

18-J008

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

平成 19 年 3 月

社団法人 情報サービス産業協会

概要

平成 18 年度 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

本調査では、開発の現場で要求工学を実践したベストプラクティスをもとに、組織的な取り組みのノウハウを整理し、「要求工学実践パターン」としてまとめた。その結果、わが国の要求工学の実態やビジネス慣行を踏まえた場合、ユーザとベンダとを仲介する「ファシリテータ主導要求開発パターン」が優れた実践パターンであることが明らかとなった。

1. 要求工学を現場で活かすために

情報システム開発の最上流工程である要求分析工程は、最も困難な工程であるにもかかわらず技術の成熟度が低いことから、最もリスクの高い工程となっている。実際、JISA の受注ソフト開発実態調査や米国の開発実態調査¹において、情報システム開発プロジェクトの成否に影響を及ぼす要因の上位を占めるのは、要求分析工程に起因するものばかりである。要求分析工程はプロジェクトの成否にかかわり、かつ、情報システムの差別化と付加価値の源泉であるといえる。

要求工学は要求分析に関わる諸技術を合理的、かつ、体系的、組織的に行うための体系である。

その対象範囲は、ハードウェアとソフトウェアを含めた情報システム全体のみならず、顧客のビジネス分析をも含むことから、従来のソフトウェア工学の枠組みを超えた幅広い技術やスキルが要求される。

図 1 に要求工学の知識体系を示す。要求工学の技術的側面を「要求開発」として、管理的側面を「要求管理」とする。さらに、「要求開発」は、要求獲得、要求分析、要求仕様記述、要求検査の4つの要素技術から構成される。各要素技術にはさまざまな技法が提案されている。

この知識体系に基づいた要求工学プロセスの例を図 2 に示す。要求仕様書に関しては、IEEE の標準 (IEEE/ANSI830-1993) が

策定されている。要求工学を実行するプロセスは標準化されたものはないが、図 1 の要求開発の各要素技術を順次実行し、段階的に繰り返すプロセスは概ね理解できるものである。

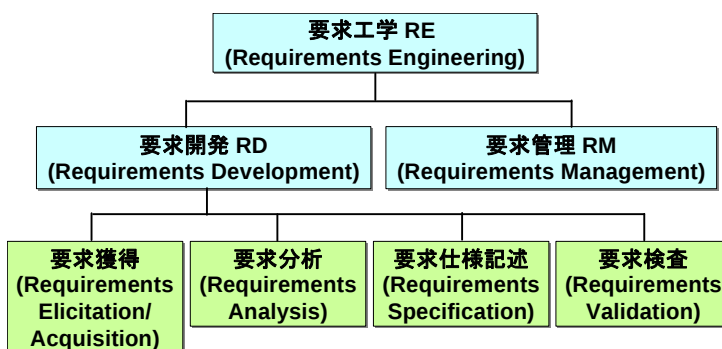


図 1 要求工学の知識体系

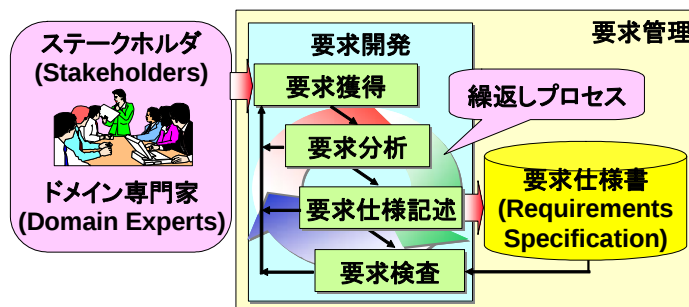


図 2 要求工学のプロセス例

¹ Standish Group, http://www.standishgroup.com/sample_research/PDFpages/chaos1994

上記のことから、要求開発プロセスの実践には相応以上の知識量と組織的な取り組みが必要であることが理解いただけよう。反面、このプロセスの困難さを克服することは、顧客満足度を高め、リスクを減らし、他社と差別化できる「付加価値の源泉」を手にするともいえる。

このような背景から、本調査委員会では、我が国における要求工学の実践状況とソフトウェア業界でのビジネス慣行に鑑み、要求工学の知識体系などの技術的基盤にたつて現場での実践を重視することとした。そのため、委員各社の現場で実践したベストプラクティスを通じて、組織的な取り組みのノウハウを整理し、「要求工学実践パターン」としてまとめた。パターン化するにあたって、次の点に注目している。

(1) 組織形態とその役割

(2) 開発形態

(3) 分析対象範囲

特に(1) 組織形態とその役割に着目し、3つのパターンを抽出した。

(1) ファシリテータ主導

ベンダとユーザの仲介者となるファシリテータを置くことにより、要求開発を円滑に行う。

(2) ユーザ主導

要求開発のスキルを持つユーザの場合、ユーザが主導して要求開発を行う。

(3) ベンダ主導

ユーザが要求開発のスキルを持たず、かつ、ファシリテータが参画できない場合、あるいは、ユーザがベンダに継続的な開発を望む場合など、ベンダが主導して要求開発を行う。

この中の「ファシリテータ主導要求開発パターン」は、現在多く見られる「ベンダ主導要求開発パターン」から段階的に脱却する、優れた要求開発実践パターンであると考えられる。要求開発を改革する手がかりとして導入・実践をお勧めしたい。

2. ユーザ主導による要求開発パターン事例

ユーザ主導による要求開発パターンのベストプラクティスでは、ユーザ企業の情報システム部門などが継続的に要求開発を主導する点に特徴がある。

事例として、損害保険業務システムを事例として取り上げ、社会情勢と同システムとの関わりの歴史をたどりながら、ビジネスシステムにおける要求定義のノウハウを紹介する。

損害保険業務システムは 1960 年代から運用が始まり、モータリゼーションの発展とともに、システム規模が拡大発展を遂げた。1970 年代には、コンピュータセンターの中で閉じていたシステムが業務の現場に「コンピュータ端末」として設置されるようになった。さらに、1980 年代にはオンラインシステムが導入され、「コンピュータ端末」は事務処理を担うすべての場所に設置されるようになった。その後、1990 年代のインターネットの普及とともに、社員一人一人がコンピュータシステムに接続できるようになった。現在では、システムの利用者は社内に留まらず、代理店や顧客も含み、端末数にして数十万台規模の情報システムとなっている。このことは一企業の情報システムが社会基盤化したともいえ、多様な利害関係者(ステークホルダ)の存在を意識せずにはいられない。

このような情報システムを取り巻く状況にあつて、情報システムを新たに企画するには従来のような情報システム開発を無視した経営戦略立案や稟議方式では、開発コストの見積りさえうまくいかない経営戦略策定段階で、情報システム開発にかかる条件を折り込む必要がある、これを「超上流」

工程と言い、発注者となる企業に必須の工程である。

また受注者である情報サービス企業と発注者との間にも、情報システムに対する大きな意識の乖離が見られる。前者は実現するシステム、すなわちどのように情報システムを作り上げるかが重要であり、後者はシステムが実現するビジネス、すなわちビジネスに貢献する情報システムが重要である。発注者はその情報システムが「できた」といえる条件を、たとえば範囲、深さ、粒度などの局面から詳細に記述し、情報システム開発の目的を共有しなければならない。これを「ベースライン」といい、変更管理も常に「ベースライン」に基づいて実施する。

情報システムにおける要求定義は「発注者と受注者が協力して行う」ことが成功の鍵であり、決して要求定義を受注者に任せきりにしてはならない。使える情報システムとはこのようなものである、と伝えない限り、受注者は作るべき情報システムがほとんどわからないと言って良いだろう。

3. ベンダ主導による要求開発パターン事例

JISA で実施しているソフトウェア開発の実態調査からも分かるように、我が国における情報システム開発の過半数のプロジェクトで、要求開発がベンダ主導で行なわれてきた。ベンダ主導による要求開発では、要求開発のリスクを主としてベンダ側が担うことになる。したがって、ベンダ主導要求開発パターンのベストプラクティスでは、リスクを低減するために、担当者個人の知識や経験などによらない組織的要求開発の仕組みに特徴がある。

ベストプラクティスとして、要求開発の社内標準を作成し、それに基づき要求開発を行なっている事例を紹介する。

まず、「ソフトウェアの要求仕様書に関する指針 (IEEE/ANSI 830-1993)」を参考に、自社独自の組織的ソフトウェア開発ガイドライン「グリーンブック」中で「要求仕様書作成要領」を策定し、これに基づき要求分析・要求獲得等を行っている事例を紹介する。当社ではかねてより知識の形式知化を進めていたが、1990 年代初頭からはソフトウェア開発のプロセス化にも取り組み、改善サイクルを回しながら「グリーンブック」を改定している。「グリーンブック」では要求定義プロセスを 3 段階に分け、各プロセスを次のように定めている。

- (1) 導出: 問題の理解 (要求の獲得に関する事項とプロセス)
- (2) 記述: 形式的描写 (要求の表現や用語の統一)
- (3) 確認: 合意形成 (要求のインスペクションと承認)

これらの標準は社内でも活用されてきた。しかし、昨今の情報システム開発は短納期、低価格が求められており、本作成要領の大幅な改定の必要に迫られている。

次は、DOA (Data Oriented Approach) を利用した手法の事例である。情報システム開発においてはデータに着目した DOA での分析が有効である。しかし、DOA のみで要求分析することは困難なため、別にステークホルダ管理や構成管理、インタビューテンプレートなどのツールが必要である。

4. ファシリテータ主導による要求開発パターン事例

ファシリテータ主導要求開発とは、発注者と受注者の間で仲介役 (ファシリテータ) が要求開発を仲介することにより、要求開発を円滑に進める方法である。

業務系情報システムの要求定義を進める上で、うまく要求獲得をできない理由の一つに、顧客にも理解可能なプロセスや形式的な技法がないことがある。下記に例示するツールを含め要求分

析や要求定義ツールを利用することで、発注者と受注者の間の仲介役（ファシリテータ）は効果的な要求分析を進めることができる。ここでは 3 つの事例を紹介する。

第一の事例は、業務要求とシステム要求の相互関連性を可視化した「業務×システム機能マトリックス」である。開発者はこのマトリックス構造に基づきドキュメントを作成し、記述レベルに一定の基準を設けることで、形式的かつ正確な顧客要求が定義でき、また顧客の要求を階層的かつ要素ごとに記述できる。マトリックスの一行が顧客の視点から見た基本要求的単位であり、基本要求的単位に詳細要求やソフトウェア要求、課題、議論などを統一的に管理する。列にはシステム機能を配し、基本要求的と機能が交差する点から、その機能に必要なとされる仕様を策定する、という手順を踏んでいく。

第二の事例は、独自の的方法論に基づく要求定義の事例である。同方法論では、要求定義の精度を低くする「不完全な記述」「最適化視点の欠落」「目的が未達成」「課題が不明確」「合意が未成立」という根源的な課題に対して、それぞれ「関係者」「課題」「目的」「手段」「業務」のモデル化技術を提唱する。モデル化のためのプロセスやツールを定義し、実システムへの適用を通じて同方法論の優位性を定量的に評価している。

第三の事例は、新たな人材を登用することで、受発注者間に横たわる要求定義の齟齬をなくするという試みである。この受発注者の間の仲介役を担う人材は「ビジネスエンジニア」と呼ばれる。ビジネスエンジニアは、受注者側の人材でありながら、本来発注者側の作業である「ビジネス戦略策定」や「課題整理／分析」の段階からプロジェクトに参加し、発注者の支援業務を行う。この人材に求められるスキルは非常に広く、ビジネススキル、業務スキル、マネジメントスキル、技術スキルなど、あらゆる面にわたる。また受注者側の人材にもかかわらず、発注者／受注者のいずれからも中立であることを要求される。しかし、ビジネスエンジニアが常に適正な要求定義をできる、という定量的な評価はこれからの課題である。

5. 要求工学分野研究の動向調査

「要求工学研究の世界的動向調査」は、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・センターによって実施された。本調査は、事前調査、文献調査、ヒアリング調査を順次行い、最後に文献調査とヒアリング調査の結果と知見をもとに、要求工学研究の現状を整理した。図 3 に文献調査とヒアリング調査の結果から得られた 8 つの主要技術に関する現状の知見をまとめた。この図を基に要求工学の要素技術の成熟度（現在の状況と実践活用の可能性）を理解することができる。

6. 要求工学分野における最新研究

要求獲得のための先進的な研究として「メタモデルに基づく要求工学手法」を紹介する。

この手法では、「個人は各自の関心に基づいて世界を見る²⁾」ことから「関係者間で相互了解するための関心の根拠となっている構造を示す」という考え方を拠り所に、妥当な要求を導出するために、以下のプロセスを経ることを提案する。

- (1) モデルはインタビューを基に個人毎に作り、
- (2) 主観的な要求の根拠を組織の関係者で共有可能とする

²⁾ 西條剛央：構造構成主義とは何か～次世代人間科学の原理～，北大路書房(2005)

- (3) そこから、関係者の間で、妥当な根拠とその根拠を了解し、
 (4) 組織としてシステムへ求める要求と、その根拠を提示する。

またビジネスにおける以下の点に着目してメタモデルを紹介する。

- ビジネスにおける協調作業とは何か
- ビジネスにおける人や組織の依存関係とは何か
- 人が担う役割、ロールとは何か
- 見解の相違は何に由来するのか
- 人が抱える課題と解は、何によって表せるのか

本メタモデルを満足する生成手法には、SSM(ソフトシステム方法論)、ゴール指向、i*(アイスター)等あるが、本手法ではそれらの断片を選択し、適所に配置し統合している。また統合の段階で既存の手法では満足できない部分には、新たなモデルを提案して補足し、手法全体を再構成する手法を採用している。

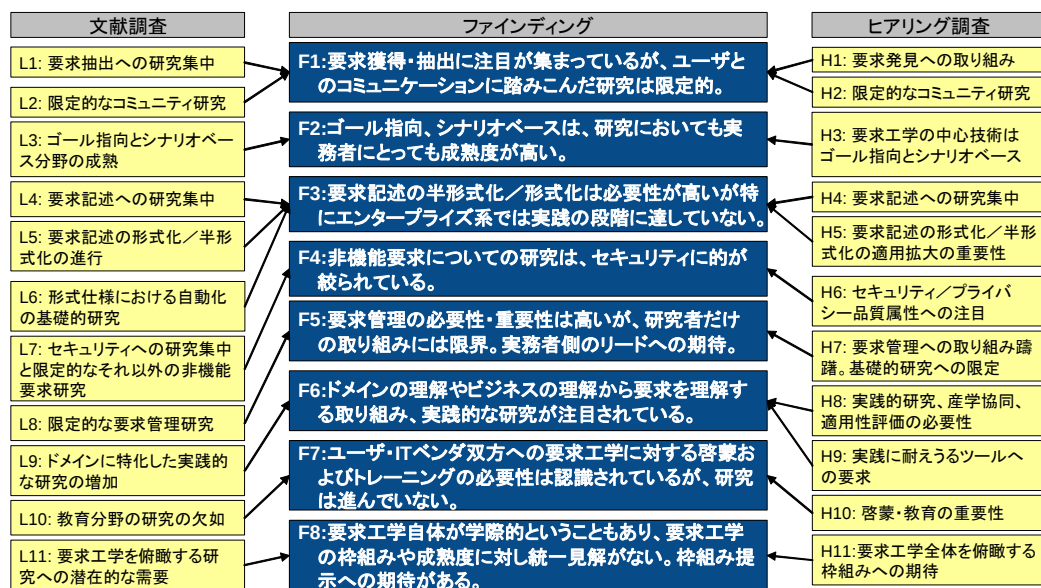


図3 要求工学分野研究の動向調査: 文献調査とヒアリング調査結果の整理

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分析とその技術体系である要求工学について、現場の実践と技術動向の両面から調査・検討した報告です。

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。これに対して、要求獲得、分析、仕様化、検査、管理、などの一連の技術を体系化した要求工学が過去 10 年余りの間に発展して参りました。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、要求工学の実践に視点を置き、現状を踏まえた要求工学の組織的活用には焦点を当てて、調査・検討を実施しました。

現場における要求工学のベストプラクティスの収集に基づき、組織的な取り組みをパターン化し、ベンダ主導、ユーザ主導の 2 つのパターンに加え、ベンダとユーザとの仲介者としてファシリテータが主導するファシリテータ主導パターンを見出しました。ファシリテータ主導パターンでは、第三者、あるいは、ユーザ内やベンダ内であっても、両者の仲介を果たす役割を積極的に組織化することにより、要求工学を円滑に実践します。今後、要求開発を実践する上で有効なアプローチであると言えるでしょう。

本調査では、最初に、要求開発の実践事例を、ユーザ主導、ベンダ主導、ファシリテータ主導の 3 パターンに分けて、整理し、まとめました。さらに、要求工学の研究動向の調査から、要求工学の技術動向と技術の事例も紹介しています。本報告書を通して、実践と基盤技術動向の両面から要求工学を実践するヒントが掴めるでしょう。

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。

また、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 19 年 3 月
(社)情報サービス産業協会
要求工学調査委員会
委員長 青山 幹雄

平成 18 年度 要求工学調査委員会 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	菊島 靖弘	(株)アイネス 金融システム本部 フェロー
	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) 技術開発事業部 事業部長
	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 生産技術強化部 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サービス基盤課 品質技術グループ 主事
	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) 社会基盤システム事業部 社会基盤システム統括部 第4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 主査
	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
	真田 圭司	富士通エフ・アイ・ピー(株) コンポーネント開発センタ
	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 課長
	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	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
	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
事務局	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 18 年度 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 目 次

概要

はじめに

委員名簿

1. 要求工学を現場で活かすために	1
1. 1. 要求工学を取巻く状況	3
1. 2. 要求工学の適切な実践がプロジェクト破綻を防ぐ	3
1. 3. 要求工学とは: 要求工学超入門	4
1. 3. 1. 要求工学の出現と発展	4
1. 3. 2. 要求工学の知識体系	5
1. 3. 3. 要求工学プロセス	6
1. 3. 4. 要求のスコープ: ビジネス要求, (情報)システム要求, ソフトウェア要求	7
1. 3. 5. 機能要求と非機能要求	8
1. 3. 6. 要求獲得技術	9
1. 3. 7. 要求分析技術	10
1. 3. 8. 要求仕様化	11
1. 3. 9. 要求検証	12
1. 4. 要求工学導入の課題	12
1. 4. 1. 要求分析の担務実態と要求工学の実践	12
1. 4. 2. 要求工学実践の課題	13
1. 5. ベストプラクティスに見る要求工学実践パターン	13
1. 5. 1. わが国のビジネス慣行を踏まえた要求工学実践パターン	13
1. 5. 2. ファシリテータ主導要求開発パターン	14
1. 5. 3. ユーザ主導要求開発パターン	15
1. 5. 4. ベンダ主導要求開発パターン	16
1. 6. 要求工学を実践する人材像	17
1. 7. まとめ	17
2. 金融システムにおける要件定義の課題	19
2. 1. 金融システムの課題	21
2. 1. 1. システムの活用は、外へ、外へと広がった	21
2. 1. 2. 多様なステークホルダーの登場	22
2. 1. 3. オープン化の流れ	24
2. 2. 要求から要件へ～要件定義の抱える課題と対策～	25
2. 2. 1. システム開発の超上流	25
2. 2. 2. システム開発の目的	26
2. 2. 3. ベースラインが重要	28
2. 2. 4. 要求から要件へ、要件定義のノウハウ	29

2. 3. 最後に	31
3. 社内標準に基づく要件定義事例	35
3. 1. 構造計画研究所におけるソフトウェア開発の現状	37
3. 2. グリーンブック	38
3. 2. 1. グリーンブック構成	38
3. 2. 2. グリーンブックにおける要件定義プロセス	39
3. 2. 3. グリーンブックにおける要求仕様書詳細	41
3. 3. 要件確定で現在抱える問題点	43
3. 4. 問題への対応	44
3. 5. 期待される要件定義	45
3. 6. 最後に	46
4. 要求定義に関する取組状況	47
4. 1. はじめに	49
4. 2. 当社で適用している開発方法論	49
4. 3. 当社で適用している要件定義手法	50
4. 4. 要件定義工程に関連する諸問題	52
4. 5. 要件定義工程の改善	52
5. 社内事例紹介と要求工学調査の方向性について	55
5. 1. はじめに	57
5. 2. 社内事例紹介	57
5. 2. 1. 要求工学プロセスの課題設定	57
5. 2. 2. 課題に対する解決仮説	59
5. 2. 3. パイロットプロジェクトへの適用効果	61
5. 3. 実践的な要求工学の為に	64
6. 「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	67
6. 1. MOYA 概要	69
6. 2. 適用事例とそのフィードバック	72
6. 2. 1. 事例紹介	72
6. 2. 2. 事例詳細	72
6. 2. 3. フィードバック	76
6. 3. MOYA2.0～MOYA の改善～	76
6. 4. 今後の課題と展望	79
7. 要求定義における価値創生に関する考察	81
7. 1. はじめに	83
7. 2. システムを構築する目的	83
7. 2. 1. システムへの期待	83
7. 2. 2. 要件の確認	83
7. 3. 要求に求められるもの	84
7. 4. ビジネスにおける利害関係	85
7. 5. ユーザの現状	86

7. 6. ビジネスエンジニア(仮称)とは	86
7. 6. 1. 役割	86
7. 6. 2. スキル	87
7. 6. 3. 担当領域	87
7. 6. 4. 位置づけ	88
7. 7. 総括	88
7. 8. 最後に	89
8. 要求工学研究の世界的動向調査	91
8. 1. はじめに	93
8. 2. 事前調査	95
8. 2. 1. 要求工学の研究分野	96
8. 2. 2. 要求工学分類の枠組み	97
8. 2. 2. ヒアリング対象項目	100
8. 3. 文献調査	101
8. 3. 1. 文献調査の準備	101
8. 3. 2. 分析	101
8. 3. 3. 文献調査のファインディング	103
8. 4. ヒアリング調査	103
8. 4. 1. 研究者へのヒアリング調査のファインディング	104
8. 4. 2. 実践者へのヒアリング調査のファインディング	106
8. 5. 文献調査及びヒアリング調査分析の結果	107
8. 5. 1. 要求工学分析の枠組みへの提言	108
8. 6. まとめ	109
9. メタモデルに基づく要求工学手法の開発	111
9. 1. 研究の背景と目的	113
9. 2. メタモデルの開発	114
9. 2. 1. ビジネスにおける協調作業とは何か	114
9. 2. 2. ビジネスにおける人や組織の依存関係とは何か	114
9. 2. 3. 人が担う役割、ルールとは何か	115
9. 2. 4. 見解の相違は何に由来するのか	115
9. 2. 5. 人が抱える課題と解は、何によって表せるのか	115
9. 3. 最後に	116

1. 要求工学を現場で活かすために

南山大学 青山 幹雄

1. 1. 要求工学を取巻く状況

情報システム開発の最上流工程である、要求分析工程が最も困難で、かつ、実務において技術の成熟度が低い。そのため、要求分析が最もリスクの高い工程となっている。一方、これは、要求分析工程が、顧客満足度を高め、リスクを減らし、他社と差別化できる付加価値の源泉となる工程であることも意味している。

例えば、情報サービス産業協会が毎年実施している実態調査を見ると、要求分析は情報システム開発における問題の最大の要因となっている。「情報システム開発プロジェクトの成否に影響を及ぼす要因」の上位3つは、次のようになっている。いずれも、要求分析工程に起因する[3]。

- (1) 要求定義(35%)
- (2) プロジェクト計画(30%)
- (3) 開発体制(15%)

この傾向は、わが国固有のものではなく、世界共通である。例えば、情報システム開発の実態調査として広く知られている、米国の Standish Group の調査がある[7]。この調査では、ソフトウェア開発の問題要因ワースト3は、次のように、要求分析工程に起因するものばかりである。

- (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%)

このように、要求分析工程は、情報システム開発の成否の鍵を握っている。

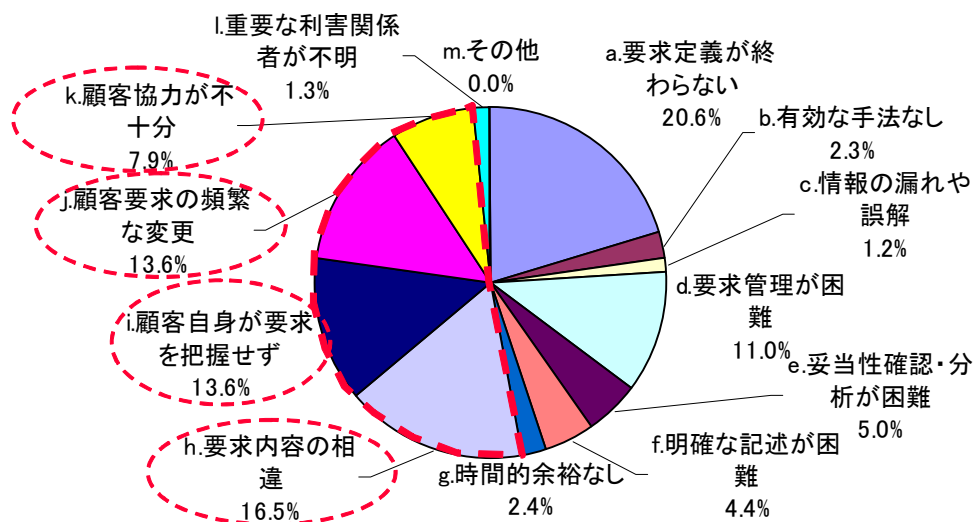


図 1. 1 要求分析の主要な課題

1. 2. 要求工学の適切な実践がプロジェクト破綻を防ぐ

要求分析工程が情報システム開発の主要な要因とすると、問題を回避するには、どうすべきか、その解決のヒントとして図 1. 2 が参考になる。全開発コストに対して要求分析工程に割り当てたコスト

トの比率を横軸に、縦軸に全開発コストの超過率を示したものである。実際には、米国 NASA で調査した多数のプロジェクトのデータがプロットされているが、図に示すように、2 つの特徴的な領域がある。

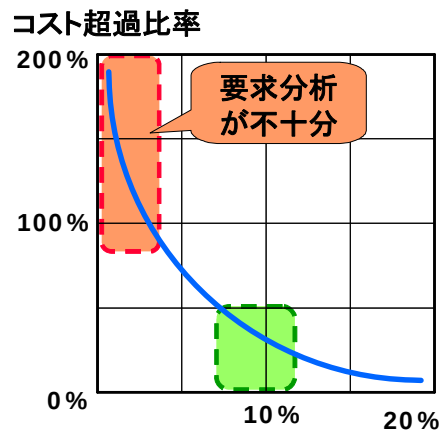


図 1. 2 要求工学の実践とプロジェクト破綻リスク

左上の領域は、要求分析にほとんどコストを割いていないプロジェクトの集まりである。この場合、開発コストが予算の 2 倍～3 倍となる確率が高い。一方、右下の領域では、開発コストの 10%程度を要求分析に割くと、ほとんどのプロジェクトでコスト超過が 50%以内に収まっている。

要求工学を適切に実践すれば、開発コストの超過を未然に防止できる可能性が高いことを示している。

1. 3. 要求工学とは: 要求工学超入門

ここで、要求工学(Requirements Engineering)とは、何か、簡潔に紹介しよう。詳しくは、参考文献[5, 8, 9]を参照願いたい。

1. 3. 1. 要求工学の出現と発展

歴史的に見ると、ソフトウェア工学の技術は下流工程のプログラミングから、上流工程の要求工学へと発展してきた。要求分析技術としては、1970 年代後半に、構造化分析・設計 (Structured Analysis/Structured Design) や実体関連モデル (ER Model: Entity Relationship Model) が提唱され、上流工程の要求分析と設計の技術が形成されてきた。その後、1980 年代からオブジェクト指向技術の発展により、上流工程もオブジェクト指向に基づいて分析・設計を行なうオブジェクト指向分析・設計が発展した。あわせて、表記法の UML も標準化された。

一方、要求分析では、顧客からの要求を獲得するために、コミュニケーション技術やファシリテーション技術も重要な技術である。また、要求工学の対象はソフトウェアのみならず、ハードウェアも含む情報システム全体、さらには、顧客のビジネス分析まで遡る。従来のソフトウェア工学の枠組みを超えた幅広い諸技術やスキルが必要である。そのため、要求分析に関わる諸技術を合理的かつ組織的に行なうための体系として、「要求工学(Requirements Engineering)」が提唱

された。1993 年からは、現在の要求工学国際会議 (International requirements Engineering Conference) の源流となる「要求工学」を冠した国際会議が立ち上がった[11]。これ以後、ソフトウェア工学の分野で要求工学が広く認知されるようになった。2004 年には我が国で要求工学国際会議が開催され、国内でも要求工学への認識が広まった。

1. 3. 2. 要求工学の知識体系

要求工学は顧客の要求を抽出し、分析し、要求仕様にまとめる一連の技術体系とする点で共通に理解されているが、その知識体系のまとめ方にはいくつかある。

ソフトウェア工学の知識体系(SWEBOK)では、要求工学とその関連技術をまとめて図 1. 3 に示すように体系化している[1]。

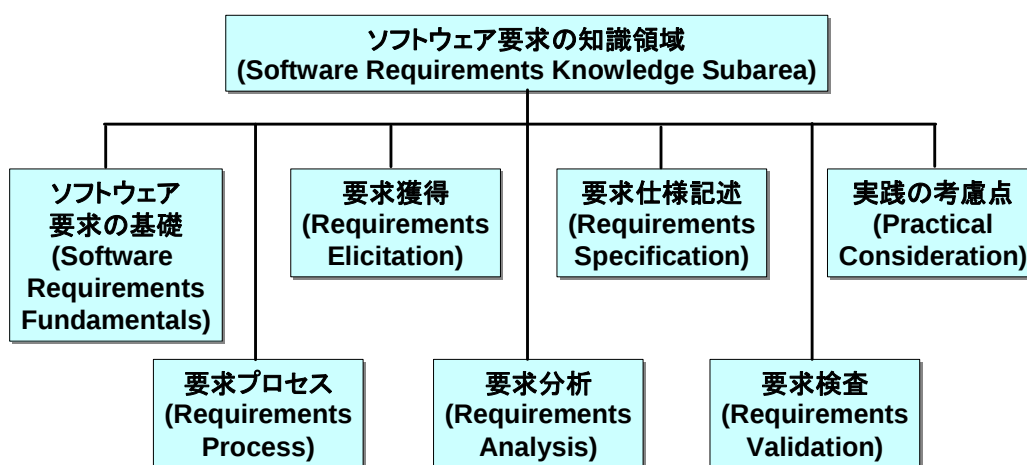


図 1. 3 ソフトウェア工学知識体系における要求工学の知識体系

要求工学を適用する上で要求管理も必要となる。そのため、図 1. 3 と同様であるが、要求工学の技術的側面を要求開発としてまとめ、要求工学の知識体系を要求管理として図 1. 4 に示す枠組みが理解しやすい。本報告書ではこれに基づいて説明する。

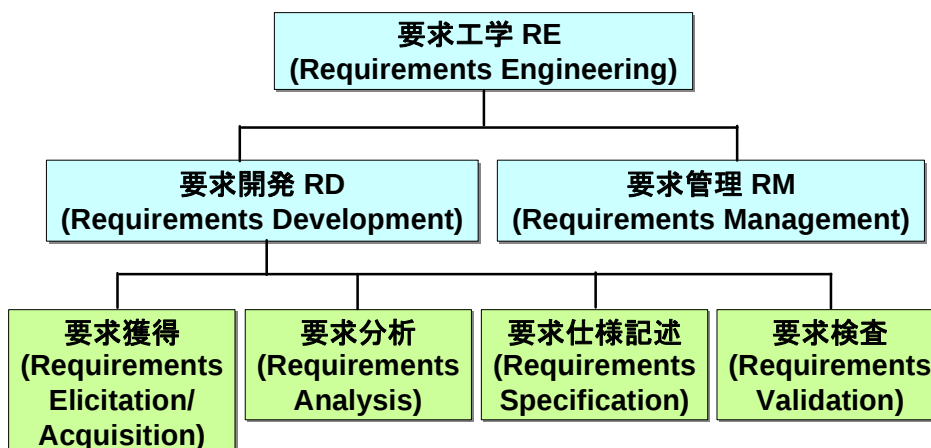


図 1. 4 要求工学の枠組み

要求開発は次の4つの要素技術から成る。

- (1)要求獲得:ステークホルダ(顧客を含む)を明らかにし、その代表から要求を獲得する。
- (2)要求分析:獲得した要求の構造(要求の項目とその間の関係)や特性を分析し、必要な要求を抽出する。
- (3)要求仕様記述:分析した要求内容を仕様書として記述する。
- (4)要求検査:要求仕様が顧客の要求を満たし、それ自体が正しいか妥当性を確認する。

1. 3. 3. 要求工学プロセス

要求工学の要素技術を用いる要求開発プロセスの例を図1.5に示す。要求開発プロセスの標準形が確立されているわけではないが、上記の要求開発の4つの要素技術を順次実行し、かつ、段階的に繰り返すプロセスは、一般におおむね理解できるものである。

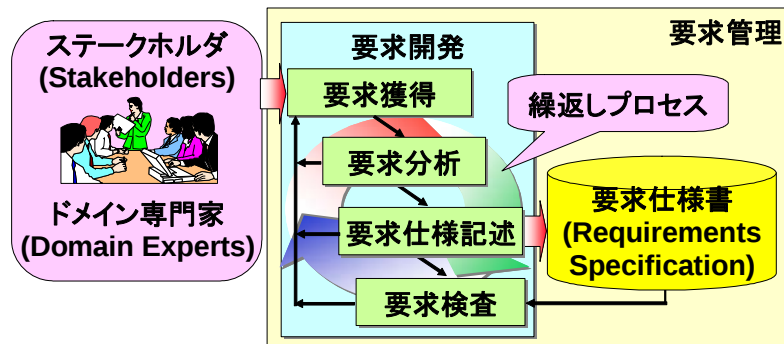


図 1. 5 要求工学のプロセス例

図1.6は、要求工学の諸技術をスパイラルプロセスに沿って実行する例[6]を示す。

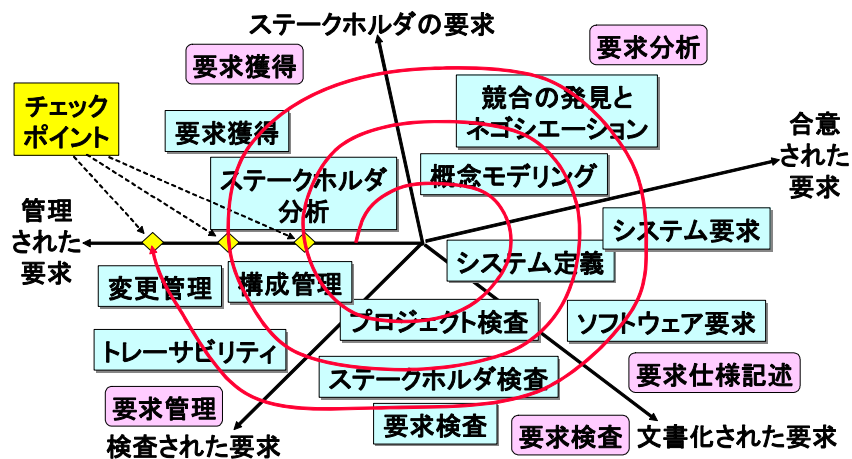


図 1. 6 要求工学のスパイラルプロセスの例

1. 3. 4. 要求のスコープ：ビジネス要求、(情報)システム要求、ソフトウェア要求

要求分析の対象範囲は、例えば、エンタープライズシステムでは、図 1. 7 に示すように、ビジネス、(情報)システム、ソフトウェアの 3 段階で捕らえることができる。

