上）を占める業務パターンなど）の業務フローを記載する。通常処理における取消、変更（一部変更、一部取消などは、業務シナリオによっては例外処理扱いにする）は、通常処理に含めて記載する。
- 例外処理を抽出する。例外処理とは、ある処理プロセスにおいて、途中で何らかのエラーを検知した際に、通常処理を中断（中止）して別の処理を行うことであり、その際に発生したエラーのことを例外と呼ぶ。業務プロセス上のある処理を実行している際に、処理を継続できないようなエラー（貸し出し状況を確認したら新規に図書の貸し出しを許可できない状態であった等）が発生した際に実施する別処理のことを、本書では例外処理と表現する。上記のような例外処理を漏れなく洗い出すことは業務フロー作成における重要目的であり、その精度が以降のシステム開発の品質に大きく影響する。そのため、例外処理として実施される業務処理の洗い出しには、漏れを極力排除する努力が必要である。洗い出しの際には、抽出した各例外処理の発生頻度や影響度、重要度も把握する。
- 特殊処理を抽出する。特殊処理とは、イベント、キャンペーンなどの不定期に実行される業務処理を本書では指す。この特殊処理も業務プロセスとしてすべて抽出する。
- 例外処理、特殊処理の抽出は、通常処理に比べて漏れやすい。そのため、入力ミス、決算またぎ、責任者不在、緊急時の事務、サービス時間外の対処など、重点的に確認すべきポイントを事前に洗い出した上で、業務担当者からのヒアリングの実施、抽出結果に対する別担当者による再確認、現場で実施されている操作の実見などにより抽出を行う。不定期の処理、イベントやキャンペーンなどにも注意する。
- 業務処理の組み合わせにおける抽出漏れを確認する。複数の処理パターンの組み合わせで処理が分類される業務処理（例えば、特定の会員種別で、特定のカテゴリの商品を購入した場合など）において、組み合わせによる処理の相違がないかについて、社

7.5 ビジネスプロセスを漏れなく、正確に記述する(DD.5)

内の業務部門や、可能なら顧客にも協力してもらい確認する。

- 分岐が発生する場合、それぞれの発生頻度（滅多に発生しない事象かどうか）、作業負荷、時刻・季節などにより発生頻度に変動がある場合にはその分布を確認する。これを実施すると、システム化対象に取り込むかどうかを決める際に役立つ。分岐発生の開始条件、終了条件などがシステム化の際に重要な意味を持つ分岐は、その条件を明確にする。
- 担当部署（アクター）は、具体的な組織名ではなく、役割（ロール）で記載する。
- 通常処理の処理量とともに、例外処理、特殊処理の発生頻度などは、漏れることがないように非機能要件定義の処理データ量の箇所に記載しておく。
- 業務が複雑な場合には、パターンごとに記載したり、さらなる階層化をして記載したりすることにより、見やすさを確保することも必要になる。
- 組織間でのやり取り、分担を明確にしたいときなどは、スイムレーンなどを使って分かりやすく表現する。
- 業務フローの記述方法には、BPMN（Business Process Modeling Notation）、フローチャート、DFD(Data Flow Diagram)、UML（Unified Modeling Language）のアクティビティ図、PFD（Process Flow Diagram）などがある。プロジェクトに適したものを選択して欲しい。

【成果物イメージ】

- 業務フローに記載する主な記載項目、イメージを表 7.14、図 7.24 に示す。

7.5 ビジネスプロセスを漏れなく、正確に記述する(DD.5)

表 7.14 業務フローの主な記載事項

分類	項番	記述内容	記述内容の説明
1. 書誌情報	1.1	業務フローの ID	業務フローを一意に識別するためのコード
	1.2	業務フローの名称	業務フローの名称
	1.3	概要	業務フローの範囲、目的を説明する
2. 構成要素	2.1	ビジネスプロセス	三階層で業務を表現したときの二階層目を中心に記載する 必要に応じて一階層目を記述しても良い
	2.2	情報の流れ	データの流れ
	2.3	順序	ビジネスプロセス間の遷移
	2.4	合流/分岐	ビジネスプロセスの分岐/合流、フォーク/ジョイン。分岐に対しては分岐条件を記載する
	2.5	補足説明	分かりやすくするために、スイムレーン、アイコン、コメントなどを自由記述して良い
	2.6	備考	特記事項など自由記述

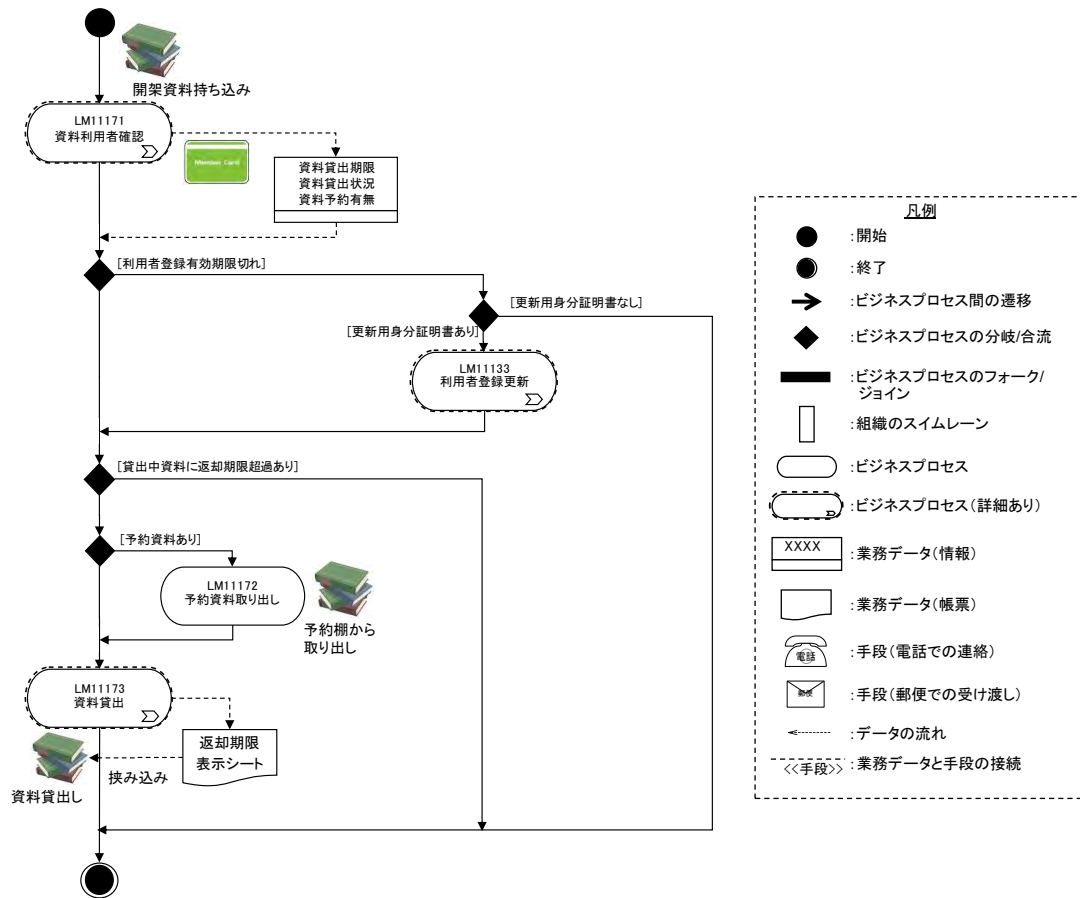


図 7.24 ビジネスプロセスフロー（業務フロー）のイメージ

7.5.4 ビジネスプロセスフロー（システム化業務フロー）(DD.5.4)

【目的】

- 新しいビジネスプロセスを創造し、To-Be システム化業務フローとして表すことにより、新たな業務の流れやシステム利用を共通認識できる状態にする。
- 必要に応じて As-Is システム化業務フローを作成し、現状のシステムを利用した業務の流れを明示することにより、現状の業務やシステム利用の問題点や課題を共通認識できる状態にする。
- システムに対する業務からのリクエスト（作成して欲しいシステム機能）を明確にして、合意形成できる状態にする。

【説明】

- ビジネスプロセスフロー（システム化業務フロー）（以下、システム化業務フロー）はフロー系モデルであり、プロセスとその間での情報のやり取り/順序を中心に描いたものである。
- システム化業務フローは、三階層で業務を表現したときの三階層目で業務を表現したものである。システムを利用した業務を表現したものであり、システムを利用した具体的な業務を作りあげるために作成する。
- 基本的には業務フローで業務レベルでの改善を検討したのち、それをもとに人の作業とシステム機能を分けて記載して、システムを利用した改善すなわち、作成して欲しいシステム機能を洗い出す流れになる。
- 具体的なシステムを利用した業務を表現することができるので、As-Is システム化業務フローの作成では現状業務の課題発見に、To-Be システム化業務フローの作成では新しい業務としてビジネスプロセスの見直しに役立つ。特に、システム利用の問題点やシステムを利用した新たな業務、新たなシステム機能を明確にするのに役立つ。

【留意事項】

- As-Is と To-Be の両面からシステム化業務を分析する。As-Is と To-Be の作成目的が異なるため、それぞれの目的を意識して作成することが望ましい。As-Is システム化業務フロー作成の目的は、現状の業務の流れをフロー図に書き出すことにより、現状の業務の流れやシステム利用を明確にし、そこに潜む問題点や課題を抽出することである。これに対して、To-Be システム化業務フロー作成の目的は、新しい業務プロセスとそれを実現するために必要になるシステム機能を対応付けることにより、業務プロセス改革内容の整合性を確認し、実施内容の合意を図ることである。
- To-Be システム化業務フロー作成にあたっては、システムを利用した新しい業務を作り上げる意識を持ち、現状追従にならないように注意することが重要である。業務自

7.5 ビジネスプロセスを漏れなく、正確に記述する(DD.5)

体の改善は業務フローで実施しているので、その改善との整合性を意識しながら描く必要がある。

- 例外処理、特殊処理に対しては業務フロー同様に意識する必要がある。業務フローとの整合性を意識して進める。
- 例外処理や特殊処理は、投資対効果や業務変更への対応の柔軟性を考慮してシステム化するかどうかを判断する。
- システム化業務フロー作成にあたっては、システム部門の見識が必要な部分もあるので、両者が協力して作成することを勧める。

【成果物イメージ】

- システム化業務フローに記載する主な記載項目、イメージを表 7.15、図 7.25 に示す。

7.5 ビジネスプロセスを漏れなく、正確に記述する(DD.5)

表 7.15 システム化業務フローの主な記載項目

分類	項番	記述内容	記述内容の説明
1. 書誌情報	1.1	システム化業務フローの ID	システム化業務フローを一意に識別するためのコード
	1.2	システム化業務フローの名称	システム化業務フローの名称
	1.3	概要	システム化業務フローの範囲、目的を説明する
2. 構成要素	2.1	ビジネスプロセス	必要に応じて、三階層で業務を表現したときの一階層目、二階層目を記載しても良い
	2.2	作業	人が行うプロセス
	2.3	システム利用作業	システムを利用しながら業務を行うときの人側のプロセス
	2.4	システム機能	システムを利用しながら業務を行うときのシステム側の機能
	2.5	ユーザインターフェース	システムを利用しながら業務を行うときの人側のプロセスとシステム側の機能の間に発生するインターフェース。通常画面か帳票である
	2.6	業務データ	利用するエンティティ
	2.7	情報の流れ	情報の流れを矢印と名前を持って明確にする
	2.8	作業の流れ	プロセス間の順序性を表す
	2.9	合流／分岐	プロセスが分かれたり合流したりすることを表す。分岐に対しては分岐条件を記載する
	2.10	補足説明	フロー図を分かりやすくするために、コメントなどを自由記述して良い
	2.11	備考	特記事項など自由記述

7.5 ビジネスプロセスを漏れなく、正確に記述する(DD.5)

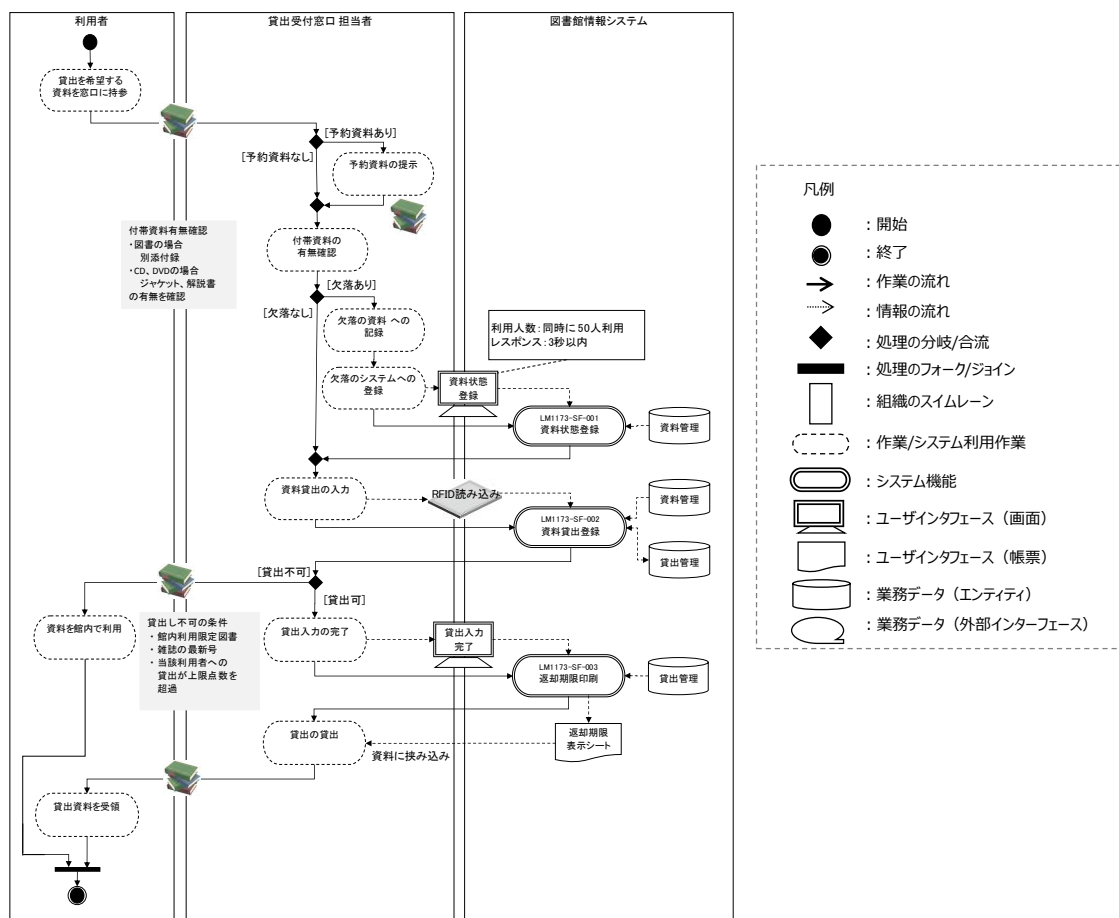


図 7.25 ビジネスプロセスフロー (システム化業務フロー) のイメージ

7.5.5 業務処理定義(DD.5.5)

【目的】

- フローでは記述しきれない業務処理の詳細内容を記述することにより、業務部門との共通認識や合意形成ができる状態にする。
- 業務ルールを明確にして、システム機能の仕様のベースラインを作り、システム開発関係者と共通認識できる状態にする。

【説明】

- 業務フロー上の各業務処理機能のうち、業務フローなどに記載しきれない、それ単独で記載した方が分かりやすい処理や判断を記載したもの。
- 業務フローは業務機能の流れを示した図であり、個々の業務機能におけるシステムの処理には言及していない。業務処理定義書は、業務フロー上の個々のボックス(業務機能)単位に作成され、業務機能とシステム機能を関係付ける成果物となる。

【留意事項】

- 業務の背景、目的、取り扱い範囲・制約事項を明確にしておく。ユーザのニーズや、特別な業務処理が存在する背景や理由を理解しておく、漏れや不整合を抽出しやすい。そのため、自明でない業務処理には理由を記載しておくことが望ましい。また、発生数や発生頻度の偏りなどがあれば、その理由も記載しておく。業務処理が存在する理由を書くことにより、その業務が何のための業務かを確認でき、要件定義漏れ(システムの実装仕様漏れ)の防止につながる。
- 発生・完了タイミングなど重要な時間や日付を確認する。業務処理が発生・完了するタイミングや、他の業務の発生との関連性などに留意すべき点があるケースは、業務の実施時間やタイミングを業務処理定義書に記載する。
- 業務上の重要なチェック処理も確認する。要件定義段階では、アプリケーションの処理詳細に関係する例外処理はチェック処理の洗い出し対象から除外しても良い。それとは反対に、業務系のエラーチェック(誤入力防止のために、購入数量が一定数量を上回っている場合には警告を発するなどの業務上のチェック処理)については、それらを取り込むことがシステム化の目的やニーズの重点になっている場合には、業務処理定義書に記載して、要件として明示する。
- 表形式にまとめると漏れや不整合を防止しやすい。業務仕様をまとめる際には、表形式で分類できるものは、表 7.16、表 7.17 に示すように表形式を積極的に活用することが望ましい。表 7.16、表 7.17 は図書館システムの会員の住居種別によるサービスの違いを示す。この表では図書の貸し出し上限と貸し出し期間を図書の種類と住居区分で分類して、まとめて表している。なお、この例では、貸し出し期間は住居区分が

7.5 ビジネスプロセスを漏れなく、正確に記述する(DD.5)

異なっても同じであるが、貸し出し点数と1つの表にしている例であり、本来は、特段の必要がない限り単純化することが望ましい。

表 7.16 図書館システム - 会員種別による貸し出し上限点数の違い

貸出点数上限	区内居住・就者	地域外
書籍・雑誌	30点	15点
CD	10点	5点
DVD	2点	1点

表 7.17 図書館システム - 複数の要素を1つの表にまとめた例

貸出期間・点数上限		区内居住・就労	地域外
書籍・雑誌	点数	30点	15点
	期間	2週間	2週間
CD	点数	10点	5点
	期間	2週間	2週間
DVD	点数	2点	1点
	期間	1週間	1週間

- 業務の要件定義において、業務担当者から「この部分は現行どおり」と言われることがある。しかし業務側も業務の詳細部分を把握していないことが多く、現行システムのドキュメントも最新でない場合も多い。「現行どおり」と言われても、要件定義においては明確な仕様に展開して要件を記述しなければならない。
- セキュリティ上の制約（例えば、顧客情報など特定の情報に対するアクセス制限、アクセスログの要否等）を確認する。IT 統制や業務監査の証跡として残す必要のある日付やデータなど留意すべきものがあれば、記載する。
- 定義された業務要件は、検討経緯を補足説明または参考資料として残しておく、後工程での要件定義結果の再確認に役立つ。

【成果物イメージ】

- 業務処理定義に記載する主な記載項目、イメージを表 7.18、図 7.26 に示す。

7.5 ビジネスプロセスを漏れなく、正確に記述する(DD.5)

表 7.18 業務処理定義の主な記載項目

分類	項番	記述内容	記述内容の説明
0. 共通情報	—	プロジェクト名、システム名、工程名、ドキュメント ID、ドキュメント名、作成者、作成日付、バージョン、更新者、更新日付	
1. 書誌情報	1.1	業務処理定義の ID	業務処理定義を一意に識別するためのコード
	1.2	業務処理定義の名称	業務処理定義の名称。
2. 構成要素	2.1	概要	業務処理定義の目的、範囲、制約などを説明する
	2.2	業務処理説明	自由記述 <ul style="list-style-type: none"> ・考え方（方針） ・業務説明 ・業務プロセス説明 ・前提条件 ・業務ルール(処理内容) ・例外業務 など 表などを用いて分かりやすく記載
	2.3	補足説明	フロー図を分かりやすくするために、コメントなどを自由記述して良い
	2.4	備考	特記事項など自由記述

業務		業務プロセス(階層1)		業務プロセス(階層2)		業務プロセス(階層3)																										
業務ID	業務名	業務プロセスID	業務プロセス名	業務プロセスID	業務プロセス名	業務プロセスID	業務プロセス名																									
BP1	図書貸出管理	貸出																														
業務処理																																
1.概要 1-1 目的 蔵書を貸し出すというサービスを提供する。 1-1 図書貸出管理の大まかな流れ 図書貸出管理とは、利用者からの蔵書の検索から予約、貸し出し、返却までを管理するとともに、利用者の管理も行う 1-2 サービスの差別化 区民に対して、サービスの差別を行う 2.業務処理説明 2-1 蔵書検索、貸出予約 利用者カードを読み込んだら、必要なチェックがまとめて実施され、エラーがあった場合にはエラー内容が画面にまとめて表示される ・有効期限切れ ・現在貸し出し中のものの期限切れ 2-2 貸し出し時のサービスの差別化 区民であるかにより、貸与される点数と貸し出し期間が以下のように変える																																
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2">貸出期間・点数上限</th> <th>区内居住・就労</th> <th>地域外</th> </tr> </thead> <tbody> <tr> <td rowspan="2">書籍・雑誌</td> <td>点数</td> <td>30点</td> <td>15点</td> </tr> <tr> <td>期間</td> <td>2週間</td> <td>2週間</td> </tr> <tr> <td rowspan="2">CD</td> <td>点数</td> <td>10点</td> <td>5点</td> </tr> <tr> <td>期間</td> <td>2週間</td> <td>2週間</td> </tr> <tr> <td rowspan="2">DVD</td> <td>点数</td> <td>2点</td> <td>1点</td> </tr> <tr> <td>期間</td> <td>1週間</td> <td>1週間</td> </tr> </tbody> </table> <p style="font-size: small;">現在貸し出し中の蔵書があっても、点数まで貸し出し可能である(点数は上限値)</p>								貸出期間・点数上限		区内居住・就労	地域外	書籍・雑誌	点数	30点	15点	期間	2週間	2週間	CD	点数	10点	5点	期間	2週間	2週間	DVD	点数	2点	1点	期間	1週間	1週間
貸出期間・点数上限		区内居住・就労	地域外																													
書籍・雑誌	点数	30点	15点																													
	期間	2週間	2週間																													
CD	点数	10点	5点																													
	期間	2週間	2週間																													
DVD	点数	2点	1点																													
	期間	1週間	1週間																													
2-3 返却処理																																

図 7.26 業務処理定義のイメージ

7.5 ビジネスプロセスを漏れなく、正確に記述する(DD.5)

コラム 11 要求とその仕様の記述 USDM

USDM [34]は、要求と仕様を階層化して表形式で表現し、各要求と理由（要求の存在理由）をペアで記述する業務処理定義書の作成方法である。図 7.27 に、USDM の形式で記載した業務処理定義書の例を示す。階層化された表形式で要求と仕様が紐付き、さらに理由を書くことにより、なぜその要求が発生したのかを記録することができる。これにより、システム化仕様の漏れ・誤りを減少させられる。

No.	要求ID	区分	機能ID	区分	内容
1		AsIs			資料貸出しの対象者は、当自治体に住居・就学・就労している方、または隣接する自治体に居住している方に限定
2		ToBe			資料貸出しの対象者を上記以外の自治体に居住している方にも拡大するにあわせて、貸出し処理において以下のサービスができるようになる。 ・当自治体の利用者に対する貸出し上限点数の拡大 ・本人認証方法の追加
3	R-1	要求			資料貸出しの受け付け時に利用状況のチェックを実施する際に、サービス拡大に関連する以下の内容変更を行う ・当自治体に住居・就学・就労している利用者に対する貸出し可能な資料の総数を増やして貸出しを行う
4		理由			チェック内容をサービス拡大にあわせて変更し、それらをまとめて実施することにより、貸出し業務にかかる処理の効率化をはかる
5			R-1-1	機能	利用者カードの有効期限が超過しているかどうかを確認する
6			R-1-2	機能	延滞がある場合には、どれが延滞しているかを示す
7			R-1-3	機能	取り置きがある場合には、何が取り置きされているかを示す
8			R-1-4	機能	貸出しが可能な図書、CD、DVDそれぞれの点数を、利用者が当自治体に住居・就学・就労しているかどうかで区別して示す
9	R-2	要求			利用者カードの代わりに、利用者が保有する交通系ICカードで本人確認ができるようにする
10		理由			図書館独自の利用者カードでの本人確認では、図書館のカードを持参する必要があるため、不携帯時の別方法での本人確認や、紛失による再発行が発生しやすい これに対して、交通系ICカードは多くの利用者が毎日使用しており、紛失・盗難時にすぐに気づくので、紛失防止だけでなく、盗用による図書盗難を防止する効果も期待できる
11			R-2-1	機能	図書館が発行する利用者カード以外に、事前に登録した本人所有の交通系ICカードにより本人確認できるようにする

