

19-J004

要求開発・管理ベストプラクティス とその体系化の調査研究

—要求開発・管理の現場力を高めるノウハウの共有—

平成 20 年 3 月

社団法人 情報サービス産業協会

概要

平成 19 年度 要求開発・管理ベストプラクティスとその体系化の調査研究 —要求開発・管理の現場力を高めるノウハウの共有—

1. 調査目的

本調査研究は、要求開発の現場で実践してきた経験から生まれたノウハウを収集し、整理・体系化を行い、ベストプラクティスとしてまとめることを目的に活動した。

現段階で、114 件のプラクティスを収集し、35 件をベストプラクティスとして整理した。ユーザからの要求を獲得・分析し、仕様書としてまとめる「要求開発プロセス」を現場で実践する上での課題を解決し、改善を図り、現場力を高めるために、これらのベストプラクティスを活用して頂きたい。

2. 要求開発プロセスの課題

要求開発プロセスは、情報システム開発の成否を握ると言っても過言ではない。しかし、現実には、要求開発プロセスが情報システム開発の、最大の問題要因となっている。

我が国の製造業の中で、高品質と低コストを両立させて、国際競争に勝ち抜いた製品の開発では、常に、上流工程から品質の作り込みに取り組んできた。情報システムの開発でも、上流工程である要求開発プロセスの改革と改善が急務である。

このような要求開発プロセスの重要性の認識は、近年、情報サービス産業全体で広がりつつある。しかし、実際に要求開発プロセスを改善するためには、解決すべき課題が多い。図 1 は、情報サービス産業協会の実態調査¹において、要求開発プロセスにおける課題を示す。複数回答の中で、次の主要 4 課題だけで 51.6%を占めることに注意すべきである。

- (1) 顧客自身が要求を把握していない(13.6%)
- (2) 顧客要求の頻繁な変更(13.6%)
- (3) 要求内容の相違(16.5%)
- (4) 顧客協力の不足(7.9%)

このような実態調査から見てきたことは、要求開発プロセスの改善が、技術面だけでなく、組織面、管理面でも様々な改善を必要とする、多面的な課題を含むことである。

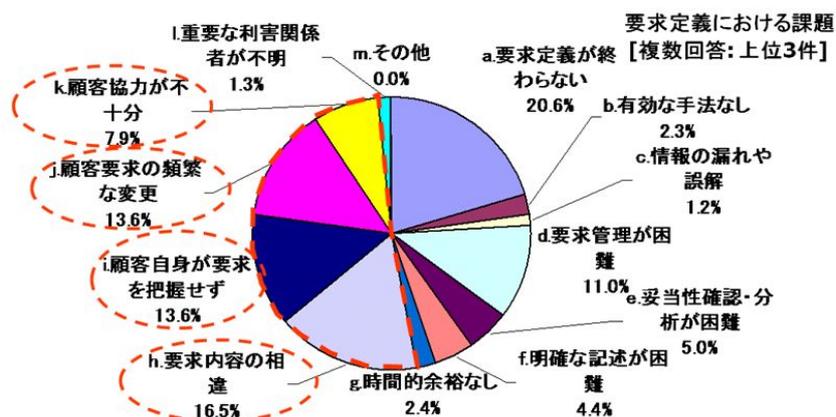


図 1. 要求開発プロセスにおける課題

¹ 情報サービス産業協会、平成 17 年度情報サービス産業におけるソフトウェア開発の実態アンケート調査結果(概要)、2007。

3. 要求工学ベストプラクティスとは

要求開発と管理における課題は、ユーザの業務や環境など、多様である。従って、要求工学を実践するためには、このような業務や環境に応じて、適切な技術の選択や調整が必要である。そのため、現場では、要求開発を実践する上で、ユーザとベンダとの関係や開発の実情、背景を踏まえて様々な工夫がなされている。このような現場のノウハウを共有することこそが、要求工学を現場で実践する助けとなる。

本報告書では、このような現場のノウハウを委員会で収集し、レビューによって適切と思われるノウハウを「要求工学ベストプラクティス」としてまとめた。

4. 要求工学ベストプラクティスの収集

ベストプラクティスをできるだけ多く収集するために、特別の条件を付けず、委員会の委員全員が経験に基づき提案する方針をとった。これは、オープンな組織によってできるだけ多くの関係者の力を結集するという、オープンソース開発や Web2.0 のフォークソノミー(大衆による分類)と呼ばれる考え方に基づいている。非オープンソース開発のように、少数の専門家や特定の企業などの組織内に閉じた情報収集は避け、多数の現場の人による自発的な情報発信と相互レビューによって、より実情に即したベストプラクティスの収集と整理が進むと期待したものからである。

現段階でのベストプラクティスの件数を問題分類毎に表 1 に示す。

表 1. プラクティスとベストプラクティスの分類

プロセス	問題分類	プラクティス	ベストプラクティス
要求獲得	要求獲得	7	3
	ステークホルダ分析	9	3
	業務知識	6	2
	真の要求(ユーザとベンダの相互理解)	7	0
	目的	6	3
	要求発散	18	6
	非機能要求	2	1
	ファシリテータ	1	1
			56
要求分析	分析・評価	9	9
	整合性・矛盾の解消	7	3
	実現可能性	1	0
	方法論	1	0
		18	12
仕様記述	仕様記述	9	2
	文書化	3	1
	文書・用語	3	0
		15	3
要求検査	検証・レビュー	7	1
要求管理	プロセス	1	0
	プロジェクト管理	2	0
	要求定義の長期化	2	0
	要求変更	2	0
	スキル・人材	5	0
	ユーザの人事異動	3	0
	取引慣行	2	0
	経営層の支援	1	0
		18	0
合計		114	35

5. 要求工学ベストプラクティスの記述:ベストプラクティスパターン

ベストプラクティスとして統一的に理解し、参照できるようにするため、ベストプラクティスの記述形式を定めた。

この記述形式は、ソフトウェアの良い設計方法を記述する形式として広く利用されているデザインパターンの記述形式を参考に、要求工学のベストプラクティスに記述に適するようにした。これを、「要求工学ベストプラクティスパターン」とも呼んでいる。

特に、要求開発・管理では、対象の業務や組織によって類似の問題であっても異なる解決方法があることから、図2に例示するように、問題と解決方法を対にして記述している。さらに、問題の背景によって解決方法も異なる。問題とその背景とをあわせて記述することとし、かつ、事例を具体的に盛り込んで記述するようにした。

「要求工学ベストプラクティスパターン」の具体的な記述方法は、付録の要求工学ベストプラクティスパターン記述形式を参照願いたい。

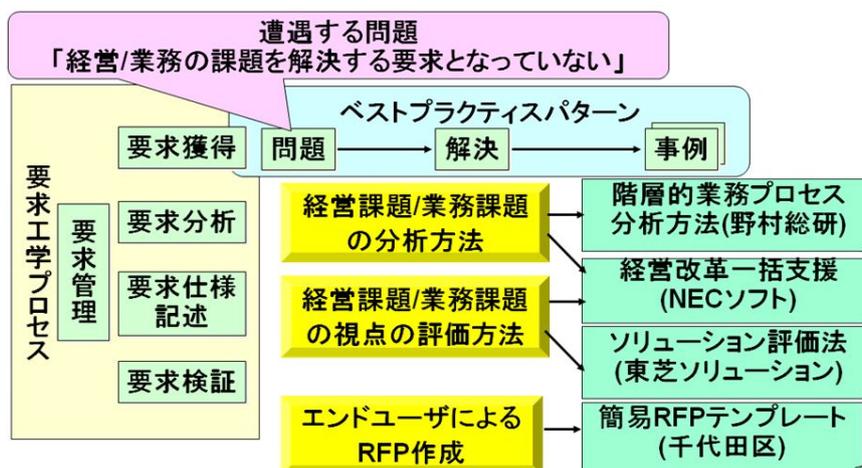


図2. 要求工学ベストプラクティスパターン

6. 要求工学ベストプラクティスの整理、体系化、レビュー

収集したプラクティスとその中から選ばれたベストプラクティスは、要求工学プロセスに対応づけて、次の5分類に分けた。さらに、特定のプロセスに限定されず、要求開発・管理全体を通して適用できるベストプラクティスを「要求工学戦略」として番号を付した。

- (1) 要求獲得
- (2) 要求分析
- (3) 要求仕様記述
- (4) 要求検査
- (5) 要求管理
- (6) 要求工学戦略

ベストプラクティスの記述の妥当性、整合性、公平性、などを委員会でレビューしている。このレビューは、今後も継続し、より多くのベストプラクティスを収集するとともに、内容の分かりやすいベストプラクティスにする予定である。

7. まとめ

要求工学の問題は大規模なシステム開発の問題の中核として、情報サービス産業を挙げて、現場における要求開発・管理への本格的な取り組みが必要である。

本調査研究によって、現場での要求開発・管理の多くの問題が抽出され、その中で、優れた解決事例をベストプラクティスとしてまとめることができた。これは、現場の実情に即したベストプラクティスを現場の視点から組織的に収集できた点で、大きな成果であると考えている。多くの現場で要求開発・管理を実践する上で参考にし、要求開発・管理の「現場力」を高めて頂きたい。

付録. 要求工学ベストプラクティスパターン記述形式

<h1>ベストプラクティスの表記法</h1>	
ベストプラクティスの内容を分類したキーワード(必須) 要求工学ベストプラクティス	ベストプラクティス名称(必須) No. A01019
番号(必須:最初の1文字は分類)	課題とその発生の前提(必須)
問題分類: 要求獲得 作成者: 前提と課題(開発形態、ドメイン、組織形態): [前提] ベンダ(SIer)のプロジェクトリーダー、担当SEが当該業務、当該システムに精通していない場合を想定している。ベンダが最適者をアサイン出来れば問題ないが、一般的には必ずしも最適者をアサインできない場合がある。 [課題] 本来あるはずの、レアケース、例外に関する要求が出てこない。顧客は自分(自部門、自社)に関心・関係のある要求は、こと細かく出てくるが、関心外あるいは視点外の要求はおろそかになり、出てこない場合が多い。(以下略、本文参照のこと)	ベストプラクティス: 要求を漏れなく獲得 作成日: YYYY年MM月DD日
ベストプラクティス [対象プロセス] 要求獲得 [主導者] ベンダ [解決の視点] 予防 [解決方法]	ベストプラクティスの内容(必須) 対象プロセス: ベストプラクティスが解決する要求工学プロセス 主導者: 解決を主導すべき人(ベンダ, ユーザ, ファシリテータ) 解決の視点: 問題発生の予防, 問題の発見, 発生した問題の解決 解決方法: 予防, 発見, 発生した問題の解決の具体的方法
備考: 関連ベストプラクティス、効果、制約: [関連ベストプラクティス] 要求を漏れなく獲得(A01-019) (以下略、本文参照のこと) [制約] 当該顧客・当該プロジェクトで重要なこと、優先すべきことはなにかを事前に判断・打合せて(以下略、本文参照のこと) [参考資料] [1] JIS X0129-1:2003、ソフトウェア製品の品質—第1部:品質モデル、2003。(以下略、本文参照のこと)	備考(オプション): 関連ベストプラクティス, 期待効果, 実践上の制約, 事例へのリンク, 参考資料,

はじめに

本報告書は、現在、情報システム開発において最も重要な課題となっている要求開発・管理を現場で実践したノウハウを整理し、「要求工学ベストプラクティス」として整理したものです。要求工学を実践する上で遭遇する問題を解決し、現場の要求開発力を高めて頂きたいと思います。

従来から、情報システム開発の最上流工程である要求分析は、システム開発の成否を握る鍵として、最も重要な工程として認識されてきました。近年の、情報システムの規模と複雑度の増大、コスト制約や納期の厳格化に伴い、要求分析の重要性はますます高まっています。

一方、要求分析はこれまで個人のスキルによるところが大きく、情報システム開発技術の中でも工学的な取り組みが遅れていた分野です。要求工学を現場で活用することが重要かつ緊急の課題です。しかし、要求分析工程は、技術面に留まらず、ユーザとベンダとの間のビジネス慣行などに大きく依存します。本報告書では、このような背景を考慮し、現場の視点から要求工学の実践のノウハウを収集し、「要求工学ベストプラクティス」として、広く利用できるように整理したものです。

本報告書は、平成 18 年度に実施した調査研究報告書「要求開発ベストプラクティスが示す成功パターン」の調査研究」の続編です。平成 18 年度は、要求開発におけるベンダとユーザの間の組織的な取り組みに着目してまとめています。平成 19 年度は、現場で要求開発を実践するノウハウに着目しています。

「要求工学ベストプラクティス」を整理するに当たり、現場における要求工学の問題とその実際の解決事例を対にしてドキュメント化する方法と採りました。これによって、現場で遭遇する問題の視点から、それを解決する具体的方法としてベストプラクティスへ辿り着けるでしょう。問題は、要求の獲得、分析、要求仕様記述、要求検査、要求管理のそれぞれのフェーズ毎に整理しています。解決方法は、具体的な方法に加え、ベストプラクティスを実践した要求開発の事例が参照できるようにしてあります。本報告書でまとめた「要求工学ベストプラクティス」を現場で是非活用して頂きたいと思います。

本調査研究が JISA 会員企業各位の開発の指針の一助になれば幸いです。

最後に、本調査研究を実施するに当たり、ご指導、ご高配いただきました経済産業省に厚く御礼申し上げます。

最後に、本報告書をまとめるにあたりご尽力いただいた委員、講師各位に深謝いたします。

平成 20 年 3 月

(社)情報サービス産業協会

ソフトウェアエンジニアリング研究会 要求工学 WG

座長 青山 幹雄

平成 19 年度 ソフトウェアエンジニアリング研究会 要求工学 WG 委員名簿

委員長	青山 幹雄	南山大学 数理情報学部 情報通信学科 教授
委員	矢野 達也	アイエックス・ナレッジ(株) 品質管理部 PMO 担当部長
	菊島 靖弘	(株)アイネス 損保システム本部 フェロー
	大竹 学	伊藤忠テクノソリューションズ(株) システムエンジニアリング室 SI 推進部 SI 開発環境推進課
	桶谷 貴弘	(株)インテック 技術本部技術部
	渋谷 照夫	NECインフロンティア(株) 国内営業事業本部 / システム開発事業部 シニアエキスパート
	加藤 久美子	NECソフト(株) 生産技術部リーダー
	小安 秀一	NECネクサソリューションズ(株) システム技術部統括マネージャー
	小池 輝明	NECネクサソリューションズ(株) 技術開発事業部 マネージャー
	後藤 卓史	(株)構造計画研究所 ソフト工学センター
	田中 等	(株)シーエーシー ビジネス支援本部 アプリケーション・アーキテクトグループ
	安田 哲平	(株)CSKシステムズ プロジェクト支援部 生産技術課 主事
	千秋 徹	(株)大和総研 人事部人材開発・採用課 次長
	杉岡 克也	ティーディーシーソフトウェアエンジニアリング(株) 社会情報システム本部 第1システム統括部 第2システム部 担当課長
	村田 尚彦	東芝ソリューション(株) IT 技術研究所 研究開発部 IT 品質ラボラトリー 主任研究員
	鎌田 真由美	日本アイ・ビー・エム(株) 東京基礎研究所 インフォメーション&インタラクション担当 部長
	新保 康夫	日本コンピューター・システム(株) 管理本部 IT 調査研究室 室長
	松井 清	日本事務器(株) クオリティ推進部 担当部長

委員	森永 善宏	日本電子計算(株) 生産管理本部 生産技術部
	荒生 知之	(株)野村総合研究所 システムデザインコンサルティング部 上級システムコンサルタント
	村山 武士	日立公共システムエンジニアリング(株) ソリューション第4事業部 研究・開発センタ センタ長
	眞木 康裕	三菱総研DCS(株) 金融カード本部 金融カード営業部システム開発室
	袴田 典之	(株)菱友システムズ ビジネスソリューション事業部 事業推進室PMOグループ
オブザーバ	塚本 英昭	(独)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 企画グループ
	中谷 多哉子	筑波大学大学院 ビジネス科学研究科 准教授 博士(学術)
	堀 昭三	安川情報システム(株) 品質保証本部 品質保証部 部長
事務局	沼田 克彦	東京電力(株) システム企画部 新配電システムプロジェクトグループマネージャー
	鈴木 律郎	(社)情報サービス産業協会 調査企画部 主任調査役

平成 19 年度 要求開発・管理ベストプラクティスとその体系化の調査研究

－要求開発・管理の現場力を高めるノウハウの共有－

目次

概要

はじめに

委員名簿

1. 要求開発・管理ベストプラクティスの収集と体系化の調査研究(南山大学 青山幹雄)……	1
1. 1. はじめに……	1
1. 2. 要求工学の概要……	3
1. 2. 1. 要求工学とは……	3
1. 2. 2. 要求開発プロセス……	4
1. 2. 3. 要求とは:業務要求, システム要求, ソフトウェア要求……	4
1. 2. 4. 機能要求と非機能要求……	5
1. 2. 5. 要求獲得……	6
1. 2. 6. 要求分析……	7
1. 2. 7. 要求仕様記述……	7
1. 2. 8. 要求検査……	8
1. 3. 要求工学ベストプラクティスとは……	8
1. 3. 1. 要求工学ベストプラクティスの収集……	8
1. 3. 2. 要求工学ベストプラクティスの記述:要求工学ベストプラクティスパターン……	9
1. 3. 3. 要求工学ベストプラクティスの整理と体系化……	10
1. 3. 4. 要求工学ベストプラクティスのレビュー……	10
1. 4. まとめ……	10
付録 1. 1 要求工学ベストプラクティスパターンの記述形式……	13
付録 1. 2 要求工学ベストプラクティスパターンの記述例……	14
付録 1. 3 要求工学ベストプラクティス一覧……	18
1. 5. 要求開発・管理ベストプラクティス……	21
2. お客様にわかりやすい情報システムの設計図作りをめざして～発注者ビューガイドライン～ (富士通 銀林純)……	73
2. 1. はじめに……	73
2. 2. 発注者ビューガイドラインの構成について……	73
2. 3. 発注者ビューガイドライン(画面編)について……	74

2. 3. 1. 発注者ビューガイドライン(画面編)の策定と公開	74
2. 3. 2. 設計書の書き方に関するコツ	74
2. 3. 3. レビューに関するコツ	74
2. 4. 今後の活動について	76
3. ユーザ自身によるRFP作成を可能にするRFPテンプレートとRFP作成プロセス及び実践事例(MPUF RFP 研究会 立花豊文)	77
3. 1. はじめに	77
3. 2. 求められるユーザ主導によるRFP作成	77
3. 2. 1. ユーザとベンダの要件定義に関する見解	77
3. 2. 2. なぜ、ユーザ自身がRFPを作成すべきなのか	78
3. 3. ユーザ主導によるRFP作成を実現する方法	79
3. 3. 1. 業務要件3点セットとRFP作成ツール	79
3. 3. 2. 作成プロセス	80
3. 3. 3. RFP作成チームの運営と合意形成	82
3. 4. 実践例	82
3. 5. 得られた知見	83
3. 6. まとめ	85
3. 7. 最後に	85
4. 要求定義におけるITコーディネータの活用事例(NECソフト 加藤久美子)	87
4. 1. はじめに	87
4. 2. ITCプロセスと要求定義	87
4. 3. NECソフトのITC活用の取り組み	88
4. 4. ITCの活用事例	90
4. 4. 1. ITCの活用事例(その1)	90
4. 4. 2. ITCの活用事例(その2)	93
4. 5. 課題	95
4. 5. 1. 個人の活用から組織的な活用へ	95
4. 5. 2. 要求定義で求められるノウハウ、スキル	95
4. 6. おわりに	96
5. 要求開発プロセスの分担制による要件品質向上事例(NECネクサソリューションズ 小池輝明)	97
5. 1. はじめに	97
5. 1. 1. 要求開発プロセスと問題点	97

5. 1. 2. 事例の問題点	97
5. 1. 3. 「要求定義」と「要件定義」	97
5. 1. 4. 問題解決のための体制	98
5. 2. プロジェクトの概要	98
5. 2. 1. 要件定義の長期化による後工程の先行着手	98
5. 2. 2. 顧客対応と開発パートナーの板挟み	98
5. 2. 3. 開発プロジェクトの規模	99
5. 3. プロジェクトの問題点	99
5. 3. 1. 要件定義フェーズ長期化	99
5. 3. 2. 五月雨式開発のリスク	99
5. 3. 3. 後工程で発生する問題	100
5. 4. 問題解決の方針	100
5. 4. 1. 要件品質の作り込み	100
5. 4. 2. 次工程へのスムーズな移行	102
5. 4. 3. 後工程で発生する問題の歯止め	102
5. 5. 2 チーム制の導入	102
5. 5. 1. 目的別チーム編成	102
5. 5. 2. 要件獲得チームと仕様作成チームの役割	103
5. 5. 3. ツールの導入	103
5. 5. 4. 要件獲得チームと仕様作成チームのインターフェース	104
5. 6. 実施作業内容	105
5. 6. 1. エラーの洗い出し	105
5. 6. 2. 要件品質の向上	106
5. 6. 3. 効果	107
5. 6. 4. 制限事項	108
5. 7. 考察	108
5. 7. 1. 要件品質について	108
5. 7. 2. 仕様作成チームの貢献	108
5. 7. 3. 効果計測	109
6. 上流工程における機能分析方法と顧客満足度評価方法の御紹介(東芝ソリューション 村田尚彦)	111
6. 1. はじめに	111
6. 2. 「PSER」分析の紹介	111
6. 2. 1. PSER による機能分析方法	111
6. 2. 1. 1. PSER とは	111

6. 2. 1. 2. PSER での分析事例	113
6. 2. 1. 3. PSER の有効性	114
6. 3. 顧客満足度評価方法	115
6. 3. 1. 顧客満足度とは	115
6. 3. 2. 家電製品の顧客満足度算出方法	115
6. 3. 3. システムでの顧客満足度の算出方法	116
6. 3. 3. 1 顧客要求の抽出、整理	117
6. 3. 3. 2. 品質特性の導出	118
6. 3. 3. 3. QFD によるシステムの顧客満足度の算出	120
6. 3. 4. 顧客満足度算出の有効性	120
6.4. まとめ	
7. CSK システムズにおける要求定義と要件定義の事例(CSK システムズ 安田哲平)	123
7. 1. はじめに	123
7. 2. よくある風景	123
7. 3. 要求定義とは	124
7. 4. 要求定義の重要性が増大している背景	126
7. 5. 『超上流工程』の本質的な難しさ	126
7. 6. 解決の基本原則	127
7. 7. 要求定義で要求が確定していく様子	129
7. 8. 要求定義終了後の状態	130
7. 9. 最後に(私見:餅は餅屋)	130
7. 10. 参考資料	131
8. 推進体制に注目した要件定義事例(シーエーシー 田中等)	135
8. 1. はじめに	135
8. 2. 求められるユーザ主導による RFP 作成	135
8. 3. 事例 1:ASP	136
8. 4. 事例 2:社内業務システム	137
8. 5. 事例 3:異なるアーキテクチャで再構築	138
8. 6. 事例 4:複数社にわたる社内業務システムの再構築と統合	139
8. 7. 事例 5:社内業務システム	140
8. 8. まとめ	141

1. 要求開発・管理ベストプラクティスの収集と体系化の調査研究

南山大学 青山 幹雄

1. 1. はじめに

本報告書は、要求工学を現場で実践するためのノウハウを収集し、整理体系化を行い、ベストプラクティスとしてまとめたものである。現段階では、114件のプラクティスを収集し、その中から、35件をベストプラクティスとして整理した。ここで示したベストプラクティスは、すべて、現場で実践してきた経験から生まれたものである。その意味で、多くの現場でこのノウハウを活かして、ユーザから要求を獲得し分析、仕様書としてまとめる、要求開発プロセスの改善を図って頂きたい。

要求開発プロセスは、情報システム開発の成否を握ると言っても過言ではない。しかし、現実には、要求開発プロセスが情報システム開発の問題の最大の要因となっている。情報サービス産業協会の実態調査においても、図 1. 1 に示すように、要求開発プロセスは情報システム開発において最も悪い影響を及ぼす可能性があるプロセスであると共に、良い影響を及ぼす可能性のあるプロセスでもある。

我が国の製造業の中で、高品質と低コストを両立させて、国際競争に勝ち抜いた製品の開発では、常に、上流工程から品質の作り込みに取り組んできた。情報システムの開発でも、上流工程である要求開発プロセスの改革と改善が急務である。

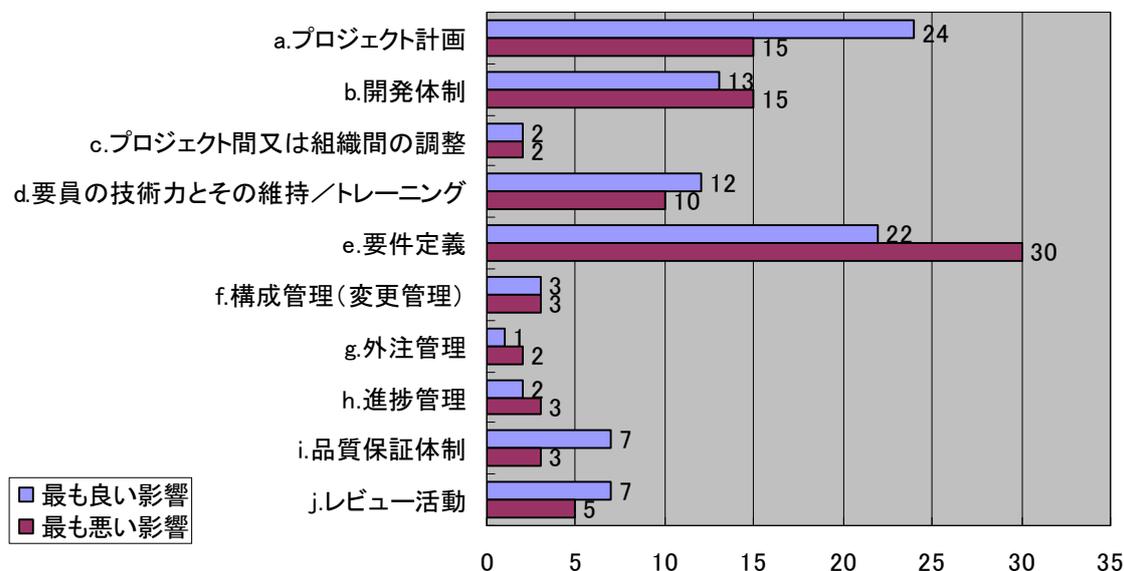


図 1. 1 情報システムの品質に影響を及ぼす要因

このような要求開発プロセスの重要性の認識は、近年、情報サービス産業全体で広がり

つつある。しかし、実際に要求開発プロセスを改善するためには、解決すべき課題が多い。図 1. 2 は、情報サービス産業協会の実態調査[2]において、要求開発プロセスにおける課題を示す。複数回答の中で、次の主要 4 課題だけで 51.6%を占めることに注意すべきである。

- (1) 顧客自身が要求を把握していない(13.6%)
- (2) 顧客要求の頻繁な変更(13.6%)
- (3) 要求内容の相違(16.5%)
- (4) 顧客協力の不足(7.9%)

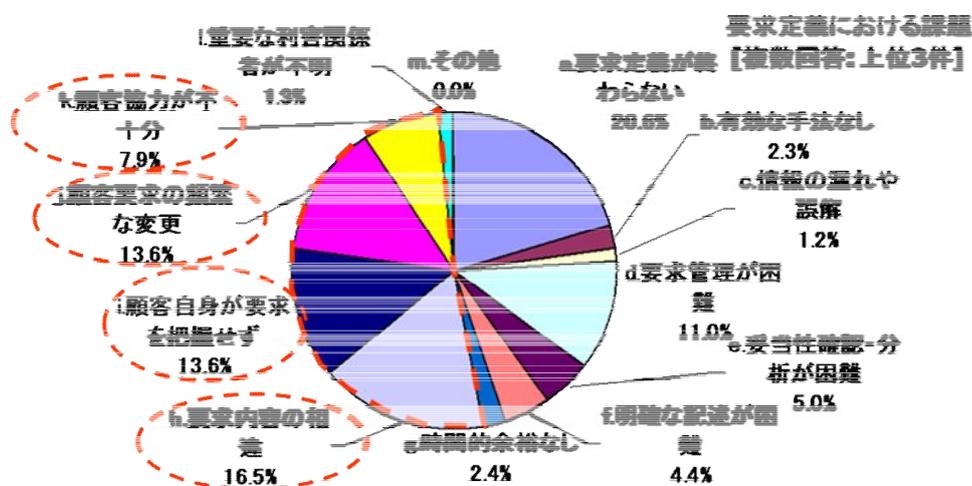


図 1. 2 要求開発プロセスにおける課題

このような実態調査から見てきたことは、要求開発プロセスの改善が、技術面だけでなく、組織面、管理面でも様々な改善を必要とする、多面的で複合的な課題となっていることである。

そのため、情報サービス産業協会の要求工学研究調査委員会では、2006 年度(平成 17 年度)に、まず、組織的な取組みについて調査を行い、ユーザとベンダの間で要求開発を仲介する「ファシリテータ」の活用などの組織的取組みの成功パターンをまとめた[3]。

2007 年度では、さらに、現場における実践技術、管理方法の経験を収集し、この中から、要求開発の課題を整理し、その解決方法の具体例とあわせて「ベストプラクティス」としてまとめた。

収集したベストプラクティスの候補は 100 件を超えたが、この中から、問題の重要性や具体的事例の内容、ベストプラクティスとしてまとめることの可能性などを議論し、35 件のベストプラクティスとして、まとめた。

ベストプラクティスはすべて、問題とその解決方法の対として、統一した形式でまとめられている。さらに、主要なベストプラクティスについては、実践の具体的内容を報告書の中で紹介している。このようなベストプラクティスを参考にして、要求定義プロセスの「現場力」を高めて頂きたい。

なお、要求工学ベストプラクティスの収集は継続して行っており、より、充実したベストプラクティス集として公開する予定である。

1. 2. 要求工学の概要

1. 2. 1. 要求工学とは

要求工学とは、情報システムに対する要求を組織的、体系的に扱う技術体系である。

要求分析や要求定義に関する技術の研究、開発、実践は、構造化分析・設計の手法が提唱された 1970 年代から取り組まれてきた。しかし、それは、特定の手法を中心に行っていた。

1993 年に、要求分析分野の研究者や実務家が集まり、要求工学(Requirements Engineering)を冠する第 1 回の国際会議を開催した。これを契機に、要求工学の概念が世界的に広まり、要求定義のための諸活動を組織的、体系的に取り組む方向に発展した。

現在では、要求工学の技術体系を図 1. 3 に示す枠組みで捉えている[1]。

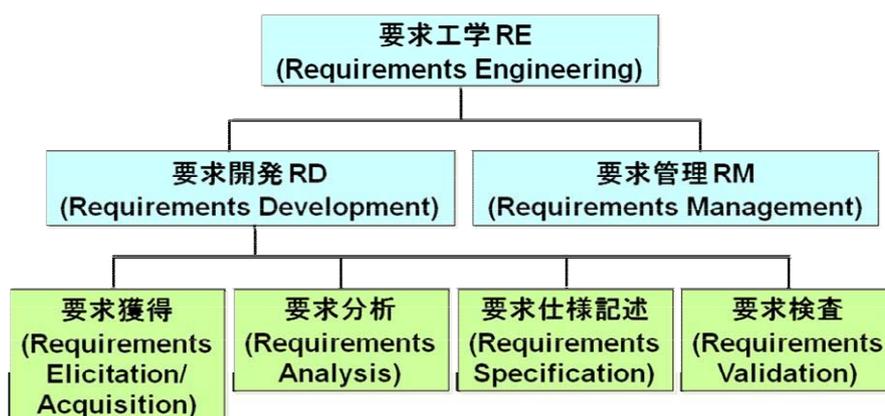


図 1. 3 要求工学の枠組み

ここで、要求定義のための活動は「要求開発」と呼び、次の 4 つの活動から成る。

- (1) **要求獲得**：顧客を含むステークホルダを明らかにし、会議やインタビューなどを通して要求を引出す。ここでは、これを要求項目と呼ぶ。
- (2) **要求分析**：要求項目を整理し、その間の関係を付け、さらに、優先順位付けなどを行い、実現すべき要求を明らかにして、絞り込む。
- (3) **要求仕様記述**：分析された要求を規定の書式や表記法で記述する。
- (4) **要求検査**：要求の間の矛盾がないことや、必要な顧客の要求項目を満たしていることなどを確認する。

1. 2. 2. 要求開発プロセス

要求開発プロセスは、図 1. 4 に示すように、要求開発の四つのプロセスが段階的に実行される。注意すべきは、この四つのプロセスは、1 回実行すれば要求が確定するというわけではない。必要に応じて、これらのプロセスが繰り返し実行される。

要求管理は、要求仕様書をはじめとする要求開発の成果物の変更管理、進捗管理、要求検査による品質確保など、要求開発プロセスの管理に関する諸活動を総称したものである。

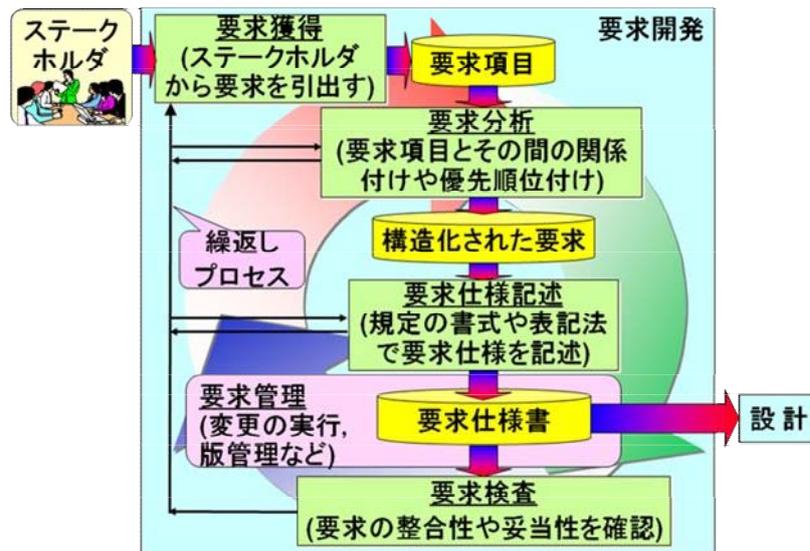


図 1. 4 要求開発プロセス

1. 2. 3. 要求とは: 業務要求、システム要求、ソフトウェア要求

情報システムに対する要求は、図 1. 5 に示すように、次の 3 段階で捉えることができる。

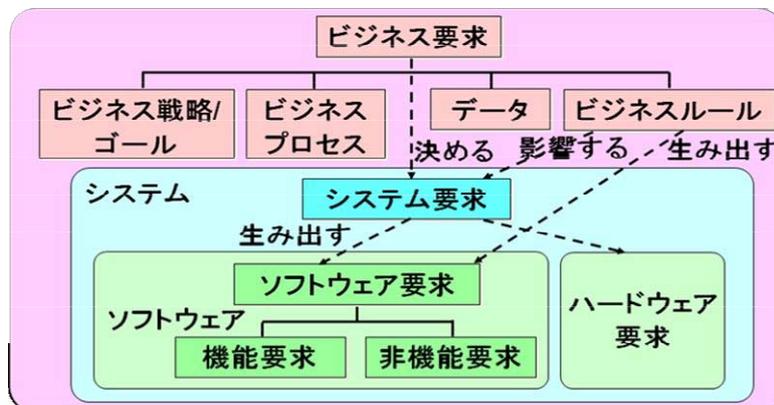


図 1. 5 情報システムの要求

- (1) **ビジネス要求(業務要求)**: ビジネスゴールを達成するために、業務の現状と達成すべき姿や製品を定義する。業務に従事する人やユーザを含めて、ビジネスルールやコスト、資産などの制約の下で、ビジネスプロセスやその実行形態を定義する。
- (2) **システム要求**: ビジネス要求を満たすために情報システムのハードウェアとソフトウェアが果たすべき要求を定義する。
- (3) **ソフトウェア要求**: 情報システムへの要求の中でソフトウェアに対する要求を定義する。

1. 2. 4. 機能要求と非機能要求

図 1. 6 に示すように、要求は機能要求(FR: Functional Requirements)と非機能要求(NFR: Non-Functional Requirements)の二つに分類できる。

機能要求とは、情報システムやソフトウェアが「行うべきこと」である。例えば、ワープロであれば、編集や印刷が機能要求となる。一方、非機能要求とは、「いかに良く機能を実行するか」である。品質特性と言ってもよい。上記のワープロの例では、処理速度や使い勝手などである。

機能要求の定義は従来の要求定義でも行われてきた。しかし、非機能要求の定義はシステム毎に異なり、必ずしも確立されているとは言えない。個別のシステム毎に、その定義方法を決めておく必要がある。

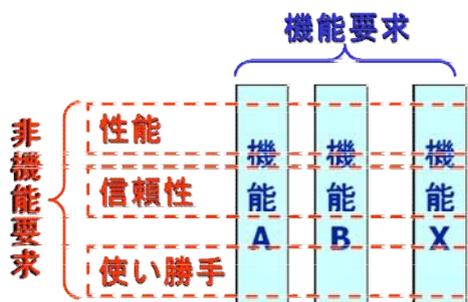


図 1. 6 機能要求と非機能要求

非機能要求を考えるには、表 1. 1 に示す、ISO/IEC 9126 の JIS 版である JIS X0129「ソフトウェア製品の品質」を参考にすると良い[4]。

表 1. 1 ISO/IEC 9126(JIS X0129)のソフトウェア品質特性

品質特性	副品質特性
機能性 (Functionality)	合目的性 (Suitability)、正確性 (Accuracy)、相互運用性 (Interoperability)、セキュリティ (Security)、適合性 (Compliance)
信頼性 (Reliability)	成熟性 (Maturity)、障害許容性 (Fault Tolerance)、回復性 (Recoverability)、信頼性標準適合性 (Compliance)
ユーザビリティ (Usability)	理解性 (Understandability)、習得性 (Learnability)、操作性 (Operability)、魅力性 (Attractiveness)、適合性 (Compliance)
効率性 (Efficiency)	時間効率性 (Time Behavior)、資源効率性 (Resource Behavior)、効率性標準適合性 (Compliance)
保守性 (Maintainability)	解析性 (Analyzability)、変更性 (Changeability)、安定性 (Stability)、試験性 (Testability)、適合性 (Compliance)
移植性 (Portability)	環境適応性 (Adaptability)、設置性 (Installability)、共存性 (Co-existence)、置換性 (Replaceability)、適合性 (Compliance)

1. 2. 5. 要求獲得

要求獲得では、獲得する要求の源泉と獲得方法の二つが鍵となる。要求の源泉としては、次の要素を考慮する必要がある。

- (1) **ビジネスゴール**：情報システムやソフトウェアがビジネスプロセスの効率化など達成すべきビジネス上の目標。
- (2) **ステークホルダ**：要求に関与する人や組織。ユーザの組織における経営者、情報システム部門、エンドユーザ、ベンダの組織、あるいは、監督官庁などを含む。
- (3) **ビジネスルール**：要求の前提となるビジネスルールであり、ビジネス上の制約、ユーザ内や業界内の諸標準など。
- (4) **ドメインの知識**：ユーザの業務やユーザの業界に関する知識、製品に関する知識など、ユーザやその情報システムに関する知識。
- (5) **実行環境**：情報システムやソフトウェアが実行するハードウェアやソフトウェアの環境。

さらに、要求獲得の方として、次のような方法が知られている。

- (1) **インタビュー**：エンドユーザや情報システム部門などステークホルダへのインタビューを行う。
- (2) **シナリオ(ユースケース含む)**：システムとは、ユーザやステークホルダとシステムとの関わりを時系列で記述したものである。ユースケースは、図的な表現で情報システムの機能とその関係を表し、各機能毎にシナリオを用いて機能の内容を記述する方法である。
- (3) **プロトタイプ**：ユーザインタフェースなど、人の感性に訴える要求に関しては、実際に使ってみないと要求を明確にすることが難しい場合がある。プロトタイプは、システ

ムの一部を簡易に作成し、ユーザが試用して要求の抽出や確認を行う。

- (4) **ファシリテーション**：ユーザとベンダの間に立って、第三者の立場から要求の抽出や分析などを仲介し、要求開発を円滑に行う組織的な方法である。
- (5) **ユーザ観察**：ユーザの作業やシステムの利用方法を観察することにより、システムの問題点や要求を洗い出す方法である。

1. 2. 6. 要求分析

要求分析では、システムとソフトウェア要求の範囲を決定し、要求間の関係を明らかにし、矛盾や競合のない 妥当な要求を抽出する。

(1) 要求の分類

(2) **概念モデリング**：要求の特性に応じたモデル化を行う。

(3) アーキテクチャの検討と要求のサブシステムやコンポーネントへの割当

要求を実現する情報システムやソフトウェアシステムは、必ずしも自由に実現選べるとは限らない。実行環境や抽出した要求とのバランスの中で、実現可能なアーキテクチャを検討し、要求仕様の実現可能性を検討する。また、必要に応じて、性能や信頼性などの非機能要求の実現可能性や、実現するための要求仕様上の課題を検討する。必要に応じて、サブシステムやコンポーネントの検討も行う。

(4) **要求のネゴシエーション**：要求仕様の実現可能性や製薬条件などをユーザと調整する。

1. 2. 7. 要求仕様記述

分析した要求を公式な要求仕様書としてまとめる。要求に対する観点によって、図 1. 7 に示す、次のような表記法がある。

- (1) **機能の観点に基づく仕様を記述**：データの流れてに沿って機能を洗い出す：データフロー図(DFD: Data Flow Diagram)など。
- (2) **データの視点に基づく仕様を記述**：データとその間の関係を整理する実体関連図(ER: Entity Relationship Diagram)など。
- (3) **オブジェクト指向に基づく仕様記述**：機能、データ、ならびに、システムの動的特性に関する表記法などの複数の表記法を統合した UML が標準となっている。

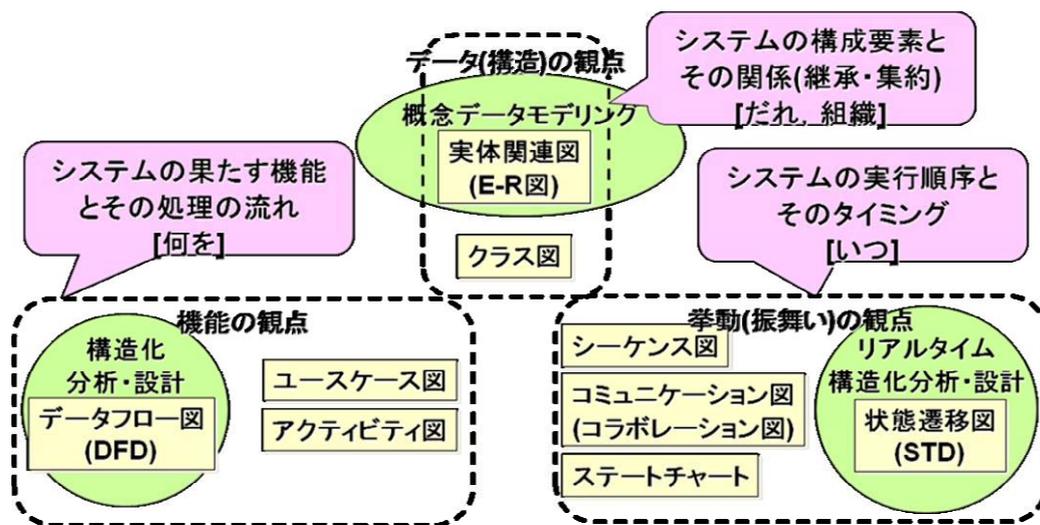


図 1. 7 三つの視点と主な要求仕様記述の方法

1. 2. 8. 要求検査

要求検査は、次の二つの側面がある

- (1) 要求それ自体に矛盾がないことや冗長でないことなどの確認：レビューによる確認や、ツールを使った確認
- (2) 顧客の要求と要求仕様とが一致することの確認：顧客とのレビューやプロトタイプによる確認、ネゴシエーションを通じた確認など。