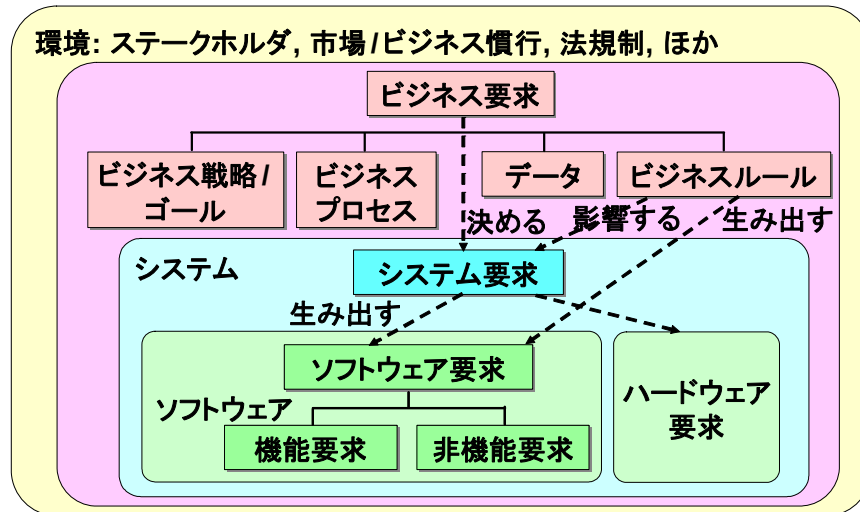


図 1. 7 エンタープライズシステムの 3 段階の要求境界

(1)ビジネス要求:対象とするビジネスに対する要求である。いわば、顧客の要求となる。ビジネス要求には、次のような要素が含まれる。

- 1) ビジネス戦略/ビジネスゴール:要求の根源となる企業のビジネス戦略やそれを具体化し、要求が満たすべきビジネスゴール。
- 2) ビジネスプロセス:ワークフローなどのビジネスを実行するプロセス。
- 3) データ:ビジネスで扱う諸データ。
- 4) ビジネスの諸活動が満たすべきビジネスルール。

(2)(情報)システム要求:対象とするビジネスの中で、情報システムとして実現すべき要求。ハードウェアに対する要求やソフトウェアに対する要求。

(3)ソフトウェア要求:情報システムの中で、ソフトウェアで実現すべき要求。

要求の要素として何があり、それらの関係がどうなっているか、予め一般化しておく、要求開発で押さえるべき点が明確になり、要求開発を体系的に行なえる。このような一般化したモデルは、要求のモデルという意味で、メタモデルと呼ぶ。要求の土俵を定義することである。文献[10]を元に作成したビジネス要求のメタモデルの例を図 1. 8 に示す。

この図の解釈は、例えば、「ビジネスは市場などの状況の中で、目的を達成するために、責務(責任)をもった職務が機能を発揮すること」となる。ビジネス要求の構成要素とその間の関係を示している、ビジネス要求を考える上で考慮すべき事柄とその関係を示していると言える。

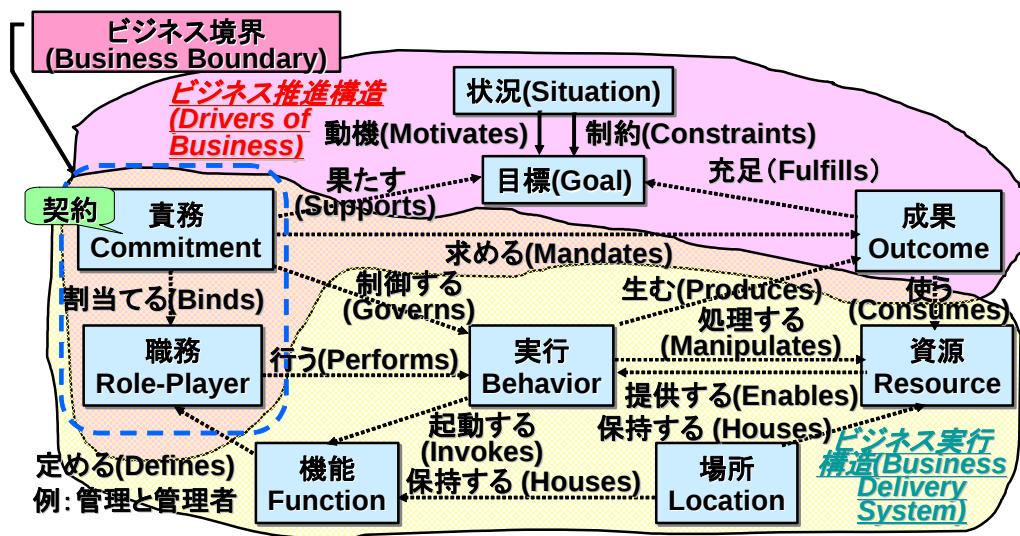


図 1. 8 ビジネス要求のメタモデル

1. 3. 5. 機能要求と非機能要求

要求は図 1. 9 に示すように、機能要求(FR: Functional Requirements)と非機能要求(NFR: Non-Functional Requirements)に分けることができる。

機能要求とは、ユーザに対して果たすべきことから、サービスである。何を行なうかという意味である。ワープロを例とすれば、編集、印刷、保存、などが機能要求である。

一方、非機能要求は「機能要求以外の要求」という意味であるが、品質要求と言っても良い。例えば、信頼性、性能、ユーザビリティ(使い勝手)など、機能をいかに提供するかという意味である。

機能要求は、ワープロの編集、印刷、保存のように、個々の働きとしてモジュール化しやすい。一方、非機能要求は、図 1. 9 に示すように、機能を横断して、システム全体として満たすべき性質であることから、モジュール化が困難である。このような非機能要求をソフトウェアの構造として実現するものをアスペクト(Aspect)と呼ぶ。さらに、オブジェクト指向に加えて、アスペクトに基づいてソフトウェアの構造を捉える方法がアスペクト指向(Aspect-Oriented)として提唱されている。

同一の機能要求であっても、信頼性や性能など非機能要求は異なる。このような非機能要求の違いは、システムやソフトウェアの制御の機構によることから、システムアーキテクチャやソフトウェアアーキテクチャが非機能要求を支配すると考えられる。同一の機能要求を満たすソフトウェアアーキテクチャは複数ありえるが、その主要な違いは、非機能要求にある。このことから、要求開発プロセスを通して機能要求と共に非機能要求の重要性も認識すべきである。

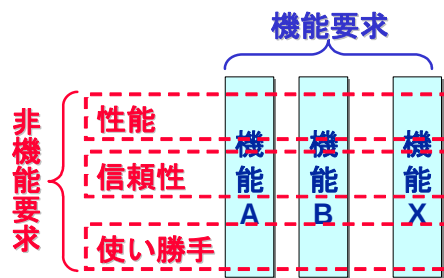


図 1. 9 機能要求と非機能要求

非機能要求の具体例として、表 1. 1 の ISO/IEC-9126「ソフトウェア品質特性」が参考になるだろう。

表 1. 1 ISO/IEC9126 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), 整合性 (Compatibility), セキュリティ (Security)
信頼性 (Reliability)	成熟性 (Maturity), 障害許容性 (Fault Tolerance), 回復性 (Recoverability)
ユーザビリティ (Usability)	理解性 (Understandability), 習得性 (Learnability), 操作性 (Operability)
効率 (Efficiency)	実行効率 (Time Behavior), 資源効率 (Resource Behavior)
保守性 (Maintainability)	解析性 (Analizability), 変更容易性 (Changeability), 安定性 (Stability), 試験容易性 (Testability)
移植性 (Portability)	適応性 (Adaptability), 移植容易性 (Installability), 規格準拠性 (Conformance, 置換性 (Replaceability))

1. 3. 6. 要求獲得技術

要求獲得技術は要求の源泉の明確化と獲得方法から成る。

(1) 要求の源泉

要求の源泉にはシステムの関係者やドキュメントなど多様である。主要な源泉として、例えば、次のようなものがある。

1) ゴール (Goal)

ビジネスゴールなど、要求が果たすべき目標。個別の要求を抽出するに先立って、ゴールを明確にすることが重要である。

2) ステークホルダ (利害関係者)

ビジネスや情報システム、ソフトウェアシステムに関与する利害関係を持つ個人、グループ、組織をいう。要求の主要な源泉である。しかも、ステークホルダはエンドユーザだけでなく、ユーザ企業の経営者や監督官庁も含まれる。そのため、場合によっては、要求に関与する主要なステークホルダを洗い出し、その間の関係や重要度を明らかにするステークホルダ分析が必要となる。

3) ドメイン知識

業種や業務などの情報システムやアプリケーションの適用範囲をドメイン(Domain)と呼ぶ。あるドメインには、固有の前提条件や共通の要求がある。このような情報をドメイン知識と呼ぶ。ドメイン知識は、ドメイン内での「常識」として要求仕様書の記述には省略されるものもあり、その記述には個人や組織などの属人性が強い場合がある。適切なドキュメント化をして共有することが重要である。

4) 環境

ビジネス、情報システム、ソフトウェアシステムを遂行あるいは実行する環境など、要求を取巻く環境に関する情報。

(2) 要求獲得方法

要求獲得方法は、対象システムの特質や要求の源泉によりさまざまな方法がある。特に、ステークホルダである人を対象として要求獲得を行う場合、コミュニケーション能力が重要となる。主要な要求獲得方法として、次のような方法がある。

1) シナリオ分析

シナリオとは、ビジネスプロセスの手順やユーザがシステムを利用する操作手順など、物事の流れを指す。どのように業務を遂行するかなどを、日本語ではあるが、一定の記述形式で記述したものである。記述方式の例としては、UMLで用いるユースケースの記述(シナリオと呼ぶ)がある。情報システムの専門化ではないエンドユーザなどと要求開発者とが共通に理解できる言葉で記述することが重要である。

2) インタビュー

要求開発者がステークホルダにインタビューする。インタビューは簡易な方法であるが、インタビューする人の能力や、インタビューする人とインタビューされる人との人間関係など、属人性が入り込みやすい。そのため、インタビューする人の人選、インタビューの方法などに留意する必要がある。

3) ファシリテーション、あるいは、ファシリテーションミーティング

インタビューがステークホルダと要求開発者との二者間での要求獲得方法であるのに対し、ファシリテーションは、仲介役としてファシリテータを交えて、三者により要求獲得を円滑に行おうとする方法である。特に、ユーザなどのステークホルダと要求開発者、あるいは、ステークホルダ間の利害の対立や要求の競合が起こりやすい場合、利害の調整を要求獲得の段階から行うことにより、要求獲得を円滑に行うことを狙いとする。

4) 観察

ユーザやステークホルダのビジネスプロセス遂行の状況、システムの利用などを観察することにより、要求を獲得する方法。

1. 3. 7. 要求分析技術

(1) 要求分析の要素技術

要求分析は、次のような技術により、要求の要素とその間の関係を明らかにすることにより、要求の構造を明らかにすることである。

1) **要求の分類整理**: 獲得した要求の要素を整理分類し、要求要素間の関係や重要性を明らかにする。

2) **要求の構造化**: 要求要素間の関係に基づき、要求の要素とその間の構造化を行う。

3) 要求の重要性和境界の明確化: 要求要素の優先度や対象とする業務やシステムの範囲を明らかにする。必要であれば、関係者とネゴシエーションする。

(2) 複数の視点から要求を分析する

要求抽出では、実世界の「もの」「こと」の中から、ビジネスや情報システム、ソフトウェアに関わる要素が抽出される。この「もの」「こと」には、様々な要素が混在する。これを分析するためには、多様な「もの」「こと」を複数の視点に分けて分類することにより、複雑度を軽減し、問題の本質を掴み取ることが容易となる。

このような複数の視点として、一般に、構造の視点、機能の視点、挙動の3つの視点がある。

1) 構造(Structure)の視点: ビジネスやシステムを構成する要素とその間の関係。ソフトウェアでは、一般に構成要素はデータとして表現されることから、データとその間の関係として捉えられる。例えば、企業の組織構成。

2) 機能(Function)の視点: システムやその構成要素が果たす機能。例えば、営業、経理、開発などの企業の構成組織が果たす機能。

3) 挙動(Behavior)[あるいは振舞い]の視点: ビジネスやシステムが機能を発揮するために、活動を起こす契機となるイベントなどの要因と活動や実行の形態。

1. 3. 8. 要求仕様化

要求仕様化とは、要求分析結果を適切な表現形式でドキュメント化することである。一般に複雑で多様な要求を1つの表現形式だけでドキュメント化することは困難である。そのため、分析で用いた、機能の視点、構造の視点、挙動の視点の3つの視点に分けてドキュメント化することにより複雑度を軽減し、正確な理解が可能となる。具体的な仕様の表現として、UMLでは、この3つの視点に分けてドキュメント化できる[2]。

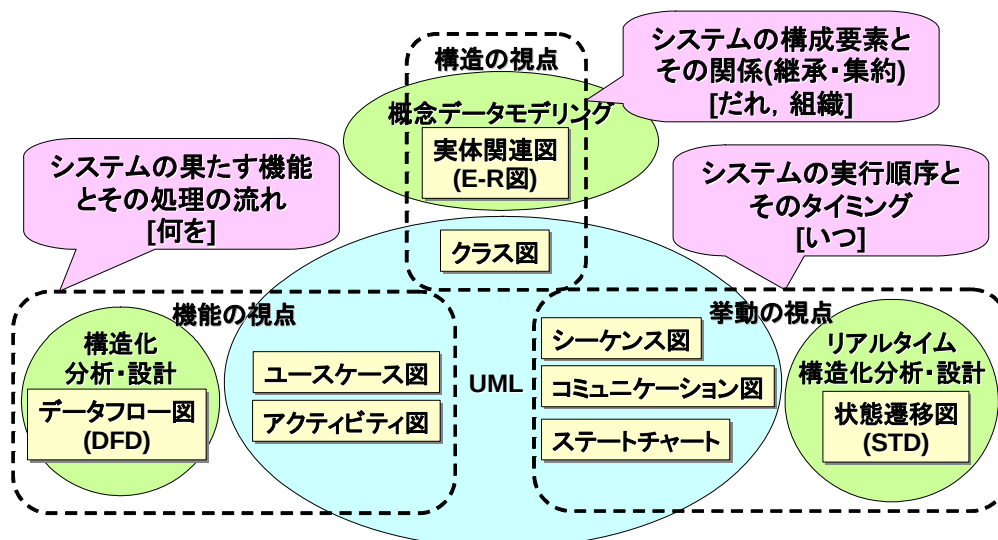


図 1. 10 複数の視点による要求仕様のドキュメント化

図 1. 10 は、構造化、概念データモデリング、UML の 3 つのドキュメント化の体系を示す。UML では、この 3 つの視点でドキュメント化できるよう考慮する。

1. 3. 9. 要求検証

要求検証(Requirements Validation)[要求の妥当性確認とも呼ぶ]は、要求仕様が顧客などの適切なステークホルダの要求を正しく定義していることを確認することである。主な要求検査の方法として、次のような方法がある。

1) レビュー

レビューは要求分析を含むあらゆる工程で有効な検証・検査方法である。要求分析で特に留意すべき点は、適切なステークホルダがレビューに参画していることである。

2) プロトタイピング

要求仕様の一部を仮に開発し確認する方法である。GUI など、人間の感性によるため、ドキュメント化が難しい要求に適用される方法である。

1. 4. 要求工学導入の課題

本調査委員会では、要求工学の導入と実践に取り組んでいる企業の事例調査を通して、要求工学の諸技術を現場で活用するための課題と実践の秘訣を探った。要求開発という上流工程では、技術面だけでなく、我が国独特のビジネス慣行の影響が強い。

1. 4. 1. 要求分析の担務実態と要求工学の実践

現場で要求工学を実践する上で大きな課題となっているのは、わが国の情報システム開発のビジネス慣行において、多くの場合、ベンダが要求分析を行っているという実態にある。図 1. 11 は、2006 年度に実施した「要求開発における顧客とベンダの役割分担の実態調査」の結果である。

この図からも明らかなように、約 60%のプロジェクトでベンダが主として要求開発を行っている。

これは、我が国においては受託開発中心のソフトウェア開発形態とそれに伴うビジネス慣行を背景として、要求分析のスキルがベンダ企業に蓄積され、多くの場合、ユーザ企業には蓄積されてこなかったことにあると考えられる。

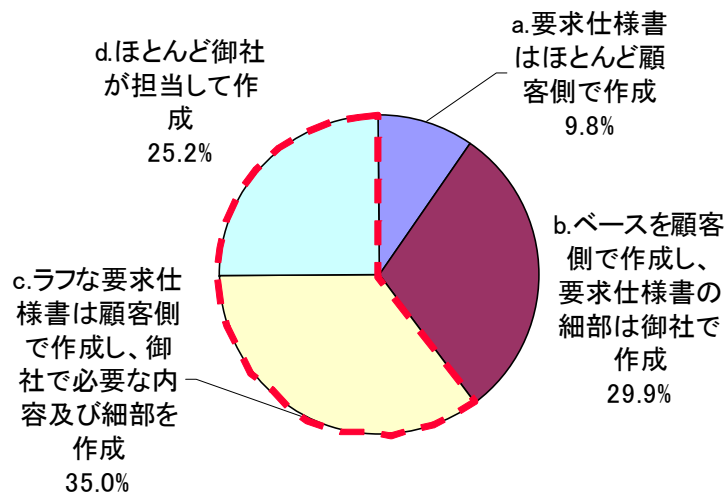


図 1. 11 要求開発における顧客とベンダの役割分担

1. 4. 2. 要求工学実践の課題

図 1. 11 に示す要求開発の実態は無視しえないので、このような状況にあることを踏まえて、要求工学を実践するためには、要求工学に関して次のような技術開発と実践のための体制作りが必要となる。

- 1) **要求工学の認識と理解**: 要求工学が必ずしも認知されていないことから、要求工学の正確な「理解」と普及を組織的に推進する必要がある。この場合、要求工学技術の中で、各企業の置かれる状況や開発対象などによって、適切な要求工学技術を組織内に導入する必要がある。
- 2) **ビジネスを考慮した要求工学の実践の仕組み作り**: 顧客企業とのビジネスの形態や顧客企業の要求工学に関するスキルに応じて、適切な実践の組織形態をとる。本報告書では、これを「要求工学実践パターン」と呼ぶ。
- 3) **要求工学を実践する高度 IT 人材の育成**: 要求工学の実践の形態や方法に応じて、必要なスキルを持つ人材の育成。要求工学を実践する人材は、IT 人材の中でも最も高度な人材と位置づけることができる。さらに、1. 3. 4 節で述べたように、対象とする、ビジネス、情報システム、ソフトウェアによって、人材が必要とするスキルが異なる点にも留意する必要がある。

1. 5. ベストプラクティスに見る要求工学実践パターン

本調査委員会では、現場で要求工学を実践した、いわば、ベストプラクティスを調査し、要求工学を実践するための組織的取り組みのノウハウを整理した。この結果を、「要求工学実践パターン」としてまとめて提示する。

1. 5. 1. わが国のビジネス慣行を踏まえた要求工学実践パターン

本調査委員会では、わが国の要求工学の実態を踏まえ、要求工学を現実的に適用する形態をパターン化した。パターン化にあたって、次の点に着目した。

(1) 組織形態とその役割

要求分析を主導する組織の形態。ユーザとベンダ、さらに、その間を仲介する役割を「ファシリテータ」と呼ぶことにする。

(2) 開発形態

受託開発かパッケージ化を行う開発か。

(3) 分析対象範囲

ビジネス、情報システム、ソフトウェアのどの範囲を対象とするか。

特に、組織形態に着目し、次の3つのパターンを抽出した。この中で、1.4.1で示した、現状のベンダ主導の要求開発を改善し、要求開発を円滑に進めるベストプラクティスとして、「ファシリテータ主導」パターンが実践されていることが明らかとなった。

表 1. 2 要求工学実践の 3 パターン

パターン	意味	組織構造
ファシリテータ主導	ベンダとユーザの仲介者となるファシリテータを置くことにより、要求開発を円滑に行なう。	
ユーザ主導	要求開発のスキルを持つユーザの場合、ユーザが主導して要求開発を行なう。	
ベンダ主導	ユーザが要求開発のスキルを持たず、かつ、ファシリテータが参画できない場合、あるいは、ユーザがベンダに継続的な開発を望む場合など、ベンダが主導して要求開発を行なう。	

表 1. 3 は本委員会で調査した事例を上記の 3 つのパターンに分類した結果を示す。

表 1. 3 調査対象のベストプラクティス

パターン	開発形態	ドメイン	事例報告
ファシリテータ主導	パッケージ	公共	(株)NTT データ[MOYA]
	受託開発	流通	(株)野村総合研究所
	受託開発		三菱総研 DCS(株)
ユーザ主導	受託開発	金融	東京海上日動システムズ(株), (株)アイネス
ベンダ主導	受託開発		(株)構造計画研究所
	受託開発		(株)菱友システムズ

1. 5. 2. ファシリテータ主導要求開発パターン

(1) 要求開発の問題点

ユーザが自ら要求開発を行なうほどには要求工学のスキルが十分でないが、要求が複雑であったり利害関係が単純でないなど、要求開発に困難が予想される。

(2) 解決策

図 1. 12 に示すように、仲介者としてファシリテータを置くことにより、適切な要求工学の技術の活用や利害関係の調整を行なえる。



図 1. 12 ファシリテータ主導要求開発パターン

(3) 必要な人材

要求工学に関して一定のスキルを持ち、ユーザとベンダの間で仲介役を果たす人材。但し、このような人材が少ないことから、専任チームとは限らず、ベンダ、ないし、ユーザの要求開発の関係者の中から、一定期間に、ファシリテータの役割を果たす責務を割り当てる。ユーザの情報システム部門は、エンドユーザとベンダとの間にあって、本来的に、ファシリテータの役割を果たしえる位置にある。

(4) ベストプラクティス

NTT データの MOYA(Model-Oriented methodology for Your Awareness)の事例では、要求工学の諸技術を組織的に実践する専門チームを置くことにより、個々のプロジェクトが個別に実践するには技術の習得や人材の点で困難であった要求工学の組織的实践を可能とした。この事例では、ファシリテータが必要な要求工学の技術の導入・実践の推進役となっている。

野村総合研究所の事例において、要求工学のスキルを持った専門家が参画することにより要求開発の技術が高まり、要求開発が円滑に進んでいる。

(5)実践のヒント

ファシリテータの育成、ならびに、ファシリテータの参画と協力して要求開発を実践する体制作りが必要である。

1. 5. 2. ユーザ主導要求開発パターン

(1) 要求開発の問題点

ユーザのビジネスの根幹に関わり、かつ、競争力の源泉となるコアビジネスプロセスの要求分析は当該ビジネスに関する高度な知識のない外部のベンダ等には分析が困難であり、かつ、競争力の観点からユーザ社内で分析をしたい。

(2) 解決策

図 1. 13 に示すように、ユーザの情報部門など、ユーザ組織が中心となって要求開発を行なうことにより、ユーザの高度なビジネスのノウハウへのアクセスが可能で、かつ、ビジネスに関するノウハウなどが企業内に蓄積できる。

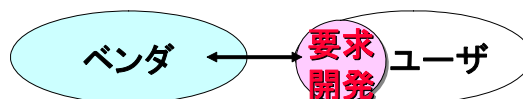


図 1. 13 ユーザ主導要求開発パターン

(3) 必要な人材

ビジネス分析などの要求工学の技術と当該ビジネス分野に関するビジネスの知識を持つ

こと。

(4) ベストプラクティス

東京海上日動システムズの事例では、長年、東京海上日動の情報システム部門としての役割を果たしてきたことから、当該業務のビジネス知識の蓄積とあわせて、ユーザ主導の要求開発が実践されてきた。

(5) 実践のヒント

本パターンで求めている人材像は極めて高度な要求分析ができるかどうか、確認が必要である。

技術：技術優位となるユーザの視点からのスキル獲得やビジネスモデリング、業務(ドメイン)知識、ビジネスとITとのマッピング、コミュニケーション/ネゴシエーション能力などが必要である。また、組織的位置づけ、トップマネジメントの参画(CIO の設置など)、経営のリーダーシップが必要である。また、情報システム部門の位置づけ、権限[部門の位置づけなど]を明確にする。

また、情報システム部門をこのように位置づけ、ビジネスの変革などを実行する戦略部門へと脱皮させることが可能となる。

1. 5. 3. ベンダ主導要求開発パターン

(1) 要求開発の問題点

ユーザに要求開発のスキルや技術が不足していたり、中小企業など、今後も要求分析の専任者を置くことが難しい。

(2) 解決策

多様なユーザの要求分析を行なえるために、ベンダ内で、ユーザの要求や IT に関する知識や理解の不足を整理し、再利用できるように、作業標準の策定、ハンドブックなどを作成する。

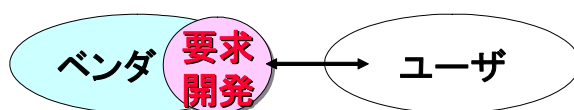


図 1. 14 ベンダ主導要求開発パターン

(3) 必要な人材

開発と関係づけてユーザの要求を理解し、一定の方法で分析、使用できる。

(4) ベストプラクティス

構造計画研究所では、要求開発のプロセスやドキュメント化などを体系化し、かつ、マニュアルとして社会で配布し、成果を挙げている。

(5) 実践のヒント

技術：要求獲得[ユーザ視点の導入、ステークホルダ分析など、]等の、特に、ユーザの視点を含む要求工学関連のドキュメントの書籍化。契約などでのリスクを回避する管理(ファシリテータ, PMO)など。

1. 6. 要求工学を実践する人材像

前節までのベストプラクティスに基づく要求開発パターンの例から、「要求工学を実践する人材像」が浮かびあがってきた。要求工学の対象を、ビジネス、情報システム、ソフトウェアの 3 段階に分けると、ビジネス要求分析など、従来の要求工学の枠を超えていることから、スキルを明確にし、人材の育成を図る必要がある。

また、下流工程が海外を含むアウトソースの対象となりつつある現状を見ると、上流工程の要求分析を担当しうる高度 IT 人材の育成は急務である。

1. 7. まとめ

我が国における要求分析/要求工学の実態調査と委員会での事例紹介/ヒアリングを通して、我が国における独特のビジネス慣行を背景として、要求工学の実践が緒に就いたばかりであることが明らかとなった。

さらに、ベストプラクティスの分析を行なった結果、要求工学を無理なく現場で実践するための組織形態を「要求開発実践パターン」としてまとめた。この中で、「ファシリテータ主導要求開発パターン」は、現在のベンダ主導要求開発パターンから段階的に脱却し、すぐれた要求開発を実践するパターンと言える、現在の要求開発を変革する手がかりとして導入・実践を頂きたい。

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] 青山 幹雄, 中谷 多哉子(編), オブジェクト指向に強くなる, 技術評論社, 2003.
- [3] IPA/三菱総研, ソフトウェアプロセス改善に関する調査, 2002 年 3 月, <http://www.ipa.go.jp/NBP/13nendo/13SPI/H13SPI-rep-index.html>
- [4] 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.
- [5] 大西 淳, 郷 健太郎, 要求工学, ソフトウェアテクノロジーシリーズ, 共立出版, 2002.
- [6] P. Sawyer, Software Requirements, R. H. Thayer, et al. (eds.), Software Engineering, 3rd ed., IEEE Computer Society, 2005, pp. 123-139.
- [7] Standish Group, http://www.standishgroup.com/sample_research/PDFpages/chaos1994
- [8] K. E. Wiegers (渡部洋子(監訳)), 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.
- [9] 山本 修一郎, 要求を可視化するための要求定義・要求仕様書の作り方, ソフトリサーチセンター, 2006.
- [10] R. Youngs, et al., A Standard for Architecture Description, IBM Systems J., Vol. 38, No. 1, 1999.
- [11] Home Page of International Requirements Engineering Conference (RE), <http://www.requirements-engineering.org/>.

2. 金融システムにおける要件定義の課題

(株)アイネス 菊島 靖弘

金融の世界は、いまやシステムの利用なくして業務を行うことができないほど、システムに依存した産業です。銀行も、証券も、生命保険も、損害保険もシステムによって発展し、支えられているといつてよいでしょう。

本稿では、損害保険会社を例に過去40年間を振り返り、システム開発が現在抱えている課題を明らかにした上で、課題解決の鍵の一つである要件定義について考察してみたいと思います。

2. 1. 金融システムの課題

2. 1. 1. システムの活用は、外へ、外へと広がった

損害保険会社における事務の機械化は、20 世紀初頭における PCS(パンチカードシステム)という会計機の導入から始まります。その後、商用コンピュータが開発され、1961 年に業界最大手の東京海上火災保険に IBM1401 というコンピュータが導入され、本格的なシステムの業務活用がスタートしたのです。

1960 年代のシステムはバッチシステムといいました。どこの会社でも、処理したいデータを、紙の原票(例えば保険申込書)の束という形(バッチ)でトラックに積み、主に大都市郊外のコンピュータセンターに送り込みました。コンピュータセンターではその紙の原票をカードパンチ会社にまわし、コンピュータが処理できるカードにパンチしたのです。カードにパンチされた情報は、コンピュータにまとめて読みこませ、電子データに変換し、電子化されたデータは磁気テープに契約マスターとして蓄積したり、決算用のデータを作ったり、統計表を作表するなどして活用しました。作表された帳票類はもちろんトラックで配送されたのです。

その後、1970 年代に入り自動車の普及が進みますと、自動車は移動するためのものですから、保険の支払いを必要とする事故が、移動先、つまり日本中で起こるようになりました。当然、事故を起こした場所で保険サービスを提供する必要がありますので、保険の加入場所に拘わらず、日本中どこでも、事故を起こした方が加入している保険の内容を把握できるようにしなければなりません。オンラインで保険契約をどこであつても把握し、保険金の支払いサービスが円滑にできるようにするため、1974 年に自動車保険の損害支払い処理サービスオンラインシステムが稼働しました。この時初めて、企業の奥の院の感じがあったコンピュータセンターからコンピュータ端末が損害保険会社の事務オフィスに飛び出したのです。台数は数百台でした。

1980 年代になりますと、銀行業界では第 3 次オンラインシステムがスタートします。損害保険業界でもオンラインシステムを用いた業務体制の構造改革が進みました。端末は日本中の営業所に導入され、契約の照会だけではなく、営業第一線から保険契約のエントリーを行うようになりました。バックオフィスの大変革が起こりました。かつて、営業から送られた大量の保険の申込書を、手作業で処理するバックオフィスは消えてしまいました。今でいう BPR(Business Process Re-engineering)が行われたのです。損害保険会社の第一線に配置された端末数は数千台になりました。こうして、コンピュータの端末を保険契約の事務処理ラインにそって、保険会社の隅々まで、誰もが使うようになったのです。

1990 年代は、言うまでもありません、インターネット技術が登場した時代です。一人ひとりがコンピュータでつながるようになりました。損害保険会社だけではなく、どの業界でも、社員

一人ひとりにパソコン端末が配られ、電子メール、電子マニュアルなどが導入され、「情報リテラシー」という言葉が飛び回ったのは記憶に新しいことです。端末数は社員の数を上回り、大企業では数万台という数になりました。場所から個人へ、全ての社員がコンピュータでつながりコンピュータを使うようになりました。

そして、21 世紀、ついにネットワークは会社の外に飛び出しました。損害保険会社の契約は代理店のオフィスから入力されます。端末の数は数十万台になりました。イントラネットで保険会社と顧客企業がつながります。成田の端末で海外旅行保険に加入できます。いまや、一企業のシステムを企業の外の人たちが使う時代です。銀行、証券などのシステムが止まったら、社会活動が停止してしまいます。システムは一企業の枠を超えて社会基盤化しました。

そして、システムの利用者はシステムの利用拡大にしたがって拡大し、その要求は利用の多様化にしたがって、高度化、複雑化したのです。

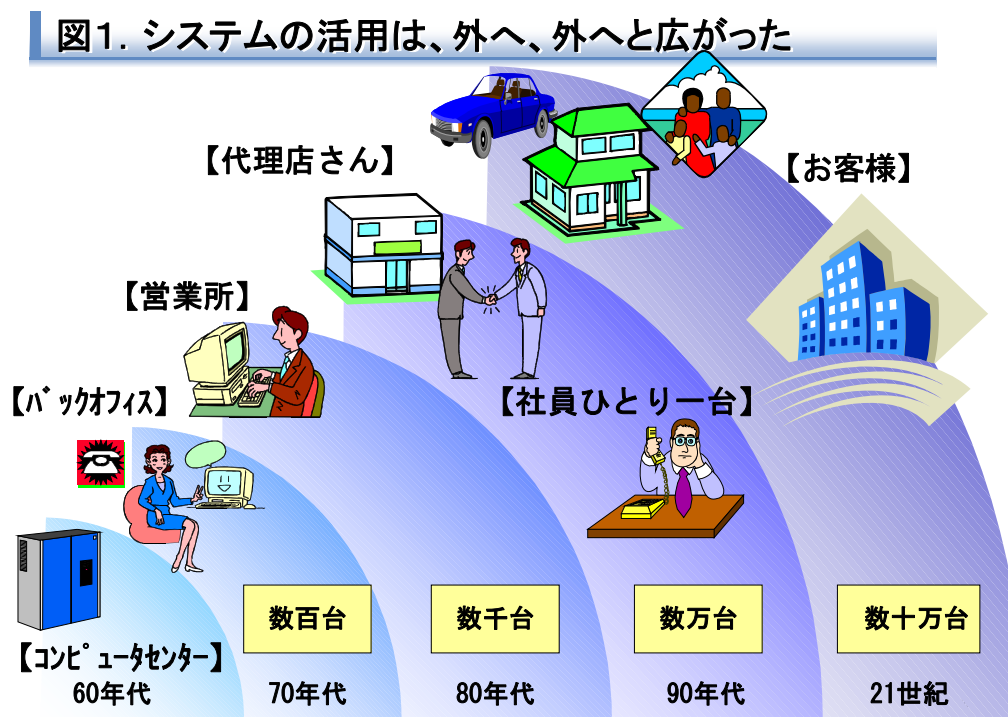


図 2. 1 システム活用は外へ、外へと広がった

2. 1. 2. 多様なステークホルダーの登場

システムの利用者が増えるにしたがって、システム開発を行う際のステークホルダーも増えました。30 年前、システム開発をする際、例えば新しい営業統計表を作りましょうなどといった場合、ステークホルダーといえ、営業統計表を作って、営業部門の管理レベルを上げようとする営業推進・企画部門と、システムを開発するシステム部門くらいでした。

その営業統計表を実際に使って仕事をする多くの営業部門の管理者と担当者が、システム開発に関与するなどということは皆無であった、といってよかったでしょう。

もちろん、業務をオンラインによって刷新する、などといった場合、つまり、大きな投資と、会社

業務の構造改革を行うようなときには、経営層を含む社内の各部門が参加するシステム開発が行われたことはいうまでもありません。

現在のシステム開発には、こんな大掛かりな話でなくとも、様々なステークホルダーが登場します。

まず、社外でシステムを使う人たちがいます。インターネットで個人のお客さんが一企業の開発したシステムを家庭で使います。お客様企業とはイントラネットで接続されました。例えば確定拠出年金の拠出検討サービスはオフィスで可能です。自動車保険はディーラーで自動車と一緒に売ってもらうのですが、その販売の仕組みも保険代理店の仕組みとして提供され使われています。そういった仕組みを提供するようになれば、使い方などについてのヘルプデスクも開設しなければなりません。

システム開発において、また運用においてアウトソーシングを活用することは今日一般的なことです。例えば、特殊な「はがき」を作る場合、印刷業者にデータを渡して印刷してもらうことになります。そうなれば、個人情報保護の観点から、暗号や、データの管理、場合によってはプリント時の不良作製物の廃棄管理までシステム開発の要件として検討しなくてはなりません。

様々な利用者が登場し、システム開発における要件定義の出発点である要求は、システム利用の広がりとともに、捉えにくい、不確実さ、曖昧さを多く含むものになりました。

(簡単にいえば、いろいろな所で、いろいろな人が使い、いろいろなことを言うようになったということです。)

図 2.2 は、損害保険会社のシステム開発を行うとき、考慮に入れなくてはならないステークホルダーを粗く洗い出したものです。これだけでもたくさんいるな、と感じますが、実際はもっと複雑で「代理店さん」一つとってみても、保険をディーラーで売る場合、銀行窓口で売る場合、一般代理店で売る場合、お客様企業の団体保険で売る場合等々、さまざまな売り手、売り方が登場するのです。