図 7.27 USDM での記述例

7.6 相互作用のモデルを漏れなく、正確に記述する(DD.6)

相互作用のモデルとは、状態遷移図や CRUD 図に代表されるモデルである。管理対象（エンティティ）のライフサイクルを表現したもので、業務上のイベント（プロセス）により、エンティティがある状態からある状態に変化することを明確にしたものである。この表現によりエンティティと業務プロセスの関係が明確になり、双方に矛盾なく業務が遂行されるかどうかを検証できる。ここでは、状態遷移図を説明する。CRUD 図は SR. システム化要求定義で説明している。

7.6.1 状態遷移図(DD.6.1)

【目的】

- あるエンティティの状態の遷移を明確にし、業務プロセスと矛盾ないものになっているかを確認できる状態にする。

【説明】

- エンティティの状態が変わっていく様子を図示したもの。エンティティの状態は業務上のイベントにより変化する。変化するときには、変化するための条件が存在する。これにより、業務とエンティティの関係が業務として成立しているかを確認できる。
- また、状態遷移図の方が業務ルールを分かりやすく表現できることがあり、その場合には状態遷移図をエンティティ定義書より先に作成し、新しい業務の検討に使うこともできる。

【留意事項】

- エンティティが持つ状態、および状態を変化させるイベントをすべて洗い出す。各イベントに対して、どのシステム化業務フローの業務タスクが対応するかを確認する。
- 1つのエンティティの状態遷移を記載するが、他のエンティティの状態遷移と同期を取って状態が遷移するエンティティがあるかもしれないので注意して確認する。
- また、本業務機能構成を先に作成してから業務フローを作成するパターンもある。

【成果物イメージ】

- 状態遷移図に記載する主な記載項目、イメージを表 7.19、図 7.28 に示す。

7.6 相互作用のモデルを漏れなく、正確に記述する(DD.6)

表 7.19 状態遷移図の主な記載項目

分類	項番	記述内容	記述内容の説明
1. 書誌情報	1.1	状態遷移図の ID	状態遷移図を一意に識別するためのコード
	1.2	状態遷移図の名称	状態遷移図の名称
2. 構成要素	2.1	開始/終了	状態遷移の開始や終了を表す
	2.2	状態	状態の名称
	2.3	遷移	状態の遷移を矢印で記載。状態遷移の補足説明を書いても良い
	2.4	分岐	業務用語を分類するためのキーワードで自由設定
	2.9	備考	特記事項など自由記述

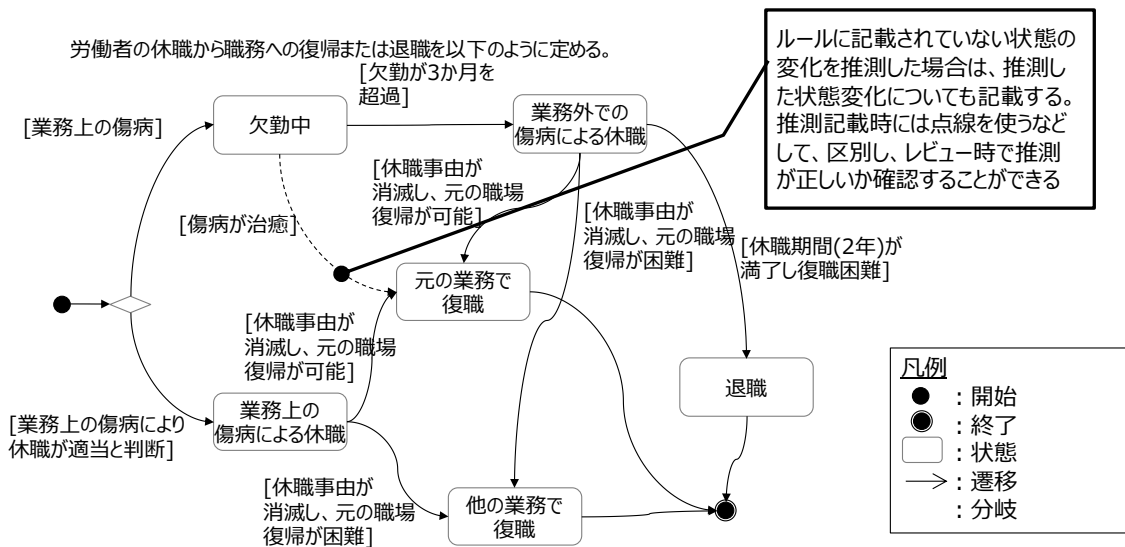


図 7.28 状態遷移図のイメージ (再掲)

7.7 コミュニケーション資料を作成する (DD.7)

システム開発に継承すべきドキュメントの作成はスケジュールや担当などが明確にされ要件定義の成果物として認識されている。しかし、経営層や業務部門長への説明資料やメンバ全員に共通認識してもらうための資料の作成はスケジュールも担当も不明確なことがある。これらのコミュニケーションを円滑にするためのドキュメントを、「業務用語定義」と「ビフォーアフター図」を例に説明する。コミュニケーションを円滑にするための作業やドキュメントはこれ以外にも考えられるので、コミュニケーション計画の立案プロセスで検討した上で、その作成を計画に盛り込む必要がある。

7.7.1 業務用語定義(DD.7.1)

【目的】

- 業務用語を定義してプロジェクトメンバがその意味を理解する。
- 業務部門やシステム開発に携わる人など関係するステークホルダに対し、共通認識をはかることにより、コミュニケーションの齟齬を低減させる

【説明】

- 業務用語を一覧化する。

【留意事項】

- 管理対象分類図で表現されたものを踏襲する。
- 上記以外の用語を洗い出し、定義する。
- 異音同義語を排除する。
例)「職制コード」と「部門コード」が全く同じ意味なら統一し、どの用語を採用するか決める。他の異音同義語は禁止語とし、成果物の記述には使用しない。
- 同音異義語を排除する。
例) 同じ「サービス商品」という用語を、ある部門では、特別価格で売るものを言い、ある部門では人の活動を有償化で行う事を言うなど、部門ごとに違う意味で使用していたら、別の用語として定義しなおす。
- 略語の用法に注意する。
正式名称ではなく略語で会話などが成り立つケース、英字の略語を多用する傾向もあるので注意が必要である。

7.7 コミュニケーション資料を作成する(DD.7)

- その他、固有名詞・ローカル用語・専門用語・業界用語・IT用語なども定義する。
- 最初からすべての用語を定義するのは時間がかかるので、定義ができたものから公開し成長させていく。
- いつでも見られるように公開場所を分かりやすくする。

【成果物イメージ】

- 業務用語定義に記載する主な記載項目、イメージを表 7.20、図 7.29 に示す。

表 7.20 業務用語定義の主な記載項目

分類	項番	記述内容	記述内容の説明
1. 構成要素	1.1	No.	業務用語定義の通し番号（行番号）
	1.2	構造	分類構造がある用語を構造番号で表す
	1.3	業務用語	業務用語の名称
	1.4	読み仮名	業務用語の読み仮名
	1.5	説明	業務用語の意味の説明
	1.6	異音同義語	異音同義語があることを示す
	1.7	同音異義語	同音異義語があることを示す
	1.8	カテゴリ	自由記述 仕分などに利用
	1.9	備考	特記事項など自由記述

NO.	構造	利用可否	業務用語	読み仮名	説明	異音同義語	同音異義語	略語	カテゴリ	備考
1	1	可	サービス商品	サービスショウヒン	人の活動を有償化で行う事		●		マスタ	
2	1	不可	サービス商品	サービスショウヒン	特別割引価格で売るもの		●		マスタ	
3	1	可	特別割引商品	トクベツワリビキショウヒン	特別割引価格で売るもの				マスタ	
4	2	可	顧客	コキヤク	売り、買いに関係なく、取引を行っている企業、個人を顧客と呼ぶ				マスタ	
5	2-1	可	仕入先	シイレサキ	買い取引を行っている企業、個人を仕入先と呼ぶ				マスタ	
6	2-2	可	販売先	ハンバイサキ	売り取引を行っている企業、個人を販売先と呼ぶ				マスタ	
7	2-2-1	可	得意先	トクイサキ	販売先の中で、取引規模が大きいなど会社が指定した優遇対象となる業、個人				マスタ	
8	2-2-2	可	一般顧客	イワンコキヤク	得意先以外の販売先				マスタ	
9	3	不可	ユーザ	ユーザ	当社の商品またはサービスを直接利用する人	●			マスタ	
10	3	可	利用者	リヨウシャ	ユーザと同じ意味。当社の商品またはサービスを直接利用する人	●			マスタ	
11	4	可	請負工事	ウケオコウジ	テナント等から依頼を受け行う請負工事。	●			仕入	
12	4	不可	請負作業	ウケオコウギョウ	請負工事と同意。本プロジェクト内では、使用しないこと。	●			仕入	
13	5	可	回議	カイギ	決裁申請事項の承認・否認行為を行うこと。				販売	
14	6	可	回議者	カイギシャ	決裁申請事項を審査し、同意、不同意の判断を行なう者のこと。				販売	
15	7	可	回議ルート	カイギルート	個々の決裁がたどる承認経路。				販売	
16	8	可	UX	ユーエクス	IT用語。User Experienceの略であり、日本語では、ユーザ経験（体験）と訳される。ユーザ経験とは、ユーザが製品やサービスを利用することによって受ける気持ちのことを指す。			●	IT用語	
17	9	可	FLL	エフエルディ	固有名詞。富士物流株式会社			●	固有名詞	

図 7.29 業務用語定義のイメージ

7.7.2 ビフォーアフター図(DD.7.2)

【目的】

- 経営層や業務部門長などに今回のシステム開発の概要を説明して、内容の合意、開発の承認を得る。

【説明】

- システム開発にともない、業務やシステムサービスがどのように現在から変わるのかまた、その効果を分かりやすく記載するドキュメント

【留意事項】

- 現在の業務と新しくなる業務を分かりやすく比較する。絵などを使うと分かりやすい。現状と新規の違いを強調するようにすると分かりやすい。
- 現状の欄には課題を明記する。
- 新規の欄には狙い、予測効果を明記する。できれば定量が良いが定性でも構わない。
- これらを実現するための前提条件や課題を明記する。

【成果物イメージ】

- ビフォーアフター図に記載する主な記載項目、イメージを表 7.21、図 7.30 に示す。

表 7.21 ビフォーアフター図の主な記載項目

分類	項番	記述内容	記述内容の説明
1. 書誌情報	1.1	ビフォーアフター図の ID	ビフォーアフター図を一意に識別するためのコード
	1.2	ビフォーアフター図の名称	課題やテーマなどを記載する
	1.3	対象プロセス	本課題/テーマの対象となる業務プロセスを記載する
2. 構成要素	2.1	狙い	経営目的や業務目的を記載
	2.2	現状業務	現状の業務を表現する
	2.3	現状の問題点	現状の問題点・課題を記載する
	2.4	新規業務	新しく変わる業務を表現する
	2.5	予測効果	定性、定量効果を記載する
	2.6	前提条件/検討事項	実現するにあたっての前提条件や今後検討しなければならない事項を記載する

7.7 コミュニケーション資料を作成する(DD.7)

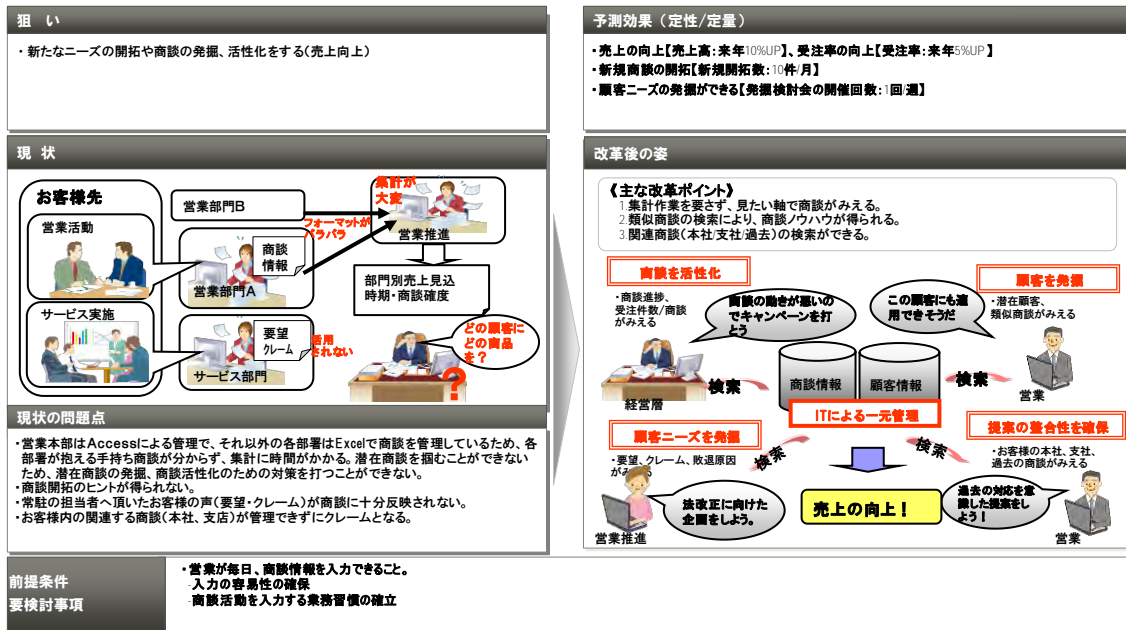


図 7.30 ビフォーアフター図のイメージ (再掲)

7.8 一覧系成果物で要求の抜け漏れを防ぐ (DD.8)

ビジネス要求定義プロセスではユーザから問題、ニーズ、課題が混在して提示され、そこからビジネス要求として目的と手段が整理される。整理された目的と手段からシステム化要求が漏れなく定義されていないと、開発工程における手戻りを引き起こすだけでなく、稼働後のユーザ満足度の低下の原因となり得る。システム化要求ならびに、その実現手段を俯瞰できる形で文書化し、漏れがないことを合意する必要がある。

7.8.1 システム化業務一覧(DD.8.1)

【目的】

- 要件定義対象の業務のうち、システム化の対象となる業務の全体を把握する。
- システム化対象規模を見積もる。

【説明】

- システム化業務一覧は、システム利用作業または機能の一覧である。業務と対応付けて、システム化業務を一覧化し、俯瞰する。

【留意事項】

- システム化業務一覧をグルーピングして、階層的に記述する。
- 業務一覧とビジネスプロセスフローを使って系統立てて記述する。
- 対応する利用者の記述をする。
- 振る舞いの想定や制約などの特徴を関係付ける。

【成果物イメージ】

- システム化業務一覧に記載する主な記載項目、イメージを表 7.22、図 7.31、図 7.32 に示す。

7.8 一覧系成果物で要求の抜け漏れを防ぐ(DD.8)

表 7.22 システム化業務一覧の主な記載項目

分類	項番	記述内容	記述内容の説明
1. 書誌情報	1.1	システム化業務一覧の ID	システム化業務一覧を一意に識別するためのコード
	1.2	システム化業務一覧の名称	システム化業務一覧の名称
	1.3	概要	システム化業務一覧の範囲、目的
2. 構成要素	2.1	ID	システム化業務の各々を一意に識別するためのコード
	2.2	システム化業務名	システム化業務（システム利用作業または機能）の名前 業務ごとのグループ化など、見やすくするために階層構造で記載することもある
	2.3	概要	システム化業務の処理内容の簡単な説明
	2.4	分類	システム化業務の処理内容の種類や方式の分類 例：バッチ、オンラインなど
	2.5	利用者	該当のシステム化業務を利用できる利用者の役割
	2.6	備考	該当のシステム化業務に関して補足する情報 (関連業務、事前条件、事後条件、業務的な例外時の情報など)

プロジェクト名	A社殿向け次期システム開発	システム名	次期システム	バージョン	1	
ドキュメントID	JS0001	作成者	田中一郎	作成日付	2007/11/8	
ドキュメント名	次期システム外部設計書	更新者		更新日付		
システム化業務一覧ID	M0001	システム化業務一覧名称	一般ユーザ向けQAサブシステムシステム利用作業一覧			
概要	次期システムにて提供するシステム利用作業を一覧で記載する。					
No	分類	システム利用作業ID	システム利用作業名	利用者	システム利用作業概要	備考
1	ユーザ	UC-001	ユーザ登録	社員	QAシステムを利用するためにユーザ情報の登録を行う。ユーザ登録はQAシステム利用の前提となる。	
2	共通	UC-002	ログイン	登録済み社員	システムにログインする。	
3	質問	UC-003	質問登録	お客様相談窓口	お客様などから受けた質問を、QAシステムに登録する。	
4	質問	UC-004	質問検索	登録済み社員	登録済みの質問を参照する。検索条件として、カテゴリ、回答の有無、時期、登録者、キーワードを指定する。	
5	質問	UC-005	質問修正	お客様相談窓口(質問登録者)	登録した質問を修正する。(未回答の質問に回答を追加登録する場合も含む。)質問の登録者のみ、修正可能とする。	
.

図 7.31 システム化業務一覧のイメージ（システム利用作業が対象）

7.8 一覧系成果物で要求の抜け漏れを防ぐ(DD.8)

(出典) IPA 「機能要件の合意形成ガイド」 [28]

プロジェクト名	A社殿向け次期システム開発		システム名	次期システム	バージョン	1			
ドキュメントID	JS0001		作成者	田中一郎	作成日付	2007/11/8			
ドキュメント名	次期システム外部設計書		更新者		更新日付				
システム化業務一覧ID	M0001	システム化業務一覧名称	一般ユーザ向けQAサブシステムシステム機能一覧						
概要	次期システムにて提供するシステム利用作業を一覧で記載する。								
No	機能名			機能ID	分類	処理タイミング	機能概要	利用者	備考 (関連業務)
	大機能	中機能	小機能						
1	予算管理	在庫機能		X01			予算立案のためのデータを集計・処理する。		
2				X0103			在庫予算データを処理する。		
3			在庫予算計算	X010301	バッチ	日次夜間	日次の在庫予算を計算する。		在庫管理
4			在庫予算入力	X010302	オンライン	予算入力期間随時	予算在庫量・金額を在庫予算データベースに登録する。	倉庫担当者	在庫管理
・			・	・	・	・	・	・	
・			・	・	・	・	・	・	
・			・	・	・	・	・	・	

図 7.32 システム化業務一覧のイメージ (機能が対象)

(出典) IPA 「機能要件の合意形成ガイド」 [28]

7.8 一覧系成果物で要求の抜け漏れを防ぐ(DD.8)

7.8.2 画面一覧(DD.8.2)

【目的】

- システム化対象の画面の全体を把握しやすくする。
- システム化対象規模を見積もる。

【説明】

- システムで使用する画面の一覧。開発する画面の一覧／俯瞰を表現する。

【留意事項】

- 画面構成の全体を把握するために、ツリー形式で作成する。
- 画面の重複、抜け漏れを防ぐために、メニュー階層別、分類別、機能別等に画面一覧を分割し、分類した項目で並び替える。
- 画面が使われる状況を明確にするために、各画面の利用者、利用目的、アクセス頻度、利用タイミング等を補足する。

【成果物イメージ】

- 画面一覧に記載する主な記載項目、イメージを表 7.23、図 7.33、図 7.34 に示す。

7.8 一覧系成果物で要求の抜け漏れを防ぐ(DD.8)

表 7.23 画面一覧の主な記載項目

分類	項番	記述内容	記述内容の説明
1. 書誌情報	1.1	画面一覧の ID	画面一覧を一意に識別するためのコード
	1.2	画面一覧の名称	画面一覧の名称
	1.3	概要	画面一覧の範囲、目的を説明する
2. 構成要素	2.1	No.	画面一覧の通し番号（行番号）。本項目によって画面数を把握できる
	2.2	画面名	画面の名称
	2.3	画面 ID	画面を一意に識別するコード
	2.4	分類	画面の種類。例えば、メニュー、入力フォーム、一覧表示等がある 後で画面の共通化を検討するために分類を記入する
	2.5	階層	画面をメニュー構造などの階層構造で整理した場合の位置付け
	2.6	説明	画面の利用者、利用目的、利用頻度（随時／特定時期）、画面の複雑さ（表示品目数、項目数等）、特徴等の説明
	2.7	機能名	関連する機能の名称。
	2.8	機能 ID	関連する機能を一意に識別するコード
	2.9	備考	特徴的な動作（スクロール、リセット、入力チェック、タイミング、ピーク時件数、想定利用者）等、特記事項

7.8 一覧系成果物で要求の抜け漏れを防ぐ(DD.8)

共通情報	プロジェクト名	A社 販向けJava Pet Store開発	作成者	Java Pet Store 画面一覧担当
	システム名	Java Pet Store	作成日付	2007/2/9
	工程名	外部設計	バージョン	2
	ドキュメントID	JPS0001	更新者	Java Pet Store 画面一覧担当
	ドキュメント名	Java Pet Store 外部設計書	更新日付	2007/3/2
書誌情報	画面一覧のID	JPS-UI LIST-0001		
	画面一覧の名称	Java Pet Store 画面一覧		
	概要	本画面一覧は、Java Pet Storeの外部設計書に含まれる画面レイアウト、入出力項目、アクション明細に記述された画面の一覧を示す。		

No.	画面名	画面ID	分類	階層			説明	機能名	機能ID	備考
				階層1	階層2	階層3				
1	ペットストアポータル画面	S-00-01	メニュー	ペットストア総合メニュー	ユーザ機能ポータルメニュー		ペットストアのポータル。購入、検索、カタログ選択などのメニューを表示する。	メニュー	F001	-
2	ペット登録画面	S-01-01	入力フォーム	ペットストア総合メニュー	ユーザ機能ポータルメニュー	ペット登録	販売者が、販売したいペットの情報(カテゴリ、名前、特徴、価格、画像、タグ)を登録する。	ペット登録	F101	登録時に入力必須情報をチェックする。
3	販売者登録画面	S-02-01	入力フォーム	ペットストア総合メニュー	ユーザ機能ポータルメニュー	ペット登録	販売者の情報(氏名、住所等)を登録する。	販売者登録	F201	登録時に入力必須情報をチェックする。
4	ペット検索画面	S-03-01	条件入力フォーム	ペットストア総合メニュー	ユーザ機能ポータルメニュー	ペット検索	ペットを検索するための条件を入力する。	ペット検索	F301	-
5	ペット検索結果表示画面	S-03-02	検索結果一覧表示フォーム	ペットストア総合メニュー	ユーザ機能ポータルメニュー	ペット検索	ペットの検索結果を一覧表示する。	ペット検索	F301	-
6	位置情報表示画面	S-04-01	地図表示フォーム	ペットストア総合メニュー	ユーザ機能ポータルメニュー	位置検索(地図)	地図上に、選択した位置(所望のペットの居場所)を表示する。	位置検索	F401	-
7	ユーザログイン画面	S-05-01	入力フォーム	ペットストア総合メニュー	ユーザ機能ポータルメニュー	ユーザ登録	ペットの売買に参加するためにシステムにログインする。	ユーザ登録	F501	パスワードエラーの場合の再入力回数は3回までとする。
8	ユーザ登録画面	S-05-02	入力フォーム	ペットストア総合メニュー	ユーザ機能ポータルメニュー	ユーザ登録	ペットの売買に参加するためにユーザ登録を行う。	ユーザ登録	F501	登録時に入力情報をチェックする。
9	ユーザ同意書表示画面	S-05-03	情報表示フォーム	ペットストア総合メニュー	ユーザ機能ポータルメニュー	ユーザ登録	ユーザ登録を完了させるために、同意書を表示し、同意の意志を確認する。	ユーザ登録	F501	-
10	プライバシーポリシー表示画面	S-05-04	情報表示フォーム	ペットストア総合メニュー	ユーザ機能ポータルメニュー	ユーザ登録	ユーザ登録を完了させるために、プライバシーポリシーを表示し、同意の意志を確認	ユーザ登録	F501	-
11	ユーザ登録完了画面	S-05-05	完了通知画面	ペットストア総合メニュー	ユーザ機能ポータルメニュー	ユーザ登録	ユーザ登録が完了したことを通知する画面。	ユーザ登録	F501	-
12	カタログ画面	S-06-01	画像付情報表示フォーム	ペットストア総合メニュー	ユーザ機能ポータルメニュー	カタログ	購入者が、ペットのカタログを表示する画面。	ペット購入	F601	-
13	ショッピングカート表示画面	S-06-02	情報表示フォーム	ペットストア総合メニュー	ユーザ機能ポータルメニュー	ペット購入	購入者が、所望のペットを選択し、ショッピングカートの内容を表示する画面。	ペット購入	F601	-
14	支払い情報入力画面	S-06-03	入力フォーム	ペットストア総合メニュー	ユーザ機能ポータルメニュー	ペット購入	購入者が、支払いの情報を入力するための画面。	ペット購入	F601	登録時に入力情報をチェックする。
15	注文完了画面	S-06-04	完了通知画面	ペットストア総合メニュー	ユーザ機能ポータルメニュー	ペット購入	ペットの購入の注文が完了したことを通知する画面。	ペット購入	F601	-