1. 3. 要求工学ベストプラクティスとは

要求開発と管理における課題は、ユーザの業務や環境など、多様である。従って、要求工学を実践するためには、このような業務や環境に応じて、適切な技術の選択や調整が必要である。そのため、現場では、要求開発を実践する上で、ユーザとベンダとの関係や開発の実情、背景を踏まえて様々な工夫がなされている。このような現場の知恵を共有することこそが、要求工学を現場で実践する助けとなる。

本報告書では、このような現場のノウハウを委員会で収集、レビューし、適切と思われるノウハウを「要求工学ベストプラクティス」としてまとめた。

1. 3. 1. 要求工学ベストプラクティスの収集

ベストプラクティスの収集にあたっては、できるだけ多く収集するために、特別の条件を付けず、委員会の委員全員が経験に基づき提案する方針をとった。これは、オープンな組織によってできるだけ多くの関係者の力を結集するという、オープンソース開発や

Web2.0のフォークソノミー(大衆による分類)と呼ばれる考え方に基づいている。非オープンソース開発のように、少数の専門家や特定の企業などの組織内に閉じた情報収集は避け、多数の現場の人による自発的な情報発信と相互レビューによって、より実情に即したベストプラクティスの収集と整理が進むと期待したものからである。

一方、委員会の定例では、毎回、要求開発の実践事例の紹介も並行して継続的に行い、事例の中で、要求開発の鍵となる要素を掘り下げて討議し、その中からもベストプラクティスを抽出して、まとめた。これらのベストプラクティスは、事例へのリンクをとることにより、理解が深まるようにした。

現段階でのベストプラクティスを分類した一覧を表 1. 2 に示す。

1. 3. 2. 要求工学ベストプラクティスの記述: ベストプラクティスパターン

ベストプラクティスとして統一的に理解し、参照できるようにするため、ベストプラクティスの記述形式を定めた。

この記述形式は、ソフトウェアの良い設計方法を記述する形式として広く利用されているデザインパターンの記述形式を参考に、要求工学のベストプラクティスの記述に適するようにした。これを、「要求工学ベストプラクティスパターン」とも呼んでいる。

特に、要求開発・管理では、対象の業務や組織によって類似の問題であっても異なる解決方法があることから、図 1. 8 に例示するように、問題と解決方法を対にして一つのパターンに記述している。さらに、問題の背景によって解決方法も異なる。問題はその背景と問題とを記述することとし、あわせて、できるだけ具体的事例を盛り込んで記述するようにした。

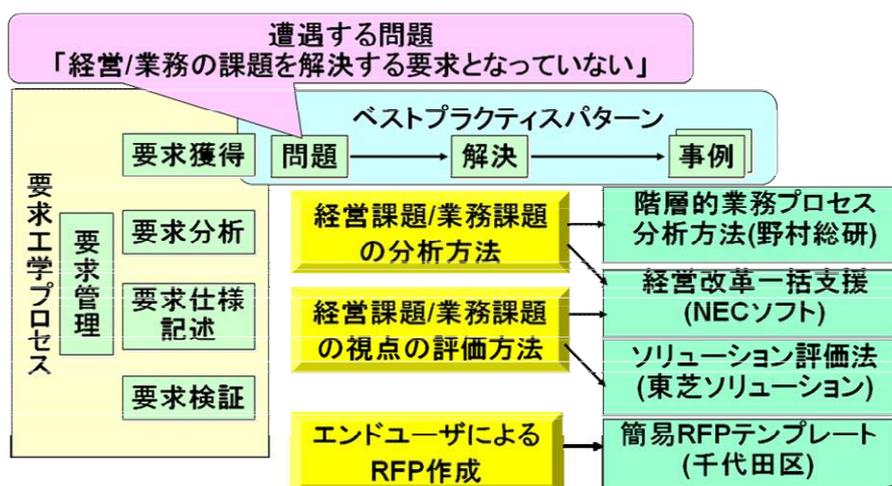


図 1. 8 要求工学ベストプラクティスパターン

また、記述形式もベストプラクティスの収集と整理のプロセスの中で改善した。

ベストプラクティスの記述形式を付録 1. 1 に、記述例を付録 1. 2 に示す。

1. 3. 3. 要求工学ベストプラクティスの整理と体系化

収集したプラクティスとその中から選ばれたベストプラクティスは、要求工学プロセスの要素に対応づけて、次の 5 分類に分けて整理し、番号を付した。

- (1) 要求獲得(No.A01~)
- (2) 要求分析(No.N01~)
- (3) 要求仕様記述(No.D01~)
- (4) 要求検査(No.V01~)
- (5) 要求管理(No.M01~)

さらに、特定のプロセスに限定されず、要求開発・管理全体を通して適用できるベストプラクティスも見いだした。これを「要求工学戦略」として、次のように分類した。

- (6) 要求工学戦略(No.S01~)

ただし、単一のプロセスに限定されないベストプラクティスや、要求開発プロセス全体を通じたベストプラクティスもあることが分かった。このため、ベストプラクティスとこれらのプロセス要素との対応付けを行っている。この結果を付録 1. 3 に示す。

1. 3. 4. 要求工学ベストプラクティスのレビュー

ベストプラクティスの記述の妥当性、整合性、公平性、などを委員会でレビューしている。このレビューは、今後も継続し、より多くのベストプラクティスを収集するとともに、より内容の分かりやすいベストプラクティスにする予定である。

1. 4. まとめ

今や、要求工学の問題は大規模なシステム開発の問題の中核として、係争の争点ともなっている[5]。情報サービス産業を挙げて、現場における要求開発・管理への本格的な取り組みが必要である。

本調査研究によって、現場での要求開発・管理の多くの問題が抽出され、その中で、優れた解決事例をベストプラクティスとしてまとめることができた。これは、現場の実情に即したベストプラクティスを現場の視点から組織的に収集できた点で、大きな成果である

と考えている。多くの現場で要求開発・管理を実践する上で参考にし、要求開発・管理「現場力」を高めて頂きたい。

ソフトウェア開発のベストプラクティスや要求工学のベストプラクティスについては、他にも収集した例がある[6、7]。これらを含めて、今後、さらにベストプラクティスの収集と整理を進める予定である。

参考文献

- [1] A. Abran, et al. (eds.), Guide to the Software Engineering Body of Knowledge, 2004 Version, IEEE Computer Society, <http://www.swebok.org>.
- [2] 情報サービス産業協会, 平成 17 年度情報サービス産業におけるソフトウェア開発の実態アンケート調査結果(概要), 2007.
- [3] 要求工学調査委員会, 要求開発ベストプラクティスが示す成功パターンの調査研究, 情報サービス産業協会, No. 18-J008, 2007 年 3 月.
- [4] JIS X0129-1:2003 ソフトウェア製品の品質 - 第 1 部: 品質モデル, <http://www.jisc.go.jp>.
- [5] 大和田 尚孝, 吉田 洋平, 小原 忍, 「動かないコンピュータ」裁判の行方, 日経コンピュータ 2008 年 5 月 1 日号, pp. 32-39.
- [6] K.E.Wiegers (渡部 洋子(訳)), ソフトウェア要求: 顧客が望むシステムとは, 日経 BP ソフトプレス, 2003.
- [7] R.R.Young, The Requirements Engineering Handbook, Artech House, 2003.

表 1. 2 プラクティスとベストプラクティスの分類

プロセス	問題分類	プラクティス	ベストプラクティス
要求獲得	要求獲得	7	3
	ステークホルダ分析	9	3
	業務知識	6	2
	真の要求(ユーザとベンダの相互理解)	7	0
	目的	6	3
	要求発散	18	6
	非機能要求	2	1
	ファシリテータ	1	1
		56	19
要求分析	分析・評価	9	9
	整合性・矛盾の解消	7	3
	実現可能性	1	0
	方法論	1	0
		18	12
仕様記述	仕様記述	9	2
	文書化	3	1
	文書・用語	3	0
		15	3
要求検査	検証・レビュー	7	1
要求管理	プロセス	1	0
	プロジェクト管理	2	0
	要求定義の長期化	2	0
	要求変更	2	0
	スキル・人材	5	0
	ユーザの人事異動	3	0
	取引慣行	2	0
	経営層の支援	1	0
		18	0
合計		114	35

付録 1. 1 要求工学ベストプラクティスパターンの記述形式

ベストプラクティスの表記法

<p>ベストプラクティスの内容を分類したキーワード(必須)</p> <p>要求工学ベストプラクティス</p>	<p>ベストプラクティス名称(必須)</p> <p>要求工学ベストプラクティス</p>	<p>番号(必須:最初の1文字は分類)</p> <p>No. A01 019</p>
<p>問題分類: 要求獲得</p>	<p>ベストプラクティス: 要求を漏れなく獲得</p>	
<p>作成者:</p>	<p>作成日: YYYY年MM月DD日</p>	
<p>前提と課題(開発形態、ドメイン、組織形態):</p> <p>課題とその発生の前提(必須)</p> <p>[前提] ベンダ (SIer) のプロジェクトリーダー、担当SEが当該業務、当該システムに精通していない場合を想定している。ベンダが最適者をアサイン出来れば問題ないが、一般的には必ずしも最適者をアサインできない場合がある。</p> <p>[課題] 本来あるはずの、レアケース、例外に関する要求が出てこない。顧客は自分(自部門、自社)に関心・関係のある要求は、こと細かく出てくるが、関心外あるいは視点外の要求はおろそかになり、出てこない場合が多い。(以下略、本文参照のこと)</p>		
<p>ベストプラクティス</p> <p>対象プロセス 要求獲得</p> <p>[主導者] ベンダ</p> <p>[解決の視点] 予防</p> <p>[解決方法] 要求獲得の際に、単純なヒアリングだけでなく、問いかけを行なって要求を引き出すことが有効である。(以下略、本文参照のこと)</p> <p>備考: 関連ベストプラクティス、効果、制約:</p> <p>[関連ベストプラクティス] 要求を漏れなく獲得(A01-019) (以下略、本文参照のこと)</p> <p>[制約] 当該顧客・当該プロジェクトで重要なこと、優先すべきことはなにかを事前に判断・打合せして(以下略、本文参照のこと)</p> <p>[参考資料] [1] JIS X0129-1:2003、ソフトウェア製品の品質 ー 第1部: 品質モデル、2003。(以下略、本文参照のこと)</p>		
		<p>ベストプラクティスの内容(必須)</p> <p>対象プロセス: ベストプラクティスが解決する要求工学プロセス 主導者: 解決を主導すべき人(ベンダ, ユーザ, ファシリテータ) 解決の視点: 問題発生の予防, 問題の発見, 発生した問題の解決 解決方法: 予防, 発見, 発生した問題の解決の具体的方法</p> <p>備考(オプション): 関連ベストプラクティス, 期待効果, 実践上の制約, 事例へのリンク, 参考資料,</p>

10 All Rights Reserved. Copyright Mikio Aoyama, 2008

付録 1. 2 要求工学ベストプラクティスパターンの記述例

要求工学ベストプラクティス		No. A01
		019
問題分類: 要求獲得	ベストプラクティス: 要求を洩れなく獲得	
作成者: 安田 哲平	作成日: 2008 年 5 月 13 日	
前提と課題(開発形態、ドメイン、組織形態):		
[前提]		
ベンダ (SIer) のプロジェクトリーダー、担当 SE が当該業務、当該システムに精通していない場合を想定している。		
ベンダが最適者をアサイン出来れば問題ないが、一般的には必ずしも最適者をアサインできない場合がある。		
[課題]		
本来あるはずの、レアケース、例外に関する要求が出てこない。		
顧客は自分(自部門、自社)に関心・関係のある要求は、こと細かく出てくるが、関心外あるいは視点外の要求はおろそかになり、出てこない場合が多い。(後工程になって初めて言い出す、あるいはベンダ側で気づいて再度ヒアリングするはめになる)		
ベストプラクティス		
[対象プロセス]		
要求獲得		
[主導者]		
ベンダ		
[解決の視点]		
予防		
[解決方法]		
要求獲得の際に、単純なヒアリングだけでなく、問いかけを行なって要求を引き出すことが有効である。		
問いかけのためには、事前のチェックリスト作成が有効である。		
(1) 業務機能要求に関するチェックリスト		
業務機能に関するものは、業務知識がないと例外やレアケースが判らないため、業務知識が		

豊富な要員が中心となってチェックリストを作成し、顧客に問いかけて要求を確認していく。(実プロジェクトでは顧客業界の業務知識の豊富な SE を専任でアサイン出来ない場合があるのでその対策を兼ねている。)

チェックリストが有効なのは主に以下のような場合である。

- 1) 業界の常識や顧客会社の常識は語られない場合が多い。(当たり前すぎて語るべきことだと意識していない。)

例: 物流と商流が異なる。(送り先は小売店直送だが請求は帳合卸商へ等。)

- 2) レアケースや例外はヒアリング相手が言い忘れている場合がある。(現行業務フローにも載っていない場合がある。)

例: ヒアリング時に、商品の返品処理機能は「不良品返品」と「注文取消し」と言われ、チェックリストを見ると返品処理: 「不良品返品」「注文取消し」「商品違い」「運送事故」等と書いてあれば、「商品違い」「運送事故」の扱いはどうするのか問いかけることができる。それぞれ返品後の業務処理が異なる場合が多い。

- (2) チェックリストの具体的な形としては、例えば、顧客業界の「業務機能一覧表」などの体裁をとる。

用語集や業務機能の説明なども一緒に整備しておくが良い。

例:

表 1 業務機能一覧表

業務機能大分類	業務機能小分類	業務機能概要説明
販売管理	見積検索	既存の見積情報の検索。 検索条件: 取引先別、見積年月日別、品目別、など
	見積入力	見積情報の入力。 入力情報: ・見出し情報(件名、売上先、送付先、納期、取引条件 など) ・明細情報(品目、単価(掛率)、数量、など)
	在庫確認
	受注確定
	出荷実績

.....

- (3) システム要求に関するチェックリスト

システムに関する例外・レアケースは多くは非機能要求なので、事前に非機能チェックリストを作成し、それに従って顧客に問いかけて要求を確認していく。

- (4) 非機能チェックリストは、JISX0129[1]などの規格や FISC(金融情報システムセンター)などの業界標準ガイドラインをベースに作成しておくのと抜け漏れがすくない。但し、JISX0129 の品質特性・副特性は分類や用語がわかりづらいので、注釈・例示を備考につけておかないと、一般的な SE 及び顧客には理解し難いので注意する。

例: JISX0129 では、現場でよく使われる「性能」、「可用性」などの用語は直接出てこない。

「機能性」「信頼性」「使用性」「効率性」「保守性」「移植性」という分類である。

例: 非機能チェックリストとして活用可能なリストの例(表 2 [2])

(実際にはもう少しブレークダウンしたものを作る必要がある)

- (5) セキュリティやアクセス権限などの要求は、システム要求であると同時に業務要求でもあり、且つ内部統制など全社運用と絡むので、関連して確認すべき部門(監査部や経営管理部などが想定される)がチェックリスト上で判るようにしておくことと便利である。

- (6) チェックリストを育てていくことも非常に重要である。現場で使用して内容をフィードバックしていく。

表 2 品質条件と RFP の整理[2]

No.	品質特性	副品質特性	影響先	主な品質要件(RFP要件)	
1	機能性	合目的性	生産量	社内ユーザーor社外ユーザー	
2			生産性	要求の記述水準および網羅性	
3		正確性	生産量	テストケース数	
4			生産性	各プロセスのレビュー回数	
5		相互運用性	生産量	他システムとのコード変換数、フォーマット変換数	
6			生産性	他システムとのインタフェース数	
7		セキュリティ	生産量	外部ネットワークとの接続の有無	
8				個人情報、機密情報の取り扱いの有無	
9	信頼性	成熟性	生産量	顧客側の規格・基準の数	
10			生産性	社外の規格・基準の数	
11	使用性	障害許容性	生産量	稼働率の設計値	
12			生産性	異常検知に必要な機能数	
13	効率性	回復性	生産量	MTTRの設計値	
14			生産性	プレゼンテーションツールの数	
15	保守性	理解性	生産量	習得時間の設計値	
16			生産性	操作性の実装水準	
17	移植性	時間効率性	生産量	システム処理能力の実装水準	
18			生産性	資源効率性の要求水準	
19	環境適応性	解析性	生産量	ソフトウェアの保守性の考慮の有無	
20			生産性	コメント行のソースコードに占めるおよその割合	
21	設置性	変更性	生産量	ソフトウェアの拡張性に対する実装水準	
22			生産性	ライフサイクル年数の設計値	
23	置換性	安定性	生産量	試験に必要な機能数	
24			生産性	環境適応性	
25	適応性	試験性	生産量	移植性の要望の有無	
26			生産性		設置性
27	置換性	適合性	生産量	置換性の要望の有無	
28			生産性		置換性

備考：関連ベストプラクティス、効果、制約：

[関連ベストプラクティス]

要求を漏れなく獲得(A01-019)

業界用語・知識の補完(A07-111)

[制約]

当該顧客・当該プロジェクトで重要なこと、優先すべきことはなにかを事前に判断・打合せしてチェックリストに重み付け(項目取捨選択)をしておかないと、「チェックのためのチェック」になって効率が悪くなる。

[参考資料]

[1] JIS X0129-1:2003,ソフトウェア製品の品質—第1部:品質モデル,2003

[2] 情報サービス産業協会,品質ベースの価格設定の可能性に関する調査研究,概要,18-J013,2007, <http://www.jisa.or.jp/report/index-h18-j.html#18-J013>

付録 1.3 要求工学ベストプラクティス一覧

No	問題分類	名称	具体的問題	対象プロセス					プロジェクト管理	スキル	主導者(問題の)			解決の観点		
				獲得	分析	記述	検査	管理			ユーザ	ファシリテータ	ベンダ	予防	発見	解決
A1	獲得	要求を漏れなく獲得	要求が出てこない。	○							○	○	○			
A2	獲得	獲得した要求量の妥当性の確認	レアケース、例外に関する要求が出てこない	○	○						○			○	○	○
A3	獲得	ユーザとベンダ間の認識の不一致解消	定量的な要求を出せない	○					○				○	○		
A4	ステークホルダ	ステークホルダによって異なる要求の調整	利害関係者間の思惑や組織間の意思疎通が要求を見出せなくする場合がある。		○						○			○		
A5	ステークホルダ	キーマンを見つける	多数のステークホルダ間の調整	○					○							
A6	ステークホルダ	キーマンを見つける	キーマンが見えない	○					○							
A7	業務知識	業界用語・知識の補完	業界用語知識の欠如が誤解・抜け・漏れにつながる。			○					○		○			○
A8	業務知識	ユーザのITスキル向上	ユーザの理解を必要とする技術要素や分野が増大している(インフラ系システムの場合)	○							○			○		
A9	目的	要求と経営戦略との整合	要求定義での作成成果物の目的(なにを確認するために作っているのか?)があいまいで、作成したものの客先との同意がとれない。			○					○					
A10	目的	要求と経営戦略との整合	ITコーディネータの活用(提案書の作成)	○	○					○	○	○(ベンダ)	○	○		
A11	要求発散	画面周りの要求を確定させる	画面周りなど、実際に使ってみないと真の要求(少なくとも、担当の顧客が満足するという意味であって、本質的な要求という意味ではない)が確定しない。	○			○	○			○		○	○	○	○
A12	要求発散	要求が止まらない状況を防止	要求が止まらない。FIXしない。	○				○			○	○	○			
A13	要求発散	要求を適切なタイミングで確定	ユーザの定期的な確認	○					○				○	○		
A14	要求発散	計画段階での要求発散や確定遅れの防止	要求(仕様)がPJ遂行中に大幅に追加または変更される。(顧客の経験不足)	○					○		○		○	○		

No	問題分類	名称	具体的問題	対象プロセス					プロジェクト管理	スキル	主導者(問題の)			解決の観点		
				獲得	分析	記述	検査	管理			ユーザ	ファシリテータ	ベンダ	予防	発見	解決
A15	要求発散	計画段階での要求発散や確定遅れの防止	要求(仕様)が[?]遂行中に大幅に追加または変更される。(顧客の経験不足)					○	○		○	○	○			
A16	要求発散		課題解決範囲選定		○						○	○	○			○
A17	非機能要求	非機能要求の形式知化	SLAを作れない	○						○		○	○			
A18	ファシリテータ	ファシリテータの活用	分担:役割分担不明なファシリテータが存在する。責任の在所が不明になる。	○							○	○	○			
N1 (19)	分析評価	要求の優先順位をビジネス全体で捉える	要求の優先順位が客とベンダで違う(重要ではないが判り易い要求から固めたがる)		○						○	○	○			
N2 (20)	分析評価	重要度・緊急度マトリックスによる要求の選別	要求の優先順位が客とベンダで違う(重要ではないが判り易い要求から固めたがる)					○				○		○		
N3 (21)	分析評価	要求の優先順位設定	要求の優先順位が客とベンダで違う(重要ではないが判り易い要求から固めたがる)		○				○			○	○			
N4 (22)	分析評価	To Beの効果を定量的に表現できる仕組み	MI(6σ改善活動)のようにTo Beの効果・KPIを定量的に表現できる仕組みが欲しい。			○					○	○				
N5 (23)	分析評価	課題の検討不足を克服	課題の原因検討		○						○	○	○			○
N6 (24)	分析評価		課題に対する解決策選定		○						○	○	○			○
N7 (25)	分析評価	要求の優先順位の管理	優先順位付けができない	○	○			○					○	○		
N8 (26)	分析評価	優先順位付けに基づいたレビューによる実現する機能の絞り込み	要件が見積りに合わない						○							
N9 (27)	分析評価	要求の規模が納期と予算を超えないようにする	納期と予算があわない						○							
N10 (28)	整合性・矛盾	要求間の矛盾解消	要求間の矛盾		○			○	○			○	○	○		

No	問題分類	名称	具体的問題	対象プロセス					プロジェクト管理	スキル	主導者(問題の)			解決の観点		
				獲得	分析	記述	検査	管理			ユーザ	ファシリテータ	ベンダ	予防	発見	解決
N11 (29)	整合性・矛盾	2チーム制による要求矛盾の回避	要求間の整合性の確認方法	○	○								○			
N12 (30)	整合性・矛盾	要求プロセス内のサイクルを回す	要求間の整合性の確認方法		○		○				○		○	○	○	○
D1 (31)	仕様記述	仕様記述のパラッキを防ぐ	要求分析で担当者個人個人の仕様書記述レベルが違う。			○							○			
D2 (32)	仕様記述	仕様記述の洩れをなくす	客がまとめた要求仕様書が使えるレベルにない。抜けがある。		○											
D3 (33)	文書化	簡易要求分析フレームワーク	見積もり根拠			○			○				○	○	○	○
V1 (34)	検証・レビュー	適切なコミュニケーションによる要求の確認	統合テスト結果の説明を行ったところ、要件定義書記述内容に対するユーザ側の誤認が判明し、仕様変更(手戻り)が発生した。	○			○				○	○	○	○		
S1 (35)	戦略	要求と経営戦略との整合	ITコーディネータの活用(開発プロセス全体の支援)	○	○	△	△	△		○	○	○(ベンダ)	○	○		

1. 5. 要求開発・管理ベストプラクティス

ベストプラクティス執筆者一覧(氏名順、敬称略)

氏名	所属
桶谷 貴弘	(株) インテック
加藤 久美子	NEC ソフト (株)
小池 輝明	NEC ネクサソリューションズ (株)
後藤 卓史	(株) 構造計画研究所
眞木 康裕	三菱総研 DCS (株)
渋谷 照夫	NEC フロンティア (株)
新保 康夫	日本コンピューター・システム (株)
杉岡 克也	ティーディーシーソフトウェアエンジニアリング (株)
袴田 典之	(株) 菱友システムズ
村田 尚彦	東芝ソリューション (株)
安田 哲平	(株) CSK システムズ

要求工学ベストプラクティス		No. A01
		019
問題分類: 要求獲得	ベストプラクティス: 要求を洩れなく獲得	
1 版		
<p>前提と課題(開発形態, ドメイン, 組織形態):</p> <p>[前提] ベンダ(SIer)のプロジェクトリーダー、担当 SE が当該業務、当該システムに精通していない場合を想定している。 ベンダが最適者をアサイン出来れば問題ないが、一般的には必ずしも最適者をアサインできない場合がある。</p> <p>[課題] 本来あるはずの、レアケース、例外に関する要求が出てこない。 顧客は自分(自部門、自社)に関心・関係のある要求は、こと細かく出てくるが、関心外あるいは視点外の要求はおろそかになり、出てこない場合が多い。(後工程になって初めて言い出す、あるいはベンダー側で気づいて再度ヒアリングするはめになる)</p>		
<p>ベストプラクティス</p> <p>[対象プロセス] 要求獲得</p> <p>[主導者] ベンダ</p> <p>[解決の視点] 予防</p> <p>[解決方法] 要求獲得の際に、単純なヒアリングだけでなく、問いかけを行なって要求を引き出すことが有効である。問いかけのためには、事前のチェックリスト作成が有効である。</p> <p>(1) 業務機能要求に関するチェックリスト 業務機能に関するものは、業務知識がないと例外やレアケースが判らないため、業務知識が豊富な要員が中心となってチェックリストを作成し、顧客に問いかけて要求を確認していく。(実プロジェクトでは顧客業界の業務知識の豊富な SE を専任でアサイン出来ない場合があるのでその対策を兼ねている。)</p> <p>チェックリストが有効なのは主に以下のような場合である。</p> <ol style="list-style-type: none"> 1) 業界の常識や顧客会社の常識は語られない場合が多い。(当たり前すぎて語るべきことだと意識していない。) 例: 物流と商流が異なる。(送り先は小売店直送だが請求は帳合卸商へ等。) 2) レアケースや例外はヒアリング相手が言い忘れている場合がある。(現行業務フローにも載っていない場合がある。) 例: ヒアリング時に、商品の返品処理機能は「不良品返品」と「注文取消し」と言われ、チェックリストを見ると返品処理: 「不良品返品」「注文取消し」「商品違い」「運送事故」等と書いてあれば、「商品違い」「運送事故」の扱いはどうするのか問いかけることができる。それぞれ返品後の業務処理が異なる場合が多い。 <p>(2) チェックリストの具体的な形としては、例えば、顧客業界の「業務機能一覧表」などの体裁をとる。</p>		

用語集や業務機能の説明なども一緒に整備しておくが良い。

例:

表 1 業務機能一覧表

業務機能 大分類	業務機能 小分類	業務機能 概要説明
販売管理	見積検索	既存の見積情報の検索。 検索条件:取引先別、見積年月日別、品目別、など
	見積入力	見積情報の入力。 入力情報: ・見出し情報(件名、売上先、送付先、納期、取引条件 など) ・明細情報(品目、単価(掛率)、数量、など)
	在庫確認
	受注確定
	出荷実績

.....

(3) システム要求に関するチェックリスト

システムに関する例外・レアケースは多くは非機能要求なので、事前に非機能チェックリストを作成し、それに従って顧客に問いかけて要求を確認していく。

(4) 非機能チェックリストは、JISX0129[1]などの規格や FISC(金融情報システムセンター)などの業界標準ガイドラインをベースに作成しておくが抜け漏れがすくない。但し、JISX0129 の品質特性・副特性は分類や用語がわかりづらいので、注釈・例示を備考につけておかないと、一般的な SE 及び顧客には理解し難いので注意する。

例:JISX0129 では、現場でよく使われる「性能」、「可用性」などの用語は直接出てこない。「機能性」「信頼性」「使用性」「効率性」「保守性」「移植性」という分類である。

例:非機能チェックリストとして活用可能なリストの例(表 2 [2])
(実際にはもう少しブレイクダウンしたものを作る必要がある)

(5) セキュリティやアクセス権限などの要求は、システム要求であると同時に業務要求でもあり、且つ内部統制など全社運用と絡むので、関連して確認すべき部門(監査部や経営管理部などが想定される)がチェックリスト上で判るようにしておくが便利である。

(6) チェックリストを育てていくことも非常に重要である。現場で使用して内容をフィードバックしていく。

表 2 品質条件とRFPの整理[2]

No.	品質特性	副品質特性	影響先	主な品質要件(RFP要件)
1	機能性	合目的性	生産量	社内ユーザーor社外ユーザー
2			生産性	要求の記述水準および網羅性
3		正確性	生産量	テストケース数
4			生産性	各プロセスのレビュー回数
5			生産量	他システムとのコード変換数、フォーマット変換数
6		相互運用性	生産性	他システムとのインタフェース数
7			生産性	外部ネットワークとの接続の有無
8		セキュリティ	生産量	生産性
9	生産性			セキュアプログラミング採用の有無
10	信頼性	適合性	生産性	顧客側の規格・基準の数
11			生産性	社外の規格・基準の数
12		成熟性	生産量	稼働率の設計値
13	使用性	障害許容性	生産量	異常検知に必要な機能数
14			生産量	MTTRの設計値
15	効率性	回復性	生産量	プレゼンテーションツールの数
16			生産量	習得時間の設計値
17	保守性	理解性	生産量	操作性の実装水準
18			生産量	システム処理能力の実装水準
19		習得性	生産量	システム処理能力の実装水準
20			生産量	資源効率性の要求水準
21	魅力性	解析性	生産量	ソフトウェアの保守性の考慮の有無
22			生産量	コメント行のソースコードに占めるおよその割合
23	移植性	変更性	生産量	ソフトウェアの拡張性に対する実装水準
24			生産量	ライフサイクル年数の設計値
25		安定性	生産量	試験に必要な機能数
	試験性	環境適応性	生産性	移植性の要望の有無
			生産性	
	置換性	適合性	生産性	移植性の要望の有無
			生産性	

備考: 関連ベストプラクティス, 効果, 制約:

[関連ベストプラクティス]

要求を漏れなく獲得(A01)

業界用語・知識の補完(A07)

[制約]

当該顧客・当該プロジェクトで重要なこと、優先すべきことはなにかを事前に判断・打合せしてチェックリストに重み付け(項目取捨選択)をしておかないと、「チェックのためのチェック」になって効率が悪くなる。

[参考資料]

[1] JIS X0129-1:2003, ソフトウェア製品の品質－第1部:品質モデル, 2003.

[2] 情報サービス産業協会, 品質ベースの価格設定の可能性に関する調査研究, 概要, 18-J013, 2007,
<http://www.jisa.or.jp/report/index-h18-j.html#18-J013>.

要求工学ベストプラクティス		No. A02
		020
問題分類： 要求獲得	ベストプラクティス： 獲得した要求の妥当性の確認	
1 版		
<p>前提と課題(開発形態, ドメイン, 組織形態):</p> <p>[前提] ベンダ(システムインテグレータ)のプロジェクトリーダー、担当 SE が当該業務に精通していない場合を想定している。あるいは、ベンダが必ずしも最適者をアサインできない場合がある。</p> <p>[課題] 顧客から要求が出てこない。規模感から考えて本来もっと要求が出てきてよいはずなのに要求が少ないと思われる。本当にこの程度の要求でよいのか、実はもっと隠された要求があるのに聞き出せていないのかがはっきりしていない。</p>		
<p>ベストプラクティス 有識者によるベンダ内での第三者レビューの実施。</p> <p>[対象プロセス] 獲得</p> <p>[主導者] ベンダ</p> <p>[解決の観点] 発見、解決</p> <p>[解決方法]</p> <p>(1) ベンダ内で有識者(顧客業界・業務に詳しい者、顧客システムに詳しい者、開発一般に知見のある者など)とプロジェクトリーダー、担当 SE による第三者レビューを実施する。 (顧客とのレビューの前に、ベンダ内で実施する。)</p> <p>(2) 主なレビューポイント 要求獲得に抜け・漏れは無い、要求のブレークダウンは十分かを確認する。 (一般的なレビューポイントである、スコープの確認や変更点の確認は当然実施する。)</p> <p>1) 同種のシステム、有識者の過去の経験事例との比較をする。 同種のシステム事例と比較して要求内容の比較、要求項目の多寡などを確認する。 要求として出ていないが、一般的にこういう機能を盛り込まないと業務面・システム面でまずいというものがないかを確認する。(過去の経験事例)</p> <p>2) 「現行どおり」などの曖昧で大括りな要求がないかを確認する。 a) 現行どおりという要求は具体的な個別要求に書き下してみると非常に大量の要求項目になる。 b) 対象業務が生産管理や SCM の例で、一言で書かれた要求が実は非常に大きい要求の場合がある。 例:「要求:生産日程計画をダイナミックに変更可能とする。」 日程計画だけでも大日程、中日程、小日程があり、ダイナミックな変更というのが 1 日の中で何回変更可能とするのかを決める必要がある。変更可能なパラメータは何と何にするのか、実績(進捗管理、在庫管理など)からのフィードバックをどのように反映するのか、そもそも生産計画の立て方はどんな方式かなど、要求をもっとブレークダウンして要求を獲得する必要がある。 実際には、スケジューラソフトウェアを呼び出すだけで、そのソフトウェアの機能の範囲内で実現できることが要求の全てである場合も可能性としてはある。</p> <p>(3) レビュー後の対策 レビューで出てきた指摘事項を整理し、不足していると思われるもの、大括りなものは顧客再ヒアリングなどの対策をたてる。</p>		

備考: 関連ベストプラクティス, 効果, 制約:

[関連ベストプラクティス]

要求を漏れなく獲得(A01)

問題分類: 要求獲得	ベストプラクティス: ユーザとベンダ間の認識の不一致解消
------------	------------------------------

1 版

前提と課題(開発形態, ドメイン, 組織形態):

[前提]

顧客とベンダ間で要件の獲得を行なっている段階において、顧客の要求、要望に対して、提供するシステムによって、なにがどのくらいよくなるのかをお互いに認識が異なったまま開発が開始してしまい、開発されたシステムが顧客の期待とは異なるシステムができてしまう。

[課題]

顧客との期待とは異なるシステムを開発してしまうことに対して、事前に調整する手段がない。顧客が要望する機能についての有無についてお互いに確認が可能であるが、機能自体の便利さ等については調整することが難しい。

ベストプラクティス:

[対象ライフサイクル]

要求獲得

[主導者]

ユーザとベンダで、関係するステークホルダが集まり作業を行なうことが望ましい。関係者が集まるのが難しい場合は、一部のメンバにおいて作業を行い、レビューを受けることが望ましい。

[解決の観点]

予防、発見

[目的]

顧客の要望に対して提供するシステムが満足させる度合いを定量的に示すことにより、顧客とベンダ間におけるシステムへの期待値を共有し、顧客の要求に対して異なるシステムの提供してしまう度合いを低減する。

[解決方法]

様々な顧客の要求に対して、満足度の定量化を行なう。定量化するための項目(例: パワーのある掃除機がほしい場合は「吸い込み率」において定量化されている。)を抽出し、満足度を抽出する。

(1) 顧客要求の抽出

顧客の要望をインタビューや KJ 法などにより抽出を行なう。

(2) 要求の整理

(1)で抽出された要求について整理を行なう。作業としては、他の要求と同意な場合はまとめたり、抽象化された要求が他の複数の要求を含んでいる場合は階層構造にするなどの整理を行ない、要求の粒度を整える。

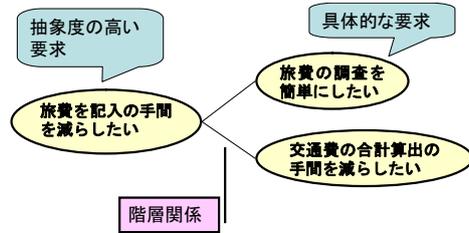


図 1 顧客の要求の整理例

(3) 定量化項目の洗い出し

上記(2)で整理された顧客要求から、ラダーリングなどの手法を利用し、この顧客の要求が必要なのはなぜという「なぜなぜ分析」を行ない、定量化を行なうための項目の抽出を行なう。

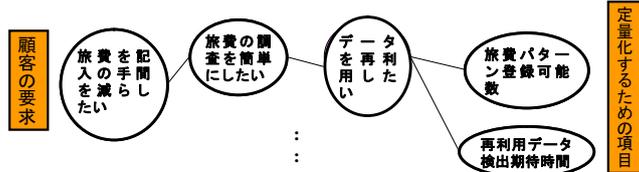


図 2 ラダーリングによる分析

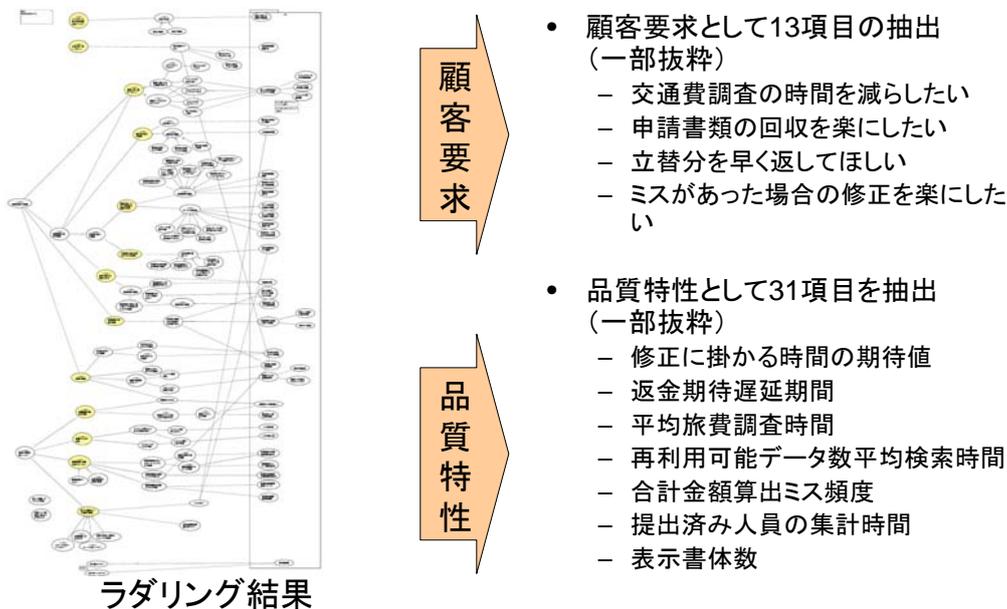


図 3 旅費清算システムについて求められた品質特性

(4) 評価

顧客の期待を定量化する項目において、それぞれ開発するシステムにおいてどのくらいの値がだせるのかを定義、もしくは顧客と協議し決定する。また、顧客要求と定量化された期待値を品質機能展開(QFD: Quality Function Deployment)等を利用すれば、従来のシステムと比較してどのくらい顧客度が上がるのかを評価することが可能となる。QFDはステークホルダ毎に作成することにより、ステークホルダ毎にどの程度満足度が向上するのかを比較することも可能となる。

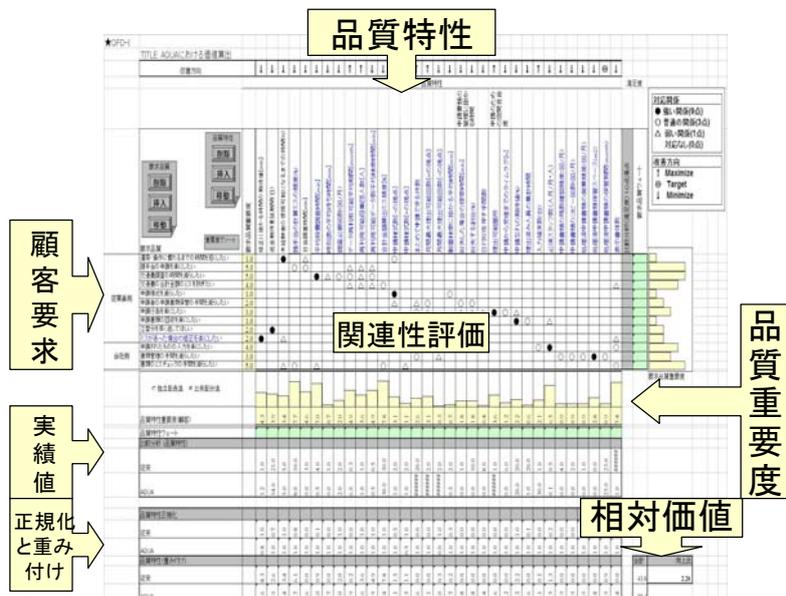


図 4 QFD の記述例

備考: 関連ベストプラクティス, 効果, 制約:

[効果]

顧客の期待値を定量値で把握することができるため、仕様等を決める際に顧客とベンダ間で仕様の同意がとりやすくなる。

[参考資料]

清水 歩、村田 尚彦, 他, 電気学会 電子・情報・システム部門大会, 2007年9月.