図2. 多様なステークホルダーの登場

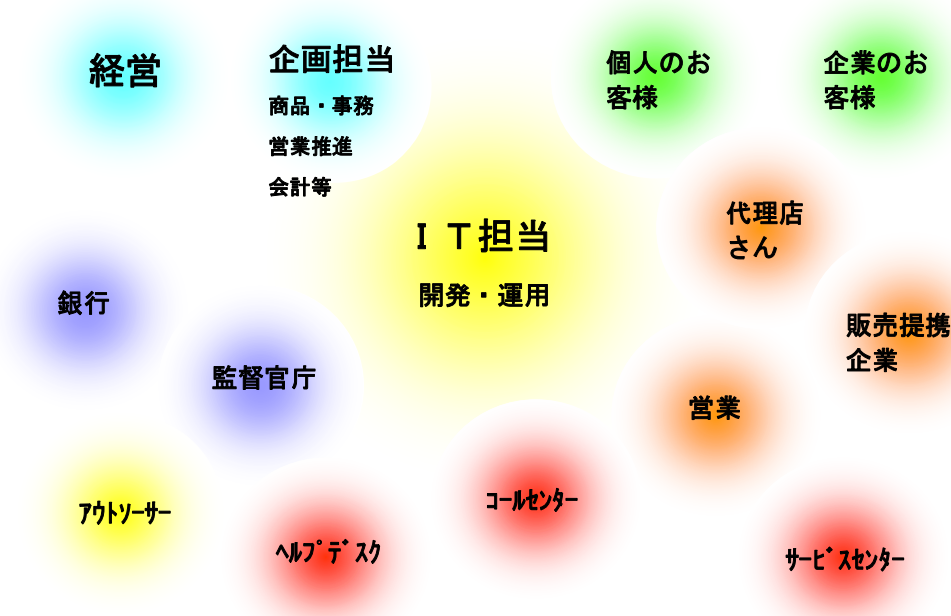


図 2. 2 多様なステークホルダーの登場

2. 1. 3. オープン化の流れ

システム開発技術も、難しくなりました。

メインフレームといわれるコンピュータでバッチ処理を行っていた時代、事務の世界では、コボルや、アセンブラといった言語を駆使してプログラムロジックを構築するというのがシステム技術でした。ダイクストラ先生の構造化技法の本や、ワーニエ先生の本を読んで、人間がそれまでやっていた事務処理や、契約の規定などの処理をコンピュータにのせることがシステム開発の目的であったのです。しかし、次のオンラインの時代になると、さすがにプログラム知識だけではシステム開発ができなくなり、オンライン OS とネットワークの技術が活用されるようになりました。回線や、端末といったインフラにかかる知識が必要になりました。今日ではインターネット技術を使わずしてシステム構築はできないでしょう。

ネットワークの時代、プログラム技術だけでは、事務システムといえどもシステム開発はできません。ソフトウェアからハードウェアまで、あらゆるシステム開発技術を駆使することが要求されます。このことがシステム開発を難しくしました。

問題は、こういった技術がリプレイスする形で変遷しなかったことです。損害保険業（おそらくは金融一般）については、バッチの時代に構築した、事務、保険契約などを処理する中核システムが現在でも 50%以上を占めます。この部分は保険業務、保険契約の処理そのものですから、変える必要もなかったのです。続くオンライン対応から後の部分は、この中核部分に重なるように追加され開発されました。外へ、外へと広がりながら、広がったところで使われた技術が、オンラインの技術であり、インターネットの技術であったのです。

今日、一つの保険商品を開発しようとなると、メインフレーム、サーバー、ネットワーク、端末

等々全てのフィールドに対し、それぞれの技術を使ってシステム開発を行います。すべてについてわかっているシステムエンジニアはそう多くはありません。多様化し、高度化した技術。要求を要件に落とし、さらにシステム構築の設計図につなげていくのが、技術面でも難しくなったというわけなのです。

2. 2. 要求から要件へ ～要件定義の抱える課題と対策～

2. 2. 1. システム開発の超上流

図 2. 3 をご覧ください。

企業でシステム開発を行うとき登場するプレイヤーです。企画・稟議の線に対応している担当責任部門がそれぞれの役割、職務権限にしたがって、業務上必要な企画を稟議決裁にかけます。新しい事業分野への進出とその手段としてシステムの活用を打ち出す場合などなら、経営層、経営企画などから付議されますし、システム開発環境などにかかる投資であれば、システム開発部門から起案されます。

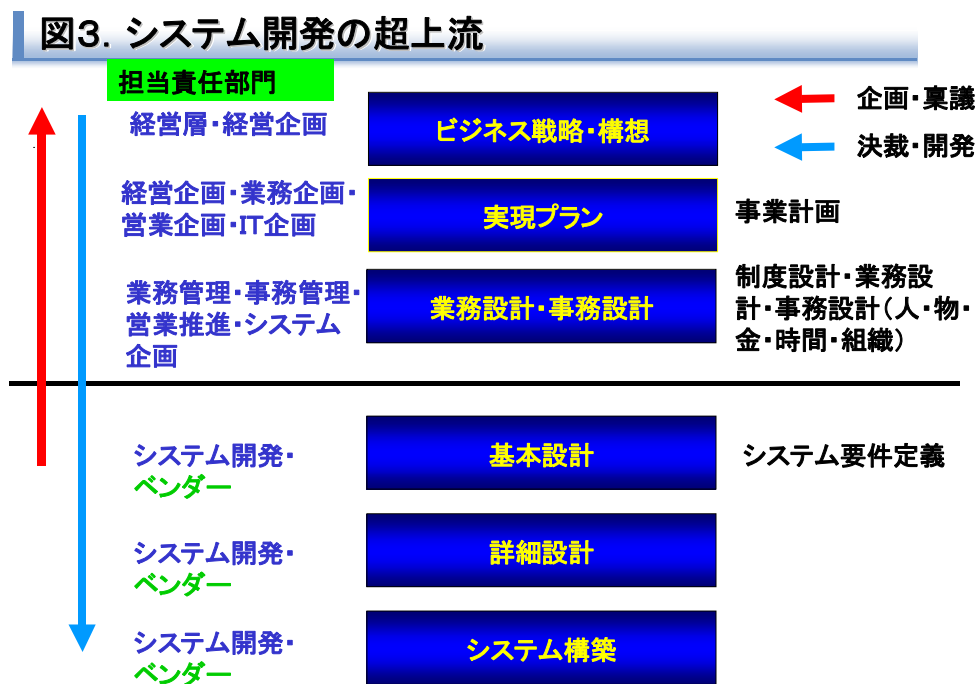


図 2. 3 システム開発の超上流

ところで一昔前には、この企画・稟議決裁の工程において、システム開発が吟味されることは、実際のところあまりありませんでした。例えば、損害保険の新商品を出そうというとき、稟議書にシステム開発のことが触れられることは、ほとんどなかったといってよかったでしょう。あっても、それほど詳細に、システムが議論の対象になることはなかったと思います。

今となっては笑い話ですが、まだシステム開発子会社ができる前、当時のシステム開発とシステム開発部門がどういうものだったか、実感できる話をご紹介します。

筆者はシステム部門にいたのですが、夜、他部門の同期入社社員と飲んでいて、新商品の開発を聞き、これは統計システムの手当てが必要になると判断し、早速翌日、商品開発部門に問い合わせたところ、まだ社外秘だからもう少し待っていて欲しい、といわれたことがあります。私は社員だったのですが?!

閑話休題。

お話したいことは、昔は、ビジネス上の企画を稟議決裁するとき、システム開発にかかるリソース、つまり人(システム開発要員の手配はできるか等)、物(今までの環境でできるか等)、金(いくらかかるか)、時間(いつまでにできるか、いつ発売できるか)や、ビジネスがどこまで実現できるかといったことが、多くの場合、大きな問題にはならなかったのです。たとえ問題になったとしてもホストコンピュータ上のアプリケーションプログラム開発だけの話でした。

今は違います。すべての仕組みがシステムにのってしまっていますから、何をするにもシステム開発が必要です。それも、アプリケーションプログラムだけにとどまる開発は、重要なものほど少数派でしょう。アプリケーション開発さえ、レガシーといわれる保有システムに、まったくうんざりするほど、山のように修正をかけなければなりません。お金も山のようにかかります。ともかく、システム開発できる範囲でしか、何も実現しないのです。したがって、稟議決裁のとき、判断材料としてシステム開発にかかる条件が極めて重要になります。システム開発にかかる見積もりがこの段階で必要になったのです。

これが、システム開発における「超上流」と名付けた工程です。システム開発が決まってから行われる工程が上流なら、その前を決めるための工程なので、「超上流」というわけです。

もうおわかりですね。システムは、いろいろな所で、いろいろな人が使い、いろいろなことを言うようになっています。システム開発に使われる技術は、多様化し、高度化し、非機能要件もある程度以上に具体的な設計をしないと見えてこないことが多い中、必要だといわれても、稟議決裁の段階で見積もりまでするとすると、これはなかなか大変なことです。事業化のための、システム開発の見積もりをすると言っても、要件が決まっていなければ見積もれるわけがありません。見積もれば、それに縛られます。しかしシステム開発のコストなど見えなければ、決裁ができない。超上流における要求獲得と要件定義の要求は極めて難しい課題なのです。

この問題に正面から取り組んだ冊子があります。情報処理推進機構から出ている SEC BOOKS「経営者が参画する要求品質の確保 ～超上流から攻めるIT化の勘所～」(オーム社)を一読してください。

2. 2. 2. システム開発の目的

一般事業会社がシステムを開発するのは、システムが実現するビジネスであったり、画期的なサービスです。システム開発に 100 億円投じる決断をした社長さんが欲しいのは 100 億円分のプログラムやコンピュータではなく、数千億円の利益を上げる(はずの)システムが実現する仕組みです。これは、社長さんでなくともシステムを発注する側にいる担当者も同じことです。

一方、システム開発を担う方が見ているものは(意識しているものは)、多くは実現するシステムそのものです。このずれが要件定義を難しくしています。システム開発を依頼する側にいる者は、多くの場合システム開発の専門家ではないので、システム開発を担う人が必要とする情報をうまく伝えられない場合が多いのです。中には最初からシステム開発はよくわからないから任せた、という人もいます。いわゆる丸投げです。丸投げでも、出来上がったものをそのまま受け入れ

るのならよいのですが、なかなかそうはなりません。人は出来上がったものを見ないとイメージが掴めないで、あとから「これでは使い物にならない」と言い出すわけです。使えるものを頼んだはずだ、しかし、使えるものとはなにか、実は相手に伝えることを怠っていることを忘れていて、ということも多いのではないのでしょうか。システム開発はオーダーメイドですから、いや、実際のところ、これでもか、これでもかというほど要求事項を正確にシステム開発担当者に伝えないと、求めるものは実現しないといってよいでしょう。

簡単にいえば、できたことになるという条件を、事細かに伝えなければ、システム開発はうまくいきません。後述しますが、システム開発には要件として定義しなければならない対象が山のようにあります。その対象について全て、依頼側がこうでなければできたことにならないという視点で、要求を要件のレベルまで明確化する必要があります。それはシステムが出来上がったときに実現する世界をシステム開発ができる要件のレベルの範囲、深さ、粒度で記述することなのです。業務要件定義といってもよいでしょう。それをシステムでどう実現するかを決めるのがシステム開発担当者による要求(業務要件)分析であり、さらにそれはシステム要件定義につながります。当然のことながら、そこから、できること、できないことも見えてきます。

(超上流ではこのできること、できないことのリスクを評価し、稟議に織り込むことが必須となります)

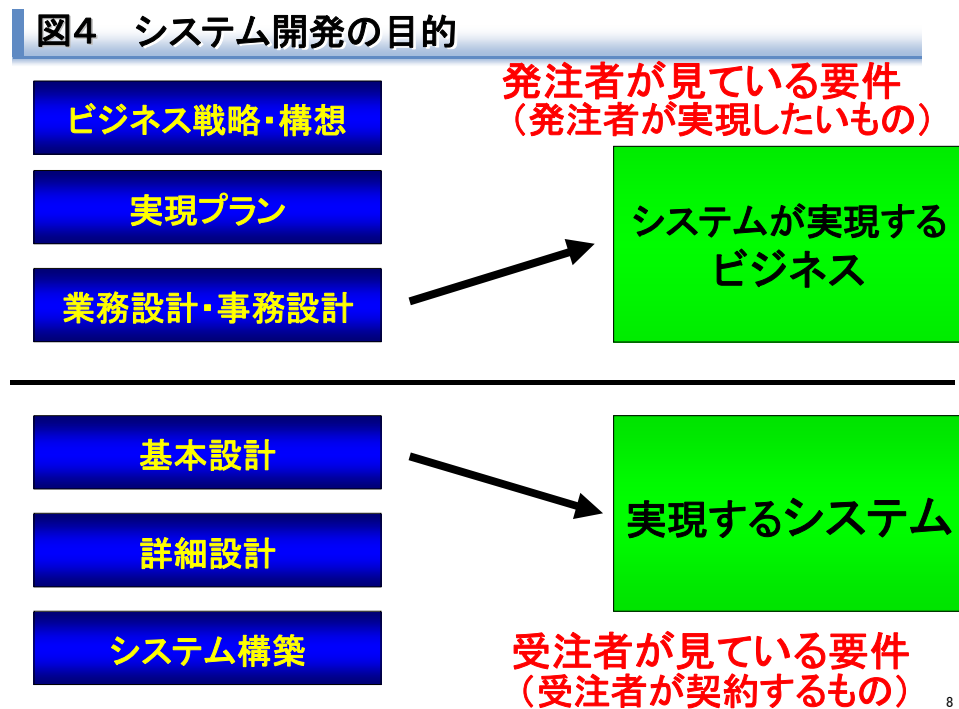


図 2. 4 システム開発の目的

発注側がシステム開発に関心がなく、システムが実現するものにだけ目がいくのはわからないわけではありません。しかし、システムが実現するものに責任を持つなら、システム開発の最初の重要な作業である、要求の明確化、具体化としての業務要件定義は役割として行わなければな

りません。システム開発によって実現するシステムが、自分の希望するものを間違いなく目論見どおりに実現するよう、自分のために、要求の具体化と、システム開発担当者への説明責任を果たさなければならないのです。

2. 2. 3. ベースラインが重要

システム開発では、先の工程のアウトプットを、次の工程のインプットにして作業を進めます。つまり、要求を聞いて、要件定義をします。要件定義を受けて、仕様書(設計書)を作ります。仕様書を読んでプログラムを作ります。設計書を見てネットワーク設計をします。前工程からの情報がいい加減なものであったら、次の工程はうまくいくわけがありません。

さらに言うなら、要件定義をきちんと行わないと、変更管理もできません。システム開発を始めるにあたって、要件定義ができない部分もあるでしょうし、曖昧なものもあるでしょう。そういったことは、できていないこと、曖昧なことを表に出して、要件定義の確定管理をしないとうまくいきません。未決や、曖昧なことも含めて変更管理ができるようなベースラインを作り、そのベースラインを軸に変更管理が必要になります。

図 2. 5 のとおり、ベースラインができて初めて変更管理が可能になります。ベースラインは発注者とシステム開発担当者の役割に応じた責任ある要件定義作業によって変更管理に耐えうるものとなります。変更管理の甘さは、今日のシステム開発においては、プロジェクト破綻リスクを増大させます。

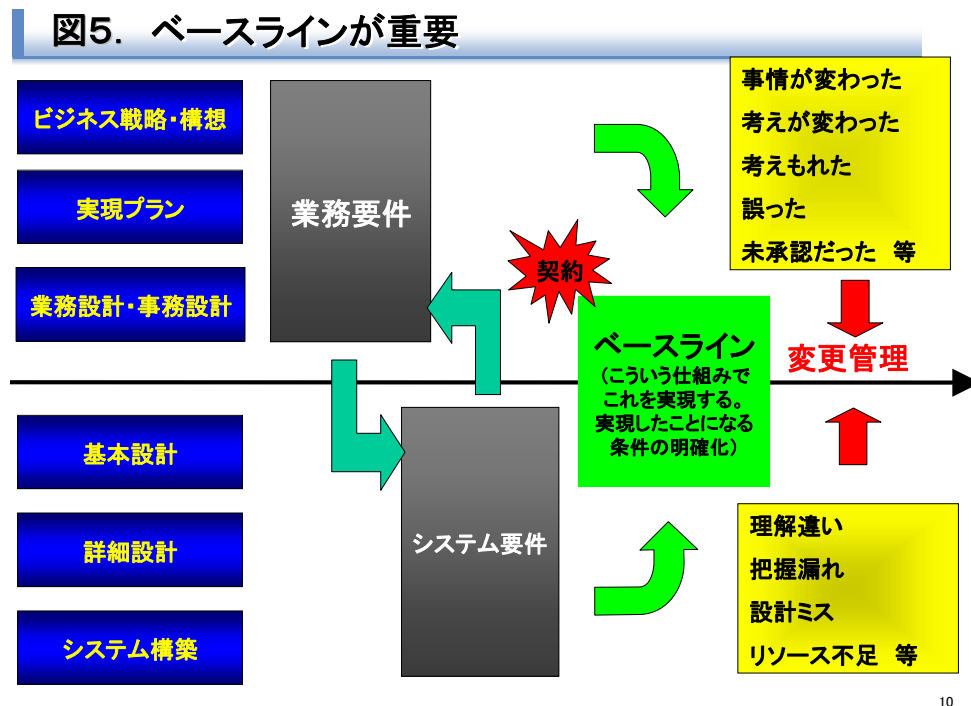


図 2. 5 ベースラインが重要

2. 2. 4. 要求から要件へ、要件定義のノウハウ

以上、今日、ビジネス要求を間違いなくシステムによって実現するためには、様々な困難な課題があること、おわかりいただけたと思います。要求を要件に落としていく技術がどんなに優れたものであっても、要求の発生源であるビジネス上の企てが、またシステムを要求する組織が抱える掟のような暗黙知的部分が見えなければ、「できたことになる」というレベルでの要件定義はできないのです。

それなら、超上流から何を見たらよいか、押さえたらよいか。

次に、5つの分野について要件定義を行うために押さえることが必要な事項を説明します。

(1) ビジネス分野

システム開発はビジネス構想、ビジネス計画に対応して行われます。しかし、このビジネス構想自体が不確実なものであることはいうまでもありません。ビジネスの企てが、企てどおりに進むなら、誰でも社長ができるでしょう。実際はなかなか思いどおりにいかないものです。

したがって、システム開発においてはビジネス構想そのものについて次のような点をしっかり押さえて進めなくてはなりません。これはリスク管理の観点から大変重要です。

<押さえるべきポイント例>

1. ビジネス構想の不完全さ対応を織り込む(把握不足、判断誤り、目論見違い、事情変更)
2. 実現したい世界の構想は往々にして、主観的、非定量的、一方的、個別・部分最適であり、結果として、みなし、決め付け、思い込み、勘違い、漏れ、誤りを含むことを評価する
3. システム化要求の内容は未承認であることが多い、承認レベルを押さえて進める

(2) システム開発対象

システム開発においては、理想の完成形を、頭の中で常に追い続けなければなりません。しかし、実際のところは人、物、金、時間といったリソースの制約を大きく受け、その時々、システム開発する範囲と内容が決まってきます。正しい答えはリソースによって異なってきます。こうしてシステム開発の範囲が決まったときには、次の点を押さえるべきでしょう。

<押さえるべきポイント例>

1. システム化要求の承認レベルを押さえる
2. システム化要求にかかる
3. 全てのステークホルダーの意見反映レベルを押さえる
4. 発注者は利用者でなくなっていることを考慮する(マーケットの声が聞こえていない)
5. 特に、担当者が勝手にこうなると思い込んでいると判断される要求は吟味・管理する
6. 最悪は同じという要求(同じにする要件定義は大変だ、同じでよいなら作る必要はない)

(3) 非機能要件

システム開発にさまざまな技術が使われるようになったことは既に指摘しましたが、ああいものが欲しい、こういうことを実現したいという要求を要件に落としていくと、次々に見えてく

るのがこの非機能要件です。いわば、最初に要件として出なかったものを含む、主に技術要件を主とした要件と言ってよいでしょう。

この、非機能要件の厄介なところは、他の要件が固まってくると、そういうことなら、と生まれてくる、あるいは決まってくるという性格を持っているところです。加えて、お金がかかることになる要件も多いのです。例えば、システムが止まったとき、すぐに立ち上げるにはそれなりのバックアップの仕組みを作らなければなりません。サーバーを二重化すれば、かなりの投資になります。困ったことに、「すぐに立ち上がる」といった、自然言語による表現は、なんとも曖昧で、聞く人によって、「すぐに」が1分のときもあり、1時間のときもあり、翌日のときもあり、といった具合なのです。そのレベルによってお金のかかり方も違います。こういった要件が、システムテストの段階で問題になったりすると、これはお金が絡むこともあり、かなり厳しい対立を生む原因にもなります。

非機能要件は、業務要件などを決めなくては決まてこないという面があるにせよ、早く押さえるにこしたことはありません。

<押さえるべきポイント例>

1. かつてに、そうなると思い込んでいるところの吟味(機構、承認、体制、事務の仕組、使い勝手、背景不明作業など、要は書いていないこと)
2. レスポンス、キャパシティ、セキュリティ、トレーサビリティ、モニタリング、メンテナンスビリティ
3. 有事対応、障害時対応、コンテンジェンシープラン
4. 環境(何台までしかつながらない、どこでしか出来ない)、人(誰しか出来ない、同時に出来ない等)、物、金(後年度負担)、24 時間 365 日等
5. 特許対応、法律・認可対応

(4) 保有システム分野

20 世紀後半、どの企業もシステム開発を日夜行い、業務の革新を行いました。その結果、今や大企業ほどシステムに依存した経営、業務運用が行われていると言ってよいでしょう。

大手の損害保険会社などでは、毎年数万本のプログラムを新設、修正します。その多くが、20 世紀作り上げられた、いわばレガシーともいべきシステムの保守といったら、驚かれるかもしれません。しかし、それが現実です。画面に一つ項目を追加しただけで、数百本もプログラムを修正する、などといったこともあるのです。

2000 年問題で保有システムを洗いざらいチェックしたことは記憶に新しいところです。大変な作業でした。大山鳴動して鼠一匹、も出ないということでしたが、とにかく保有システムの長大さは万里の長城のごとくです。

新規システムも、保有システムとつながなければその機能を発揮できない場合が多いわけですから、この保有システムにかかる手当て要件の洗い出しは大変重要な作業といってよいでしょう。

注意すべきは、対象となるのは保有システムだけではないということです。システムを使っている人、システムをささえるインフラなども吟味の対象にしなければいけません。

<押さえるべきポイント例>

1. 現行システムとその利用、運用(前提、内容、事情、制約、研修、運用、特殊運用、担当、契約、過去のイベント対応等)
2. アウトプット活用内容(追加加工、データ二次活用)
3. アウトプットステークホルダー対応(代理店、客先(個人、企業)、社員、利用管理者、監督官庁、コールセンター)
4. 関連システム(含む社外、インターフェイス、内容、事情、制約、運用等)
5. インフラ(キャパシティ、OS バージョン、ネットワーク、認証契約、アウトソース関連等)

(5) プロジェクト

システム開発プロジェクトを進めている際にも、実に様々な要件定義を必要とするシステム手当てが発生します。これはリスク管理の観点から検討してもよいのですが、可能な限り、システム開発を始める前に吟味検討しておく必要があります。

例えば、共用プログラムに他プロジェクトと同時に修正を入れなくてはならないときなど、通常とは異なった修正反映管理が必要になります。時にはプログラムを別に新設するという判断になることもあります。

また、システム開発中に言語や OS などのバージョンアップがあったり、ハードウェアの入れ替えなどがあると、テストなどのロードが増すだけでなく、プログラム開発について対応することが必要になることもあります。

こういった、システム開発期間中の影響イベントはなるべく広く、正確に捉え検討すべきでしょう。

<押さえるべきポイント例>

1. 発注者、マーケット、客先要望事情等、ステークホルダーの個別事情
2. 同時期イベント、立法、通達、認可、契約完了、契約条件変更等
3. 社内インフラ改定、関連システム改定、制度改定、機構改革等
4. OS バージョンアップ、保守切れ等必須対応要素
5. 社外利用インフラ改定(郵便番号等)、料金改定等外部環境条件
6. 開発力確保、パートナー・アウトソーサー・客先の開発力
7. 開発中・展開時有事対応(地震等天災、展開季節、利用パッケージ会社買収)、
8. 同時開発・プログラム共有システムの有無

2. 3. 最後に

一般的にシステム開発は、こういうものを作りたいという要求の説明から始まります。しかし、業務要件を納得感のあるものにするためには、こういうことを実現したいという企画の説明が必要です。業務要件については「システム化の目的」、「システム化の方針」、「システム化の理由」、「システム化の期待効果」といった情報が不可欠となります。これは一般事業会社では、部門長への伺いに書かれる内容と同じです。システム開発規模によっては経営会議の付議資料に記載される内容です。

次に、システムを導入する組織の文化、どこにも書いていない掟的ルールも把握する必要があります。

ます。例えば、毎年構造変更を伴う機構改革(県別体制から、東北といった地域単位への変更、本部制の導入など)が頻繁にある会社の場合なら、毎年の機構改革に耐えられるように、変更しやすいシステムを作らなければなりません。

一年契約の満期日は、翌年の同じ日でしょうか、その前の日でしょうか。このように業界によって違う内容、意味を持つ言葉もあります。

さらに、規定だけ読んでも、現場で臨機応変に対応している事務処理は見えてきません。営業所の出納帳簿といった会計帳簿の付け方も、システム化される前はずいぶん場所によって違ったものです。「どうしたいのか」がわからないと要件は決まりません。

このように、システム開発の要件と言うものは、なかなかシステム開発の受託側では把握できませんし、どのように扱えばよいのか適正な物差しを持てないものが多いのです。

一方、システム開発にかかる要件には、システム技術力と知識がなくては出来ないものがたくさんあります。これはなかなか一般事業会社の社員にはわからない。

もうおわかりだと思います。つまり、今日における、システム開発のための要件定義は、発注者と受注者が共に、業務のプロ、システムのプロとして知恵を出し合って決めていかないとよいものできないのです。

先にあげた「押さえるべきポイント例」についても、その扱いについて決めるには、業務の専門家とシステムの専門家の密接な連携が不可欠といえましょう。

要件定義は、発注者、受注者が協力して行う。これが成功の鍵なのです。

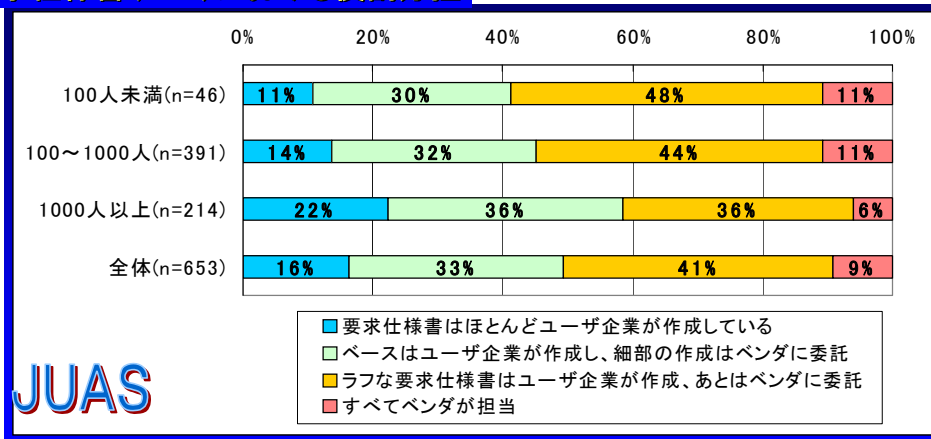
最後に、大変興味深い調査をご紹介します。

JUAS(日本情報処理ユーザー協会)が2005年に発表した要件定義関係の調査です。

図2.6はRFPの役割分担です。要求仕様書をユーザー企業(依頼側)で作成しているのは16%にすぎません。図2.7はシステム開発が終わった後の反省点です。約80%の会社が、システム仕様の定義が不十分のまま発注してしまったと回答しています。これからどのように要件定義を進めて行けばよいのか答えがあるように思います。

図6. 発注者は自分の欲しいものがわからない？

要求仕様書(RFP)における役割分担



・ラフな要求仕様だけで、残りはベンダに書いてもらっている企業や、ベンダにすべて書いてもらっている企業が、全体の半数

JUAS IT動向調査2004

図 2. 6 発注者は自分の欲しいものがわからない？

図7. 発注者は反省している

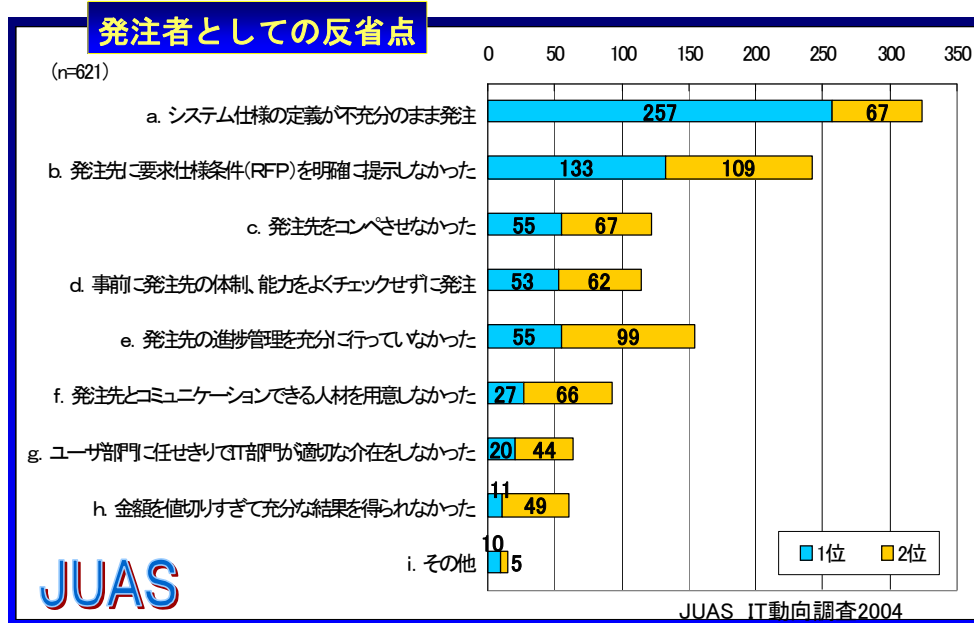


図 2. 7 発注者は反省している

3. 社内標準に基づく要件定義事例

(株)構造計画研究所 後藤 卓史

概要:

構造計画研究所は 90 年代初頭からソフトウェア定量化と知識の形式知化の重要性に気づき、FP 法をベースとしたソフトウェア定量化と、熟練者の知識を集大成したグリーンブックをまとめた。その中で要件定義については「要求仕様書作成要領」としてまとめられている。

昨今、Time to Market を重視するソフトウェア開発が求められている。その結果、短納期と仕様未確定状態でのプロジェクト開始が当たり前になってきている。それに対して構造計画は開発規模コントロールと提案ベースでの要件定義で対応しようとしている。だが、根本的な解決には発注者と開発者のコラボレーションが必要である。

3. 1. 構造計画研究所におけるソフトウェア開発の現状

構造計画研究所の顧客業種は運輸情報通信業、製造業、電気・ガス業、建設業、サービス業、官公庁等であり、建築関係 CAD や構造計算関連ソフトウェアに強みを持っていた。

そもそも構造計画の前身は一設計事務所であった。建築設計には構造物の耐震強度計算が必須であり、それを自前で計算しようということになって計算機が導入された。それが計算プログラム開発も手がけるようになり、次第にさまざまなソフトウェア開発に広がっていった。よって強みはその分野であった。しかしながら、近年は製造業関連の受注の伸びが著しく、そのソフトウェアの内容も多岐にわたって来ている。それらのソフトウェア開発の多くは改造プロジェクトであり、新規開発の占める割合は少ない。したがって特定の発注者からの受注が多いといえる。また、従来から通信関連の研究開発支援ソフトウェア開発も手がけていたが、最近は運用系ソフトウェア開発も増えている。

そのような背景のもと、要件仕様についてはシステム全体をすでに理解している事を前提として、改造部分だけの要件仕様を渡されることが多く、かつ短納期になってきた。そのため、当該プロジェクトにはその開発経験を十分に保有する技術者が必須となる。なぜならば、そのプロジェクトの経験がなければ全体を把握することも困難であり、スタートダッシュも出来ない。結果として、短納期・高品質はまず達成できないからである。

その現実ソフトウェア技術者の特定プロジェクトへの固定化につながり、さまざまなキャリアを積んで熟練した SE を目指そうとしている技術者にとっては厳しいものとなっている。

また、短納期ということもあり、経験のあるソフトウェア技術者に業務が集中し、長時間労働を強いられることになった。構造計画のみならず、現在の情報処理産業界にとってこの状況は、いずれにおいても同様であると思われ、ソフトウェア企業における人材のキャリアパス形成に大きな困難を生み出している。また、それは長い目で見ると情報処理産業全体への優秀な人材の流入を阻害しているとも言えよう。そのような状況を鑑みると、現在は要件定義だけではなく、ソフトウェア開発のあり方そのものに対して、見直しの時期が来ていると言える。

3. 2. グリーンブック

3. 2. 1. グリーンブック構成

構造計画は 90 年代の初めからソフトウェアに関する定量化の重要性に気づいてプロジェクトデータ収集と FP 法の導入を図ってきた。また知識の形式知化[1]の緊急性を認識し、社内の熟練者が保有する知識の集約を開始した。それらは図 3.1 に示すようなソフトウェアライフサイクルに沿った形でまとめられ、表紙の色が緑色で統一されたところからグリーンブックとして社内で活用されてきた。構造計画がソフトウェア開発において曲がりなりにもプロセス化が進められたのは、このグリーンブック制定に負うところが大きかったといえよう。

ソフトウェア開発のプロセス化は、2000 年 4 月の ISO9001 認証取得でグリーンブックの優れた部分を取り込むという形で概略完成し、年々改善サイクルをまわしながら進展している。ISO9001 の目指すところは奇しくもグリーンブックの目指すところとほぼ一致しており、ソフトウェア開発部門への ISO9001 導入に対しては大きな困難は感じなかった。

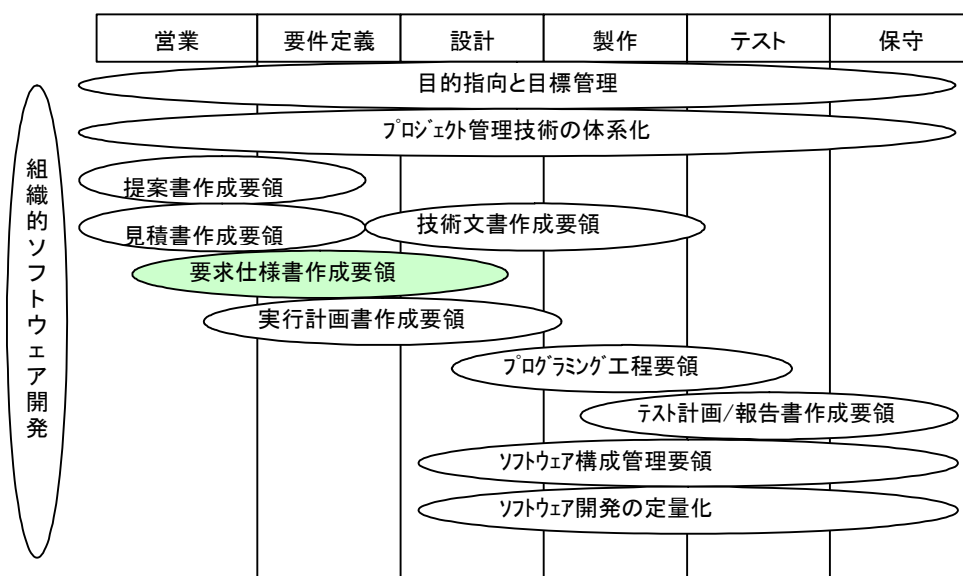


図 3.1 グリーンブック体系

さて、グリーンブックにおける要件定義は、図 3.1 に示す「要求仕様書作成要領」に詳細に記述されている。構造計画の要件定義はこのプロセスにしたがっているので、以下詳細を述べる。

注記)グリーンブックでは「要求仕様、要求定義」という表現を用いている。近年はそれを「要件仕様、要件定義」と呼称するようになってきた。英文では requirement であるが、日本語では要求と要件の違いを明確にしないまま要件という表現の方が一般化されてきている。しかし筆者としては、要求はあくまでも発注者がシステムに抱く期待であり、それを具現化できるように分析し優先順位化して具体的なシステム化のために記述するものが要件であろうと感じている。ちなみに ISO9001 邦訳[2]では requirement は要求と訳されている。

3. 2. 2. グリーンブックにおける要件定義プロセス

グリーンブックでの要件定義プロセスを図 3.2 に示す。プロセスの概略フローは要件を理解するための導出、要件の形式的な記述、記述した要件に対する確認の3つの部分から成っている。フローには現れていないが実際には繰り返しを含んでいる。発注者(図 3.2 中では顧客)の構想を引き出し、かつ実現可能な要件に切り出すには、発注者との繰り返しのコミュニケーションがもっとも大事である。これらのプロセスの遵守とスピード化がプロジェクト成功への近道である。