図 7.33 画面一覧のイメージ (一般的な表形式)

(出典) IPA 「機能要件の合意形成ガイド」 [28]

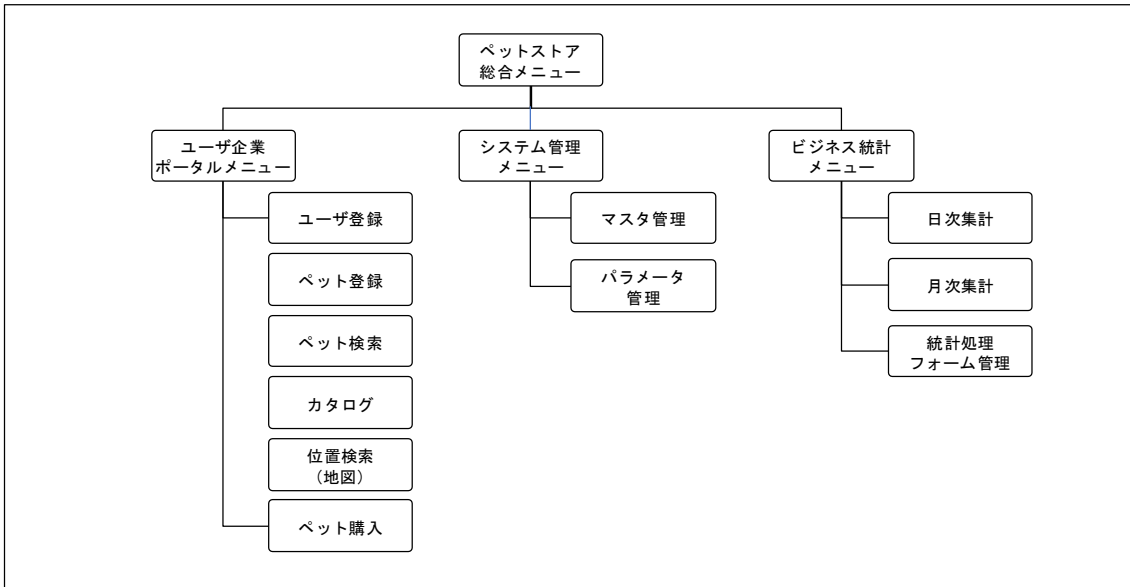


図 7.34 画面一覧のイメージ (ツリー形式で表現した例)

(出典) IPA 「機能要件の合意形成ガイド」 [28]

7.8.3 帳票一覧(DD.8.3)

【目的】

- システム化対象の帳票の全体を把握しやすくする。
- システム化対象規模を見積もる。

【説明】

- システムが作成する帳票の一覧。開発する帳票の一覧／俯瞰を表現する。

【留意事項】

- 帳票が使われる状況を明確にするために、利用する主な業務と利用者、各帳票の出力タイミングを記載する。

【成果物イメージ】

- 帳票一覧に記載する主な記載項目を表 7.24 に示す。

表 7.24 帳票一覧の主な記載項目

分類	項番	記述内容	記述内容の説明
1. 構成要素	1.1	帳票 ID／名称	帳票を特定する識別子または名称を記述する
	1.2	出力タイミング	当該帳票の出力タイミングを記述する。
	1.3	出力様式	当該帳票の出力様式（汎用紙、専用紙、PDF など）を記述する
	1.4	用紙サイズ	当該帳票の用紙サイズと印刷向きを記述する
2. 任意項目	2.1	帳票利用業務	当該帳票を利用する主な業務名称
	2.2	帳票出力システム機能	当該帳票を出力する主なシステム機能名称
	2.3	帳票形式の識別	当該帳票の形式（一覧表、詳細表、集計表など）の識別
	2.4	帳票データの作成方法	当該帳票の作成方法（バッチ、オンライン）を記述する
	2.5	改ページ条件	当該帳票の改ページ条件を記述する
	2.6	出力制約条件	当該帳票の出力制約条件を記述する

7.8 一覧系成果物で要求の抜け漏れを防ぐ(DD.8)

7.8.4 外部インターフェース一覧(DD.8.4)

【目的】

- 要件定義対象システムと他システムとのデータ連携インターフェースを洗い出す。
- システム化対象規模を見積もる。

【説明】

- 外部インターフェース一覧とは、当該システムと他システムとの間でデータの連携が行われるインターフェースを一覧形式にまとめたものである。

【留意事項】

- 外部インターフェースの抜け漏れを少なくするために、外部インターフェースが発生するシステム単位に記述範囲を絞り、外部インターフェース一覧を作成する。

【成果物イメージ】

- 外部インターフェース一覧に記載する主な記載項目、イメージを表 7.25、図 7.35 に示す。

7.8 一覧系成果物で要求の抜け漏れを防ぐ(DD.8)

表 7.25 外部インターフェース一覧の主な記載項目

分類	項番	記述内容	記述内容の説明
0. 共通情報	—	プロジェクト名、システム名、工程名、ドキュメント ID、ドキュメント名、作成者、作成日付、バージョン、更新者、更新日付	
1. 構成要素	1.1	外部インターフェース ID / 外部インターフェース名称	外部インターフェースを特定する識別子および名称を記述する
	1.2	接続先相手システム名	当該システムと関連する外部システムを識別する名称
	1.3	出力 / 入力の識別	関連する外部システムとやり取りされる情報の方向（出力 [=送り出し]、入力 [=受け取り]）を示す
	1.4	既存 / 新規の識別	外部インターフェースが既存にあるのか、新規に作成されるインターフェースかの識別
	1.5	接続方式	外部インターフェースが具体的にどのような仕組みで実現されるかの方式の説明
	1.6	処理タイミング	外部インターフェースを使ったデータの送り出し / 受け取りがどのような処理タイミングで実行されるかの説明

No	外部インタフェースID	外部インタフェース名	出力 (To)	入力 (From)	相手先システム情報		接続方式			処理タイミング		備考
					システム名	既 / 新	受渡し手段	データ形式	タイミング	送受信時間		
1	7800001	受注情報	○		販売管理システム	新規	ftp	XML	バッチ (定時)	日中		
2	7800002	商品出荷状況情報		○	販売管理システム	新規	ftp	XML	バッチ (日次)	夜間		
3	7800003	商品情報		○	販売管理システム	既存	ftp	XML	バッチ (定時)	日中		
4	7800004	新規顧客情報	○		顧客管理システム	既存	ftp	XML	バッチ (定時)	日中		
5	7800005	顧客更新情報	○		顧客管理システム	既存	ftp	XML	バッチ (定時)	日中		
6	7800006	受注処理ログ情報		○	処理証跡管理システム	既存	API	テキスト	オンライン (即時)	日中		
7	7800007	信用照会問合せ情報	○		クレジット信用照会システム (外部)	既存	https (SSL利用)	テキスト	オンライン (即時)	日中		
8	7800008	信用照会結果情報		○	クレジット信用照会システム (外部)	既存	https (SSL利用)	テキスト	オンライン (即時)	日中		

図 7.35 外部インターフェースのイメージ

(出典) IPA 「機能要件の合意形成ガイド」 [28]

7.8 一覧系成果物で要求の抜け漏れを防ぐ(DD.8)

7.8.5 エンティティ一覧(DD.8.5)

【目的】

- システム化対象業務に関連するすべてのエンティティについて、その分類や属性などを体系的に整理し、網羅性を検証可能にする。
- エンティティの概要を容易に把握できるようにする。
- システム化対象のスコープで扱うデータの規模を見積もる。

【説明】

- エンティティを識別する名称の一覧、エンティティの補足説明、エンティティ定義の目次として作成する。

【留意事項】

- データモデルに業務量を反映するためにデータ件数や保存期間を記述する。
- エンティティを定義した根拠を示すために、現行システムなどの具体的なエンティティの抽出元を記述する。
- データモデルの変更点を確認するために、現行システムのデータモデルからの変更点を明記する。

【成果物イメージ】

- エンティティ一覧に記載する主な記載項目、イメージを表 7.26、図 7.36 に示す。

表 7.26 エンティティ一覧の主な記載項目

分類	項番	記述内容	記述内容の説明
1. 書誌情報	1.1	エンティティ一覧の ID	エンティティ一覧を一意に識別するためのコード
	1.2	エンティティ一覧の名称	エンティティ一覧の名称
	1.3	概要	エンティティ一覧の範囲、目的
2. 構成要素	2.1	エンティティ名	エンティティの名称
	2.2	エンティティ種別	エンティティ種別 (イベント系エンティティ、リソース系エンティティなど)
	2.3	定義	エンティティの定義
	2.4	備考	担当部署、業務名、保存期間、抽出先など

7.8 一覧系成果物で要求の抜け漏れを防ぐ(DD.8)

プロジェクト名		システム名		バージョン	
ドキュメントID		作成者		作成日付	
ドキュメント名		更新者		更新日付	
エンティティ一覧ID		エンティティ一覧名称			
概要					
No.	エンティティ名	エンティティ種別(注)	定義		
1	顧客	リソース系	顧客の個人情報		
2	製品分類	リソース系	製品分類コードの定義		
3	製品	リソース系	各製品の基本情報		
4	受注	イベント系	受注の基本情報		
5	受注明細	イベント系	受注における明細部分		
6	売上高	サマリ系	商品別売上集計		

図 7.36 エンティティ一覧のイメージ

(出典) IPA 「機能要件の合意形成ガイド」 [28]

7.9 インターフェースの確認を中心に要求の妥当性、実現性を確認する(DD.9)

業務部門とのシステム化要求確認は機能に対する要求の確認が中心となる。その際には、ビジネス要求を入力として利用者とシステムとの間のインターフェースを中心に確認すると良い。まずは、利用者とシステムとの相互作用を中心に確認し、その相互作用を実現する画面や帳票などのユーザインターフェースの確認へと段階的に詳細化していく。定義内容のバラつきによるユーザビリティ低下を回避するために、まずUI標準を策定して、それを合意した上で個々のユーザインターフェースを定義することが望ましい。UI標準を策定するためには、利用者の特性や求められる要求を分析する必要がある。

なお、ユーザインターフェースの実現イメージが明確になってくると、レイアウトの細かいところなど枝葉に目が行きがちである。しかしユーザインターフェース定義の本来の目的は、定義したユーザインターフェースでビジネス要求が実現できるか、さらにはビジネス目的を達成できるかを業務部門と確認することである。その点に留意してほしい。

7.9.1 システム化要求仕様(DD.9.1)

【目的】

- 個々のシステム化業務の内容（使い方、処理の流れ）をステークホルダ間で合意形成する。特に現行システムから変更が発生したシステム化業務を中心に確認する。
- 抽出したシステム化業務について、アクターとシステムとの相互作用やエンティティとの関わり（どのエンティティのレコードを生成、参照、更新、削除しているか）を明確にする。

【説明】

- 利用者とシステムとの間の相互作用に関する仕様書である。システム化業務ごとに、その使い方、あるいは必要な入力・処理の流れ・出力を表現する。

【留意事項】

- ビジネスプロセスフローに書かれているシステム化業務と詳細に説明する文章・図表を互いに関連付け、お互いに相手を追跡できるようにする。
- システム化業務の実行前後の条件を記述する。
- 利用者とシステムのそれぞれのアクションに分けて記述する。
- 1つのシステム化業務の振る舞いに対して、基本・代替・例外の3種類のシナリオを

7.9 インターフェースの確認を中心に要求の妥当性、実現性を確認する(DD.9)

考える。

【成果物イメージ】

- システム化要求仕様に記載する主な記載項目、イメージを表 7.27、図 7.37 に示す。

表 7.27 システム化要求仕様の主な記載項目

分類	項番	記述内容	記述内容の説明
1. 書誌情報	1.1	システム化業務説明のID	システム化業務説明一意に識別するためのコード
	1.2	システム化業務説明の名称	システム化業務説明の名称
	1.3	概要	システム化業務説明の範囲、目的
2. 構成要素	2.1	入力データ	システム化業務への入力情報 データには、以下の区分も記述する。(画面／エンティティ／ファイル／テーブル)
	2.2	出力データ	システム化業務への出力情報 データには、以下の区分も記述する。(画面／エンティティ／ファイル／テーブル)
	2.3	システム化業務の振舞い	シナリオ：システム外部の人とシステム利用作業の間のやり取り・相互作用のシーケンス（複数の場合に分けて記述しても良い） 処理プロセス：機能の内部の処理ステップの流れ（それぞれの処理ステップの入出力データを、前後に併記しても良い）
	2.4	事前条件	システム化業務の開始前にシステムに期待される状態
	2.5	事後条件	事前条件が守られたときに、システム化業務の終了後に保証されるシステムの状態
	2.6	備考	該当のシステム化業務に関して補足する情報(関連業務、トリガ（起動のタイミングを含む）、あるいはビジネスルールに基づくこのシステム化業務の振舞いの根拠など)

7.9 インターフェースの確認を中心に要求の妥当性、実現性を確認する(DD.9)