要求工学ベストプラクティス		No. A04																																																				
		005																																																				
問題分類: ステークホルダ	ベストプラクティス: ステークホルダによって異なる要求の調整																																																					
1 版																																																						
<p>前提と課題(開発形態, ドメイン, 組織形態):</p> <p>[前提] 複数のステークホルダから要求を獲得する。</p> <p>[課題] 「複数の現場」、「システム部と現場」などステークホルダによって要求が異なったり、ステークホルダ間で要求が矛盾したりする。</p>																																																						
<p>ベストプラクティス:</p> <p>[対象プロセス] 要求分析</p> <p>[主導者] 要求分析者</p> <p>[解決の観点] 予防、解決</p> <p>[解決方法] ステークホルダ分析を実施し、システムにおける役割や位置づけ、ミッションや期待といったステークホルダに関する直接的なものだけではなく、それらから導き出される意思決定における重みづけなど、要求の競合解決に必要としている情報をステークホルダ要求リストとして整理し、調整したい要求に適用し要求の優先順位を決定することで、要求の競合を解決する。 このためには競合解決のためのトレードオフについて合意を形成しておくことが望ましい。 要求の競合が解決できない場合は、複数の要求として、分けて管理する。たとえば、要求から導き出される画面や帳票も複数になることを共通の認識とし、それに伴い開発工数も異なることをステークホルダで合意を形成しておくことが必要である。 課題の具体例として、セキュリティ関係の要求の場合、現場からの要求は使い勝手の面からセキュリティの制約が低く、例外の多いものになりやすく、システム部門からの要求は例外を認めないものになることが多いことが挙げられる。</p> <p>ステークホルダ要求リストの例</p> <p style="text-align: center;">表 1 ステークホルダリスト 1</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>ステークホルダ</th> <th>要求</th> <th>利害</th> <th>要求に与えるインパクト</th> <th>コスト</th> <th>開発期間</th> <th>緊急度</th> <th>優先度</th> </tr> </thead> <tbody> <tr> <td>ステークホルダ 1</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>ステークホルダ 2</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>ステークホルダ 3</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table> <p style="text-align: center;">* 実現可能性のないものは除いておく</p> <p style="text-align: center;">表 2 ステークホルダリスト 2</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>ステークホルダ</th> <th>要求</th> <th>効果</th> <th>実現可能性</th> <th>実効性</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table> <p style="text-align: center;">* 実効性 = 効果 * 実現可能性</p>			ステークホルダ	要求	利害	要求に与えるインパクト	コスト	開発期間	緊急度	優先度	ステークホルダ 1								ステークホルダ 2								ステークホルダ 3								ステークホルダ	要求	効果	実現可能性	実効性															
ステークホルダ	要求	利害	要求に与えるインパクト	コスト	開発期間	緊急度	優先度																																															
ステークホルダ 1																																																						
ステークホルダ 2																																																						
ステークホルダ 3																																																						
ステークホルダ	要求	効果	実現可能性	実効性																																																		

【事例】

ステークホルダ分析の例:MOYA のステークホルダ分析(リッチピクチャ)

ステークホルダが抱える課題をSSM(Soft Systems Methodology)のリッチピクチャで簡潔に図式化し、実務者と経営者、システムの利用者とその提供者など、ステークホルダそれぞれの立場の想いを分析する。リッチピクチャ作成時に考慮すべき点は次のとおりである。

- (1)問題領域には、どのようなステークホルダがいるのか
- (2)ステークホルダ間の関係は、どのようになっているのか
- (3)各ステークホルダは、どのような課題を抱えているのか
- (4)各ステークホルダ間に、意見の対立などはあるか

MOYA においては、ステークホルダ分析を単独に活用するのではなく、分析領域定義の XYZ 公式、課題分析の CATWOE(顧客(Customer)、行為者(Actor)、変換プロセス(Transformation Process)、世界観(World View)、所有者(Owner)、環境制約(Environment))を相互に活用することで本質的な課題に近付かせるようにしている。

* 企画プロセスで利用している。

【参考資料】

[1] 平岡 正寿, 「気づきから要求を導く」～要求定義方法論 MOYA～その概要と適用事例, 要求工学調査委員会, 要求開発ベストプラクティスが示す成功パターンの調査研究, 情報サービス産業協会, 18-J008, 2007 年 3 月.

備考: 関連ベストプラクティス, 効果, 制約:

【関連ベストプラクティス】

「分析・評価」ベストプラクティス

【制約】

優先度の低い要求を対象外として扱える

要求工学ベストプラクティス		No. A06
		090
問題分類: 要求定義	ベストプラクティス: キーマンを見つける	
1 版		
前提と課題(開発形態, ドメイン, 組織形態):		
[前提]		
発注側企業のプロジェクト推進体制		
<ul style="list-style-type: none"> (1) 複数の要求提供部門がプロジェクトに参画している。 (2) 各要求提供部門が要求に対する決定権を有しており、システム部門は要求部門間の調整は行うが、決定権は有していない。 (3) 受注側への要求の伝達はシステム部門より行われる。 		
[課題]		
<ul style="list-style-type: none"> (1) 複数の要求提供部門が存在する場合、一方の要求を満足しようとするれば他方が成り立たないケースが散見される。 (2) ファシリテータ (システム部門) は一方のシステム担当者のみしか参画していないことが多いことより、要求の不備が発見できないケースがある。 		
ベストプラクティス		
[対象プロセス]		
要求抽出		
[主導者]		
発注者推進者が主導するが、明確になっていない場合は受注者が発注者に要請する。		
[解決の観点]		
(1) プロジェクト運営		
1) プロジェクト推進体制を明確にする。		
複数の要求提供部門が存在する場合は、以下の事項を明確にしておく。		
a) プロジェクトオーナー		
システム化要件の優先順位付けを行なうプロジェクトオーナーを明確にしておき、要求が拡散した場合の最終決定を委ねる。		
b) ファシリテータ (システム部門)		
ファシリテータの役割と責任、権限を明確にしておくことで、無責任な発言の防止や要求提供部門間の混乱を抑止する。		
2) 要求提供部門とのコミュニケーション		
受注側への要求の伝達は、ファシリテータ同席のもと、要求提供部門と直接行なう。		
これにより、ファシリテータのスキル不足や誤解、誤認による要求の漏れ、矛盾等を防止することが可能となる。		
理想的なプロジェクト推進体制は以下の通りである。		
i) 提供された要求に対する調整及び確定は、ファシリテータ同席のもと、各要求提供部門と直接行なう。		
ii) 費用、工数、スケジュール等プロジェクトリソースの確定に関しては、ファシリテータ経由でプロジェクトオーナーと行なう。		
(2) ウォークスルーレビュー		
関連する業務、システム担当者にて、データの流れを中心としたウォークスルーを行い、担当業務だけではなく、周辺業務に対する要求に対する漏れや不整合の洗い出しを行う。		
ウォークスルーの観点は以下の通り。		

1) 想定している業務事象の確認

(業務事象例)

他社より受領するデータが期限までに到着しなかった場合

2) 代替案の確認

なお、ウォークスルーレビューにて確定した内容に対しては、「合意」したものと取り扱い、変更する場合のルールを明確にしておく。

備考: 関連ベストプラクティス, 効果, 制約:

[事例]

眞木 康裕, 要求定義における価値創生に関する考察, 要求工学調査委員会, 要求開発ベストプラクティスが示す成功パターンの調査研究, 情報サービス産業協会, 18-J008, 2007年3月.

要求工学ベストプラクティス		No. A07
		111
問題分類: 要求獲得	ベストプラクティス: 業界用語・知識の補完	
1 版		
<p>前提と課題(開発形態, ドメイン, 組織形態):</p> <p>[前提] ベンダ(SIer)側のメンバに顧客および顧客業界の用語知識が欠如している。</p> <p>[課題] 業界用語知識の欠如が要求の誤解、抜け、漏れにつながる</p>		
<p>ベストプラクティス</p> <p>[対象プロセス] 要求獲得</p> <p>[主導者] ベンダ</p> <p>[解決の観点] 予防、発見、解決</p> <p>[解決方法] 顧客の文化と自社の文化を橋渡しする用語集を作成する。 ここで言う「用語集」は、顧客を含むプロジェクト全体での正式用語集ではなく、ベンダ側のドラフト・備忘録としての用語集である。</p> <p>(1) 業界用語集と顧客用語集の 2 種類の用語集を作成する。 1) 用語は業界(業種)毎に独自なものがあり、また個々の会社ごとにも独自なものがあるので 2 種類の用語集を作成する。 2) 顧客が既に社内用語集を作成維持している場合もあるので、その場合はそれを活用する。</p> <p>(2) 業界用語集は事前に収集し作成しておく。 1) 業界用語集は当該業種のシステムを手がける場合のベンダとしての一般常識という認識で作成する。 2) 専門図書およびインターネット上の用語集、既存プロジェクトで作成した用語集などを参考に作成する。 インターネット上の用語集は誤りも多いので注意する。 インターネット上の用語集で信頼度の高いのは、業界団体が作成したもの、業界大手企業が作成したものなどである。 複数の情報源を元に作成すること。(内容の信頼度を高めるためと丸ごと引用すると著作権上問題になるため) 3) 顧客との打合せの中で業界用語の抜け・漏れがあれば追加していく。</p> <p>(3) 顧客用語集は実際の顧客ヒアリングの中で整備していく。 1) 顧客用語は厳密な定義がないものや、ヒアリング相手によって異なる定義をしている場合があるので注意する。 誰に確認したのか(どの部門に聞いたのか)も顧客用語集に記入しておくが良い。 2) 顧客の新入社員研修用テキストは非常に参考になる。わかりやすさと顧客社内でオーソライズされているという 2 点で特に有用である。</p>		

(4) 同じ言葉で業界用語・顧客用語とシステム用語、ベンダ用語が異なるものは特に注意する。顧客とベンダと双方で誤解する可能性が高い。

1) アルファベット 2 文字や 3 文字の略語は業界によって異なる意味を持つ場合が多いので注意する。

例: POP という用語

業界	フルスペル	意味
ネットワーク(システム開発系)	Post Office Protocol	電子メール受信プロトコル
ネットワーク(通信インフラ系)	Point Of Presence	集線局、アクセスポイント
小売業、広告業	Point Of Purchase	売場で商品を直接宣伝する広告物
製造業	Point Of Product	生産時点情報管理

2) システム開発のフェーズに関する用語は顧客とベンダ、複数ベンダ間で異なる場合が多いので注意する。

また、違う用語が似たような意味や同一の意味で使われる場合があるので注意する。

例: 単体試験の定義はモジュール単体試験かプログラム単体試験か。

外部設計と基本設計と概要設計と方式設計は同じ意味・内容なのか、細かい差異があるのか。

(5) 業界用語集と顧客用語集を分けて維持するとよい。

ベンダが顧客と同業他社の案件を手がけるときに業界用語集はそのまま再利用できる。

(6) 記載用語の範囲、記述のレベル

1) 業界用語は、自社(ベンダ)のメンバのレベル、知識範囲に応じて、用語を選択し、記述する。単に語数が多く詳しければ良いというものではない。

2) 顧客用語は、出現したもの全てを記載しておく。

備考: 関連ベストプラクティス, 効果, 制約:

[関連ベストプラクティス]

要求を洩れなく獲得(A01)

要求工学ベストプラクティス		No. A08
		006
問題分類: 業務知識	ベストプラクティス: ユーザの IT スキル向上	
1 版		
前提と課題(開発形態, ドメイン, 組織形態):		
[前提] 長期間にわたり複数のプロジェクトに携わるお客様		
[課題] IT 動向や他社事例を参考に、自社システムの実態とは乖離した要求となることがある。 (1)システム利用部門において、自社システムに対する IT 知識が不足している。 【例】 Web システムにおいて、ハードウェアの仕様を無視し、3 秒以内に 1 画面に複数の表やグラフを表示する要求 (2)業務フローに対する知識が不足。 【例】 月次処理でしか集計できない項目に対して、日次処理の要求。		
ベストプラクティス:		
[対象プロセス] 要求獲得		
[主導者] 発注者と受注者の共同作業		
[解決の観点] 予防		
[解決方法] (1)ファシリテータとしてシステム部門の有識者を介在させ、自社システムでの実現が不可能な要求に対しては、要求の変更をお願いする。 (2) ファシリテータの役割として、システム利用部門に対して、自社システムの実態に関する教育を以下のカリキュラムに基づき定期実施する。 1) 適用可能、対応可能な IT 技術の共有 自社で適用可能、対応可能な IT 技術を共有することで、要求の逸脱を防ぐ。 特に、先端技術を導入した際には、システム開発におけるリスクに加え、システム運用におけるリスクも共有しておくことが重要である。 2) 周辺業務に関する業務フローとシステムの関連 担当業務だけではなく、周辺業務に対する業務フローやシステム間連携についても、十分理解いただくよう努める。 【例】 理解しやすい例として、お金の流れを中心にウォークスルーすることがある。 3) IT 動向や他社事例の取得 自社の有識者や取引先から IT 動向や他社事例に対する情報を取得する。 自社の有識者や取引先より有益な情報が取得できない場合は、以下の方法を試みる。 a) 外部有識者(同業他社、外部委託先)に助言を求める。 b) RFI を提示し、要求実現方法の提供を求める。		
備考: 関連ベストプラクティス, 効果, 制約:		

問題分類: 目的	ベストプラクティス: 顧客要求に合った提案書の作成
----------	---------------------------

1 版	
-----	--

前提と課題(開発形態, ドメイン, 組織形態):

[前提]
顧客が要求定義時にITコーディネータもしくはコンサルタントなどを活用しようと考えている。

[課題]
経営改革も必要と考えているが、システム化にあたり、その必要性を当初はあまり認識していない。

ベストプラクティス

[対象プロセス]

要求獲得、要求分析

[主導者]

ベンダの IT コーディネータ(以下、ITC と略す)。
顧客にコンサルタントとしての関わりも持っており、役割としてはファシリテータにも近い。

[解決の観点]

ベンダへ発注する際、システム化の目的より手段が重視され、顧客の真の要求をうまく引き出せないままシステム開発に突入することで、できたシステムがその会社(組織)に役立たないシステムになってしまったり、システムが動いている実物を見てから開発の終盤で要求がでてきて、開発作業におけるオーバーヘッドが大きくなってしまい、ベンダ、顧客ともに疲弊してしまうケースがある。こういったことを予防できる。

[解決方法]

本プラクティスでは、提案書作成までのベンダ側の対応として、ITC プロセスを活用することにより、顧客の真の要求(経営改革を考慮したシステム提案がほしい)にあった提案書を作成する。

- (1) 事前の信頼関係構築の段階で ITC プロセスを顧客に紹介し、認識してもらうことで、その後の ITC プロセスを用いた提案をスムーズにできた。
- (2) ITC プロセスで、SWOT、5Force(図 1)、BSC(バランススコアカード)などを活用し、顧客の経営改革を念頭に置きつつ、現状分析、新システム構想を行い、提案書作成に活用。
 - 1) 外部環境分析
 - 2) 内部環境分析: SWOT 分析、戦略マップなど(図 2)
 - 3) システム化のポイント: 戦略マップから具体的な提案内容への説明

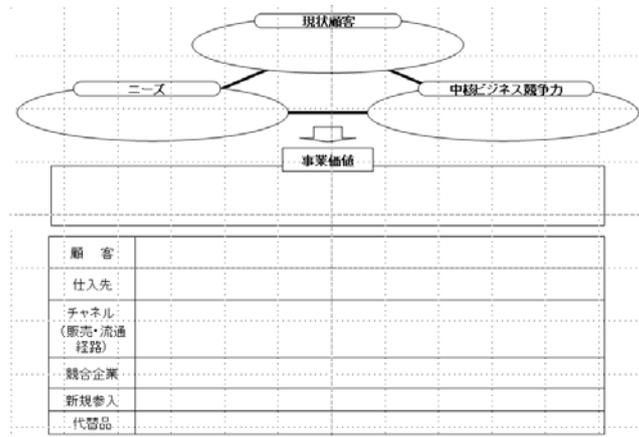


図 1 事業ドメイン分析と業界特性分析(5Force)

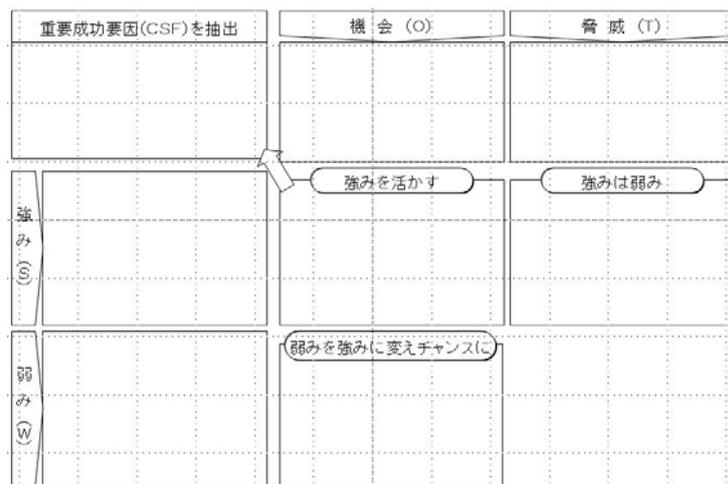


図 2 SWOT 分析・重要成功要因(CSF)抽出

- (3) システム化検討より前の経営戦略段階から参加し、関係者間での目的(目標)の共有
日ごろからの顧客との信頼関係の構築により、顧客の真のニーズの把握とそれに基づく提案の実施。

備考: 関連ベストプラクティス, 効果, 制約:

[事例]

要求定義における IT コーディネータの活用事例(本報告書所収)

[参考資料]

[1] IT コーディネータ協会, <http://www.itc.or.jp/>.

[2] ITコーディネータ協会(編), 成功したい社長が読むIT経営のススメ, アイテック, 2007.

要求工学ベストプラクティス		No. A11
		014
問題分類: 要求発散	ベストプラクティス: 画面周りの要求を確定する	
1 版		
前提と課題(開発形態, ドメイン, 組織形態):		
[前提]		
<p>(1) 顧客側の予算が限られている。(当然、スケジュールも厳しい)。従って、設計書を作成し、仕様を確定してから実装をすることは間に合わない状況にある。</p> <p>(2) 顧客が真に欲するのは、実際に動くシステムと操作マニュアルである。(各種設計書が欲しいわけではない)</p> <p>(3) アジャイル的な開発手法で要件を確認&確定していく必要がある。</p>		
[課題]		
<p>(1) 画面の細部に顧客の注意がいきまわり、本質的な要求でないことに発散する</p> <p>(2) 要件確認の初期段階において、本質的なデータ項目の確定に注力できない</p>		
ベストプラクティス		
[対象プロセス]		
要求の獲得、分析		
[主導者]		
ユーザ、ベンダ双方		
[解決の観点]		
下記「実際の進め方」のような段階を踏んで、要求を獲得&確定していくイテレーション計画をたて、特に、フェーズ 2 からは実システムイメージ(実際に実装を実施)して確認を行う。		
[解決方法]		
以下の 4 フェーズに工程を分けて実施し、フェーズ 3 以降は、ユースケース単位に実装～試験までを実施し、試験が完了したものから、順次、顧客にプレゼンテーションし、要件を確定する。		
<p>(1) フェーズ 1: 使用するアウトプット(画面・帳票)からデータ項目と業務要件をヒアリングする。 この段階でデータモデルを確定</p> <p>(2) フェーズ 2: 最終イメージそのもの(実物)の画面レイアウトと画面遷移を実装し、要件を確定する。 デザイン会社に発注した最終デザインを取り込んで、実際に実装したもので要件を確認</p> <p>(4) フェーズ 3: チェック仕様を実装し、チェックロジック及びメッセージ内容を確定した。(その段階での操作仕様も含む)(実際にコーディングと試験を実施)</p> <p>(5) フェーズ 4: ビジネスロジック及びデータ操作(検索&更新)を実装し、最終仕様を確認する。</p>		
[実践の留意点]		
<p>(1) 当初の要求獲得(データ項目の確定)においては、あくまでも画面/帳票の項目は入出力の確定であることを顧客に申し入れておいた。(それ以外の観点での要求抽出は実施しないということ)</p> <p>(2) データを如何に早く固められるかが、アジャイル的な開発手法を用いるときのポイント(データが確定しないとロジックの実装ができない)であり、早期にデータの確定を行っていく必要がある。</p>		
備考: 関連ベストプラクティス, 効果, 制約:		
[関連ベストプラクティス]		
<ul style="list-style-type: none"> ・要求が止まらない状況を防止する(A12) ・要求を適切なタイミングで確定(A13) 		

【効果】

実システムのイメージを早期に顧客に提示すること、暫定でも実システムを提供することで、要求の齟齬が発生しにくく、満足度が向上する。(顧客側も最終イメージをつかみながら要件を確認できる。)

【制約】

ユーザ側のステークホルダが XP でいう「オンサイト顧客」に近い対応できることが条件
また、中小規模(50ks 程度までの規模、開発体制は最大で 8 名程度)のプロジェクトでなければ難しい。(開発側のグループが分かれ、開発プロジェクト内に、何らかの統制が必要になる規模になると、ドキュメントや細かい進捗管理なしで開発を行うことは、難しいと思われる。)

要求工学ベストプラクティス		No. A12
		016
問題分類: 要求発散	ベストプラクティス: 要求が止まらない状況を防止	
1 版		
前提と課題(開発形態, ドメイン, 組織形態):		
[前提]		
<p>(1) 顧客側の予算が限られている。(当然、スケジュールも厳しい)。従って、設計書を作成し、仕様を確定してから実装をすることは間に合わない状況にある。</p> <p>(2) 顧客が真に欲するのは、実際に動くシステムと操作マニュアルである。(各種設計書が欲しいわけではない)</p> <p>(3) アジャイル的な開発手法で要件を確認&確定していく必要がある。</p>		
[課題]		
<p>(1) 要件定義書を作成せずに顧客からの要件獲得をどのように実施するか</p> <p>(2) 要件定義書を作成せずに、獲得した要件をどうやって確定させていくか</p>		
ベストプラクティス		
[対象プロセス]		
要求の獲得、検査、管理		
[主導者]		
ベンダ		
[解決の観点]		
下記「実際の進め方」のような段階を踏んで、要求を獲得&確定していくイテレーション計画をたて、特に、フェーズ 2 からは実システムイメージ(実際に実装を実施)して確認を行う。		
[解決方法]		
以下の 4 フェーズに工程を分けて実施し、フェーズ 3 以降は、ユースケース単位に実装～試験までを実施し、試験が完了したものから、順次、顧客にプレゼンテーションし、要件を確定する。		
<p>(1) フェーズ 1:使用するアウトプット(画面・帳票)からデータ項目と業務要件をヒアリングする。 この段階でデータモデルを確定</p> <p>(2) フェーズ 2:最終イメージそのもの(実物)の画面レイアウトと画面遷移を実装し、要件を確定する。 デザイン会社に発注した最終デザインを取り込んで、実際に実装したもので要件を確認</p> <p>(3) フェーズ 3:チェック仕様を実装し、チェックロジック及びメッセージ内容を確定した。(その段階での操作仕様も含む)※実際にコーディング&試験を実施</p> <p>(4) フェーズ 4:ビジネスロジック及びデータ操作(検索&更新)を実装し、最終仕様を確認する。</p>		
[実施上の留意点]		
<p>(1) 契約時に、暫定納品後の顧客確認期間を設け、それ以降の変更要求が発生した場合は、追加費用が発生することを事前に調整しておくこと。</p> <p>(2) 難しい要件変更要望が発生した場合は、制約(スケジュール等)から見て難しいことを開発側から、申し入れ、調整を実施することも必要。</p>		
備考: 関連ベストプラクティス, 効果, 制約:		
[関連ベストプラクティス]		
<p>(1) 画面周りの要求を確定させる(A11)</p> <p>(2) 要求を適切なタイミングで確定(A13)</p>		

【効果】

実システムのイメージを早期に顧客に提示することと、暫定でも実システムを提供することで、要求の齟齬が発生しにくく、満足度が向上する。(顧客側も最終イメージをつかみながら要件を確認できる。)

【制約】

ユーザ側のステークホルダが XP でいう「オンサイト顧客」に近い対応をしてくれることが条件
また、中小規模(50ks 程度までの規模、開発体制は最大で 8 名程度)のプロジェクトでなければ難しい。(開発側のグループが分かれ、開発プロジェクト内に、何らかの統制が必要になる規模になると、ドキュメントや細かい進捗管理なしで開発を行うことは、難しいと思われる。)

要求工学ベストプラクティス		No. A13
		096
問題分類: 要求発散	ベストプラクティス: 要求を適切なタイミングで確定	
1 版		
前提と課題(開発形態, ドメイン, 組織形態):		
[前提]		
<p>(1) 顧客側の予算が限られている。(当然、スケジュールも厳しい)。従って、設計書を作成し、仕様を確定してから実装をしては間に合わない状況にある。</p> <p>(2) 顧客が欲するのは、実際に動くシステムと操作マニュアルである。(各種設計書が欲しいわけではない)</p> <p>(3) アジャイル的な開発手法で要件を確認&確定していく必要がある。</p>		
[課題]		
要件定義書を作成せずに顧客への要件を確定するタイミングをどうやって設けるか		
ベストプラクティス		
[対象プロセス]		
要求獲得、検査、管理		
[主導者]		
ユーザ、ベンダ双方		
[解決の観点]		
下記の解決方法の段階を踏んで、要求を獲得&確定してクイテレーション計画をたて、特に、フェーズ2からは実システムイメージ(実際に実装を実施)して確認を行う。		
[解決方法]		
以下の4フェーズに工程を分けて実施し、フェーズ3以降は、ユースケース単位に実装～試験までを実施し、試験が完了したもものから、順次、顧客にプレゼンテーションし、要件を確定する。		
<p>(1) フェーズ1:使用するアウトプット(画面・帳票)からデータ項目と業務要件をヒアリングする。 ※この段階でデータモデルを確定</p> <p>(2) フェーズ2:最終イメージそのもの(実物)の画面レイアウトと画面遷移を実装し、要件を確定する。 ※デザイン会社に発注した最終デザインを取り込んで、実際に実装したもので要件を確認</p> <p>(3) フェーズ3:チェック仕様を実装し、チェックロジック及びメッセージ内容を確定した。(その段階での操作仕様も含む)(実際にコーディングと試験を実施)</p> <p>(4) フェーズ4:ビジネスロジック及びデータ操作(検索&更新)を実装し、最終仕様を確認する。</p>		
上記の各イテレーション単位で暫定納品を行い、2週間程度の顧客確認期間を設け、それ以降の変更要求が発生した場合は、追加費用が発生することを事前に調整しておく。(確認期間が仕様確定タイミング)		
備考: 関連ベストプラクティス, 効果, 制約:		
[関連ベストプラクティス]		
<p>(1) 画面周りなどを実際に提示することで要件確定につなげる(A11)</p> <p>(2) 要求が止まらない状況を防止する(A12)</p> <p>(3) 後工程で要求が出てくる(完成した画面を見てから等)ことを防止する(A15)</p>		
[効果]		
実システムのイメージを早期に顧客に提示することと、暫定でも実システムを提供することで、要求の齟齬が発生しにくく、満足度が向上する。(顧客側も最終イメージをつかみながら要件を確認できる。)		

【制約】

- (1) ユーザ側のステークホルダーが XP でいう「オンサイト顧客」に近い対応をしてくれること
- (2) 中小規模(50ks 程度までの規模、開発体制は最大で 8 名程度)のプロジェクトでなければ難しい。(開発側のグループが分かれ、開発プロジェクト内に、何らかの統制が必要になる規模になると、ドキュメントや細かい進捗管理なしで開発を行うことは、難しいと思われる。)

要求工学ベストプラクティス		No. A15
		103
問題分類: 要求発散	ベストプラクティス: 計画段階での要求発散や確定遅れの防止	
1 版		
前提と課題(開発形態, ドメイン, 組織形態):		
[前提]		
ユーザ部門にシステム開発の経験が少ない場合、複数のシステム担当 SIer 体制である場合など関係部門が多い場合、要件・仕様変更、追加やスケジュール変更が発生することが多い。		
[課題]		
要件や仕様が、プロジェクト遂行中に大幅に追加または変更される場合が多々発生する。開発工程が進んでからその追加や変更を調整しようとしてもお互いの合意や妥協点が見出し難い。従って、当該プロジェクトに関して、開発の初期段階からユーザ、ベンダ双方で拠りどころとなる何らかの合意文書を持ち、それに沿ってプロジェクトを進めることが望ましい。		
ベストプラクティス		
[対象プロセス]		
要求獲得時(下記対応策(1)、(2)、(3))、要求管理時((4)、(5))		
[主導者]		
ベンダ(問題関与者はユーザ、ベンダ双方であるがユーザ側にシステム開発経験が少ないため。)		
[解決の観点]		
予防: 要求の発散及び要件の確定遅れの予防		
[解決方法]		
「システム開発計画書」の運用による客先との役割分担、開発プロセスの合意形成・維持		
(1) ユーザ部門への役割、SI・開発プロセスの説明、教育		
1) ユーザ側のシステム開発経験が少ない体制である場合、複数のシステム担当の SIer 体制である場合、客先予算と要求機能にギャップがある場合などのケースにおいては、要件・機能の調整・決定がプロジェクト成功へのポイントとなることが多い。		
2) 施策として、ユーザ側にシステム構築の全体責任があることを認識させることと、早期に要件・仕様を確定させて SI・開発プロセスのルールを遵守してプロジェクトを進めてもらうことを目的に、ユーザ部門の役割(計画レビュー、仕様レビュー、データ移行作業など)及び、開発プロセス: 開発工程、品質管理、仕様変更ルールなどの説明および教育を行なうこととする。		
(2) 開発プロセスの明確化・合意形成を目的に、提案書を入力に「システム開発計画書」を作成し、ユーザ承認を得て合意		
1) 「システム開発計画書」は契約書を補完する役割を持つ。「システム開発計画書」のインプットは「提案書」であり、客先との合意を得るため、前提となる提案書からの内容一貫性を保つ。		
2) 「システム開発計画書」の内容は、「システム開発の目的、システム化範囲、納入要件、推進体制、役割分担、開発環境、大日程計画・品質保証計画、プロジェクト運営、仕様変更手順等」を記述する。特に、「お客様との作業分担を明確にすること」、「要件・仕様変更に対する扱い(文書化、別途費用等)を明確にすること」にポイントを置いている。		
3) システム開発計画書を運用することで、次のような効果を図っている (NEC ソフト社全体方針より)。		
a) システム構築の範囲、役割分担、作業条件を明確にすることにより、問題発生時の早期解決が図れ、プロジェクトの損益予測の確度向上にもつながられる。		
b) ユーザと当社との間で開発計画の合意、確認を行うことにより、責任範囲の明確化、双方で計画の		

共有化、進捗管理方法、要件・仕様変更管理方法の明確化が図れる。
c)作業範囲を確認することにより、業務単位の要員計画の明確化が図れる。
上記 a)～c)により、プロジェクト遂行中でのユーザとの誤解等のトラブルを防ぎ、プロジェクトを成功裡に進めることが可能となり、顧客満足度の向上も期待できる。

(3) プロジェクトのステークホルダ全員に対する開発計画説明会の実施

- 1) プロジェクト発足時、極力、早い段階で開発計画書の作成と説明の場設定を行なう。
- 2) マルチベンダ体制の場合、ユーザのみならず、スポンサ、既存システム、関連システム対応SIerなどの他のベンダ、協力会社等会社間にまたがるステークホルダなどが多数存在する。システム構築に当たっての目的や基本的な合意事項を共有し、周知徹底させる場として説明会を実施することで、以降のプロジェクト運営を円滑進められるようにする。

(4) プロジェクト憲章として基本合意事項を活用

- 1) システム開発作業における基本合意事項を記した「システム開発計画書」をプロジェクト憲章として活用。プロジェクトの途中で基本合意事項を立ち返って確認することができるため、要件・仕様変更発生時や体制変更発生などの場合に顧客や関係者との課題調整時に、役立てることができる。

(5) フェーズ毎で活動計画書・WBSを作成し、そのタイミングで提示することで基本事項を顧客と再確認

- 1) システム開発計画書をベースとして、役割分担、大日程計画等を前提条件として参照しながら、フェーズ毎に現実的に詳細化した活動計画書と中日程WBSをとりまとめ、その都度、ユーザに提示する。工程が進むにつれて、開発項目が、要件⇒機能⇒モジュール⇒コード化、とブレイクダウンしてゆく開発プロセスをユーザに認識してもらうことで、要件・仕様の発散や無理な追加・変更を防止している。
- 2) システム開発計画書作成時点では詳細化できていない項目について、または明確になっていなかった事項を具体的にフェーズ毎の活動計画書で補完していく。

個別プロジェクトでのフェーズ毎の活動計画書の例:

- －システム分析フェーズ活動計画書&中日程
- －システム構築フェーズ活動計画書&中日程
- －結合テストフェーズ活動計画書&中日程
- －総合テストフェーズ活動計画書&中日程

フェーズ毎のタイミングでユーザ、ベンダ側との作業の明確化、基本事項の確認を行なうことにより、トラブルを未然に防ぐことができる。

備考: 関連ベストプラクティス, 効果, 制約:

[関連ベストプラクティス]

チーム制による要求矛盾の回避(N11)

問題分類: 検証・レビュー	ベストプラクティス: 非機能要求の形式知化
---------------	-----------------------

1 版

前提と課題(開発形態, ドメイン, 組織形態):

[前提]

(1) ベンダ側担当者はシステム部門から要求仕様書(または要求に関するメモ)の説明を受けた。その後、経験者によるレビューを受けずに要件定義書を作成し、確認のためシステム部門に送付した。

(2) システム部門の承認を得たのでベンダ側は開発に着手し、統合テストまでプロジェクトが進行した。

[課題]

ベンダ側担当者は要件定義段階や設計段階で熟練者によるレビューを計画したが熟練者の業務多忙でかなわず、そのまま開発を続行した。ところが統合テストの段階になってシステム部門に説明を行ったところ、同席したユーザから仕様にはない非機能要件を実現できていないことを指摘され、大きな手戻りが発生した。

ベストプラクティス

[対象プロセス]
要求仕様記述

[主導者]
ベンダ: 非機能要件に対するベンダ側の問題なので、主導者はベンダである。

[解決の観点]
解決: 業界知識習得に努めるとともに、非機能要件の洗出しのためにさまざまな手法を採用する。

[解決方法]

開発の終盤になって初めてユーザやシステム部門の想定する要件と、出来上がったシステムとの違いが明らかになる場合、非機能要件に起因することが多々ある。いわゆる行間に書かれていた大事なことを見逃していたのであるが、それを防ぐためには熟練者による詳細なレビューが必要である。しかし、本課題のように熟練者が業務多忙でそれがかなわない場合もある。そのような事態を避けるために以下のようなさまざまな手法を採用する。

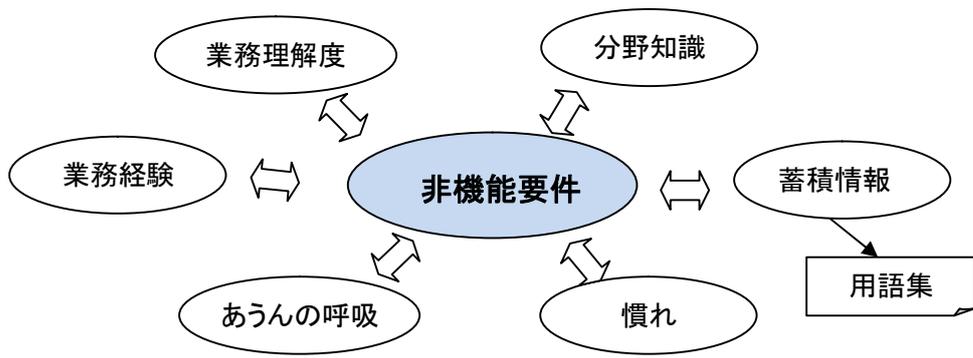


図1 非機能要件に関連する要因

- (1)組織的対応:レビュー専任者の任命
類似プロジェクトが多い場合、経験豊富かつ教育的素養を持つ人を特定プロジェクトに属さないレビュー専任者として任命し、各プロジェクトで必要な場合にレビューを担当できるようにする。
- (2)知識・経験の継承:情報の蓄積や継承の機会の設定
経験者の手により用語集や技術資料を蓄積し、経験で得られた知識を情報蓄積や勉強会の機会を通じて

広めていく。しかしながら、文章化が困難な感覚に基づいたものや独自文化も多いので、教育的観点に立ったレビューを通じて知識の継承を図ることが効果的である。

(3) 密接なコミュニケーション:信頼の構築

非機能要件に対する課題の本質は単に業務知識やコミュニケーションだけの問題ではなく、同じ業界に生きるものとしてお互いが理解を深め合い信頼を築くことができるかどうかということもある。当然、業界の経験や業務理解の差が大きく影響するし、あるいは、顧客とベンダ間の永年の付き合い(あうんの呼吸とも言うべきか)や慣れで構築される面も大きい。それらも含めた信頼関係の構築を行う。

(4) 直接的なコミュニケーション:出向や駐在

ベンダ側にとって付合うのが初めての業界や企業、または初めて取扱う業務システムの場合、これまでとはまったくの別世界に踏み出したとベンダは覚悟してかかる必要がある。それを克服する手段として、例えばユーザの仕様確定部門(システム部門)へ数週間から数ヶ月の単位で出向または駐在し、顧客側担当者とともに要件を確定することで非機能要件の形式知化を行う。場合によっては要求の再検討にまで関与する。

この副次的効果として、協働で作成した仕様に基づいて開発見積りを実施することで、機能要件の漏れはおろか仕様的大幅な増加や変更も防ぐことができる。この場合、顧客への出向(駐在)段階と開発で契約を変えることが必要である。出向(駐在)期間は期間単位での契約とし、開発は通常通りに一括で契約する。

(5) 間接的な確認:ユーザ内部資料の入手

ユーザがまとめた資料はそのものがユーザの文化であり、ユーザの非機能要件を最も的確に表したものである。そこで、ユーザの業務マニュアルや要領書などの内部資料を入手し確認する。

非機能要件が課題になることのもう一つの側面は、どこまで相手が知っているかを確認しないことであり、お互いが相手は知っていて当然という意識しかないところである。その「無意識」がシステム開発へ不確定さを導入する。業務の都合上、未経験者を割当てせざるを得ないなどのさまざまな理由をベンダ側は抱えているので、打合せの場に出席した担当者すべてが分野知識をもつものではないということをユーザまたはシステム部門は知っておくべきである。一方、ユーザまたはシステム部門がすべての情報をベンダに伝えている訳ではないとベンダは知っておくべきである。

しかしながら、非機能要件の難しさは、完成したシステムがユーザにとって十分満足ゆくものであれば問題なく、一方、何かしらの不満足があれば突如として大問題になるという側面を持っているところにもある。したがって、非機能要件といえども気づいたときは文書化の努力を惜しんではいけない。それが信頼獲得の第一歩である。

備考: 関連ベストプラクティス, 効果, 制約:

[関連ベストプラクティス]

適切なコミュニケーションによる要求の確認(V01)

要求工学ベストプラクティス		No. D03
		034
問題分類: 文書化	ベストプラクティス: 簡易要求分析フレームワーク	
1 版		
<p>前提と課題(開発形態, ドメイン, 組織形態):</p> <p>[前提] Web をベースとしデータベースを操作する一般的なシステムに適用を行なう。</p> <p>[課題] 要件を獲得した段階で、ベンダは顧客に対して開発規模を示す必要があるが、ベンダ側にとっては開発規模の根拠をこの段階においては示しづらく、また顧客側にとってもその良否を判断するための手段がない。そのため、この時点での開発規模の見積もりについては顧客、ベンダ双方において同意しがたいものがある。</p>		
<p>ベストプラクティス</p> <p>[対象プロセス] 要求獲得</p> <p>[主導者] ベンダが分析を行なう。ただし、方法自体は簡易であるためユーザも理解可能である。(ユーザ側において分析を行なうことも可能。) 分析後はユーザとベンダ双方によって、開発規模の検討を行なう。</p> <p>[解決の観点] 予防、発見、解決</p> <p>[目的] 簡易化された機能分析方法を利用することにより、顧客の要望に対して要求定義の段階において機能分析を行うことによって、顧客とベンダ双方が開発規模の判断を可能とする。</p> <p>[解決方法] 機能分析を行なうための方法として「PSER」の提供を行なう。 (1) 要件から PSER への抽出 顧客から提示された要件から PSER の分析を行なう。顧客から示される要件は顧客毎に示し方が変わってくるので、顧客毎に分析ルール(顧客要件→PSER へのマッピング規則)を定めておき分析を行なう。 (2) PSER の規模によって開発規模を検討 PSER 分析した結果、PSER のそれぞれの個数によって開発規模を推定する。</p> <p>[PSER の概要] (1) PSER の概要 PSER は図 5 に示すように一つの業務を 4 つの観点に分類して分析を行なう。 (2) PSER の記述方法 PSER については図 5 で示された観点を図 6 のように組み合わせて分析を行なう。例として業務プロセスはシステムのメニュー項目の一つのレベルとし、そのメニュー項目を選択した際に行なう個々の作業(例:「商品一覧を表示」「一覧の中からショッピングカートへ選択」)を業務サービスとする。また、業務サービスがデータベースにアクセスする場合はエンティティを利用し、業務ルールが存在する場合は業務ルールを各観点到に差込をする。 また、図 7 に旅費清算システムにおける分析事例を示す。例に記述しているのは、出張における清算費を申請した後に、その内容を変更する部分を主に PSER で分析を行っている。</p>		

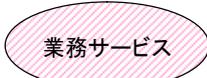
- P**  - 一つの業務機能にまつわる一連の処理の流れや状態遷移を制御するコンポーネント
- 例: Webショッピング=商品選択→注文確認→注文確定
-
- S**  - 業務プロセスが業務処理を実行する上で必要となるロジックを提供するコンポーネント
- 例: 商品検索、ショッピングカート商品入力、注文発行、など
-
- E**  - データベースに保存される業務データ
- 例: 注文明細、配送伝票、など
-
- R**  **業務ルール** - チェックロジックや計算式を提供するコンポーネント
- 例: 注文可否チェック=[カートが空でない]AND[合計金額<100万円]AND[商品在庫有り]
- その他、非機能要件事項を記述

図 5 PSER の 4 要素の位置付け

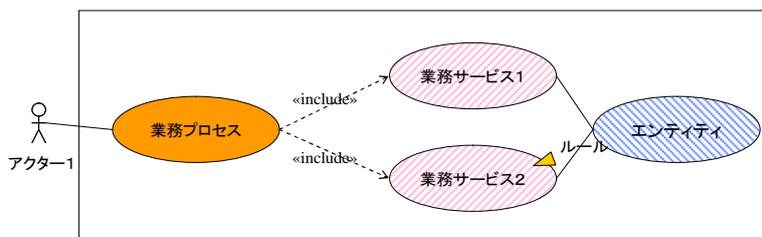


図 6 PSER の組み合わせ

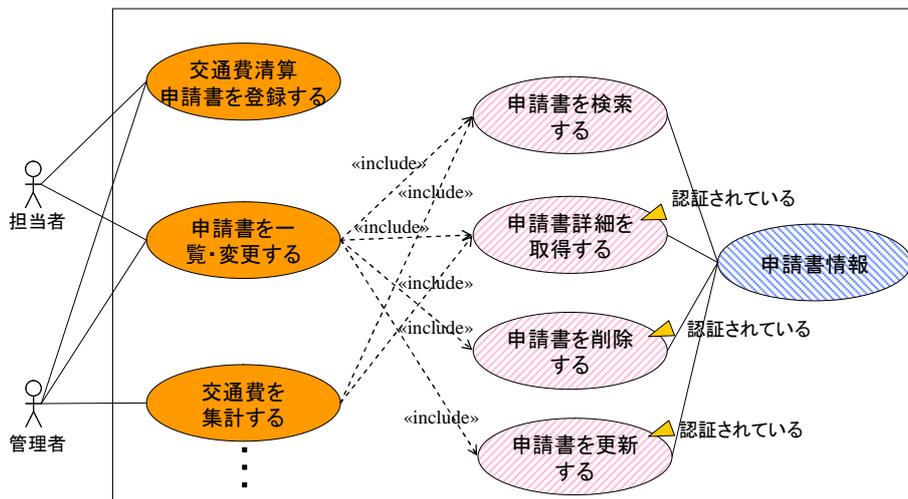


図 7 PSER での分析事例

備考: 関連ベストプラクティス, 効果, 制約:

[効果]

(1)分析時に抽出した機能が実装されているかのトレーサビリティ

抽出された PSER の構造は、設計、実装時においてもそのまま活用できる。よって、要求分析の段階で抽出された機能を実現しているのかについて確認が容易になる。

問題分類: 分析・評価	ベストプラクティス: 要求の優先順位をビジネス全体で捉える
-------------	-------------------------------

1 版

前提と課題(開発形態, ドメイン, 組織形態):

[前提]