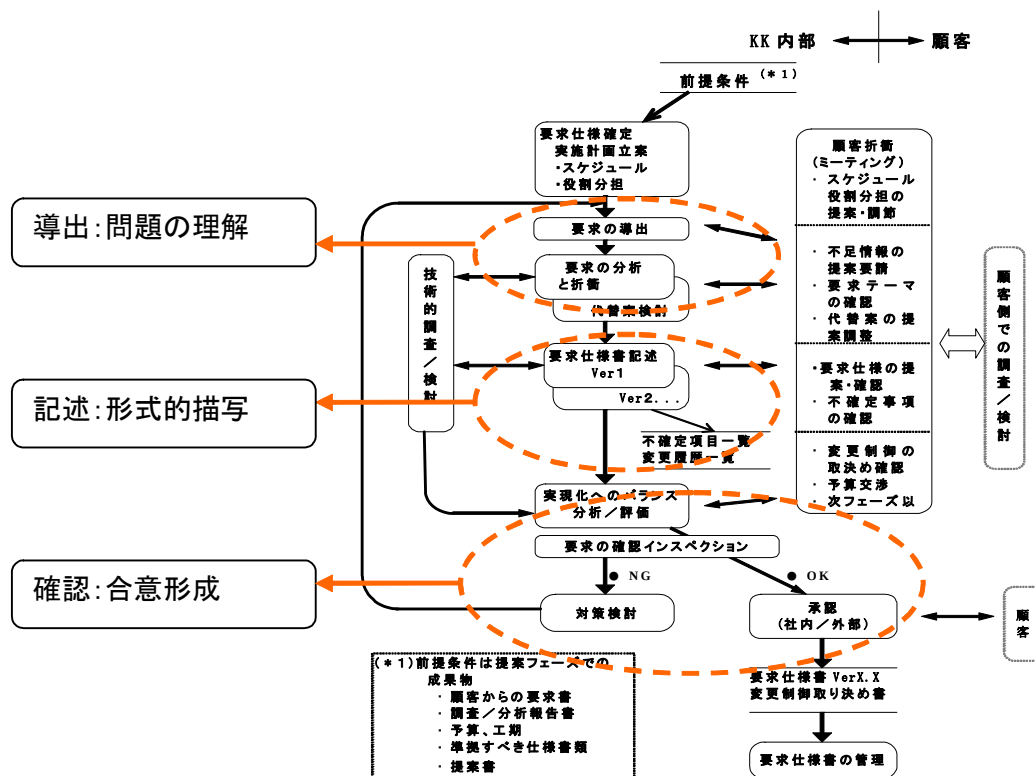


図 3.2 グリーンブックにおける要件定義プロセスフロー

(1) 導出：問題の理解

問題領域の要求モデルを作成するために必要な関連知識をすべて取得(導出)するプロセスであり、基本的に以下を実行する。

- ・要求に対する知識の情報源の特定
- ・該当知識の獲得
- ・該当知識と対象とする問題の関連性の特定
- ・関連する知識の意義と、それが要求に与える影響の理解

また、プロセスは、1)要求の分析と折衝、2)矛盾・重複・欠落解決のための折衝、の2つに分かれている。

1) 要求の分析と折衝

- ・要件範囲(システム化境界)の定義
- ・チェックリストを用いた要求分析

2) 矛盾・重複・欠落解決のための折衝

- ・矛盾、重複、欠落の特定
- ・解決のためのミーティング
- ・代替案検討と折衝

(2) 記述:形式的描写

導出プロセスから導かれた要件を形式化して文書化するプロセスであり、ここでは誤解や恣意的な導入を防ぐために徹底的な形式化と用語の統一を行う。このプロセスは以下の3つに分かれている。

1) 図表表現、標準目次、標準テンプレートの利用

要件定義はわかりやすく、誤解がうまれない正確な表現を行う必要がある。そのため図表表現を用いて理解しやすくすることが必要である。また、標準目次を設定して要件定義の一貫性維持と欠落が無いようにすることも重要である。また記述に際しては標準文書テンプレートの適用も必要である。

2) 業界専門用語の取り扱い

通常、顧客は業界専門用語を用いて業務を行っている。したがって、ソフトウェア開発者が業界特有の専門用語を理解できるように用語の定義を行うことが必要である。

3) 簡潔かつ一貫した言語の使用

誤解のない文章を作成するには、簡潔で一貫した正しい日本語表現の適用が必要である。概して行間を汲み取るということがよくあるが、それは誤解のリスクを埋め込むことになり、必ず避けなければならない。読む人によって解釈が異なる文章の作成が求められる。

(3) 確認:合意形成

このプロセスでは、「記述」により表現された要件定義が、発注者の意図に反していないことを確認し保証する。このプロセスも以下の3つに分かれている。

1) 要求仕様インスペクション

作成した要求仕様書に矛盾・誤り・欠落がないことをインスペクションで確認する。

＊現在は ISO9001 の規定の中で要求仕様インスペクションのチェックリストを規定しており、必要に応じて実施するとしている。

2) 要求仕様インスペクション結果に伴うアクション

インスペクション結果は要求仕様書への反映だけではなく、発注者との折衝やプロジェクト関係者との調整も含まれる。場合によっては逆提案に及ぶこともあり、重要なプロセスである。

3) 承認

作成した要求仕様書は社内はもちろん、発注者の承認も確実に取っておく必要がある。このプロセスはプロジェクトの成功にとって欠くべからざるものである。

3. 2. 3. グリーンブックにおける要求仕様書詳細

グリーンブックの要件定義は米国の IEEE 規格である IEEE/ANSI830-1993 (Recommended Practice for Software Requirements Specifications:ソフトウェアの要求仕様に関する指針)を参考に記述されている。その IEEE/ANSI830-1993 の概略構成を表 3.1 に示す。

表 3.1 参考)IEEE/ANSI 830-1993 の構成

1. はじめに
1.1 要求定義文書の目的
1.2 成果物の範囲
1.3 定義、頭字語、および略語
1.4 参照資料
1.5 その他の文書の概要
2. 一般記述
2.1 成果物の概要
2.2 成果物の機能
2.3 ユーザの特性
2.4 一般的制約条件
2.5 仮定および依存要因
3. 機能的要求、非機能的要求、インタフェースなどの要求
外部インタフェース、機能的特性、性能要求、 論理データベース要求、設計制約条件、 システム特性、品質特性など
付属資料
索引

表 3.1 に示すように、この規格は 1993 年当時において、現在問題になっているような項目についてどのように考えるべきかをすでに規定しており、非常に参考となる先進的な規格である。

ただし、顧客として軍を意識しており、厳格すぎる面も見受けられる。したがって、グリーンブックとしてはこれをそのまま取り入れるのではなく、参考にしつつも現実に合った手法に変更した。

3. 2. 3. 1. 要求仕様書の全体構成

グリーンブックの要求仕様書では「導出」と「記述」内容については以下Ⅰ～Ⅶの目次構成が必要であると述べている。その中から、プロジェクトの実態に応じて必要な項目を抜き出して要件定義を記述するようになっている。

- Ⅰ システム化の背景と構想
- Ⅱ 機能要求仕様
- Ⅲ データ要求仕様
- Ⅳ 性能要求仕様
- Ⅴ 信頼性要求仕様
- Ⅵ セキュリティ要求仕様
- Ⅶ 運用・保守要求仕様

また、「確認」についてはインスペクション、アクション、承認について述べてある。

(1) 導出

これは「Ⅰ システム化の背景と構想」の部分である。システムを開発する際に、システムが動くことだけを考えて開発しても実際には当初目的である業務効率化推進を達成できないことが多い。発注する顧客にはシステム化の前にビジネス計画や方針が必ず存在する。そのビジネスを成功させるには何をなすべきかを念頭におきながら、そのためには、何のためにどのようなもの

をいつまでにシステム化するということを明確にしなければならない。

したがって、システム化を決意した背景とそのテーマ・目的を明確にしておく必要がある。また、顧客内でのシステムの位置づけや範囲、運用形態についての記述も関連情報として必須である。P. LoucopolosとV. Karakostasはその著書 [3]の中で要求の導出について「他人の問題を解決しなければならないときにまず第一に行う必要があるのは、その問題についてさらに多くのことを見つけ出すことである」と述べている。単なる開発ではなく、顧客の要求の背景にまで深く掘り下げた上で要件定義を行うことは今後のソフトウェア開発企業にとって必要なことであろう。

(2) 記述

これは「Ⅱ 機能要求仕様」～「Ⅶ運用・保守要求仕様」について、その目次と記述方針について述べてある。この記述についてはプロジェクトは千差万別なので、内容を詳細に書くことはしないで大まかな方針のみにとどめてある。さらに、記述した内容の網羅性を高めるためのチェックリストが添付されている。

表 3.2 要求分析チェックリスト概略例

- ・基本設計に実現に必要な情報を含んでいるか
- ・要求が分割できるものであれば分割したか
- ・要求定義に修辭が含まれていないか
- ・ビジネス目標との合致は
- ・要求の意味は他社にとって一意に理解できるか
- ・システムが現実的に開発可能か
- ・テスト計画を作り出せる要求仕様か

表 3.2 の内容を見ればごく当然のことばかりである。しかしながら、プロジェクトリーダーがこのようなことを毎回反復できるかを考えると、そこにチェックリストの存在意義がでてくる。

(3) 確認

確認については、インスペクションを実施し、結果についてアクションを起こす。最終的に承認プロセスでの承認を取り付けるとある。それぞれについて留意事項が述べられている。

以上がグリーンブックで述べられている要求仕様書を作成するための内容である。しかしながら、「要求仕様書作成要領」は第1版が 1995 年、第2版が 1999 年に発行されたものの、第2版発行の時期においてはソフトウェアを取り巻く環境は大きく変化してしまった。次章では 90 年代後半ぐらいからソフトウェア開発企業が抱えるようになった問題点について述べる。

3. 3. 要件確定で現在抱える問題点

昨今は Time to Market を競う時代となり、ソフトウェア開発は多くの問題を抱え込んだ。特に仕様確定は品質や生産性(=利益)につながる重要なアクティビティである。その重大なアクティビティについて、走りながら考えるしかなくなってきたのである。このような時代になって開発者が直面する要件確定に関する問題を挙げてみよう。

(1) 仕様未確定でのプロジェクト開始

現在、発注者から受ける要求は納期が非常に短く、かつ要求内容は不定で順次確定するというものが多い。経験から言えば、90年代から比べると納期は確実に半減している。当時はプロジェクトが早く終了した場合には社外のセミナーを受けたり、休暇をとったりしてリフレッシュをしていた。また、先輩から後輩への技術の継承が行われていたのは余裕がなせるわざでもあった。

そのような時代はもはや過去の話であり、今やいかに短期に高品質のものを次々に出せるかを競う時代になっている。したがって、仕様は決まっていない、記述したものは何も無いというような構想のみでの発注ということもめずらしくない。それでも仕様が開発初期に確定すればまだ良いほうで、中には納期直前に最終仕様が決まる例もある。

(2) 仕様の漸増

プロジェクト初期に仕様確定していないということは開発が進むにつれて確定するということである。また、そのような決め方であれば、当初は構想になかった仕様が次々に発生し、ややもすると当初の見積り規模を大幅に上回るようになる。多くの場合、それら追加の仕様は開発担当者にとって十分に許容できる程度のちょっとした追加仕様であり、営業を間に挟んで決めるまでもないと判断し安易に受け入れてしまう。その積み重ねで、やがては気づかないうちに許容範囲を超えてしまうのである。開発担当者にとって発注者側担当者のちょっとした追加を受け入れることは、良いものを作りたいという技術者のプライドであり、また顧客満足度向上の一環でもある。したがって、ずるずると受け入れてしまうことになる。

さらに、プロジェクトマネジメントとして、いつまでの仕様変更は許容する(当然上限はある)が、それ以降の変更は次回へシフトするというコントロールも行っている。これは品質の悪いものを遅れて納入するというお互いに避けたい事態への対応であるが、いつも認められるわけではない。

(3) これまでの標準化の崩壊

仕様確定がプロジェクト後期にずれ込むに従って、先に述べてきたグリーンブックの要求仕様書作成要領では到底追いついていけない状況になっている。グリーンブックや要求定義工学関連書籍に見られる記述は、納期要求がこれほどまでに厳しくなかった時代の標準化であったといえるだろう。最終的にプロジェクトとして決めるべき項目は同じなので、要求仕様書は結果的に出来上がるとしても、プロジェクトの初期に要件をモデル化し分析をした後に確定仕様を記述することは不可能に近くなってきている。したがって、グリーンブックも含めてこれまでの要件定義における標準化やプロセスは見直す時期に来ていると言える。

(4) 開発工数の増大

短納期かつ要件は漸次確定するという状況下では手戻りや重複が多発し、開発工数は増大す

る。なぜならば、例えば「仕様が決まっていないので、できるところから手をつけよう。仕様が決まってから手直しするほうが、決まるまで待つよりは早い」となるからである。また、仕様確定が遅れると、納期に圧迫されてどこかで必要な時間を省略せざるを得なくなる。それは本来時間のかかる詳細な設計工程、もしくはテスト工程が対象になることが多いので、当然ながら品質は低下し、バグ対応に追われて工数は増大するという負の循環に陥る。

したがって、これまでの要件定義では標準的な文書化が特に重要視されており、それが基本とされているが、現在ではドキュメント品質を向上させて欠陥を防止するなどは困難な話である。

(5) 非機能要件の増大

非機能要件には発注者の業務分野に特化したもの、応答速度などのようにシステム内容に関わるもの、法令に基づくものなどがある。特に近年はシステム管理基準や情報セキュリティ管理基準が出されており、開発者としてもセキュリティ問題を無視できなくなってきた。セキュリティ問題は発注者の IT 統制にかかわることなので、ソフトウェアの機能仕様に非機能要件として考慮しなければならないだろう。

* ISO90017.2.1 b)にも「明示してはいないが既知である事項も明確にする」とある。

3. 4. 問題への対応

3. 3. で述べたような問題に対応するために、開発者側でやれることは限られている。しかし、構造計画では以下のような対応を図っている。

(1) 開発規模のコントロール

組織におけるソフトウェア開発能力を前提にした規模見積りとコントロールはマネジメントの基本である。したがって、要件が決まっていなければ開発可能なボリュームでコントロールするのが現実的であろう。受注の際、多くの場合において発注額と納期が最初に確定する。そこで、組織の生産性(h/FP)から開発可能な FP 規模をおおまかに決めることができる。

$$\text{開発可能規模} = \text{対象システム機能量 (FP 値)} \times \text{生産性 (h/FP)} = \text{工数}$$

このようにして要件の機能量(FP 値)が決まれば開発に必要な工数がおのずと決まり、納期に間に合わせるための人員も決まってくる。その開発可能規模をにらみながら漸次確定する要件定義に対応しようというのである。多くのプロジェクトリーダーは継続プロジェクトであれば経験的に発注額と納期から開発可能規模を把握できる。とはいっても、数値的な裏づけは繰り返し精度を高める上で非常に重要である。

しかし、初期段階の要件には未定部分が多く、その段階で精密な見積りを行うことは大変困難なことである。幸いにも継続プロジェクトの場合データファイルは自明なことが多い。したがって、データファイル数から概算する NESMA の FP 試算法[4]が適用しやすい。この手法は下記式を用いるが、それには適用条件が付いており算出精度もそこそこなのであくまでも参考値であることを忘れてはいけない。

【参考:NESMA の FP 試算法[3]】

FP 値 = 内部論理ファイル(ILF)数 × 35 + 外部インタフェースファイル(EIF) × 15

条件：・ ILF には平均して3つの EI と2つの EO、1つの EQ を伴う。

・ EIF には平均して1つの EO と1つの EQ を伴う。

(2) 提案ベースでの要件定義

概要だけが決まっている要件の場合、それはソフトウェア開発にとって問題でもあるが、逆に開発者が得意な領域へ要件定義を誘導するチャンスでもある。発注者への提案に開発者の蓄積技術や再利用可能性、経験者の有無を盛り込む。その条件で要件を定義できるならばプロジェクトの効率向上や利益を確保することができる。開発者の提案ベースで要件が確定できれば、それは開発者が品質を保証し、かつ納期も確実に守るという前提なので、発注者と開発者は Win-Win の関係を結ぶことができる。

近年の成功プロジェクトを見るとこのような提案が成功した多いように感じる。発注者と開発者の間に横たわる要件定義の一番の問題はコミュニケーションの問題である。その問題を効果的に解決する手段として、開発者が発注者の理解できる言葉で要件を記述するというのは良い方法であると思われる。

3. 5. 期待される要件定義

要件定義にはさまざまな手法やモデル化[3][5]が提案されている。しかしながら、ソフトウェア開発はまずビジネス達成が目標であり、そのソフトウェアは人が作って世の中に出すものである。それを前提として考えた場合、以下のことが併せて期待される。

(1) ビジネスプランと投資効果をもとにした要件定義

社会の複雑化や競争激化で、企業にとっては1日でも早いビジネスモデルの構築と運用開始が期待される。そのためにハードウェアとソフトウェアの調達が行われる。ハードウェアに関してはその価格、機能や能力、精度が明らかであり調達に際しては適切な選択が行われる。しかしながら、ソフトウェアに関しては、ビジネスモデルの重要な機能を具現化し、ユーザとの直接的なコミュニケーションをとる手段であるにも関わらず、期待する投資効果を得るための最適機能の検討やその規模の予測が十分にはなされていないように思える。詳細な検討に基づいた発注であれば、場当たりの仕様の漸増などは起こらないであろう。

ビジネスが最高の効果をもたらすための最適機能を決め、そのボリュームを予測して投資を行うということは投資の基本である。それにはシステムに対する事前の要件定義が必要であり、その中で優先順位をつけて投資を行う。つまり、ソフトウェア開発はビジネスプラン達成のための数ある投資アクティビティの中の一つであり、その要件定義なくしてはビジネス達成はありえない。

(2) 発注者と開発者のコラボレーション

システムに対する「要求」はさまざまであるが、システムの「要件」を決めるにはどのような最新技術があるか、技術上の制約は何かなどを詳細に知り、ビジネス適用を検討する必要がある。そのためには要件定義の段階から開発者を交えたビジネスモデル達成の最適要件を確定することが肝要であろう。その上で、①発注者は投資規模に見合った機能を吟味して、発注範囲(機能範囲)を

明確に決めること、②開発者はその最適要件を実現するために定量的なソフトウェア信頼性の実現と把握および保証を行なう、の2点が重要である。

また、要件を確定しても、なかには希望する運用開始時期に間に合わない要件も必ず含まれる。また開発中に追加変更せざるを得ない場合も発生する。社会は変化し開発者の能力にも限界があるので、これらは避けられないことである。したがって、一旦要件を確定したら、次には要件仕様のコントロールが必要になってくる。

「仕様をコントロールするという、発注者と開発者のお互いの努力があってこそ競争に打ち勝つソフトウェア開発が可能になる」[6]が示すように今後のソフトウェア開発は、単に何かのソフトウェアを言われたままに作るというのではなく、発注者のビジネスプランのもと開発者も交えて成功裏に持っていくという発想が必要ではなかろうか。

(3) 教育の必要性

要件定義手法にかかわらず、教育の「場」でソフトウェアに関する講座があまりにも少ない。日本の工業製品のほとんどにはソフトウェアが関与しており、ソフトウェアの出来不出来が製品の性能を左右するといっても過言ではない。ソフトウェアは今後の日本を支える大事な要素である。したがって、ソフトウェアライフサイクル全般にわたった教育を充実させる必要がある。

3. 6. 最後に

ソフトウェア開発の第一歩は要件定義である。品質や生産性はそこで決まるといっても良い。その要件定義に存在する問題を明らかにし、それに始まるソフトウェア開発現場の現状を改善することは、日本の情報処理産業を3K産業にしないために急いで対応すべきことである。今後の要件定義工学の発展に期待したい。

参考文献

- [1]野中郁次郎＋竹内弘高共著「知識創造企業」東洋新報社 1996 年
- [2]対訳 ISO9001 品質マネジメントの国際規格 2001 年1月 日本規格協会
- [3]要求定義工学入門 Pericles Loucopoulos, Vassilios Karakostas 著 富野壽監訳 共立出版 1997 年
- [4]NESMA(Netherlands Software Metrics Users Association)
<http://www.nesma.nl/sectie/home/>
- [5]要求定義工学プラクティスガイド Ian Sommerville, Pete Sawyer 著 富野壽監訳 共立出版 2000 年
- [6]ファンクションポイント法の導入と効果的活用 荒木貞雄／後藤卓史著 ソフト・リサーチ・センター 2006 年

4. 要件定義に関する取組状況

(株)菱友システムズ 袴田 典之

4. 1. はじめに

筆者は、情報系ビジネスアプリケーションのシステム開発、運用を担当してきたが、近年、業務プロセス改善活動を担当することとなり、システム開発工程の改善などに取り組んでいます。

本稿では、筆者の属する会社における要件定義工程への取組状況を紹介させていただきたいと思います。

4. 2. 当社で適用している開発方法論

当社が標準として適用しているシステム開発工程は、ウォーターフォール型の開発方法論を元としている。システム開発プロジェクトに適用する場合には、適宜、開発・管理工程のテラリングを施しつつ、インクリメンタルモデルやイテレーティブモデルを組み合わせることにしている。

ウォーターフォールモデル(図 4. 1)は、滝が流れ落ちるようにシステム開発を進めて行く開発方法論で、最も基本的なモデルと言える。システム開発プロジェクトを要件定義や設計、開発、テスト、導入移行の工程に分割し、各工程での成果物(定義書や仕様書など)を明確に定義し、その成果物で後工程を順次実施していく方法論である。

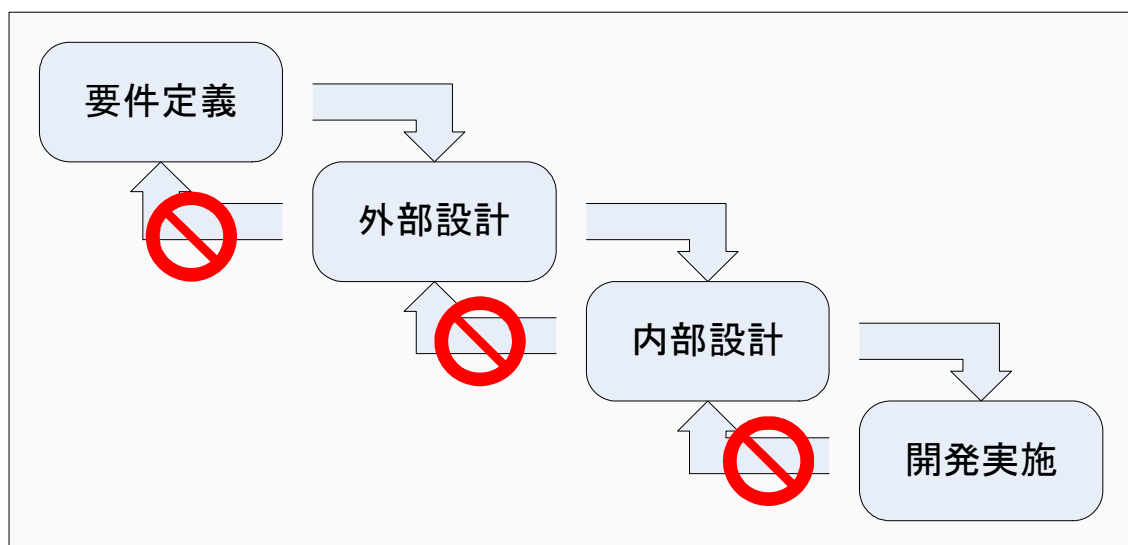


図 4. 1 ウォーターフォールモデル

また、前工程の定義書、設計書に従って工程を実施していくところからドキュメント駆動型モデルと呼ばれることもある。

ウォーターフォールモデルに対しての当社の評価は次の通りである。

- メリット：
- ・大規模開発にも適用できる。
 - ・プロジェクト管理が行いやすい。

- デメリット：
- ・基本的にフェーズの後戻りが出来ない。
 - ・実装面でのリスクを早期にヘッジすることが難しい。

4. 3. 当社で適用している要件定義手法

当社のシステム開発工程では、要件定義工程が「要求開発」と位置づけられる。要件定義においては DOA (Data Oriented Approach) 型を標準としており、特徴を列記すると次の通り。

- ・データ項目に重点を置いた要件定義と設計
- ・データ中心の手法／技法と表記法（DFD、ERD、データの正規化）
- ・開発局面間のスムーズな接続／容易な局面化
- ・変更に強いプログラム構造にすることが可能

また、DOA を実施していく場合の手法は次の通りである。

- ・データフローダイアグラム Data Flow Diagram（図 4. 2）

業務におけるデータの流れを4種の単純な記号でモデル記述する。

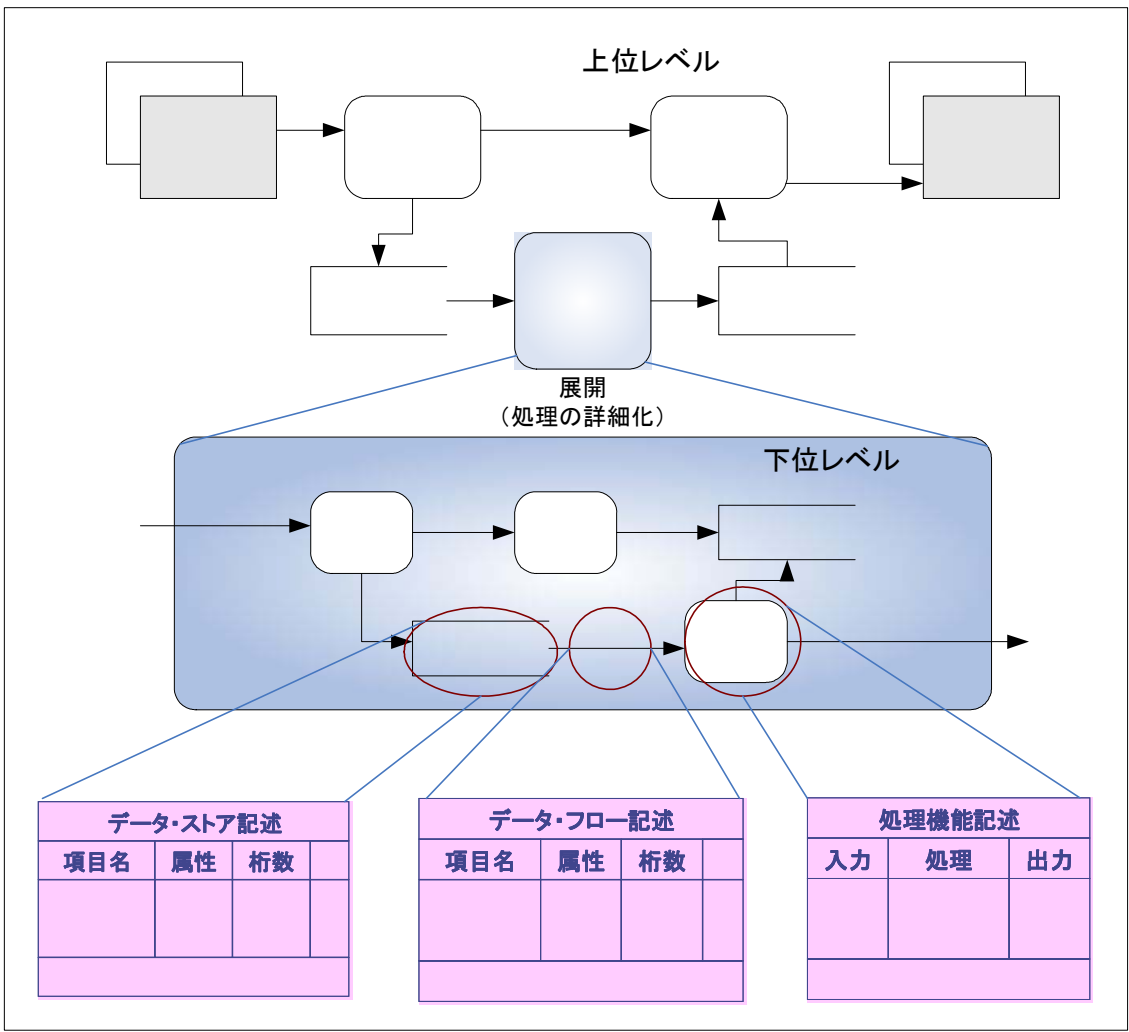


図 4. 2 DFD4 点セット

・データの正規化 Data Normalization(図 4. 3)

DB 設計に先立って、関連性の高いデータ項目のくくり方を理論に基づき最適化する。

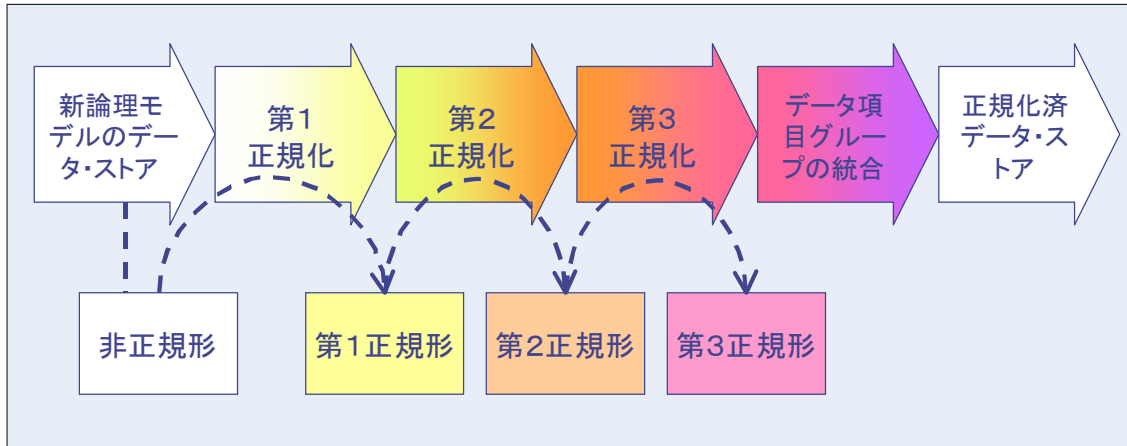


図 4. 3 正規化の流れ

・ER 図 Entity-Relationship Diagram(図 4. 4)

エンティティ(管理対象データ)とエンティティ間の関連を簡易な記号で図示する。

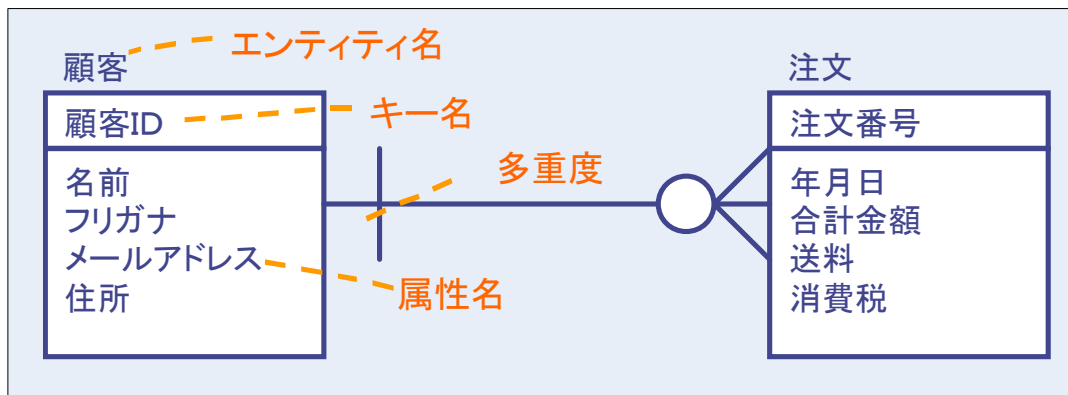


図 4. 4 E-R 図 (IE 法)

・データディクショナリー Data Dictionary

ーデータに関する属性の辞書または貯蔵庫

ーDOA のキーワード「ONE FACT ONE DATA ONE PLACE」

ーホモニム(同音異義語)、シノニム(異音同義語)の排除、適切な用語定義(業務用語集の作成・管理)

・複合設計／構造化設計 Composite Design / Structured Design

ーモジュール分割の良否の尺度(強度／凝縮度、結合度)、複合分析

4. 4. 要件定義工程に関連する諸問題

システム開発プロジェクトで一般的に言われる内容ながら、当社においても、要件定義工程での作業、成果物を原因とする問題が発生している。ユーザ曰く、「こんな仕様は聞いていない」「ここはこうなると思ってた」「この要求が入っていない」等。

なぜ、そうなるのか、原因となったポイントの調査を行ったので、調査結果から代表的なものを列記する。

代表的ポイント：

- (1) 「要件定義書」の確認を怠り、ステークホルダー間の正式な合意があいまいであった。
 - (2) 標準化不足や技術者スキルのばらつきが原因で、顧客要求事項の粒度にばらつきが発生した。すなわちユーザ要求と機能要求が混在した。
 - (3) 隠れたステークホルダーの要求事項があり、「要件定義書」に漏れがあった。
 - (4) 要求事項への変更が管理されていず、合意済みの「要件定義書」が陳腐化していた。
 - (5) 要件定義以降、要求事項の変更を考慮なく受け入れてしまった。
 - (6) 「要件定義書」に基づいたシステムの妥当性確認をしていなかった。
- 等

ウォーターフォールモデルにおける要件定義は、顧客要求(ビジネス要求やユーザ要求)から機能要求を導き出しつつ、開発スコープを明らかにする工程であるべきだが、現実的には、実現できていないことが上記「代表的ポイント」からも認められる。ポイントによっては、ステークホルダー管理や構成管理などのプロジェクト管理系プロセスに課題を持つと考えられるものもあるが、発生タイミングなどの性質から、(1)～(3)は「要件定義能力」、(4)～(6)は「要件管理能力」と分類することが出来る。

4. 5. 要件定義工程の改善

ベンダー側となる当社だけでは解決が難しい代表的ポイントもあるが、CMMI をモデルとした業務プロセスの見直しの一環として、要件定義能力、要件管理能力の向上にも取り組んでいるので、その改善点の一部を紹介する。

(1) 要件定義工程の見直し

顧客との合意を形成するアクティビティの重要性を高め、「事前社内レビュー」アクティビティや「顧客との合意」アクティビティを、テンプレートやチェックシートと組み合わせて実施するように標準工程を見直しする。

(2) DOA 手法の利用勉強会

要求事項のあいまいさや粒度の統一を図るため、DOA手法の説明から例題演習、および習熟者によるノウハウ伝達などの勉強会を開催する。

(3) プロジェクト管理ノウハウの勉強会

システム開発で必要なプロジェクト管理項目についてPMBOKを基準に整理し、標準化する。習熟したプロジェクトマネージャーを講師に、勉強会を開催し、重要ポイントであるステークホルダー管理の意識を要件定義の重要性と絡めて展開する。

(4) 要件管理プロセスの設定

CMMI モデルと当社業務プロセスを比較して、弱いポイントでもあった要件管理プロセスの見直しを行い強化した。従来は変更管理プロセスとして、要件以外のアイテムに関しても同様に管理していたが、要件のみを切り出して、スコープにある要求事項の管理や変更管理、トラッキングなどに関して強化を行っている。

(5) 変更に対して取るべきアクションの紹介

要件に対する変更要求の中には、変更しないままではシステム化の目的や価値を満たせないケースもある。従って、変更要求に対する対処方法をガイドラインとして取りまとめ、社内に展開する。

それぞれの改善活動において、習熟者のノウハウについても重視しており、個人の能力を組織の能力へ移転させることを期待している。なお、原因調査において、習熟者は、開発プロセスや管理プロセスが求める本質を理解し、そのプロジェクトに適切にアレンジして実施していることが確認された。このことはシステム開発プロジェクトごとに、システム開発標準へのテーラリングが重要であると考えられ、要件定義においても同様に思われる。

5. 社内事例紹介と要求工学調査の方向性について

(株)野村総合研究所 荒生 知之

5. 1. はじめに

システム開発が、部分的な事務処理の効率化・自動化から、事業そのものと一体化する形にその性格を変化させていく中で、システムの要件定義がますます困難かつ複雑になっていることは、いまさら繰り返すまでもないだろう。それに呼応する形で、(独)情報処理推進機構 ソフトウェア・エンジニアリング・センターの開発プロセス共有化部会が「超上流」という言葉とともにガイドラインを提案するなど、システム開発における上流工程の重要性に対する認識は近年特に高まり、方法論や各種ツールにも様々なものが提案されてきている。しかし、課題・問題としての重要性は強く認識されているものの、手法や方法論の面では、具体的な解決策について決定的なものがまだ提示されていないのが現状である。