プロジェクト名	A社殿向け次期システム開発	システム名	次期システム	バージョン	1
ドキュメントID	JS0001	作成者	田中一郎	作成日付	2007/11/8
ドキュメント名	次期システム外部設計書	更新者		更新日付	
システム化業務ID	M0001	システム化業務名称	一般ユーザ向けQAサブシステムシステム利用作業		
概要	ユーザの情報を変更する				
事前条件	管理者権限を持ったユーザでログインしていること	事後条件	ユーザ情報が変更されること		
入力データ	ユーザ情報(画面)	出力データ	ユーザ情報(テーブル)		
基本シナリオ: ユーザの情報を変更する					
■使用者のアクション			■システムのアクション		
1. ユーザ情報の一覧を要求する。			2. ユーザ情報の一覧を利用者へ返す。		
3. 変更するユーザを選択し、ユーザの詳細情報を要求する。			4. ユーザの詳細情報を利用者へ返す。		
5. ユーザ情報の変更を要求する。			6. 該当するユーザ情報を変更する。		
~~~~~					
例外シナリオ3: 指定されたユーザが不在					
■使用者のアクション			■システムのアクション		
1. ユーザIDを指定誌、ユーザの詳細情報を要求する。			2. ユーザ不在のエラーメッセージをユーザへ返す。		
備考	なし				

図 7.37 システム化要求仕様のイメージ

(出典) IPA 「機能要件の合意形成ガイド」 [28]

## 7.9.2 UI 標準(DD.9.2)

### 【目的】

- 統一的なユーザインターフェースによるユーザビリティ低下防止のために画面のレイアウト、振る舞いを統一する。

### 【説明】

- 画面にまたがる前提や制約、記述の便宜に基づく約束事を記載する。

### 【留意事項】

- 各画面個別に合意すべきことを少なくするために、共通して合意すべきことは UI 標準として定義する。
- 各画面での確認ポイントを絞るために、画面遷移パターン、エラー表示方法、エラーメッセージを標準として定義する。
- 共通ルールの意図を理解しやすくするために準拠すべき社内外の基準・標準、ガイドライン、特筆すべき要求、開発コンセプトなどがあれば明記しておく。

### 【成果物イメージ】

- UI 標準に記載する主な記載項目、イメージを表 7.28、図 7.38～図 7.44 に示す。

## 7.9 インターフェースの確認を中心に要求の妥当性、実現性を確認する(DD.9)

表 7.28 UI 標準の主な記載項目

分類	項番	記述内容	記述内容の説明
1. 書誌情報	1.1	共通ルール ID	ルールを記述するドキュメントを一意に特定するための ID
	1.2	共通ルールの名称	ルールを記述するドキュメントに与える名称
	1.3	概要	何のためのルールなのかを説明する
2. 構成要素	2.1	ページの構成要素	1つの画面を複数のエリアに分割する場合の画面レイアウト、エリアに与える名称、エリアが使用される用途、例
	2.2	ページの表示イメージ	ページの構成要素が組み合わさったときの表示イメージ、実際の画面デザインを用いて表現
	2.3	配色	キーカラーの指定、アクセントカラーの指定、エリアごとの基本配色、フォントカラーの指定
	2.4	文字フォント・サイズ	画面に表示する文字フォントの指定、サイズの指定、行間の指定
	2.5	全体構造とナビゲーション	画面のもつ階層構造、メニュー構造、分類方法の基準、ナビゲーション表示方法や表示変更方法を示す説明、および図
	2.6	画面遷移パターン	単一ブラウザ/複数ブラウザの使用可否の定義、フレームにまたがる遷移の有無、そのコントロール方法の定義、自らブラウザを閉じる画面の有無と遷移への影響
	2.7	エラー表示方法、エラーメッセージ	画面内のエラーメッセージ表示、エラーメッセージダイアログの要否、エラー画面レイアウト、エラーメッセージへのアクセントカラー使用、エラー固有の書式、メッセージの可変部と固定部の配置
	2.8	ユーザの定義	アクセス可能範囲のユーザ分類と命名、説明
	2.9	制約事項	性能要件、セキュリティ要件、その他による制約事項

## 7.9 インターフェースの確認を中心に要求の妥当性、実現性を確認する(DD.9)

1 ページの構成要素  
 ページ構成要素を以下のように定義する。  
 なお、原則としてフレームを使用しないこととする。(ピクセル値は目安)

各ページ構成要素には、主として以下の用途に用いられる。

項番	要素	用途
1	ヘッダエリア	サイトロゴ(タイトル、カバー画像)を常に表示し、Java PetStoreのサービスであることを利用者に明らかにする。 ※すべての画面で表示するものとする。
2	ユーティリティエリア	利用頻度の高い機能(商品検索)を常に利用可能にしておく。 ※すべての画面で表示するものとする。
3	ナビゲーションエリア	主要コンテンツへのリンク(顧客情報、カート、ログイン/ログアウト)を常に利用可能にし、任意にログイン/ログアウトができるようにする。 ※すべての画面で表示するものとする。
4	コンテンツエリア(sub1)	ペットメニューのサービス(共通ペットカテゴリ)を常に表示する。 ※すべての画面で表示するものとする。
5	コンテンツエリア(main)	サービス別の詳細コンテンツを表示する。エラーメッセージを画面上に表示する場合にはここに表示する。
6	コンテンツエリア(sub2)	ログインしている顧客に対して、ペットメニューのサービス(ログイン顧客の好きなペットカテゴリ)を常に表示する。 ※ログイン中のすべての画面で表示するものとする。
7	フッタエリア	クレジット表記、コピーライト表記、問合せ先 ※すべての画面で表示するものとする。
8	余白	余白

2 ページの表示イメージ  
 ページの構成要素を組み合わせた場合の表示イメージを以下に示す。

図 7.38 UI 標準のイメージ (ページの構成要素、ページの表示イメージ)

(出典) IPA 「機能要件の合意形成ガイド」 [28]



## 7.9 インターフェースの確認を中心に要求の妥当性、実現性を確認する(DD.9)

3 配色

(1) キーカラー  
背景色は□白(“#FFFFFF”)を基調とする。

(2) ページ構成要素への基本配色

項番	要素	配色(バックグラウンド)
1	ヘッダエリア	□白(“#FFFFFF”)
2	ユーティリティエリア	□白(“#FFFFFF”)
3	ナビゲーションエリア	□白(“#FFFFFF”)
4	コンテンツエリア(sub)	□白(“#FFFFFF”)
5	コンテンツエリア(main)	□白(“#FFFFFF”)
6	フッタエリア	□白(“#FFFFFF”)
7	余白	□白(“#FFFFFF”)

(3) アクセントカラー  
アクセントカラーは■赤(“#FF0000”)とする。

(4) フォントカラー  
基本色は■黒(“#000000”)とする。

図 7.39 UI 標準のイメージ (配色)

(出典) IPA 「機能要件の合意形成ガイド」 [28]

4 文字フォント・サイズ

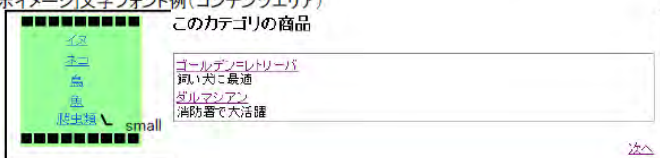
(1) 文字フォント  
書体はシンプルな書体を適用する。  
欧文部分は一般的な Helvetica または Arial、sans-serifとする。  
和文、漢文部分は指定せず、ブラウザのフォント指定に従う表示とする。  
[参考] sans-serif表示イメージ

sans-serifです。

(2) 文字サイズ  
ページの主題や項目、重要な内容を示す文字サイズはブラウザの標準サイズとする。  
詳細内容を示す文字サイズはやや小さめ(x-small;)とする。  
ユーティリティエリアとナビゲーションエリアは小さめ(small;)とする。

x-small です  
small です  
標準サイズ です

(3) 行間サイズ  
行間サイズは指定しないこととする。  
各画面で指定することは許可しない。  
[表示イメージ]文字フォント例(コンテンツエリア)



このカテゴリの商品

- イロ
- エニ
- 高
- 魚
- 豚骨湯small

ゴールドソレトリーバ  
飼い犬に最適  
オランダシアン  
消防着で穴浴槽

図 7.40 UI 標準のイメージ (文字フォント・サイズ)

(出典) IPA 「機能要件の合意形成ガイド」 [28]

## 7.9 インターフェースの確認を中心に要求の妥当性、実現性を確認する(DD.9)

5 全体構造とナビゲーション

(1) 階層構造、メニュー構造

ペットストア総合メニュー

- └ ユーザ機能ポータルメニュー
  - └ ペット登録
  - └ ペット検索
  - └ 位置検索(地図)
  - └ ユーザ登録
  - └ カタログ
  - └ ペット購入
  - └ その他画面グループ(ログイン画面など)

(2) ナビゲーションの表示パターン

	ナビゲーション	表示パターン
1	顧客情報	表示固定
2	カート	表示固定
3	ログイン/ログアウト	ユーザのログイン状態により表示を切り替え

別途、画面レイアウトおよびアクション明細を記す。

図 7.41 UI 標準のイメージ (全体構成とナビゲーション)

(出典) IPA 「機能要件の合意形成ガイド」 [28]

6 画面遷移パターン

(1) 画面遷移パターンの定義

	エリア種別	イベントアイテム、操作	遷移パターン
1	ヘッダエリア	「タイトルロゴ」イメージをクリック	ペットストアポータル画面へ遷移する。
2		「検索」ボタンを押下	ペット検索画面へ遷移する。
3		「顧客情報」リンクを押下	未ログインのユーザはログイン画面へ遷移する。ログイン済みのユーザはユーザ更新画面へ遷移する。
4		「カート」リンクを押下	ショッピングカート表示画面へ遷移する。
5		「ログイン」リンクを押下	ユーザログイン画面へ遷移する。
6		「ログアウト」リンクを押下	御礼画面へ遷移する。
7	コンテンツエリア	基本的に画面遷移による遷移に従う。	
8	フッタエリア	各種リンクを押下	表記しているサイトへ遷移する。

図 7.42 UI 標準のイメージ (画面遷移パターン)

(出典) IPA 「機能要件の合意形成ガイド」 [28]

## 7.9 インターフェースの確認を中心に要求の妥当性、実現性を確認する(DD.9)

### 7 エラー表示方法、エラーメッセージ

#### (1) エラー表示方法分類

エラー表示方法の分類を次の表のとおりとする。

種別	概要	エラー表示方法
1 簡易な入力エラー	クライアント側でチェック可能な簡単な入力エラー	ダイアログによる英語メッセージ
2 サーバ送信後のエラー	サーバ側のリクエストに対する処理でエラー検知	画面遷移したのち、各国語によるメッセージをコンテンツエリア(main)に表示する。
3 上記以外の致命的エラー	何らかの不具合により処理が続行できない。	共通のエラー画面に遷移し、メッセージを表示する。

入力エラー通知に用いるメッセージは理解容易な英語にできることから、英語メッセージに統一する。サーバ側で処理した結果のエラーはユーザへの通知を明確にするために、各国語によるメッセージとする。

#### (2) エラー画面レイアウト図

特に定めない。個別の画面レイアウトに従う。

#### (3) エラーメッセージ書式定義

コンテンツエリア内に示す主たるメッセージ部分にはアクセントカラーを用いる。

図 7.43 UI 標準のイメージ (エラー表示方法、エラーメッセージ)

(出典) IPA 「機能要件の合意形成ガイド」 [28]

### 8 ユーザの定義

#### (1) アクセス可能範囲のユーザ分類

ユーザ分析より英語圏、日本語圏、中国語圏のユーザを対象とする。

#### (2) ユーザ分類の説明

分類	未ログイン時のユーザ分類名称	ログイン時のユーザ分類名称
1 英語圏ユーザ	未ログイン英語ユーザ	ログイン英語ユーザ
2 日本語圏ユーザ	未ログイン日本語ユーザ	ログイン日本語ユーザ
3 中国語圏ユーザ	未ログイン中国語ユーザ	ログイン中国語ユーザ

### 9 制約事項

#### (1) 画像・動画の制限

インターネットを介して利用することを想定して、通信量を極力避ける。

画像の多用、画像ファイルの肥大化を抑制する。

画像ファイルはGIFを使用する。

1ファイルあたりのファイルサイズは特別な場合を除き、10KBを越えないこととする。

特別な場合とは、初期表示画面のアピールといった、固有の目的のために

必要な大きな画像を用いる場合をいう。

動画ファイルは使用しない。動画GIFも同様である。

#### (2) スクリプトファイルの制限

スクリプトの制限事項については別途定める。

#### (3) 画面レイアウトの制限

フレームを使わない。

画面を快適に使用できるモニタサイズについては、1024×768ピクセルとする。

縦スクロールによる操作は許容し、横スクロールが必要となる操作は排除する。

#### (4) ブラウザ制御の制限

シングルウィンドウ内の操作で完結する制御に留める。

別のブラウザを立ち上げることが必要なアクションは使用しない。

ブラウザのアドレス欄の隠蔽・非表示を伴う制御は行なってはならない。

図 7.44 UI 標準のイメージ (ユーザの定義、制約事項)

(出典) IPA 「機能要件の合意形成ガイド」 [28]

### 7.9.3 画面遷移図(DD.9.3)

#### 【目的】

- 各処理での画面の流れを整理し、遷移パターンの統一性、抜け漏れ（特に逆方向に戻る遷移）の検出、遷移の妥当性（特に連続操作時の戻り先の妥当性）確認などを行う。特に現行システムから変更が発生した画面遷移を中心に確認する。

#### 【説明】

- アプリケーションを実現する画面の流れを表す図表である。
- 発注者が理解する業務の流れや業務遂行におけるシステムの一連の操作に沿って、画面の位置付けと流れを、フロー図形式で表現する。

#### 【留意事項】

- 一覧性を高めるために印刷 1 ページに 1 つの完結した業務を記述する。1 ページに収まりきれない場合は、階層化した上で、別ページに記述する。
- 画面遷移の流れを理解しやすくするために、上から下、左から右に遷移するように配置する。
- 正常系と異常系の遷移を明確に区分するために、正常系と異常系の判別がしやすいかき分け方をする。

#### 【成果物イメージ】

- 画面遷移図に記載する主な記載項目、イメージを表 7.29、図 7.45 に示す。

## 7.9 インターフェースの確認を中心に要求の妥当性、実現性を確認する(DD.9)

表 7.29 画面遷移図の主な記載項目

分類	項番	記述内容	記述内容の説明
0. 共通情報	—	プロジェクト名、システム名、工程名、ドキュメント ID、ドキュメント名、作成者、作成日付、バージョン、更新者、更新日付	
1. 書誌情報	1.1	画面遷移 ID	ひとまとまりの画面遷移を一意に識別する識別子
	1.2	画面遷移の名称	ひとまとまりの画面遷移を一意に識別する名称
	1.3	概要	画面遷移の概要を記述する
2. 構成要素	2.1	画面	遷移元、遷移先の個々の画面を表す。画面レイアウトを一意に識別できる名前を記述する。画面イメージがある場合には、そのイメージを用いても良い
	2.2	遷移矢線	実現される画面の順序関係を矢線で記述する 遷移矢線は交錯しないように描画する
	2.3	アクション	画面遷移に伴って起動される動作
	2.4	イベント	画面の遷移を引き起こすきっかけとなるもの 例えば、検索ボタンが押されたときは“検索”というイベントが発生する 開発都合で必要になる内部イベント等のイベントは記述しない
	2.5	条件分岐／合流	条件により遷移が分岐する場合に記述する。 合流は分岐した複数の遷移が単一の遷移に集約されることを表す
	2.6	開始点	画面遷移の開始画面を明示する 自明の場合は不要
	2.7	終了点	画面遷移の終了画面を明示する。 自明の場合は不要
	2.8	他画面遷移への接続	画面遷移を複数の図に分割して記述する場合に、他画面遷移への接続点を明示する。接続する画面遷移を一意に識別できるように記述する
	2.9	下位画面	部品化された遷移が存在することを表すための記号。画面記号やイメージに《下位遷移》を付加する

7.9 インターフェースの確認を中心に要求の妥当性、実現性を確認する(DD.9)

分類	項番	記述内容	記述内容の説明
	2.10	遷移動作	遷移時に起こる動作の名称を記述する（画面アクション明細を特定する名前を記述するのが望ましい） 構成要素 2.3 のアクションを示す記号を用いる代わりにこの記法でアクションを記述しても良い 内部動作の記述は行わない
	2.11	異常系	例外処理等の正常系処理を実行できない場合の処理方法を記述する
	2.12	注釈	遷移図だけでは表現できない場合、設計情報として必要なものを記述する

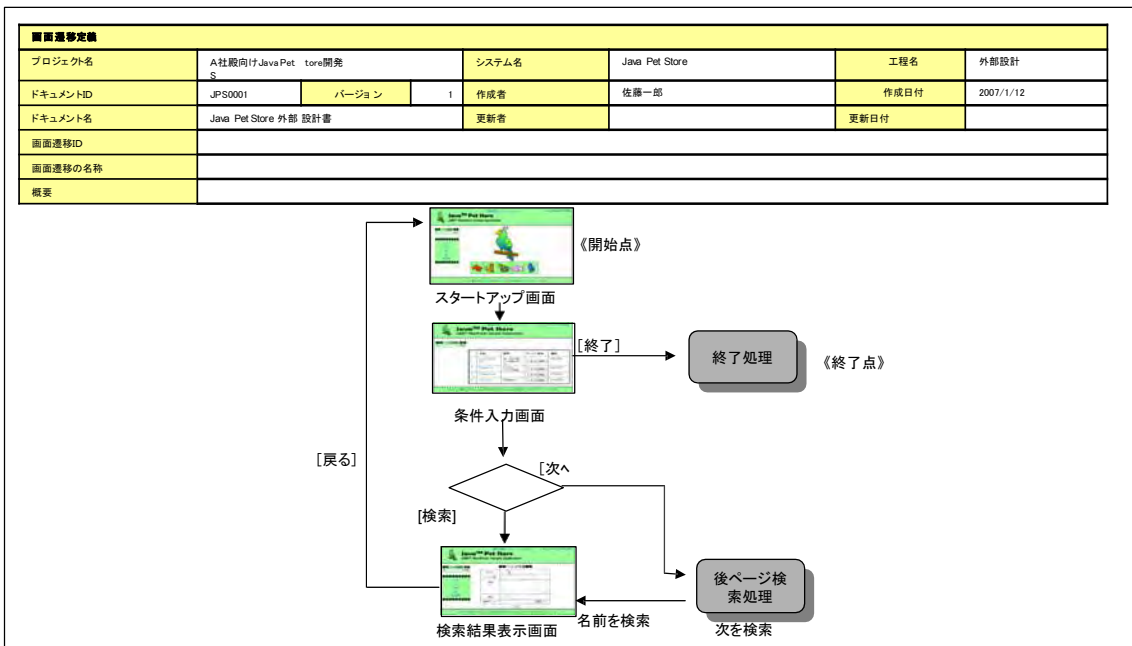


図 7.45 画面遷移図のイメージ

(出典) IPA 「機能要件の合意形成ガイド」 [28]

### 7.9.4 画面レイアウト(DD.9.4)

#### 【目的】

- 画面での処理内容に応じた大まかなレイアウトを仮決めし、画面の項目数の規模感を把握する。特に現行システムから変更が発生した画面を中心に確認する。

#### 【説明】

- 業務で使用する画面のレイアウト、入出力項目、画面遷移を作成し、業務部門の確認を取る。

#### 【留意事項】

- 要件定義、基本設計で粒度が異なることを意識して、記述粒度を定義する。
  - 要件定義：画面に対するステークホルダの要求として、画面レイアウトのイメージや必要な情報（データ項目）、画面に必要な処理（ボタンなど）などを明確にする。
  - 基本設計：実装する画面として性能や規約を考慮し再設計し、ステークホルダとの合意を行う。
- 利用目的や使い方を明確にする。

各画面の利用シーン（どの業務においてどのような人が使用するのか、業務の習熟者か、初心者か、顧客かなど）を明確にする。各画面には、当該画面が取り扱うデータの種類や、参照なのか更新なのかなど、目的用途が明確に分かるような名称を付与することを命名規約で定めることが必要である。また、利用者の権限により、画面に表示する内容や処理などが変わる可能性があるので確認をしておく。想定利用者や検討経緯、利用シーンが明らかになっていれば、基本設計工程で各画面に配置する項目の選・項目の並び順・読みやすさ・使いやすさを検討する際に、誤りの発生を抑止することができる。
- 利用目的や操作性を考えた画面レイアウトになっているか確認する。

利用目的によっては、画面デザイン時に企業のブランドカラーや企業ロゴなどに配慮することも必要となる。

  - 短時間に多くのデータを入力する業務（例えば、伝票を短時間で10キーだけで高速に入力する業務）に、複数の選択肢から選ぶチェックボックスや階層を順次掘り下げるドリルダウンなどのサポート機能を駆使した画面を設計しない。
  - 顧客や初心者の使用に向けて使いやすさが特に重視される画面の要件定義の際には、画面レイアウト、画面遷移、入力補助機能を漏らさず記載し、使いやすさに関してユーザの要件確認の手戻りがないようにする。
- 画面仕様は要件定義の段階で業務部門と合意する。
  - 入力チェック仕様などは、データ項目定義書に記載された条件以上のものがなければ、この段階では不要である。しかし、チェック機能が特に重要な場合は、確認のために記載することが望ましい。また、XXXや9,999などの桁数やデータ属性表現

## 7.9 インターフェースの確認を中心に要求の妥当性、実現性を確認する(DD.9)

ではなく、具体的なデータの例で表現することにより、ユーザの理解を確かなものとする。ただし、最大表示桁数の確認のためには、XXX や 9,999 のような表示も必要になる。

- ▶ 画面レイアウトや画面遷移は、ツールを活用してイメージや使いやすさを確認する。ツールを使用すれば比較的容易に簡易図（モックアップ）が作れるので、この段階で、簡易図（モックアップ）で業務部門に画面の操作性、画面遷移、補助機能などを確認してもらうと手戻りが少なくなる。
- なぜこの画面レイアウトになったかの検討経緯を残す。  
画面レイアウトの結論のみの記述の場合、後続工程でなぜこの画面レイアウトになったか、検討経緯が分からなくなり、技術的な問題等で修正の必要が生じた場合に判断がつかなくなり、工期に遅れが生ずるといった問題が起きる。検討経緯を画面レイアウトの注釈欄などに残しておくが良い。画面レイアウト決定までの経緯を記録に残しておく、何の目的のための画面かが確認でき、責任者や担当者（窓口）が変わった時にも担当業務の画面がどうしてこのようなレイアウトになったのかをより正確に伝達できる。
- B2C の業務や社外公開 Web ページなどについては、要件定義の段階で画面に表示する項目がほぼ確定するようであれば、リーガルチェック（法規違反や知財侵害等のチェック）を行う。

### 【成果物イメージ】

- 画面レイアウトに記載する主な記載項目、イメージを表 7.30、図 7.46 に示す。



## 7.9 インターフェースの確認を中心に要求の妥当性、実現性を確認する(DD.9)

表 7.30 画面レイアウトの主な記載項目

分類	項番	記述内容	記述内容の説明
1. 書誌情報	1.1	画面 ID	画面を区別するための番号
	1.2	画面の名称	画面の名称
	1.3	概要	何のための画面なのかを説明する
2. 構成要素	2.1	画面レイアウト	図によってレイアウト案を提示
	2.2	識別 ID	図表内で識別するため、画面部品ごとに ID を振る。画面遷移や画面入出力項目一覧、画面アクション明細などと連携して用いる
	2.3	ラベル	画面に表示される時の名称
	2.4	画面部品の種類	どのような画面部品を用いるのかの画面部品名。他にボタンなどがある
	2.5	表示範囲	表示する桁数や、テーブルのカラムの幅などを提示
	2.6	画面部品の説明	その画面部品がどのような役割を果たすのか、の意味や説明を記述する。入出力が絡む場合はこちらに説明をつける。項目の細かい定義や処理内容は別扱い
	2.7	操作手順	入力してボタンを押すなど、操作のイメージが掴めるような操作手順を記述する

## 7.9 インターフェースの確認を中心に要求の妥当性、実現性を確認する(DD.9)

画面レイアウト定義				
基本情報	画面ID S-05-01			
	画面の名称 ユーザログイン画面			
	概要 ペット購入システムを利用するために必要なログイン画面			
レイアウト図				
使用する部品				
ログイン画面で使用する部品には、ユーザ名・パスワードの入力にテキストボックスとログインボタンがある。				
識別ID	ラベル	画面部品の種類	表示範囲	画面部品の説明
①	ユーザ名	テキストボックス		ユーザ名入力するテキストボックス
②	パスワード	テキストボックス		パスワードを入力するテキストボックス
③	ログイン	ボタン		ログイン処理を起動するボタン
操作手順				
①ユーザ名をテキストボックスに入力する。				
②パスワードをテキストボックスに入力する。				
③「ログイン」ボタン押下する。				

図 7.46 画面レイアウトのイメージ

(出典) IPA 「機能要件の合意形成ガイド」 [28]

### 7.9.5 帳票レイアウト(DD.9.5)

#### 【目的】

- 帳票の大まかなレイアウトと目的を仮決めし、帳票の項目数の規模感を把握する。特に現行システムから変更が発生した帳票を中心に確認する。

#### 【説明】

- 帳票レイアウトを作成し、出力項目の過不足、表示形式、表示順序などについて、業務部門に確認する。帳票はこれまでは紙に印刷していたが電子化が進んでいる。帳票の提供形態を検討する際には、電子化を行うかを含めて検討する。

#### 【留意事項】

- 要件定義、基本設計で粒度が異なることを意識して、記述粒度を定義する。
  - 要件定義：帳票に対するステークホルダの要求として、帳票レイアウトのイメージや必要な情報（データ項目）などを明確にする。
  - 基本設計：実装する帳票として性能や規約を考慮し再設計し、ステークホルダとの合意を行う。
- 紙で出力するか、電子化するかを考える。

従来は紙媒体で実施してきた業務処理を画面を使用したワークフローで済ませ、業務効率化とペーパーレス化を実施するケースが増えてきている。業務の電子化により、業務簡素化、業務処理時間の短縮、出力紙媒体の削減の効果が得られることに加え、帳票をデータとして保管することにより、検索、加工、分析等による業務効率化（高度化）を進めることができる。
- 帳票については、独自フォーマット等による完全な電子化以外にも、コストを意識して別の選択肢を取ることも検討する。
  - 画面のハードコピーで済ませられるものはそれで済ませるくらいに割り切る。
  - ビジネスの変化に合わせて業務で必要とする情報が変化すると、業務部門からは効率化のためにこのような情報を出して欲しい、配置をこうして欲しいなどの変更要求が発生する。これに対しては、帳票データを CSV などの形式でファイルに出力し、業務部門で適切な形式に加工してもらうなどの対応を検討する。これにより、利用環境の変化に柔軟に対応することができ、帳票設計、実装にかかる工数の大幅な削減も同時に実現できる。
- 利用目的や利用シーンを想定して帳票設計をする、また用途が明確に分かる帳票名を付与する。
  - 現場での利用シーンに合わせて帳票を設計する必要がある例
    - ◆ 拠点別営業業績の管理帳票：拠点別の営業業績を、明細⇒小計⇒中計⇒総計、

## 7.9 インターフェースの確認を中心に要求の妥当性、実現性を確認する(DD.9)

という管理帳票で出力することにした。これを業務部門のマネージャに見てもらったところ、「この帳票は、現場ではまず総計を見て、例えば近畿地方の売上は前年比でどうかと確認する。前年比で下回っていると、どの支店の売上が減少しているか、その支店を商品別に見るとどうかと、掘り下げて行く。」との使用方法であり、総計⇒中計⇒小計⇒明細の順で見られるように情報の出力順番を決定する必要があることがわかった。

- ◆ 部門別／個人コード別のリストを業務部門に配布する際に、個人ごとに改ページしない形式だと、配布先の現場で個人別の切り離しが必要になるが、控えを保存する際の収納効率を高めるため、個人ごとの改ページは行わない、などの注釈があると分かりやすい。
  - 用途の分かる名称の例
 

「契約明細リスト」ではなく「当月未入金契約者抽出リスト」にすると、用途や利用シーンが分かりやすい。
- 帳票レイアウトの確認においては、現実のデータに近いデータ例で表示することにより利用者との理解の齟齬をなくすることができる。
  - 新設帳票の場合は、帳票レイアウト、項目説明をすべて記載する。
  - 既存帳票の場合は、既存帳票のレイアウト図を利用して、変更部分の説明を加える。