- (1) 新システム導入に向けて、ユーザはシステム化への要求をベンダに伝え、ベンダはシステム提案を行っているような場合で、システム構築の契約は締結していない状況。
- (2) ユーザ側が To-Be モデルを明確に描いていない、かつ、技術的制約に考慮していない。

[課題]

- (1) ユーザのビジネス要求の優先順位と要求に対するシステム化の優先順位が異なる。
- (3) ユーザ間で部門単位のビジネス要求の優先順位が異なり、ベンダが優先順位を整理できない。

ベストプラクティス

[対象プロセス]

要求分析

[主導者]

ベンダ

[解決の観点]

予防: 事業に対する課題を整理し、それに対する IT 化の課題を明確にし、整合性のとれたシステム化の実現を図る。

[解決方法]

ベンダは、ユーザに対して事業課題の解決やそれに伴う IT 化の課題を解決するアプローチ方法を支援し、要求を確定させる。

例えば、全体最適化(EA)の手法を活用して、課題の整理と解決を図る。

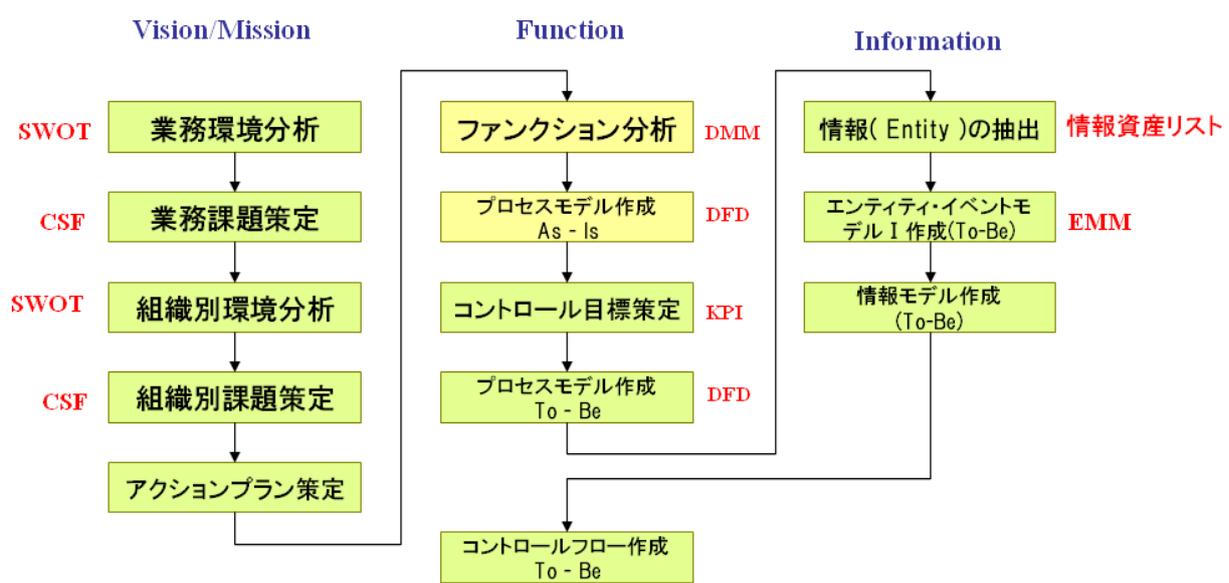


図 1 EA 作成ステップ

要求の優先順位付けについては他のプラクティスを参照。

備考: 関連ベストプラクティス, 効果, 制約:

[関連ベストプラクティス]

重要度・緊急度マトリックスによる要求の選別(N02)

[参考資料]

経済産業省 EA ポータル http://www.meti.go.jp/policy/it_policy/ea/index.html

問題分類: 分析・評価	ベストプラクティス: 重要度・緊急度マトリクスによる要求の選別
-------------	---------------------------------

1 版

前提と課題(開発形態, ドメイン, 組織形態):

[前提]

顧客との間で同一の手法による、要求の優先の順位付けを行うことが同意されていること。

[課題]

優先順位付けの手法とルールを具体的に定めて、維持管理すること。

ベストプラクティス

[対象プロセス]

要求分析

[主導者]

ベンダ

[解決の観点]

予防: 事業に対する課題を整理し、それに対する IT 化の課題を明確にし、整合性のとれたシステム化の実現を図る。

[解決方法]

顧客からの要求について、重要度、緊急度で判断し、下図に示すマトリクス[1]にマップする。重要度は顧客の業務上での重要度を 3 段階に分け、S は最重要、A は重要、B は重要でない、に分ける。緊急度は、現在開発中のシステムに反映すべき内容であるかどうかで判断し、小(緊急ではない)、中(緊急だが代替え手段あり)、大(緊急かつ代替え手段なし)に分類する。

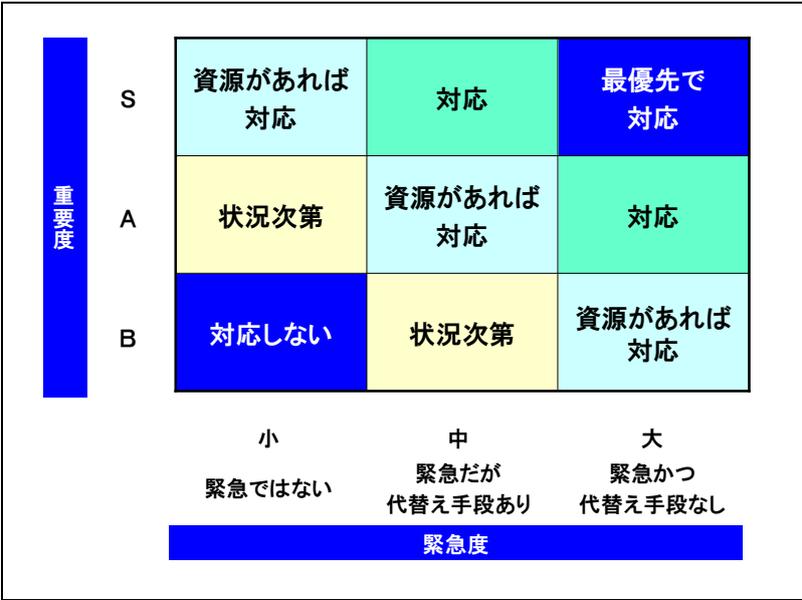


図 1 重要度・緊急度マトリクス

その結果、対応方法は次の 5 つに分けられる。

- (1) 最優先で対応: 他の要求の優先度を下げても採用する要求。
- (2) 対応: 現在のシステムに採用する要求だが、最優先ほど高くない要求。

- (3) 資源があれば対応:開発コスト、スケジュール、リソースに余裕があれば採用を検討する要求。
- (4) 状況次第:開発プロジェクトの状況に応じて採用するかどうか判断する要求。
- (5) 対応しない:現在は採用すべきではない要求。

マトリックスにマップした要求については、顧客内での見解が統一されるまで議論し、その結果ユーザとベンダとの調整に使用する。

備考: 関連ベストプラクティス, 効果, 制約:

[関連ベストプラクティス]

優先順位付けに基づいたレビューによる実現する機能の絞り込み(N08)

[効果]

システム化範囲が明確になり、顧客満足度の高いシステム構築に貢献できる。

[制約]

重要度の決定ルールが顧客と開発間で合意されていること。

システムライフサイクルを通して、マトリックスの保守を継続的に行うこと。

[参考資料]

[1] 井上 潤吾, IT投資の見える化, ビジネスインパクト, Vol. 14, 2007, pp. 32-33,

<http://www.bcg.co.jp/article/index.php?y=2007&page=2>.

要求工学ベストプラクティス		No. N03																				
		049																				
問題分類: 分析・評価	ベストプラクティス: 要求の優先順位設定																					
1 版																						
<p>前提と課題(開発形態, ドメイン, 組織形態):</p> <p>[前提]</p> <p>(1) ユーザとベンダとで要求項目についてシステム化に向けて検討を行っており、契約段階まではいたらない状態である。</p> <p>(2) ユーザはベンダに対して、要求項目における優先順位を自ら明確に提示していない。</p> <p>(3) または、ベンダが、要求項目における優先順位を把握していない。</p> <p>[課題]</p> <p>(1) ユーザ側の要求をベンダ側の都合による解釈により、要求の優先順位を決める。</p> <p>(2) ユーザの要求に関わる費用とビジネス効果による費用対効果とシステム化に関わるベンダ側の費用および提供品質を考慮されていない。</p> <p>(3) このことにより、システム構築中以降に仕様の見直しが発生や、運用開始時にユーザ要求とは異なる仕様であることが発覚する。</p>																						
<p>ベストプラクティス</p> <p>[対象プロセス] 要求分析、要求管理</p> <p>[主導者] ベンダ</p> <p>[解決の観点] 予防: 要件の合理性や事業課題に対する整合性を見極め、要件に対する品質を確保できるよう事前に対応を図る。</p> <p>[解決方法] ベンダは、要求に関わる費用と価値を分析し、その結果を管理する。</p> <p>(1) ベンダ側は、その要求により、ユーザ側がどの程度のビジネス効果、価値を想定しているかの情報を得て、システム化に対して想定される費用と要求される品質について分析し、要求が達成された効果とシステム化に関わる影響度を明確にする。</p> <p style="text-align: center;">表 1 優先順位付けの例</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>優先順位</th> <th>要求項目</th> <th>ニーズ</th> <th>経済性</th> <th>実現性</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table> <p>(2) 解決策の実現性と課題の重要度で重み付けをし、取り組むべき課題を選定する。</p> <p>(3) それをもとに、要求の優先順位について、ユーザと合意形成をはかる。</p> <p>(4) 利害関係者への承認を得る。</p> <p>(5) 合意形成後は、ベンダは、要求に対する品質を確保する方法を選択し、品質を管理する。</p>			優先順位	要求項目	ニーズ	経済性	実現性															
優先順位	要求項目	ニーズ	経済性	実現性																		
<p>備考: 関連ベストプラクティス, 効果, 制約:</p> <p>[関連ベストプラクティス] 要求の優先順位をビジネス全体で捉える(N01)</p>																						

[効果、制約、関連等]

業務スキル・業務ノウハウの習熟度

ROI 分析

要求工学ベストプラクティス		No. N04
		057

問題分類: 分析・評価	ベストプラクティス: To Be の効果を定量的に表現できる仕組み
-------------	-----------------------------------

1 版

前提と課題(開発形態, ドメイン, 組織形態):
[前提]
 ユーザ側において、以下のものが求められている。
 (1) 利害関係者からシステム化に伴う費用対効果が求められている。
 (2) 経営者等から数値指標が求められている。

[課題]
 ベンダ側として、
 (1) 要求そのものが、例えば、費用にあった効果があるのか、要求を実現したシステムが本当に要求に対して効果があるのかを捕捉する。
 (2) 要求に対する効果を捕捉する。
 (3) MI(6σ 改善活動)のように To Be の効果や KPI を定量的に表現できる仕組みが必要。
 つまり、システム上に捕捉できる仕組みとデータベース上に確保できる項目が必要。

ベストプラクティス

[対象プロセス]

要求仕様記述

[主導者]

ベンダ

[解決の観点]

解決: ビジネスプロセスの ROI をモニタリング/コントロールできるシステム構築を要求の時点で埋め込む。

[解決方法]

要求に対する効果を捕捉できる情報を決定し、要件に定義する。
 (1) ユーザにおけるビジネスプロセスを含めた範囲で要求の実現効果を考慮し、捕捉のタイミング(いつ、だが、どこで)と捕捉するもの(何を、どのように)を明確にし、それを記述することが必要。
 例えば、バランス・スコアカードや IT コーディネータのコントロール目標の活用。
 (2) 上記をユーザに依頼する場合、ユーザ自身だけで難しい場合は、ファシリテートを行う。
 (3) ファシリテート上、留意することは、システムの視点に入り込まず、ビジネスプロセスの視点に立って、ユーザを支援する。捕捉する項目については、ベンダ側で検討し、実現可能なものにしておくこと。
 (4) 捕捉しただけでは、データにすぎないので、評価できる方法を事前に検討し、システム化の必要がある場合は、要求項目として記述しておくこと。

表 1 バランススコアカードシート

視点名	戦略目標	重要成功要因	業績評価指標	ターゲット(数値目標)	アクションプラン
財務					
顧客					
業務プロセス					
人材と変革					

表2 ITコーディネータ コントロール目標シート

QCT	サービス目標	目標値	何が	どの様な状態

備考: 関連ベストプラクティス, 効果, 制約:

[関連ベストプラクティス]

[参考資料]

[1] 吉川 武男, バランス・スコアカード構築、生産性出版, 2003.

[2] 前川 寿住(編), IT コーディネータの中小企業経営改善実践マニュアル, 清文社, 2002.

要求工学ベストプラクティス		No. N05
		065
問題分類: 分析・評価	ベストプラクティス: 課題の検討不足を克服	
1 版		
前提と課題(開発形態, ドメイン, 組織形態):		
[前提]		
(1) ユーザ側が要件をうまく説明しきれていない、もしくは、要件を把握していない場合		
(2) ベンダ側が要件を安易に考えてしまったか、もしくは、要件を把握していない場合		
[課題]		
(1) 課題に対する原因を十分に検討せずに、要求をシステム化している場合		
(2) 課題に対する原因を検討する方法がわからない場合。 これにより、課題に対する間違った解決策を選択する。		
ベストプラクティス		
[要求ライフサイクル]		
要求分析		
[主導者]		
ベンダ		
[解決の観点]		
予防: ユーザの課題に対するシステム化の整合性を保つために、原因を把握できる環境や手段を考える。		
[解決方法]		
課題に対する原因を検討する方法を選択し、それにもとづいて原因を把握する。		
ユーザ側の参画、例えば会議への参加やインタビューへの参加などの協力を依頼する。		
(1) 課題を分析するにあたり、メンバが取り組みやすい原因を検討する方法を選択する。 例えば、KJ 法、QC の手法、「なぜ、なぜ、5 回」など。		
(2) 課題に対してメンバ全員が理解し、選定した方法で原因を検討し、結果に対して合意形成を図る。		
(3) ワークショップ形式の会議や図式化などで課題に対する原因の検討が行い易い環境を作る。		
(4) ベンダは、課題に対する原因を検討するための資料の図式化やインタビューなどにより、原因を導出する。		
(5) このとき得られた原因は、ユーザと共有すること。		
備考: 関連ベストプラクティス, 効果, 制約:		
[関連ベストプラクティス]		
[効果、制約、関連等]		
業務スキル、インタビュースキル、プレゼンテーションスキル。		

要求工学ベストプラクティス		No. N07
		086
問題分類: 分析・評価	ベストプラクティス: 要求の優先順位の管理	
1 版		
前提と課題(開発形態, ドメイン, 組織形態):		
[前提]		
<p>(1) 要件に対する優先順位付けを最初から行わずに進められた場合。</p> <p>(2) 打合せを重ねる上で、要件の優先順位が変わったのを変更していなかった場合。</p> <p>(3) 主となる要件の優先順位が高く、その要件に関連する要件の優先順位を変更していなかった場合。</p>		
[課題]		
ユーザの要求の優先が高いものがシステム化されていない、もしくは、リリース時期が遅くなり、ユーザ要求を反映したものとならない。		
ベストプラクティス		
[対象プロセス]		
要求獲得、要求分析、要求管理		
[主導者]		
ベンダ		
[解決の観点]		
予防: 要件の優先順位を付けて管理することが、システム化に際し要件単位での追跡可能性を高める。		
[解決方法]		
<p>(1) 要件リストと優先順位を常にメンテナンスする。</p> <p style="padding-left: 20px;">このとき、優先順位付けを複数の方法で提示し、合意の得られる方法を採用する。</p> <p style="padding-left: 20px;">例えば、要件リストは、要求属性項目表などで管理し、要件の内容については、ユースケース記述などで管理する。</p> <p style="padding-left: 20px;">要求属性項目表は、要求項目、優先度(相対的価値)、要求の出处、根拠など</p>		
<p>(2) 優先順位付けについては、他のプラクティス参照。</p>		
<p>(3) 要求管理における変更管理や構成管理については、ツールを利用して管理する。</p>		
備考: 関連ベストプラクティス, 効果, 制約:		
[関連ベストプラクティス]		
重要度・緊急度マトリックスによる要求の選別(N02-048)		
[効果、制約、関連等]		

要求工学ベストプラクティス		No. N09
		101
問題分類: 分析・評価	ベストプラクティス: 要求の規模が納期と予算を超えないようにする	
1 版		
前提と課題(開発形態, ドメイン, 組織形態):		
[前提]		
<p>(1) ユーザはシステム化の対象となる業務が少なく、関わる関連部署が少ないため、各部署の要件を一覧にし、業務のシステム化をベンダ側に依頼してきた。それをもとにベンダは概算の金額と納期を提案し、ユーザはその金額と納期で予算と運用開始日を決定した。</p> <p>(2) ベンダ側は、ユーザの要件一覧からヒアリングをし、要件に対する分析を始めると、納期と予算が要求における規模に合わないことが判明した。</p> <p>(3) ユーザとベンダは、システム構築における契約は締結していない。</p>		
[課題]		
<p>(1) 要件の分析が不十分であるにも関わらず、予算や納期が先行している。</p> <p>(2) 要求が納期と予算合わない。</p>		
ベストプラクティス		
[対象プロセス]		
要求管理・プロジェクトマネジメント		
[主導者]		
ベンダ		
[解決の観点]		
解決: ユーザに事業課題からシステム化への整合性と要求の合理性の理解を得て、プロジェクトを円滑に開始できるようにする。		
[解決方法]		
<p>ユーザとの調整を図り、納期と予算内で実現可能な範囲で要件の優先順位付けを再度行い、ユーザと合意形成を図る。</p> <p>例えば、ベンダ側は以下のことを行うことが必要となる。</p> <p>(1) 個々の要件に関わる開発期間と開発コストを算定する。</p> <p>(2) 要件の優先順位に基づいて、WBSを作成し、範囲内かを判定する。</p> <p>(3) 代替案を提案する。</p> <p>(4) ベンダは、営業・システム間での意思統一を徹底する。</p> <p>(5) 決定事項を文書化し、ユーザの利害関係者を含め合意形成を図る。</p> <p>※ 調整が不調な場合に、中止することも考慮しておく。</p> <p>※ 当初見積額より大幅にコストがかかる場合は、最終予定コストを算出し、そのコスト内でシステム構築するように、プロジェクトマネジメントを強化すること。</p>		
<p>要求の優先順位付けについては、他のベストプラクティスを参照。</p>		
備考: 関連ベストプラクティス, 効果, 制約:		
[関連ベストプラクティス]		
重要度・緊急度マトリックスによる要求の選別(N02)		
[効果、制約、関連等]		
プロジェクトマネージャーのネゴシエーションスキル		

要求工学ベストプラクティス		No. N10
		068
問題分類: 整合性・矛盾	ベストプラクティス: 要求間の矛盾解消	
1 版		
<p>前提と課題(開発形態, ドメイン, 組織形態):</p> <p>[前提] 要求提供者が複数存在し、各提供者間での調整は図られていない。</p> <p>[課題] 各提供者の要求間で矛盾が発生する。 (1)ある人曰く「シンプルな見易い画面。操作もシンプルにしてくれ。」 別の人曰く「1 画面、1 機能で実現してくれ。」 また別の人曰く「あの情報とこの情報を一度に見れるようにしてくれ。」 (2)営業部長曰く「売上機会損失を減らすために在庫切品ゼロを目標にする。」 経理部長曰く「キャッシュフロー経営が求められているので、在庫は少なく、短くする。」</p>		
<p>ベストプラクティス</p> <p>[要求ライフサイクル] 獲得、分析、記述、検査</p> <p>[主導者] ベンダ</p> <p>[解決の観点] 予防、発見、解決 要求間の矛盾が後工程になって明らかになることを防止するとともに、ステークホルダの要求事項評価を統一する。</p> <p>[解決方法] (1)RFP のレビューやヒアリングを通して要求事項を収集し、お客様視点の要求事項(要望)を、システム開発完了後に実現できているか評価できるレベルに詳細化する。 1)お客様の要望から、2)システムへの要求事項、3)管理される要求(要件) 補足 A: 1)から 2)への詳細化 顧客の要望からシステムが解決する問題、適える目標に変換して、顧客と合意する。 変換の手法としては、RFP やインタビュー結果のレビュー、ブレーンストーミング、プロトタイピング、ロールプレイなど。 補足 B: 2)から 3)への変換 顧客の要望と、システム化の目標から、システム機能(入力、処理、出力)への要求や非機能(性能、操作性)への要求、制約条件に変換する。 UP(Unified Process)の FURPS+[ファープスプラス] (Functionally(機能)、Usability(ユーザビリティ)、Reliability(信頼性)、Performance(性能)、Supportability (サポート性)、+(制約))、などの分類を切り口として、変換していくと整理がしやすい。 他の尺度として ISO/IEC 9126(JIS X 0129)や、次の 5 つの特性をまとめた RASIS「レイシス」もある。 R(Reliability): 信頼性 A(Availability): 可用性 S(Serviceability): 保守性 I(Integrity): 保全性 S(Security): セキュリティ</p>		

(2) 詳細化された要求事項を、要求事項一覧表による要求事項の一覧化を行う。(一覧表の管理要件までを作成)

ID	顧客要求	顧客要求補足	システム化要求	管理要件	関連管理要件	計測項目	達成値
1	A	Aaaaaaaaa	A-1	A-1-1	C-1-1と並立	クリック数	<3cnt
2	---	---	---	A-1-2	D-3-1と対立	項目数/画面	<20cnt
3	---	---	A-2	A-2-1		処理時間	<3sec

(3) 要求事項を分析し、親子や兄弟関係の構造化、ビジネスゴールへの寄与度、システム化のニーズ、実現コスト、削減効果等の分析を実施し、優先度を判定する。

例)システム化のニーズ:システム化の主目的、効果を高める、副次効果有り、代替でも変わらず、などこのときに、要求一覧表のベンダ内部レビューを実施し、管理要件間での関係性も一覧に表記する。矛盾・対立がある管理要件と、実現のためには開発部分が重複する管理要件を明確にしておく。

ID	管理要件	外部設計書	内部設計書	モジュール	検証	妥当性確認
1	A-1-1	入力画面標準設計書	内設 A			
2			内設 B			
3	A-1-2	A21 入力画面				

補足 C:矛盾の発見

管理要件を図式化することで発見の手助けとすることができる。

DFD、ER 図、状態遷移図、ユースケース、オブジェクト指向分析、など

補足 D:要求一覧表の記載事項

本文中に記載した事項とともに、要求から設計、実装、検証までの垂直性を含めておくことと要求への変更発生時に、どこに影響があるかが管理しやすい。

また不具合があった場合に、どの不具合を優先的に対応しなければならないかの判定に対する1インプットとして利用できる。

(4) 要求事項一覧表に基づいて、お客様とシステム化要求事項の合同レビューを実施する。

各要求事項の分析結果、優先度を評価し、実施や見送りの取り扱いを決定する。

矛盾する要求事項の取り扱いは、中間点近くでバランスを取るケースや、矛盾要求間から要求を択一するケースなどが考えられる。

膨張しすぎた(納期やコスト、体制などと矛盾のある)要求事項も、優先度や実現期間、実現コストなどから今回のシステム開発プロジェクトの範囲外とすることを申し入れる。

結果に従って要求事項一覧表を更新するとともに、議事録にて関係者に相違ないことを確認する。合意にたどり着けない場合は、QCD への影響を説明し、契約の改訂も念頭に置いておく。

備考: 関連ベストプラクティス, 効果, 制約:

[関連ベストプラクティス]

ステークホルダによって異なる要求の調整(A04)

問題分類: 整合性・矛盾

ベストプラクティス: 2 チーム制による要求矛盾の回避

1 版

前提と課題(開発形態, ドメイン, 組織形態):

【前提】

- (1) 要求フェーズでの成果物が複数作成され、概要設計のインプットにされること。
- (2) 要求フェーズから概要設計にかけてのフェーズにおいて、要求獲得スキルの高い要員のもとにワーク作業を担当できるメンバが手配できること。

【課題】

ドキュメント間での機能名称などの管理を矛盾無く実施すること。

№82,93: 要求間の整合性の確認方法: 整合を維持するツールの導入、タイムリーな問い合わせ可能な窓口設定、要求プロセス内のサイクルを回す。

ヒアリングした要求を分析、文書化する際に、要求間での矛盾が発覚し、後の設計に影響が波及する。

ベストプラクティス

【対象プロセス】

要求開発プロセス

【主導者】

ベンダ

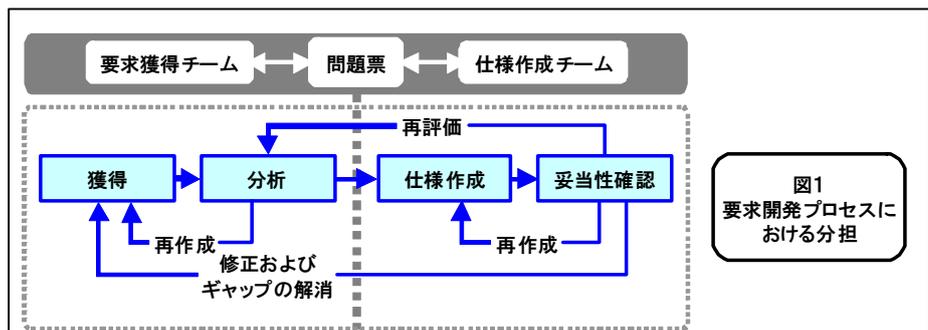
【解決の視点】

解決(プロセス)

【解決方法】

要求フェーズにおいて要求獲得・分析担当と仕様作成・妥当性確認担当の 2 チーム制を導入し、要求間の矛盾の洗い出しと改善を行い、要求開発プロセスを効率化する。

- (1) 要求の獲得と分析を担当する「要求獲得チーム」と、仕様作成と妥当性確認を担当する「仕様作成チーム」の 2 チーム制にする。仕様作成チームは要求開発ツール(以下、ツール)への入力時に発せられる警告により妥当性確認を担当する。(図 1)
- (2) 仕様作成チームが精査して発覚した問題点は「問題票」に記入し、要求獲得チームと問題点を共有する。



作業概要

- (1) 要求獲得～分析: 要求獲得チームは、顧客要求のヒアリング結果を Excel で記述する。
- (2) 分析～仕様作成: 仕様作成チームは、Excel に書

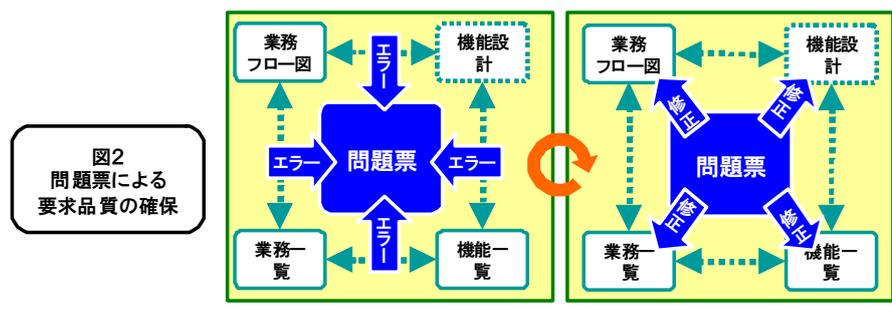


図 2 チーム制による要求開発プロセス

かれた内容を、ツールに入力する。

(3)仕様作成～妥当性確認:ツールへの入力時に、要求間の矛盾があると入力できないか、警告が発せられるので、その内容を問題票に記述する。

(4)妥当性確認～分析または獲得:要求獲得チームは問題票に記入された問題点の解決方法を問題票に記入する。仕様作成チームは、問題票に記入された解決方法により、ツールへの入力を行う。

要求フェーズのライフサイクルでは継続的に実施、開発側(受注側)で主導して行う。また、本プラクティス実施において、顧客視点を持ち要求内容を精査する事を同時に実施する必要がある。

備考: 関連ベストプラクティス, 効果, 制約:

[効果]

システム全体の 13%相当の機能で要求矛盾を発見修正 (ファンクションポイント換算)。

[制約]

2 チーム制を導入するのに見合うシステム規模であること(事例では約 4000FP)。

ツールは、要求間の矛盾を自動的に検出でき、オペレーションが容易であること。

ツールの導入コストの費用対効果を算定済みであること。

ここでの要求の矛盾とは、作成されるドキュメント間の矛盾であり、ビジネスロジック上の矛盾は含まれない。

[参考資料]

[1] K. E. Wiegers (渡部洋子(訳)), ソフトウェア要求:顧客が望むシステムとは, 日経 BP ソフトプレス, 2003.

要求工学ベストプラクティス		No. V01
		011
問題分類: 検証・レビュー	ベストプラクティス: 適切なコミュニケーションによる要求の確認	
1 版		
前提と課題(開発形態, ドメイン, 組織形態):		
[前提]		
<p>(1) ユーザ側システム部門 (以降システム部門と表記) は、ユーザの要求を元に要求仕様書、又は要求に関するメモを作成した。</p> <p>(2) システム部門から要求仕様の説明を受け、ベンダ側は要件定義書を作成し、確認のためシステム部門に送付した。</p> <p>(3) システム部門の承認を得たのでベンダ側は開発に着手し、統合テストまでプロジェクトが進行した。</p>		
[課題]		
<p>ベンダが統合テスト結果の説明を行ったところ、同席したユーザから誤認を指摘され、仕様の変更が発生した。また、要件定義書記述内容に対するシステム部門もしくはベンダ側の誤認が判明し、大きな手戻りが発生した。</p>		
ベストプラクティス:		
[対象プロセス]		
要求獲得、検査		
[主導者]		
ユーザ、ファシリテータ、ベンダそれぞれにおいて適切な時期や工程でのお互いのコミュニケーションが必要なので、3 者共に主導者と言える。		
[解決の観点]		
<p>予防: システム部門にて要求仕様書等を作成した場合、ベンダに提示する前にユーザとシステム部門で要求確認レビューを行い、要求に対する認識の統一を図り、開発時の手戻りを予防する。また、システム部門とベンダは適切なコミュニケーションを行い、要件に対する認識を確認する。</p>		
[解決方法: ユーザ/ファシリテータ側]		
<p>この課題は、システム部門がユーザの要求を取りまとめ、そのままベンダに開発を依頼した場合に起こることが多い。ベンダにとっては要求仕様を元に要件定義したわけであるから、開発の最終段階で仕様変更となることは納期及び利益の圧迫や士気の低下のみならず、品質の低下につながる最も大きい要因になる。それはユーザにとって運用上の大きなリスクとなってはね返ってくる。</p> <p>したがって、この課題はユーザとシステム開発をベンダに依頼するシステム部門間の検証の問題として捉えられる。システム部門は効率性と確実性を求めるが、ユーザは現実性と利便性を求めている。その思いの違いからユーザとシステム部門の意見は食い違うことが多い。その状態でシステム開発を依頼することは無駄になりかねない。したがって、十分な話し合いにもとづく歩み寄りと合意が必ず必要である。そのためにベンダへ要求仕様書等を提示する前段階において、</p> <p>(1) ユーザとシステム部門との間で要求確認レビューが必要不可欠である。</p> <p>(2) レビューの成果物としての要求仕様書は、現在よくあるような要求仕様メモではなく、発注者としての公式な文書化を行うべきである。</p>		
[解決方法: ベンダ側]		
<p>この場合、要件定義書作成以降、必要な工程 (例えば外部設計やテスト計画段階) での成果物のレビューや上流工程でのプロトタイプ作成・検討を行い、要求の実現度合いや妥当性について確認し合意を重ねることが必要である。</p>		

この課題は、主にシステム部門とベンダ間の、要求を要件に展開する際、およびその後の設計段階における認識の差の問題である。その差を埋めるためにさまざまなコミュニケーションを行わなければならない。

また、要求仕様を実現化するための要件化や設計書作成には技術力や理解力もさることながら、文章による伝達能力も大きく影響することを忘れてはいけない。さらに、出来上がり具合をユーザが随時確認することも必要である。

これらに対応するには、以下のコミュニケーション手法が挙げられる。

(1)レビュー

開発の初期段階の成果物(例えば要件仕様書や基本設計書)を、システム部門と分野の熟練者を含むベンダ側との間にて十分にレビューし、要件の実現性と整合性を確認する。また、特殊な条件を持つ機能や制限事項が多い場合には、さらにテスト計画書のレビューも必要である。

レビューは以下の3点に留意する。

- 1) 立場が異なれば同じものでも見る視点が異なってくる。したがって、レビューはその視点・観点を明確にして実施する(例:インタフェースに着目する、または画面遷移に着目する等)。
- 2) 文章が誤解を生むような構成になっていないかに留意する(例:主語と述語が離れている)。
- 3) 記述のない非機能要件について、誤認を避けるためにできる限りの洗い出しと文書化を行う。

(2)プロトタイプ作成

コミュニケーションの効果的なもう一つの手法としてプロトタイプ作成がある。

この手法はシステムの動きや表示を目で実際に見ることができるので、システム部門のみならずシステムの設計書を読み解くことが不得手なユーザにとっても効果的な検証手段である。プロトタイプは本格開発の前に行うので、ベンダ側にとっても要件の確定とともに、実現化手法の可能性確認を行うことができるので有益である。

ただし、注意すべき点は、プロトタイプを見てさらに仕様が増大または大きく変わる場合である。この場合、再見積に基づいた納期および経費の見直しと合意が必要となる。

いずれにしてもユーザ及びシステム部門とベンダは、レビューやプロトタイプを通じてコミュニケーションを十分にとり、要件に合致したシステムを協働して作り上げなければならない。

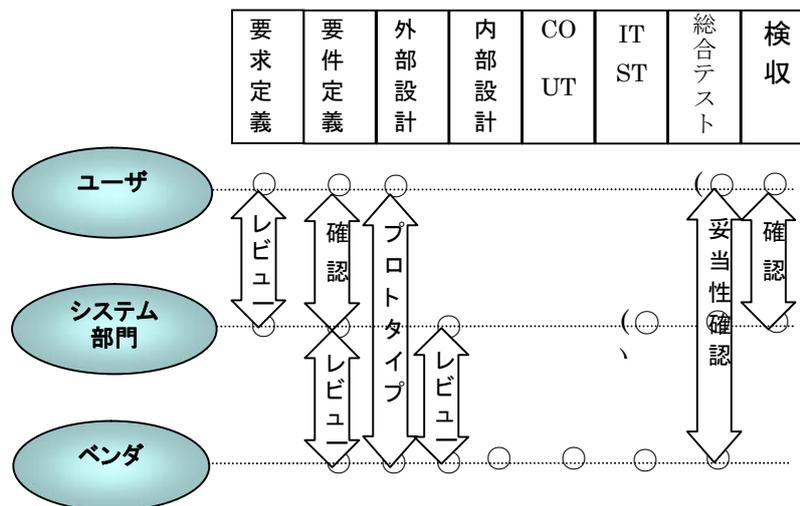


図1 ステークホルダ間で必要な主なコミュニケーション

備考: 関連ベストプラクティス, 効果, 制約:
[関連プラクティス]

[本文内容と共通フレーム 2007 との対比]

本文	共通フレーム 2007
レビュー	2.6 共同レビュープロセス
要求定義	1.4 企画プロセス、1.5 要件定義プロセス
要件定義	1.6.2 システム要件定義、1.6.4 ソフトウェア要件定義
外部設計	1.6.5 ソフトウェア方式設計
内部設計	1.6.6 ソフトウェア詳細設計
CO、UT	1.6.7 ソフトウェアコード作成およびテスト
IT、ST	1.6.8 ソフトウェア結合、1.6.9 ソフトウェア適格性確認テスト
総合テスト	1.6.10 システム結合、1.6.11 システム適格性確認テスト
検収	1.6.12 ソフトウェア導入、1.6.13 ソフトウェア受入れ支援

要求工学ベストプラクティス		No. S01
		088a
問題分類: 戦略	ベストプラクティス名: 要求と経営戦略との整合	
1 版		
<p>前提と課題(開発形態, ドメイン, 組織形態):</p> <p>[前提] 顧客のシステム導入経験があまりない。</p> <p>[課題] (1) コンサルティングフェーズを受注。顧客課題解決が必要。 (2) 顧客がシステム構築も含め、長期的な支援を希望。</p>		
<p>ベストプラクティス</p> <p>[対象プロセス] 要求開発・管理プロセス全体(要求獲得時、要求分析、要求仕様記述、要求検査、要求管理)</p> <p>[主導者] ベンダ(IT コーディネータ(以下、ITC と略す))。 要求のライフサイクル全体を主に実施するのはユーザ。ITC はそれを支援、助言する立場となる。 本プラクティスにおいて ITC はベンダ側の人材であるが、顧客にコンサルタントとして入っており、役割としてはファシリテータにも近い。</p> <p>[解決の観点] (1) 要求獲得時において十分抽出したと思われた要求が、経営戦略を考慮されず、システム化の目的が明確化されないゆえにシステム構築途中で要求変更が頻発してしまったり、できたシステムがその会社(組織)の業務改善に役立たないシステムとなり、あまり使われないケースがある。ITC が経営戦略の策定からかわり続けることで、システム構築プロジェクト遂行中も経営戦略との整合性を意識することができ、こういったことを予防できる。 (2) ユーザ企業の中で要求がまとまらないときにも ITC がファシリテータの役割を持つことで要求定義作業をより効率的、効果的に進めることができる。</p> <p>[解決方法] 本プラクティスでは、図 1 に示すように、要求のライフサイクル全体を ITC が支援している。 ITC のプロセスでは、ユーザ企業の経営戦略からの策定、展開からプロセスとして扱っており、その後も経営戦略との整合性をチェックしながら進めていく。システム化の目的を的確に導き出すプロセスとしても有効である。</p> <p>(1)ITC プロセス(以下の 5 つのフェーズ)全体を支援する担当者を置き、ライフサイクル全体を通じた支援を行った。</p> <ol style="list-style-type: none"> 1) 経営戦略フェーズ 2) IT 戦略策定フェーズ 3) IT 資源調達フェーズ 4) IT 導入フェーズ 5) IT サービス活用フェーズ <p>なお、RFP の作成は、3)IT 資源調達フェーズで行なう。</p>		

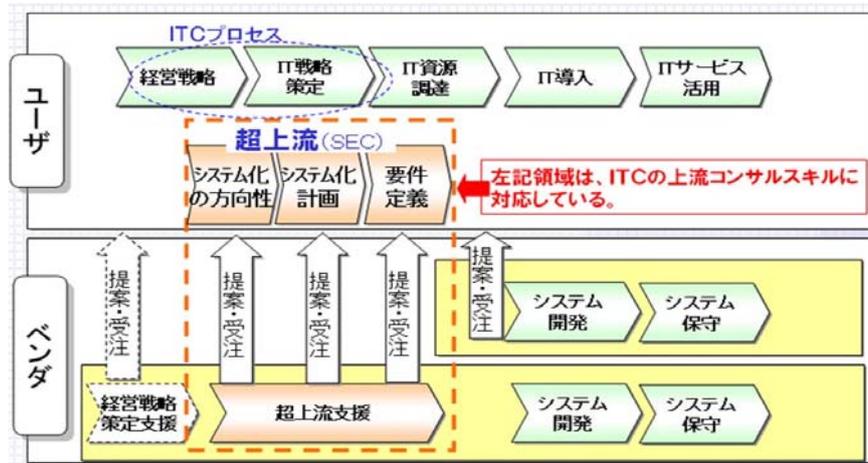


図1 ITCによる要求ライフサイクルを通じた支援

(3) 経営戦略フェーズの主要成果物

- 1) 事業特性・業界特性の分析(5Forceと呼ぶ)(図2参照)
- 2) 現状認識・課題認識
- 3) SWOT(自身の強み弱み、外部の機会と脅威)分析(図3参照)
- 4) 保有する業務ノウハウなどの強み、業界の変化に対する弱みなどを分析し、主要成功要因(CSF)を抽出。
- 5) 新しい業務モデルの図式化

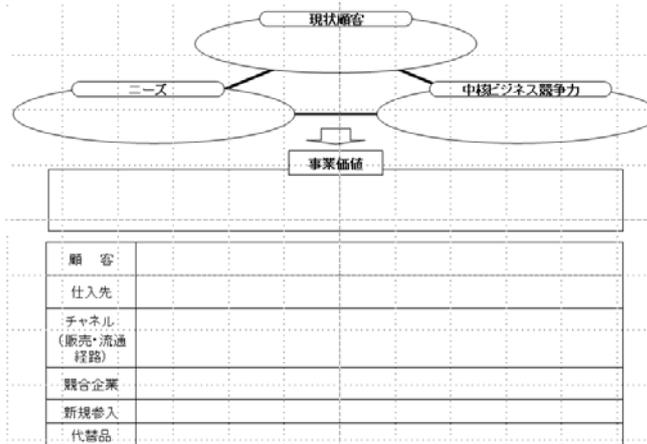


図2 事業ドメイン分析と業界特性分析(5Force)

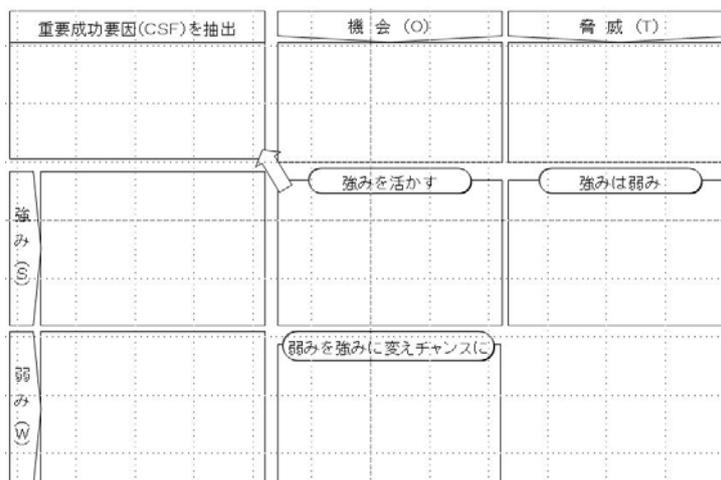


図3 SWOT分析・主要成功要因(CSF)抽出

(3)IT 戦略策定フェーズの主要成果物

- 1) 業務モデル分析、検討(機能分析、ワークフロー可視化ツール等の利用)
- 2) 代替案の分析、検討
- 3) グランドデザインに関する提言のとりまとめ

備考: 関連ベストプラクティス, 効果, 制約:

[事例]

要求定義における IT コーディネータの活用事例(本報告書所収)

[参考資料]

[1] IT コーディネータ協会, <http://www.itc.or.jp/>.

[2] ITコーディネータ協会(編), 成功したい社長が読むIT経営のススメ, アイテック, 2007.