そのような課題認識の下、筆者は、2003 年に要求工学プロセスに関する調査研究を端緒とし、社内にて要求工学プロセスの調査・改善活動を継続的に実施してきた。この報告書では、その2003年の調査プロジェクトの紹介と以降の活動を踏まえ、今後、要求工学をより実践的な分野にしていくための考え方を提示させて頂きたい。

5. 2. 社内事例紹介

5. 2. 1. 要求工学プロセスの課題設定

要求工学プロセスを調査するに当たり、我々自身が考えるあるべきプロセスと典型的な現実のプロセスを比較し、要求工学プロセスにおける課題仮説を設定した。

まず、あるべき要求工学プロセスを「顧客満足度の高いシステムを構築するのに必要な要求を、ある程度の継続的かつインタラクティブなプロセスを踏んで次第に明らかにし、正確に定義するプロセス」と置いた。さらに、プロセスの範囲を「システム化に対する目的・ビジョンが定まり、それに対する現場からの課題とそれを実現する業務イメージが漠然と固まっている段階から、システムに対する具体的な要求を定義するまで」と設定し、あるべきプロセスを次の4ステップに分けて定義した。

- ① システムの前提となる業務イメージを顧客と合意する。
- ② システムに対する顧客要求を早く確実に引き出す。
- ③ 引出された要求を確実に設計に反映する。
- ④ 設計段階以降に変更された顧客要求に対し、早く正確に設計に反映する。

翻って実際の現場では、上記プロセスはどうなっているだろうか。品質・スケジュールが遅延している理由については、現場プロジェクト分析により、主に「システム担当の顧客が、現場の業務をあまり確認しないままシステム要求を決めようとする」「打合せが、画面・帳票を中心に実施され、なぜ、そのようなシステムを作るのかという視点にかけているため、後に現場の利用者にシステムを説明したときに手戻りが多く発生し、スケジュールを逼迫させる」などがあがった。簡単にまとめると下記のようなになる。

- ① 新システムが稼動する前提となる新業務のイメージが漠然としたまま、
- ② 顧客要求を漠然としたレベルで曖昧に抽出し、
- ③ 設計を進める中で場当たりに顧客と要求を確定させ、
- ④ 手戻りが多く発生し、スケジュールや工数が逼迫する。

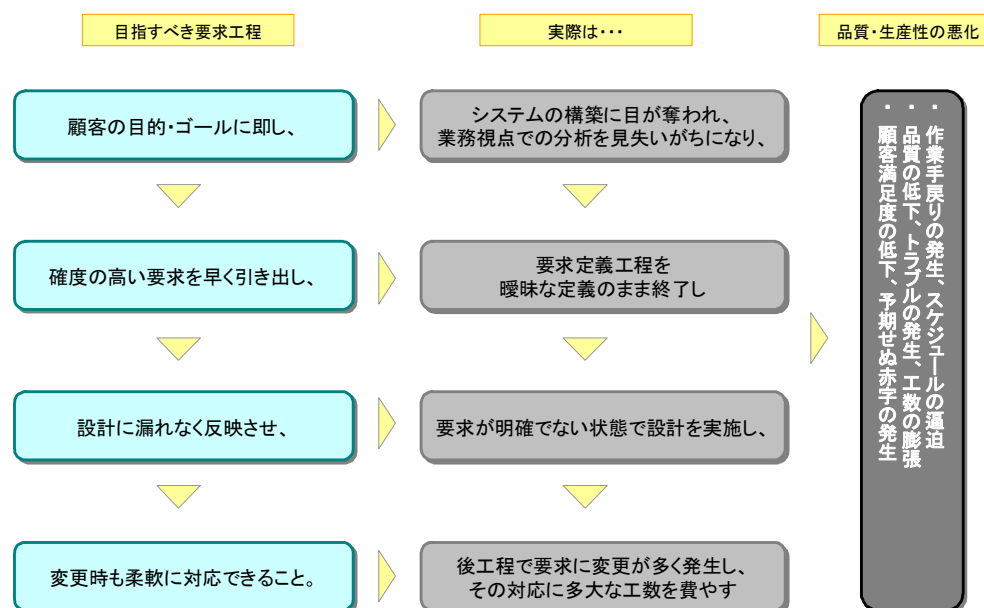


図 5.1 要求工学プロセスのあるべき姿と実際

誰も品質の低いシステムの構築を目指してはいない。正確な要求を定義することの重要性は皆の認識するところだろう。それでは、なぜ現実には要求定義がうまくいかないのだろうか。そこには、現在の技術、考え方では解決しえていない何らかの構造的、組織的な制約などがあるためではないだろうか。調査では、現場のプロジェクト経験や周囲のメンバーからのヒアリングを通じ、ギャップを作っている問題仮説を下記のように整理した。

第一に特に我々が扱っているサービス産業で利用されるシステムの多くは、業務視点で要求が出されるが、システム要求はシステム機能視点で定義されるため、ビジネスレベルの顧客要求とシステムに対する具体的な仕様の間には、複雑な直交性が発生してしまうという点である。即ち、一つの業務の単位に対する要求は、必ずしも、一つのシステムに対する要求ではなく、複数のシステムにわたることがある。逆に、一つのシステムに対する仕様は、複数の業務に対する要求が反映されているということである。（実際のプロジェクトで縦軸と横軸にビジネスとシステム機能との関連性を比較すると、図 5.2 のようにシステムと複雑に絡み合うようになる。）

第二に、システム開発の現場では、システムの構築が優先されるため、機能の設計に関心の中心がいく。現場では、まずはスケジュール内にシステムを構築すること、次の工程にシステムを進めることを目標とする。そのため、画面、帳票、処理概要といった内容を直接決めていくことに

対する優先度が強く、それがなぜ必要なのか、どのように使われるか、という業務の視点を失いがちになってしまう。この結果、業務視点での要求の定義・検証の実施が不十分になる可能性がある。

第三に、顧客要求とシステム要求の整合性を表現する過度に形式的でなく、顧客にも理解可能な技術が存在しないことである。即ち、顧客とのコミュニケーションが可能な具体性を持ちつつ、形式的で正確に要求を記述できるプロセスや技術が無いからである。

	基本機能	Biz Mart 取引先出荷予定データ送信 Biz Mart 取引先発注データ受信 Biz Mart 出荷実績伝票データ再受信 Biz Mart 納品明細表 J/F 照会 Biz Mart 本部発注追加 センター別日付更新 ファイルアップ＆クリア 仮納品明細表発行 共配ルートマスタ照会 共配ルートマスタ登録 検収データ再作成 検収データ作成 検収終了データ修正 検収終了データ組み込み 作業進捗状況確認 仕入確定宣言 取引先機能サービス制御 出荷金額確認（月締め業務） 出荷金額算出処理 出荷実績伝票データ受信情報作成 数量管理（DPT 別） 数量管理（店舗別） 通過型お取引先判定 納品明細表条件指定発行																			
T02_003	通過型のお取引先向けサービス		○	○	○		○														○
T02_005	便別作業への対応				○	○		○	○	○	○		○	○	○		○	○		○	○
T02_008	障害時に対する考慮	○			○		○		○												
T02_011	予約商品受け渡し表の機能改善																				
T02_013	納品明細書再発行の機能改善											○	○	○	○						○
T02_014	作業進捗確認画面の機能改善													○							
T02_016	出荷実績伝票データ配信サービス			○	○			○							○			○			
T02_017	非EDI伝票組み込みサービス							○													
T02_023	非EDI伝票一覧検索																				
T02_024	出荷金額確認（月締め業務）の他法人対応																○	○			
Z02_005	05 本部発注追加						○	○									○				

図 5. 2 業務×システム機能マトリックス

5. 2. 2. 課題に対する解決仮説

5. 2. 1. の課題仮説に対する、具体的な解決方針を下記の 3 点とした。即ち、業務要求からソフトウェア／システム要求までの要求レベル・要求間の関連性の構造化と、その構造を前提とした要求工学プロセスの定義、そして、それを実現するツールの開発である。

a) 要求構造の定義

要求の構造を定義することとは即ち、顧客のゴール、業務的な要求、システム的な要求を、業務とシステム機能が直行的であることを前提とした形で構造化し、業務的な要求、システム的な要求との相互関連性を可視化することである。同時に、構造に基づいた、ドキュメンテーションの構成を作成し、その記述レベルなどに一定の基準を設けることにより、形式的かつ正確な顧客の要求定義が出来るよう、顧客要求を階層的に、かつ、要素ごとに記述することができることを前提とした。また、普段システムにもさほど慣れ親しんでいない顧客にもその内容を理解してもらえよう、わかりやすく、シンプルな構造にすることを重視した。同様に、相互関連性についても、業務とシステムとの関連性以外の関係性についてはシンプルなものとし、トレーサビリティもより単純なものとした。

検討した結果の構造は次のようなものになった。

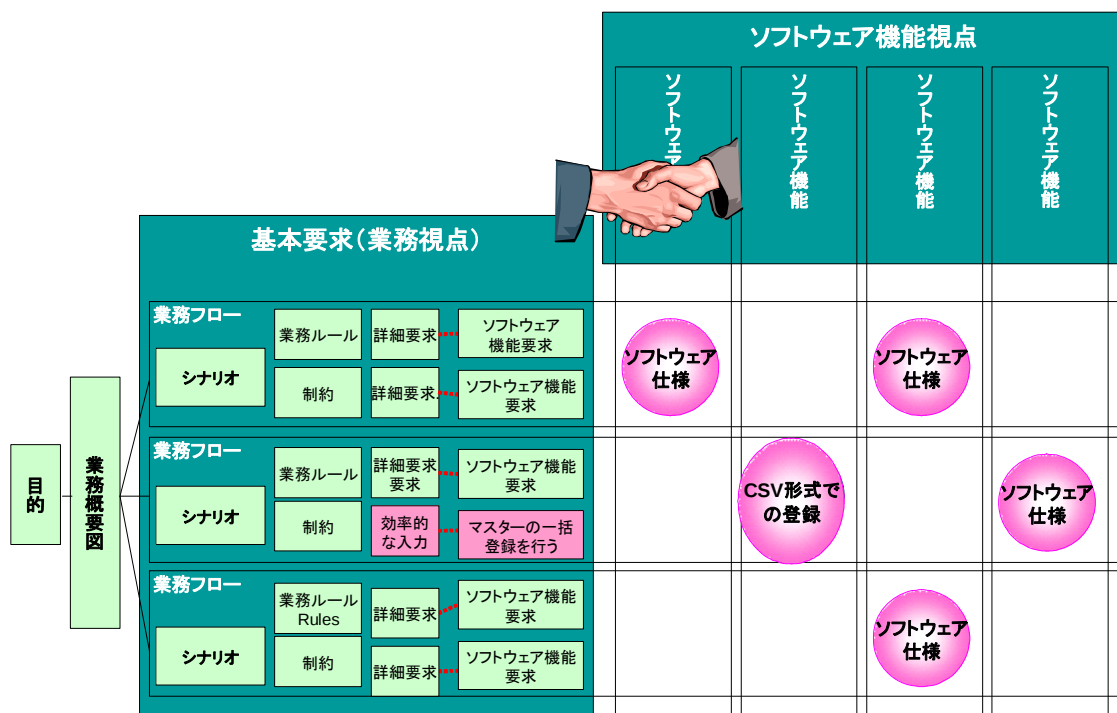


図 5.3 要求構造の図

まず、顧客からの視点を、基本要素とし、それをマトリクスの行単位として定義した。基本要素単位に検討する要件を、その基本要素の配下に、各要素として、詳細要件、ソフトウェア要件として定義した。また、その中で発生した、課題や議論についても、基本要素単位に管理するようにした。引き出した要求は、一つ一つの要素として定義するようにした。

別途、画面、帳票、処理などシステム機能の単位を列として配置した。システム詳細仕様は、機能単位のみで考えるのではなく、必ず、基本要素と機能のクロスする点で、関係する基本要素からその機能に必要なとしている仕様は何かという手順を踏んで決定することを前提とした構造とした。

そのクロスするポイントに要求が落ちれば、あとは、そのクロスポイントを単位に設計を詳細化していけばよく、それ以降の基本設計以降の手順については、既存の開発プロセスに任せればよいと考えた。

b) プロセスの定義

次に、上記の構造に基づいて、顧客からの要求獲得を実施するプロセスを下記の3ステップに定義した。そのプロセスを構造と連携させることで、顧客や当方にとって、各プロセス

で明確にするべき要求のレベル・要素を形式的に意識させることを目指した。

1. 基本要求和システム機能定義

まず、業務の流れに添って顧客視点の要求単位である「基本要求和」を抽出し、顧客と合意をとる。同時に範囲内で必要となる業務支援機能を整理し、システム機能定義として定義する。

2. 基本要求和視点での、業務視点での要求獲得・検証

次に顧客からの要求獲得を実施する。その際は、常に「基本要求和」を意識して、基本要求和配下に詳細な要求を詰めていく。実際に適用したプロジェクトでは、基本要求的のなかも、顧客の業務上して欲しいこと、システムとして具体的に実現したいこととその階層を2段階にわけ、さらに、要求が決定するまでに発生する課題も含めて、全て基本要求和単位に決定・管理した。

詳細な業務要求を定義する際は、シナリオや業務フローを用いて要求獲得を実施し、顧客に業務視点を意識して要求を発想してもらうよう心がける。顧客と検証する際は、その要求要素の上位レベルの要求、下位レベルの要求との関連性に注視しながら、その内容を確認してもらう。

3. 各機能に必要なとされるソフトウェア詳細要求の獲得・検証

システム要求については、獲得された業務要求にしたがって、当社で想定される機能単位にどのような仕様が必要となるかを検討し、顧客とは、上記にあげたマトリックスを見せ、関連する基本要求和を意識しながら、なぜ、そのような機能にしたのかという視点で、レビューを実施する。

c) ツールの作成・構築

第三に、上記の構造とプロセスを実現するツールの開発を実施した。市販のツールに対して、上記の要求の構造を反映するのが困難だったため、結果として、マイクロソフトアクセスを利用した独自ツールを開発するに至った。

5. 2. 3. パイロットプロジェクトへの適用効果

上記アプローチ方針に基づいて、社内のあるシステム構築プロジェクトへの適用を実施した。プロジェクトの概要は下記の通りである。

a) パイロットプロジェクトの概要

弊社がメンテナンスしているある流通関係のシステムに対して、新たな商品カテゴリーに関連した業務・サービスを追加するシステムプロジェクトで、規模は40MM、期間は3ヶ月のエンハンスプロジェクトであった。利害関係者は、顧客の業務部門の担当がメインであり、打合せにほとんど出席いただくことができた。当社は、打合せを通じて、彼らから要求を獲得し、それを要求仕様書としてまとめた。

5. 2. 2. で述べたアプローチ方法は、実際には、このパイロットプロジェクトと実質的に並

行して構築し、下記のような効果を得ることができた。

b) 適用結果

要求獲得の場面では、以前の打合せでは多く見られた、議論の手戻り、蒸し返しが減った。たとえ蒸し返したことであったも、以前の資料をデータベースに入力し、それを基本要求単位に全て整理しながら、顧客と打合せを進めたことにより、過去の議論や検討経緯を常に参照しながら打ち合わせすることが出来た。また、機能面からこのような機能が必要であろうと検討を進めていたことも、業務視点で考えたときに不必要であろうと判断されカットされた要求もあった。このように実際に、プロジェクトを進めている中で、いままでのやり方とは違い、かなり早い段階から、無駄なく、多くの要求が獲得されていることを実感することができた。

以下に、要求獲得・検証のスピード・精度の向上、設計品質の向上という二つの視点からその効果を数量的に示したい。尚、比較対照としたプロジェクトは、同一の顧客・開発メンバーで開発した、同環境における同規模のプロジェクトである。

< 要求獲得・検証の精度向上 >

「システムに対する顧客要求を、早く確実に引き出した」ことを検証することを試みた。しかしながら、以前のプロジェクトでは、要求を構造にしたがって定義していなかったため、要求の数を単純に比較することは困難であった。代わりに、議事録の量(バイト数)を顧客と決定した量＝要求の量と同義と捉え、それを、工程ごと、時系列に表現した。

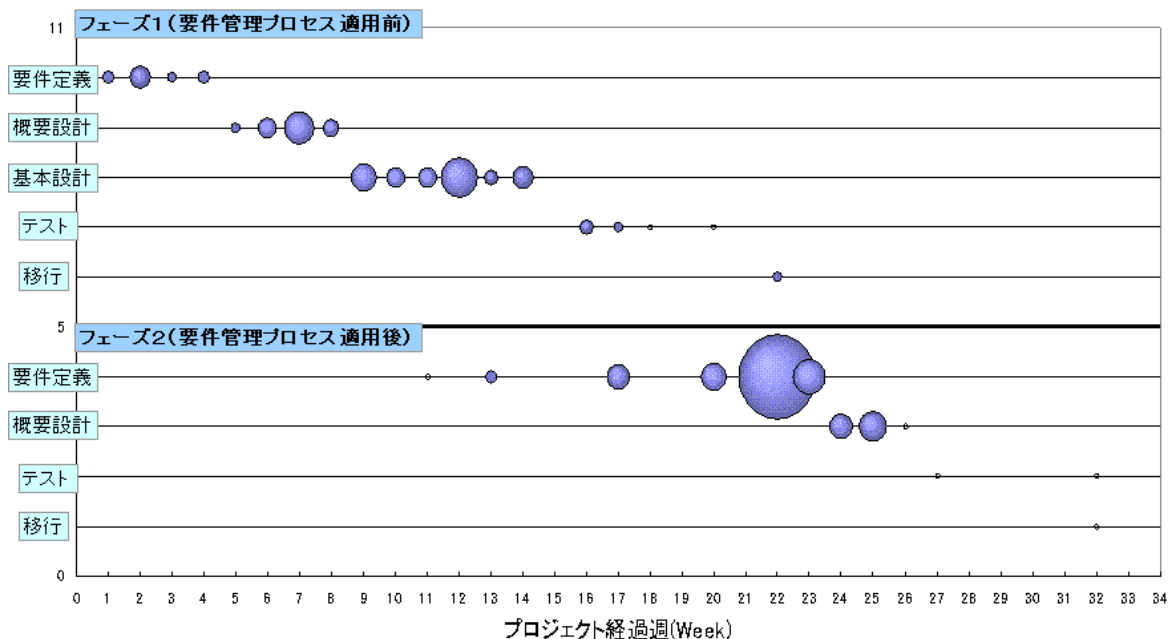


図 5. 4 議事録の量の推移

その結果、今回のプロセスを適用した場合は、以前と比較し、非常に早い段階で、多くの要求を定義でき、反対に後工程では変更がほとんど発生しなかったことがわかる。次にそれがどのように、設計品質に影響を与えたかを見てみる。

<設計品質の向上>

設計品質を見るために、総合テスト時に発生した障害の数とフォロー工数を、障害原因が埋め込まれた工程別にグラフ化したのが下記の図である。

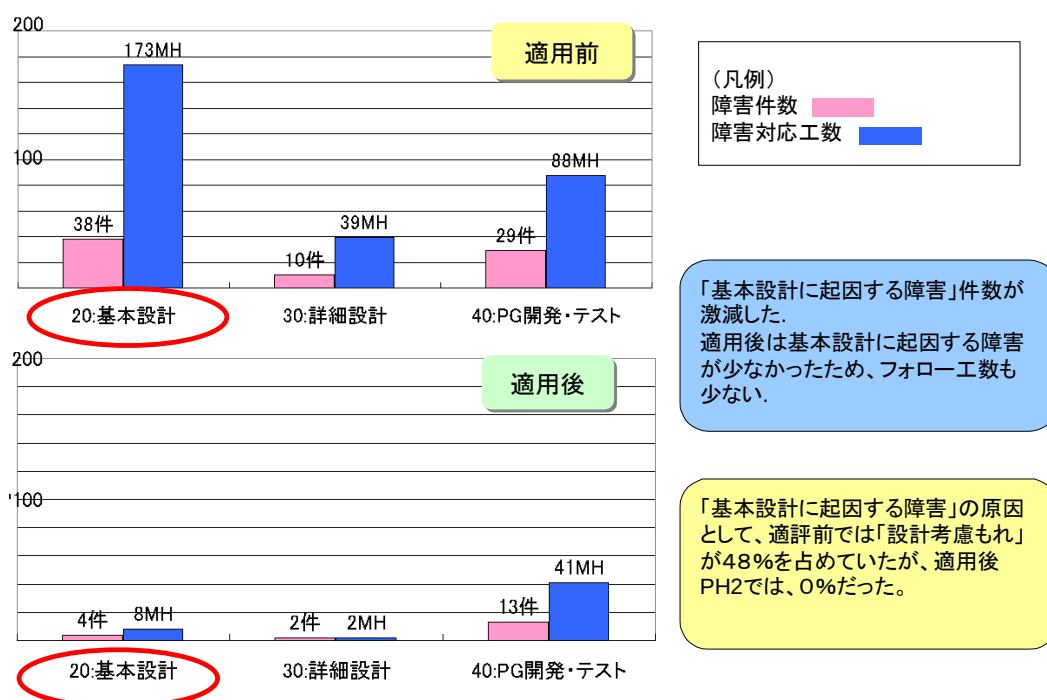


図 5. 5 総合テストにおける障害件数比較

図を見てわかるように、適用後のプロジェクトでは、相対的に見て、基本設計に起因する障害件数が大幅に減少した。基本設計を起因とする障害はフォロー工数も多くなるため、トータルのフォロー工数は非常に多く削減されたことがわかる。

<顧客の声>

担当した利害関係者の評価も非常によく、理解も非常に早かった。

c) 適用効果の分析・前提

上記で述べたように、パイロットプロジェクトにおいて、「顧客視点での要求」視点で要求獲得・検証を維持することが品質・生産性向上に寄与すること、エンジニアリング的工夫で

それが実現できることがわかった。

しかし、今回のプロセスをその後いくつかの他のプロジェクトに適用する中、このパイロットプロジェクトは、適用が成功する要因に恵まれていたこともわかった。

まず、実現したい業務イメージがある程度明確であったことである。今回は、既存の業務をどのように変更すればよいか、比較的イメージがわかりやすいプロジェクトであった。これが、システムの全面再構築など、業務目的が漠然としている場合は、具体的な新業務イメージを描くまでに多くの時間を要するであろうし、その業務イメージへの適切なブレークダウンが実施できていなければ、今回述べている要求工学プロセスにそもそも繋がらない結果となり、プロセスの適用の効果は薄れる。

次に、主たる利害関係者が、業務内容をよく把握し、かつ、それと利害が対立する利害関係者が少なく、また、対立があっても、利害関係者内でうまく調整してくれたことがあげられる。結果として、我々と打合せに出席している利害関係者間での打合せで要求をFIXさせることさえできれば、その要求はほとんど覆ることがなかった。これが、大規模なプロジェクトになると、ある場所で決めたことが、他の場所に持っていくと容易に覆されることが頻繁に発生する。結果、人々は決まったプロセスについては、再度追うことが億劫となり、結論さえ決まればよいと考えてしまう。結果として、要求の構造が維持できなくなる可能性がある。適用する際は、プロジェクト全体で要求の構造について合意できていることが前提となる。

最後に、要求分析者が比較的有能な人間であり、また、彼がほとんどの要求を分析し、整理していったことがある。特に規模が大きく、新規に立ち上がったプロジェクトになると、様々な能力レベルの人間が混在せざるをえない場合が多い。また、要求定義段階で統率が崩れ、リーダー不在の中、各自がばらばらに要求を検討するような形になることも多い。要求工学プロセスは、手順どおり決めることを促し、また、相互の整合性・関連性を可視化、要求を管理しやすくするが、要求を決めていくのは人間である。特に、要求獲得・検証プロセスは、コミュニケーションや発想が必要とされる人間的な作業であるから、要求分析担当の資質・素養が大前提になるのはいうまでも無い。

5. 3. 実践的な要求工学の為に

上でも述べたように、上記適用事例は成功したが、社内に適用を拡大する中で、効果がみられるプロジェクトもあれば、反対にプロジェクトを混乱させてしまう場合もあった。効果がまばらであった要因は、上記の手法が要求工学プロセスに関するさまざまな課題の一部に対する解決策であり、実際の要求工学プロセスは他にも、様々なレベルにおいて課題・問題を抱えている場合が多いからである。また、上流工程の課題・問題に相互依存性があり、たとえば顧客との合意プロセスに問題があるプロジェクトにおいては、仕様記述がどれほどうまく実施されても、それがリニアにシステム全体の品質をあげることにもつながらない。

要求工学プロセスに関連する課題・問題の領域を下記に例示する。

1. ユーザー側の要求定義の品質
2. ユーザー側のプロジェクト体制・利害関係者の参画
3. ユーザー・ベンダー間の要求合意・変更プロセス

4. プロジェクト特性に応じた上流プロセスのカスタマイズ
5. 現行業務・システムの把握・整理
6. 上流工程プロセス&手法の体系化
7. 上流工程での工数・工期見積
8. ドメイン別業務・設計ナレッジの蓄積・共有化
9. 要求の構造化・トレーサビリティ
10. 要求管理手法・技術
11. 上流工程の定量化・計数化
12. 要求定義仕様の記述・レビュー手法
13. 上流工程と設計・テストプロセスの連携
14. 上流工程を担当人材の育成

5. 2. で説明したパイロットプロジェクトは、上記一覧の1～5や8、12、14については、すでに問題がないプロジェクトであった。それが故、6、9、11 の課題についての解決仮説の適用と効果を見出すことができたものであるといえる。

しかし、多くのプロジェクトにおいては、1～3 の顧客との要求合意の観点で課題があったり、8 や14 などベンダーが側のノウハウや提案力に課題を持ったりする。

今後、要求工学をより実践的にしていくためには、上記であげたような要求工学プロセスにおける課題について、プロセスモデルなどを利用して網羅的に整理し、解決技術とのプロセスをマッピングすること、さらに、プロジェクトに適用するに当たっては、プロジェクトのリスク要因と解決策を結びつけることによって、各技術を効果的に適用し、プロジェクトのリスク・課題を効果的に解決するフレームワークを構築していくことが重要になるであろう。

6. 「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

(株)NTT データ 平岡 正寿

6. 1. MOYA 概要

ソフトウェアを開発するためには、いかに作るか(How)、何を作るか(What)、なぜ作るか(Why)を明らかにすることが重要であるといわれています。

このうち、いかに作るか(How)については、フレームワークや開発プロセス、自動化など、多くの企業で積極的な取り組みがなされ、かつ数多くの知見が蓄積されており、かなりのレベルにあると思われる。

しかし一方、何を作るか(What)、なぜ作るか(Why)についてはその重要性の認識とは裏腹に知見が蓄積されているとはいいがたい状況です。

そこで、NTT データでは 2004 年に要求定義の精度を向上させるために、SSM(ソフトシステムズ方法論)やゴール指向分析、UML による多視点モデリングなどといった様々な手法をうまく組み合わせ、統合したものとして MOYA を策定しました。

MOYA とは、Model-Oriented methodology for Your Awareness の頭文字をとったもので、要求定義における問題を、ステークホルダの気づきを導くことによって、要求定義の精度をあげようというものです。

要求定義の精度が高くないために、起こる問題とは何でしょうか。

我々は大きく以下のような問題があると考えており、それらを解決することがよりよい要求定義につながるのではないかと仮定しました。

- ・不完全な記述
- ・最適化視点の欠落
- ・目的が未達成
- ・課題が不明確
- ・合意が未成立

上記の5つの問題の根源に存在する以下の5つの要素(ピンクの丸印の部分「関係者」「課題」「目的」「手段」「業務」)をモデル化することこそが、よりよい要求定義であると考えました。

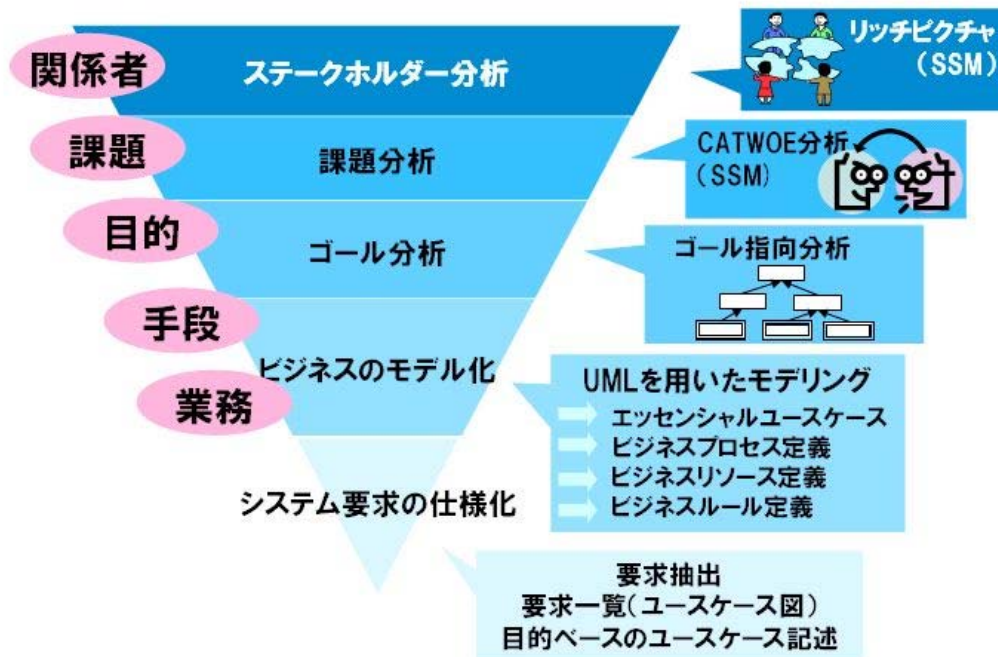


図 6. 1 要求定義を行うためのモデル化技術

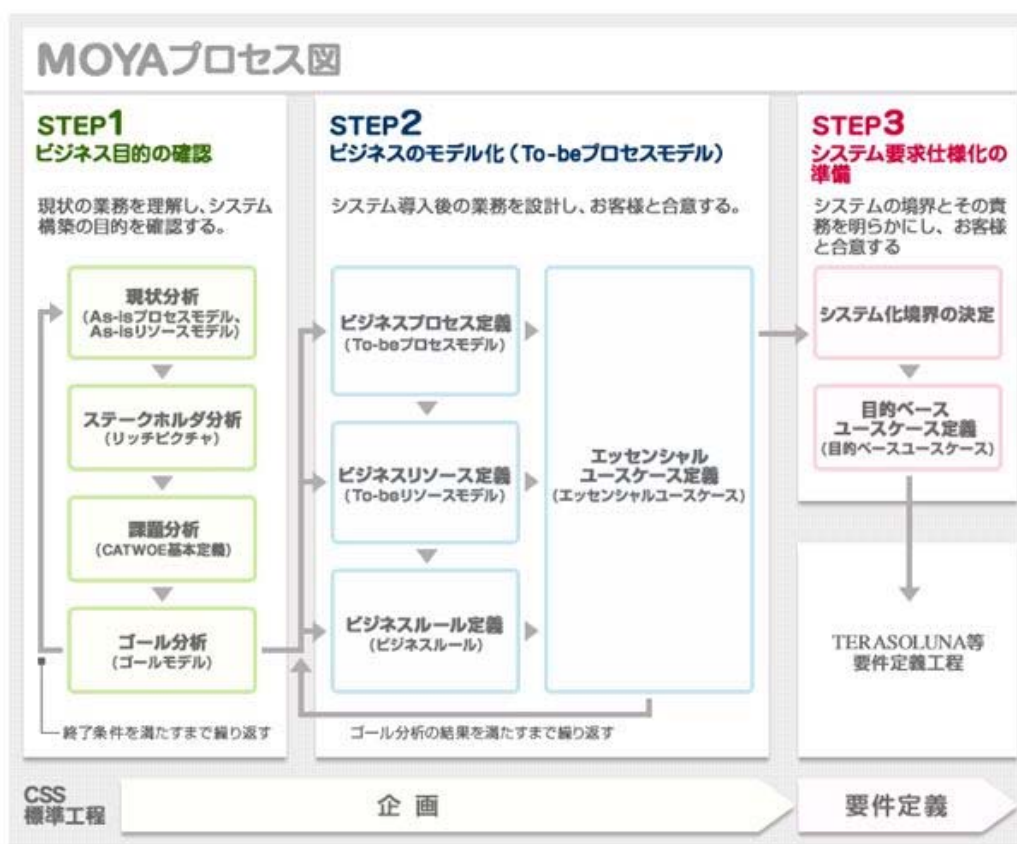


図 6. 2 MOYA 全体像とプロセス

(1) 関係者

【主な分析プロセス名】

ステークホルダ分析

【利用する主要なモデル】

リッチピクチャ

【概要】

リッチピクチャと呼ばれる自由に記述できる絵を用い、分析したい状況に登場するステークホルダの立場、課題や思い、お互いの関係・影響などを描く。特に記述の制約はなく、いかに見る人に対して一目でステークホルダ間の関係や、課題、思いが伝わるように書くか、ということが重要となります。

(2) 課題

【主な分析プロセス名】

課題分析

【利用する主要なモデル】

CATWOE 基本定義

【概要】

課題を分解する CATWOE 基本定義という手法を用いて、課題を6項目の要素に分解し、その課題に関わるステークホルダ、その課題と課題状態をそう考える価値観、その課題の解決状態、制約条件などを把握することで、価値観を共有するものです(CATWOEとは、以下の項目の頭文字をとったもの)。

C: Customers…課題が解決するとうれしい人々

A: Actors…課題を実際に解決する人々

T: Transformation Process…課題状態 T(Pre)から、その課題が解決した状態 T(Post)の記述

W: Weltanschauung…課題解決を正当だと考える価値観

O: Owners…課題解決を許可(もしくは拒否)できる人

E: Environment…課題解決のための資源や制約、環境条件など

(3) 目的

【主な分析プロセス名】

ゴール分析

【利用する主要なモデル】

ゴールモデル

【概要】

課題分析で抽出した課題解決策やその価値観などを、要求定義における1つの要求(=ゴール)とみなし、ゴール群をツリー構造として表現することで、より上位のゴール(真の解決策)やより下位のゴール(解決の手段)を導き出します。

(4)(5) 手段・業務

【主な分析プロセス名】

ビジネスプロセス定義・ビジネスリソース定義・ビジネスルール定義・システム境界の決定

【利用する主要なモデル】

UML(アクティビティ図、クラス図)

【概要】

上記で求めた手段レベルのゴールを達成するために必要となる業務モデルや情報構造を UML を利用し表現します(ただし、ビジネスルールは自然言語記述としています)。

6. 2. 適用事例とそのフィードバック

6. 2. 1. 事例紹介

【事例名】

かがやきぷらん II (介護事業所業務支援パッケージの開発)

【業務概要】

もともと NTT データは、「かがやきぷらん」という介護サービス事業者に向けての事業経営効率化のためのパッケージを 2000 年に提供していましたが、開発後(当時)5年が経過し以下のような課題を抱えていました。

- (1) 2006 年に介護保険法が改正される
- (2) 介護サービス事業者の業務運用形態が変化
- (3) システムへの機能要請・性能改善要請
- (4) 回線の大容量化・ハードウェア含めた低価格化によるシステムのあり方の変化

【MOYA プロジェクト概要】

2004.10～2005.03 の 6 ヶ月間で業務分析とシステム要求定義、営業方針を策定するプロジェクトとして、開始しました。

プロジェクトのチーム約 10 名、MOYA チーム約 4 名の体制で実施したプロジェクトであり、以下の最終成果物を作成しました。

- (1) ゴールモデル
- (2) 業務フロー
- (3) ユースケース図
- (4) 機能一覧
- (5) 営業プラン

6. 2. 2. 事例詳細

(1) ステークホルダ分析

現パッケージに対してのヘルプデスクや、現パッケージ販売を行う営業担当者に蓄積された多様な情報と、パッケージを開発する内部関係者の様々な視点をステークホルダ間で共有し、整理することで、お互いの持つ問題意識・全体状況の理解を進めることができました。

この段階で重要なことは、ここでは解決を導かないことです。リッチピクチャでは、解決に向けて、お互いの状況を共有・理解・把握することが重要であり、解決を急がないことが重要です。