- お客様先など外部に提供する帳票については、要件定義の段階で帳票に表示する項目がほぼ確定するようであれば、リーガルチェック（法規違反や知財侵害等のチェック）を行う。
- なぜこの帳票レイアウトになったかの検討経緯を残す。
 

最終帳票だけが成果物として残ると、検討経緯が分からなくなるため、なぜこの名称になったかなど、検討経緯も分かるようにする。

### 【成果物イメージ】

- 帳票レイアウトに記載する主な記載項目、イメージを表 7.31、図 7.47 に示す。

表 7.31 帳票レイアウトの主な記載項目

分類	項番	記述内容	記述内容の説明
1. 構成要素	1.1	帳票 ID／名称	帳票を特定する識別子または名称を記述する
	1.2	レイアウト情報	帳票への出力項目の配置や文字のサイズ 固定文言や表の罫線などの配置



## 7.10 データの洗い出しだけでなく、標準化や管理方法も含めて仕様化する(DD.10)

「BR. ビジネス要求定義」では、概念データモデルの策定を通じてビジネス要求を実現する上で必要となるエンティティや属性の洗い出しを行った。「SR. システム化要求定義」では洗い出したエンティティに対して属性の明確化、さらに必要に応じて詳細化を行う。また、洗い出したエンティティや属性に同名異義(ホモニム)や異名同義(シノニム)があったり、同義の項目のデータ型が異なっていたりすると業務内容の認識齟齬を生み出すだけでなく、認識齟齬による漏れ、誤りを引き起こす。したがって、エンティティや属性の標準化をする必要がある。

また、各データの主管ユーザを明確にする必要もある。データは業務そのものであり、そのデータを主管する業務部門があるはずである。どの業務部門がどのデータをどのように管理するかを検討しないでいると、稼働後に不要な例外業務や特殊運用を引き起こし、ビジネス目的の達成を阻害する原因になる。

### 7.10.1 エンティティ定義書/データ項目定義書(DD.10.1)

#### 【目的】

- 業務に必要な情報を洗い出し、システム化対象の範囲をデータの視点で明確にする。特に現行システムから変更が発生したエンティティを中心に確認する。
- システム化対象の範囲で扱うデータの規模感を把握する

#### 【説明】

- エンティティの格納単位ごとのレイアウトと従属する項目内容を表したもの。

#### 【留意事項】

- 要件定義、基本設計で粒度が異なることを意識して、記述粒度を定義する。
  - 要件定義：概念データモデルの理解を補完する情報として、エンティティに属する主要なデータ項目を明示する。
  - 基本設計：概念データモデルをデータベース設計のインプットとするために、業務で必要とされる情報（データ項目）を洗い出す。
- データ項目を使用するのは業務部門であるため、主要なデータ項目については該当の業務部門が中心になってまとめる。複数部門で共同利用する場合はシステム部門が主幹業務部門を決定すると良い。

## 7.10 データの洗い出しだけでなく、標準化や管理方法も含めて仕様化する(DD.10)

- 初期値の有無、データの範囲、扱いを明確にしておく。  
(例) コード値の設定  
性別：1 (男)、2 (女) だけでなく、3 (不明)、4 (その他特記あり) など考慮  
商品種別：001～700、700～800 新商品のための予約、900 番台は使用しない、等
- 項目名称の付け方をルール化して、同名異義 (ホモニム)・異名同義 (シノニム)、等、項目名称を整理して、極力統一し、数を減らす。  
例) 「日」「YMD」「年月日」 ⇒ 「年月日」に統一  
「発令日」「着任日」「異動日」 ⇒ 「発令年月日」に統一  
「管理項目 1、管理項目 2」 ⇒ 番号つき項目禁止
- 区分、記号、コード、表示などの違いを明確に定義し、関係者間で合意する。
- 1つの項目に対して複数の意味を定義しない。
  - 部門コード4桁に、上2桁は「部」、下2桁は「課」の意味を持たせてまとめて定義している例がある。このような場合、異なる意味である部と課に強い依存関係が発生してしまうため拡張性が低くなる。部コードと課コードを別項目にすることで、拡張性が保たれる。
  - コードと数値、数量を区別する。  
数値と数量の混同例：「期間月数」には通常は月数が入るが、0000 なら未定、9999 なら無期を意味するといった例外は使用しない。この場合は、「期間区分=1:有期 2:無期 3:未定」といった区分を別途設け、3を期間区分の初期値とする、等
- データ、コードの管理部門 (データ管理者) を決めておく。
  - データ項目の変更権限や複数の関連部門にまたがるコードの主幹部署については、その管理部門とシステム部門が調整をすると、関連する業務、およびシステムへの影響を把握できる。
  - 桁数等のデータ属性、コード項目の値等を定める時は、連動先や利用者・用途などを視野に入れ、齟齬が出ないようにする。

### 【成果物イメージ】

- エンティティ定義書/データ項目定義書に記載する主な記載項目、イメージを表 7.32、図 7.48 に示す。

7.10 データの洗い出しだけでなく、標準化や管理方法も含めて仕様化する(DD.10)

表 7.32 エンティティ定義書/データ項目定義書の主な記載項目

分類	項番	記述内容	記述内容の説明
0. 共通情報	—	プロジェクト名、システム名、工程名、ドキュメント ID、ドキュメント名、作成者、作成日付、バージョン、更新者、更新日付	
1. 書誌情報	1.1	エンティティ定義の ID	エンティティ定義を一意に識別するためのコード
	1.2	エンティティ定義の名称	エンティティ定義の名称
	1.3	概要	エンティティ定義の範囲、目的
2. 構成要素	2.1	属性名	エンティティの情報項目、属性名
	2.2	型	数値、テキストなどの属性の型
	2.3	属性説明	属性の説明
	2.4	備考	長さ（属性のバイト数など）、精度（データ精度）、必須（必須データか否か）、主キー（主キーか否か）など

エンティティ定義ID		エンティティ定義名称					
概要							
No.	属性名	型	属性説明	長さ	精度	必須	主キー
1	製品番号	数値型	製品ごとに振られた識別子	10	0	Y	1
2	製品名	テキスト型	製品名称	255	0	Y	
3	製品単価	数値型	製品の単価	8	0	Y	
4	製品分類番号 (FK)	数値型	この製品の分類番号	10	0	Y	

図 7.48 エンティティ定義書/データ項目定義書のイメージ

(出典) IPA 「機能要件の合意形成ガイド」 [28]



7.10 データの洗い出しだけでなく、標準化や管理方法も含めて仕様化する(DD.10)

## 7.10.2 ドメイン定義書(DD.10.2)

### 【目的】

- データ項目の定義、取りうる値の範囲（定義域）を明確にする。特に現行システムから変更が発生したドメインを中心に確認する。

### 【説明】

- データ項目のグループ単位ごとの名称、型、形式、値域を表したものの。

### 【留意事項】

- ドメインを適切に管理することでデータ項目の保守性を高めることができる。一方で手作業による管理は煩雑であり、二重管理に陥るなど保守性の向上を期待することが難しくなる。したがって、ドメインをサポートしたモデリングツールを利用することが望まれる。

### 【成果物イメージ】

- ドメイン定義書に記載する主な記載項目、イメージを表 7.33、図 7.49 に示す。

表 7.33 ドメイン定義書の主な記載項目

分類	項番	記述内容	記述内容の説明
1. 構成要素	1.1	ドメイン定義の ID	定義を一意に識別するためのコード
	1.2	コード/非コード	ドメインを「コード系」と「非コード系」のいずれかに分類する
	1.3	分類	ドメインを分類して管理する場合は分類記述する
	1.4	ドメイン名	ドメインの名称を記述する
	1.5	型	ドメインの型と桁数、文字数を記述する
	1.6	説明	ドメインの意味、用途など記述する
	1.7	記述形式	ドメインの取る値の形式および補足説明を記述する
	1.8	最小値	値を列挙できないが最小値を規定できる場合は記述する
	1.9	最大値	値を列挙できないが最大値を規定できる場合は記述する
	1.10	既定値	既定値がある場合は記述する

7.10 データの洗い出しだけでなく、標準化や管理方法も含めて仕様化する(DD.10)

ID	コード/ 非コード	分類1	分類2	ドメイン名	型		説明	記述形式	最小値	最大値	デフォルト値
					データ型	桁数 文字数					
C001	コード			企画コード	文字型	4	ツアー基本情報を識別するコードを示す。		-	-	-
C002	非コード			企画日	日付型	-	年月日を示す。		-	-	-
C003	コード			企画番号	文字型	4	企画情報を識別するコードを示す。		-	-	-
C004	非コード			企画名称	文字型	300	ツアーの名称を示す。		-	-	-
C005	コード			出発地コード	文字型	2	出発地を識別するコードを示す。	JIS X 0401の都道府県コードに従う。	0	47	-
C006	コード			目的地コード	文字型	2	目的地を識別するコードを示す。	JIS X 0401の都道府県コードに従う。	0	47	-

図 7.49 ドメイン定義書のイメージ

7.10 データの洗い出しだけでなく、標準化や管理方法も含めて仕様化する(DD.10)

### 7.10.3 コード体系定義書／コード内容定義書(DD.10.3)

#### 【目的】

- 桁ごとに意味を持つコードの定義を標準化する。特に現行システムから変更が発生したコードを中心に確認する。

#### 【説明】

- コードの構成、ならびに桁別の内容、文字種別、値域を定義する。

#### 【留意事項】

- 桁別に意味を持たせる場合には、運用時に桁数が枯渇することがないように、今後のデータ量を見据えた定義をする必要がある。また、開発においてコードの内容を判別して業務ロジックを作り込む必要があり、障害の温床になるケースがある。したがって、複数のコードを連結した単一のコードを定義するのではなく、個々のコードに分割するなど桁別に意味を持たせない定義をすることが望ましい。

#### 【成果物イメージ】

- コード体系定義書／コード内容定義書に記載する主な記載項目、イメージを表 7.34、図 7.50 に示す。

表 7.34 コード体系定義書／コード内容定義書の主な記載項目

分類	項番	記述内容	記述内容の説明
1. 書誌情報	1.1	ドメイン ID	コードに対応するドメイン ID を記述する
	1.2	ドメイン名	コードに対応するドメイン名を記述する
	1.3	コード説明	コードの意味、用途などを記述する
	1.4	構成	コードの各項目の区切り位置を記述する
2. 構成要素	2.1	項番	コードの各項目に対応する連番を記述する
	2.2	内容	コードの各項目の内容を簡潔な単語で記述する
	2.3	文字種別	桁別コードの各項目で利用するすべての文字種別を記載する
	2.4	桁数	コードの各項目の桁数を記述する
	2.5	桁説明およびコード値	コードの各項目の意味説明を記述する

7.10 データの洗い出しだけでなく、標準化や管理方法も含めて仕様化する(DD.10)

ドメインID		D0001														
ドメイン名		社員コード														
桁別コード説明		社員をシステム内で一意に識別するコード。正社員に対してのみ付与する。														
構成	桁	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	X ₁₀	X ₁₁	X ₁₂	X ₁₃	X ₁₄	X ₁₅
	項番	1	2			3										

項番	内容	文字種別	桁数	桁説明およびコード値
1	採用地域	数字	1	社員が採用された地域を示す。 1:関東/2:関西/3:その他
2	採用年度	数字	4	社員が採用された年度を西暦4桁で示す。 最小値:1960/最大値:9999
3	連番	数字	5	[採用年度]内の連番。採用地域に関係なく、通し番号で付与する。 最小値:00001/最大値:99999

図 7.50 コード体系定義書/コード内容定義書のイメージ

## 7.11 機能とデータとの間の整合性を検証する (DD.11)

ビジネスプロセスならびにその実現手段であるシステム機能はデータのライフサイクル（作成～照会～更新～削除）と一貫性を保たなければならない。主要なデータがどこで作成され（C:Create、作成）、どこで使用され（R:Read、照会）、どこで更新され（U:Update、更新）、どこで削除される（D>Delete、削除）かを表す CRUD 図を作成する。CRUD 図の作成を通して、主要データのライフサイクルと機能との間の関係性を確認し、データや機能の抜け漏れ、矛盾を検証する。

CRUD 図記載の粒度にはエンティティレベルとデータ項目レベルがある。エンティティレベルの CRUD 図は、主に CRUD 図に記載すべきエンティティや機能の抜け漏れ、エンティティと機能との関連性の矛盾を検証するために作成する。一方でデータ項目レベルの CRUD 図は、データ管理上の矛盾を検証するために作成する。表 7.35 に CRUD 図作成を通して検証できるエンティティ、データ項目、機能の抜け漏れ、矛盾の例を示す。

表 7.35 CRUD 図作成により検証できるエンティティ、データ項目、機能の抜け漏れ、矛盾の例

記載粒度	抜け漏れ／矛盾	例
エンティティ	機能の抜け漏れ	あるエンティティを更新、削除する機能があるにも関わらず、登録する機能がない
	エンティティの抜け漏れ	更新機能であるにも関わらず、更新・削除対象のエンティティがない
	矛盾	先行して実行される機能で参照しているエンティティが後続して実行される機能で登録されている
データ項目	矛盾	先行して実行される機能で更新しているデータ項目が後続して実行される機能で登録されている

### 7.11.1 CRUD 図(DD.11.1)

#### 【目的】

- どのシステム化業務が、どのエンティティのインスタンスを作成(C:Create)、参照(R:Read)、更新(U:Update)、削除(D>Delete)しているかを明確にし、システム化業務の抜け漏れ・矛盾を除去する。

#### 【説明】

- エンティティのインスタンスの作成、参照、更新、削除をエンティティとシステム化業務とのマトリックスで表現したもの。

#### 【留意事項】

- 要件定義、基本設計で粒度が異なることを意識して、記述粒度を定義する。
  - 要件定義：業務フローやシステム化業務フローと概念データモデルを関連付けて、業務フローやシステム化業務フローの漏れや矛盾、概念データモデルのエンティティやデータ項目の漏れや矛盾を発見し是正するために作成する。また、データの発生源（データの所管部門）を明確化する。
  - 基本設計：どのシステム機能によって、どのエンティティやデータ項目が生成・参照・更新・削除されるのかを明確にし、システム機能の漏れや不整合をなくす。
- 要件定義段階で、主要データについて CRUD 図を作成するのは、情報を登録・参照・更新・削除する機能の配置を確認し、機能面、データ面での抜け漏れを検証するためである。すべてのデータについて CRUD 図を作成する必要はなく、企業活動の主要データや複数の機能から更新されるような項目だけで良い。

例として、受注システムにおける主要なデータとは「受注」などであり、複数の機能から更新されるような項目とは「数量」「金額」「主要仕様」などである。受注後、生産を開始する時点までシステムで変更を受け付けることを認めるかどうかなど、エンティティレベルやデータレベルの情報のライフサイクルについては概念データモデルなどで確認する。
- 項目を作成、更新するタイミング(更新してはいけないタイミング)を整理した上で、制約がないかどうかを確認する。D(削除)については、業務的な削除(論理削除)とシステム的な削除(物理削除)がある。要件定義ではデータの業務的なライフサイクルを確認することが目的であるため、業務的な削除を考慮して検討する。
- データ項目を正しく定義する。

データ項目レベルで CRUD 図を書くことにより、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。

## 7.11 機能とデータとの間の整合性を検証する(DD.11)

### 【成果物イメージ】

- CRUD 図に記載する主な記載項目、イメージを表 7.36、図 7.51 に示す。

表 7.36 CRUD 図の主な記載項目

分類	項番	記述内容	記述内容の説明
1. 書誌情報	1.1	CRUD 図の ID	CRUD 図を一意に識別するためのコード
	1.2	CRUD 図の名称	CRUD 図の名称
	1.3	概要	CRUD 図の範囲、目的
2. 構成要素	2.1	エンティティ	エンティティ名
	2.2	システム化業務	システムの機能（または業務、組織、画面）など
	2.3	CRUD	エンティティのインスタンスの作成 (C:Create)、参照 (R:Read)、更新 (U:Update)、削除 (D>Delete)

CRUD図ID	CRUD図名称	概要																																		
		<table border="1"> <thead> <tr> <th colspan="2" rowspan="2"></th> <th colspan="4">エンティティ</th> </tr> <tr> <th>顧客</th> <th>製品</th> <th>受注</th> <th>受注明細</th> </tr> </thead> <tbody> <tr> <th rowspan="4">機能</th> <th>顧客登録</th> <td>C</td> <td></td> <td></td> <td></td> </tr> <tr> <th>顧客検索</th> <td>R</td> <td></td> <td></td> <td></td> </tr> <tr> <th>受注登録</th> <td>R</td> <td>R</td> <td>C U</td> <td>C</td> </tr> <tr> <th>受注検索</th> <td></td> <td>R</td> <td>R</td> <td>R</td> </tr> </tbody> </table>						エンティティ				顧客	製品	受注	受注明細	機能	顧客登録	C				顧客検索	R				受注登録	R	R	C U	C	受注検索		R	R	R
		エンティティ																																		
		顧客	製品	受注	受注明細																															
機能	顧客登録	C																																		
	顧客検索	R																																		
	受注登録	R	R	C U	C																															
	受注検索		R	R	R																															

図 7.51 CRUD 図 (エンティティレベル) のイメージ

(出典) IPA 「機能要件の合意形成ガイド」 [28]

## コラム 12 要件定義における CRUD 図の役割

## 【悩み】

「CRUD 図は作成するのに時間を要するので作成しない」というプロジェクトは少なくない。模造紙のような大きな紙に定規を当てながら苦勞して CRUD 図を作成したことはないだろうか。100 プロセス×1,000 項目のデータのマトリクスなら 100,000 の要素が存在し、C,R,U,D のどれも入らないのが正しいのかを含めてすべての要素を確認しなければならないが、これは容易なことではない。

## 【解決案】

要件定義において CRUD 図を何のために作成するのかを明確にしてから作成する。

設計工程では機能とデータの整合性の検証を目的として大きなマトリクスで検証をすることがあるが、要件定義における CRUD 図作成の目的は、新しいビジネスプロセスが業務として成立しているかどうかの検証である。

要件定義では、業務の実世界を業務フローや概念データモデルを使ってモデル化し、可視化された業務の上で、課題分析や新しい業務の検討、業務の理解・共通認識を行う。業務フローなどは機能モデル、概念データモデルはデータモデルになり、機能とデータの関係は下図に表すように、クロスするような関係になる。

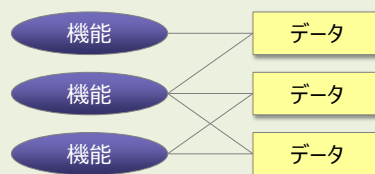


図 7.52 機能とデータの関係

CRUD 図は、この両者の関係を表形式でモデル化したものであり、相互関係のモデルなどと言われている。機能とデータの相互関係のモデル化は、状態遷移図などでも実施できるので、状態遷移図を用いる方法もある。

CRUD 図の利点は、縦軸、横軸の各要素を軸に網羅的に内容を検証できることである。縦軸、横軸を検証の目的に合わせて選択することにより、マトリクスを必要最小限に絞り込んで機能とデータの相互関係をモデル化することが、CRUD 図作成のコツになる。要件定義での軸の候補としては、以下の2つが考えられる。

- データ側：エンティティ、主要なデータ項目
- 機能側：ビジネスプロセス、システム化業務



## 7.11 機能とデータとの間の整合性を検証する(DD.11)

横軸はプロセスであり、大きなレベルではビジネスプロセスになり、詳細なレベルではシステム機能になる。データのライフサイクルが長い、例えば企業全体に及ぶような場合は、大きなプロセスでライフサイクル全体を可視化できるようにビジネスプロセスを横軸に選択した方が良い。

縦軸はエンティティにするか、データ項目にするかの選択になる。エンティティはデータ項目に比べると圧倒的に数が少ないのでエンティティを縦軸に選択した方が良い。しかし、作成者がエンティティを正確に理解できていない場合が少なくない。このような場合は、主要なデータ項目を厳選し、縦軸に設定すると良い。具体的な項目で検証できるので理解しやすい。また、データ項目で CRUD 図を作成することでエンティティ自体を明確化することができる。

縦軸（プロセス）、横軸（データ）のレベルが決定したとして、プロセス、データの双方を漏れなく列挙し CRUD 図を作成しなければならないのか、という疑問が次に発生する。冒頭に述べたように、CRUD 図作成の目的は業務フローやシステム化業務フローの漏れや矛盾、概念データモデルのエンティティやデータ項目の漏れや矛盾を発見することであり、すべてを漏れなく検証しなくても良い。例えば、オーダーを中心とした販売、製造、倉庫、出荷、請求業務の全体最適システムへの見直しが課題なら、それに関係した部分だけを検証できれば良い。課題が複数あれば、無理に 1 枚の CRUD 図にせず、課題ごとに分けても良い。全業務を検証できるわけではないが、このように的を絞って検証するのも一つの方法である。

また、CRUD 図は、例えば以下のようなテーマを設定して、それが検証できるビジネスプロセスやデータに絞って作成するのも良い。

- 「オーダーのビジビリティの追求」
- 「販売計画から生産計画の立案から予実管理」など

すべての業務を検証できるわけではないが、このように的を絞って検証するのも一つの方法である。

本文にも述べたように、業務フローやシステム化業務フローの機能の抜け漏れや、概念データモデルのエンティティの抜け漏れ、ビジネスプロセスとデータの関係の矛盾を発見できることが CRUD 図作成の効果である。どのように作成すれば効果が得られるかを検討した上で、プロジェクトに必要な十分な範囲で CRUD 図を作成し、有効に活用していただきたい。

## 7.12 十分に妥当な非機能要求を定義する (DD.12)

近年、非機能要求の重要性が増してきており、経営や業務の重要な関心事となっている。もともと非機能要求には技術的な要素が強く、これまではシステムの専門家が設計工程に入ってから苦労しながら決定することが多かった。しかしながら今日では、非機能要件は要件定義工程において経営層、業務部門、システム部門やベンダ企業が協力して決定することが重要視されてきている。

非機能要求を定義する際には、IPA が公開している「非機能要求グレード 2018 (以下、非機能要求グレード)」 [13] が役に立つ。非機能要求グレードでは、非機能に対する要件定義が 238 種類の項目に分類され、それぞれに目標値/指標値 (メトリクス) が設定されている。示された項目のメトリクスを決めれば良いので、非常にシンプルで分かりやすい。経験の少ない人でも、漏れなく実施しようとしたときに役立つ。基本は、この非機能要求グレードを有効利用していただきたい。

なお、非機能要求グレードを適用する際には、技術的な設定項目も多いので、経営部門や業務部門などと合意を形成し、効率的に進め、仕様化していくための工夫が必要になる。

### 7.12.1 非機能要件書(DD.12.1)

#### 【目的】

- 現行システムですでに実現している要求や、新システムで新たに求められる可用性や性能・拡張性などの非機能要求を明らかにする。特に、コスト (見積り) への影響が大きい事項や、現行システムと新システムで要求が異なる非機能要求の意識合わせを行う。
- コスト (見積り) 感をユーザ部門と共有し概算見積りを出せるようにする。

#### 【説明】

- 当該プロジェクトの要件定義工程で決めておくべき非機能要件を決める。

#### 【留意事項】

非機能要求グレードをベースにした要件定義の実施上の工夫ポイントを述べる。

ポイントの 1 つは、要件定義工程で経営層、業務部門、システム部門やベンダ企業などがいかに協調して合意形成していくかである。方針を確認し、経営層や業務部門から引き出すべき要求を明確にし、そこからメトリクスにつなげていくという工夫が必要である。

## 7.12 十分に妥当な非機能要求を定義する(DD.12)

2 つ目は、非機能要求のあいまい性や矛盾をなくすことである。性能の要求を聞くと、「早いに越したことはない」などと言われる。とかく非機能要求はオーバースペックになりがちだ。また、セキュリティ強化のために暗号化を適用すれば、利便性が低下する。こうしたトレードオフ関係、優先順位関係に決着をつけなければならないものもある。

### (1) 非機能要求グレードを活用する

まずは、IPA が公開している非機能要求グレードの活用を検討していただきたい。非機能要求グレードのイメージを図 7.53 に示す。

(注) 詳細は IPA のホームページを参照 [13]

大項目	可用性	性能・拡張性	運用・保守性	移行性	セキュリティ	システム環境・エコロジー
「項目一覧」 238のメトリクス						

図 7.53 非機能要求グレードのイメージ