2. お客様にわかりやすい情報システムの設計図作りをめざして～発注者ビューガイドライン

富士通株式会社 銀林 純

2. 1. はじめに

発注者ビュー検討会とその最初の成果物であるガイドラインについて紹介する¹。

発注者ビュー検討会は、正式名称は「実践的アプローチに基づく要求仕様の発注者ビュー検討会」という大変長い名前です、略して発注者ビュー検討会と称している。設立趣旨は「お客様にわかりやすい仕様記述方法および合意方法を共同で検討する」となっているが、注意頂きたいのは、仕様記述方法だけでなく合意方法を検討する点も含むことである。仕様記述方法についてはこれまでいろいろなところで検討・追究されてきたが、その仕様について発注者と開発者の間の合意をどう形成していくかという観点に踏み込んだ取り組みはなかったと思われる。

検討会は2006年4月にNTT データ、富士通などの6社で設立し、2007年8月からさらに3社が加わり、計9社の検討会となっている。この9社を見て頂くと、すべてベンダー企業（開発者側の企業）である。NTT データが幹事役を務め、検討会の公式ホームページもNTT データのサーバで運営している。なお、2008年度以降は、活動の中心が独立行政法人 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（IPA SEC）に移る予定である。

検討会が対象とした工程は外部設計の工程である。これは要件定義が終わりシステム設計に入った最初に、作るべきシステムの外部仕様を設計していく工程となる。要件定義は概ね発注者の責任で作成・完成させるものである。しかし、外部設計になると開発者が設計書を作成し発注者にその内容をレビュー・承認してもらうという作業分担になることが往々にしてある。つまり、この検討会は、開発者がどう設計書を書き、どう発注者にレビューしてもらえば誤解や齟齬が減るかという観点で、開発者が努力すべき事項をまとめようという想いでベンダー9社が集まったものである。

2. 2. 発注者ビューガイドラインの構成について

ガイドラインは次の3編から成る。

(1) 発注者ビューガイドライン(画面編)

画面設計を対象として、コツをまとめる。

(2) 発注者ビューガイドライン(システム振舞い編)

システム化の範囲がわかる業務フロー図やそれに付随する設計要素を対象として、コツをまとめる。

(3) 発注者ビューガイドライン(データモデル編)

ER 図や CRUD 図の記述を対象として、コツをまとめる。

¹ 2007年9月の検討会参加企業9社による合同記者会見の際のプレゼン資料と、記者会見同日に一般公

2. 3. 発注者ビューガイドライン(画面編)について

ここでは、発注者ビューガイドラインの中で最初に策定した画面編について紹介する。

2. 3. 1. 発注者ビューガイドライン(画面編)の策定と公開

発注者ビューガイドライン(画面編)は、発注者と開発者の間の誤解を防ぐ、あるいは認識のずれを見つけ出すためのコツ(工夫)や留意点をまとめたドキュメントである。当初から参加の 6 社がそれぞれ実際の設計事例からコツの候補を持ち寄った。それを 6 社で吟味し、最終的に計 80 個ほどのコツがガイドラインというかたちでまとまった。さらに、それを東京証券取引所、AGS に事前レビュー頂き、コメントを反映して一般公開した。

発注者ビューガイドライン(画面編)の内容は以下のようにになっている。

第 1 部:表現

画面に関連する外部設計書の記述のコツを集約

画面一覧、画面遷移、画面レイアウト、画面遷移・レイアウト共通ルール、入出力項目、アクション明細

第 2 部:記述確認

外部設計書の記述を確認する際に留意すべき事項をまとめたチェックリスト

第 3 部:レビュー

外部設計書の記述内容を発注者と開発者が確認する際に、認識のずれを起こさずに合意するためのコツ

2. 3. 2. 設計書の書き方に関するコツ

実際のガイドライン(画面編)では、まず、設計書の書き方のコツがひとつひとつ具体的に説明されている。いくつかの例を紹介する。

(1) FD1002「画面一覧に記述する分類には、あらかじめ定義された分類を使用する。(以下省略)」

このコツにより、発注者が画面一覧という設計書をレビューしやすくなる。

(2) FD2002「画面名が、業務的に意味のある識別名になっていないと、画面遷移上で実際の画面が想起しにくく確認しづらい」

これは、例えば、「ペット登録画面1」、「ペット登録画面2」のようについつい通し番号を振って画面名にしてしまうこともあるが、それは望ましくないという意味である。「ペット登録ヘッダ【入力モード】」、「ペット登録詳細【入力モード】」のように、しっかり意味のある名前を付けて区別しやすいようにしようというコツである。

(3) FD2003「画面遷移の遷移矢線が錯綜していると、処理の流れが煩雑となり、ユーザを混乱

開したガイドライン(画面編)に基づく。

させる」

(4) FD2004「画面遷移が上から下、左から右に遷移するように配置する」

これらは、いずれも、言われてみれば当たり前のことであるが、このようにドキュメントとしてまとめられてはいなかった。

次のコツは非常に重要である。画面レイアウトという設計書において、例えば何かを検索した結果を表示する画面があったとき、検索結果を「必ず 4 件表示」するのか「最大 4 件表示」するのかは、外部仕様では大きな違いとなる。検索結果が 3 件以下のときにも 4 件分の枠が表示されるか、それとも枠が小さくなるのか、という違いとなる。このコツは、画面レイアウトにおいて枠が固定長か可変長かをしっかり区別し、それを発注者に確認すべし、と主張している。

画面レイアウトに関して、もうひとつ紹介したいコツがある。画面レイアウトというと、Power Point や Excel 等で矩形を組み合わせて描いた画面イメージを作成する場合と、HTML 等で見栄えを含めた実際の画面を作成する場合がある。どちらも有用である。一見、HTML 等で作成した実際の画面の方が、出来上がり画面により近いので発注者のレビュー・確認にも良さそうに思われるが、実はそうとは限らない。設計の初期では、画面の色や文字フォントといった見栄えよりも、その画面でどういう入力項目がありどういう出力項目があるかを洗い出すことが重要である。このような初期の段階で HTML 等の画面をレビューしだすと、どうしても見栄えに気が取られ、本当に必要な項目の洗い出しに集中できない、といった弊害も起こりえる。より素朴な矩形を組み合わせた画面イメージの方が都合が良い場合もある。

2. 3. 3. レビューに関するコツ

このガイドラインでは発注者のレビューを大きく次の 3 段階(仕掛期、充実期、完成期)に分け、それぞれの段階でのコツ(工夫)をまとめた。

表 2.1 合意形成の 3 段階フェーズ

仕掛期(前期)	合意形成を開始。レビュー対象となる工程成果物の量が大幅に増加するフェーズ
充実期(中期)	合意形成が大幅に成熟。工程成果物の量は小幅に増加、バグ発見と修正が主となるフェーズ
完成期(後期)	合意の成熟を達成。工程成果物の量や修正数は安定、承認が主となるフェーズ

これは、同じ設計書、例えば画面遷移という設計書ひとつをとっても、時期によってその期待される完成度が異なるからである。例えば画面遷移は、仕掛期には「すべての画面と、典型的な遷移矢線が記述されている」ことが期待されるのに対し、充実期には「すべての画面とすべての遷移矢線が記述され、バグと対策が明らかになっている」、さらに完成期には「完成の状態にあり、審査・承認の作業のみ管理される」という状態になっていることが期待される。つまり、発注者がレビューするポイントが時期によって異なる。

レビューの進め方のコツをひとつだけ紹介する。

FR0002「他の画面あるいは工程成果物への影響度に留意して説明順序を決定する」

充実期におけるコツのひとつで、例えば画面レイアウトの共通部分や画面遷移の共通部分(エラー処理など)のような何回も登場する共通的な仕様を先に説明・レビューすることで、合意形成を効率化し手戻りを防ごうとするものである。このようなコツも、聞けば当たり前のことであるが、あらかじめ知っているのと役に立つ場合が多い。

2. 4. 今後の活動について

最後に今後の活動について紹介する。

まず、今回のガイドライン(画面編)については、当初から参加の 6 社はそれぞれ、今後、自社の開発標準に組み入れることを表明した。例えば、富士通ではSEにSDEMという開発標準が浸透しているが、SDEMにガイドラインの中身を組み入れる作業を既に開始している。また、新規の3社もそれぞれ検討に着手している。

発注者ビュー検討会は2008年3月に活動を終了し、2008年度からは、作成したガイドラインを改善し普及する活動を、IPA SECで引き継いで行う予定である。

[参考文献]

- [1] 実践的アプローチに基づく要求仕様の発注者ビュー検討会, 発注者ビューガイドライン(概説編), Ver. 1.0, 2008年3月, <http://www.nttdata.co.jp/cview/>.
- [2] 実践的アプローチに基づく要求仕様の発注者ビュー検討会, 発注者ビューガイドライン用語集, Ver. 1.0, 2008年3月, <http://www.nttdata.co.jp/cview/>.
- [3] 実践的アプローチに基づく要求仕様の発注者ビュー検討会, 発注者ビューガイドライン(画面編), Ver. 1.0, 2008年3月, <http://www.nttdata.co.jp/cview/>.
- [4] 実践的アプローチに基づく要求仕様の発注者ビュー検討会, 発注者ビューガイドライン(システム振舞い編), Ver. 1.0, 2008年3月, <http://www.nttdata.co.jp/cview/>.
- [5] 実践的アプローチに基づく要求仕様の発注者ビュー検討会, 発注者ビューガイドライン(データモデル編), Ver. 1.0, 2008年3月, <http://www.nttdata.co.jp/cview/>.

3. ユーザ自身による RFP 作成を可能にする RFP テンプレートと RFP 作成プロセス及び実践事例

MPUF RFP 研究会 立花 豊文

3. 1. はじめに

情報システム開発に関わるトラブルは後を絶たず、納期の遅延、大幅な予算超過だけでなく、折角構築されたシステムも発注側の期待に反し、最悪の場合は使われないということが起きている。この原因が要件定義に起因しており何らかの対応が必要だという認識は多くの関係者に一致するところである。MPUF RFP 研究会[1]はこのような現状に対して「ユーザ主導の戦略的 IT 利活用における RFP・要件定義フェーズについての研究、および成果の普及活動」を目的に 2005 年 5 月に立ち上げられた。ベンダ、官・民のユーザ、コンサルタントという多様な参加者から構成された本研究会の研究によると要件定義に関わる問題の多くはユーザが要件定義フェーズにおいて主体性を発揮せず、コンサルタントやベンダに任せていることに起因しており、ユーザ自身が現状分析、課題抽出、課題解決の検討、関係者の合意、新システムの要件整理といったプロセスを経た RFP (Request For Proposal) を作成し、ベンダに提示することがシステム開発に関わるトラブルを解消する最良の方法であるとの認識に達した。

本報告は、研究会で提案しているユーザ自身が自らの手を使って(主体性をもって)RFP を作成するための方法とユーザの実践例の研究を通して得た知見を紹介するものである。

3. 2. 求められるユーザ主導による RFP 作成

3. 2. 1. ユーザとベンダの要件定義に関する見解

情報システムの開発において要件定義は、発注側(ユーザ)と受注側(ベンダ)双方に大きな課題を突きつけており、このことはいくつかの調査結果が示している。

例えば、情報サービス産業協会が実施している調査[3]によると、通常ベンダにおける情報システム開発における関心は、品質、コスト、納期の 3 点であるが、この 3 点に影響を及ぼす主要な要因は以下ようになっており、要件定義が全てにおいて影響ありとされている。つまり、ベンダにとって、要件定義は情報システム開発の成否を握る重要課題なのである。

(1) 品質に及ぼす要因

1) 要件定義(23.8%) 2) 要員の技術力(17.4%) 3) 開発体制(15.3%)

(2) コストに及ぼす要因

1) 要件定義(21.8%) 2) 開発体制(15.4%) 3) プロジェクト計画(14.7%)

(3) 納期に及ぼす要因

1) 進捗管理(25.2%) 2) プロジェクト計画(24.3%) 3) 開発体制(13.6%) 4) 要件定義(13.3%)

さらに、その要件定義のどこに問題があるかという点について言えば、情報サービス産業協会の調査研究[2]によれば、ソフトウェア開発の問題要因ワースト3は以下のとおりであり、このことからベンダは顧客側に問題があると考えていることがわかる。

- (1) 要求内容の相違(16.5%)
- (2) 顧客自身が要求を把握せず(13.6%)
- (3) 顧客要求の頻繁な変更(13.6%)
- (4) 顧客の協力不足(7.9%)

一方、MPUF が 2007 年に行ったアンケートによればユーザが抱えている IT 調達上の課題には次のようなものがあり、課題の原因は自分達にもあるとユーザも認識していることがわかる。

- (1) IT 投資を本当の業務改革に結び付けられない。
- (2) あいまいな仕様での発注による手戻りが多々発生している。
- (3) 特定ベンダに依存している。
- (4) 適正な価格、調達の費用対効果をどのように出すのかわからない。
- (5) 要件定義書・仕様書が書けない、書き方がわからない。

以上のことから要件定義に関わる課題解決にはユーザが主体性を発揮する必要があるという、この点においてベンダ側もユーザ側も一致していると言える。

3. 2. 2. なぜ、ユーザ自身が RFP を作成すべきなのか

要件定義においてユーザが主体性を発揮するということは、どういうことか。

情報システム開発における要件定義フェーズはベンダの開発工程に組み込まれていることが多く、業務要件作成に当たってベンダが行うユーザからの聞き取り、理解、記述の各段階で内容が一部欠落することが考えられる。この欠落がユーザの期待するシステムと出来上がったシステムとの間にギャップを生み、前述のような問題を引き起こす大きな要因になっているのではないかとすれば、次のような状況に置かれていると推測されるユーザ自身が自ら要件定義を行うこと、つまり RFP を作成することこそ本来のあるべき姿である。それによって、現在抱えている要件定義に関わる問題の解決、ひいては情報システム開発の問題解決につながると考えることは自然であろう。

- (1) ユーザはビジネス競争力強化のための業務改革を支援する IT を必要としている。
- (2) しかし、ユーザは現状の IT 調達に悩みを持っており、対応が急務だと思っている。
- (3) そして、何よりユーザは業務に精通しており業務上の課題を認識している。

しかも、ユーザ自身も先のMPUFのアンケートでわかるように自らRFPを作成すべきではないかと思っているのである。しかし、書けない。では、どうすればよいのか。

3.3. ユーザ主導によるRFP作成を実現する方法

ユーザ自らRFPを書くことが出来るようになるためには次の3つの課題を解決する必要がある。

- (1) RFPに関する知識の獲得
- (2) RFPの作成方法の習得
- (3) 経営者をはじめとする関係者の思いを調整し、RFPに反映する方法

これらの課題は知識や技術だけではなく、関係者の意識を変え、合意を形成するといった課題を含んでいる。これらを解決しユーザ自身によるRFP作成を実現するには、次の3つが揃う必要がある。

- (1) RFP作成ツール
- (2) RFP作成プロセス
- (3) RFP作成チームの運営と合意形成の仕組み

以下、この概要を紹介する。

3.3.1. 業務要件3点セットとRFP作成ツール

研究会では、Webのページで公開している「簡単RFPテンプレート」と呼ばれているRFPの作成ツールを提供している。これは、ユーザ自身がRFPを作成するために必要と思われるRFPの基礎知識や各種雛形などで構成されており[4]、業務要件作成時に使用する「業務要件3点セット」(以下3点セット)はこの中に含まれている。

3点セットは業務を表現するための次の三つのシートと記述ガイドからなる。業務をこの三つのシートに表現(可視化)することで、業務主体者自身による業務と課題の整理を行うとともに第三者に業務を理解してもらうためのベースを作る。

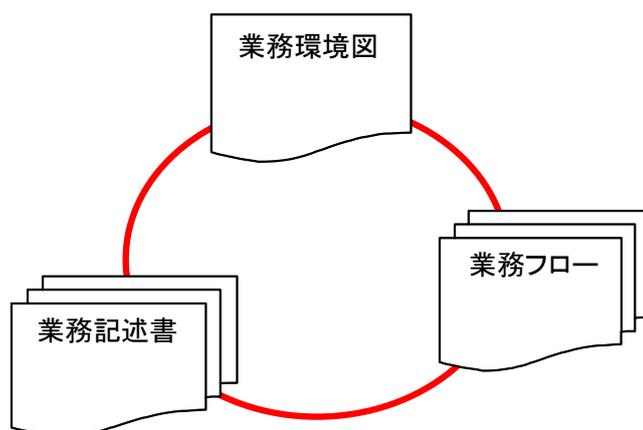


図 3. 1 業務要件の 3 点セット

(1) 業務環境図

IT 調達の対象業務の範囲を概観するためのシートで、作成することで業務主体者とその業務範囲が明らかになる。なお、このシートは業務フロー作成時の前提資料となる。

(2) 業務記述書

業務内容を記述したシートで、作成により対象業務の業務ルール、業務手順が明確になる。なお、このシートは業務フローを作成するときの前提資料となる。

(3) 業務フロー

業務の流れ(登場人物と業務内容の関係)を記述するためのシートで、作成により業務の全体像が明らかになる。

3. 3. 2. 作成プロセス

プロジェクトマネジメントにおける実行フェーズに当たる RFP 作成プロセスは大きく分けて次の 7 つのステップからなる。

(1) As-Is 作成

As-Is 3 点セットを作成する。

<作成の要点>

業務主体者を中心にして、あるがままに書く。

(2) 課題抽出

As-Is 3 点セットから課題を抽出し、課題一覧表を作成する。

<作成の要点>

システム開発の目標や経営課題もまとめておく。

(3) 課題の整理

課題を整理し、調達するシステムで解決すべき課題、解決の方向を明らかにする。

<作成の要点>

システム開発の目標や経営課題もまとめておく。

(4) 新システム概要図作成

課題の整理の結果に基づき新システムの概要図を作成する。

<作成の要点>

新システムのスコープ、機能及び現状(As-Is)システムとの相違に留意する。

(5) To-Be 作成

To-Be3点セットを作成する。

<作成の要点>

新システムを中心にして、システム調達後、つまり課題が解決されたあとの業務を表し、業務要件が遂行できるか確認する。

(6) 業務要件の作成

To-Be3点セットから業務要件を作成する。

<作成の要点>

調達に当たっての優先順位を決め、リストにしておく。

(7) RFP の作成

別に作成した非業務要件と合わせて、RFP に纏めあげる。

MPUF RFP 研究会

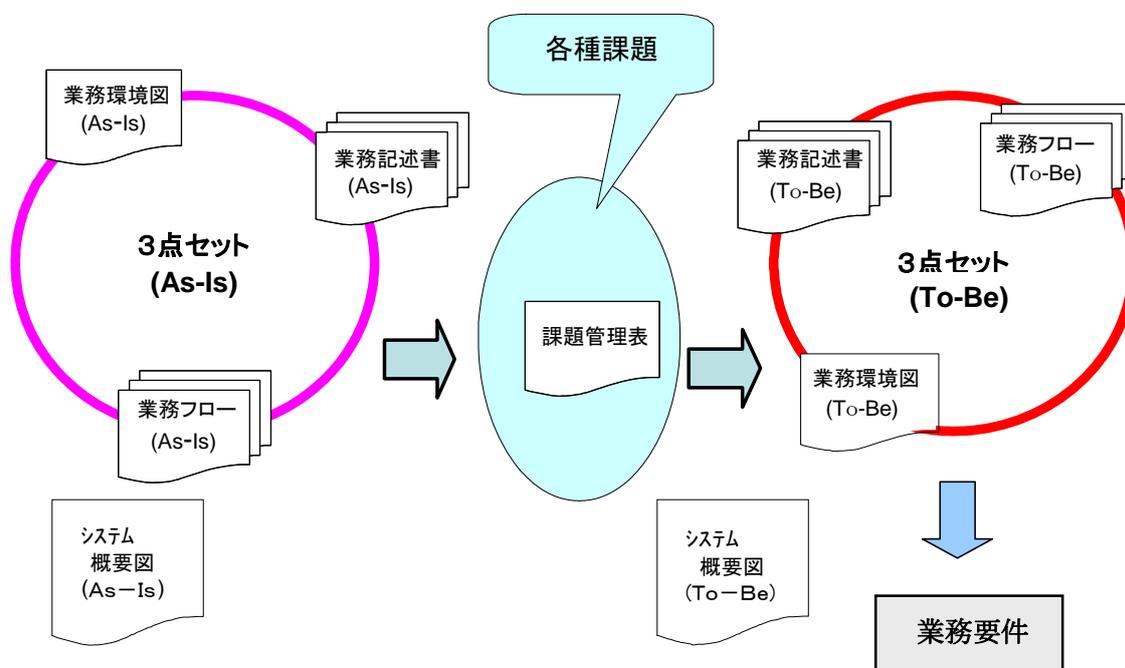


図 3. 2 業務要件作成の流れ(概要)

3. 3. 3. RFP 作成チームの運営と合意形成

多くのプロジェクト運営と同様、RFP 作成においてもチーム運営は成否を左右する。研究会におけるプロジェクトマネジメント経験者とRFP作成経験者との議論から、合意形成を含むチーム運営におけるプラクティスとして以下を提案する。

- (1) 業務主体部門とシステム部門の代表で業務要件作成チームを構成し、事務局を置く。
- (2) チーム事務局は、RFP を取りまとめるとともにファシリテーターの役割を担い、要件の優先順位付け等の合意を形成するための仕組みを提供する。
 - 1) 経営者
 - 2) 業務部門(ミドル、現場担当者)
 - 3) 情報システム部門
- (3) 合意形成のための仕組みとして、作成プロセスの各ステップにおけるレビューと、ステップからステップに移るタイミングでのトップを含めた関係者による優先順位等の意見調整ミーティングを設ける。
- (4) 非業務要件については、専門家の協力を得てシステム部門が作成する。
- (5) RFP はシステム部門または事務局が業務要件と非業務要件を纏める形で作成する。



図 3. 3 RFP 作成担当例

3. 4. 実践例

「ユーザ主導の RFP による調達」の実践事例として、東京都千代田区統合事務管理(共通基盤)システムの導入がある[5]。この事例はユーザ主導による要求開発パターン事例としても重要である。以下、要点を纏める。

(1) ユーザ自ら RFP を作成するに至った経緯

主導の RFP 住民情報システムを再構築した際に以下の教訓を得られた。

千代田区統合事務管理(共通基盤)システムの導入にあたり、この教訓を生かし業務要件に注力、つまり自ら RFP を作成することにした。

教訓:業務改革に時間がかないと、運用後に業務が煩雑になる等の問題が発生する。

(2) 業務の可視化と改革

1) 業務主管課担当者主体による現状業務の可視化(As-Is)、改善項目の抽出、To-Beモデルの作成

2) ファシリテーション、記述指導など RFP 作成支援のためにコンサルタントを活用。

(3) 検討委員会による業務要件作成推進

1) 検討委員会に業務主管課の担当者も参加。

2) 委員会の活動

・業務フロー等のレビュー

・連携部分等のレビュー

・問題点の明確化

・成果物のブラッシュアップ

(4) 実践による知見、他

1) 委員会による RFP 作成を円滑に推進するためには、トップの了承、ボトム(担当者)の合意だけでなくミドル層(業務主管課長)の同意も必要不可欠である。

2) システム更新時には時間をかけて業務も見直すべきである。

3) 新たなシステム導入には「思い」を持ったキーマンが必要である。

4) ベンダ任せにせず、発注者として果たすべき責任を果たすことが重要である。

3. 5. 得られた知見

RFP 研究会が提案している RFP テンプレートとそれを使った RFP 作成方法は、実際にユーザに使われ事例として公開されるに至っている。このような状況を受けて行なった研究で得られた知見を紹介する。

(1) ファシリテーターの重要性

要件定義におけるファシリテーターの重要性については、要求工学実践パターンのベストプラクティスの一つとしてファシリテーター主導要求開発パターンがあげられているとおりである [5]が、ここでは 2 つのファシリテーターが必要であることを指摘しておきたい。ユーザ内の関係者を仲介するファシリテーターとベンダとユーザを仲介するファシリテーターである。いずれの役割も重要であるが、研究会での千代田区の実施例等の研究を通して、ユーザ内部の仲介を受け持つファシリテーターの役割こそが、決定的に重要であると認識したしだいである。

(2) 業務要件 3 点セットの有効性

業務要件 3 点セットを作成することは業務を可視化することであることから、その成果物である業務要件 3 点セットは社内外の業務に関わる各種要請に対応するための基礎資料になる。たとえば、J-SOX や ISMS といった外部要請や業務手順の整備や業務改革(BPR)といった内部要請に対してである。

(3) 改善サイクルとしての作成プロセス(図 3.4)

千代田区の実践例でも As-Is3 点セット ⇒ 課題 ⇒ To-Be3 点セットというプロセスで業

務要件は作成されている。この流れから、最後の To-Be3 点セットは次回システム開発時には As-Is3 点セットとして使うというアイデアがでてくるのは自然である。

こうすると、As-Is に改善が行われて To-Be に至り、そこから As-Is に戻るという一種の改善サイクルを形成できるだけでなく、この改善サイクルは経営課題、業務遂行上の発生する各種課題、システム上の課題などを課題一覧という形で管理、改善対象とすることで、業務改善とシステム改善をセットにして扱うことができるという利点がある。

(4) 自分で書くことがすべての始まり

ユーザ自身の手による RFP 作成を推し進める上でのハードルのひとつに、業務要件の書き手である業務主体者への意識の問題がある。最初は受身で自ら手を動かすことへの抵抗が見られるが、ある時期から主体性をもって行動するようになる。研究の結果、自ら書くことが意識に変化をもたらし、持っている力を発揮するようになるという興味深い結論に達した。

自分で書くと、いろいろなことが見えてくる。
そうすると、人に説明できるようになるし、相手の言っていることも理解できるようになり、業務に対する責任や意識の共有が進むようになる。
結果、日常思っている業務課題や改善策が表出し、実際の改善として実を結ぶ。

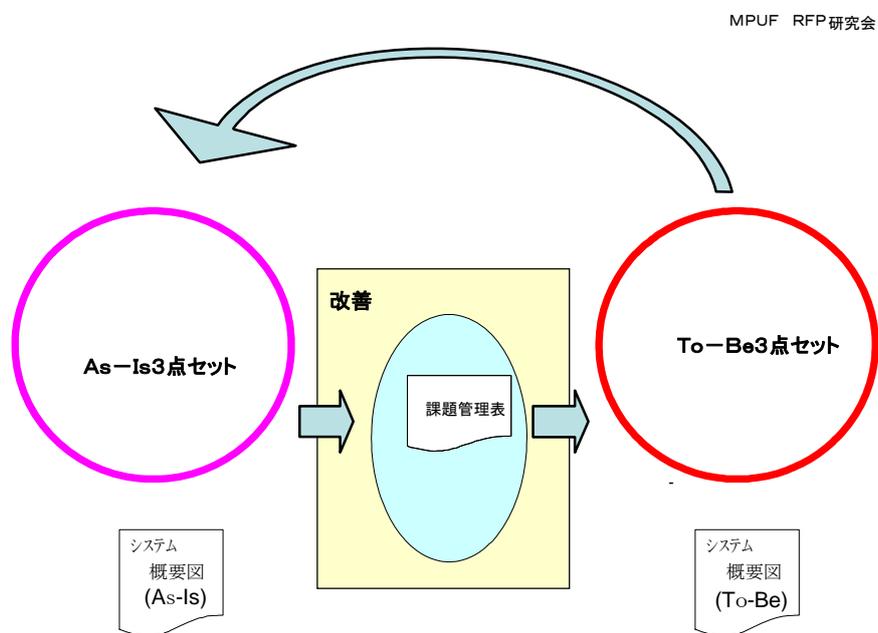


図 3.4 改善サイクルとしての作成プロセス

3.6. まとめ

- (1) RFP テンプレートは、ユーザ自ら RFP を作成する場合のツールとして有効である。
- (2) RFP 作成プロジェクトの重要成功要因として次の要因を挙げることができる。
 - 1) 業務主体者から構成された RFP 作成チーム
 - 2) ファシリテーターの存在
 - 3) 委員会業務理解と課題解決および合意形成の場の存在
- (3) RFP テンプレートに含まれる業務要件 3 点セットを使って業務の可視化ができる。
- (4) 業務要件 3 点セットによる RFP 作成プロセスは、業務とシステムの改善サイクルとしても有用である。

3. 7. 最後に

千代田区の実践例から、RFP 研究会が提案している RFP テンプレートを使うことによってユーザ自ら RFP を作成できることが確認できたが、まだ普及に向けてのスタートをきったところである。今後は、さらに普及活動を推進するために次の課題に対応していく必要がある。

- (1) 事例を含めた RFP 作成に関する情報の発信
- (2) 普及に関係団体との連携
- (3) 刻々と変化する IT 調達環境に対応できるようユーザとベンダの役割を含む作成プロセスとツールの継続的な見直し

参考文献

- [1] MPUF RFP 研究会, <http://www.mpuf.org/study/mpuf-rfp.aspx>
- [2] 情報サービス産業協会, 要求開発ベストプラクティスが示す成功パターンの調査研究, 情報サービス産業協会, pp3-17 (2007).
- [3] 情報サービス産業協会, 情報サービス産業における受注ソフトウェア開発の技術課題に関するアンケート調査結果, 2004, http://www.jisa.or.jp/bulletin/tech_problem2004.pdf
- [4] MPUF RFP 研究会, 簡単！RFP テンプレート, <http://www.mpuf.org/rfp/#about>
- [5] 総務省, 自治体事例紹介 ケース I 千代田区 2, http://www.soumu.go.jp/joho_tsusin/top/local_support/pdf/jirei_i.pdf

4. 要求定義における IT コーディネータの活用事例

NECソフト株式会社 加藤 久美子
NECソフト株式会社 岡野 勝俊
NECインフロンティア株式会社 渋谷 照夫

4. 1. はじめに

IT コーディネータ(以下、ITC と略す)のプロセスはユーザ経営者の経営戦略策定のサポートから、構築した IT 資産の導入、運用までユーザの立場で幅広く支援できるよう定義されている。経営の視点から自社に必要な IT 戦略を考え、システム要求を導き出す過程において、まさに「要求定義」の支援を請け負う役割を持つ人材と考えられる。

2006 年度からの本要求工学 WG の中での事例発表や議論の中でも、「要求定義」「ユーザ、ベンダ間のやりとり」のテーマについて語られる機会が多くあった。ITC のノウハウやスキルにはこれらのテーマについて有効に活用できる場面が多いのではないかと考え、また、社内には多数の ITC が存在することから、本テーマに焦点をあて、活用の実践例と今後の課題についての考察を行なった。

4. 2. ITC プロセスと要求定義

ITC とは、経済産業省推進資格で、「経営者の立場に立って、経営と IT を橋渡しし、真に経営に役立つ IT 化投資を推進、支援するプロフェッショナル」[1]である。

ITC の 5 つのプロセスと、ユーザ、ベンダの関係を図にしたものが以下の図 4.1 である。

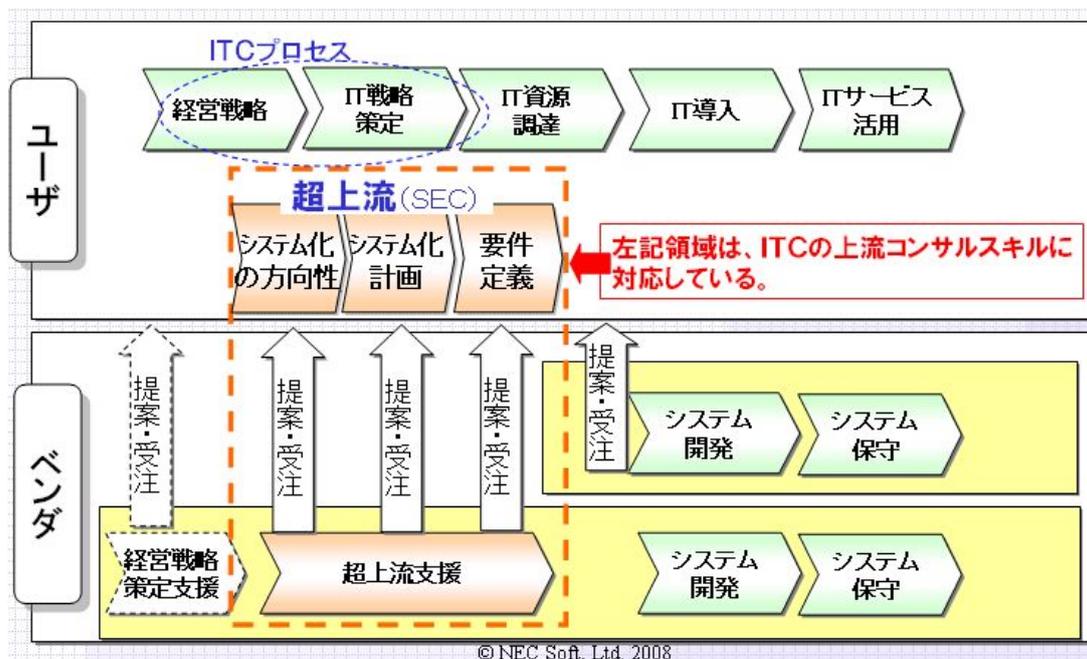


図 4. 1 ITC プロセスと超上流

『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～』(ソフトウェア・エンジニアリング・センター編)では、「システム化の方針」「システム化計画」「要件定義」を『超上流』のプロセスとして定義している[2]。ITC プロセスにこれをあてはめてみると、ITC の上流コンサルのプロセスがその部分に相当する。ITC プロセスでは「IT 資源調達」の中で、RFP の発行および提案書の受領を行なっていることから、本稿では、「経営戦略」から「IT 資源調達」の前半部分までを「要求定義」と位置づけることとする。

図 4. 1 で示したように、ITC プロセスは 5 つのプロセスから成っている。この 5 つのプロセスを実行する役割として、IT コーディネータ協会では、以下のような役割を定めている([3]より引用)。

- (1) 情報システムを基盤として組み込んだ経営戦略策定の支援。
- (2) 経営戦略を実現する情報化企画書の作成。
- (3) 情報化に必要なIT資源の調達を経営戦略企画責任者と共に行い、情報システムの開発および組織への導入時にマネジメント支援を行なう。
- (4) 運用時にはIT資源および経営活動全般における情報の活用度を定期モニタリングする。
- (5) これらのプロセスで問題があれば対策を提案する。

ユーザ側の立場に立ち、一つのシステムの構築、運用、活用に至るまでの広い範囲を支援範囲としていることがわかる。

また、IT コーディネータに求められる3つの能力は以下のように体系化されている([3]より引用)。

- (1) 各フェーズの活動を最適な方向に導く「プロセス&プロジェクトマネジメント」能力
- (2) 経営者をはじめユーザや IT ベンダーとの「コミュニケーション」能力
- (3) 活動の成果を「評価基準」に基づいてチェック、監理する「活動のモニタリング&コントロール」能力

4. 3. NEC ソフトの ITC 活用の取り組み

以下、当社での ITC の活用状況、および事例を紹介する。

社内の ITC 取得経験者(約 170 名)に対し、どのような役割・業務において ITC プロセスおよびスキルの活用が有効であるかを実感したか、アンケート調査を行なった結果を簡単にまとめたのが図 4. 2 である。

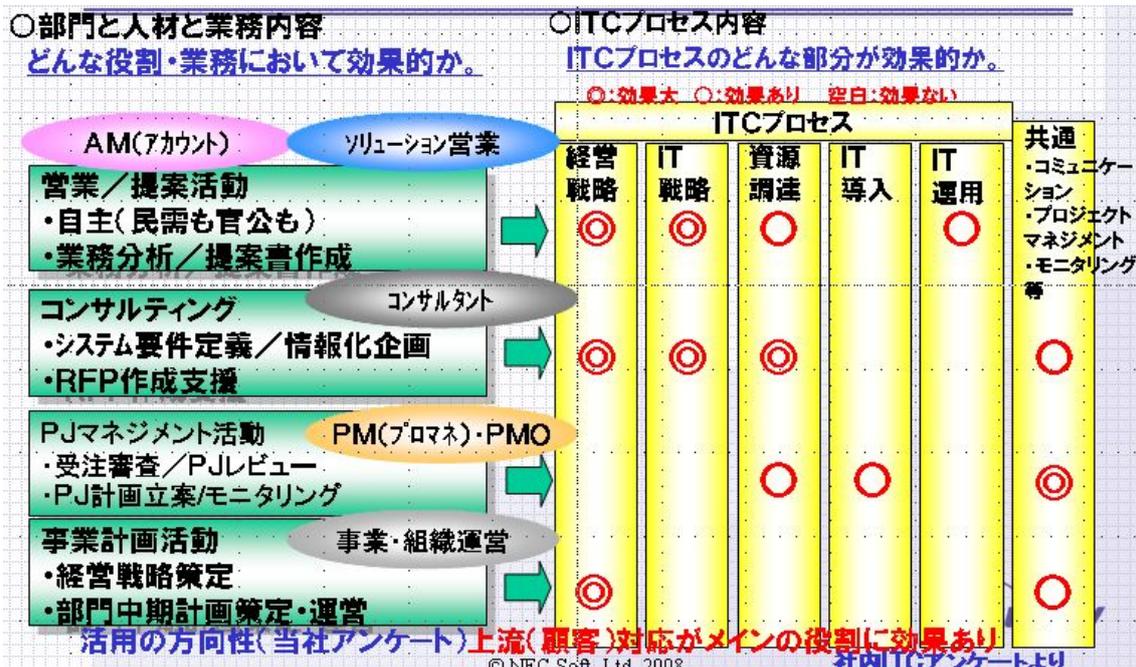


図 4. 2 社内 ITC アンケートの活用の方向性結果まとめ より

当社の ITC 取得経験者の業務内容は、大別して「営業/提案活動」、「コンサルティング」、「PJ マネジメント活動」、「事業計画活動」の 4 つに分けられる。

図 4. 2 は、それぞれの人材が ITC の 5 つのプロセス、3 つの共通能力の主はどこで効果的と認識しているかを大雑把に示したものである。AM(アカウント)は、「アカウントマネージャ」という職種でユーザ対応窓口を行なうマネージャのことである。

このアンケート結果で、効果大(◎)と主に捉えられているプロセスは、「経営戦略」、「IT 戦略」、「資源調達」であり、ユーザ対応がメインの役割ほど効果大と実感している、という結果が出ている。「PJ マネジメント活動および PJ マネジメント支援活動」という層では、ITC プロセスそのものよりも、プロジェクトマネジメントを含む共通能力面で効果的と実感されることが多くなっている。

また、具体的には以下のような効果も挙がっている。

- (1) ユーザ満足度が向上
 - －経営改革を織り込んだ提案
 - －システム要件の整理段階からのアプローチ
 - －業務課題を見据えた提案
- (2) コンサル、課題解決力を新たに備えることができた
- (3) ITC プロセスや PMBOK の実践によるプロジェクトマネジメントの成功
- (4) ITC イースト東京*1などの会社としての地域貢献

*1 ITCによるIT活用に関する地域(東京都東地区)企業支援を目的とした任意団体

特に、中小企業のユーザに対し、今までになかった経営改革を織り込んだ提案など、ユーザの高度化、多様化したニーズへの対応によるユーザ満足度の向上を図れたことが効果として現れてきている。

次に実際のプロジェクトまたは業務への活用シーンを以下の3つのパターンに整理した。

(1) 経営改革プロジェクト一括支援

ITC の 5 つのプロセス全体を一括支援する。

(2) 経営視点による部分コンサルティング型

典型的なベンダコンサルティングで、内部統制、セキュリティ、業務改革など一部サポート

(3) ITC スキルを活用した例

ITC プロセス、スキルを使って、提案書の充実、プロマネ、PMO 審査視点などの業務に活用した例。

以下、4. 4「ITC の活用事例」でご紹介する 2 つのプロジェクト事例は、次の二つである。

事例(その1): (1)プロジェクト一括支援

事例(その2): (3)提案書の充実

4. 4. ITC の活用事例

4. 4. 1. ITC の活用事例(その1)

一つ目の事例は、ある社団法人のプロジェクトである。平成 14 年夏、コンサルティング業務 (ITC プロセス)を提案、受注したプロジェクトである。

<事例概要>

- (1) 期間 第一期 平成 14 年 9 月～平成 18 年 3 月(事前アプローチは平成 14 年 6 月から)
第二期 平成 19 年 4 月～現在
- (2) 体制(主担当) ITC 1 名 が客先に入り込み、全期間支援。
(コンサルティング支援) ITC 6 名(最大時)
(開発支援) XXX 事業部
- (3) 概要 平成 14 年度 業務分析、戦略提言など
平成 15 年度 新システムイメージ策定
平成 16 年度 AsIs、ToBe、設計
平成 17 年度 新システム開発、導入
平成 18 年度 追加開発、運用支援
平成 19 年度 業務モデル改善検討

本事例では、一通りの ITC5つのプロセスが回り、サイクルとしては二巡目に突入したところとなっている。第一期稼働後2年間、システムを停止させることなく、トラブルなく順調に続いているプロジェクトである。

この事例のポイントは、一人の主担当が全体において継続して関わっており、まさに ITC のライフサイクルすべてを支援することができている、というところである。

この事例の当時 ITC プロセスは V0.9 であり、使用している用語、ツール等も当時のものであるが、プロセスを活用した主要成果物を以下に紹介する。

<経営戦略策定フェーズ>

「****コンサルティング報告書」にまとめてユーザへ報告。

- 1) 事業特性・業界特性の分析(5Force)
- 2) 現状認識・課題認識
- 3) SWOT(自身の強み弱み、外部の機会と脅威)分析
- 4) 保有する業務ノウハウなどの強み、業界の変化に関する弱みなどの分析を行い、主要成功要因を抽出
- 5) 新しい業務モデルの図式化。

特に 5Force は要求が出てくる視点を、「新規参入」「競合特性」「代替商品」「買い手(ユーザ)」「売り手(供給者)」という5つの視点で業界の特性を分析したものである。(図 1. 3)

SWOT 分析においては、当初、ユーザの支部を実際に訪問し、ヒアリングを行っていたが最終成果物としては、ユーザ自身による、分析、作成を行なっていただくこととし、支援を行なった。ユーザ自身が作成することで、要求の背景・根拠が明確になるという効果が見られた。

<戦略情報化企画フェーズ> (現:IT 戦略策定フェーズ)

「****コンサルティング報告書」にまとめてユーザへ報告。

- 1) 業務モデル分析、検討(DMM²、ActiveModeler)
- 2) 代替案の分析・検討
- 3) グランドデザインに関する提言のとりまとめ

DMM は、3 行×3 行の格子様式を用い、業務・システムを階層的に分解し、対象領域を明確にする。各格子に書かれた機能はさらに 3 行×3 行の格子様式に展開され、詳細化され、要求を機能に落とし込むことができる。全体が見渡せることが特徴である。(図 1. 4)As-Is、To-Be それぞれ作成する。

ActiveModeler は、ワークフロー可視化のツールであり、プロセス分析に使用した。当社オリジナルの分析方法であるが、業務の流れを鮮明にすることができた。

² DMM: Diamond Mandala Matrix 機能分析ツール

(参考) 業界特性分析(5Force)

NEC エンタープライズシステム		売り手(供給者)
新規参入	<p>過去10年間の売上増減率の推移を比較しているところ、業界平均の増減率は約10%であるが、NECは約20%と高い水準を維持している。</p> <p>また、新規参入者の増加傾向が顕著である。</p>	<p>主要な供給者として、ハードウェアメーカー、ソフトウェアメーカー、システムインテグレーション会社などが挙げられる。</p> <p>これらの供給者は、NECの競争優位性を脅かす可能性がある。</p>
競争特性	<p>競争の激化により、価格競争が顕著である。</p> <p>また、サービス品質の向上が求められる傾向がある。</p>	<p>供給者の競争優位性を維持するためには、技術革新とコスト削減が求められる。</p>
代替商品	<p>クラウドサービスの普及により、従来のオンプレミス型システムに代替されるリスクがある。</p>	<p>代替商品の出現により、市場の構造が変化する可能性がある。</p>
買い手(顧客)	<p>顧客のニーズが多様化しているため、カスタマイズされたソリューションが求められる。</p> <p>また、コスト削減と効率化が求められる傾向がある。</p>	<p>買い手の要求に応えるためには、柔軟な対応と高い技術力が求められる。</p>