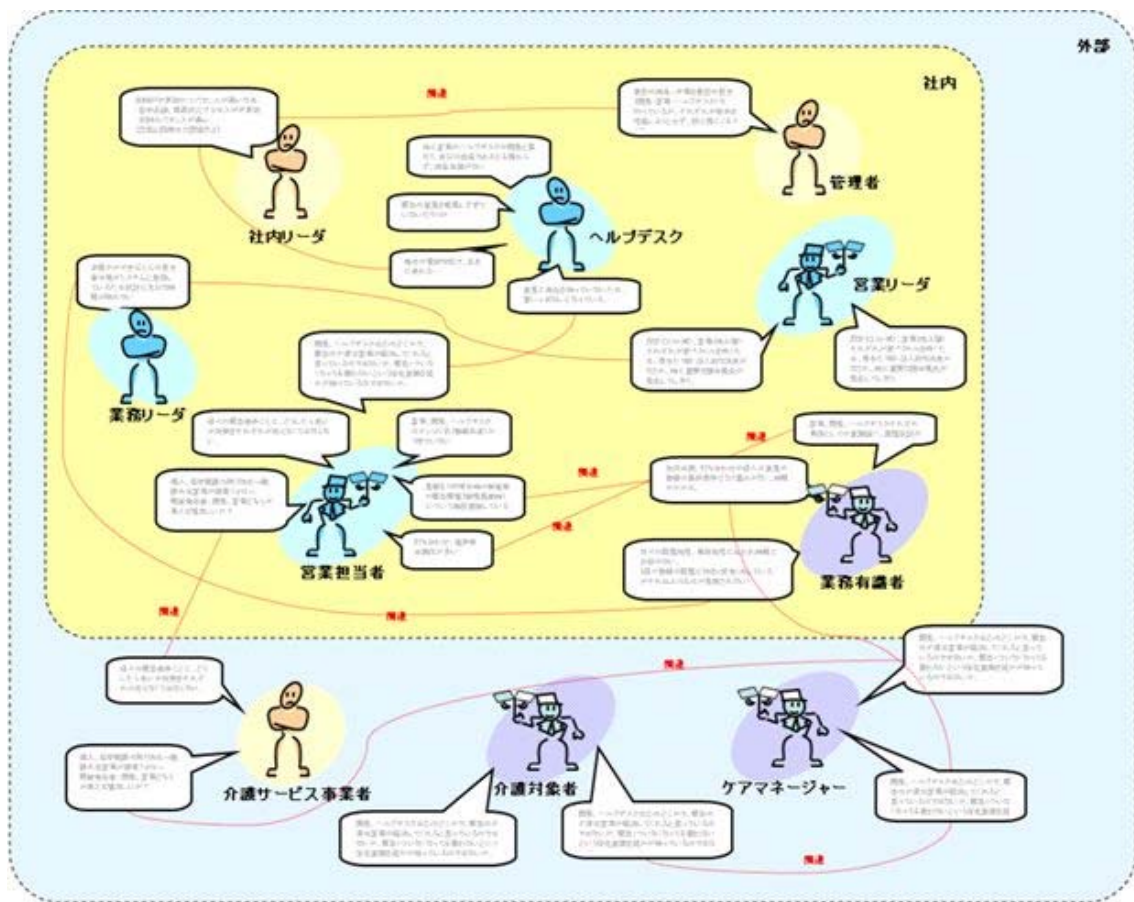


図 6. 3 リッチピクチャによるステークホルダ分析

(2) 課題分析

リッチピクチャで抽出された重要と思われる問題を詳細に分析し、そこに横たわる価値観を共有できるまでディスカッションを進めることで、問題となる状況が、ある価値観を共有することで、よい状況に導けることを共有しました。一例としては、営業の「カスタマイズしないから売れない」、開発の「カスタマイズはリソース・品質等の問題があるからやらない」という二者択一の話し合いではなく、その双方の思いを汲み取った価値を共有することができました。

◆ 重要課題-パッケージのカスタマイズ

C:パッケージを利用する事業者様

A:NTTデータ 営業担当、開発担当

T: (Pre)お客様からの要望について、要件の中身を十分吟味しないまま「カスタマイズできない」として販売機会が失われている

(Post) Plug-inによるカスタマイズが実施できている状態。また、共通的に展開できるものについては、機能拡張されている状態

W : ・機能面、心理面の双方において、“安い／早い・速い／使いやすい”の三要素が揃うことが、お客様に満足してもらえるということだ。
・お客様に共通の必須機能を追加すれば、競争力強化になるはずだ
・お客様からの要望をきちんと吸い上げることで、「お客様と共にシステムを作り上げる」イメージをもっていただけることは望ましいことだ。
・それにより、地域の他の事業者を呼び込んでもらうことはよいことだ。

図 6. 4 パッケージのカスタマイズにおける、CATWOE 基本定義の抜粋

(3) ゴール分析

重要な課題について分析した結果の課題解決や価値観をゴールモデル化し、上位の重要なゴールと、そのゴールを達成するために必要な手段(下位ゴール)を明確にしました。

これまでヘルプデスクでの課題やクレーム、社内外からの要望・不満、組織間の課題が渾然とした状態から、ゴールモデルで要求の構造化が果たせました。

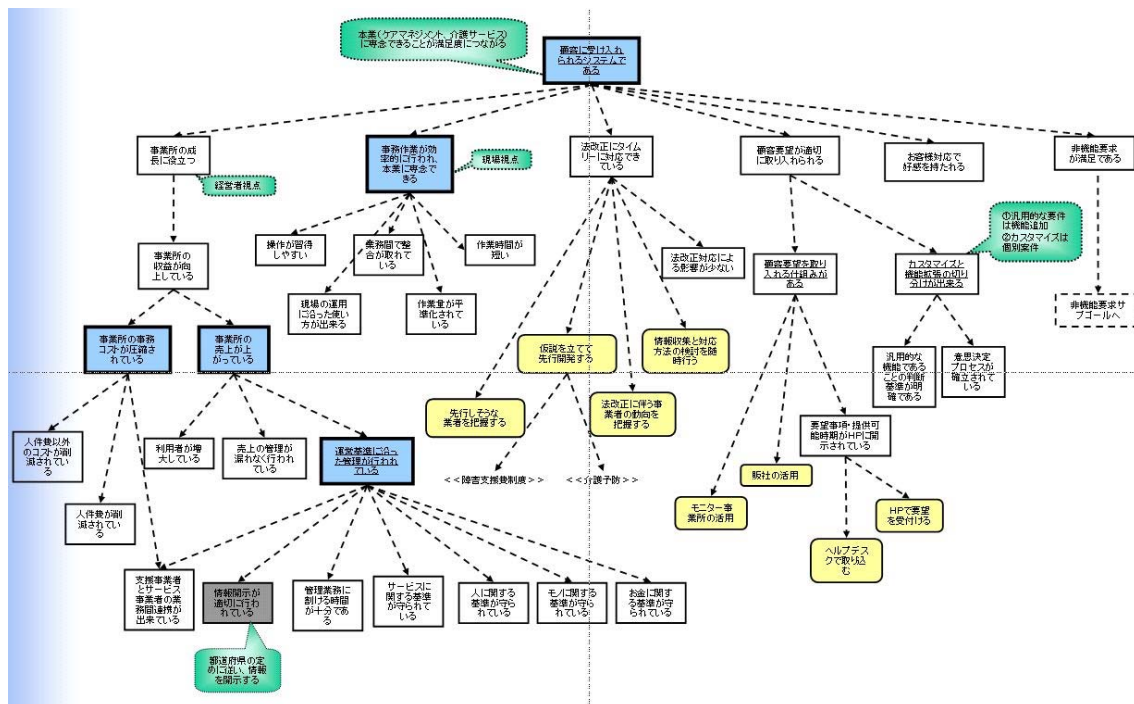


図 6. 5 ゴールモデルの顧客満足に関する部分の抜粋¹

(4) システム設計分析へ

ゴールモデルから、かがやきぷらん II の機能を導出し、それらを実現するための ToBe プロセスモデル・ToBe リソースモデルをアクティビティ図と ER 図で表現しました。ここでは、通常クラス図を用いますが、内部のメンバーのスキルにマッチした ER によってリソース構造を表現することにしました。

この結果を受け、実際のシステム開発の工程が開始されました。

¹実際にはもう少し定量的な内容やゴールの達成確率などの内容が入っている部分もありますが、今回はその部分は除いています

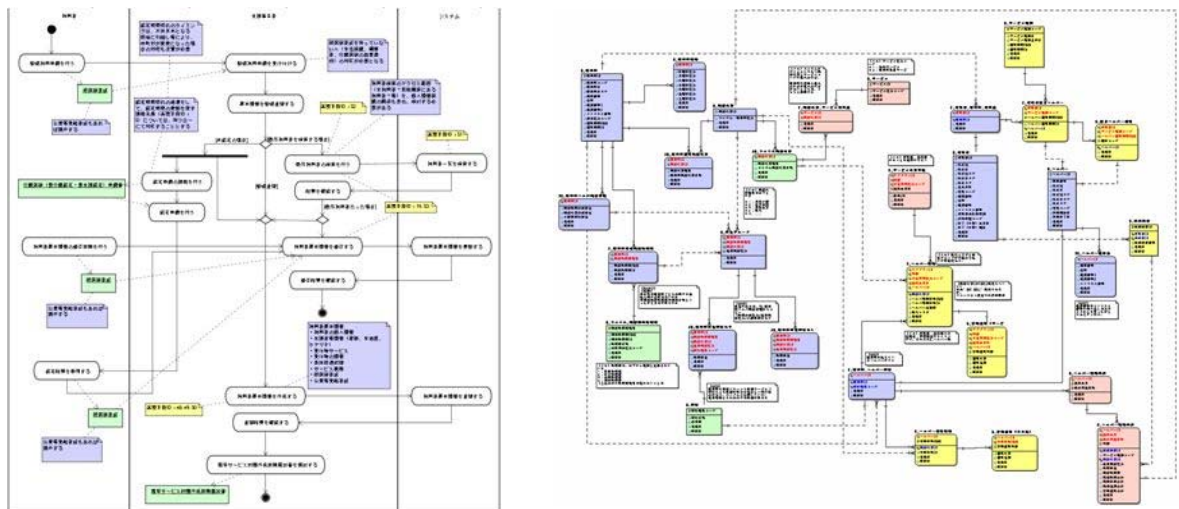


図 6.6 アクティビティ図と ER 図

6. 2. 3. フィードバック

前述の適用事例を含めた事例から多くの要望をもらい、その結果を MOYA の改善につなげました。

改善内容は次章とし、ここでは主なフィードバック内容をリストしておきます。

- (1) MOYA で、システム要求定義以外のことは実施できないものか
- (2) ゴール分析を実施していると、誰のためのゴールを作成しているのかわからなくなる
- (3) MOYA のタスクの進め方が属人性が高いため、もう少し汎用化できないか
- (4) MOYA STEP1 には時間がかかってしまう(もう少し時間を短くできないものか)

6. 3. MOYA2.0～MOYA の改善～

前述のようなフィードバックに対して MOYA の改善を以下のように図り、新しい MOYA2.0 を策定しました。

(1) MOYA で、システム要求定義以外のことは実施できないものか

「MOYA 自体をシステムの要求定義だけでなく、方針策定やひいては提案活動に活用できないか」など、かなり多くのフィードバックがあった内容です。

システムに関する要求定義のプロセスと、戦略・方針に関する要求定義のプロセスとを分け、2つのルートを用意することで、具体的なシステムを前提にした分析だけでなく、部署の方針策定や戦略策定、提案活動などに対しても MOYA が活用できるように定義しました。

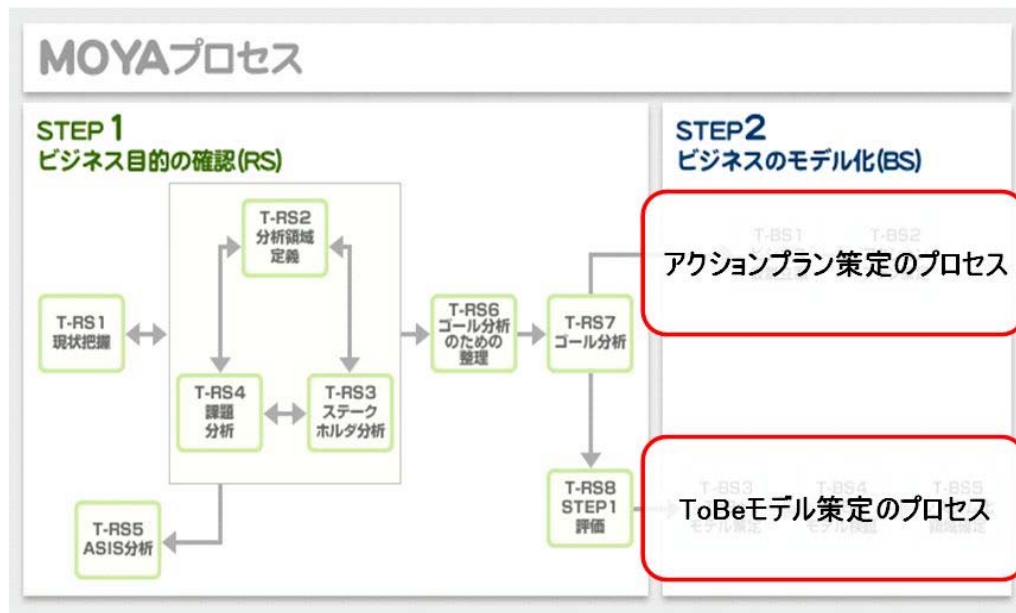


図 6. 7 2つのプロセスを使ったビジネスのモデル化²

- (2) ゴール分析を実施していると、誰のためのゴールを作成しているのかわからなくなる
 分析スコープを検討するタスク(分析領域定義策定)を用意し、「誰のための、何のための」分析であるかを明確に意識できるようにしました。



図 6. 8 分析領域定義タスクを用意する

² アクションプラン策定のプロセス・戦略・方針に関する要求定義 -と、ToBe モデル策定・システムに関する要求定義-のプロセスとの2つのルートを用意

(3) MOYA のタスクの進め方の属人性が高いため、もう少し汎用化できないか

MOYA では、ある程度属人的になるのは仕方ない部分もありますが、属人性を少しでも排除できる可能性のある部分として、プロセスのカスタマイズを取りあげました。

MOYA を進めるプロセスのカスタマイズについて、一般的に利用する可能性が高いプロセス＋成果物を定義し、なるべく誰でも使えるようなプロセスと成果物を定義しました。

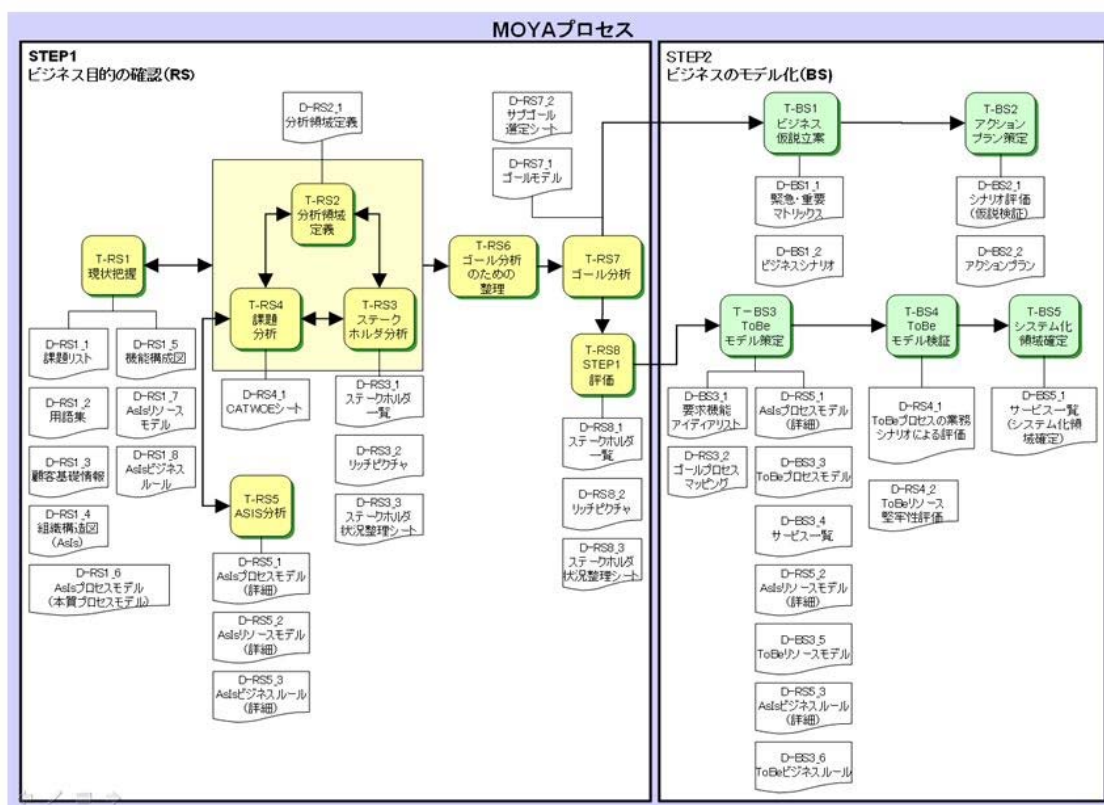


図 6. 9 MOYA プロセス全体像

(4) MOYA STEP1 には時間がかかってしまう(もう少し時間を短くできないものか)

MOYA の本来の目的から考えると、ステークホルダとのディスカッションや分析作業自体の時間を単純に削るべきではありません。これらの時間は必要だと思われるのであれば、それだけの時間をかける必要があります。

そこで、削ることができそうなプロセスのカスタマイズや成果物の定義などの時間を短縮するために、典型的なプロセスと成果物のテンプレートを用意しました。この内容は(3)と同様となります。

上記を中心にして様々なフィードバックをもとに MOYA を改善し、2007 年 1 月に MOYA2.0 をリリースしました。

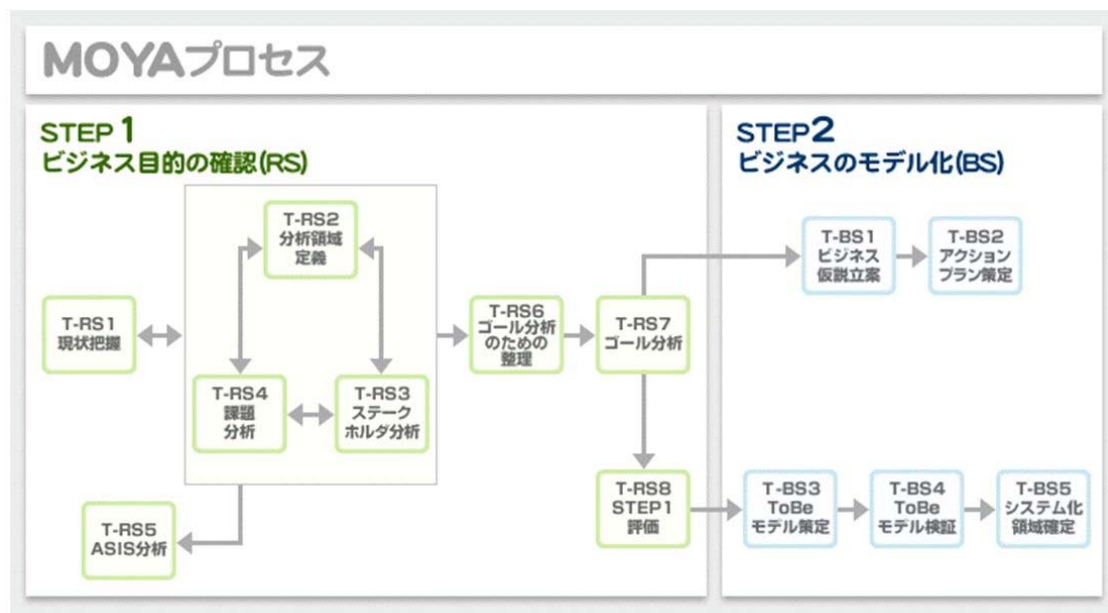
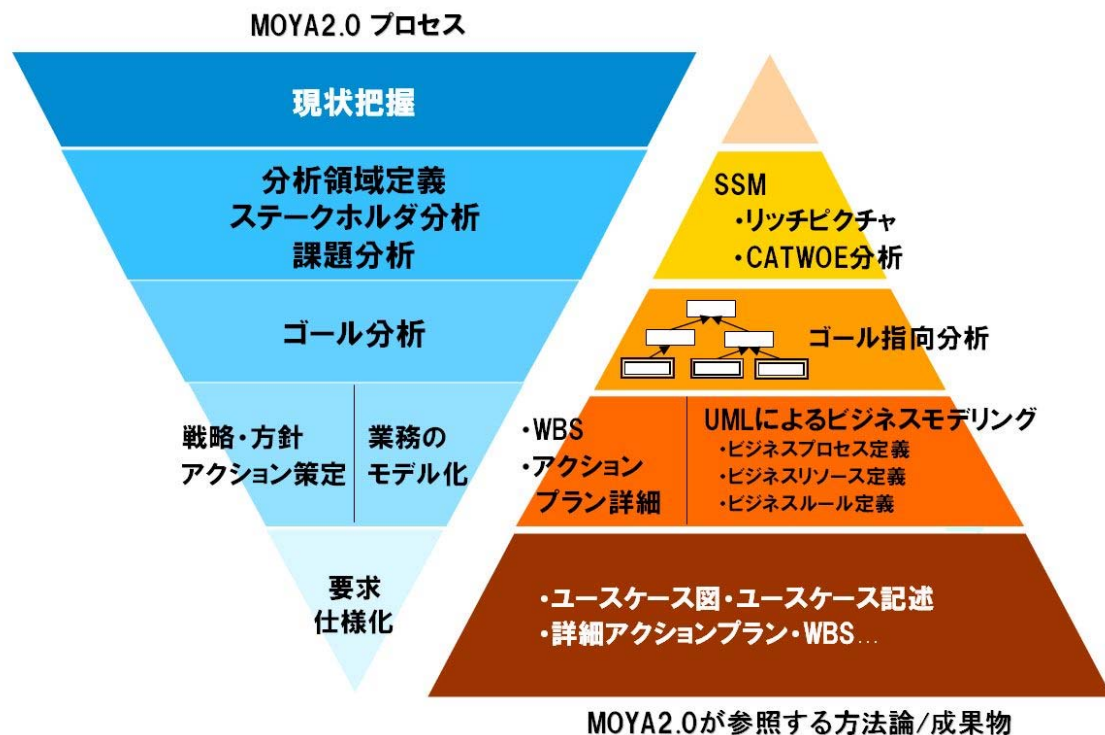


図 6. 10 MOYA2.0 全体像とプロセス概要

6. 4. 今後の課題と展望

今後の課題としては、大きく3つあげることができます。

(1) ツールの提供

MOYA2.0 のプロセスを実行し、トレースするためにはツールのサポートが不可欠です。

現在は手作業で行っている番号付けやトレース作業が自動化されることによるメリットは非常に大きいと思います。ただし、専用のツールが必要となったり、専用のツールがないと成果物が後工程で利用できなかったりしては普及のさまだけになると思われますので、いかに手軽に提供できるかが重要となります。

(2) 効果測定

「MOYA が要求の品質にどう寄与しているか、後工程にどういう影響をもたらすか」についていかに数値化するかは非常に重要な課題です。

これらの数値がある程度明確になることが、利用する側に対しての大きなインセンティブになるからです。

今後は、

- ・品質を図るための指標の設定・後工程への影響を図るための指標の設定
- ・それらの数値を取得するための仮説設定
- ・数値の取得と仮説の検証

を繰り返し実施し、常に設定指標と設定数値をリファインしていくための活動が重要と考えます。

(3) 見積もりへの寄与

我々SIerとしての立場からすると、「このような要求に対してシステム開発を行う際の見積もり」の指標にMOYAが使えることには非常に意味があります。

現在は、この部分にはまったく手をつけていないところであり、見積もりに対しての指標と数値を設定し、常にそれらをリファインしていくことが今後の大きな課題と考えます。

7. 要求定義における価値創生に関する考察

三菱総研 DCS(株) 眞木 康裕

7. 1. はじめに

構築したシステムに対する満足度が決して高くなっていない現状において、顧客満足度の向上並びにSI業界における新たなビジネスモデルの創生について考察する。

7. 2. システムを構築する目的

7. 2. 1. システムへの期待

70 年代に事務処理の代替手段として本格的に利用されてきたコンピュータは、現在ではあらゆる分野で身近なものとして利用されている。

ここに至って、利用者の立場によってシステムへの期待(=要求)は異なってきている。

7. 2. 2. 要件の確認

情報システムに求められる要求を決めるプロセスを考えてみる。

複雑なシステムでは、要求が数百件以上になるのは当然である。要求にはさまざまな種類、レベルが存在していても、ある一定の基準に従い要求を整理し、必要度が高いと判断された要求を要件としてとりまとめ、システムを構築するのが一般的なプロセスである。

つまり、利害関係者のさまざまな要求に対し、ある一定の基準で整理した「要件」とは、「何を作ればよいのか」を定義した成果物であり、「要件定義」はプロジェクトの成功基準を定義したバイブルであるといえる。

通常のプロジェクトでは、発注者が要件の整理を行い「要求」を定義するが、自らが定めた要求にもかかわらず、構築されたシステムに対する満足度は決して高くなく、プロジェクトの約 20% はシステムの出来に不満を感じているとの調査結果も発表されている。これは、必要性が低いと判断した要求の中に、真に必要な要求が含まれていたと考えることが出来よう。

この不満を解消するためには、受注者は発注者から提示された要件を鵜呑みにせず、あらゆる利害関係者の要求を検討した上で取りまとめられた要件であることを検証し、要件へフィードバックすることが重要であるといえる。

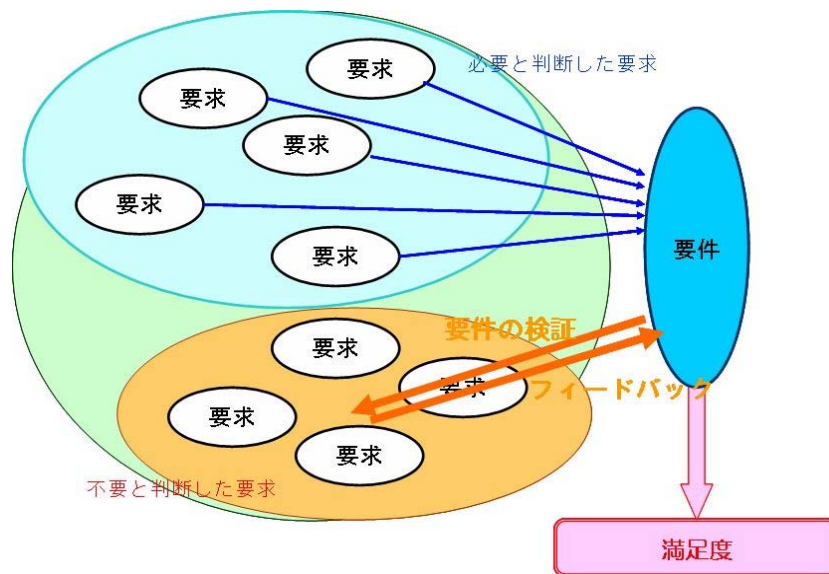


図 7.1 要求の整理

7. 3. 要求に求められるもの

一般的に、システムを構築する際に発注者は要求分析工程として、要求を整理し機能要件を定義するが、後工程であるシステム設計においては、総じてレビューのみに参画するケースが多い。

つまり、実際に構築されたシステム(=要求の集合)の検証は、運用テスト段階になって初めて可能なのである。

とはいえ実際に運用テスト段階で要求が適切に反映されていないと判明したとしても、開発費用は確定し、新たな費用や再調整の時間を伴う上流工程へはもはや戻ることは出来ない。

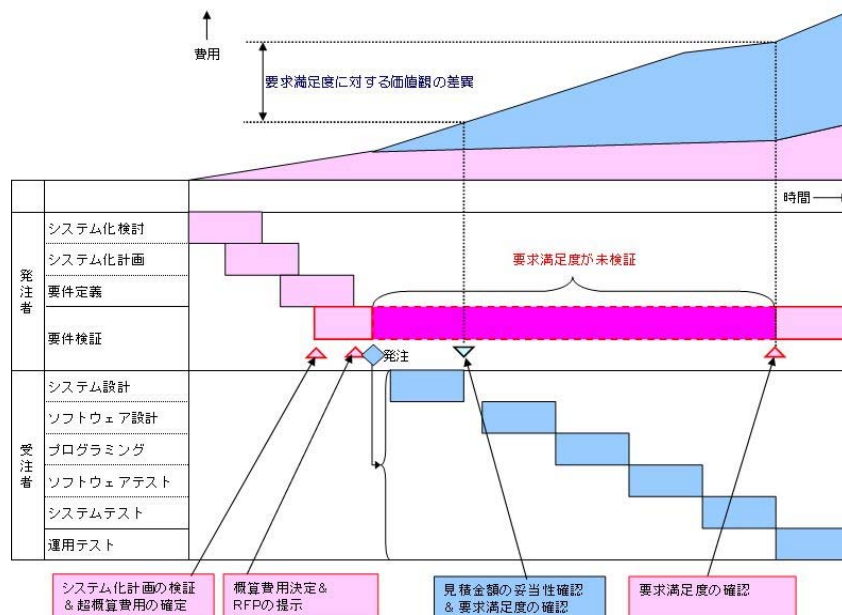


図 7.2 開発スケジュールとコストの相関

超上流工程

超上流工程の範囲として要件定義まででは不十分

業務

システム

ソフトウェア

要求

戦術の立案
→ システム化検討

システム化計画

要件定義

システム設計

ソフトウェア設計

プログラミング

システムテスト

ソフトウェアテスト

運用テスト

要求の検証

(1) 発注者

システムを利用することで自社のビジネスに貢献することを目的としている。

①經營者

経営方針の反映による会社への貢献

②企画部門

③利用部門

①經營者

②技術者

この利害関係の衝突が、システムの仕様が不完全な状態でプロジェクトに着手する主要因となっており、結果的に満足度の低いシステムを構築してしまっている。

7. 5. ユーザの現状

要件の定義、実現、検証はユーザの責務であるが、以下の理由にて実施できていない場合が多い。

(1) 要求の多様化

- ・利害関係者の増加(社内→社外)
- ・インフラ(メーカ固定→オープン)
- ・レスポンス、セキュリティ

(2) 技術スキル不足

(3) 業務スキル不足

専門領域への細分化

(4) 時間不足

競争力強化による担当領域の拡大、効率化

この問題の解決に向けては社内有識者の提案が求められているが、多くのユーザは社内有識者に対して、「企画提案力」が不足していると感じていることより、外部からの支援が必要な状況となっている。

7. 6. ビジネスエンジニア（仮称）とは

7. 6. 1. 役割

企業における活動領域において、受注者側のエンジニアは主に「ソリューション設計」、「コンポネント設計」、「ソリューション構築」に従事しており、発注者側のシステム部門と共同で作業する場合がほとんどである。

従って、「ビジネス戦略策定」、「課題整理／分析」における決定事項がソリューションに反映されているかを検証できるのは、発注者側のシステム部門のみである。

しかしながら、「ソリューション構築」の主体は受注者側のエンジニアであり、発注者側のシステム部門はほとんど関与していないのが実態である。

このため、完成したソリューションがビジネス戦略とミスマッチであっても、受注者側のエンジニアが検証する手立てがないのが現状となっている。

このギャップを解消するために、「ビジネス戦略策定」、「課題整理／分析」の段階からプロジェクトに参画するエンジニア(ここでは「ビジネスエンジニア」と呼ぶ)の役割が重要となってくる。

企業における活動領域（責任分担）

活動領域		経営戦略策定		戦略的情報化企画		開発	
		経営目標/ビジョン策定	ビジネス戦略策定	課題整理/分析	ソリューション設計	コンポーネント設計	ソリューション構築
発注側	ユーザ部門	目標/ビジョンの確認	ビジネス戦略の確認				
	システム部門		ビジネス戦略策定の助言	プロジェクト基本計画の策定	プロジェクトの管理統制 ソリューションアーキテクトの設計	プロジェクトの管理統制	プロジェクトの管理統制 アプリケーションコンポーネントの検証
受注側	営業		ビジネス戦略策定の助言	ソリューション提案	プロジェクト基本計画の策定		
	PM				プロジェクト基本計画の策定	プロジェクトの管理統制	プロジェクトの管理統制
	SE/PG				アプリケーション開発計画の策定	アプリケーションコンポーネントの設計	アプリケーションコンポーネントの開発
ビジネスエンジニア BE			ビジネス戦略策定の助言	ソリューション策定の助言	ソリューションアーキテクトの検証	アプリケーションコンポーネントの検証	アプリケーションコンポーネントの検証

図 7. 4 企業における活動領域

7. 6. 2. スキル

ビジネスエンジニアには、オールマイティなスキルが求められるが、特に、ビジネススキルと技術スキルを兼ね備えることが望まれる。

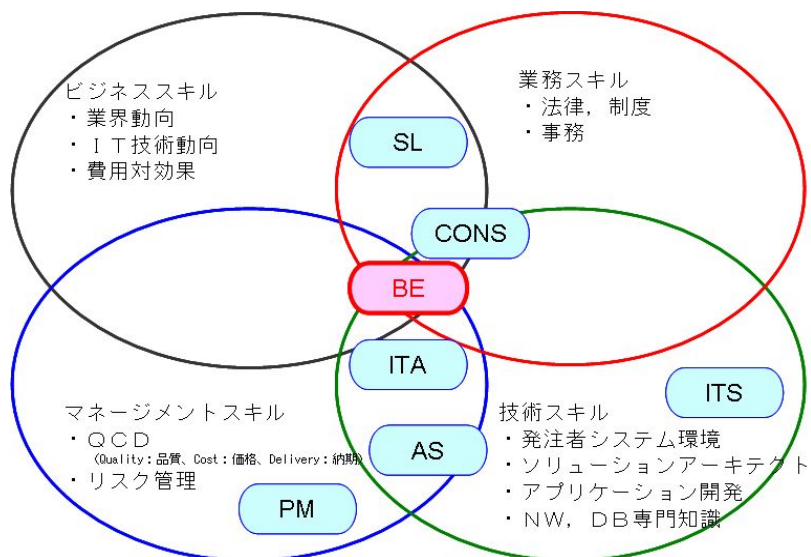


図 7. 5 ビジネスエンジニアに求められるスキル

7. 6. 3. 担当領域

ビジネスエンジニアは「ビジネス戦略策定」、「課題整理/分析」、「ソリューション設計」、「コンポーネント設計」、「ソリューション構築」において、要件定義に対する相違箇所が判明した際は、受注者内部への指摘を行うとともに、発注者側システム部門のみならずユーザ部門に対しても説