非機能要求グレードは、可用性、性能・拡張性、運用・保守性、移行性、セキュリティ、システム環境・エコロジーの6領域に対し、要件定義工程で明確にすべき238のメトリクスを提示している。

例えば、運用時間（通常）というメトリクスでは、

レベル0：規定なし

レベル1：定時内（9時～17時）

レベル2：夜間のみ停止（9時～21時）

レベル3：1時間程度の停止あり（9時～翌朝8時）

レベル4：若干の停止あり（9時～翌朝8時55分）

レベル5：24時間無停止

## 7.12 十分に妥当な非機能要求を定義する(DD.12)

というガイドを示し、決定を導いている。

非機能要求グレードは、ただ単に 238 の項目を定義せよではなく、以下に示すような扱い方を提示してくれているのが特徴である。

### ① モデルシステムの参照

実現したいシステムのイメージに最も近いモデルシステムを選択することにより、設定したい項目に対するメトリクス的一般値を引用できる。

### ② 段階的詳細化

検討の順番や重要項目を示す樹系図が用意されている。重要項目では、品質やコストに大きな影響を与える事項を明示している。

## (2) 基本的な情報を要求として引き出す

工夫ポイントの1つ目は、経営層、業務部門、システム部門やベンダ企業などがいかに協調して合意形成していくかである。

機能要件の定義は、利用部門が主体で実施することが基本であるが、非機能要件の定義には、システム部門が決定するメトリクスもある。協調して合意形成していく際には、経営層や業務部門がそれぞれどの項目に強い関心を持つかを識別する必要がある。そうすることにより、どのステークホルダからどの非機能要求を抽出し合意形成すべきかが明確になる。また、非機能要求は、トレードオフ関係になるものがあるので、どちらを優先するかを判断をしなければならない。コストだけではなく、ビジネス目的達成を考慮した判断をするためにも以下の選別は役立つ。

### ① 非機能グレードの項目をステークホルダ別に分けよ

合意形成しなければならないステークホルダによって、非機能要求の項目を振り分ける。まず、ステークホルダによって大きく3つの領域を設定する。

- a) ビジネス領域：経営層や利用部門の関心の高い領域。ビジネスとして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。
- b) 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。
- c) 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。

次に、各々の領域に非機能要求の項目を振り分ける。項目の種類を振り分けた例を表 7.37 に示す。

## 7.12 十分に妥当な非機能要求を定義する(DD.12)

表 7.37 非機能要求の振分け例

領域	大項目	中項目	備考
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	非機能要求グレードに追加
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	非機能要求グレードに追加
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	非機能要求グレードに追加

このように、同じ可用性でも、詳細内容によって合意形成をすべきステークホルダが異なることが分かる。また、非機能要求グレードには、非機能要求につながるビジネス要求／基本情報や、昨今非機能として注目されている、ユーザビリティやソフトウェア保守性・移植性なども追加されている。

### ② 聞きだす要求を明確にする

上表のように3つの領域に振り分けても、238項目を提示してメトリクスを決めていくのは困難であることから、基本的な「要求」を抽出するところから出発するのが良い。ここでは、最終的に確定したメトリクスなどを「要件」と呼び、その元になる段階のものを「要求」と呼ぶ。各々の領域に要求抽出シートを用意しておくが良い。要求抽出シートとは、非機能のメトリクスを決める際に基本情報や重要な意思表示を抽出するためのものである。例えば、「セキュリティ」の「リスク分析範囲」では、個人情報の流出が検知できる対策を講ずるかどうかの意思

## 7.12 十分に妥当な非機能要求を定義する(DD.12)

(要求)を聞き出すにとどめる。ソリューション領域も同様に、例えば、情報セキュリティに対しては、第三者による診断を実施したいかどうかの意思(要求)を聞き出すにとどめる。その後でネットワーク診断、Web診断、DB診断はどうするかというようにメトリクス(要件)として具体化していく。要求を抽出することにより、その中で、優先順位や矛盾やトレードオフ関係を検討できるようになり、238項目ものメトリクスを直接決めていく方法より効率的に進めていけるようになる。なお、サービス利用者数等、非機能要求のメトリクスそのものでもある基本情報もビジネス領域に位置付けており、ここで抽出する。このような「要求」レベルで確認するためのシートを準備しておくが良い。

表 7.38 にビジネス領域の要求抽出シートの例を示す。

表 7.38 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

要求の抽出は、ヒアリングにより行うことが基本だ。ただし、セキュリティや業務継続などは、システム開発プロジェクト以前に企業としての方針が設定されている。方針書やガイド等を入手し理解し、そこから抽出することも必要になる。

### (3) 矛盾、トレードオフを解消する

#### ① 制約や前提条件を明確にする

法令、商慣習、予算や特定技術採用、技術人材、再利用素材などビジネス上の制約や技術的な制約を明確にしておく。また、未確定の要素があれば仮定し前提条

## 7.12 十分に妥当な非機能要求を定義する(DD.12)

件として設定して検討する。

### ② 現状を正確に把握する

現状から改善したいというのが要求のベースであり、どのくらい良くなったかがシステム化の成果である。まず現在の状況を正確に把握することが重要である。一気にジャンプし過ぎて現場がついていけないという事にも注意が必要になる。段階的なステップアップも検討要素に入れなければならないし、逆もある。現行と同じで良いと言われたのに、現行より悪くなることもある。Web化したら性能が悪化するというのは、システム部門側は仕方ないと思うかもしれないが、業務部門にとっては現状より悪化することは許せないはずである。現状を把握して判断することが重要である。

### ③ 優先順位を判断する

制約や前提条件、現在の状況などをもとに優先順位を決定する。ビジネス領域の要求が優先順位の高い要求であると判断すると良い。昨今、非機能要求はビジネス要求達成のための重要なファクタになってきている事は前述のとおりである。

### ④ 要求のレベルでトレードオフ関係を判断する

非機能要求相互の依存関係を明らかにし、依存関係があるものについて、矛盾した要求がないかを確認する必要がある。非機能要求の間にトレードオフがある場合には、要求度や優先順位をもとに、どちらの要求を優先して採用するかを判断する。優先順位付けには、業務部門だけでなく、経理部門、管理部門、法務部門など間接部門への確認も必要かを見極めると良い。

### ⑤ 一律にメトリクスを決めない

性能要求などでは、目標値が1秒であるが遵守率90%というような決め方や、ある処理だけは3秒を目標とするなど、条件に応じたメトリクスを決める必要がある。一律の値に決定しないことである。業務復旧やセキュリティなども、完全に社内に閉じたシステムや、基幹システムではないものにまで一律に高い水準を求めないことである。システムによっては贅沢な機能になってしまう。

### ⑥ 実際に体感してもらう

性能などは、余裕があれば体感してもらう。現行業務のレスポンスが3秒で、目標値が1秒であったとしても、実際には2秒で十分速いと感じることがある。利用者に体感してもらった上で合意形成を取るということも効果がある。難易度の高い方式を検討するよりリスクは下がる。

## (4) 関心が高い非機能要求設定に関するポイント

### ① 可用性

BCP(事業継続計画)とは、企業が自然災害、事故、テロ等の予期せぬ緊急事態に遭遇した場合に、重要業務に対する被害を最小限にとどめ、最低限の事業活動の

## 7.12 十分に妥当な非機能要求を定義する(DD.12)

継続、早期復旧を行うために事前に策定する行動計画である。大きな震災やテロなどが身近にあるという認識が強まり関心を高めている。電力や通信、金融など、社会的影響が大きい業種だけでなく、製造業などでも業務停止による影響を重要課題として捉え、BCPを設定する企業が増加している。

これらの要求は、非機能要求グレードの「可用性」「継続性」の領域の中に「業務継続性」／「目標復旧水準」などとして位置付けられている。

### a) 稼働率、稼働品質の評価指標を定義する

システムの稼働率、稼働品質には、表 7.39 に示すようにいくつかの評価指標がある。重要インフラシステムか、基幹業務システムかの違いや、そのシステムの特質により、稼働率、稼働品質の評価基準は異なるため、表 7.39 のような表を準備し要件定義の段階で確認する必要がある。

表 7.39 稼働率、稼働品質

区分	評価項目	評価式	評価	参考目標
稼働率	稼働率	実績稼働時間/ 計画稼働時間	1に近いほど良い	99.999%(5分停止/年)以上のような厳しい目標の場合は特別な対策が必要(99.5%程度は通常確保)
	延べ稼働率	(延べ時間-計画停止時間-障害停止時間)/ 延べ時間	1に近いほど良い	顧客の使用可能時間の評価(99.95%以下になると不満が出るなど)
稼働品質	業務停止回数	業務停止回数/ 年	0に近いほど良い	基幹業務システムは0.06件/1年間で運用費程度は確保される
	規定時間外停止回数	規定時間以上に停止した回数/ 年	0に近いほど良い	重要インフラシステムで特に重視(15分以上の停止は0回/年が目標)
	オンライン平均応答時間	規定内応答回数/ 全応答回数	1に近いほど良い	顧客からの直接受注入力など主要な入力データに限って管理するのが良い
	バッチ処理異常終了率	異常処理回数/ 年	0に近いほど良い	障害報告の迅速性などの目標も必要

### b) 長時間のシステム停止を考慮する

昨今は、システム障害などによるシステム停止時にどう対応するかを決めるだけでは大きな災害に対応できないことがわかってきた。大きな災害の発生を考慮するなら、1週間や1カ月という長期間のシステム停止時にどうするかまで検討しておかなければならない。

### c) 運用を考慮する

バックアップシステムを遠隔地に配備したとしても、そこで業務を遂行するた



## 7.12 十分に妥当な非機能要求を定義する(DD.12)

めの運用体制が確立できなければ業務が停止したままになる。バックアップシステムでの運用方法や体制も検討しておく必要がある。

### d) 訓練する

運用方法を立案しても、実際には機能しない場合がある。人が中心になる場合には訓練を実施して運用の見直しも同時に行う必要がある。

## ② セキュリティ

インターネットの普及とともに、情報の盗難やコンピュータシステムの破壊などの犯罪が増加している。システム停止だけでなく、もし情報漏えいなどの事件に発展した場合には企業として大きな責任問題になる。経営的なダメージも大きくなることから、セキュリティの確保は企業としての重要課題である。

これらの要求は、非機能要求グレードの「セキュリティ」領域に多数記載されている。セキュリティは非機能要求として以前から強く認識されている項目だが、昨今は、セキュリティ犯罪の多様化への対応に苦慮しているのが実情である。

### a) 入口対策／内部対策／出口対策の視点で検討する

i) 利用者の認証、権限管理、アクセス権管理、ログ管理、暗号化機能などの要求を明確にする。

ii) データや業務へのアクセス権限を適切に設定し、許可されていない者がアクセスできないようにする。万が一不正アクセスが発生した場合には、情報漏洩や妨害行為などの事象を引き起こされた事実が確認できる仕組みにしておきたい。悪意のある相手によるデータの窃盗や改ざんに対し、これを防止してデータが健全な状態を保つ対策が必要である。これには専門的な知識が要求されるため、場合によっては外部の専門機関の活用を検討する。

iii) 業務の監査ログ、証跡の保持、各種アクセスログの採取と保持、ユーザ ID とパスワード以外により利用者本人であることの認証方法についても、検討する必要がある。

### b) 守り以外も考慮する

ガード、検知、ログ／証跡、分析、対処、教育といった、プロセスの全体を見通した設計を行わなければならない。守りきれないことを想定した対処方法についても検討が必要である。

### c) 対組織だけでなく、対個人に備える

組織として仕組みを作るだけでは、アクセス権のある社内の人員による悪意を持った攻撃は防ぎきれない。個人の教育やモラルの向上などの取り組みも必要である。

### d) IPA の情報セキュリティ読本などを参考にする

IPA セキュリティセンターでは、情報セキュリティ強化に対する対策として、

## 7.12 十分に妥当な非機能要求を定義する(DD.12)

「情報セキュリティ読本」 [35]や「情報セキュリティ教本」 [36]などを出版している。またその他にも、ホームページで常に最新情報を発信しているので、ぜひ参考にさせていただきたい。

### ③ ユーザビリティ

最近、ユーザビリティに対する関心が高まっている。ユーザビリティとは、「使いやすさ」のことであり、ITシステムの画面に例えると、色や配置の工夫等による見やすさ、操作性などでのことである。従来のシステム開発は目的の機能を実現することが最優先で、システムの使い方は、人が訓練して習得すれば良いという発想であった。しかし、インターネットやスマートフォンなどの普及にともなうITシステムが直接ユーザと触れることが多くなってきたことにより、機能が備わっていてもユーザビリティを考慮していなければ、ユーザに見放されても仕方ない時代に、さらにはユーザビリティだけでは競争優位を築けない時代になってきている。「楽しさ、驚き、心地よさ」というUX (User Experience) が脚光を浴びてきている。

これらの要求は、残念ながら現行の非機能要求グレードには記載されていない。非機能要求グレードの基本は「製品品質」であり、「製品品質」はISO/IEC25010 [12]における「品質モデル」の中で定義されているが、ISO/IEC25010には品質モデルとして「製品品質」だけでなく「利用時品質」が追加された。「利用時品質」とは、「有効性」、「効率性」、「満足性 (実用性、快感性、快適性)」などのことである。ここでは製品 (システムやソフトウェア) 自体の品質だけでなく、それを利用した際の品質も意識する必要があると説明されている。なお、本ガイドではこれら利用時品質の向上方法については割愛する。

#### a) ユーザビリティ、UXを考慮せよ。

ユーザビリティの設計方法やUX要求を獲得する手法、人間中心設計などについては、一般書物として出版されていたり、セミナーが開催されたりしているので、そちらを参考にすると良い。

### ④ 性能

#### a) 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

##### i) 条件による性能要求の違いを考慮する

画面レスポンスは「一律2秒以内」などとまとめ過ぎてはならない。性能は条件によって異なる。利用者数やデータ量などによって異なるとともに、一日の時間帯や一年の時期などによっても異なる。ピーク時を見つけ出し、その検証と合意形成を行うことが重要である。また、「1分間に最大500件の入力があった場合に、2秒以下で返答できる確率を95%にする」などと、条件

## 7.12 十分に妥当な非機能要求を定義する(DD.12)

と要求レベルを定義する。

主なデータ量として以下が挙げられる。

- 業務量：利用ユーザ数（稼働当初は最大）、商品数（稼働当初は最大）を設定した上で、稼働当初に予想される最大量（システムのライフサイクルを考えた最大量）
- トランザクション量（入力量）：平常時、ピーク時、緊急時、障害時に予想される量
- データ保存量：稼働当初に予想される最大量を確認する。

詳細は非機能要求グレードの項目を参照願う。

### ii) 将来の状態を予測する

上記のようなデータを見極めるとき、現状のデータだけでは問題がある。将来まで使用するシステムとして、稼働時や1年後、2年後といったデータで判断しなければならない。将来のデータ量を予測するためにも、能力不足による障害を現システムで発生させないためにも、日ごろの運用で、定期的なデータ量の測定を実施し、システムの利用状況の変化を先読みしておくことが大切である。

### iii) 前提としているハードウェア構成を明確にする

性能要求を検討する際には、ハードウェア構成・スペックの前提を明確にした上で検討する。前提が変更になった場合は、性能要求の見直し、あるいは、性能要求をそのままとするのであれば、アプリケーション設計の見直しを行う。また、他の業務とサーバー、ネットワークを共同使用する場合、他の業務による性能への影響の条件がある。他の業務のスケジュールや性能実績を参考に性能要求を検討することもできるが、実際の評価は実稼働環境での検証が必要であることを、ユーザもベンダも理解しておくことが必要である。

### iv) 想定外を検討する

性能はピーク時などの最大量を想定して検討するが、それでも想定外の事態が発生した際の対応方法を検討しておく。これは、性能だけに言えることではない。

### v) 当たり前の思い込みによる齟齬を払拭する

インターネットの普及にともない、Web画面を使用するオンラインシステムが多くなっている。システム部門はWebになったら速度が遅くなるのは当たり前だと思っているが、利用部門は違う。現行システムと同等の速度は当たり前で、むしろ速くなると思っている。この齟齬は、特に現行システムと機能的には変わらない時によく発生する。変更する業務の要件定義に注力し、変更しない部分の要件定義は「現行と同じ」で終わらせると、機能要件は問題なくとも、非機能要件が意図せず変わってしまっていることがあるので注

## 7.12 十分に妥当な非機能要求を定義する(DD.12)

意しなければならない。

## b) 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

## i) 現行システムと比較してみる。

レスポンスタイムを1秒以内と決定したとして、それは本当に達成が必要な要求なのだろうか。「早い」に越したことはないが、比較対象の現行システムや他のシステムでは現状どの程度のレスポンスで運用されているのかを確認してみることが必要である。例えば、現行システムのレスポンスタイムが5秒ぐらいであり、性能改善が3秒に止まったとしても十分に満足度が得られるのであれば、3秒を目標値と設定することは、性能を出す難易度が高い場合の合意形成方法の一つとして有効である。

## ii) 体感する。

オンラインの処理性能は感覚による評価によるところが大きいいため、実際に速度を体感できる環境を整えると良い。いわゆる、プロトタイピングである。すなわち要件定義工程でプロトタイピングを実施し、体感してもらう。要件定義の計画時にプロトタイピングの実施を意識した進め方、体制を盛り込んでおくことが重要である。

## 2) 運用・保守性

## a) 開発容易性も検討する

システム改修が自分たちでできない、あるいは少しの改修に高い費用がかかるといった問題はよくある。例えば、「受注登録画面に新たな入力項目を追加したい」など、稼働後の機能追加要求は必ず発生する。これらの要求に、短時間、低コストで対応するためには、最初からプログラムを簡単に修正できるようにしておくことが重要であり、そのためにはプログラムの開発方法にまで踏み込む必要がある。要件定義において、保守性を向上させる手段として、適当な開発ツールやフレームワークの導入を検討するのも有効である。最近の開発ツールには保守ドキュメントを自動的に出力できる製品も多い。

## 3) 移植性

## a) 移植性を検討する上でのポイント

i) 既存のサーバーに相乗りする場合は、対象サーバーの制約を確認する。

ii) バックアップを含めた総コストを見積もり、投資判断に加えることも重要である。

iii) クラウド活用のシステム構成の場合、オンプレミスのシステム構成とシステムの挙動が変わってしまい、思わぬ障害に遭遇することがある。事前にクラウド環境のシステム動作に慣れておくことが望ましい。

## 7.12 十分に妥当な非機能要求を定義する(DD.12)

- iv) ハードウェアや OS、データベースなどのミドルウェアは保守期間満了に伴うバージョンアップが避けられない。その場合もアプリケーションは変更せずに済ませられるようアプリケーションへの影響を見極める。
- v) 国内で作成したシステムを海外展開する場合も、言語や OS の違いを吸収できるような考慮が必要となる。特に、機能が特定のハードウェアや OS に大きく依存すると移植性が下がる。できるだけ、世の中に普及している標準的な技術を組み合わせるのが好ましい。

### (5) 企業内の標準的な非機能要求基準を定める

企業内のシステム環境や設備を共通化している場合、非機能要求レベルを標準化することができる。個々の要求のレベルは、標準のレベルをもとに、業務特性、ユーザ特性に基づいて調整をすると良い。顧客ニーズの変化、技術進歩により、非機能要求や実現費用は変わるため、基準となるレベルは定期的に見直すこと。基準を見直す時は、現有の共通基盤などだけでなく、自社業務の本来の要求をもとに実施する。

#### 【成果物イメージ】

- 非機能要求グレードを参照されたい。

## 事例 26 各ステークホルダが協力しながら進めていく 非機能要件定義の進め方 (富士通株式会社)

### 【取り組みの背景】

富士通の要件定義手法 Tri-shaping [37]は、要件定義工程でのビジネス要件の定義、システム化機能要件の定義や、非機能要件の定義をガイドしている。ここでは、そのうち非機能要件定義の進め方について紹介する。

Tri-shaping では、IPA から提示されている非機能要求グレード [13]を基本におき、要件定義工程において経営層や業務部門と協力しながら重要な要求を引き出し、そこを中心に最終的に非機能要求グレードに示されているメトリクスを決めていくプロセスで非機能要件定義を進める。非機能要求が合意形成に至るまでの全体の流れを以下に示す。

### 【非機能要求グレードの拡張】

Tri-shaping は非機能要求グレードを非機能要求の基本に置いているが、要件定義工程で円滑に合意形成ができるように工夫している。

- 最終結果のメトリクスの提示だけではなく、そのメトリクスに行き着くまでのプロセスを可視化している。
- 非機能要求グレードはシステム基盤をスコープに 238 のメトリクスを設定しているが、Tri-shaping は業務アプリケーションも含めたシステム全体をスコープとしていることから、メトリクス設定は 337 に増えている。

### 【非機能要件定義の流れ】

図 7.54 に Tri-shaping の非機能要件定義の流れを示す。

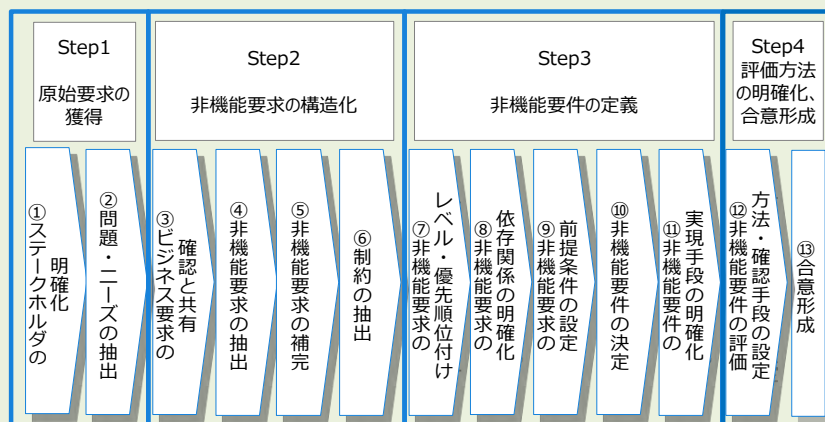


図 7.54 Tri-shaping での非機能要件定義の流れ

## 7.12 十分に妥当な非機能要求を定義する(DD.12)

### 【タスクの概要】

非機能要件定義には大きく4つのステップがあり、それらは13のタスクに分類される。以下に、各ステップおよびタスクの概要を説明する。このうち特にポイントとなるStep2、Step3については、イメージを図7.55、図7.56に示す。

#### Step 1

##### ① ステークホルダの明確化

業務の実現の観点だけでなく、業務を遂行するための一連の作業（業務運用）や IT サービス提供、システム運用・保守の観点で、要求を抽出するステークホルダを洗い出し、各ステークホルダの役割、権限、関係などを明確にする。

##### ② 問題・ニーズの抽出

システムに関わるすべてのステークホルダの満足度を高めるために、業務運用の観点だけでなく IT サービス提供、システム運用・保守の観点で、各ステークホルダのさまざまな関心事から問題・ニーズを洗い出す。

#### Step 2

##### ③ ビジネス要求の確認と共有

経営ビジョン・経営方針、部門ビジョン・部門方針、システム化方針、スローガン、経営目的・経営施策、業務目的・業務施策などの情報から、数量・時間・場所などを観点到、要求抽出シート（7.12.1項で紹介）を使って、ビジネス要求に含まれている業務特性・業務継続・業務連携・業務量・運用方針・セキュリティなど、トップダウンの非機能要求を確認し、その結果をステークホルダ間で共有する。

##### ④ 非機能要求の抽出

要求抽出シートを使って、非機能特性である可用性、性能・拡張性、運用・保守性・移行性などを観点到、ボトムアップの非機能要求を抽出する。

##### ⑤ 非機能要求の補完

現行システム仕様や類似システム仕様、あるいはソリューション仕様など、ステークホルダが参照し共有できるモデルを具体的に例示し、明示的に示されなかった暗黙の非機能要求を抽出することにより、ボトムアップの非機能要求を補完する。

##### ⑥ 制約の抽出

法令や組織規定、提携先とのインターフェースなど、要求を実現する上でのビジネス観点の制約や、利用するサービスや適用する製品など、技術観点の制約を抽出し、システム化や業務・システム運用に制限を与える条件を明らかにする。

## 7.12 十分に妥当な非機能要求を定義する(DD.12)

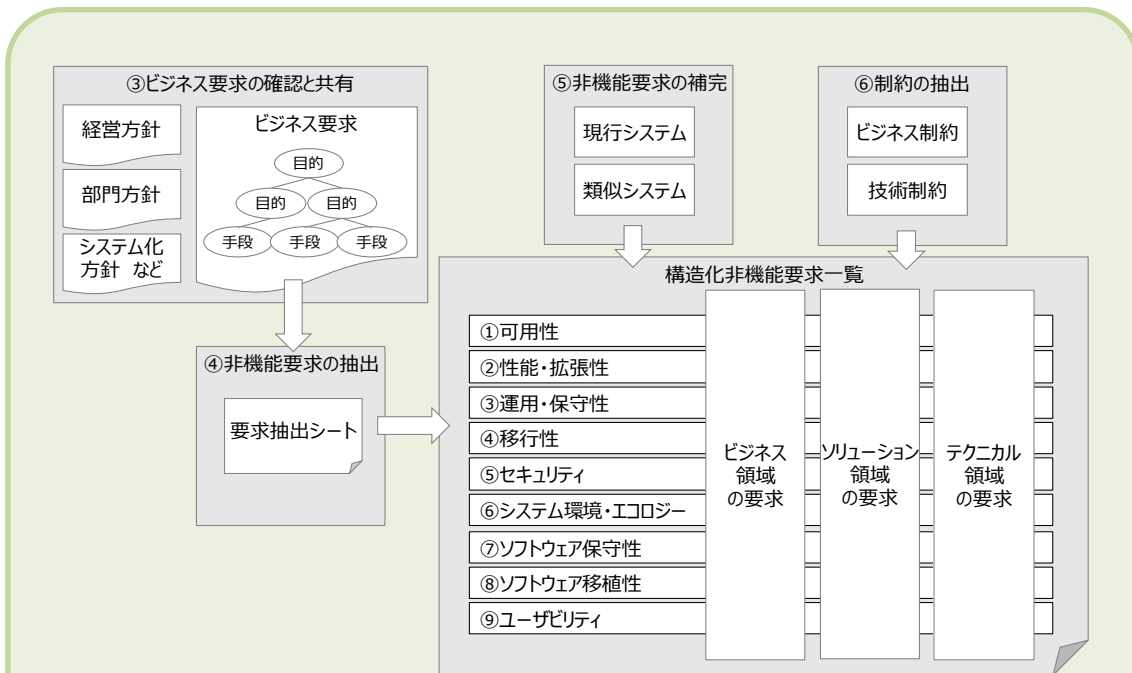


図 7.55 Step2 のイメージ

### Step3

#### ⑦ 非機能要求のレベル・優先順位付け

制約による非機能要求の実現可否や、要求の重要度や緊急性などから「Must」「Need」「Want」「Hope」のように実現の優先度を決めていく。その際、一般的にはビジネス領域にある非機能要求の優先度は高いものと判断する必要がある。

#### ⑧ 非機能要求の依存関係の明確化

制約とともに非機能要求相互の依存関係を明らかにする。依存関係があるものについて、矛盾した要件がないか確認する。非機能要求間でトレードオフがある場合は、要求度合いや優先順位付けをもとに、どちらの要求を優先して採用するかを判断する。