図 4.3 5Force で業界分析を行なった例

(参考) 機能分析(DMM)

機能分析(DMM) - レベル1, 2								
1-1 ルーティング	1-2 業務運用	1-3 発生	2-1 申請書受付	2-2 受理番号付与	2-3 書類審査	3-1 申請内容のチェック	3-2 関連支店への依頼申請	3-3 ナンバーの購入
1-4 モニタリング	1-5 振替	1-6 収支	2-4 モニタリング	2-5 書類作成・受理(受取の場合)	2-6 データ入力	3-4 モニタリング	3-5 振込代行	3-6 重要税の申告
1-7 コントラクト	1-8 支払い	1-9 納品	2-7 納品管理(立寄金集計等)	2-8 振込代行	2-9 申請書類作成	3-7 完了通知書作成	3-8 決算処理と後援機関等の連携	3-9 地方税の申告
4-1 評価	4-2 印刷購入申込み	4-3 印刷受領(ハスラ含む)	1 振替	2 書類作成・受理	3 振込代行	4-4 評価	4-5 印刷受領(ハスラ含む)	4-6 支払
4-7 振込代行	4-8 印刷受領	4-9 印刷受領(一括)	6 モニタリング	7 代行政センター	8 税・手数料の納付	4-10 モニタリング	4-11 振・手配書の納付	4-12 支払
5-1 納品物の管理	5-2 データ入力	5-3 業者への受付票	7-1 振込代行	8-1 データ入力	8-2 納品	4-13 印刷受領(一括)	4-14 印刷受領(一括)	4-15 印刷受領(一括)
5-4 モニタリング	5-5 振込代行	5-6 納品	8-3 モニタリング	8-4 振込	8-5 入金	5-7 印刷受領(一括)	5-8 振込代行	5-9 納品
5-10 印刷受領(一括)	5-11 印刷受領(一括)	5-12 印刷受領(一括)	8-6 データ入力	8-7 入金通知書発行	8-8 入金チェック	5-13 印刷受領(一括)	5-14 振込	5-15 納品

© NEC Soft, Ltd. 2008

図 4.4 DMM で機能分析を行なった例

4. 4. 2. ITC の活用事例(その2)

二つ目の事例は、流通業 A 社の事例で、ITC フェーズでは「戦略情報化フェーズ」(現状分析・新システム構想)「情報化資源調達フェーズ」(RFP、ベンダ選定)に該当する事例である。

<事例概要>

概要 約2年にわたり継続訪問、SI 受注により信頼関係構築

ITC として RFP 作成支援

客先経営戦略、他社を意識した差別化提案を行い、受注

本事例では、信頼関係構築の土台作りに ITC プロセスをユーザに紹介している。それにより、スムーズな提案活動につながった事例である。日ごろのユーザ訪問および RFP 支援を行なう中で、ユーザが「業界 No1 になるためには経営改革が必要」という認識を持っていたことを感じとっており、要求をよりの確につかむことができていた。

その上で SWOT、BSC、5Force などの ITC のノウハウをフル活用し、提案書を作成している。

特に、図 4. 5 の BSC(バランススコアカード)に関しては、当社内の事業運営に使用しているツールであり、管理職以上の社員は業務で使用経験のある経営管理ツールである。使い方に精通していることも活用の要因となった。

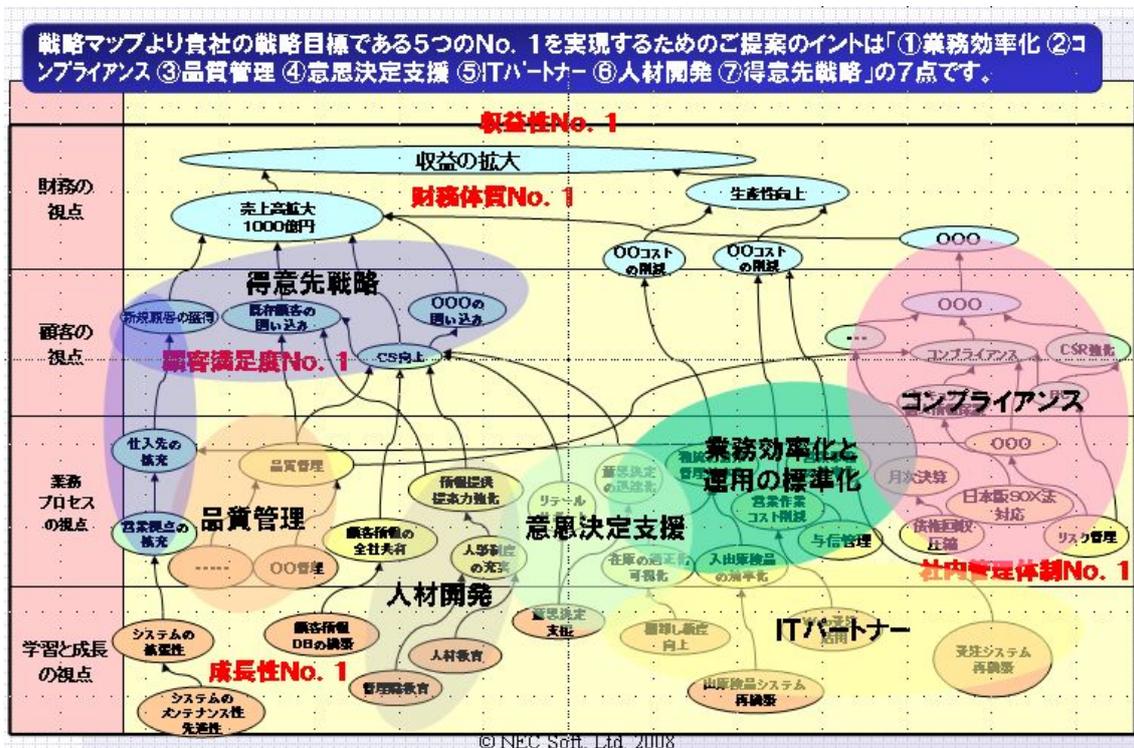


図 4. 5 バランススコアカードを活用した提案書の一部

以下に、提案書の目次例、提案書の一部を示す。(図 4. 6、4. 7)

実際の提案書目次	
1. BSCを用いた戦略的経営の実践 1-1. 経営改革のご提案 1-2. 経営改革の進め方 1-3. バランス・スコアカードの構成要素 1-4. バランス・スコアカードの5つの本質	4. システム化のポイント 4-1. 新システムの日進月歩の方向性 4-2. 事業戦略マップと新システム 4-3. 事業戦略を達成するシステム 4-4. 業務効率化と運用の態 4-5. コンプライアンス 4-6. 品質管理 4-7. 意思決定支援 4-8. ITパートナー 4-9. 人材開発 4-10. 得意先戦略 4-11. 事業戦略達成を支援する実現手段
2. 貴社を取り巻く事業環境 2-1. 市場環境 2-2. 法の改正 2-3. ...の影響 2-4. 月次決算の早期化 2-5. 業界の流通構造トピックス 2-6. 競争環境の分析 2-7. 業界の主な再編 2-8. 他業界における動向と企業戦略 2-9. 他業界の企業としての取組み事例 2-10. 他業界の営業改革事例 2-11. 貴社を取り巻く環境(まとめ)	5. ご提案システム概要 5-1. ソリューションのポイント 5-2. ソリューションマップ 5-3. DWH(データウェアハウス) 5-4. ワークフロー・アプリケーション開発基盤 (以下省略) 6. ご提案システムインフラストラクチャーご紹介 7. 情報セキュリティに関する弊社からのご提案 8. ご提案システム 開発方針 9. システム移行と教育について 10. スケジュール及び体制 11. 保守 12. 今回ご提案におけるご依頼事項 13. お見積に関する前提条件
3. 貴社次期基幹システム再構築の目的 3-1. 貴社課題 3-2. 事業環境のまとめ	6. IT構築提案
提案書全体 = 260p	戦略MAPから提案内容への落とし込み 今回提案範囲の明示

© NEC Soft, Ltd. 2008

図 4. 6 提案書目次例

図 4. 7 SWOT 分析例では、図 4. 6 の目次例にある「2-11. 貴社を取り巻く環境(まとめ)」、「3-1. 貴社課題」、さらに RFP 作成作業支援時の情報などから SWOT 分析を作成している。この分析が、ユーザより「当社のキーワードが整理されている」と評価された。

SWOT分析	
3-2. 事業環境のまとめ	
内部環境	外部環境
強み <ul style="list-style-type: none"> 自社でシステム開発する能力がある 特定エリア・専門分野でトップシェアを持つ 	機会 <ul style="list-style-type: none"> 販売後の安全・品質管理が必要である ※注活動だけでなく、アフターサービスも充実させる必要がある SCM全体の最適化が求められる メーカーの生産型や輸入型にバラつきがあり、在庫の原因となっている 〇〇〇の改正でマーケットが分割された
弱み <ul style="list-style-type: none"> 営業拠点が地域に限定されている 営業利益率が低い 販売費及び一般管理費が高い 	脅威 <ul style="list-style-type: none"> 価格競争が激しくなっている 共同購買により低価格を採られる 各種〇〇費削減政策が実行されている 〇〇〇見直しによる経費削減が実行されている 大手〇〇〇から新規参入がある 総合商社が参入している

図 4.7 SWOT 分析例

これらの ITC ノウハウをフル活用した分析手法により、ユーザ要求を結果として反映できた提案書を作成できた。その結果、競合他社提案との明確な差別化提案となり、ユーザより高評価を得ることにつながったと考えられる。

4.5. 課題

4.5.1. 個人の活用から組織的な活用へ

ここまで、当社の ITC を活用した取り組み例を中心に、活用例を挙げてきた。

しかしながら、プロジェクトで必要なときに ITC の有資格者が常にそのノウハウ、スキルが有効活用できる立場でプロジェクトに参加できているかという、現実的には実現できないことも多い。

また、当社の中でも、問題化したプロジェクトの要因として依然として「途中で作業範囲を拡大したことが要因」とされるプロジェクトは存在する。社内の施策として、品質会計制度³、PMO、PN (プロジェクトナビゲータ)制度⁴、などさまざまな施策が功を奏し、また、生産革新活動も進んでいるが、主に設計以降の工程やプロジェクト管理に関する施策であり、さらに上流(=超上流)についての施策の充実を考える必要がある、との声も出てきている。

よって、ITC の活用を個人の有資格者の活用、という観点から視野を広げ、そのスキル、ノウハウに着目し、もっと広い範囲で活用する、という視点で考えてみたい。

4.5.2. 要求定義で求められるノウハウ、スキル

既に述べたように、ITC のスキル、ノウハウと要求定義で求められるスキル、ノウハウは共通するところも多い。

その観点で、我々が双方に必要とされるスキルをまとめた表が以下である。

表 4.1 必要とされるスキル(ITC、要求定義)

	ITC	要求定義策定者
①経営環境分析	◎	○
②業種ノウハウ	○(得意業種でよい)	○(得意業種でよい)
③業務モデリング	◎	◎
④ITアーキテクチャ	△	△
⑤ファシリテーション技術、コミュニケーション・合意形成力)	◎	◎
⑥プロセス&プロジェクトマネジメント	◎	○
⑦モニタリング、評価技術	◎	△

³ コーディング工程までに作り込んだバグを負債とみなし、レビュー/テストによるバグ抽出によりこの負債を返済し、負債が0となった時点で出荷する制度

⁴ プロジェクトマネージャ(PM)がプロジェクト(PJ)管理に専念できるように、PJに常駐しPMを補佐する役割

(1) ユーザ支援の視点をベンダ側視点へ変換し活用する

ITC のプロセス自体はユーザ側の経営者の立場に立ち、支援を行なう役割を規定されている。よって、ユーザ側のプロセスであり、考え方やプロセスをITベンダが活用するためには、変換、具体化がある程度必要となる。これは、ノウハウのプロセス化を考える際に、重要な考慮点となる。

だが、ITベンダの立場であっても、ユーザの立場の支援プロセスを理解していることで、ユーザにとってのより良い提案につながるなど、経営課題を一緒に発見、解決を図る方向性を導くことができるようになり、お互いに Win-Win の関係を構築できることが期待できる。また、要求定義後に受注した案件であっても、要求定義でのプロセスなりノウハウなりがわかっていると、その時点で必要なことが明確になっているか、設計までに不足している確認事項は何か、といったポイントが会得しやすくなり、プロジェクトの失敗のリスクが少なくなる、というメリットも考えられる。

(2) 業界標準プロセスを実践するために ITC プロセスを適用する

プロセスという視点では、世の中の動きとして共通フレーム 2007 や経済産業省のモデル契約の提示など、「超上流」に対する考え方が広まりつつあり、認識されてきている。そのプロセスとともに、実業務にどういった手法で活かすのかを考えたとき、ITC有資格者などが個人で取組んできたITCプロセスやノウハウの情報共有範囲を組織に広げていくことも活用範囲を広げることと言えるであろう。

(3) 教育や人材育成に活用する

また、スキルとして「ファシリテーション」や「マネジメント」は、ユーザに対してはもちろん、組織の中でも幅広く活用できるスキルである。教育や人材育成という観点でも、ITC のスキル、プロセスを活用し、教育内容の見直しやキャリアパスの明確化、等により、技術者のスキルアップにつなげていくことも可能である。

4. 6. おわりに

以上、説明してきたことから分かるように、結論として「要求定義力」を上げるために、ITC の「プロセス」「スキル」等の視点での活用することが、一つの有効な方法となり得ると考えられる。今後も 4.5 で述べたいいくつかの課題を中心に解決を図りながら取り組んでいきたい。

参考文献

- [1] 特定非営利活動法人 IT コーディネータ協会, IT コーディネータ(ITC)プロセスガイドライン, Ver1.1 (2006).
- [2] 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, 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, オーム社 (2006).
- [3] IT コーディネータ協会, <http://www.itc.or.jp/>.

5. 要求開発プロセスの分担制による要件品質向上事例

NEC ネクサソリューションズ株式会社 小池 輝明

5. 1. はじめに

5. 1. 1. 要求開発プロセスと問題点

要求開発プロセスは、要求の獲得、分析、文書化、妥当性確認のフェーズに分けられるが、開発規模が大きくなるとこのプロセスを一人の作業者によって実行することは困難になる。

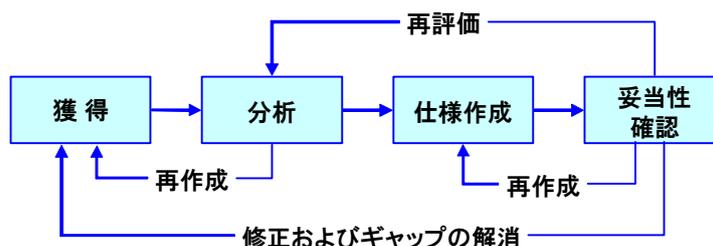


図 5. 1 要求開発の繰り返しプロセス

この理由として、要求開発プロセスの上位にあたる顧客要求の獲得に作業比率をかける傾向により、分析以降のプロセスにかかる作業リソースが不足しがちになることがあげられる。本来必要な、分析から獲得、妥当性確認から文書化、妥当性確認から分析または妥当性確認から獲得という繰り返しプロセスが機能しなくなり、要求品質が低下する。

要求品質が低下することにより次のような弊害がある。

- (1) 後工程での修正工数の増加とそれに伴うテスト工数の増加、
- (2) プロジェクトが進むにつれて作業工数が指数関数的に増大
- (3) 顧客要求が適確に捕らえられていないために、導入後のシステムに対する顧客満足度が期待できない
- (4) 修正要求が出される可能性が高くなる

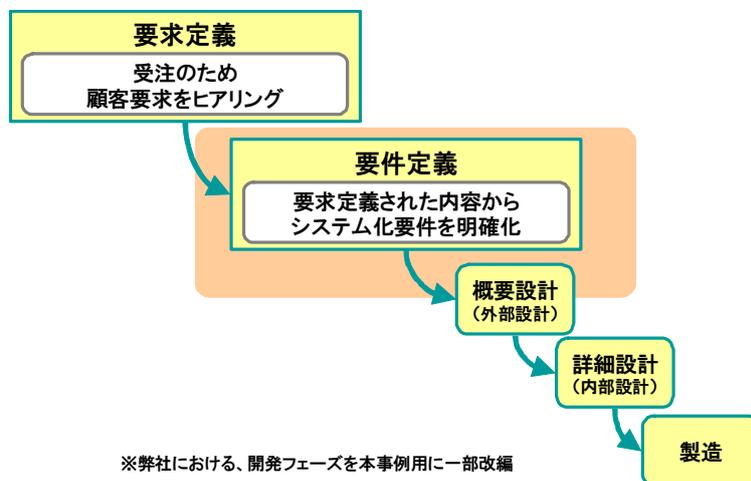
5. 1. 2. 事例の問題点

ここで紹介する事例は、こうした危機を認識し始めたプロジェクトにおいて要求分析フェーズでどのように回避し、後のフェーズへの問題の持ち込みを回避する取り組みである。この取り組みでは計画的な部分とアドホック的な部分が混在しているため、要求開発のプロセスの視点から整理して紹介する。

5. 1. 3. 「要求定義」と「要件定義」

本事例の中で「要求定義」と「要件定義」の2つの用語を用いる。要求定義は、受注のために顧客要求をヒアリングするフェーズで受注のための見積もり内容を明確にする。要件定義は、要求定義のアウトプットを元に、システム化要件を洗い出す。

要求開発のプロセスで言えば、要求定義は獲得と分析、要件定義は要求の文書化と妥当性確認が該当するが、実際には明確な規定はない。



※弊社における、開発フェーズを本事例用に一部改編

図 5.2 要求定義と要件定義の位置づけ

5. 1. 4. 問題解決のための体制

今回の開発プロジェクトは受注した部門で着手していたが、プロジェクトのアラームがあがった時点で生産技術の支援部門に支援依頼が出された。そのため、開発プロジェクトの外部からメンバが支援を行った。

5. 2. プロジェクトの概要

5. 2. 1. 要件定義の長期化による後工程の先行着手

この開発プロジェクトは 3 ヶ月で要件定義フェーズを終了し、概要設計に移行する予定であった。開発規模の見積り終了し、詳細設計以降の開発作業を担当する開発パートナーとの調整も終了し、詳細設計のスケジュールに合わせて必要なリソースが確保されていた。しかし、3 ヶ月を過ぎても要件定義は終了せず概要設計が並行で開始され、6 ヶ月が経過する時点でもまだ収束しない状況であった。

5. 2. 2. 顧客対応と開発パートナーの板挟み

要件定義の長期化に伴い開発チームは、顧客対応と開発パートナーの板挟みになっていた。要件定義が長期化することで顧客側からは次々と要件が出され収束の兆しが見えず、そのために概要設計も遅れ始めていた。

開発パートナーは当初のスケジュールに合わせて開発メンバを確保していた。スケジュールの変更が発生すると再度同じメンバのアサインが行われるかどうかの保証はない。しかも、概要設計も遅れているために発注条件の概要設計書を期日まで渡せる保証もなかった。仮に先行発注しても、開発パートナーの開発者は作業待ちになり、その費用もムダになりかねなかった。

5. 2. 3. 開発プロジェクトの規模

当初正確な開発プロジェクトの規模は明確に提示されなかったが、画面数などでおおよそ下表のような規模が知らされていた。このような規模からファンクションポイントを概算すると、4,000FP～5,000FP になることがわかっていた。

表 5. 1 開発規模の概要

機能種別	想定機能数
画面入力	100
画面照会	20
帳票出力	200
パッチ更新	20

5. 3. プロジェクトの問題点

5. 3. 1. 要件定義フェーズの長期化

要件定義フェーズの長期化における問題は次の 3 点に集約される。

(1) 開発スケジュールの遅れ

上流工程である要件定義が遅れると次工程以降に波及的に影響する。しかし、要件定義が遅れたスケジュールを次工程以降で取り戻すことは困難である。

(2) 要件定義内容の肥大化

顧客からみると実に都合の良い状態である。いつまでも要件定義を続けていけば、顧客は要求をいつまでも聞いてシステムに反映してくれるものと思いきや、要求は時間とともに変化していくので、時間をかければかけるほど当初の要求から変わっていくことになる。

(3) 開発コストの増大

スケジュールが遅れて、要件定義が肥大化すれば、開発コストが増大するのは火を見るよりも明らかである。スケジュールと同様に要件定義で増大した要求を次工程以降で取り戻すことはできない。

5. 3. 2. 五月雨式開発のリスク

詳細設計以降を開発パートナーに発注することを決めていたこのプロジェクトは、開発パートナーからの発注の遅れを指摘され非常に厳しい状況におかれていた。そこで、開発チームは要件が確定した部分から五月雨式に発注することになった。

要件をマスタ系のように変更が少ない内容を静的な要件とし、ビジネスの変化に伴い頻繁に変わることが予想される要件を動的な要件と分けた場合、静的要件から発注する場合と、動的な要件から五月雨式に発注する場合は、その後のリスクはまったく変わってくる。

五月雨式発注の場合は、確固たるポリシーのもと、リスクを最小限にしなければならない。

5. 3. 3. 後工程で発生する問題

不適切なポリシーに基づいた五月雨式発注はいくつかの致命的な事態を招く。

まず、要件の正規化が行われないことによる、ムダな要件が次工程に混入してしまうリスクがあげられる。要件を俯瞰し全体の視点から要件間の整合性やビジネスプロセス視点での重要性の判断が不在のまま次工程に移行すること、要件の優先順位付けがないためにスケジュール遅延時の対策立案が遅れる、もしくは、適切に行われない可能性がある。

この結果、不要な要件を満たすために限られた工数が割り当てられることになり、工数超過と不要なバグを発生させることになる。また、不要な要件であっても実装されればテストをする必要がありテスト工数も増大する。

ムダな要件の次工程への混入による問題は、スケジュール、費用、品質と全てに渡ってマイナスの影響を及ぼす。ムダな要件を次工程に混入させないためには、要件の品質を向上させる必要がある。

5. 4. 問題解決の方針

5. 4. 1. 要件品質の作り込み

顧客のビジネスプロセスへの貢献がなければ、システムの存在価値はない。その前に、論理的に説明できる機能が実装されていなければならない。前者は魅力的品質、後者は当たり前品質に位置づけられる。魅力的品質は、顧客のビジネスプロセスをどれだけ理解しているかにもよるため、ここでは論じないこととする。

本事例では、当たり前品質をどのように維持するかに焦点をあてた方針を立てた。ソフトウェアの品質属性については既に多方面で論じられているのでここでは割愛する。

(1) 要件定義フェーズにおける品質

要件定義における成果物間で論理的な矛盾をなくすことが要件定義の品質を向上させる。例えば、「図 5. 3 要件品質について(1)」のように要件定義が行われたときの品質を考えてみる。

業務フロー図を作成し、業務が業務一覧に反映される。業務フロー図の機能が全て設計されて、その機能が機能一覧に反映される。業務一覧と機能一覧の間においては、業務に機能がマッピングされている必要がある。これらのことが間違いなく行われたとき、整合性が維持され、要件品質が確保されていると言える。

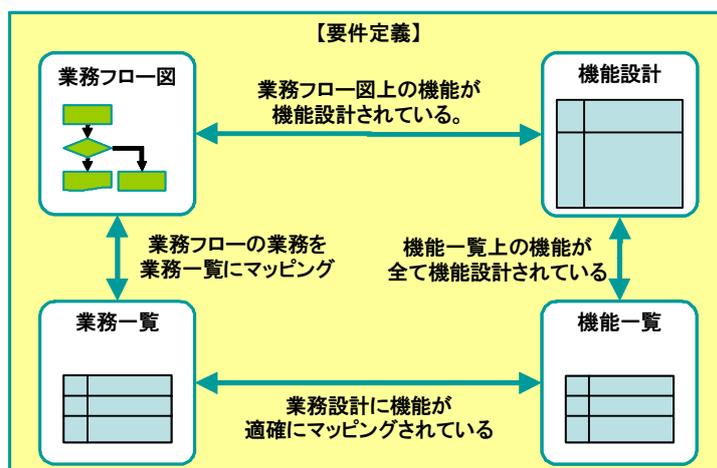


図 5.3 要件品質について(1)

(2) 要件定義のアウトプット

要件定義のアウトプットは概要設計とテストのインプットになる。要件定義で書かれた要件は設計されなければならないし、要件定義で定められたものは顧客が受入テストでの確認対象となるからである。

また、要件定義で出された業務一覧や機能一覧はプロジェクトマネジメント上正確でなければならない。例えば、機能一覧をプロジェクト管理要素として用いる場合、機能間関連を考慮してWBSを作成し、リソース配分とスケジュールを作成する。機能一覧で項目の重複があれば余分な機能の作り込みによりムダにリソースを浪費することになり、機能が足りなければ検収されない(つまりは売上げにつながらない)という事態になりかねない。以上のように、コスト、スケジュール、リソース管理する際に、項目一覧は重要な要素になる。

要件定義のアウトプットがその後のプロジェクト全体に与える構図は「図 5.4 要件品質について(2)」を参照欲しい。

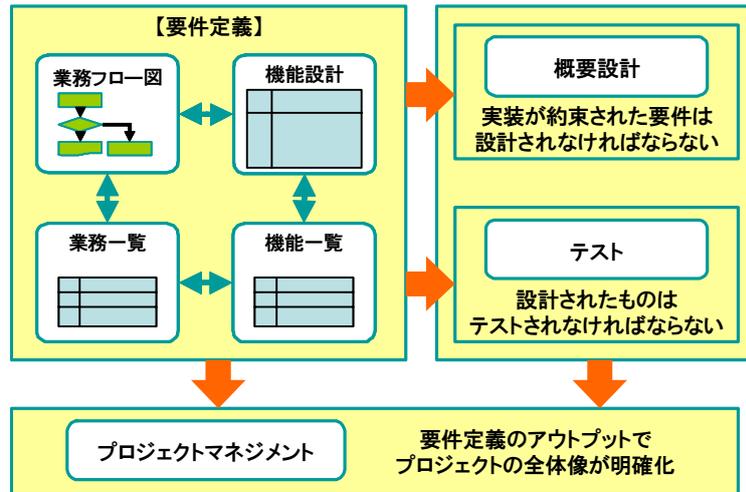


図 5. 4 要件品質について(2)

このように要件品質は重要な意味を持っている。このプロジェクトにおいては、要件定義フェーズにおける要件品質を向上させ、プロジェクト全体へ要件定義フェーズの誤りを持ち越さないようにする必要があった。

5. 4. 2. 次工程へのスムーズな移行

通常、品質を確保するために徹底したレビューによるバグの洗い出しが行われる。しかし、既にスケジュールを 3 ヶ月超過しているプロジェクトに対してレビューやインスペクションを徹底して品質を確保してもスケジュールの遅れを取り戻すことは期待できず、むしろスケジュール的には遅れがさらに増大する可能性がある。そのため、要件定義フェーズを一刻も早く収束させ、次工程へのスムーズな移行が望まれる。

5. 4. 3. 後工程で発生する問題の歯止め

要件品質を維持しスムーズに概要設計フェーズに移行できたとしても、後工程で同じ事態を招くことになると、取り組みの努力が水泡に帰することになってしまう。そのため、後工程で新たな要件が発生した際に同様の問題が発生しないように歯止めをしておく必要がある。

そこで、2 チーム制にして、各チームの役割を明確にルール化しておくことで、後戻り発生の抑止になるのではないかと考えた。

5. 5. 2 チーム制の導入

5. 5. 1. 目的別チーム編成

現在のプロジェクトの成果物の品質を向上させつつ次工程への移行をスムーズに行い、さらに、品質向上のための工数のオーバーヘッドを最小限にしつつ、要件定義ドキュメント作成作業の担当として、仕様作成チームを編成することとした。現在の作業を進めている開発チーム

は要件獲得チームとし、2チーム制にすることとした。

5. 5. 2. 要件獲得チームと仕様作成チームの役割

要件獲得チームと仕様作成チームの要求開発プロセスにおける分担を「図 5. 5 要件開発プロセスにおける分担」に示す。要件開発プロセスでの、要件の「獲得」と「分析」は要件獲得チームが担当し、仕様作成チームは「仕様作成」と「妥当性確認」を担当する。要件獲得チームは顧客対応に注力し、仕様作成チームは獲得された要件の文書化と品質の確保に注力する。要件開発プロセスにおいては、各フェーズからのフィードバックやフィードフォワードが絶えず発生するため、インタフェースの取り方とルールを決めておく必要がある。

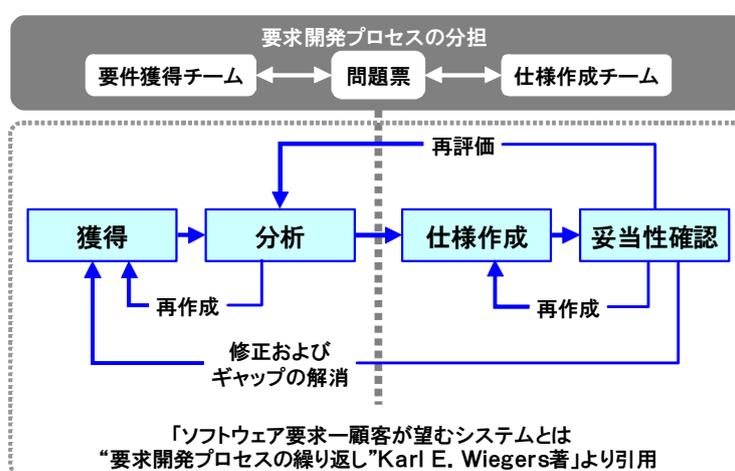


図 5. 5 要件開発プロセスにおける分担

5. 5. 3. ツールの導入

仕様作成チームが効率的に作業を遂行できるように、要件仕様作成ツール(以下、ツール)を導入することとした。このツールは主に以下のような機能を備えている。

(1) 機能名称不一致の検出

業務フロー図に記入されている画面名、帳票名などが機能一覧に存在しない、または、機能一覧に記入されている機能名が業務フロー図に記載されていない、という不一致を検出できる。また、機能名称の類似や重複の検出が可能。仕様作成チームはこのような不一致を検出した際には、問題票に記入して、解決方法を要件獲得チームに問い合わせることになる。

(2) 業務フロー図の作成ルールの統一

業務フロー図の作成上のルールに合致しない記載を抑制する。例えば、画面入力アイコンからはなんらかのデータストアにつながっていなければならないため、画面入力から帳票出力に直接フロー線は引けない。これは DOA 的手法によりデータの概念モデルの作成を意図しているための機能である。仕様作成チームは、このルールに反した作業はできないので、問題票に記入し、要件獲得チームの判断を仰がなければならない。

5. 5. 4. 要件獲得チームと仕様作成チームのインタフェース

要件獲得チームは顧客の打ち合わせ調整や一部先行開発のための作業があり、仕様作成チームとの作業進捗をいつも合わせることはできないため、お互いの作業がオーバーヘッドになりかねない。また、要件品質を確保する必要もあったが、これも2チーム制によるオーバーヘッドを発生しかねないリスクがあった。

そこで、要件獲得チームと仕様作成チームのインタフェースは問題票を使用し、共同作業が効率よく進められるようにした。問題票には、要件定義での各ドキュメント間の整合性を仕様作成チームがレビューした結果発見されたエラーを記録し、要件獲得チームとのレビューを通して、再度ドキュメントに反映させる際に使用した。「問題票」を「かんばん」のように利用し、作業進捗・スケジュールが両チームへの影響を最小にした。

具体的には、以下の4つの作業が絶えず回るようにした。(図5.6参照)

- (1) 要件獲得～分析: 要件獲得チームは、顧客要求のヒアリング結果をExcelで記述する。
- (2) 分析～仕様作成: 仕様作成チームは、Excelに書かれた内容を、ツールに入力する。
- (3) 仕様作成～妥当性確認: ツールへの入力時に、要件間の矛盾があると入力できないか、警告が発せられるので、その内容を問題票に記述する。
- (4) 妥当性確認～分析または獲得: 要件獲得チームは問題票に記入された問題点の解決方法を問題票に記入する。仕様作成チームは、問題票に記入された解決方法により、ツールへの入力を行う。

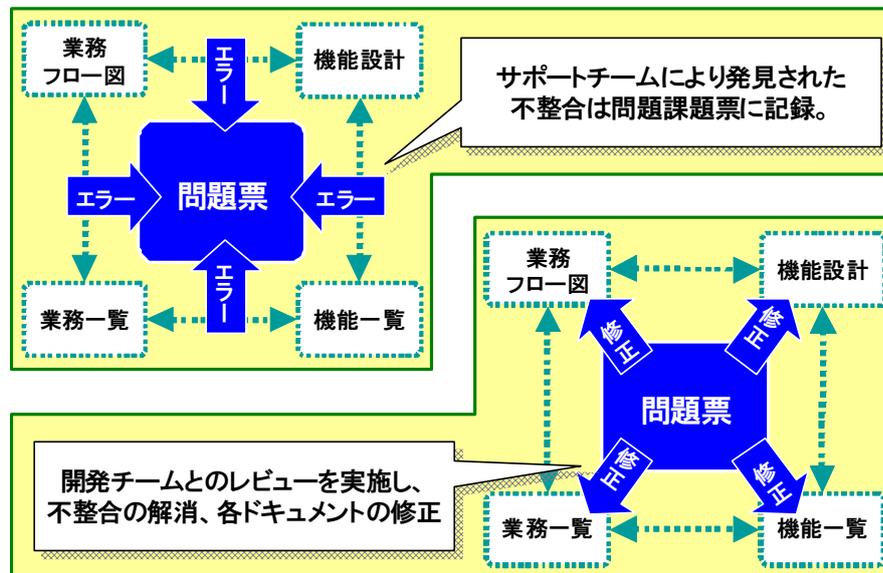


図 5. 6 問題票による要件品質の確保

この問題票の記録から、品質に影響のある内容がどの程度発見できたかを分析できるようにしておくことで、後工程へのリスク削減を定量化できるようにした。

5. 6. 実施作業内容

5. 6. 1. エラーの洗い出し

問題票に記録されたエラーについて分析をしてみる。実際に記録されたエラーの一部を「表 5. 2 エラーの洗い出し(機能名称の不一致の一部)」と「表 5. 3 エラーの洗い出し(業務フローの一部)」に示す。

表 5. 2 エラーの洗い出し(機能名称の不一致の一部)

No.	種別	問題
8	名前・ID 違い	【業務フロー】2-1-24 にて"———"の画面番号違い
9	重複	【業務フロー】2-1-24 にて画面番号 402 が別名で 2 個存在している
13	重複	【業務フロー】2-1-12 にて画面番号 226 が別名で 3 個存在している
14	名前・ID 違い	【業務フロー】2-1-12 にて画面番号 673"———一覧"は、637 で登録済み
57	名前・ID 違い	【ビジネスフロー】(2-1-12)「部品発注書」該当 No なし
58	名前・ID 違い	【ビジネスフロー】(2-1-12)「製造番号票」該当 No なし
59	名前・ID 違い	【ビジネスフロー】(2-1-12)「(673) ———」No 違い
62	名前・ID 違い	【ビジネスフロー】(2-1-17)「(618) ———」の ID が違う
63	名前・ID 違い	【ビジネスフロー】(2-1-18)「———確認票」該当 No なし
64	名前・ID 違い	【ビジネスフロー】(2-1-18)「———作成」該当 No なし
65	名前・ID 違い	【ビジネスフロー】(2-1-18)「(470) ———」「(585) ———」の表記名違い
66	重複	【マトリックス】管理番号 539:テーブルは割当明細となっているが、エンティティには本社割当明細と拠点割当明細の 2 種類ある
68	重複	【マトリックス】管理番号・470・478・482・485:同じ画面が 2 つある

「表 5. 2 エラーの洗い出し(機能名称の不一致の一部)」に関しては、比較的軽微なエラーなので、仕様作成チームで判断できる内容が多く、最終的な確認のみ要件獲得チームに依頼する程度で対応が可能であった。軽微とはいえ、こうした内容が後工程に混入すると似たような機能が違う名前で実装されるなど、リソースの浪費につながる可能性があり、決して軽視されてよい内容ではない。

表 5.3 エラーの洗い出し(業務フロー)の一部

No.	種別	問題
1	ルール違反	【業務フロー】仕切り線が引かれている場所があるが、どう対応するか
2	ルール違反	【業務フロー】四角にて囲まれている部分があるが、どう対応するか
4	ルール違反	【業務フロー】画面入力系 → 帳票の矢印が引けない
5	ルール違反	【業務フロー】説明のないオブジェクトが存在している
6	ルール違反	【業務フロー】2-1-5にて"受注データ更新"がデータとして登録されているが、これは処理ではないのか
7	ルール違反	【業務フロー】帳票 → 画面入力系の矢印が引けない
10	ルール違反	【業務フロー】2-1-25にてフロー図上では画面164“———検索照会”からデータの変更がされるのだが、これは入力または更新系では？
11	ルール違反	【業務フロー】手作業 ⇄ 画面入力系の矢印が引けない
16	ルール違反	【業務フロー】画面入力系 → 画面入力系 の矢印が引けない
17	ルール違反	【業務フロー】画面入力系 → 画面入力系 の矢印が引けない
18	ルール違反	【業務フロー】オブジェクト → テキスト の矢印が引けない
19	ルール違反	【業務フロー】太い矢印があるが、どう対応するか
60	ルール違反	【ビジネスフロー】(2-1-14)「(586)支払入力」から人間系「支払」へ線が引けない
61	ルール違反	【ビジネスフロー】(2-1-14、15)「受注データ」から「受注データ」へ線が引けない
79	ルール違反	【ビジネスフロー】(2-1-20)入力から手作業への線が引けない

一方、「表 5.3 エラーの洗い出し(業務フロー)の一部」では、要件内容を把握している要件獲得チームでないと判断できない部分が多かった。これらの内容に関しては、要件獲得チームと仕様作成チームのレビュー時に対応方法が決まるといことが多かった。

5.6.2. 要件品質の向上

本事例での要件品質を、今回の作業で検出・修正したエラーがどのような規模であったかを示すことで把握することとする。

本事例のシステム機能は画面入力、画面検索、帳票出力、バッチ更新の4つに分類されるので、各分類でどれだけエラーが発見されたかを検証してみる。最もエラー発見の多かったのは機能数も多い帳票出力系であり、もっとも少ないのは画面参照系機能であった。要件として帳票機能が多いのは知らされていたが、機能あたりのエラー数でみても、比率も非常に高い(図 5.7 機能数とエラー数の関係参照)。

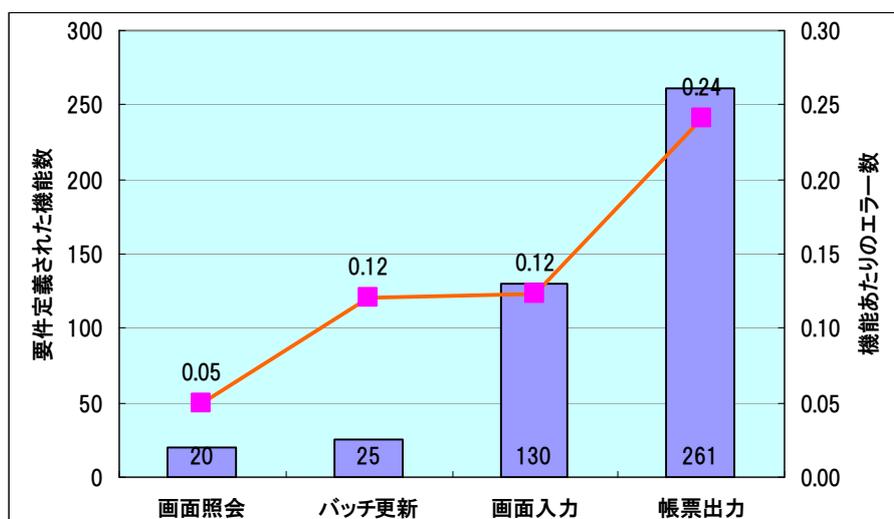


図 5.7 機能数とエラー数の関係

5.6.3. 効果

今回の事例の効果については、当初目標だった下記の事項が達成できたと言える。効果の概要については、「図 5.8 効果の概要」を参照

(1) 要件品質の作り込み

要件上のエラーは 83 機能に及び、その機能の開発規模は 540FP にのぼった。この規模はシステム全体の 13%に相当する。これらの機能について要件定義フェーズで発見できたことは品質向上に貢献できたと言える。

(2) スムーズなフェーズ移行のサポート

仕様作成チームを編成してエラーの除去、ドキュメント作業を行い、要件獲得チームのアウトプットの品質向上に貢献したが、プロジェクト予算を圧迫していたのでは意味がない。そこで仕様作成チームの作業工数を計算してみると、約 3 人月であることが判明した。今回は 3 名で構成していたので約 1 ヶ月で作業を終了することができた。

また、2 名はオペレータレベル作業だったため SE 作業単価と比較して約 1/3 の安いメンバをアサインできたため、プロジェクト予算全体へのインパクトは最小限におさえつつスケジュールへの影響も最小限に留めることができた。

(3) 後工程で発生する問題の歯止め

仕様作成チームでの作業は随時、「作業ルール」としてドキュメント化し、要件獲得チームにも徹底を促した。このルールがあることで、追加要件が発生した場合には、要件品質を維持したまま対応できると期待できるので、歯止めとしての効果は十分に期待できる。

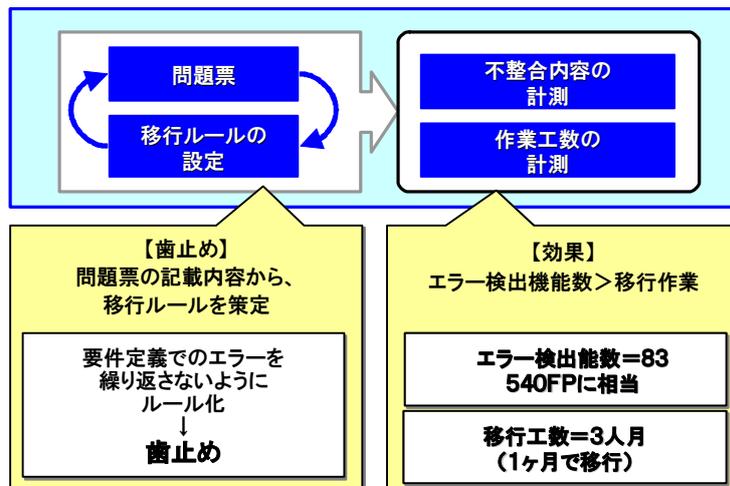


図 5.8 効果の概要

5.6.4. 制限事項

今回の事例に関して効果があがるための条件として下記事項があげられる。

- ・ 2 チーム制を導入するのに見合うシステム規模であること。
- ・ ツールは、要件間の矛盾を自動的に検出できオペレーションが容易であること。
- ・ ツールの導入コストの費用対効果を算定済みであること。

5.7. 考察

5.7.1. 要件品質について

要件の品質に関しては、品質特性で機能の品質を捕らえる「当たり前品質」と、ドキュメント化された要件定義で顧客がどの程度満足できるか「魅力的品質」が考えられる。しかし、要件定義段階ではシステムが動作しているわけではないので、正確に品質を計るのは難しい。

要件定義でのアウトプットが、開発プロジェクトに対してもっとも重要であることを考えると、論理的な誤りが排除されていることが望ましい。品質の定量化は困難であるにしても、エラーを発見された機能の数やその機能の規模(ライン数、ファンクションポイントなど)を計測しておくことは重要である。

要件定義が収束しないのであれば、時系列的に獲得要件数をプロットして傾向をつかむことも重要であると思われる。

5.7.2. 仕様作成チームの貢献

一般に、品質保証活動とは要件獲得チームのアウトプットに対するレビューという形で進められる。しかし、品質保証活動が要件獲得チームとの距離をおけばそれだけ工数のオーバーヘッドが発生し、本来の目標を脱し易い。

本事例で、仕様作成チームを品質保証チームとしなかったのは、第三者がレビューしてエラ

一を発見しつつ、その内容の対処まで行うことを当初から目指していたからである。要件定義フェーズは顧客とのインタフェースが最も濃密である必要があり、短時間で終了する必要がある。そこで、要件のドキュメント化や検証作業に工数が割かれると、要件開発プロセスの周期が長くなってしまふ。

今回は作業範囲外であったが、要件の検証にはアーキテクトが参画して、技術的な見地からの検証も同時に行うとよりよい結果につながったと思われる。

5. 7. 3. 効果計測

機能数や機能の定量化について最初から計画して取り組んだため、効果も定量的に示すことができた。仕様作成チームのように第三者チームをプロジェクトに招聘することがなくても、要件定義フェーズでの定量的な計測を計画しておくことと下記のような効果があると思われる。

- (1) 時系列的な獲得要件数の把握による、要件の収束状況の把握
- (2) 品質の定量化による、要件定義の進め方に関するモニタリング
- (3) 獲得要件を規模として把握し、開発リソース、予算の確保

参考文献

- ・ K. E.Wiegers, ソフトウェア要求, 日経 BP 社, 2003.