明、交渉を担当する。

7. 6. 4. 位置付け

役割による組織に対する貢献は以下の通りである。

①統括

自社の事業全体に対して責任を持つ

②営業

自社の売上、収益に対して責任を持つ

③PM

自社のQCDに対して責任を持つ

④ビジネスエンジニア

発注者要求の実現に対して責任を持つ

つまり、ビジネスエンジニアは顧客からの信頼度を向上させることにより、自社商域の拡大を目指す役割を担う。

これより、ビジネスエンジニアは開発チームと分離独立した立場に位置することで、要求の実現に対して中立性を保つことが可能となる。

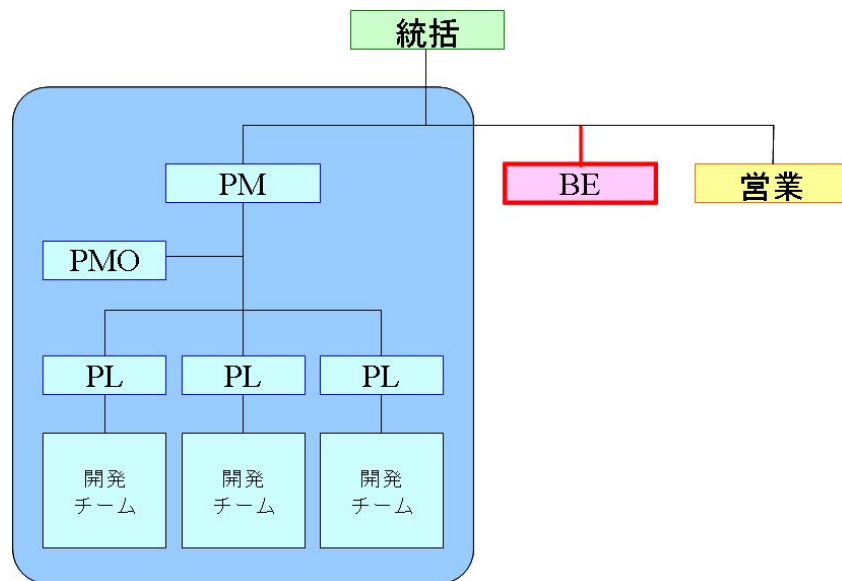


図 7. 6 ビジネスエンジニアの位置付け

7. 7. 総括

システム構築における満足度の向上により自社商域の拡大を目指す上で、ビジネスエンジニアの役割が重要となりつつある。

ビジネスエンジニアがプロジェクトに参画するメリットとしては、以下の事項が挙げられる。

①発注者

- ・システム部門におけるマンパワー不足
- ・スキル不足(2007年問題)の解消

- ・システム構築投資の適正化

②受注者

- ・更に顧客満足度の高いシステムの構築
- ・新たなビジネスモデルの組成

また、ビジネスエンジニアに対する課題としては、以下の事項が挙げられる。

①発注者

- ・業務スキルジャッジの適正化
- ・妥当なフィーの認定

②受注者

- ・要員の育成
- ・社内処遇の適正化

システム構築の早い段階で、発注者自身が自ら定めた要件の検証が実現できていれば、満足度の高いシステム構築へ繋げていけるはずであるが、残念ながら、要件の検証が出来ていないのが現実である。

受注者側においても発注者の要件がシステムに適切に盛り込まれているか、要件の反映度を検証していくことも不可欠であろう。

ビジネスエンジニアの存在が普及および浸透するためには、要件定義並びにシステム構築に対する責任の所在を、発注者、受注者それぞれの立場で明らかにしていく必要がある。その役割を担うビジネスエンジニアの意義については、今後も議論を深めていく価値があるものと考ええる。

7. 8. 最後に

これまでの情報システム開発者は、決められた仕様に沿ってシステム構築していただくに終始することが多かったように思われる。これからは自らの卓越した技術スキルとユーザの視点に立った戦略提案能力を発揮できる新たなエンジニア像を描き出す必要があるのではないだろうか。

8. 要求工学研究の世界的動向調査

日本アイ・ビー・エム株式会社 東京基礎研究所 鎌田(板倉)真由美

この調査報告は、2005年12月から2006年3月にかけて実施した要求工学の世界的な動向に関するものである。ソフトウェア開発において最も曖昧で困難といわれる要求定義を研究する要求工学について、広く現状を概観し将来の方向性を探った。当調査は独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センターによって行われた。また、本稿は情報処理学会ソフトウェア工学研究会にて発表した論文に手を加えた。

8. 1. はじめに

要求工学は、この数年最も関心が高まっているソフトウェア工学の一分野である。ソフトウェア開発の現場では、通常開発プロセスの初期段階に要求定義作業が実施されている。そこではエンドユーザやビジネスオーナーから主に非形式的な表現の要求を獲得することに始まり、それら要求に対してシステム開発者らが中心となって、分析・モデル化・システム化可能性の精査・優先度付け・要求の仕様化などの作業を行い、次の設計プロセスの入力資料となる要求仕様書を作成している。要求工学では研究の対象が、多様かつ広範囲にわたり、適用する技術もビジネス分析や対人コミュニケーションといった人間系からユースケースやデータベース論理仕様といったシステム設計に非常に近いものまで、広範囲かつ多種類に及んでいる。

ソフトウェア開発の各プロセスにはそれぞれの困難さがあるが、多様な入力からシステム開発の基本となる要求仕様書を作成する要求定義プロセスは、困難の度合いが最も大きいと言える。近年注目される理由のひとつは、その困難さゆえに研究者にとっては知的好奇心とやりがいを感じる分野であることと、実務者にとっては無視できない根源的で重要な課題であると認識されてきたためである。

当調査は、図 8. 1に示す手順で識者へのインタビューと文献調査を柱とし、要求工学の現在の全体像を明らかにし、さらに今後の進むべき方向について考察した。

Step1 事前調査: 日本国内の要求工学研究の第一人者を中心に事前調査を行い、調査対象研究の絞り込みを実施した。

Step2-A 文献調査: 主要国際学会を中心とした文献調査を通じて、要求工学の研究動向を探った。

Step2-B ヒアリング調査: 文献調査だけでは得られない知見を得る目的で実施した。対象としたのは研究者と実務者双方である。要求工学研究者に対しては、対象研究エリアの成熟度と適用度、対象研究の将来的なビジョンなどをヒアリングし、企業内の要求工学実務者へは、実際に要求工学にどのように取り組んでおり、今後何を期待するかを中心に調査した。

Step3 文献調査およびヒアリング調査の分析: 文献調査とヒアリング調査の結果と知見を基に、要求工学の現状について整理した。

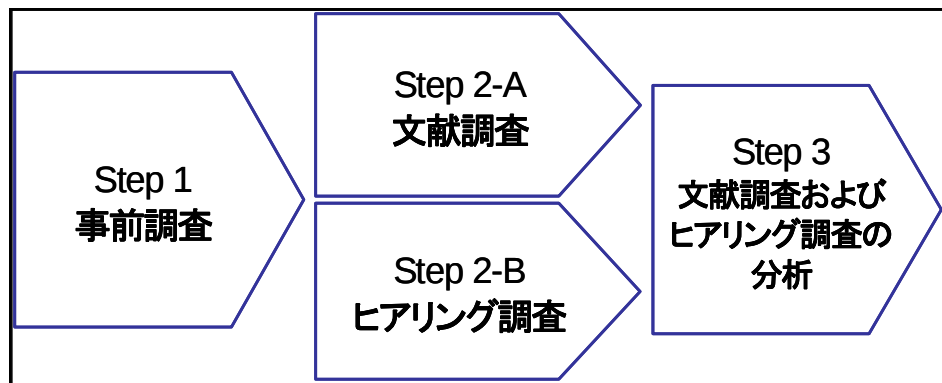


図 8. 1 調査手順

また、要求工学の全体像を把握するために、既存研究からいくつかの定義やフレームワークを参照した。まず要求工学の定義については、大西らの定義「ソフトウェア要求をいかにしてまとめるかといった技術や技法の集大成」および、Zape[2]の定義「広義に捉えればシステム工学の一分野とも言える」を前提とした。

また、要求工学全体を可視化する目的で、2004 年の Working Conference on Requirements Engineering: Foundation for Software Quality (以降 REFSQ) [3]の 10th Anniversary Booklet より、Rengnell らによるワークショップで使用された Framework for making the context of each talk explicit[4]を参照して要求工学メタモデルを新たに作成した (図 8. 2)。

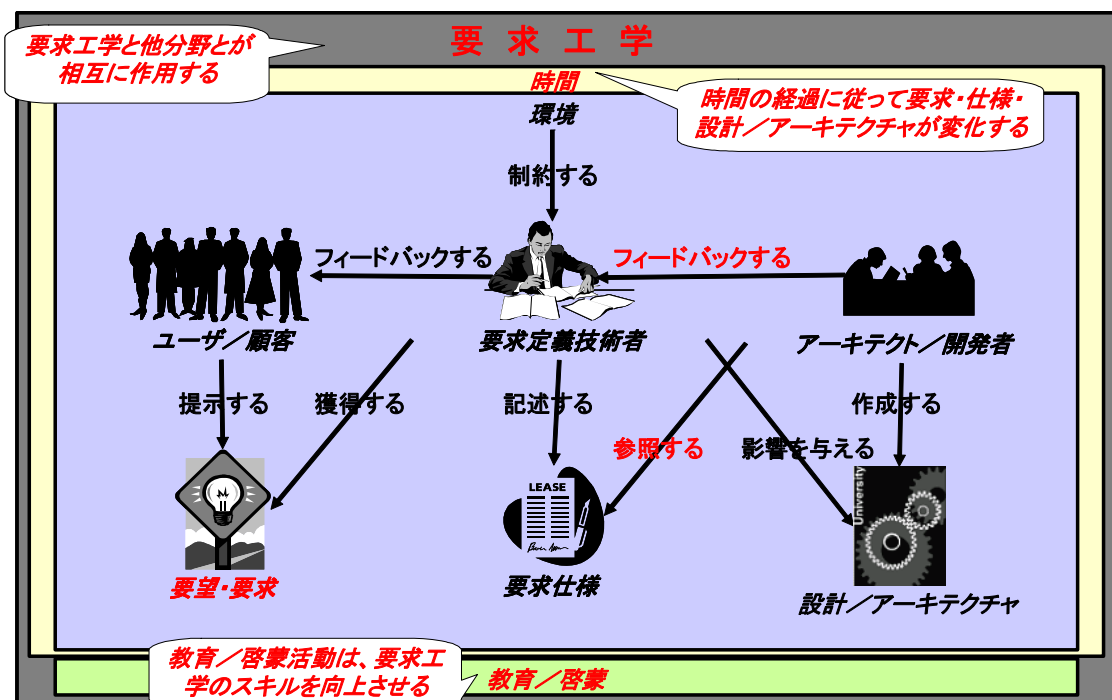


図 8. 2 要求工学メタモデル

また、要求定義プロセスについては既存研究でさまざまな提案がされているが、国際標準や規格などは存在しない。よく知られているプロセスモデルには、Thayer[5], Sommerville[6], Robertson[7]などがあるが、当調査では日本の実態に即したプロセスを重視し、情報処理学会ソフトウェア工学研究会要求工学ワーキンググループが提案した要求工学プロセス[図 8. 3]を基にした。

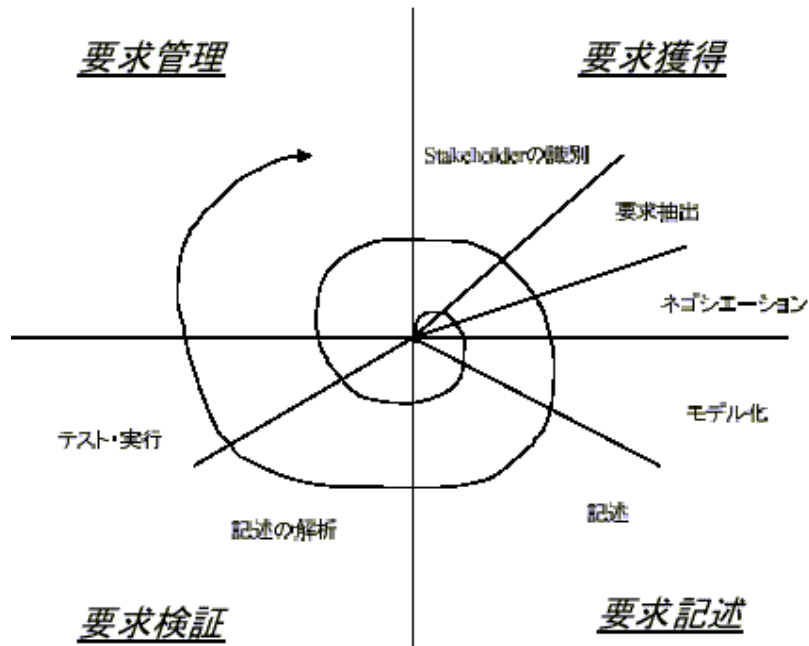


図 8. 3 要求定義プロセス [8]より引用

8. 2. 事前調査

事前調査では、日本における要求工学研究の第一人者の方々から、文献調査およびヒアリング調査の対象の抽出、また調査内容および要求工学全体の枠組みの策定への助言を得た。事前調査は対面式で行い、①現在の要求工学の状況、②要求工学の枠組み、③要求工学が解決しようとしていること・取り組むべき方向、④これから注目すべき要求工学の分野、についてインタビューを行った。

表 8. 1 事前調査対象者

研究者	主な貢献	研究テーマ	所属組織	インタビュー日
青山 幹雄	RE'05 Paper Accept RE'04 General Co-Chair	組込みソフトウェア ソフトウェア工学	南山大学	2005/12/13 2005/12/21
玉井 哲雄	RE'01-'06 Program Committee 「ソフトウェア要求と仕様」(マイケル・ジャクソン著) 訳者	オブジェクト指向 モデル化手法 ソフトウェア工学	東京大学	2005/12/12
佐伯 元司	RE'04 General Co-Chair RE'02 Paper Accept	ゴール指向 要求工学	東京工業大学	2005/12/26
大西 淳	RE'05 Paper Accept, RE'04 Local Arrangement Chair & Program Committee 情報処理学会要求工学ワーキンググループ主査 (1998-2004)	シナリオベース 要求工学	立命館大学	2005/12/26
海谷 治彦	RE Journal (Springer) 2005 Accept RE'04 Paper Accept, RE'02 Paper Accept 情報処理学会要求工学ワーキンググループ主査 (2004-2005)	ゴール指向 組込みソフトウェア 要求工学	信州大学	2005/12/26
中谷 多哉子	情報処理学会要求工学ワーキンググループ幹事 (1997-2005)	オブジェクト指向 要求工学 ソフトウェア工学	筑波大学	2005/12/29

8. 2. 1. 要求工学の研究分野

事前調査の結果、一口に要求工学といっても研究分野は細分化されており、要求工学の範囲についても認識が統一されていないのが現状であることを確認し、まず要求工学の全体像を把握するために、以下の手順で要求工学の範疇にある研究エリアを抽出・整理した。

- 1) 要求工学の主要国際学会である RE および REFSQ の 2001-2006 年の Call for Papers テーマ(論文募集要項に載っているテーマ)より、頻出語(単語・熟語)を抽出整理する。
- 2) 要求工学における手法／テクニックとして、[9]にて分類されている研究分野名を抽出整理する。
- 3) 上記 1), 2) の結果を元に、主要研究エリアを選定する。

この分類の結果、表 2 で挙げた 18 の研究エリアが要求工学に含まれるとした。

表 8. 2. 要求工学の主要研究エリア(順不同)

No.	研究エリア	No.	研究エリア
1	ステークホルダ分析	10	非機能要求モデリング
2	ゴール指向	11	無矛盾性チェック ¹
3	シナリオベース	12	要求とアーキテクチャ
4	アспект指向	13	要求の社会的／文化的側面
5	問題フレーム	14	要求管理
6	ネゴシエーション・モデル	15	追跡可能性 ²
7	要求パターン	16	ドメイン特化研究
8	ドメイン・モデリング	17	学際的研究
9	形式仕様	18	要求工学教育

これら 18 の要求工学研究エリアを図 8. 2 のメタモデル上に配置し、可視化した(図 8. 4)。

¹ 一貫性(Consistency)チェックともいう

² トレーサビリティ(Traceability)ともいう

18 の研究エリアを要求工学メタモデル上に配置することにより、現在の要求工学研究でどの分野に研究エリアが集中しているのか、また、あまり手をつけられていない箇所はどこなのかを明らかにすることができる。ほとんどの研究エリアで、複数の人と複数のもの(ドキュメント)を扱っているが、当調査では現状把握を目的とするために、もっとも焦点をあてているエリアを選択した。

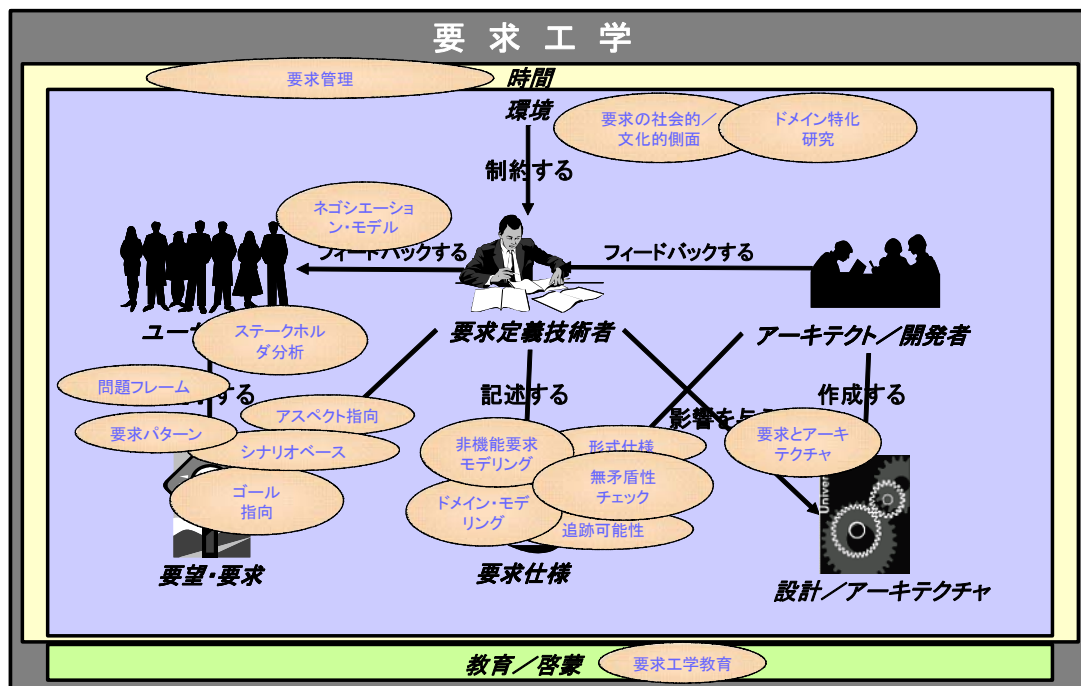


図 8. 4 要求工学研究エリア

8. 2. 2. 要求工学分類の枠組み

次に、要求工学の全体像を把握するために 4 つの視点による要求工学分類の枠組みを設定した。これら 4 つの視点は、いずれも当調査の目的を重視して事前調査にて設定したものである。いずれの視点も表形式で表しており、横軸は 4 つの要求定義プロセス(図 8. 3)で共通であるが、縦軸は以下に挙げる 4 つの視点である。

- 1) 表 8. 3. **ステークホルダの視点**: 研究エリアが対象としているステークホルダを把握することで、要求定義現場におけるベンダとユーザの協業促進の可能性を把握する。ステークホルダの種類は「SEC BOOKS 経営者が参画する要求品質の確保 ～超上流から攻める IT 化の勘どころ～」[10]の図 4.1 に一部の役割を追加した。
- 2) 表 8. 4. **要求定義品質の視点**: 研究エリアが着目している要求定義における品質特性およびその理由を知ることで、日本のモノ作りに欠かせない高品質な成果への貢献可能性を把握する。要求定義の品質特性は[11]の Software Requirements Specifications(SRS)の品質特性を代用特性とし、日本語訳は[1]から引用した。
- 3) 表 8. 5. **ソフトウェア品質の視点**: 研究エリアが要求定義の結果作成されるソフトウェア自身にも、視野を広げて品質を考慮しているかを知ることで、日本のモノ作りに欠かせない高品質な成果への貢献可能性を把握する。ソフトウェアの品質特性は、[12]の Quality

characteristics and guidelines for their use より6つの品質特性を引用し、日本語訳はJIS X0129[13]を使用した。

- 4) 図 8. 5. 成熟度の視点:研究エリアの先進性・成熟度・現場への適用可能度を知ることにより、日本の現場に現在適用可能な状況か、将来適用可能性があるかを把握する。4 象限の表で表し、各研究エリアの位置をプロットする。

表 8. 3 要求工学分析の枠組み – ステークホルダ・ビュー –

ステークホルダ*(役割) / 要求定義プロセス		要求獲得	要求記述	要求検証	要求管理
顧客	社長／担当役員				
	業務部門				
	情報システム部門				
	システム子会社				
エンドユーザ	内部ユーザ				
	オペレータ				
	外部ユーザ** (匿名性の高いユーザを含む)				
システム提供者	プロジェクトマネージャ				
	コンサルタント／アナリスト				
	アーキテクト				
	開発者				

* ステークホルダは、SEC BOOKS 経営者が参画する要求品質の確保 ～超上流から攻めるIT化の勘どころ～ p.44 図 4.1 要件の定義と役割を基に、一部の役割を追加。

** 外部ユーザとは主に、要求獲得を直接行えないインターネットを介してシステムを利用するユーザや組込みシステムのユーザなど。

表 8. 4 要求工学分析の枠組み — 要求定義の品質ビュー —

要求定義の品質特性(SRS)* / 要求定義プロセス	要求獲得	要求記述	要求検証	要求管理
妥当性				
非あいまい性				
完全性				
無矛盾性				
重要度と安定性のランク付け				
検証可能性				
変更可能性				
追跡可能性				
(その他:)				

* ISO/IEC9126(JIS X 0129) as characteristics of SW products. Information technology software product evaluation :Quality characteristics and guidelines for their use より 6 つの品質特性を引用。日本語訳は JIS X0129 を使用した。

表 8. 5 要求工学分析の枠組み — 最終成果物ソフトウェアの品質ビュー —

ソフトウェア・プロダクトの品質特性* / 要求工学プロセス	要求獲得	要求記述	要求検証	要求管理
機能性				
信頼性				
使用性				
効率性				
保守性				
移植性				
(その他:)				

* ISO/IEC9126(JIS X 0129) as characteristics of SW products. Information technology software product evaluation :Quality characteristics and guidelines for their use より 6 つの品質特性を引用。日本語訳は JIS X0129 を使用した。

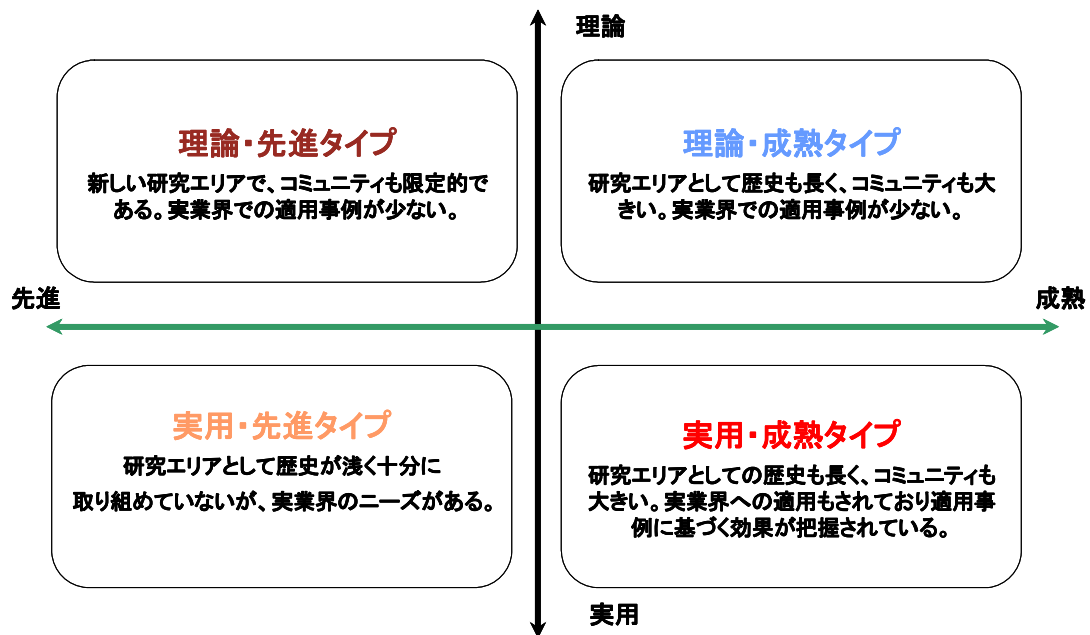


図 8. 5 研究の適用状況と先進性ビュー

8. 2. 3. ヒアリング対象項目

STEP2-B のヒアリング調査では、研究者の方々に対して共通質問を設けてインタビューしている。共通質問は、事前調査の結果を参考に作成したものであり、その領域の研究者としての見解や今後の自分の研究の方向性を確認すると同時に、要求工学全体像についても質問している。一方、実務者の方々には主に現場で要求工学がどう使われているか、どのような意義を感じているのか、また今後要求工学はどうあるべきか、などを質問した。以下に質問事項を挙げる。

研究者を対象とした共通質問(大項目のみ)

- Q1. 研究の主たる目的および、その研究によって解決しようとしている課題(ステークホルダ、要求定義品質、ソフトウェア品質の視点から)
- Q2. 研究の前提条件・制約条件
- Q3. 研究で、顧客やプロジェクトオーナーに要求や期待する結果についての確認を行うことを考慮しているか?どんな方法で行うのか
- Q4. 現在の研究の成熟度はどの位置か(成熟度の視点で)
- Q5. 将来の研究の成熟度はどの位置か(成熟度の視点で)その実現方法は?

要求工学全般についての共通質問(大項目のみ)

- Q6. 現在の要求工学における課題とは何か?その理由は?
- Q7. 今後 3 年間で注目されそうな要求工学の研究エリアは何か?その理由は?
- Q8. 学際的研究への意見および自分の研究への取り込み状況・予定

その他の質問項目

- Q9. 要求工学分析の枠組み 4 種についてのコメント
- Q10. SEC の今後の研究テーマへの提言

8. 3. 文献調査

STEP2-A の文献調査の対象は、表 6 の 4 学会および 1 論文誌に 2000 年から 2005 年の 6 年間に発表された論文である。国際会議についてはワークショップに投稿された論文も含むが、6 年間で計 717 論文が調査対象となった。

表 8. 6 文献調査対象一覧

No.	国際会議 / 論文誌	略称	説明・選択理由
1	IEEE International Requirements Engineering Conference	RE	IEEE主催で行われる、要求工学分野で最もメジャーかつ権威のある国際会議。年1回開催。
2	Working Conference on Requirements Engineering: Foundation for Software Quality	REFSQ	ヨーロッパ中心で行われている、要求工学分野専門の国際会議。年1回開催。
3	Requirements Engineering Journal	REJ	要求工学専門の論文誌。Springer-Verlag London Ltd.により年4回発行。
4	International Conference on Software Engineering	ICSE	ソフトウェアエンジニアリングにおける最も権威のある会議。競争率は5～8倍。年1回開催。要求工学分野の論文のみを抽出。
5	Conference on Advanced Information Systems Engineering	CAISE	システムエンジニアリングにおける会議で、要求工学に関する論文が多い。年1回開催。要求工学分野の論文のみを抽出。

8. 3. 1. 文献調査の準備

まず 717 論文を、2000-2005 年に発表された論文であるグループ A と 2005 年に RE と REFSQ に発表された論文であるグループ B に分類した。グループ A については、他論文に引用された数を Google Scholar[14]を用いて集計し、引用数の多い上位 51 論文を影響力がある論文として抽出した。グループ B は、グループ A の抽出方法では下位になりがちな新しい論文の傾向を読み取る目的で、そのまま 53 論文を分析対象とした。

これら 104 論文を、事前調査で設定した 18 の研究エリアに分類し、メタモデルや枠組みに配置していくことで、要求工学の研究エリアの活性度や影響度を明らかにした。しかし、グループ A のように他論文から多く引用されている論文を選択する場合には、どのような論文が引用されやすいかを理解し、限界を理解して使う必要がある。一般に概説的な論文や、基礎的な領域を研究している論文は引用されやすく、事例などは引用されにくい。こうした欠点を補うために当調査では、2005 年の最新論文については、対象を要求工学に特化した国際会議に採択された論文についてはすべてを分析対象とした。

8. 3. 2. 分析

文献については以下のように事前調査で設定した枠組みに沿った分析を行った。

1. 要求工学メタモデル上への配置：どこの問題を解決しようとしているのかがわかる（図 8. 6 および図 8. 7）
2. 要求工学プロセスとの対応付け：重視されているプロセスがわかる
3. 実プロジェクトへの適用状況と適用ドメインの把握：インタビュー対象研究エリアの成熟度合い、実用度合いの裏づけを取る

グループ A から抽出された 51 の論文を要求工学メタモデル上に配置すると、18 研究エリアを配置した際に集中した箇所である”要求の獲得”と、”要求仕様書関連”に、多くの論文が集まっていた。グループ B の 53 論文を要求工学メタモデル上に配置すると、グループ A では見ら

れないネゴシエーション・モデルやエンドユーザとの係わり合い、また固有ドメインに関連する研究が新たに出てきている点が時系列の変化である。

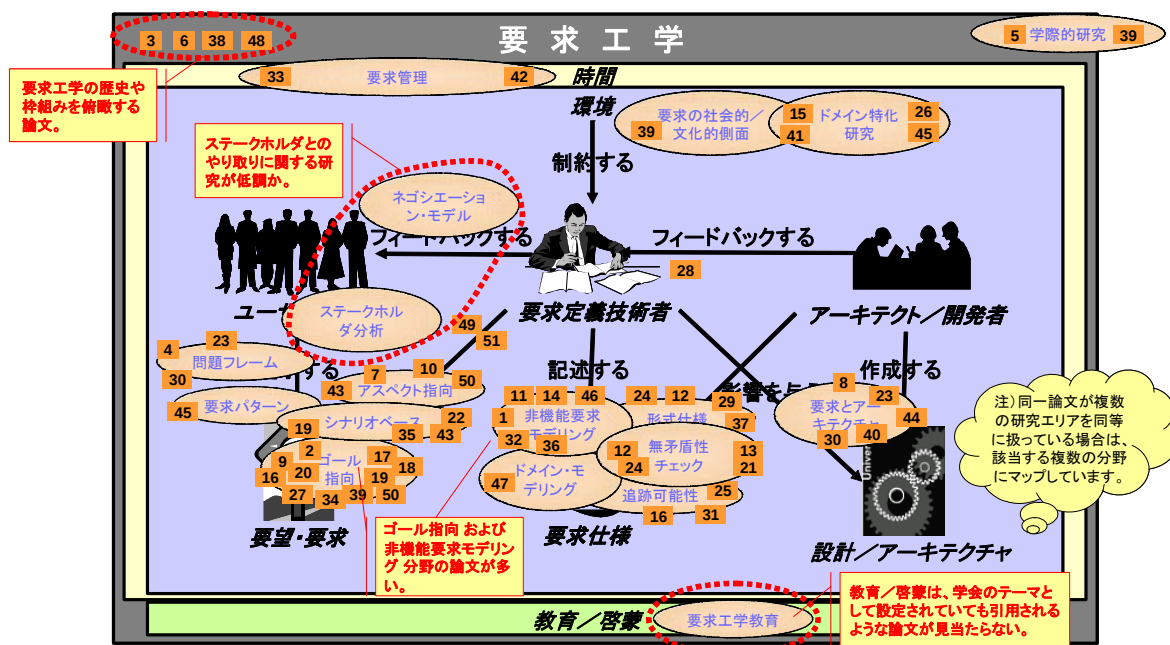


図 8. 6 要求工学メタモデル上に配置したグループ A(2000-2005 の引用数の多い論文)

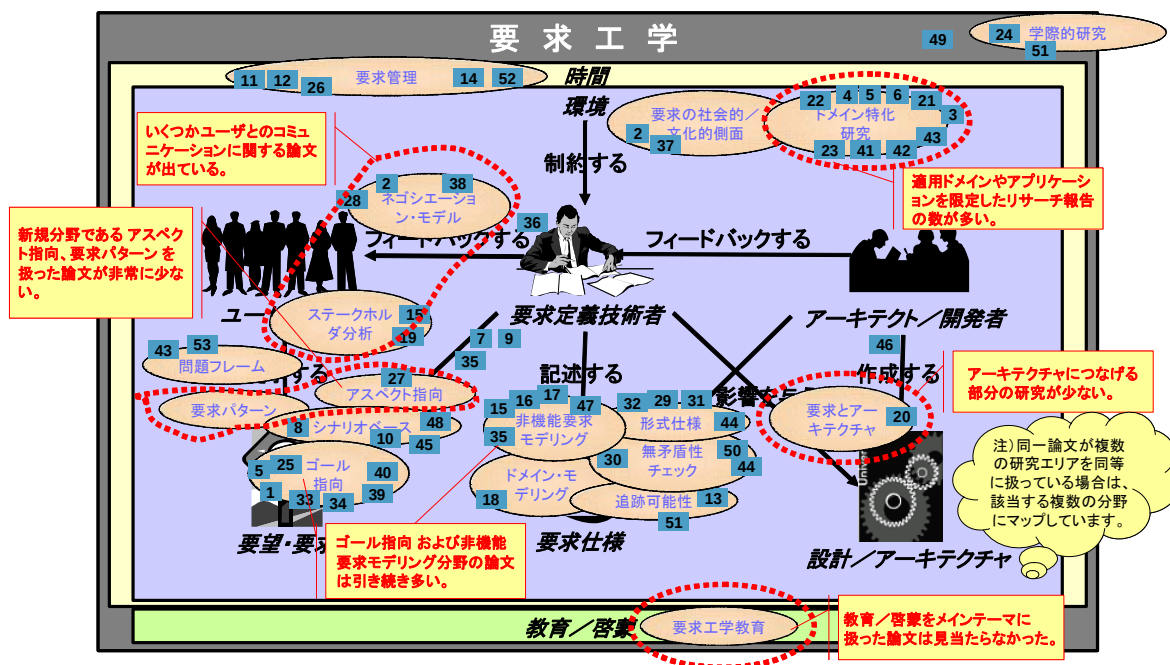


図 8. 7 要求工学メタモデル上に配置したグループ B

8. 3. 3. 文献調査のファインディング

これら3種類の文献調査を行った結果、以下の13項目がファインディングとして観察された。

- 1.. 要求獲得、特に「要求の抽出」に研究が集中している傾向がある。
2. ステークホルダ分析やネゴシエーションといった、ユーザ／顧客とのコミュニケーションに関する研究は限定的である。
3. ゴール指向とシナリオベースは、論文数も実ケース数も多く、他分野に比べて成熟している傾向がある。
4. 要求記述に関する研究も、研究が集中しているプロセスである。
5. 要求記述の形式化／半形式化が進んでいる傾向がある。
6. 形式仕様はツールのサポートが前提となっており、自動化の試み（特に検証）が進められている。ただし、実プロジェクトへの適用報告がほとんどない。
7. 非機能要求の中でも、セキュリティに研究が集中している。実プロジェクトへの適用報告も多く、実践的な研究が多い。
8. 要求管理に関する研究は、現時点では限定的。ただし、直近では実践的な研究が出てきている。
9. ドメイン特化研究が盛んになっている傾向があり、より特定のエリアに特化した実践的な研究が進む可能性がある。
10. 要求工学教育エリアをメインに扱った研究論文はほとんどないが、ここ数年ワークショップのテーマとして扱われており関心が高まっている。
11. 要求工学全体を俯瞰する研究への潜在的な需要がある。
12. 要求からアーキテクチャにつなぐ研究発表は一定の数が出ているが波があり、'05年は低調だった。
13. 新興研究エリア（アスペクト指向、要求パターン）は、調査範囲内から見る限り、研究コミュニティが限定的な可能性がある。

8. 4. ヒアリング調査

ヒアリング調査の対象者は、事前調査により候補者として挙げた方々のうち、現時点での研究状況なども加味して依頼し、承諾を得た方々である。研究者13名(表8. 7)および実践者7名(表8. 8)に対して、事前調査で設定した要求工学の枠組み図や説明文書をあらかじめ送付して、対面式のインタビューにより調査した。

表 8. 7 ヒアリング対象研究者

No	研究者	研究テーマ	該当研究エリア	所属	インタビュー日
1 2	Eric Yu Steve Easterbrook	Goal Oriented Approach (*) Non Functional Requirements (NFR)	ゴール指向	University of Toronto	2006/01/23
3	Matts Heimdahl	Formal Model-Based Development	形式仕様	University of Minnesota	2006/01/27
4 5	Lars Hagge Kathrin Lappe	RE Pattern	要求パターン	DESY (Deutsches Elektronen-Synchrotron)	2006/02/06
6	Neil Maiden	Scenario-based approach Discovering Requirements	シナリオベース	City University London	2006/02/08
7 8	Bashar Nuseibeh Michael Jackson	Problem Frame Aframes	問題フレーム	The Open University	2006/02/10
9 10 11	Joao Araujo Ana Moreira Pedro Guerreiro	Aspect Oriented Requirements Engineering	アスペクト指向	Universidade Nova de Lisboa	2006/02/13
12	三瀬 敏朗	Embedded Software Exception Analysis	ドメイン・モデリング	松下電工株式会社 品質部	2006/02/24
13	Colette Rolland	Map-driven Modular Method Requirements Evolution	ゴール指向 + シナリオベース	Universite Paris-Sorbonne	2006/03/05