#### ⑨ 非機能要求の前提条件の設定

システム化や業務・システム運用に必要な条件に漏れがないかどうかを確認し、要求や制約として抽出されていなかった非機能要求を要求として設定する。また、非機能要求を達成するときに前提となる条件があれば設定する。設定した前提条件は、リスクとして管理する。

#### ⑩ 非機能要件の決定

抽出された要求と制約、前提条件を整理し、非機能要件定義書に、目標となるメトリクスを設定していく。



## 7.12 十分に妥当な非機能要求を定義する(DD.12)

### ⑪ 非機能要件の実現手段の明確化

非機能要件の実現性や技術的根拠を示すために、非機能要件を実現する手段を明らかにする。その際は、システム構成要素ごとに、実現手段の実装範囲に他との隙間や重複がないように実現性を検討する。なお、具体的な実現方式は設計工程で具現化する。

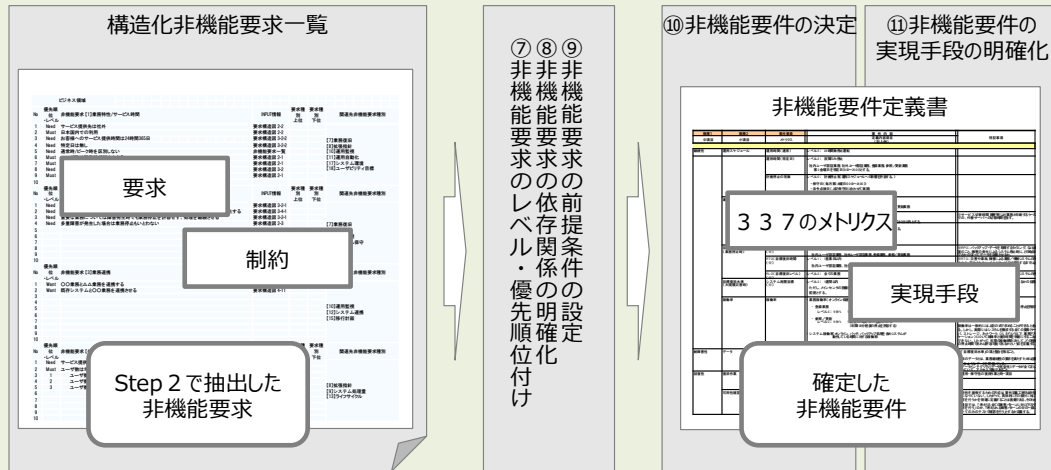


図 7.56 Step3 のイメージ

### Step4

### ⑫ 非機能要件の評価方法・確認手段の設定

非機能要求が満たされていることの妥当性を確認するため、測定する項目や評価方法、評価タイミングを決める。定性的な内容の要件など、ステークホルダによって認識が異なる可能性があるものは、要件定義段階であらかじめ評価項目を定めておき、ステークホルダ間で合意する。また、評価実施のために必要なツールなどの確認手段を明らかにする。

### ⑬ 合意形成

非機能要件定義書、評価方法・確認手段について、ステークホルダ相互の意見の一致を図る。なお、未確定要素がある場合は、前提条件を見直すほか、次工程に影響をおよぼさないよう、非機能要件を確定する時期をステークホルダと合意する。

上記で、非機能要求の進め方についての概略イメージを掴んでいただけたと思う。Tri-shaping は、非機能要件の各メトリクスを決定していくまでのプロセスをガイドし、必要な道具を用意している。

## 7.13 運用や移行、総合テストの要求も定義する (DD.13)

機能要求や非機能要求を定義することは要件定義工程の主たる作業ではあるが、それだけではない。プロジェクトライフサイクルやシステムライフサイクルの視点では、開発したシステムのテスト、移行を経て問題なく完遂することや、サービス終了まで安定運用を維持し続けることが求められる。それに向けた準備として、後続工程で予定される総合テスト、移行、運用の実施に向けた計画作りもまた、要件定義工程の重要な作業である。

### 7.13.1 運用要件書(DD.13.1)

#### 【目的】

- サービス提供（運用）に関する要求の意識合わせを行う。特に、コスト（見積り）への影響が大きい事項は必ず行う。
- コスト（見積り）感をユーザ部門と共有し、運用準備、および新システムでの運用コストの概算見積りができるようにする。

#### 【説明】

- 業務の遂行要求、システムの運用要求・障害時の対応についてまとめる。

#### 【留意事項】

- 要件定義の段階で、運用・操作要求を検討する。  
運用要件書の検討は、システム実装後まで後回しになることが多いが、要件次第ではシステムの基盤設計、運用保守体制やアプリケーション設計に大きな影響を与える場合も多い。要件定義の段階で運用要求を作成することで、運用に関する要求の漏れや手戻りを少なくすることができる。
- 運用部門を検討に巻き込む。  
運用部門が主体的に運用要件書を記述したほうが、定義している機能要求、非機能要求に対して運用部門ならではの視点で要求が抽出できたり、矛盾が抽出できたりする場合がある。また、引き継ぎもスムーズに進められるなどの利点もある。運用部門を巻き込むためには、要件定義プロセス定義計画において役割分担として明確にする必要がある。

#### 【成果物イメージ】

## 7.13 運用や移行、総合テストの要求も定義する(DD.13)

- 運用要件書に記載する主な記載項目を表 7.40 に示す。

表 7.40 運用要件書の主な記載項目

分類	項番	記述内容	記述内容の説明
1 構成要素	1.1	運用範囲	サービスを提供する範囲を記述する
	1.2	サービスレベル 目標	非機能要件定義書に基づき当該システム、ならびに運用として目標とするサービスレベルを記述する  例えば、以下に挙げる項目を記載する <ul style="list-style-type: none"> <li>・ システムのサービス時間、メンテナンスなどの休止に関する要件</li> <li>・ 年次、月次、週次、日次の運用スケジュール、特殊日のスケジュールなどを確認</li> <li>・ 地域による違いや、グローバルの運用の場合のサービス時間帯や、運用要件の違い</li> <li>・ ヘルプデスク・サービスの運用要件</li> <li>・ 関係する他システムの運用要件との整合性、制約条件、スケジュール</li> <li>・ 障害発生時の業務継続条件、対応方針（サーバー復旧、データの整合性、システム間の整合性）</li> </ul>
	1.3	運用体制	ユーザ、保守開発プロジェクト、その他関係者を含む運用体制を記述する
	1.4	運用・保守作業	想定される運用・保守作業を記述する
	1.5	IT サービス管理 方針	運用・保守作業を管理する対象（インシデント管理、キャパシティ管理など）と管理レベルを記述する

## 7.13.2 全体移行計画書(DD.13.2)

### 【目的】

- 移行に関する作業スコープを明確にする。
- 移行に関する作業工数の規模感を把握する。

### 【説明】

- 近年システム規模が増大し、現行システムからの移行をトラブルなく実施できるかどうか、システム構築の成否に大きな比重を占めるようになった。移植性に関する要求は、内容によっては、業務設計、システム基盤設計、稼働日などに、非常に大きな影響を与える場合がある。しかし、システム移行の検討は、システム実装後へと後回しにされることが多く、移行方法を検討した結果、機能の見直しが発生する場合がある。要件定義の段階で、システム移行計画のうち作成できる部分は作成しておくことと、それらを基本設計以降の各工程において順次見直すことが重要である。

### 【留意事項】

- システム利用者全体を対象にして一斉に全機能を切り替えるか、できるだけ機能や利用者を狭めてスモールスタートするかを確認する。スモールスタートの方が障害発生時の影響を局所化できるため、そのための仕組みを要求に組み込むことを推奨する。
- 並行稼働の有無を確認する。並行の場合は、現時点で想定する並行稼働の対象システムと想定期間を明記する。また、旧システムの運用凍結時期と新システムへの移行時期を明確にしておく。
- データ移行（コンバージョン）を早い段階で検討しておく、データの洗い出し時点で漏れていたデータの存在を発見でき、移行時だけでなく、要件定義、設計段階で、早期に対応を検討できる。なお、移行対象データはできる限り絞り込むと良い。5年に1回使用される程度のデータの移行は行わないなど、移行の必要性を確認する。
- データベースの中から誤りや重複を洗い出し、異質なデータを取り除いて整理する、データクレンジングが重要であり、次の点に留意することが必要となる。
  - データクレンジングの責任者を明確にしておく。
  - データクレンジング作業を、ベンダ任せにしない。
  - 対象データの取り扱いを明確にしておく。
  - 修正履歴を保管する。
- 移行しないデータ、手修正するデータ、確認用に紙または別ファイルに保管するデータを確認する。少量のデータや組織独自で管理しているデータなど、手作業でも移行できるものは、移行プログラムの数を増やさない考慮も必要である。
- 移行テスト、移行リハーサルを計画に入れておく。

## 7.13 運用や移行、総合テストの要求も定義する(DD.13)

- 障害対応や切り戻しを含めて検討しておく。移行作業に入ってからでの予期せぬトラブルに対応するため、システム移行作業途中の検証ポイントを明確にし、システム戻しの手順を検討しておく。
- 体制面等での考慮事項
  - 移行要件検討チームを独立したチームとして運用部門も巻き込んで組織し、開発チームより先に設置する。
  - 業務/システム/データに分けて検討する。データクレンジングを検討しておく。
- 次の項目は、後回しになることが多いため、この段階で方向性を検討し、要件定義に反映する。
  - 訓練、研修について、次の検討を行っておく。
    - a) 訓練・研修の要否、訓練・研修時の要求（個人情報表示、印字しない等）、時期および対象機能。
    - b) システム切替前に研修を開始する場合には、環境準備、開発スケジュール等。
    - c) システム運用に携わる現場要員の教育訓練・習熟計画を忘れない。
  - 機械処理系以外の移行可否判断基準(教育・習熟度、監査部門の承認、経理処理、企業内部への広報、セレモニー、資源準備、組織発令など)を検討する。
- 要件定義の段階で検討できていない事項は、未決事項として管理する。

### 【成果物イメージ】

- 全体移行計画書に記載する主な記載項目を表 7.41 に示す。

表 7.41 全体移行計画書の主な記載項目

分類	項番	記述内容	記述内容の説明
1. 構成要素	1.1	移行範囲	移行対象のハードウェア、データ、アプリケーションシステム、業務などの範囲を記述する
	1.2	移行実施概要スケジュール	移行実施日（段階移行の場合は移行完了日までの移行の流れを記述する
	1.3	移行方式	移行や切り替えの方式を記述する。
	1.4	移行作業概要	移行フローを大枠で整理し、業務部門、他システム、プロジェクトの役割分担を記述する
	1.5	移行支援ツール作成範囲	移行作業のうち、ツールを作成して実施する範囲を記述する
	1.6	移行準備スケジュール	手順作成、ツール作成、リハーサルなどのスケジュールを記述する
	1.7	移行作業に関する作成ドキュメント一覧	移行作業について作成するドキュメントを洗い出し、記述内容、業務部門への提示時期、承認ポイントを記述する

### 7.13 運用や移行、総合テストの要求も定義する(DD.13)

分類	項番	記述内容	記述内容の説明
	1.8	移行テスト方針	移行支援ツールのテスト、移行後業務連動テスト、移行リハーサルにおける範囲、利用環境、体制を記述する
	1.9	移行結果業務部門確認項目一覧	移行結果に関する業務部門確認項目を記述する
	1.10	移行実施体制	移行実施に向けた準備段階における個々の作業に関する体制（窓口、連絡体制など）を記述する
	1.11	会議体系	移行に関する会議体を記述する。
	1.12	移行対象外データの扱い	移行対象外データの運用フローを記述する
	1.13	移行異常時の対応方針	異常発生時における切り戻し、作業続行の指針や、縮退運転の方針を記述する
	1.14	移行期間中のシステム運用条件	移行期間中における、オンライン、バッチなどシステムの運用条件を記述する
	1.15	移行終了時の現行システムの扱い	現行システムの切り離し、撤去、保有データの保存期限、アクセス手段を記述する。
	1.16	移行セキュリティ方針	媒体搬送、データクレンジング、リハーサルなどの移行作業におけるセキュリティ対策を記述する
	1.17	移行処理/データ量定義	移行対象となるデータ量・処理量を記述する
	1.18	移行バックアップ方針	移行作業におけるバックアップ範囲、保存期間などの方針を記述する
	1.19	移行ハードウェア一覧	移行に必要な資源（メモリ/ワークエリア）の見積りデータと、その資源を確保できるか検証したハードウェアの一覧を記述する
	1.20	データクレンジング方針	データクレンジング対象についての方針を記述する
	1.21	エンドユーザ教育・訓練方針	業務やシステムの移行に伴う、エンドユーザに対する教育・訓練方針（内容、スケジュール、場所、環境、回数など）
	1.22	運用部門教育・訓練方針	システムの移行に伴う、運用部門に対する教育・訓練方針（内容、スケジュール、場所、環境、回数など）

### 7.13.3 総合テスト計画書(DD.13.3)

#### 【目的】

- 合意された品質レベルに達するためテストを効率よく進める計画を、プロジェクトメンバーに示す。
- 総合テストの具体的な準備を早期に進めるための基礎情報にする。

#### 【説明】

- 総合テスト計画書は作成検討の開始が遅くなりがちであるが、要求の内容によっては業務設計、運用設計などに非常に大きな影響を与える。総合テスト計画書の記載事項のうち、要件定義の成果から決定できる部分は作成しておくことにより、後続工程に入ってからの手戻りを少なくできる。

(注) 要件定義、基本設計等、それぞれの設計段階でテスト仕様を固め、漏れの早期発見と後続工程からの手戻り防止を図る方法はWモデルと呼ばれる。

- 総合テストシナリオの作成は以下を前提として開始する。
  - 要件定義書の根幹部分(システム化業務フロー、業務機能構成表など)がほぼ確定していること。
  - 全システム一括本稼働一斉切り替えなのか、部分的に移行していくのか、などの、総合テスト後の移行方式が決まっていること。

#### 【留意事項】

- 総合テスト計画書には正常系だけでなく、例外系、特殊系を含め、一連の業務が正しく実行できるかを確認するための総合テストシナリオを記載する。
  - ビジネスプロセス関連図や業務流れ図(業務フロー)を活用して総合テストシナリオを可視化する。
  - システムが複数のサブシステムから構成される場合、全サブシステムを通して業務が成り立つかを確認できるシナリオを作る。
  - テストケース作成については、通常業務で一般的に発生する業務や利用頻度の高い業務を優先し、特殊ケースや難易度の高い機能の確認はその後で作成する。
- 総合テストシナリオの作成には、業務を熟知した業務担当者と、同じく業務を熟知してテスト対象の業務全体を網羅するテストシナリオが作成できるシステム担当者のペアで担当させる。
- 総合テストの事前検証として、新システム全体が構成されたテスト環境(参照するテーブルおよびシステム動作環境もなるべく本稼働環境に合わせて設定する)で、基本的な業務データの流れテストを実施することにより、技術的環境の動作検証も含めて一連のデータの流れを検証できるので、致命的な動作不良を早期に除去できる。これ

### 7.13 運用や移行、総合テストの要求も定義する(DD.13)

により、以降の詳細シナリオに沿った総合テストを円滑に実施することができる。総合テストでのテストケースを要件定義で作成しておく、単体テスト、結合テストでもそれらを参考にすることができ、テストケースの抜けが減少する。

- 要件定義と並行して総合テスト計画を作成することは難しいが、総合テスト計画書を作成することにより発見できる要求の漏れは多い。要件定義作業の最後にこの作成を義務付けると良い。
- 総合テストの完了判断基準を決定しておく。
  - 期待する品質目標値（例. 不具合件数/規模）を記載しておく。
  - 期待するレスポンスタイム（例. 秒/件）や、バッチ処理時間等を記載しておく。
- 要件定義の段階で検討されていない事項は、未決事項として管理する。

#### 【成果物イメージ】

- 総合テスト計画書に記載する主な記載項目を表 7.42 に示す。



## 7.13 運用や移行、総合テストの要求も定義する(DD.13)

表 7.42 総合テスト計画書の主な記載項目

分類	項番	記述内容	記述内容の説明
1. 目的	1.1	総合テストの目的	総合テストで検証する目的、完了基準を記述する
2. 概要	2.1	対象、範囲	総合テストのテスト対象、ならびにテスト範囲を記述する
	2.2	前提条件	総合テストの開始基準、ならびに実行における前提条件を記述する
	2.3	環境	総合テスト環境の種類と種類ごとの構成（ハードウェア、ソフトウェア、ネットワーク）を記述する
	2.4	期間、スケジュール	総合テストの期間、スケジュールを記載する。テスト観点に応じて多段階で実行する場合は、それぞれのスケジュールも記載する
	2.5	体制	総合テストの推進体制と役割分担を記述する
3. テスト方針	3.1	業務間連動テスト方針	複数の業務を連動させるテストの対象業務、テスト観点（通常処理、異常処理、例外処理、特殊処理）を記述する
	3.2	サイクルテスト方針	日回し、年回しなどの業務サイクル通じたテストの対象業務、テスト観点（通常処理、異常処理、例外処理、特殊処理）を記述する
	3.3	業務運用テスト	締め処理（日次、月次、年次）などビジネスプロセスフローで定義した内容のテストの対象業務、テスト観点（通常処理、異常処理、例外処理、特殊処理）を記述する
	3.4	システム間連動テスト方針	ファイル連携などシステムをまたいだテストの対象、テスト観点（通常処理、異常処理、例外処理、特殊処理）を記述する
	3.5	非機能テスト方針	非機能要件定義書で定義した非機能要求に対するテストのテスト対象、テスト観点を定義する
	3.6	ユーザテスト方針	ユーザテストの対象、テスト観点を記述する

# おわりに

---

おわりに

## おわりに

本書は、システム構築の上流工程における重要プロセスである要件定義を適切に実施することにより、それに続く開発プロジェクトの失敗を減らし、構築するシステムに対する品質要求に応え、対象システムにより実現されるビジネスや新たなサービスに、より高い価値をもたらすことをガイドすることを目的として作成した。本書では、この「適切に」という言葉を、システム化の目的を達成するための要求を誤りなく選択し、そこから新たな価値を創造するビジネス要求を過不足なく定義し、システム化要求定義に抜け漏れなく展開する、そして、これらのプロセスを初期の計画どおりの QCD で実施できるようマネジメントすること、と定義してノウハウを記載した。

ノウハウは、要件定義の現場でよく発生する問題点を起点に、各問題が発生した際にどう対処するか、発生させないためにどうするかなどにつき、問題と解決の勘どころを対応させる形式で記載した。発生する問題は、現場の有識者の経験から特に重要なものを選択している。

ただし、要件定義において発生する問題は、このガイドに記載したものだけではない。むしろ、まだ本書に記載されていない問題の方が多いと思われる。また、ガイドに示した勘どころについても、本書とは異なるご意見をお持ちの方も少なくないと思われる。これについては、みなさまの経験をもとに補完・訂正して、ご自身の作業環境でのより良い要件定義のあり方として改善していただければ幸いである。また、本書に追記すべきノウハウをお持ちの方も多いと思われる。本ガイドの今後の成長のために、そして今後の要件定義をより実りのある活動にするために、ぜひお手持ちのノウハウをお寄せいただきたい。

IT の役割は“Support Business”から“Do Business”へと変化してきている、と言われ始めてからもう何年も経過している。このような状況においてビジネスの変化に対応した情報システムを構築・運用をするためには、今後、ますます要件定義の品質が重要になってくる。システム構築の手法は IT の進歩と歩調を合わせて常に進化し続けており、要件定義において作成する具体的な成果物の種類や記載内容も同様に変化し続けることが予想されるが、本書に示したさまざまな事項は、要件定義の勘どころとして今後も普遍性を保ち続けると考える。本ガイドがみなさまの要件定義の実施における道標として少しでも役立つことを願う。

# 付録

---

## 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

要件定義で作成すべき成果物を決めるためには、要件定義や前後の工程の目的を理解する必要がある。ここでは「共通フレーム 2013」 [27]に従って解説する。

共通フレーム 2013 は、発注者と開発者の双方の視点から情報サービス産業の標準的なライフサイクルプロセスを定めている。これは、情報システムの開発や活用を通じて発注者と開発者双方がともに成功を得るための規範となっている。また、共通フレームでは標準とされる成果物を規定しておらず、役割の観点でまとめたプロセスをアクティビティ、タスクに詳細化して提示している。

共通フレーム 2013 が規定する要件定義関連のプロセスを付図 1.1 に示す。また、同図に記載された各プロセスの目的を以下に示す。

### (1) 「企画プロセス」

#### ① 「システム化構想の立案プロセス」

経営上のニーズ、課題を実現、解決するために、新たな業務の全体像とそれを実現するためのシステム化構想を立案する。

#### ② 「システム化計画の立案プロセス」

システム化構想を具現化するために、運用や効果等の実現性を考慮したシステム化計画、プロジェクト計画を具現化し、ステークホルダの合意を得る。

### (2) 「要件定義プロセス」

利用者および他のステークホルダが必要とするサービスを提供するためにシステムで実現する業務の要件を定義する。(業務要件定義)

ステークホルダのニーズおよび要望を分析し、以下の対応を行う。

- 相互作用を表現する
- システムがニーズを満たすことを確かめる
- 合意を取る

### (3) 「システム開発プロセス」

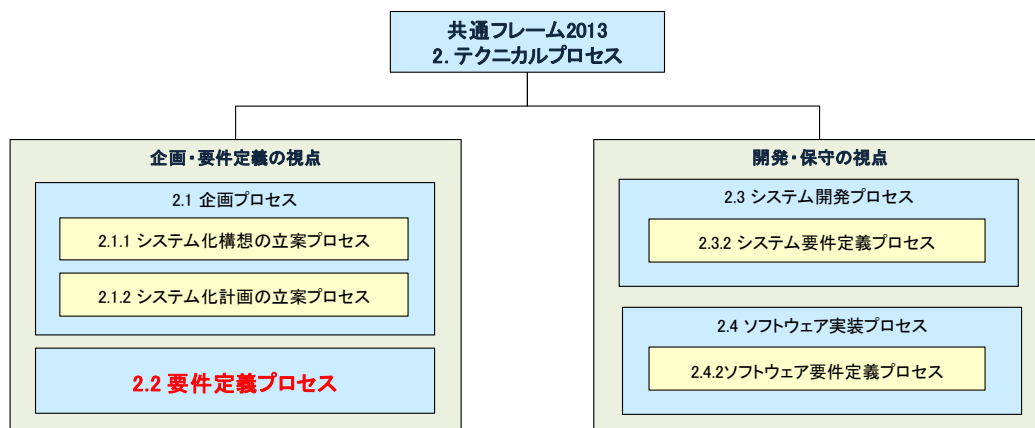
#### ① 「システム要件定義プロセス」

定義されたステークホルダの要件をシステムの技術的要件へ変換する。(システム要件定義)

### (4) 「ソフトウェア実装プロセス」

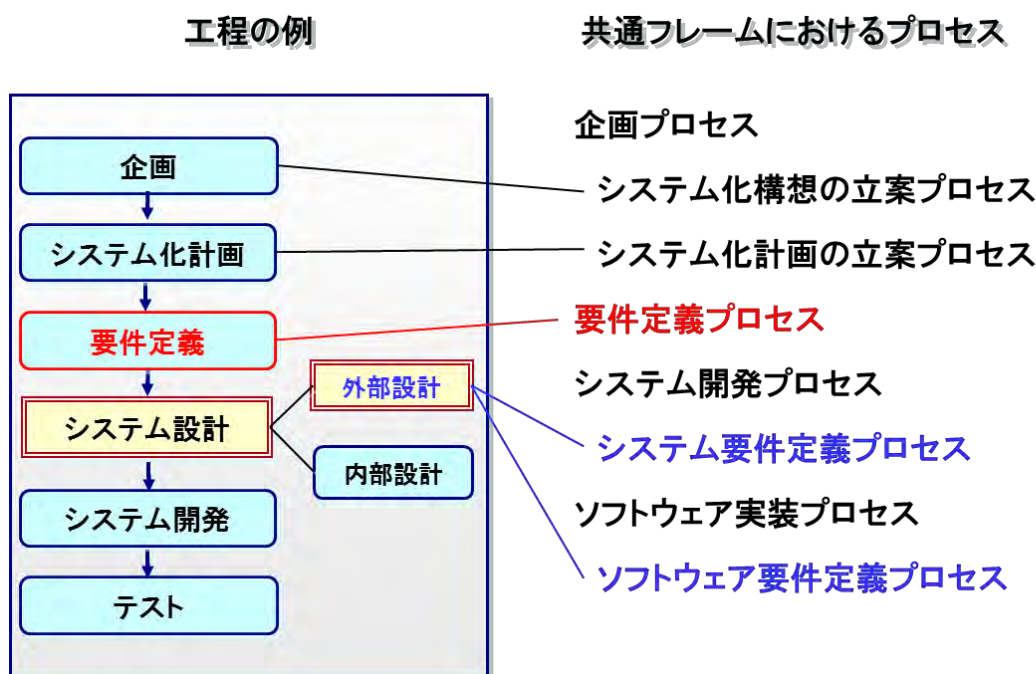
#### ① 「ソフトウェア要件定義プロセス」

システムのソフトウェア要素の要件を確立する。(ソフトウェア要件定義)



付図 1.1 共通フレーム 2013 における要件定義関連プロセス

ここで注意すべきことは、共通フレームでは、一般に使用される「工程」を、要件定義プロセスというように「プロセス」と表現しているということである。これは、システム開発を役割の観点からまとめているためであり、「プロセス」は期間や順序を規定したものではないことを意味している。「工程」は、一般に企業ごとにその呼称や定義が異なる。各企業は必要に応じ、自身の定義する「工程」と共通フレームの「プロセス」とのマッピングを行っている。このような工程とプロセスとの対応例を付図 1.2 に示す。



付図 1.2 工程とプロセス（共通フレーム）との対応例

## 付録

プロジェクトの状況や目的によって、どの工程にどのプロセスを配置するか、工程の目的を達成するためにどのような成果物を選択するかが異なる。

例えば、プロジェクトの目的が「経営上のニーズや課題を解決するための業務プロセスの改善」であれば、企画工程で業務フローを作成し、業務プロセスのニーズや課題を業務フロー上で確認することになるかもしれない。また、プロジェクトの目的が「ステークホルダが必要とするサービスを提供できる IT システムの明確化とそれによる業務の変化の明確化」であれば、要件定義工程でシステム化業務フローを作成し、設計工程では作成したシステム化業務フローを使って開発部門と「ユーザと IT システムの役割分担」について合意形成することになるかもしれない。

また、一般的にシステム化業務一覧やシステム化業務説明は、システム機能の外部仕様をまとめる成果物として開発プロセスに位置付けられる。しかし、プロジェクトの特性によっては「利用部門が主として参画している要件定義工程で実施した方が効率的である」、「見積り精度を上げるために要件定義工程で作成する」などと判断して、作成する工程を変更して良いわけである。

各プロジェクトでは、それぞれが置かれている状況に基づいてプロジェクトの目的を明確にし、作業プロセスを配置し、作成するドキュメントを決定していくことが求められる。

## 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」

IPA が公開している「機能要件の合意形成ガイド」 [28]は、システム要件定義に関わる成果物作成上のコツが示されているので参考になる。

機能要件の合意形成ガイドは、実現したい情報システム像を伝える側（発注者）と、それを設計し開発する側（開発者）との間で起こる認識の齟齬を防止することを目的としたガイドであり、発注者と開発者が仕様を確認する最後の段階である基本設計（共通フレームの用語では、システム・ソフトウェア要件定義）での発注者-開発者間の合意形成のコツを紹介している。このガイドには、成果物に対し「何を書けば良いか」「どう書けば良いか」についても多く記載されている。

機能要件の合意形成ガイドは、基本設計工程での活用を想定して記載されており、そこには、システム化する機能の視点での合意形成のコツが書かれている。例えば、システム化業務フローでは、システム化する業務の範囲や機能、システムとの接点の合意形成のためのコツが書かれており、システムを利用した業務がどれだけ業務改善に貢献できるかという視点とは異なる。しかし、要件定義（業務要件定義）の成果物は基本設計へのインプットであり、基本設計へのつながりを意識して要件定義を実施することが重要であることから、合意形成ガイドをぜひ参考にしていきたい。参考として、要件定義に関連する機能要件の合意形成ガイドの成果物に対するコツの概要を付表 2.1 に掲載した。詳細は、IPA のホームページからダウンロードできる。

また、このガイドで提示している成果物は、基本設計で合意形成をする必要があると説明しているが、必ずしも基本設計で作成すべきであるとは述べていない。要件定義プロセスや企画プロセスで作成すべき成果物もあるので注意が必要である。



付表 2.1 本ガイドの成果物で活用可能な「機能要件の合意形成ガイド」の  
施策（コツ）の例

<システム振舞い編>

工程 成果物名	コツ分類	コツID	施策（コツ）
システム 化業務一 覧	書き方	02D101	システム化業務一覧の中のシステム化業務をグルーピングして、階層的に記述する。
		02D102	システム化業務一覧は、要件定義で作られた業務一覧と業務フローを使って系統立てて記述する。
		02D103	システム化業務一覧のそれぞれのシステム利用作業に、対応する利用者の記述を追加する。
		02D105	システム化業務一覧に記述する機能一式は、業務処理の流れに沿って整理させる。
		02D106	システム化業務一覧に記述された各機能に対して、対応する業務を関係付ける。
		02D107	システム化業務一覧に記述された各システム化業務に対して、その振舞いの想定や制約などの特徴を関係付ける。
		02C201	システム化業務一覧を使って、システム化業務の名前が一意であることを確認する。
	確認 レビュー	02R201	システム化業務一覧を使って、業務と対応付けながら、業務とその業務に関連する機能との対応を確認する。
	システム 化業務フ ロー	書き方	02D201
02D202			システム化業務フローの詳細を隠蔽した、大きな流れを記述する。
02D203			システム化業務フローの各業務を、部門などの役割に基づいた区画に記述する。