6. 上流工程における機能分析方法と顧客満足度評価方法の御紹介

東芝ソリューション株式会社 村田 尚彦

6. 1. はじめに

上流工程において利用する当社独自の二つの手法について紹介を行なう。

最初に開発対象となるシステムの機能分析を行ない、システムの規模を算出する手法を紹介する。

次に、開発するシステムの顧客の要件から顧客要求をどのくらい満たすことができるのかを評価し顧客満足度を算出する手法を紹介する。

6. 2. 「PSER」分析の紹介

6. 2. 1. PSER による機能分析方法

顧客の要件を定義する上流工程では、開発するシステムの機能がおぼろげに見えてくるが、実際に開発するシステムの規模を把握することは難しい。

今回紹介する手法は、Web をベースとしたエンタープライズ系の業務システムへの適用を前提とし、上流工程において機能の抽出や複雑さについて分析を行ないシステムの規模を算出する手法である。

6. 2. 1. 1. PSER とは

PSER とは、システムが支援する業務の機能を中心に分析を行なう方法である。業務を4つの観点から分析し、コンポーネントとデータに分類する。4つの観点とは図 6. 1 に示すように、業務プロセス(Process)、業務サービス(Service)、業務エンティティ(Entity)、業務ルール(Rule)からなり、それぞれの頭文字をつなぎ合わせて PSER と呼ぶ。最初に PSER それぞれの概念について紹介を行なう。

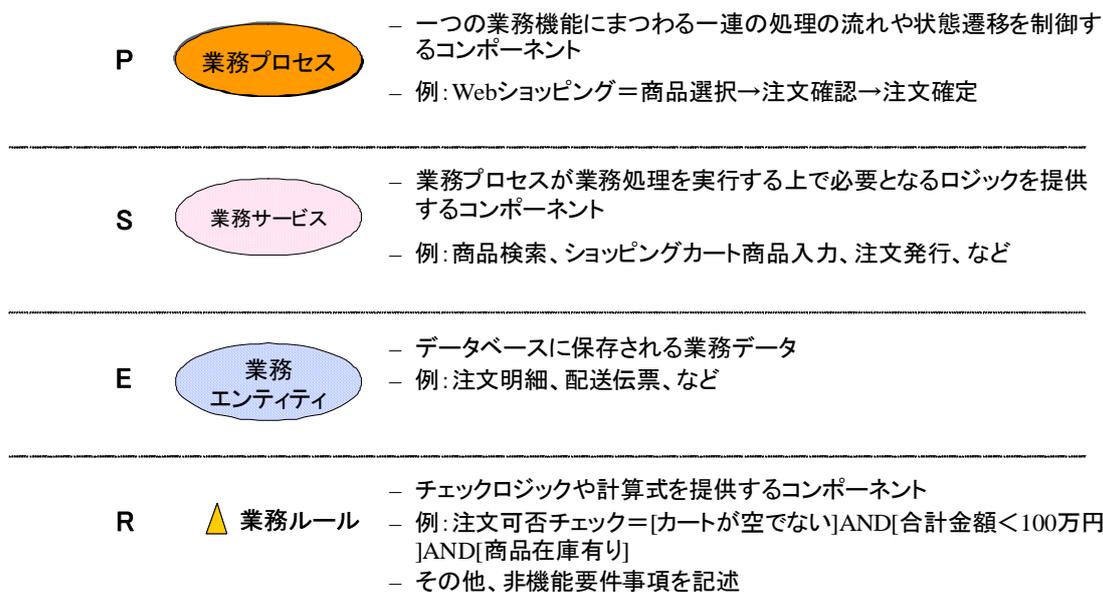


図 6.1 PSER の概要

① 業務プロセス(Process)

業務機能に関する一連の処理の流れや状態遷移を制御するコンポーネントである。図 6.2 に示すように、業務プロセスは複数の業務サービス、業務エンティティ、業務ルールで構成される。

② 業務サービス(Service)

業務サービスは業務プロセスが業務処理を実行する上で必要なロジックを提供するコンポーネントである。業務プロセスは複数の業務サービスの組み合わせにより一連の処理を実行する。

また、業務プロセスと業務サービスの数により、そのシステムの規模が予測できる。

③ 業務エンティティ(Entity)

業務エンティティはデータベース等に保存される業務データを示す。業務エンティティの単位は例えば従業員データ(従業員番号、姓名で構成)など適切な単位で構成される。

業務エンティティが対象とするのは基本的に永続データとする。

④ 業務ルール(Rule)

業務ルールは業務におけるチェックロジック、計算式等を表している。業務プロセスや業務サービスにおいてチェックロジック等が存在する場合は図 6.2 に示すように、業務プロセスや業務サービスに差し込むような形で記述する。業務ルールが複数ある業務プロセスや業務サービスはそのロジックが複雑になる可能性が高いことがわかる。

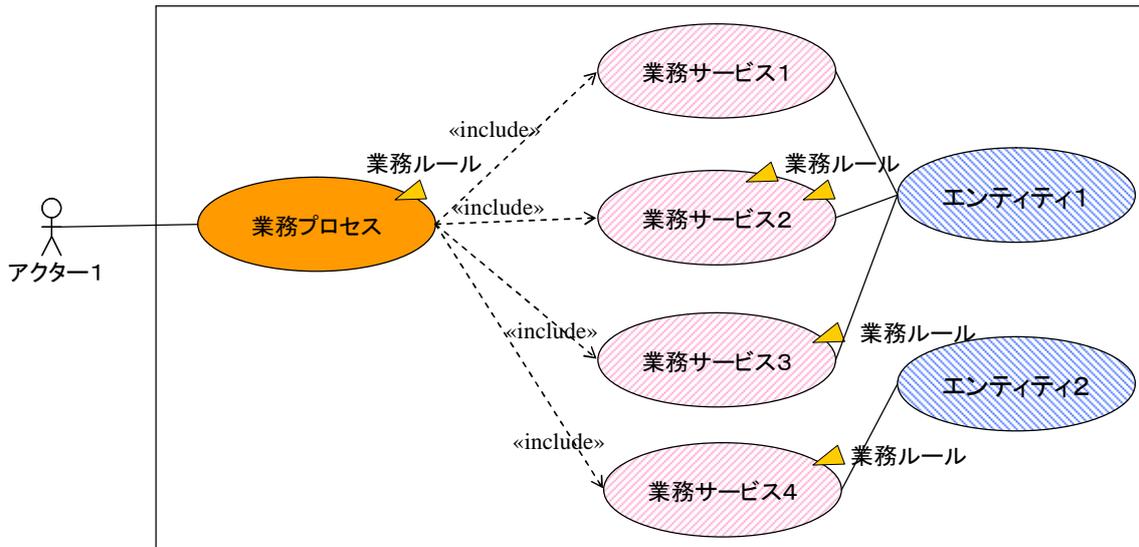


図 6. 2 PSER でのユースケース図

6. 2. 1. 2. PSER での分析事例

PSER で分析を行った事例を図 6. 3 に示す。これは旅費清算システムの機能の一部を PSER で分析した結果である。対象とした機能は交通費の清算を申請した後に、その申請書に対して改めて変更を行う部分について PSER で示した。

P、S、E、R それぞれをどのように分析し、抽出したかについて説明する。

■ 業務プロセスの抽出

本稿では外部仕様レベルの業務機能単位を業務プロセスとして抽出した。交通費の清算システムの場合は「交通費清算申請書を登録する」「申請書を一覧・変更する」「交通費を集計する」などが抽出される。

この時点では、業務プロセスの粒度が大きい場合や小さい場合などもあるが、これは業務サービスを抽出する時点で見直しを行う。

■ 業務サービスの抽出

業務プロセスに続いて、業務サービスを抽出する。業務サービスを抽出するには、各業務プロセスの内部で、どのような機能が順番に実行するかを分析し、その機能を業務サービスとする。業務プロセス「申請書を一覧・変更する」は「申請書を検索する」「申請書詳細を取得する」「申請書を更新する」という機能を順に実行することで一連の処理を行なうことが可能であることがわかる。

■ 業務エンティティの抽出

今回の業務サービスは交通費を清算するための申請書データをデータベースに登録し、それにアクセスすることが必要である。このデータを業務エンティティ「申請書情報」として抽出する。

業務エンティティの単位は、まとまったデータの集まりを想定しており、データベースにおけるテーブルレベルの粒度を想定している。

■ 業務ルールの抽出

業務ルールは業務プロセスや業務サービスにおけるチェックロジックを記述する。図 6.3 では、いくつかの業務サービスがユーザ認証において認証されている必要があるため、「認証されている」の業務ルールを抽出している。

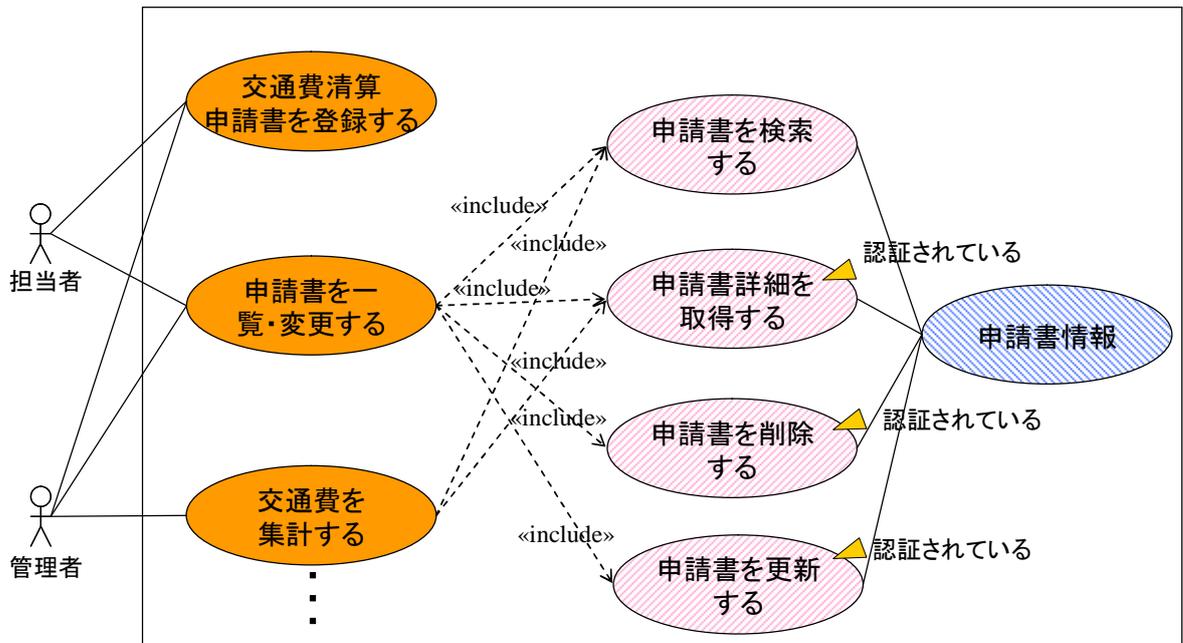


図 6.3 PSER 分析の例

6.2.1.3. PSER の有効性

PSER が顧客の要求する機能を抽出し整理する方法であることを説明した。

本稿で抽出した、業務プロセス、業務サービス等の個数によりシステムの開発規模の見積もり精度について検証を行なってみた。

見積もり方法は、基本的に業務プロセスと業務サービスの数に対して、それぞれの単位あたりの平均的な開発工数(それまで社内に蓄積されたデータから算出)を乗算することにより行なう。業務プロセスや業務サービスに多数の業務ルールが差し込まれている場合や業務エンティティへのアクセスが多い場合は、重み付けによって、平均的な工数より多くしている。

図 6.4 は要件定義完了時、システム設計完了時の見積もり工数と、システム開発に実際にかかった工数を比較したグラフである。開発工数の規模がかなり近く、PSER による見積り精度がよいことがわかる。

また、実際に PSER の手法自体がシンプルであるため、顧客が PSER を理解でき、PSER による分析結果の妥当性を評価できるため、システムの開発規模の見積もりについて同意がとりやすくなる。

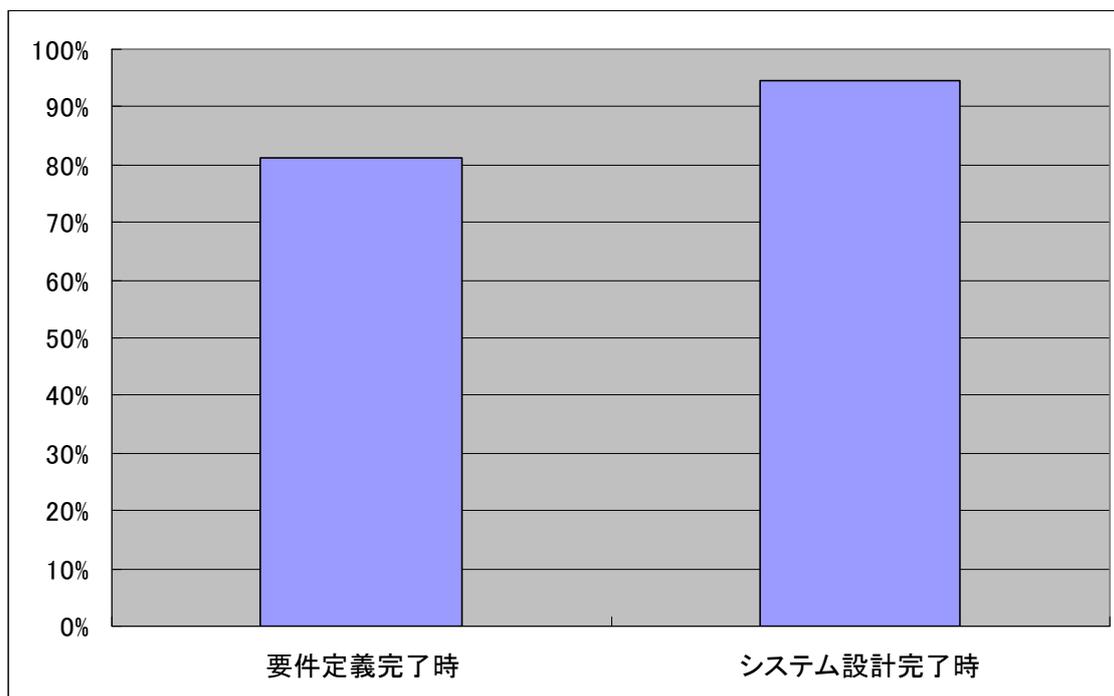


図 6. 4 開発終了時に対する PSER による見積もりの割合

6. 3. 顧客満足度評価方法

6. 3. 1. 顧客満足度とは

システムにおいて顧客の満足度を算出する場合、顧客の要件に対する機能の有無により判断するケースが多い。もちろん顧客が要望する機能がなければ、顧客の要求を満足させることは難しいが、機能の有無だけで、提供されたシステムに顧客が満足するとは限らない。例えば提供する機能が難解で顧客が利用できないということがある。この場合、機能の有無では顧客の満足を満たしているが、実際には利用できない機能であるため結局は顧客の満足を満たしていないことになる。

6. 3. 2. 家電製品の顧客満足度算出方法

家電製品では、QFD(Quality Function Deployment)を利用して顧客満足度を算出する方法が確立されている。この考え方をシステムの顧客満足度の算出方法への応用について検討を行った。

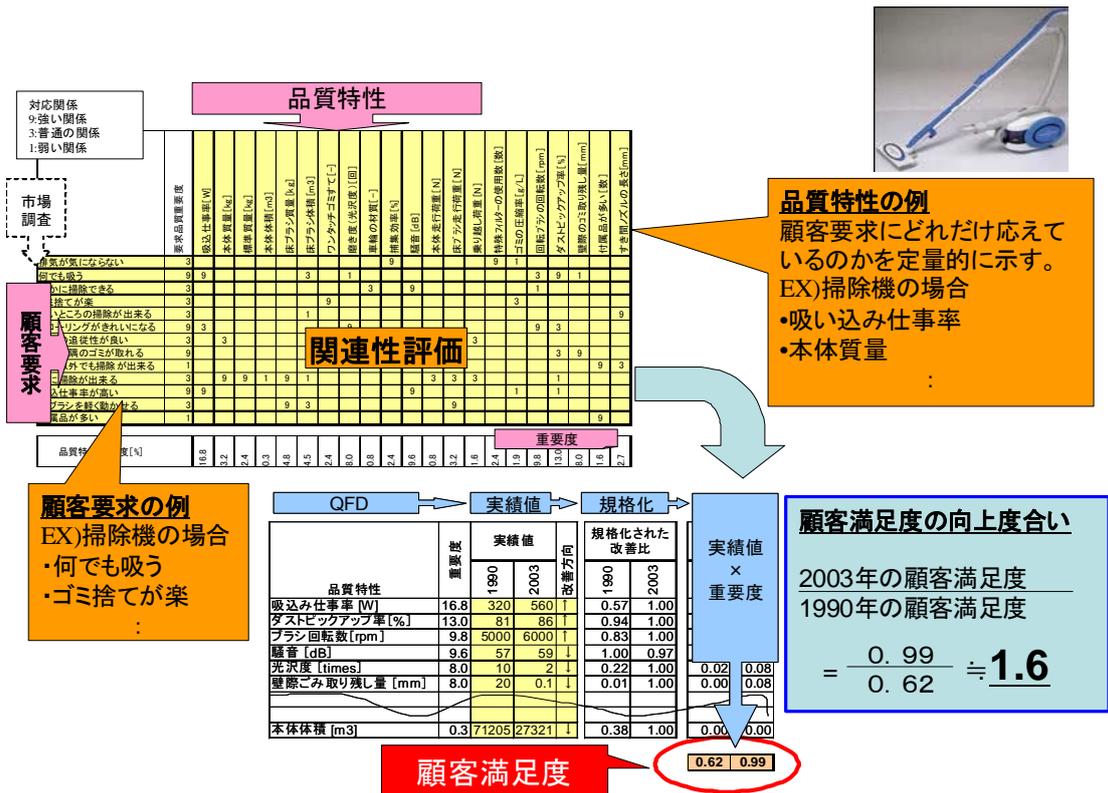
最初に、掃除機を評価した事例を紹介する。

QFD では製品の顧客要求に対する顧客満足度を品質特性で定量的に評価を行なう。図 6. 5 に掃除機の顧客要求と品質特性を示す。

例えば、顧客要求の一つである「ごみを何でも吸う」の顧客満足度は、ごみの吸い込み力を示す品質特性「吸い込み仕事率」を利用して評価できる。この場合、「吸い込み仕事率」が従来の掃除機の 320W に対して新製品が 560W である場合は顧客満足度が 1.75 倍に向上したこ

とになる。

掃除機の一部の機能の顧客満足度の算出ではなく、掃除機全体の顧客満足度を算出するには、複数の顧客満足度と品質特性から総合的に算出する必要がある。この場合は QFD¹を利用して評価する。図 6.5 の例では、1990 年と 2003 年の掃除機の顧客満足度の向上度の比較を QFD を利用して評価を行った結果、約 1.6 倍と向上することがわかる。



出展元: 小林由典, 小林英樹他: J. Industrial Ecology, Vol9(4), 2005

図 6.5 家電製品における顧客満足度の評価方法

6.3.3. システムでの顧客満足度の算出方法

掃除機のように、家電製品の品質特性は顧客要求を定量的に示すことができるため顧客満足度の算出が可能である。システムでも顧客満足度を算出するためには、家電製品のように顧客要求を定量的に示すことができる品質特性を抽出する必要がある。

そこで、システムの品質特性の算出手法を開発したので紹介する。

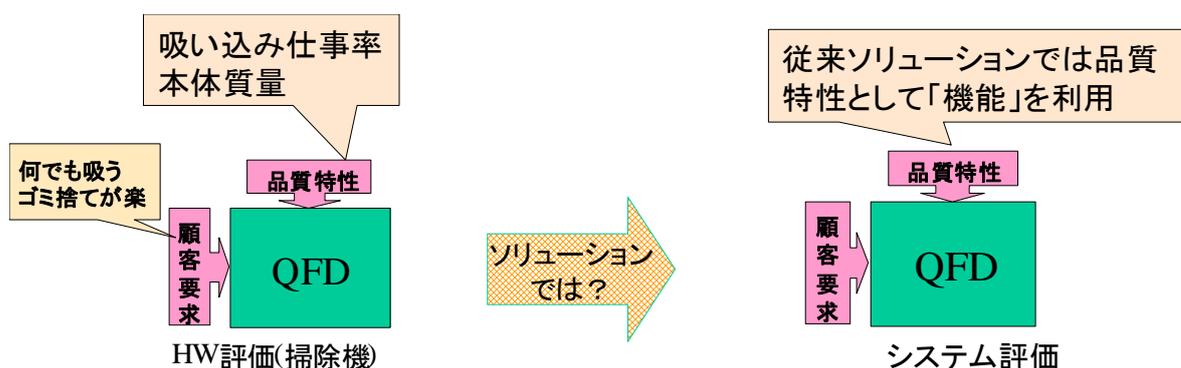
品質特性を抽出するには、なぜこのような顧客要求がだされたのか、その根拠を分析し、定量的に示すことができるのであれば、それを品質特性として採用できるのではないかと仮説を立てた。例えば、「システムの処理をもっと早くしてほしい(つまりは売上げにつながらない)」と

¹ QFD の利用方法については、参考文献[7]を参照。

いう顧客要求の根拠について分析を行なった結果、「従来のシステムにおける処理時間が 100 秒もかかり遅いので 10 秒以下にしてほしい」と具体的になる。この場合、処理時間が品質特性の候補となり、処理時間が早くなればなるほど顧客満足度は向上することになる。

このように顧客要求を満たすために、具体的にどうなればいいのかを分析する手法としてラダーリングという手法がある。そこで、顧客要求から品質特性を求める手法としてラダーリングを採用した。

以下、この方法について旅費清算システムで評価を行い、有効性を検証した。



出展元:清水歩、村田尚彦他:東芝ソリューション(株):電気学会 電子・情報・システム部門大会:2007年9月

図 6.6 HW 製品(家電製品)とシステムとの違い

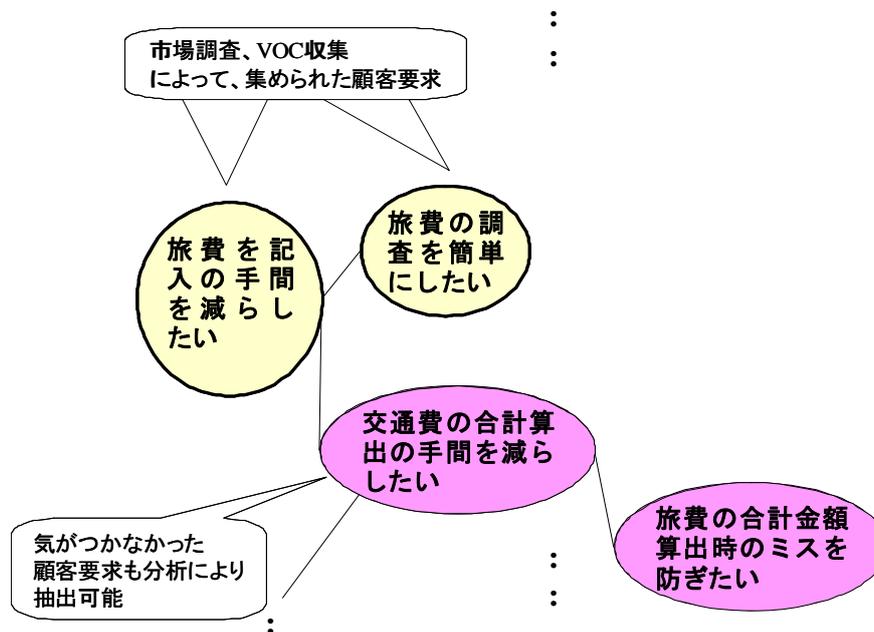
6.3.3.1 顧客要求の抽出、整理

顧客満足度を算出するために旅費清算システムを利用する一般従業員をステークホルダーとした。

最初に顧客要求の抽出を行なった。顧客要求自体はインタビューなどにより抽出する。抽出された顧客要求は、それぞれの間に関係が存在したり、表現は異なるが同じ顧客要求である場合がある。このような顧客要求を整理についてもラダーリングを利用した。例えば、図 6.7 にラダーリングで顧客要求を整理した例を記述する。顧客の要求である「旅費の記入の手間を減らしたい」に対して「旅費の調査を簡単にしたい」という顧客の要求が階層関係になっていることがわかる。

また、ラダーリングにより「交通費の合計算出の手間を減らしたい」という新しい顧客要求が推測でき導き出すこともできる。

以上のように、インタビューを通して導き出された顧客要求をラダーリングによって整理を行なった。



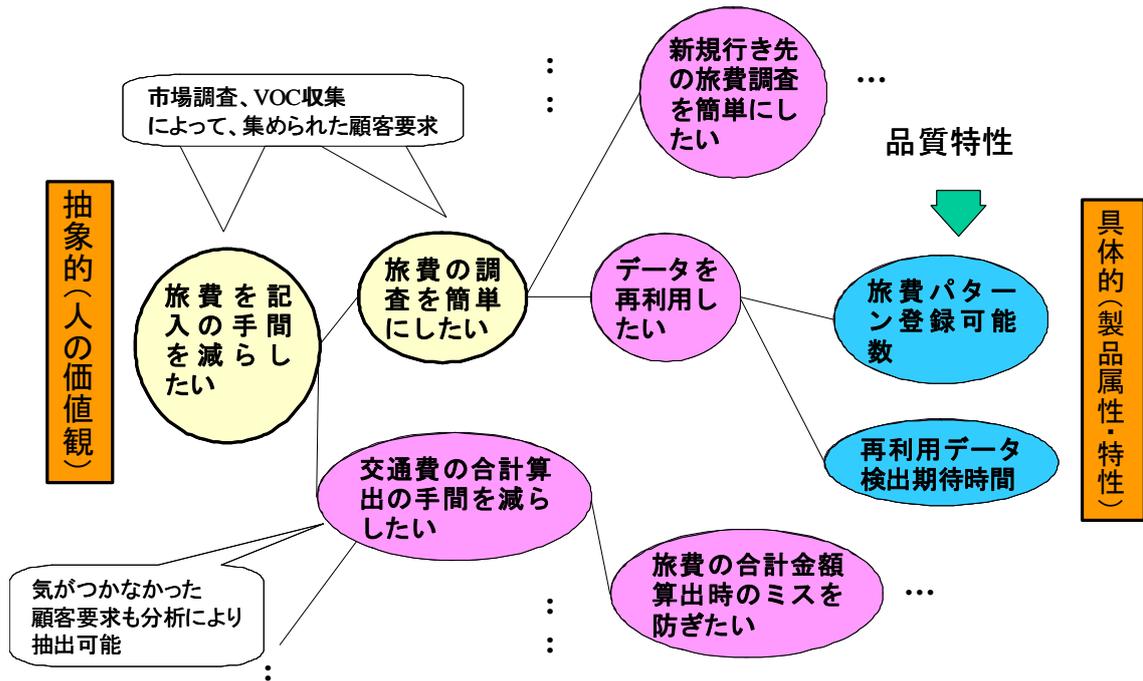
出展元: 清水歩、村田尚彦他: 東芝ソリューション(株): 電気学会 電子・情報・システム部門大会: 2007年9月

図 6.7 顧客の要求の整理

6.3.3.2. 品質特性の導出

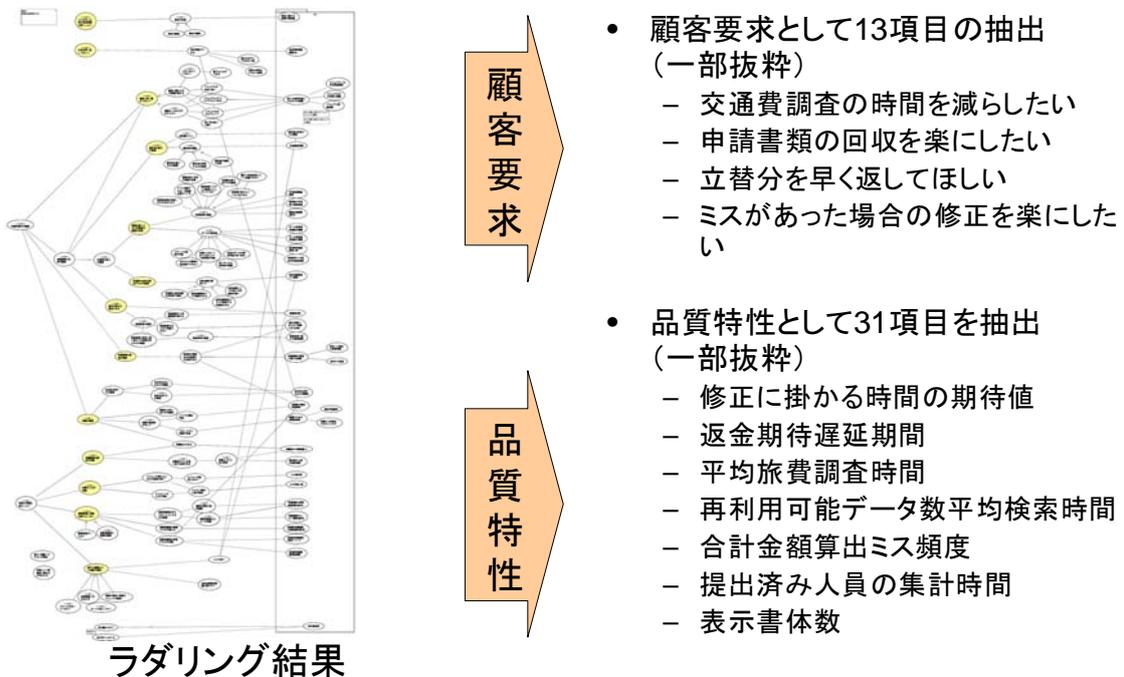
図 6.8 はラダーリングにより顧客要求から品質特性を求めた例である。例えば、「旅費の調査を簡単にしたい」から「データを再利用したい」とラダーリングされている。これは、「旅費の調査を簡単にしたい」に対して、具体的な手段を考えると、過去に同じ場所に出張したことがある場合には過去のデータが再利用できれば調査が簡単になると考えることができるので「データを再利用したい」が抽出される。さらに同様の分析を行っていくことで最終的に品質特性を抽出することができる。

図 6.9 に旅費清算システムに対して抽出した品質特性を示す。



出展元: 清水歩、村田尚彦他: 東芝ソリューション(株): 電気学会 電子・情報・システム部門大会: 2007年9月

図 6.8 ラダーリングによる品質特性の抽出



- 顧客要求として13項目の抽出 (一部抜粋)
 - 交通費調査の時間を減らしたい
 - 申請書類の回収を楽にしたい
 - 立替分を早く返してほしい
 - ミスがあった場合の修正を楽にしたい

- 品質特性として31項目を抽出 (一部抜粋)
 - 修正に掛かる時間の期待値
 - 返金期待遅延期間
 - 平均旅費調査時間
 - 再利用可能データ数平均検索時間
 - 合計金額算出ミス頻度
 - 提出済み人員の集計時間
 - 表示書体数

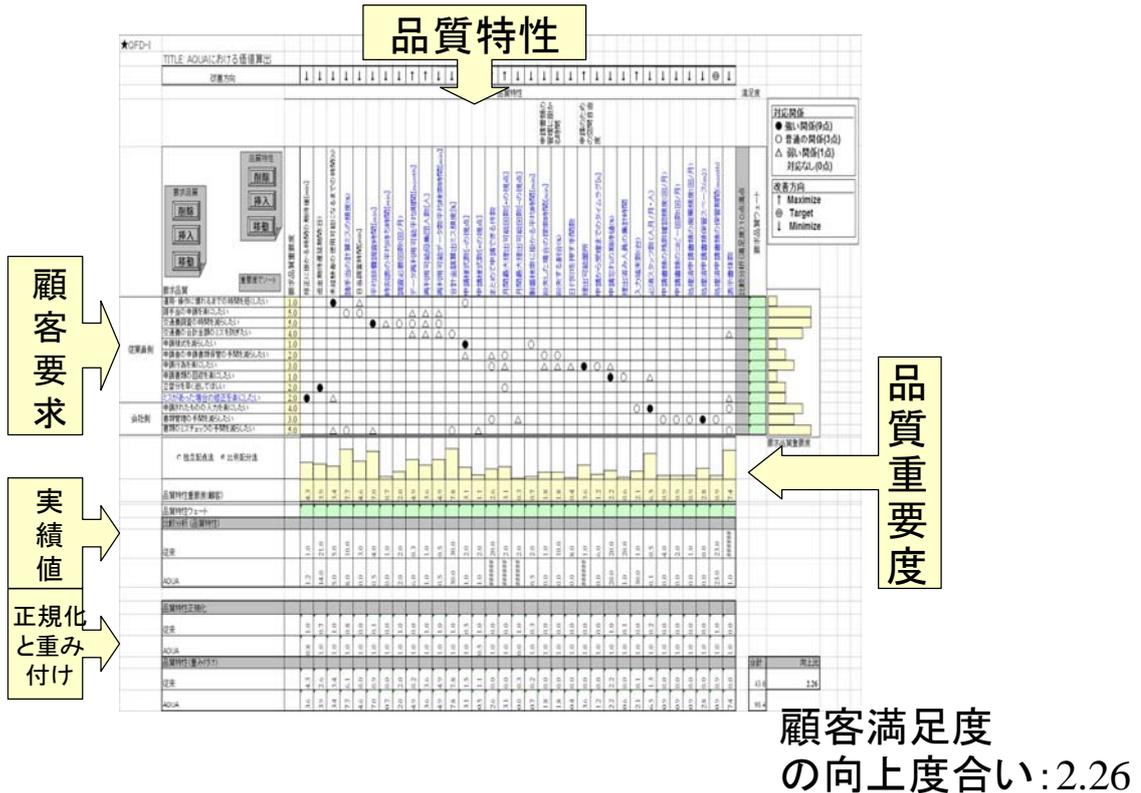
出展元: 清水歩、村田尚彦他: 東芝ソリューション(株): 電気学会 電子・情報・システム部門大会: 2007年9月

図 6.9 ラダーリング結果

6. 3. 3. 3. QFD によるシステムの顧客満足度の算出

求められた、顧客要求と品質特性をもとに、図 6. 10 に示す QFD を作成した。本稿では旅費清算の申請を紙ベースで行う運用と旅費清算システムを利用した運用とを比較し顧客満足度の向上度合いを算出した。

その結果、2.26 倍向上していることがわかった



出展元: 清水歩、村田尚彦他: 東芝ソリューション(株): 電気学会 電子・情報・システム部門大会: 2007 年 9 月

図 6. 10 旅費清算システムの顧客満足度

6. 3. 4. 顧客満足度算出の有効性

システムでも家電製品のような品質特性を抽出し、顧客満足度を算出することが可能であることを示した。機能の有無によって顧客満足度の評価を行なう手法と比較すると、品質特性を顧客にわかりやすい項目でかつ定量的に示すことができるため、顧客はシステムを利用した時点での満足度を容易に推測できる。また、開発着手前に顧客とベンダー間において、品質特性の数値について調整を行い、同意を得ることができれば、その数値を目標にシステム開発が行うことができ、顧客が満足するシステムを提供することが可能となる。以上のことから、紹介した手法で顧客満足度を算出することは有効であると言える。

6. 4. まとめ

本稿では PSER での機能分析によるシステムの規模を算出する手法とシステムを対象に顧客満足度を算出する方法について紹介した。

PSER は顧客の要件からの機能の抽出と、その規模を示すことが可能であり、その根拠が顧客にもわかりやすい方法であることを示した。

また、システムの顧客満足度の算出方法については、顧客のシステムへの要望に対する度合いを示す方法を示した。この二つの方法を組み合わせることにより、ベンダーが顧客に提供するシステムの機能、規模や、顧客にとっての満足度を示し、それらを顧客も容易に理解することができるようになる。

この二つの方法を組み合わせることにより、システム開発の早い段階に、顧客とベンダーが開発規模等を調整時する際、システムにどのような機能を作りこめば、顧客満足度を最大にするかを検討できる。今後はこれらの手法を組み合わせ、その活用方法について検討する。

[参考文献]

- [1] 細谷竜一、村田尚彦:「ビジネスアプリケーション開発へのモデル駆動型アーキテクチャ (MDA)の適用」、東芝レビュー、Vol59 No11、2004
- [2] 清水歩、村田尚彦:「ICT ソリューションのためのファクター算出方法の提案」、電気学会 電子・情報・システム部門大会、2007/9
- [3] 細谷竜一他:「MDA 大研究!」、株式会社アイ・ディ・ジー・ジャパン、JavaWorld、2004/5
- [4] Y. Kobayashi, H. Kobayashi, A. Hongu, and K. Sanehira, “Practical Method for Quantifying Eco-efficiency Using Eco-design Support Tools”, Journal of Industrial Ecology vol.9, no.4, pp.131-144, 2005
- [5] 東芝:「製品の新しい指標「ファクターT」のお話 東芝グループがすすめる「新たな豊かさ」の追求」,
http://www.toshiba.co.jp/env/jp/products/pdf/factor_t_basic.pdf, 2005
- [6] 「情報通信技術 (ICT) の環境効率評価ガイドライン」、日本環境効率フォーラム,
<http://www.jemai.or.jp/japanese/eco-efficiency/pdf/guideline.pdf>, 2006
- [7] 赤尾洋:「新製品開発と品質保証-品質展開のシステム. 標準化と品質管理」, 日本規格協会, vol.25, no.4, pp. 9-14, 1972
- [8] 山本修一郎:「システム要求管理技法—ゴール指向による」、ソフトリサーチセンター、2007/5
- [9] 清水吉男:「要求を仕様化する技術・表現する技術 - 入門+実践 仕様が書けていますか?」、技術評論社、2005/10

7. CSK システムズにおける要求定義と要件定義の事例

株式会社 CSK システムズ 安田 哲平

7. 1. はじめに

CSK システムズにおける開発標準(プロセス標準ガイド)は、「EPISODE/II¹」といい、「ソフトウェア開発上における効果的かつ、有意義な情報」という意味の略称である。業界標準ともいえる、SLCP(Software Life Cycle Process)、共通フレーム 98、PMBOK(Project Management Body of Knowledge) 等に準拠し、マニュアル、フォーム、サポート資料、特性格標準で構成されている。

2007 年に EPISODE/II の上流工程を改訂した際の要求定義フェーズを中心に、要求分析の考え方を説明する。EPISODE/II 全体の体系を図 7.1 に示す。

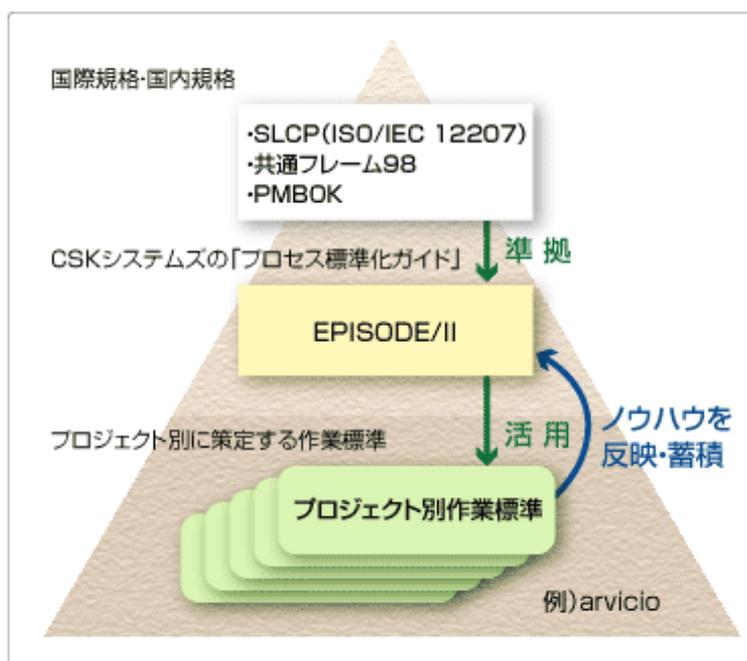


図 7. 1 EPISODE/II の体系

7. 2. よくある風景

実際のシステム開発の現場では図 7. 2 のような状況をよく見かける。

具体的には、企画で予算・効果・施策の概略程度しか明らかになっていないのに、十分検討せずに企画から RFP を作成しているケースなどである。

その場合、そのまま要件定義(システム要件定義)に突入すると、要件が固まらないことがよくある。あるいは業務が固まらないままシステムの話を詰めていって土壇場でひっくり返ったりする。

¹ EPISODE/II : Effective and Profitable Information on Software Development の略

こういう状況に陥ってしまう一つの要因として、多くのシステム開発標準が要求定義フェーズにおいて、受注側（ベンダ・SIer）の視点に偏っていることが挙げられる。もちろんそれは、システム開発受注時におけるリスク低減・回避という当然のことではあるのだが、発注側でシステムを纏めきるパワーが低下している状況にあっては、問題を孕んでいるといえよう。