02D204			システム化業務フローにおいて、システム化する部分を、システム化業務を表すアイコンを使って記述し、システム化しない業務と区別する
02D205			システム化業務フローにおいて、システム化する範囲を、区画の1つを使って記述し、区画の境界を通してシステム化の内と外を区別する。
02D206			システム化業務フローにおいて、システム化業務の流れ、データの流れを表す矢印は、それぞれ異なる線の種類を用いる。
02D207			システム化業務フローのフローを交差しないようにする。
02D208			システム化業務フローの全体を俯瞰できるように、図表1枚に記述するシステム化業務を階層化する。
02D209			システム化業務フローにおいて、開発対象と開発対象外のシステム化業務を、異なるアイコンを使って記述し、両者を区別する。
02D210			システム化業務フローに説明文も併せて記述する。
02D211		システム化業務フローに書かれているシステム化業務と、その各々を詳細に説明する文章・図表を互いに関連付け、お互いに迎れるように配慮する。	
レビュー		02C101	システム化業務フローで、使用される画面や帳票の対応関係を確認する。
		02R101	発注者は、社内他部門、社外関係者など、当該業務に関与する関係者（社）の業務処理内容、処理・作業順序を、業務フローとシステム化業務フローおよびシステム化業務一覧などと比較・検証する。
		02R202	システム化業務フローを使って、業務上のタイミングを確認する。
		02R203	システム化業務フローを使って、分岐の境界条件を確認する。
		02R301	レビューは、要件定義書に登場する主要な登場人物（利用者、関係者）を確認する。
		02R402	最初に、サブシステム間、開発対象と開発対象外部、システム化業務と作業、などの境界に着目し、境界部分の入出力データをレビューする。
		02R403	システム化業務一覧とシステム化業務フローで、異なる利用者から使われる同じ名前のシステム利用作業に着目して、それらが本当に同じであることを確認する。
		02R407	画面・帳票をあげながら、業務とシステムを対応させて、システム化業務フローをレビューする。
		02R408	重要な日付／期間に応じたシステム化業務を確認する。
		02R501	システム化業務フローに手順や手続きを明記し、順序を意識しながらレビューする。
	04R201	システムが扱う情報を俯瞰できる資料を作成し、システムが扱う情報の範囲を確認する。	
システム 化業務説 明	書き方	02D301	システム化業務説明に、システム化業務の実行前後の条件を記述する。
		02D302	システム化業務のシナリオを、利用者とシステムのそれぞれのアクションに分けて記述する。
		02D303	1つのシステム化業務の振舞いに対して、基本・代替・例外の3種類のシナリオを考える。
		02D304	システム化業務のシナリオの中の個々のアクション（箇条書き）の先頭に、連番ではなく枝番を使用する。
	レビュー	02R404	システム化業務説明のシステムのアクションにより、要件定義書の現行システムの課題が解決していることを確認する。
		02R405	利用者の役割・権限に着目して、システム化業務説明の基本シナリオのバリエーションを確認する。
		02R406	システム化業務フローに書かれている、システム化の対象範囲外の作業間で引き渡されるデータの種類と状態を確認する。
		02R502	画面一覧と帳票一覧を用いて、画面や帳票を通して利用者が関わるシステム化業務が何かを確認する。
		02R503	画面レイアウトと帳票レイアウトを用いて、画面や帳票を通して利用者がかわるシステム化業務がどう動くかを確認する。
		02R504	システム化業務説明の例外シナリオに関係する画面レイアウト（エラー処理、エラー表示）を確認する。
		02R505	システム化業務説明の事前条件・事後条件・例外シナリオが、要件定義書の業務レベルの条件や制約に対応していること、および考慮されていないケースがないことを確認する。

## &lt;画面編&gt;

工程 成果物名	コツ分類	コツID	施策（コツ）
画面レイ アウト	書き方	03D301	画面レイアウト上で、画面部品などの表現を整理・工夫する。
		03D303	画面レイアウト上の操作手順に、操作に対するシステムのアクションも記述する。
	確認	03C301	構成要素の配置やボタン押下時における簡単な動作のみの記述に留まるのではなく、入力情報や制約条件等の詳細情報も併せて記述する。
		03C302	概要に情報項目を記載する際には、なるべく情報項目群の単位で記述する。
		03R205	レビューの際にエラー表示を具体的に説明する。
	レビュー	03R401	画面イメージを提示して、実際の配色を確認する。
		03R402	権限や条件と対応して画面レイアウトを説明する。
画面レイ アウト (標準化)	書き方	03D600	個別画面に依らず、共通して合意すべきことは、画面レイアウト共通ルールとして定義する。
		03D601	画面遷移・レイアウト共通ルールの1つとして、あらかじめページの構成要素を定義する。
		03D602	画面遷移・レイアウト共通ルールの1つとして、あらかじめ画面遷移パターンを定義しておく。
		03D603	画面遷移・レイアウト共通ルールの1つとして、あらかじめエラー表示方法・エラーメッセージを定義する。
		03D604	準拠すべき社内外の基準・標準、ガイドライン、特筆すべき要件、開発コンセプト等があれば明記しておく。
		03D606	複数の画面で表示される一組の入出力項目については、レイアウトを共通ルールとしてパターン化する。
		03D607	制約を示す項目やルールは可能な限り数値化する。
	確認	03C401	各エリアの役割を明記されているかを確認する。
		03C402	各エリアの名称に加えて、配色やレイアウト例を施した全体イメージが示されているかを確認する。
		03C403	色名称に加え、実際の色で着色したフォントなどを併記しているかを確認する。
		03C404	使用できるカラーの範囲、基本的な配色を定義されているか、また、色彩の使い方について一定の法則が設けられているかを確認する。
		03C405	フォント名に加えて、フォント表示の具体例を示しているかを確認する。
		03C406	共通にするべき範囲が明確にされていることを確認する。

## &lt;データモデル編&gt;

工程 成果物名	コツ分類	コツID	施策（コツ）
概念デー タモデル	書き方	04D101	エンティティを分類でタイプ分けして、色、網掛け、枠囲みで表現する。
		04D102	ER図中のエンティティにインスタンスの作成、および更新の時期を記述する。
		04D103	業務単位にER図を作成する。
		04D104	イベント系エンティティをデータの発生順で記述する。
	レビュー	04R301	データモデルに、実データを当てはめて補足する。
		04R302	画面レイアウト、帳票レイアウトとER図を対応づけて確認する。
		04R307	新規要件に伴うER図の変更は、変更前/後を比較して効果を説明する。
		04R308	旧システムと新システムのデータベース変更の方針について業務内容の観点で図示して説明する。
		04R309	業務とエンティティのまとまりとの対応を示し、データモデルの読み方を説明する。
		04R309	現行システムなどの具体的なエンティティおよび属性の抽出元を記述する。
データ項 目定義書	書き方	04D204	現行システムなどの具体的なエンティティおよび属性の抽出元を記述する。
CRUD図	書き方	04D301	Create, Read, Update, Deleteの記載欄を明確に分けて（固定させて）表現する。
		04D302	CRUD図のエンティティをイベント系エンティティとリソース系エンティティに分離して記述する。
		04D303	CRUD図のビジネスプロセスを実際の業務の時系列順に並べる。
		04D304	CRUD図のイベント系エンティティを、エンティティを作成する順番に沿って並べる。

## &lt;帳票編&gt;

工程 成果物名	コツ分類	コツID	施策（コツ）
帳票レイ アウト	書き方	07D001	印刷量によって印字する行数が増減する場合、改ページ条件を考慮した帳票レイアウトを作成する。
		07D002	帳票レイアウトに印字される種類（テキスト、表部品、画像、バーコード）を明示する。また、具体的な部品の属性を明示する。
	レビュー	07R002	実業務で用いている紙の伝票と帳票項目説明及び帳票レイアウトの対応付けを確認する。
		07R003	プレ印刷部、固定文字列及び可変文字列の識別をし、それに基づいて発注者に確認する。
		07R004	帳票レイアウトに関する詳細な要件を確認するために具体的なイメージを提示しそれを基にレビューする。
		07R006	利用目的や利用者・利用シーンと照らし合わせ、実際の利用者を交え帳票のレビューを行う。

### 付録3 CMMI 1.3 と本ガイドの記載の関係

CMMI (Capability Maturity Model Integration : 能力成熟度モデル統合) [38]は組織の保有する能力の成熟度をモデル化したものであり、システム開発を行う組織が自己のプロセス改善を行う際の指標になる。CMMI では、製品、サービスの開発、調達能力などを5段階で表す。組織プロセス改善の到達度が、このレベルにより表される。

CMMI における成熟度レベルは、以下のように分類されている

- レベル1 初期
- レベル2 管理された
- レベル3 定義された
- レベル4 定量的に管理された
- レベル5 最適化している

CMMI では組織の成熟度に応じて、レベルごとにさまざまなプロセス領域が定義されている。このうち、要件定義に関係の深い2つのプロセス(『要件管理』と『要件開発』)について、ガイドとの関連を付表3.1に記載する。要件管理の各項目などCMMIに記載されているプロセスに対応していない事項もあり、このガイドに記載されている事項を実施すればそれだけでCMMIの組織成熟度モデルに適合するというものではないが、要件定義を実施する組織に要求される活動の事例として参考にしていただきたい。

付表3.1 CMMI とガイドの各記載との対応

カテゴリ	目的	固有ゴール	ゴール	固有プラクティス	プロセス	ガイドの記載
1. 要件管理 (REQM) 成熟度レベル2	プロジェクトの成果物の要件および成果物構成要素の要件を管理すること、およびこれらの要件とプロジェクトの計画および作業成果物との間の整合性を確保すること	1. 要件を管理する	要件が管理され、プロジェクト計画および作業成果物との不整合が特定されている。	SP 1.1 要件を理解する	要件の意味に関して要件提供者と共に理解する。	
				SP 1.2 要件に対するコミットメントを獲得する	プロジェクト参加者から要件に対するコミットメントを獲得する。	
				SP 1.3 要件変更を管理する	プロジェクト実行中の要件の進化に伴い、要件への変更を管理する。	
				SP 1.4 要件の双方向の追跡可能性を維持する	要件と作業成果物との間に双方向の追跡可能性を維持する。	6.3.6
				SP 1.5 プロジェクト作業と要件の間の整合性を確保する	プロジェクトの計画および作業成果物が要件と整合している状態が保たれているようにする。	

カテゴリ	目的	固有ゴール	ゴール	固有プラクティス	プロセス	ガイドの記載
2. 要件開発 (RD) 成熟度レベル3	顧客要件、成果物要件、および成果物構成要素の要件を引き出し、分析し、そして確立すること	1. 顧客要件を開発する	利害関係者のニーズ、期待、制約、およびインタフェースが集められ、顧客要件に変換されている。	SP 1.1 ニーズを引き出す	成果物ライフサイクルのすべてのフェーズについて、利害関係者のニーズ、期待、制約、およびインタフェースを引き出す。	4.1 4.2.1 4.2.2
				SP 1.2 利害関係者のニーズを顧客要件に変換する	利害関係者のニーズ、期待、制約、およびインタフェースを優先付けされた顧客要件に変換する。	4.2.3
		2. 成果物要件を開発する	成果物要件および成果物構成要素の要件を開発するため、顧客要件が精緻化され詳細化されている。	SP 2.1 成果物要件と成果物構成要素の要件を確立する	顧客要件に基づいて、成果物要件および成果物構成要素の要件を確立し保守する。	6.3.6
				SP 2.2 成果物構成要素の要件を割り当てる	各成果物構成要素に要件を割り当てる。	7.5.2
				SP 2.3 インタフェース要件を特定する	インタフェース要件を特定する。	7.9
		3. 要件を分析し妥当性を確認する	要件が分析され、その妥当性が確認されている。	SP 3.1 運用の考え方とシナリオを確立する	運用の考え方および関連するシナリオを確立し保守する。	7.5.3 7.5.4 7.13.1
				SP 3.2 必要とされる機能性と品質属性の定義を確立する	必要とされる機能性および品質属性の定義を確立し保守する。	7.12.1
				SP 3.3 要件を分析する	要件が必要かつ十分であるようにするため、それらを分析する。	4.1.3
				SP 3.4 つり合いをとるために要件を分析する	利害関係者のニーズと制約をつり合わせるため、要件を分析する。	5.2.1
				SP 3.5 要件の妥当性を確認する	得られた成果物が最終利用者の環境で意図されたとおりに稼動するようにするため、要件の妥当性を確認する。	5.2.2 6.4.2

## 付録

付表 3.1 の内容は、CMMI Institute からの特別な許諾に基づく、開発のための CMMI®1.3 版（原文：CMMI® for Development Version 1.3 Model, Japanese Translation, ©2010）からの引用を含む。

CMMI Institute のいかなる文書も、『現状有姿のまま』提供されるものである。CMMI Institute は、特定目的適合性、商品性、独占性、および使用結果についての保証を含む、いかなる明示または黙示の保証も提供しない。CMMI Institute は、特許権、商標権、または著作権侵害の問題が生じても、いかなる責任も負わない。

CMMI® は、CMMI Institute の登録商標である。

## 参考文献

- [1] IPA, 経営者が参画する要求品質の確保 ～超上流から攻める IT 化の勘どころ～ (第 2 版), IPA, 2006: <https://www.ipa.go.jp/sec/publish/tn05-002.html>
- [2] 経済産業省, DX(デジタルトランスフォーメーション)レポート ～IT システム「2025 年の崖」の克服と DX の本格的な展開～, 経済産業省, 2018: <https://www.meti.go.jp/press/2018/09/20180907010/20180907010-3.pdf>
- [3] 経済産業省, デジタルトランスフォーメーションを推進するためのガイドライン (DX 推進ガイドライン), 2018: <https://www.meti.go.jp/press/2018/12/20181212004/20181212004-1.pdf>
- [4] IPA, 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, 2017
- [5] 日本情報システム・ユーザー協会 (JUAS), デジタル化の取り組みに関する調査 2018, JUAS, 2018
- [6] 日本情報システム・ユーザー協会 (JUAS), 企業 IT 動向調査報告書 2019, JUAS, 2019
- [7] 日本情報システム・ユーザー協会 (JUAS), ソフトウェアメトリックス調査 2016, JUAS, 2016
- [8] 経済産業省, 情報システム・モデル取引・契約書 (2007・4), 2007: [http://www.meti.go.jp/policy/it_policy/keiyaku/model_keiyakusyo.pdf](http://www.meti.go.jp/policy/it_policy/keiyaku/model_keiyakusyo.pdf)
- [9] IPA, 「グローバル化を支える IT 人材確保・育成施策に関する調査」概要報告書, 2011: <https://www.ipa.go.jp/jinzai/jigyou/global-report.html>
- [10] 総務省, ICT による経済成長加速に向けた課題と解決方法に関する調査研究, 2014: [http://www.soumu.go.jp/johotsusintokei/linkdata/h26_02_houkoku.pdf](http://www.soumu.go.jp/johotsusintokei/linkdata/h26_02_houkoku.pdf)
- [11] Hasso Platter Institute of Design at Stanford, 柏野 尊徳 (訳), 中村 珠希 (訳), スタンフォード・デザイン・ガイド: デザイン思考 5 つのステップ, (一社) デザイン思考研究所 (編), 2012: [https://designthinking.or.jp/index.php?dt_text](https://designthinking.or.jp/index.php?dt_text)
- [12] ISO/IEC 25010 Systems and software engineering -Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models, 2011
- [13] IPA, 非機能要求グレード, 2010: <https://www.ipa.go.jp/sec/softwareengineering/reports/20100416.html>
- [14] 日本情報システム・ユーザー協会 (JUAS), ソフトウェアメトリックス調査 2019,

JUAS, 2019

- [15] 石谷 靖, ソフトウェア開発見積りの課題と解決方法, 2012:  
<https://www.ipa.go.jp/files/000004077.pdf>
- [16] B. Boehm, Software Engineering Economics, Prentice-Hall 社
- [17] ISO/IEC/IEEE 29148 Systems and software engineering - Life cycle processes  
- Requirements engineering, 2011
- [18] JIS X 0166 システム及びソフトウェア技術-ライフサイクルプロセス, 財団法人 日本規格協会, 2014
- [19] ピーター・チェックランド, ジム・スクールズ(著), 妹尾 堅一郎 (訳), ソフト・システムズ方法論, 有斐閣, 1994
- [20] スティーヴン・レヴィット(著), 他, 0 ベース思考---どんな難問もシンプルに解決できる 単行本 (ソフトカバー), ダイヤモンド社, 2015
- [21] アレックス・F. オスボーン(著), 豊田 晃(訳), 創造力を生かすーアイデアを得る 38 の方法 単行本, 創元社, 2008
- [22] JUAS(編), 福田 修編(著), SE を極める 仕事に役立つ文章作成術, 日経 BP 社, 2005
- [23] ソフトウェアイノベーションセンタ, NTT データ, NTT, 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, 2015
- [24] IPA, 高信頼化ソフトウェアのための開発手法ガイドブック, 独立行政法人情報処理推進機構 ソフトウェア・エンジニアリング・センター, 2011
- [25] ANSI/IEEE 1028 Software Reviews and Audits, 1998
- [26] IPA, 情報処理システム高信頼化教訓集 IT サービス編, IPA, 2019:  
<https://www.ipa.go.jp/ikc/reports/20190315.html>
- [27] IPA, 共通フレーム 2013~経営者、業務部門とともに取組む「使える」システムの実現~, IPA, 2013
- [28] IPA, 機能要件の合意形成ガイド, IPA, 2010:  
<https://www.ipa.go.jp/sec/softwareengineering/reports/20100331.html>
- [29] 日本情報システム・ユーザー協会 (JUAS), ソフトウェアメトリックス調査 2012, JUAS, 2012
- [30] 長谷川 智亮, 池田 訓子, 柴原 英明, 山中 啓之, 藤貫 美佐, 上流工程での要件定量化によるスコープマネジメント, プロジェクトマネジメント学会 第 26 回秋季研究発表大会
- [31] IPA, 先進的な設計・検証技術の適用事例報告書 2017 年度版 事例 81 総合的技術によるユーザ主体のソフトウェア開発手法「モデル指向開発(MOD)」の紹介 株式会社

- 日立産業制御ソリューションズ 提供, 2017:  
<https://www.ipa.go.jp/files/000064274.pdf>
- [32] 吉川 武男, バランス・スコアカード (Balanced Scorecard) の本質とは, 2002:  
<https://www.technofer.co.jp/downloadfiles/pdf14.pdf>
- [33] アレックス・オスターワルダー, イヴ・ピニユール(著), 他, ビジネスモデル・ジェネレーション ビジネスモデル設計書, 翔泳社, 2012
- [34] 清水 吉男, 「【改訂第2版】[入門+実践] 要求を仕様化する技術・表現する技術」改訂版のポイント, 技術評論社, 2010
- [35] IPA, 情報セキュリティ読本, 実教出版:  
<https://www.ipa.go.jp/security/publications/dokuhon/2006/>
- [36] IPA, 情報セキュリティ教本, 実教出版:  
<https://www.ipa.go.jp/security/publications/kyohon2/index.html>
- [37] 富士通株式会社, 要件定義手法 Tri shaping, 2011:  
<http://pr.fujitsu.com/jp/news/2011/02/9-2.html>
- [38] Carnegie Mellon University Software Engineering Institute CMMI 成果物チーム, 開発のためのCMMI® 1.3版, 2010:  
[https://resources.sei.cmu.edu/asset_files/WhitePaper/2011_019_001_28778.pdf](https://resources.sei.cmu.edu/asset_files/WhitePaper/2011_019_001_28778.pdf)
- [39] 一般社団法人 情報サービス産業協会(JISA), 要求工学知識体系 第1版(REBOK: Requirements Engineering Body Of Knowledge)
- [40] IIBA 日本支部 BABOK v3 翻訳プロジェクト, ビジネスアナリシス知識体系ガイド Version3.0, IIBA 日本支部
- [41] Project Management Institute(著), プロジェクトマネジメント知識体系ガイド PMBOK ガイド 第6版(日本語), 2018
- [42] ヴィジェイ・クーマー(著) 渡部 典子(訳), 101 デザインメソッド — 革新的な製品・サービスを生む「アイデアの工具箱」, 英治出版, 2015
- [43] 片岡 雅憲, 小原 由紀夫, 光藤 昭男(著), アジャイル開発への道案内, 近代科学社, 2017
- [44] 室脇 慶彦, IT負債 基幹系システム「2025年の崖」を飛び越えろ, 日経BP, 2019
- [45] 安藤 昌也, UXデザインの教科書 - もっとうれしい体験を。 , 丸善出版, 2016
- [46] エリック・リース(著), 他, リーン・スタートアップ, 2012
- [47] IPA, つながる世界の利用時の品質～IoT時代の安全と使いやすさを実現する設計～, 2017: <https://www.ipa.go.jp/sec/reports/20170330.html>



## 執筆（敬称略）

システム構築上流工程強化部会

「システム化要求 WG」

主査	：森崎 修司	国立大学法人名古屋大学
委員	：太田 忠雄	株式会社ジャステック
	河尻 直己	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	日鉄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
	森田 功	富士通株式会社
事務局	：村岡 恭昭	独立行政法人情報処理推進機構

## 監修（敬称略）

システム構築上流工程強化部会

主査	：山本 修一郎	国立大学法人名古屋大学
委員	：小野 修一	株式会社エヌ・ティ・ティ・データ
	小浜 耕己	スミセイ情報システム株式会社
	崎本 壮	株式会社日立製作所
	長山 一	一般社団法人日本情報システム・ユーザー協会
	福田 靖	日本電気株式会社
	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
	森崎 修司	国立大学法人名古屋大学
	横山 隆介	株式会社日本取引所グループ

## 部会/WG 運営

事務局	：山下 博之	独立行政法人情報処理推進機構
	村岡 恭昭	独立行政法人情報処理推進機構

## ユーザのための要件定義ガイド 第2版

要件定義を成功に導く128の勘どころ

---

2019年12月10日 2版1刷発行

2021年11月1日 2版2刷発行

監修者 独立行政法人情報処理推進機構 (IPA)

社会基盤センター

発行人 片岡 晃

発行所 独立行政法人情報処理推進機構 (IPA)

〒113-6591

東京都文京区本駒込二丁目28番8号

文京グリーンコート センターオフィス

<https://www.ipa.go.jp/ikc/>

©独立行政法人情報処理推進機構

※本書の図は、第三者の著作物を利用して作成しています。

---

ISBN 978-4-905318-72-9 Printed in Japan





独立行政法人情報処理推進機構  
社会基盤センター