図 7. 2 よくある風景

7. 3. 要求定義とは

弊社開発標準 EPISODE/II における要求定義の位置付けは、「要求定義フェーズは企画と要件定義の間に位置し、プロジェクトの要求及び業務・システムの範囲を確定させるための工程である。」と定義している。



図 7. 3 要求定義フェーズの位置づけ

ここでいうプロジェクトは顧客視点でのプロジェクトであり、ベンダによるシステム開発プロジェクトではない。例えば、製造業における『在庫削減プロジェクト』というようなものが、ここでいうプロジェクトであり、業務プロセスの改善、生産方式の転換、工場の新設や組織の統廃合、情報システムの再構築などを含んだプロジェクトである。情報システム開発はプロジェクトの一部であることが理解できよう。企画～要求定義フェーズにおいては単なるシステム開発プロジェクトよりも広い視点での検討が求められる。

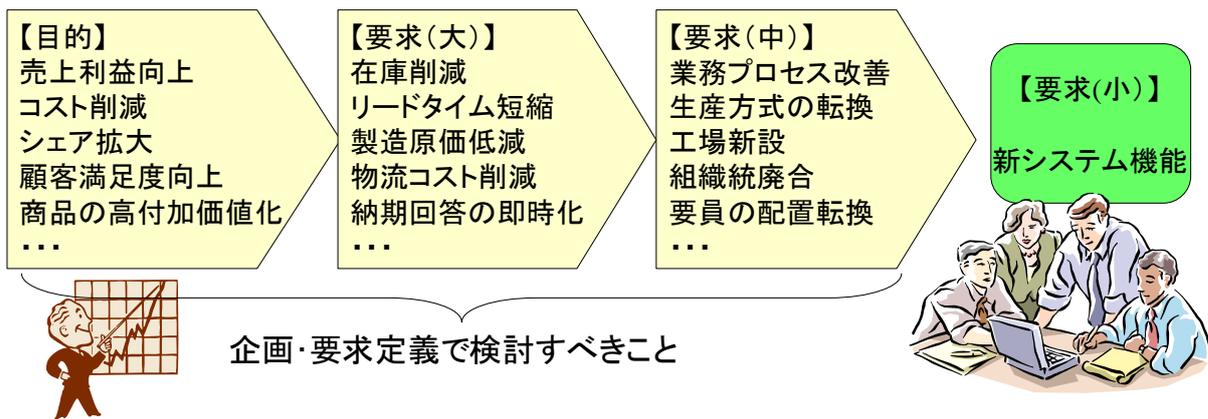


図 7. 4 プロジェクト全体と情報システム

要求定義の主な内容は、次の 4 つである。

- (1) 経営の要求が正しくプロジェクトの要求として展開されていることを確認する。
- (2) 要求実現後の姿を描き、新業務に必要な業務機能を明らかにする。
- (3) 各種の制約の下でその要求を実現する案を検討しプロジェクトとしての案を選択する。
 (この結果としてプロジェクトへの要求、及び業務・システムの範囲が確定する)
- (4) 選択された実現案をブレイクダウンし実現後の姿を具体化する。

『超上流』と呼ばれる企画、要求定義、要件定義におけるプロジェクトステークホルダとそのパワーバランスを図 7. 5 に示す。企画から一足飛びに情報システム部門主体の要件定義に突入した場合の問題点が図からも見て取れる。要求定義を省略した場合、明らかに「業務を決める」ステップが不足している。先に情報システムを決めてしまうと、後で「これでは業務ができない、回らない」とひっくり返される、或いは使われないシステムになってしまう可能性が高い。

工程	企画	要求定義	要件定義
誰が牽引者か			
単純化すると	予算を取る	業務を決める	情報システムを決める

図 7. 5 超上流のパワーバランス

いまや、情報システム化されていない業務は無いので、業務を決めようとするれば必然的に情報システムも検討しなければならないが、工程により視点や力点が異なることに注意していただきたい²。

7. 4. 要求定義の重要性が増大している背景

要求定義で実施すべき内容は、過去においては暗黙的に発注側が実施していたものであるが、あえて『要求定義』フェーズを明示的にした背景には、以下の 2 点が挙げられる。

(1) 情報システムが大規模化・肥大化し、情報技術自体も日進月歩なので、情報システムの要件定義だけでも複雑化している。端的に言って、業務と情報システムと同時に全ての要件を詰めていくのは無理がある。

→ 『要求定義と要件定義のフェーズを明確に分けて、効率的に進める効果』

(2) 要求定義は本来当該企業(発注側)でやるべき内容だが、情報システム部門が弱体化してそこまで出来ていない場合が多い。そのため要件定義フェーズになっても元々の要求が揺れ動くことがよくある。

→ 『要件定義に入る前に要求定義の完了確認(手戻り防止)の意味合い』

7. 5. 『超上流工程』の本質的な難しさ

この章では『超上流工程』の本質的な難しさについて述べる。

(1) 次工程以降でないと確定しない事柄を事前に決める、決めざるを得ない工程である。

たとえば見積についていえば、まだ確定していないシステム要件に対してそれを実現するためのシステムの開発費やインフラ費用を見積もらなくてはならない。実現可能性についても、具体的手段が不明確な時点で、実現性の可否を判断しなくてはならない。プロジェクトリーダーが進退を賭してコミットせざるを得ない局面といえる。

(2) プロジェクトステークホルダの利害対立の調整がまだついていない工程である。

これらの工程はプロジェクトステークホルダの利害調整を行うための工程といえる。

プロジェクトステークホルダは当然のごとく、立場毎に要求内容が違い、要求の優先度が違う。よく言われる「声の大きい人の要求が通る」とか「偉い人の要求が最優先」ではなく、本来の目的、目標にかなった要求を重視しなければいけない。

ただし、現実問題として、情報システムの顧客業務上で位置づけや開発上での意思決定な

² 業務と情報システムの決め方については、1. 10. 参考資料、参照のこと

どを考慮すれば、「偉い人の要求」を考慮する必要もあるので、落としどころを探り、政治的根回しが必要になるなど、プロジェクトマネジメント的に難しい局面である。

(3) 見えない要求、見えない制約を吸い上げなければならない工程である。

要求は人によって違うため、また当たり前のことはことさら言わないことが多いため、ヒアリング等で表に出てこない要求がある。また、成文化されていない要求もあるので部外者には分からない暗黙の要求、暗黙の制約があることが多い。特に「現行どおり」と言う要求は暗黙の要求の塊と考えて間違いない。

これらは、意識的に且つ具体的に聞き出さないと出てこない。「暗黙の要求はありませんか」ではなく、「この業界ではこれこれこういう要求がある場合が多いのですが御社の場合はどうですか」と例示できるレベルで聞き出さないと、抜け漏れが発生しやすい。

プロジェクトメンバに業務知識が必要な局面である。

(4) 共通理解の困難な工程である

まず、各人が理解できる領域・分野が異なるというのを認識しておかなければならない。経営者や業務部門の人にシステムの、あるいは、実現レベルの詳細を話しても理解してもらえない。

また、言葉による理解の限界があり、「判った」と言っても本当に理解しているかどうか分からない場合も多い。各人が持つイメージのギャップも相当大きい。

具体物・完成物が無い状態、つまりまだ出来上がっていないものに対して共通理解を求めざるを得ない局面である。プロトタイプ的な手法で確認する場合もあるが、気をつけないと、「もうシステムができていないか」と勘違いされる場合が多い。いくら説明しても「プロトタイプで出来ていたのに、何故、本番システムで出来ないのか」と詰め寄られると、「共通理解は難しい」と再認識する羽目になる。

(5) やりたいことはたくさんあるが、できることは少ない(できることが少ないことを確認する工程)

制約としてコスト、納期、人、技術、などがあるのは全員承知して検討を始めている筈だが、検討しているうちにどんどん要求が膨れ上がってしまう。より良いものを作りたいという思いで各自が要求を出しているのに、カットするには忍びないが、心を鬼にして「できること＝やらねばならないこと」を決めていかなければならない。

7. 6. 解決の基本原則

前章で述べたように要求定義フェーズは本質的に難しい工程であるが、難しいからといって手をこまねいている訳にはいかない。要求定義に焦点を絞って、その解決策を述べる。

要求定義の原則は『当たり前のことを当たり前に実行する』である。

原則 1: まず範囲を決める。深堀はそれから

要求が膨らんでいき收拾がつかなくなる場合の多くは、対象とする範囲が不明確なケースが多い。プロジェクトステークホルダ(経営者、企画部門、業務部門、情報システム部門など)の誰が見ても判るような切り口で対象範囲を定義して、議論がそこからあふれ出ないようにすることが大事である。

内容を深堀するのは範囲が確定してからである。往々にして自分が知っている分野について、対象範囲がまだ決まっていないのに先走って深堀しがちである。

原則 2: 深堀は概略から詳細へ、大から小へ

深堀する際も、概略から詳細、大から小へという手順を踏まないと、抜け漏れが発生する可能性が高い。

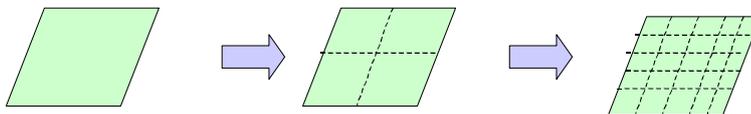
原則 3: 皆が理解できるもので合意を取る

情報システム部門またはベンダが音頭をとって進める場合が多いが、その際、システム担当者しか判らない用語や図式を多用してドキュメントを作りがちである。全プロジェクトステークホルダが理解できるものは、フロー(業務フローなど)、表(業務機能一覧表など)などに限られることに注意して作業を進めるべきである。シンプルな絵や図・図式も判りやすいものではあるが、説明の必要な絵・図・図式(例えば DFD、ERD、ユースケース図など)は理解してもらえない、あるいは間違った理解をする可能性があるので注意が必要である。

原則 1: まず範囲を決める、深堀はそれから



原則 2: 深堀は概略から詳細へ、大から小へ



原則 3: 皆が理解できるもので合意を取る



図 7. 6 要求定義の原則

7.7. 要求定義で要求が確定していく様子

企画書を受け取って要求定義を開始してから終了するまでの間で、要求がどのように確定していくのかを図 7.7 で示す。

Step1: 企画の確認

企画書で定義されたプロジェクトの目的・目標を確認する。投資の範囲・総額も確認する。

※企画書は人によって解釈が異なる(自部門に都合の良いように解釈する)ので気をつける。

Step2: 現行の確認

現行システム、現行業務で実現されている範囲を確認する。

※「現行どおり」をきちんと定義するためにも重要である。プロジェクトステークホルダによって、現行の認識が異なる場合も多い。

Step3: 要求の確認(当初要求)

プロジェクトの目的・目標と現行との差異が新規要求となる。また現行の中で不要なものの廃止や変更も要求として出てくる。

※新規の要求や変更要求は細かくでてくるが、一言で「現行どおり」といわれる要求が実は一番大きな要求である。

プロジェクトの目的・目標に紐づかない要求(ついで要求)が出てくるがよくある。

※範囲外の要求をカットするには、その要求がプロジェクトの目標・目的と紐づいているかついていないかを図式化して見せるのが良い。

Step4: 実現案候補の作成

制約(人、モノ、金、時間、環境など)の範囲内で要求を最大限実現する案を複数作成する。

全ての要求を実現するのは通常無理なので、要求の優先度・重要度を考慮して決定する。

※制約(特に予算面)により実現されない要求があることをプロジェクトステークホルダに納得させなければならない。

Step5: 実現案の選択

複数の実現案候補を客観的なものさしで比較検討し、「一つの実現案」を選択する。

※実現案候補は複数出てくるが、必ず「一つ」を選ばなければならない。

Step6: 要求実現範囲の確定

実現案によって実現される要求の範囲を宣言する。

※選択された実現案を元に、確定した実現範囲をプロジェクトステークホルダに指し示し、念を入れて確認する(儀式が必要)。

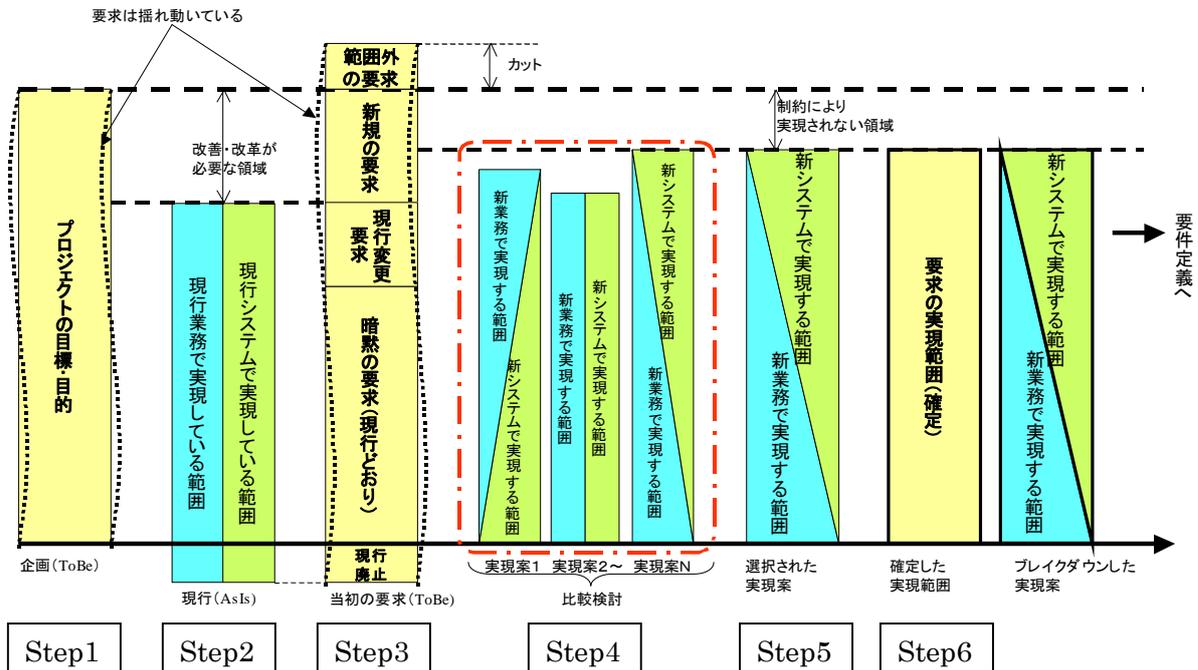


図 7.7 要求定義で要求が確定していく様子

7.8. 要求定義終了後の状態

要求定義終了時に「要求定義書」というアウトプットが作成される。要求定義書は、RFP（提案依頼書）と同等内容または RFP よりはやや詳しいレベルの成果物である。

ただし、要求定義書は、各種検討が実施され、経緯も含めて関係者に合意されているドキュメントだということに意味がある。

情報システム開発の委託経験が豊富な企業は、『要求仕様書』を作成している。どういう経緯や経過でこの要求が出てきているのかとか、どういう検討をしてこの案に決めたのかとかいう疑問に答えられる文書になっている。

不十分な RFP はこのような経緯や経過が記載されていないため、最終的に要求が固まらず、プロジェクトの運営が不安定になりがちである。例えば情報システム部門だけの思い込みで RFP を作成しているケースがこれにあたる。

7.9. 最後に(私見:餅は餅屋)

情報システムを開発する場合、効果のないシステム(目的・目標に結びつかないシステム)を開発するのは大きな無駄である。一方、情報システム部門が付け焼刃で業務分析を行っても効果が薄い(業務分析ができる情報システム部門は当然存在する)。

企画は企画者やコンサルタントが取り組むべきだし、業務は業務部門が検討すべきである。要求定義は 7.3 章でも述べたように、業務部門が主体となって実施するフェーズである。従ってベンダ(SIer)がこのフェーズを受注することは少ない。

しかしベンダとしてはこのような要求定義のプロセスを経て、要求が出てきているのかどうかを確認すべきである。後工程の情報システム部門は、前工程がきちんと実施されたのかを確認し、検討内容を把握して、情報システムの開発に全力を注ぐべきである。場合によっては、このままでは要件定義に入れないと(現実には難しいかもしれないが)受入検査で差し戻すことがあってもよい。

それぞれがそれぞれの立場でそれぞれの工程でベストを尽くすこと、餅は餅屋である(ただし、餅の原材料は糯(もち)米ということぐらいは知っておこう)。

7. 10. 参考資料

『何故、業務視点から要求の確定をしていくのか?』

要求定義ではまず業務要求(業務視点での要求)から決めていく。システム側から要求を決めていくと、本来何のためにやるのかという目的を見失いやすい。業務はプロジェクトの目的・目標と密接に関連している場合が多いので、業務から決めるほうがブレにくいし、要求の優先順位や重要度も設定しやすい。

またそれ以外にも、業務という切り口は全てのプロジェクトステークホルダに理解されやすいこと、業務とシステムはある程度対応しているので業務の範囲を決めれば自ずとシステムの範囲も決まってくるということが挙げられる。

業務要求が確定した後にシステム要件を深掘していけば、システム要件がブレたり無用のシステムを作ってしまう危険性が抑えられる。

例外的にシステムから決めていくのは、「業務上の不都合は無いがレガシーシステム(ホストコンピュータ)の保障期間が終了するのでオープン化しなくてはならない」、「親会社の指示により指定の会計パッケージを導入する」など特殊な場合に限られる。

【参考】業務とシステムの対応のイメージ

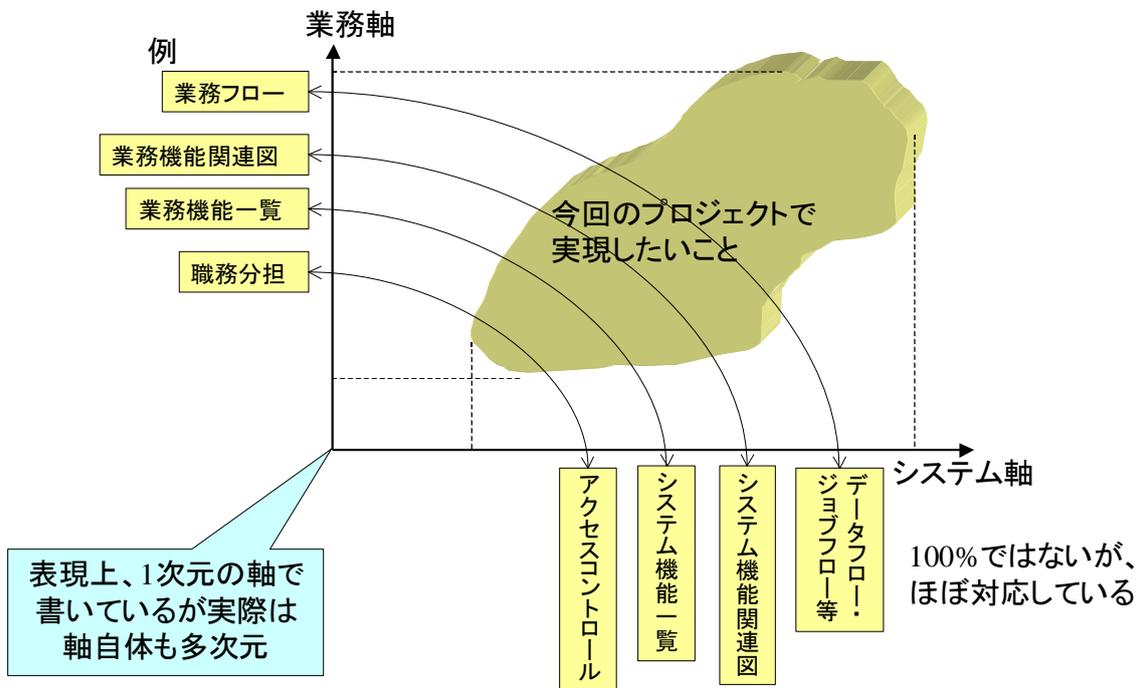


図 7. 8 業務とシステムの対応のイメージ

【参考】段階的検討1(要求定義)

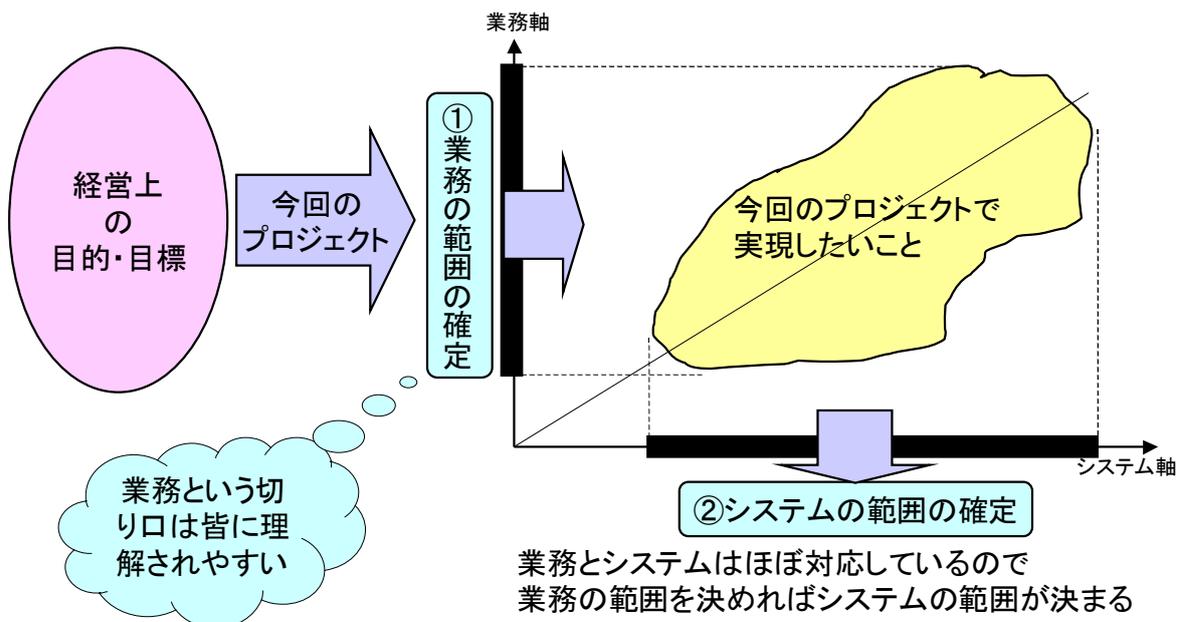


図 7. 9 段階的検討1(要求定義)

【参考】段階的検討2(要件定義)

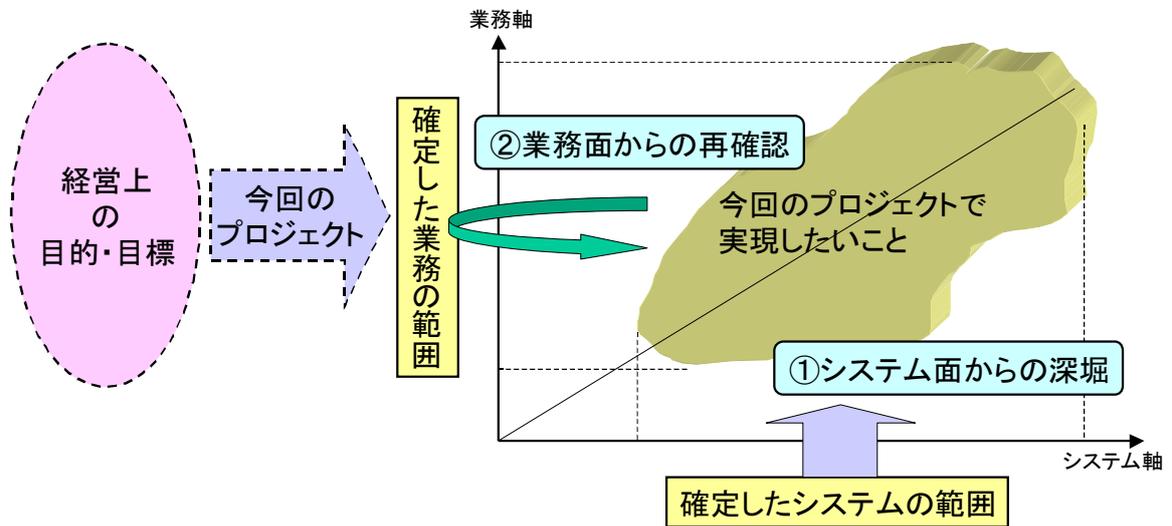


図 7.10 段階的検討 2(要件定義)

以上

8. 推進体制に注目した要件定義事例

株式会社シーエーシー 田中 等

8. 1. はじめに

要件定義では、要件を引き出し、分析し、まとめるために、多くのコミュニケーションや意思決定が必要になる。そのため、要件定義に関わる組織がどのような体制を取るかが非常に重要である。システムの背景や組織の事情により、プロジェクト毎に要件定義の推進体制は異なるが、本稿では代表的なパターンのプロジェクト事例を挙げてみたい。

8. 2. 要件定義の要素

プロジェクト事例を紹介する前に、まずは要件定義にはどのような人や組織が関わるのか、またどのような要素が重要なのかを挙げてみた。プロジェクトによって、これらの特性や関係は異なる。

(1) 要件定義に関わる組織

1) 顧客企業

システム開発を発注し、最終的にシステムを所有する企業。

2) エンドユーザ

開発されたシステムを実際に利用する人または組織。なお、顧客企業に所属するとは限らない。

3) システム部門

顧客企業のシステムを管理する部門。要件定義やシステム開発の際、ベンダとの窓口になることが多い。

なお、顧客企業の一部門ではなく子会社の場合もある。

4) ベンダ

顧客企業から要件定義作業を受注し、要件をまとめる。引き続きシステム開発を受注し、要件に基づいてシステムを開発する。

(2) 要件定義に必要な要素

要件定義に必要な要素は様々なものが考えられるが、筆者の実務経験から、以下の三つが特に重要と考える。

1) 予算

顧客企業において、要件定義やシステム開発を発注するための予算。予算を負担する部門は、エンドユーザが所属する部門の場合、システム部門の場合などがある。

2) 業務知識

システムが対象とする業務分野の知識。

3) IT スキル

IT スキルは、システムを開発するために必要であるが、要件定義段階でシステム像を描くためにも必要である。

8.3. 事例 1: ASP

(1) システムの背景

本事例では、システムを所有する顧客企業自身はシステムを利用しない。社外のエンドユーザに対しシステム機能を提供し、利用料を徴収することにより、利益を上げる。いわゆる、ASP (Application Service Provider) の形態である(図 8.1)。

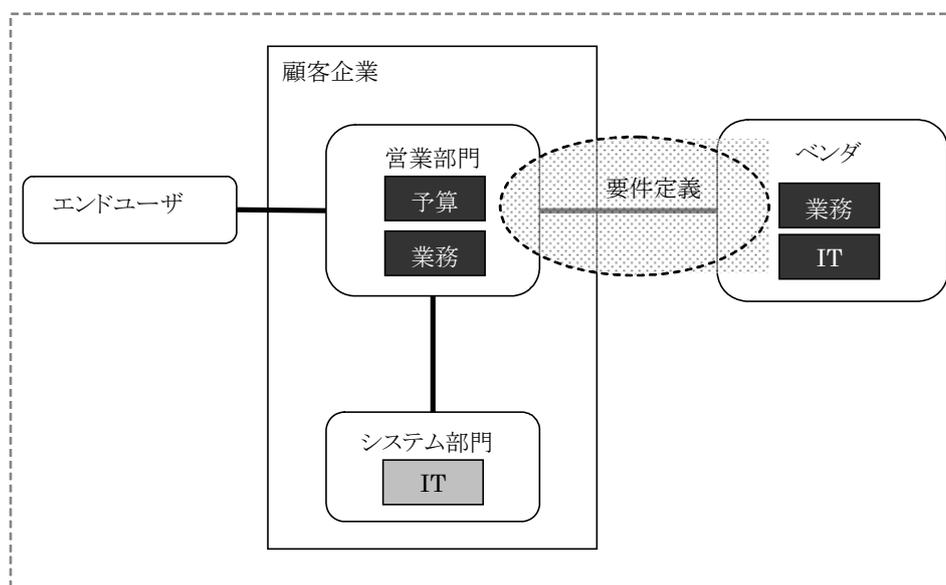


図 8.1 Application Service Provider ビジネス

(2) 組織

ベンダは、IT スキルは当然持ち、高い業務知識も保有している。

顧客企業の営業部門は、業務知識は持つが、IT スキルはほとんど無い。但し、システム開発の予算を持っている。

顧客企業のシステム部門は、業務知識も予算も持たない。そのため、要件定義などシステムの具体的な内容にはほとんど関わらない。

エンドユーザは、顧客企業外の組織である。顧客企業にとっての顧客である。

(3) 要件定義の経緯

ベンダが業務プロセスやシステム仕様を提案し、それを顧客企業の営業部門が確認する形で、要件定義を実施した。顧客企業のシステム部門は、業務に詳しくなかったため、ほとんど要件定義に関わらなかった。また、エンドユーザは顧客企業外の組織ということもあり、エンドユーザから

のヒアリングは行わなかった。しかし、顧客企業の営業部門とベンダが十分に業務を理解していたため、特に不都合は無かった。

要件定義は順調に行われ、システム開発に進むことができた。システム稼働後のエンドユーザからのクレームも少なかった。

(4) 考察

ベンダは高い業務知識と IT スキルを保有していたため、顧客企業にとって価値のあるシステムを提案することができた。顧客企業の営業部門も、十分な業務知識を保有していたため、そのシステム提案を正しく評価することができた。また、予算も保有していたため、順調に要件を確定していくことができた。

ベンダが高い業務知識を保有していたことが、大きな成功要因だったと言える。

8. 4. 事例 2: 社内業務システム

(1) システムの背景

本事例のシステムは、顧客企業の社内業務システムである(図 8.2)。

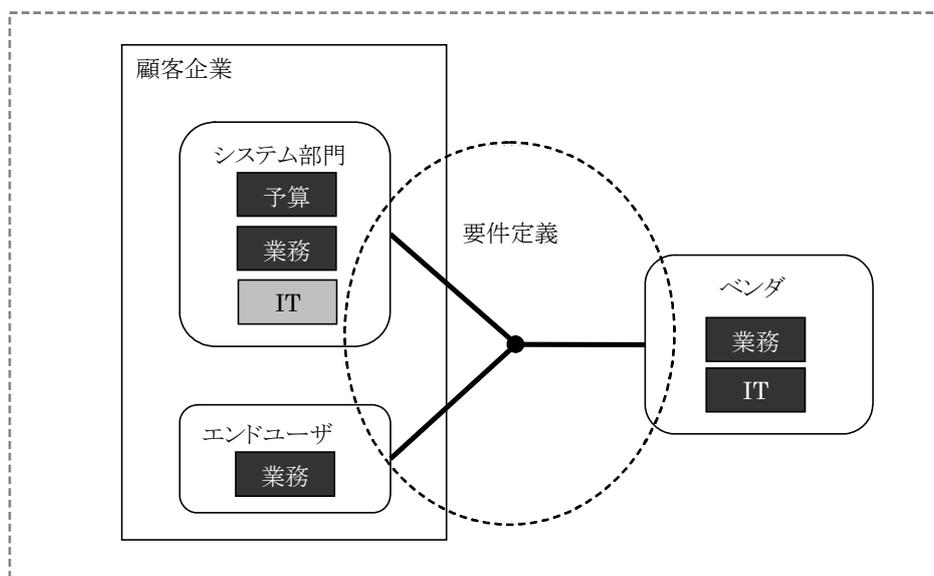


図 8. 2 顧客企業の社内業務システム

(2) 組織

ベンダは、IT スキルは当然持ち、業務知識も十分に持つ。

顧客企業のシステム部門は、業務知識を十分に持ち、ある程度の IT スキルも持つ。更に、システム開発のための予算も持っている。

エンドユーザは、顧客企業内の組織である。

(3) 要件定義の経緯

要件定義の際には、顧客企業のシステム部門とベンダが共同でエンドユーザからのヒアリングを行った。システム部門は、エンドユーザからの各要件に対して、費用対効果を判断した。その判断には、ベンダも協力している。費用対効果が小さいと判断した場合は、その要件を却下した。その結果、システム開発費用を当初予算内に収めることができた。

(4) 考察

要件の増加を抑える有力な対策の一つは、費用対効果が小さいものを排除することである。但し、費用対効果の判断は簡単ではなく、費用を評価するには IT スキル、効果を評価するには業務知識が必要となる。

本事例の顧客企業のシステム部門は十分な業務知識を持っていた。また、IT スキルについてはベンダの協力を得ることができたため、費用対効果を正しく判断することができた。更に予算も保有していたため、要件を的確にコントロールすることができた。

予算負担部門が費用対効果を判断できる能力を持っていたことが、大きな成功要因と言える。

8.5. 事例 3:異なるアーキテクチャで再構築

(1) システムの背景

本事例は、既に稼動している顧客企業の社内業務システムを、老朽化などの理由により、異なるアーキテクチャで再構築するものである(図 8.3)。

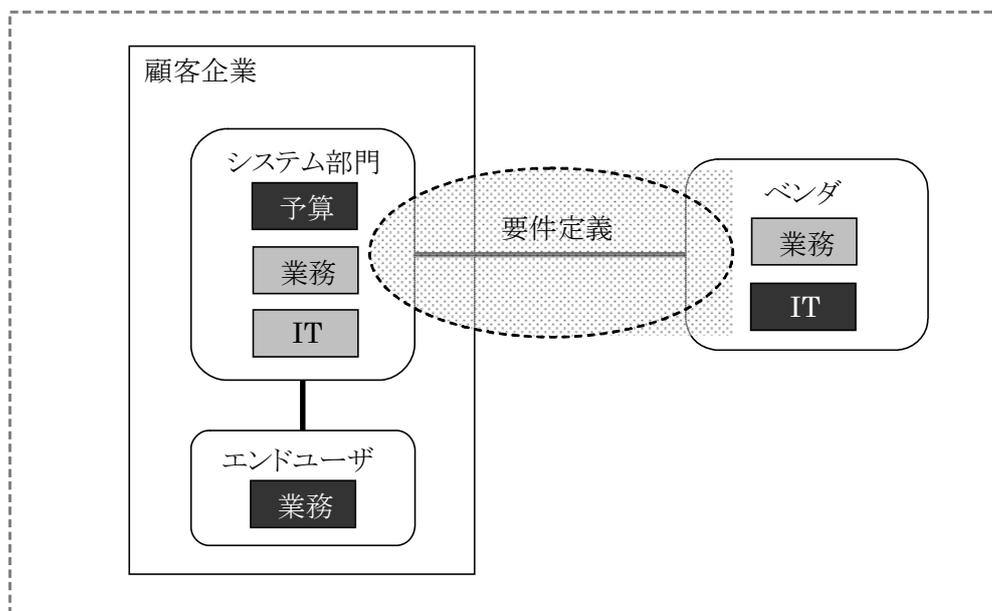


図 8.3 異なるアーキテクチャで再構築するシステム

(2) 組織

ベンダは、IT スキルは当然持つが、業務知識は不十分である。

顧客企業のシステム部門も、業務知識は不十分である。また、IT スキルも不十分である。但し、システム開発の予算は持っている。

エンドユーザは、顧客企業内の組織である。

(3) 要件定義の経緯

基本的な要件は、「機能はそのままに現行システムを再構築」というものであった。新たな要件は無いという考えだったため、エンドユーザからのヒアリングは行わなかった。現行機能の仕様は、現行システムのドキュメントやソース等を参照することにより、難航したものの明確化した。

システム開発時、現行システムと新システムとのアーキテクチャ差異を吸収するため、様々な工夫や処理の追加などが必要になった。そのため、当初予定よりも工数が増大した。

(4) 考察

前事例でも述べたように、工数の増大を防ぐ鍵は費用対効果の判断である。ベンダが IT スキルを保有しているため、費用を判断することはできた。しかし、要件定義に関わった組織、すなわちシステム部門とベンダのいずれも業務知識が不十分だったため、効果を判断することができなかった。

要件定義時にエンドユーザを巻き込み、業務要件を改めて明確化すると同時に、システム部門およびベンダが業務知識を吸収する必要があったと考えられる。

8. 6. 事例 4: 複数部門にわたる社内業務システムの再構築と統合

(1) システムの背景

本事例は、既に稼動している顧客企業の複数の社内業務システムを、再構築して統合するものである(図 8.4)。

(2) 組織

ベンダは、IT スキルは当然持ち、ある程度の業務知識も持つ。

顧客企業のシステム部門は、ある程度の業務知識や IT スキルを持つ。また、システム開発のための予算も持つ。

エンドユーザは、顧客企業内の複数の組織である。

(3) 要件定義の経緯

要件定義の際には、顧客企業のシステム部門とベンダが共同でエンドユーザからのヒアリングを行った。エンドユーザは複数の組織に跨るが、ヒアリングは組織毎に行った。ある組織から出た要件の中には、別の組織から出た要件と競合するものもあった。そのため、要件の調整は難航し、

機能が複雑化してしまった。結果として、当初予定よりも工数が増大した。

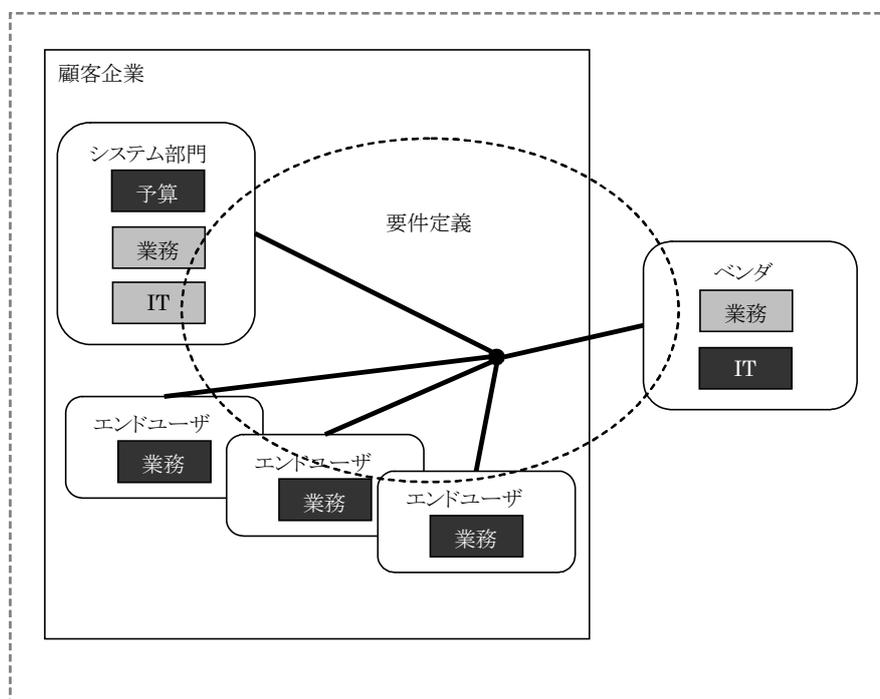


図 8.4 再構築および再統合する社内業務システム

(4) 考察

本事例のシステムのエンドユーザは複数の組織に跨るため、要件も複数の業務に関わるものがあった。そのため、一つの組織業務を対象とするシステムに比べて、より高い業務知識が要件定義に必要となる。顧客企業のシステム部門およびベンダがある程度の業務知識を持ち、更にエンドユーザからのヒアリングも行ったが、要件定義に必要なレベルの業務知識までは得られなかったと考えられる。

エンドユーザにより深く参画して頂けば、より深い業務知識を得、要件の増加を抑えることができたと考えられる。具体的には、エンドユーザ自身に業務分析を行なって頂く、エンドユーザの組織間で業務についてディスカッションを行なって頂く、といったことが挙げられる。

8.7. 事例 5: 社内業務システム

(1) システムの背景

本事例のシステムは、顧客企業の社内業務システムである(図 8.5)。

(2) 組織

ベンダは、IT スキルは当然持つが、業務知識は不十分である。

顧客企業のシステム部門は、業務知識も IT スキルも不十分である。

エンドユーザは、顧客企業内の組織であり、システム開発のための予算を持っている。

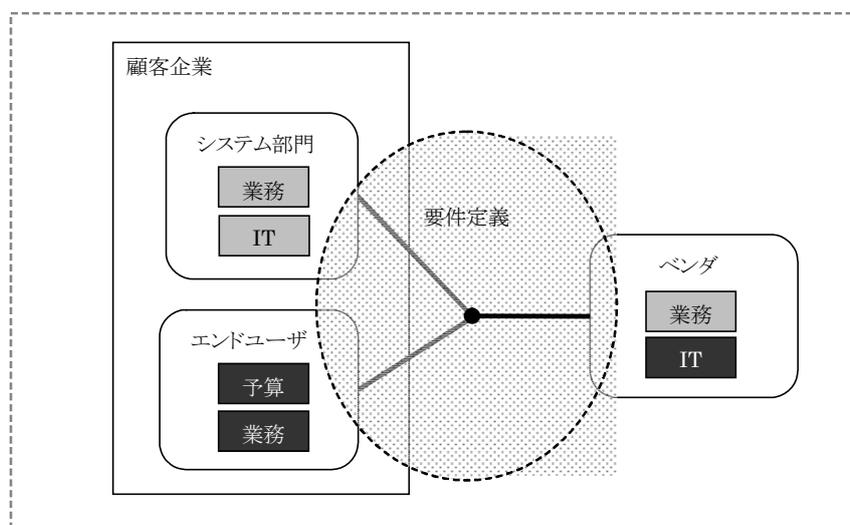


図 8.5 顧客企業の社内業務システム 2

(3) 要件定義の経緯

要件定義の際には、顧客企業のシステム部門とベンダが共同でエンドユーザからのヒアリングを行った。当初想定よりも要件が増加する傾向だったため、システム部門とベンダは調整を試みたが、エンドユーザの合意を得られなかった。

最終的には増加した要件に基づいてシステムが開発されたが、保守性、運用性、拡張性は低いものとなってしまった。

(4) 考察

本事例ではエンドユーザが予算を持っているため、要件の最終的な決定権もエンドユーザが持っていた。しかし、エンドユーザは IT スキルを持っていないため、業務的な要件以外については適切な判断ができなかった。

顧客企業においては、IT スキルを持つシステム部門に、要件をコントロールできるだけの権限を持たせるべきだと言える。

8.8. まとめ

ここまで見てきた通り、要件定義の推進体制はプロジェクトにより様々である。そして、それによって要件定義の進め方や結果は大きく変わってくる。

プロジェクトには様々な事情があり、理想的な体制を築くことはなかなか難しい。しかし、体制の特性や弱点を理解していれば、最大限の対策を取ることができる。本稿がそのための一助になれば幸いである。

－ 禁 無 断 転 載 －

要求開発・管理ベストプラクティスとその体系化の調査研究
－要求開発・管理の現場力を高めるノウハウの共有－

平成 19 年 3 月発行

発行所: 社団法人 情報サービス産業協会

〒135-8073 東京都江東区青海 2 丁目 45 番 TIME24 ビル 17 階

TEL (03) 5500-2610 FAX (03)5500-2630

— 禁 無 断 転 載 —

要求開発・管理ベストプラクティスとその体系化の調査研究
— 要求開発・管理の現場力を高めるノウハウの共有 —

平成 20 年 3 月発行

発行所: 社団法人 情報サービス産業協会

〒135-8073 東京都江東区青海 2 丁目 45 番 TIME24ビル 17 階

TEL (03) 5500-2610 FAX (03)5500-2630