表 8. 8 ヒアリング対象実践者

No	名前	関心対象領域	所属	インタビュー日
1	Theresa Kratschmer	要求管理メソドロジー	Researcher, IBM (Watson Research, Hawthorne)	2006/01/25
2	Jamshid Vayghan	要求工学とデータ、およびアーキテクチャ	Architect, IBM (Rochester)	2006/01/26
3	Chris Winter	要求工学全般	Distinguished Engineer, IBM (Hursley)	2006/02/10
4	上原 三八	エンタープライズ・システム	富士通研究所	2006/02/03
5 6	Lars Hagge* Kathrin Lappe*	REパターン	DESY (Deutsches Elektronen-Synchrotron)	2006/02/06
7	三瀬 敏朗*	ドメイン・モデリング	松下電工株式会社 品質部	2006/02/24

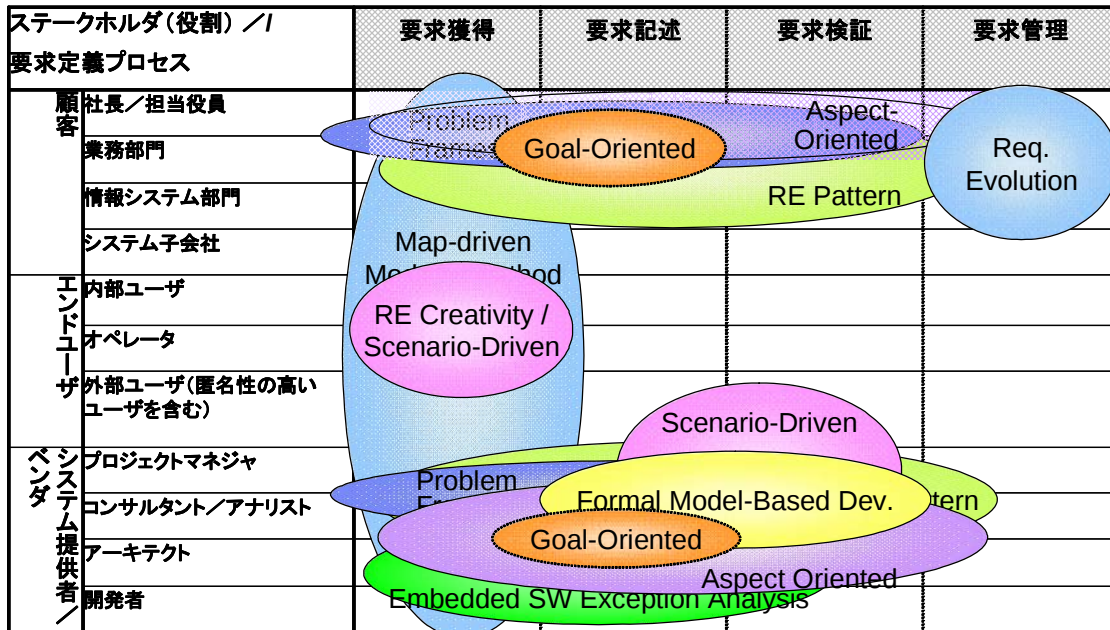
* は、表 7 の研究者との重複あり。

8. 4. 1. 研究者へのヒアリング調査のファインディング

13 人の研究者へのヒアリング調査結果に基づいて、4 つの要求工学の枠組み(表 8. 3、8. 4、8. 5 および図 8. 5)を更新した。以下にファインディングとともに更新された表を示す。

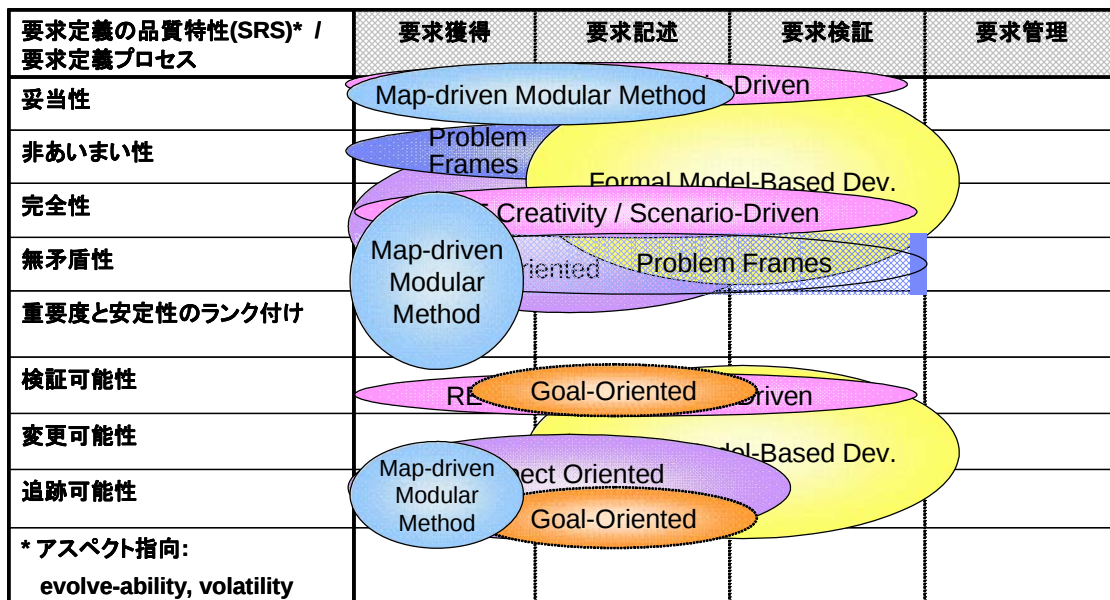
- ステークホルダとプロセスの枠組みについては、研究対象の中心は、要求記述と要求検証であり、要求獲得が次いで対象とされていた。研究対象ステークホルダはシステム提供者／ベンダのコンサルタント／アナリスト、アーキテクトが多く、次いで業務部門であった。(表 8. 9)

表 8.9 研究で焦点をあてているステークホルダと該当プロセス



- 要求定義の品質とプロセスの枠組みについては、それぞれの研究テーマにより要求定義の品質で着目しているところは異なり、特定品質への集中は見られなかった。しかし、重要度と安定性のランク付けという品質特性についてはほぼ全ての研究エリアで対象外とされていた。(表 8. 10)

表 8. 10 研究で充実させようとしている要求定義の品質特性と、該当プロセス



- 要求定義の最終成果であるソフトウェア品質特性とプロセスの枠組みについては、最終成果物の特定の品質を意識している研究自体が限定的であった。機能性、使用性、保守性に着目した研究は少数存在したが、最終成果のソフトウェアの品質特性についてはほとんど

ど着目されていないといえる。(表 8. 11)

表 8. 11 研究で視野に入れている最終成果物のソフトウェア品質特性と該当プロセス

ソフトウェア・プロダクトの品質特性／要求工学プロセス	要求獲得	要求記述	要求検証	要求管理
機能性	Map-driven Modular Method			Req. Evolution
信頼性	Aspect Oriented			
使用性	Map-driven Modular Method			Req. Evolution
効率性	Scenario-Driven	Goal-Oriented		
保守性	Aspect Oriented			
移植性				
* ゴール指向: プライバシー * 問題フレーム: セキュリティ * Map-driven Modular Method: variability				

- 成熟度の枠組みについては、各研究者は明確な指針を持っていなかったため、インタビュー対象者の自己申告では、妥当な配置ができなかった。

8. 4. 2. 実践者へのヒアリング調査のファインディング

実践者へは定型の質問項目を事前に送っておらず、実務に関しての会話が中心であった。そのため、調査結果の 13 件もファインディングのみを対象とした。

- 要求工学自体が目的ではなく、システム構築や IT 環境の中に位置づけて、より広い枠組みの中で考えるべき。
- ビジネスレベルから要求を理解することが重要。
- 要求管理はプロジェクトマネジメントの一環として捉えている。
- 要求獲得ではユーザとのコミュニケーションが重要である。
- ユーザの要求を引き出し共に作り上げること、およびユーザの背景やシステム状況を把握し可視化することが、システムの品質向上につながる
- 要求は検証可能でテスト可能に定義され、書かれていなくてはならない。(組込み以外でも)仕様化・形式化の技術を適用することが重要になる。
- 既存システムを伴う開発と新規開発では要求定義のアプローチも変わる
- 要求獲得は、インタビューやオープン質問などのコミュニケーション主体のアクティビティとして行われる。また、ドメイン知識が重要である。
- 実務で再利用する目的で、パターン化およびリポジトリ利用を進めている。
- 実践的なツールが不足している。現場では自ら必要とするツールを作成して使っている例も多い。
- 価値(Value)に着目し、顧客満足を高めるソフトウェア開発に取り組んでもらいたい。

8. 5. 文献調査およびヒアリング調査分析の結果

ここまで述べてきた文献調査とヒアリング調査の結果を統合し、共通するファインディングを抽出し、特徴的な項目についての知見を整理する。文献調査とヒアリング調査双方から図 8. 8 にて裏づけされたファインディングは 8 点であった(図 8. 9)。

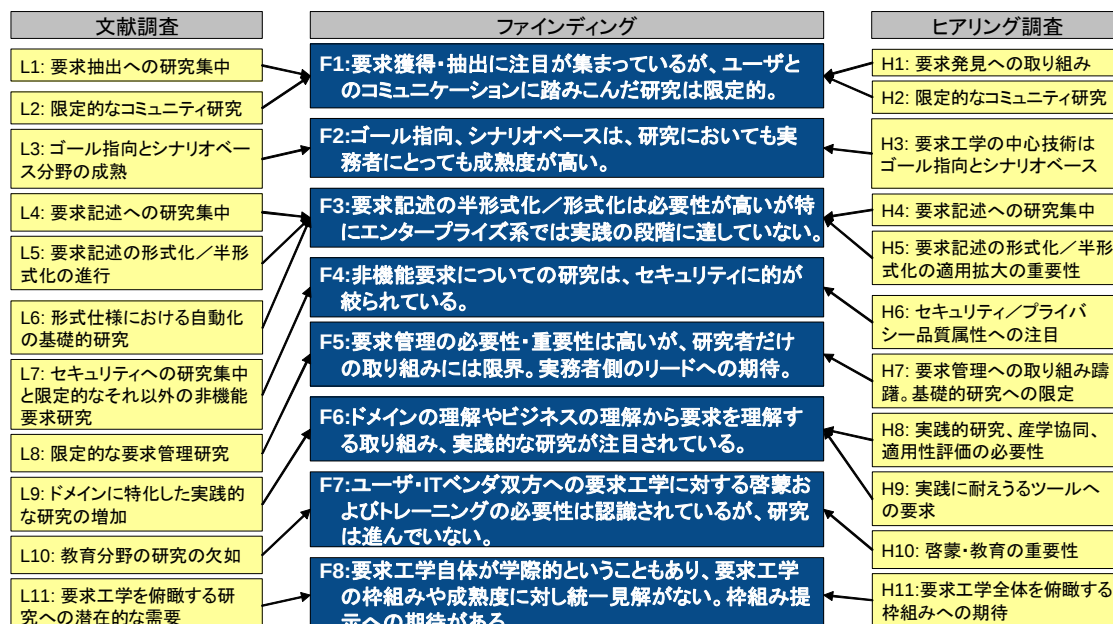


図 8. 8 文献調査とヒアリング調査結果の整理

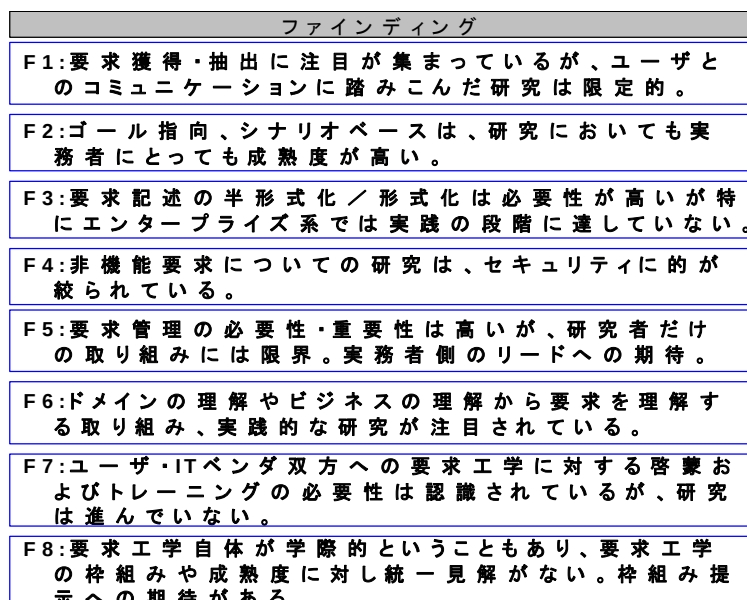


図 8. 9 ファインディング 8 項目 (図 8. 8 から抜粋)

文献調査とヒアリング調査の結果得られた研究エリアの成熟度や研究コミュニティの大きさを図 8. 5 の枠組みに配置すると、図 8. 10 のようになった。この図の○の大きさは、文献調査の結果得られ

た研究エリアの大きさ(コミュニティとしての広がりや研究者の層の厚さ)を加味したものである。また、横軸の先進・成熟の判断指標は、該当研究エリアの研究がどれくらいの期間(活動している実期間)なされているかを考慮して配置した。活動期間は具体的には、その研究エリアが発案・作成されてからの年数を基準のひとつにした。

- ・0・5 年以内 (先進)
- ・5・10 年以内 (やや先進)
- ・10 年以上 (やや成熟)
- ・15 年以上 (成熟)

図 8. 10 をその成熟度と実用度の視点で眺めると、たとえばシナリオベースの研究エリアは、成熟度も高く、実際に適用した事例もある研究といえる。このような研究エリアは実際のプロジェクトに応用するのにリスクが低いと考えて良い。また、ネゴシエーション・モデルは、まだ研究が開始して十分に時間が経っていないが、実践的な事例などが出てきていると読める。図 8. 10 を活用することで、要求工学の各研究エリアが今現在どのような状況にあり、実践に活用可能かどうかがわかるのである。

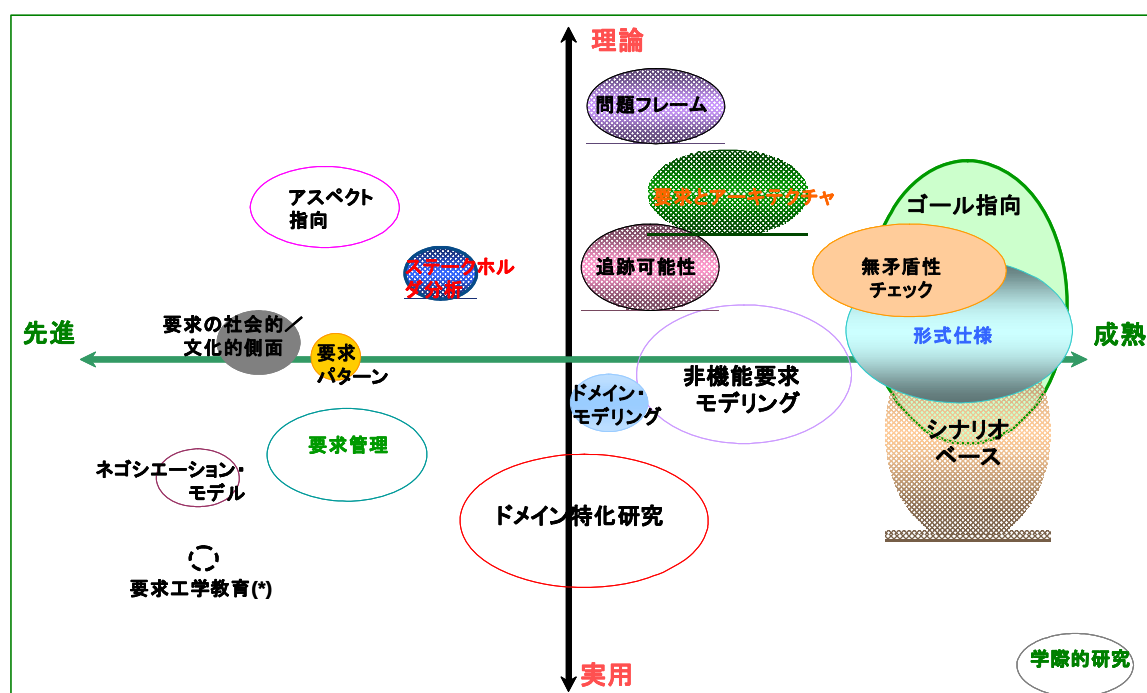


図 8. 10 要求工学研究エリアの成熟度

8. 5. 1. 要求工学分析の枠組みへの提言

その他特筆すべきものとして、事前調査で設定した 4 つの枠組み(表 8. 3、8. 4、8. 5 および図 8. 5)について、聞き取り調査の中でいくつかのコメントが出された。表 8. 4 および表 8. 5 に関するコメントであったが、それらをまとめると以下の 3 点に集約される内容であった。

1. 顧客の満足度を品質の指標に加える(表 8. 3 および 8. 4 について)
2. 変化する要求に対応する指標が必要(表 8. 3 および 8. 4 について)
3. セキュリティは独立した品質として必要(表 8. 4 について)

ISO や IEEE の品質指標では不足する、新たな品質の概念が必要とされているという興味深いコメントである。IT システムに対する社会的な信頼性やセキュリティの期待や、産業界からの変化への即応性への強い要請などが、要求工学にも影響を及ぼしていると考えられる。

8. 6. まとめ

本調査報告では、短期間に文献調査と第一人者および実践者への聞き取り調査を行い、現在の要求工学を概観する枠組みを提示した。また、共通の要求工学の枠組みを提示して研究者と実践者双方から聞き取り調査を行ったことにより、研究者が焦点を当てているエリアと、実践者が期待するエリアの差異も明らかになった。こうした両者のギャップを今後どう埋めていくのかといったことも重要な課題である。この調査結果を、最終的な目的であるユーザと IT ベンダ双方にとって有益な指針へとつなげていくことが必要であろう。

今後も、このような動向調査を定点観測的に行っていくことは日本の IT 産業にもユーザにとっても意義のあることである。

謝辞

事前調査にご協力をいただいた青山幹雄先生(南山大学)、玉井哲雄先生(東京大学)、佐伯元司先生(東京工業大学)、大西淳先生(立命館大学)、海谷治彦先生(信州大学)、中谷多哉子先生(筑波大学)に、厚く御礼を申しあげたい。

また、ヒアリング調査に快く応じてくださった以下の方々(敬称略): 上原三八(富士通研究所)、三瀬敏朗(松下電工株式会社)、Eric Yu, Steve Easterbrook, Mats Heimdahl, Lars Hagge, Kathrin Lappe, Neil Maiden, Bashar Nuseibeh, Michael Jackson, Joao Araujo, Ana Moreira, Pedro Guerreiro, Colette Roland, Theresa Kratschmer, Jamshid Vaygahn, Chris Winter の各氏のご協力に心からの感謝と敬意を表する。

参考文献

- [1]大西淳. 郷健太郎., 要求工学, 共立出版, p2, p.16, 2002
- [2]P.Zave., Classification of Research Efforts in Requirements Engineering, ACM Computing Surveys, 29(4): 315-321, December 1997. 287, 1997
- [3]<http://www.refsq.org/>
- [4]Brorn Rengnell et al., Summary of 10th Anniversary Workshop on Requirements Engineering, Foundation for Software Quality, 2004, p.7
- [5]Rechard Thayer. Merline H.Dorfman., Software Requirements Engineering, 2nd ed.Los Alamitos, CA: IEEE Computer Society Press,1997
- [6]Ian Sommerville. Pete Sowyler., Requirements Engineering, John Wiley & Sons Addison,1997
- [7]Suzanne Robertson. and James Robertson., Mastering the Requirements Process, ACM Press, Addison-Wesley,1999
- [8]情報処理学会ソフトウェア工学研究会要求工学ワーキンググループ, ウィンターワークショップ・イン・金沢報告, 2001

- [9]Bashar Neseibeh and Steve Easterbrook., Requirements Engineering: A Roadmap, ICSE 2000, 2000
- [10]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・センター編, SEC BOOKS 経営者が参画する要求品質の確保 ～超上流から攻める IT 化の勘どころ～ , オーム社, p.44, J2005
- [11]Software Engineering Standards Committee of the IEEE Computer Society, IEEE Recommended Practice for Software Requirements Specifications IEEE Std.830-1998, 1998
- [12]ISO/IEC9126 as characteristics of SW products. Information technology software product evaluation :Quality characteristics and guidelines for their use, <http://www.iso.org/iso/en/ISOOnline.frontpage>
- [13]JIS X0129, <http://www.jisc.go.jp/jis-act/index.html>
- [14]<http://scholar.google.com/>

9. メタモデルに基づく要求工学手法の開発

筑波大学大学院ビジネス科学研究科 中谷 多哉子

ビジネス領域を分析する目的は、システムへの要求の背景を明らかにして、要求が発せられた根拠を示すことにある。我々はビジネスシステムの要求を獲得するための手法 RODAN を開発してきた。本稿では、RODAN の開発に先立って定義した、ビジネス領域の構造を表すモデルを、手法のメタモデルとして紹介する。

9. 1. 研究の背景と目的

ビジネス領域を分析する目的は、システムへの要求の根拠を、その背景と共に明らかにすることである。したがって、このような分析の結果として得られるモデルは、システムが導入されるビジネス世界を表すものとなる。

それでは、システムが導入されるビジネスとはどのような世界なのであろうか。我々はビジネス領域を分析する手法を開発するにあたって、それ自身の成果物の構造も開発している[5,6]。ビジネス領域の構造を表すモデルは、分析手法を適用することによって得られる成果物の構造でもある。したがって、このモデルは手法の成果物視点におけるメタモデルということになる[1]。

RODAN を適用することによって得られる成果物は、クラス図やアクティビティ図、ゴール図などが想定されているが、実は、このような表現手段はあまり重要ではない。重要なのは、要求の根拠として表されるビジネス領域のモデルには、何を記述すべきかということである。また同時に、これらのモデルがインタビューを受けた人、個人の見解に基づいて記述されているという点にも着目する必要がある。これらのモデルが表しているのは、個人が抱えている課題の根拠、およびその課題の解の構造である。主観的なモデルをいかにして組織で共有されるシステムの妥当な根拠とするかも開発上の課題であった。

西條の構造構成主義[7]によると、個人は各自の関心に基づいて世界を見ているとされている。ここから関係者間で相互理解するためには、関心の根拠となっている構造を示す必要があるとされている。我々はこの考え方を哲学的拠り所とし、妥当な要求を導出するために、以下のプロセスを経ることを提案している。

1. モデルはインタビューを基に個人毎に作り、
2. 主観的な要求の根拠を組織の関係者で共有可能とする
3. そこから、関係者の間で、妥当な根拠とその根拠を了解し、
4. 組織としてシステムへ求める要求と、その根拠を提示する。

本稿では、ビジネスに於ける以下の点に着目して、RODAN のメタモデルを紹介する。

- ビジネスにおける協調作業とは何か
- ビジネスにおける人や組織の依存関係とは何か
- 人が担う役割、ロールとは何か
- 見解の相違は何に由来するのか
- 人が抱える課題と解は、何によって表せるのか

9. 2. メタモデルの開発

上記の検討項目に従って開発されたビジネスの構造を表すモデル、すなわち、RODAN のメタモデルを図 1 に示した。点線で囲った部分は、既存の手法が網羅している領域である。RODAN は、既存の手法を手法断片として統合し、不足部分を新たに補った[1]。本節では、各着目点に基づく考察を紹介し、メタモデル構築のプロセスを紹介する。

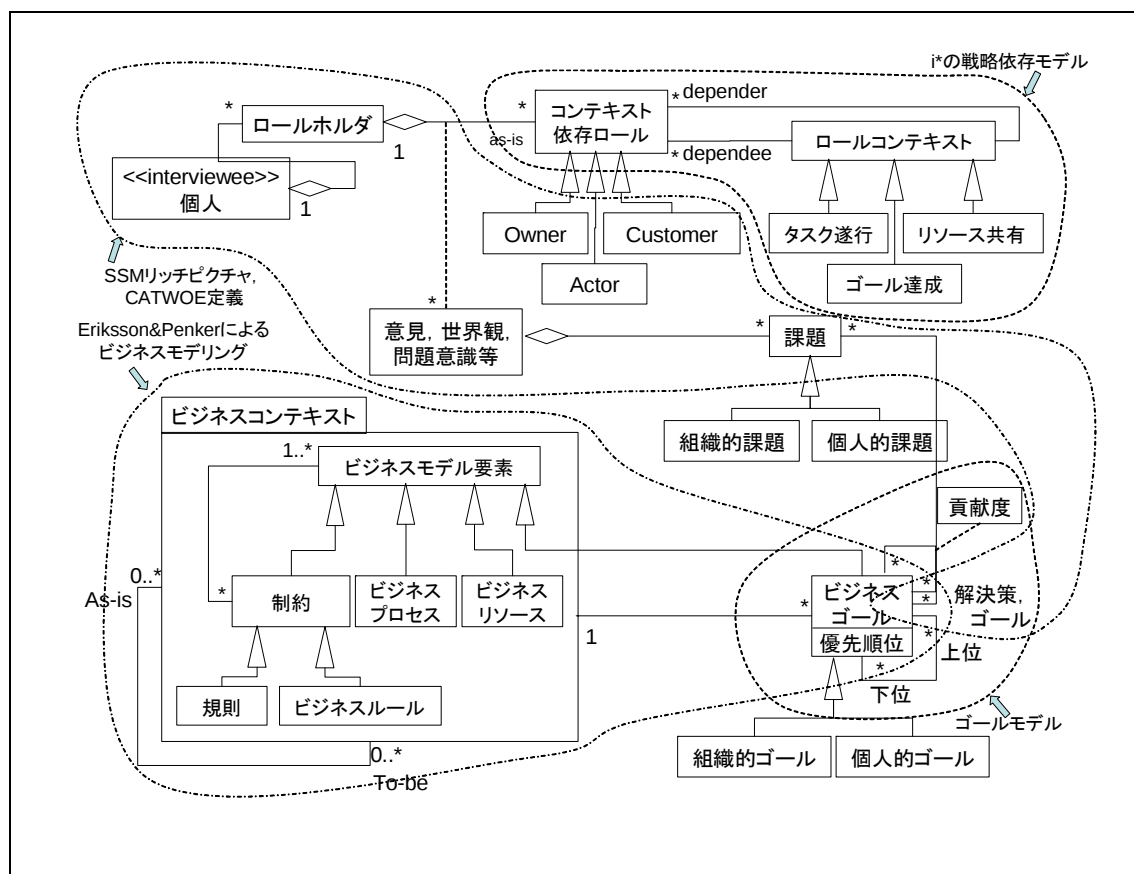


図 9. 1 成果物視点における RODAN のメタモデル

9. 2. 1. ビジネスにおける協調作業とは何か

ビジネスとは、複数の人々が関与し、各々の人が各自に与えられた責務を、自分たちの意思によって達成し、同時に、組織としての目的を達成する営みである。ソフトシステムズ方法論(以下、SSM)によると、ある問題状況を改善させるという活動には、3種類の人や組織が関係するとされている[2]。三種類のロールとは、Customer(受益者)、Actor(遂行者)、Owner(指示権限者)である。これらのロールを担う人や組織は、状況が変われば別のロールを担うこともある。そのため、これらのロールはコンテキストに依存するロールであるとみなすことができる。ビジネスにおける協調作業とは、これらのロールの相互作用である。

9. 2. 2. ビジネスにおける人や組織の依存関係とは何か

協調作業を行うということは、何らかの情報の交流や業務遂行の指揮命令、目的達成といった

関係が、その作業に関与する人や組織の間に存在することを意味する。i*を開発した Yu らは、これらの依存関係を 4 種類(すなわち、リソース共有、タスク遂行、ゴール達成、ソフトゴール満足)に分類している[8]。Yu らが主張するロールの間の依存関係は、協調作業を担うコンテキスト依存ロールの間の関係に他ならない。個人や組織が担うロールは、コンテキストがあることによって、その存在意義を説明することが可能となる。

たとえば、A 氏が担っている顧客接待という役割には、それを指示する上司である B 氏、A 氏が職務を全うすることによって恩恵を受ける顧客の C 氏が存在している。A 氏の役割の意義は、これらの人々との間の依存関係によって説明可能である。同様に、B 氏、C 氏においても、A 氏へ指示する根拠や、C 氏が恩恵を受けることを B 氏が望んでいる根拠によって示すことができる。

9. 2. 3. 人が担う役割, ロールとは何か

溝口らのオントロジーによると、ロールは、基本概念、ロールホルダ、ロールの三層構造の一部を占める概念である[4]。この構造を用いると、ある実体(たとえば、人や組織)が複数のロールを持つ場面が複数存在するという状況をうまく表すことができるようになる。たとえば、鈴木一郎氏は、町内会では会員であり、倫理委員でもあるが、会社では課長であり、プロジェクトの管理者も担っているという状況には、人という基本概念に町内会向けと会社向けという二つのロールホルダを与え、個々のロールホルダに具体的なロールを対応づけた構造を対応づけることができる。

9. 2. 4. 見解の相違は何に由来するのか

何らかのシステムをビジネスに導入しようとする際、様々な利害衝突が発生し、要求を容易には確定できないことも少なくない。また、一度確定した要求が、担当者の異動によって変更されることもある。このような状況を回避するために、契約を交わすことも一つの手段であろう。しかし、現実的に役に立つシステムを開発するためには、ビジネスに関係する人々の相互了解した要求を定義する必要がある。

相互了解には、他者の立場でビジネスを理解する必要がある。ここで、他者の立場が理解できれば課題を理解できるということは、課題というものが、立場、すなわちロールと関係づけられていることを示唆している。

たとえば、人は、状況に応じて様々なロールホルダを保持する。そして、人は、特定のロールホルダを保持することによって、すべきこと、してはならないこと、そして、すべきことの優先順位付けを決定している。この判断基準となるものが、世界観と呼ばれるものである。価値観とも言う。

世界観は、ある人が自己を取り巻くビジネスを観察したときに選択され、現在の状況が問題状況であると判断される。したがって、他者の課題を理解するためには、その人の世界観に基づいてビジネスを観察しなければならないはずである。このような「個人の立場」を表す構造は、SSM の CATWOE の構造とも矛盾しない[2]。CATWOE では、世界観は「W: World View」として説明されている。

9. 2. 5. 人が抱える課題と解は、何によって表せるのか

人が課題を抱え、それを解決しようとしている状況があるとき、我々が観察すべき対象は、シス

テムの導入、それ自体が課題の解決策とされる領域である。したがって、課題の解は、「システムを導入することによって、現状のビジネスプロセス、ビジネスルール、リソース共有の形態、リソース構造などを変更する」ものとして提案される。課題を抱えている現状は、これらのビジネスプロセス、ビジネスルール、ビジネスリソースによって、可視化することが可能である。もちろん、課題が解決された将来の状況も同様に可視化することも可能である[3]。ここで、ビジネスゴールは、課題を解決された状況や現状を維持し続けている状況を実現することに相当し、いわば、現状と将来のモデルを結びつける根拠となるものとして位置づけることができる。

我々は、これらのビジネスのモデルを、課題のコンテキストを表すものであるという意味で、ビジネスコンテキストと名付けた。

9. 3. 最後に

本稿で示したメタモデルを満足する成果物を生成する手法には、様々なものがある。RODAN は、SSM、ゴール指向分析、i*、Eriksson&Penker 法をはじめとした様々な手法を手法断片として選択し、それらを適所に配置して統合している。また、統合時に既存の手法では満足できない部分は、新たなモデルを提案して補足して、手法全体を再構成するという方針を採用した。手法の適用事例は、随時報告を行っている。

参考文献

- [1] Brinkkemper, S., Saeki, M. and Harmsen, F.: Meta-Modelling Based Assembly Techniques for Situational Method Engineering, Information Systems, vol. 23, no.7, pp. 489—508 (1998).
- [2] Checkland, P. and Scholes, J.: Soft Systems Methodology in Action, John Wiley & Sons (1991).
- [3] Eriksson, H.-E. and Penker, M.: Business Modeling with UML, John Wiley & Sons, New York (2000).
- [4] 溝口理一郎, 池田満, 来村徳信: オントロジー工学基礎論～意味リンク, クラス, 関係, ロールのオントロジー的意味論, 人工知能学会誌, Vol. 14, No. 6, pp. 1019-1032 (1999).
- [5] NAKATANI, Takako and FUJINO, Terunobu: Role and Owner Based Business Domain Analysis, Proc. of the Asia Pacific Software Engineering Conference'2005 (APSEC2005), IEEE, pp.130—137 (2005).
- [6] 中谷 多哉子, 藤野晃延: RODAN のためのビジネスゴール分析手法の提案, ソフトウェア工学の基礎ワークショップ 2005, ソフトウェア科学会, pp.129-134 (2005).
- [7] 西條剛央: 構造構成主義とは何か～次世代人間科学の原理～, 北大路書房(2005).
- [8] Yu, E. S. K.: Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering, Proc. of the 3rd International Symposium On Requirements Engineering (RE'97), IEEE, pp. 226-235 (1997).

メタモデルに基づく要求工学手法の開発

(有)エス・ラグーン, 筑波大学大学院
中谷多哉子

新しい要求分析手法の開発 に向けて

- 手法開発のヒント
 - 問題発見型::問題の解法をプロセス, プロダクトの両面から解析する
 - 経験蓄積型::ベテランの解を定式化
- 新手法構成法
 - 新手法開発型::新しいモデルを考案
 - 新しいゴールモデル, 新しいシナリオ構造
 - 新しい折衝手法など
 - 既存手法統合型::既存手法を組み合わせて適用. 不足部分の拡張, 補足
 - 成果物やプロセスの構造を定義し, 既存手法をメソッドフラグメントとして新手法を構成する

ビジネス領域分析手法： RODANの場合

- RODAN: Role, Owner based Business Domain Analysis Method
 - 問題発見型＋既存手法統合型の手法
 - 併せて、経験蓄積による精緻化
- 手法の着眼点
 - ビジネスは個人によって構成されている
 - ビジネスモデリングによる、個人のビジネス領域の構造を明らかにする
 - 関係者間の信念対立をモデルによって解消することを目指す
 - ∴信念対立が要求変更の原因となっていることがある

2011/8/31

NAKATANI Takako

3

問題：個人は矛盾する問題、解を持っている

- 人は立場によって異なる価値観、信念を持ち得る.
 - 信念の根拠として、個人の立場、環境をモデルとして表す
 - 関係者が、モデルを介して他者が提案する解の妥当性を了解することができるようになる
- 個人の立場、環境、解をモデルで表すための構造を構成
 - >モデルのモデル＝メタモデル

2011/8/31

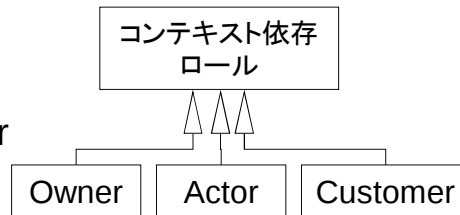
NAKATANI Takako

4

協調作業とは何か

- 業務(コンテキスト)毎に3つのロールが存在する考える

- 遂行者: Actor
- 指示者: Owner
- 受益者: Customer



- 協調作業＝3種のロールの相互作用である

2011/8/31

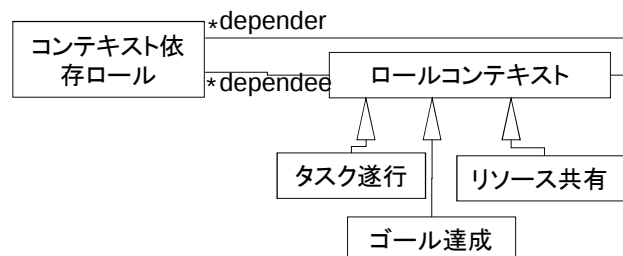
NAKATANI Takako

5

ロールの間に三種類の依存関係を識別する

- 業務には, 以下の依存関係がある

- タスク遂行
- リソース共有
- ゴール達成



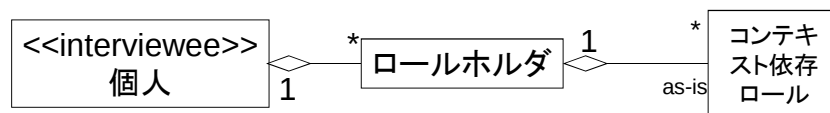
2011/8/31

NAKATANI Takako

6

人は複数のロールを持つことができる

- ロールを担うことによって、責務が発生し、それによってビジネス領域を観察する視点が定められる。



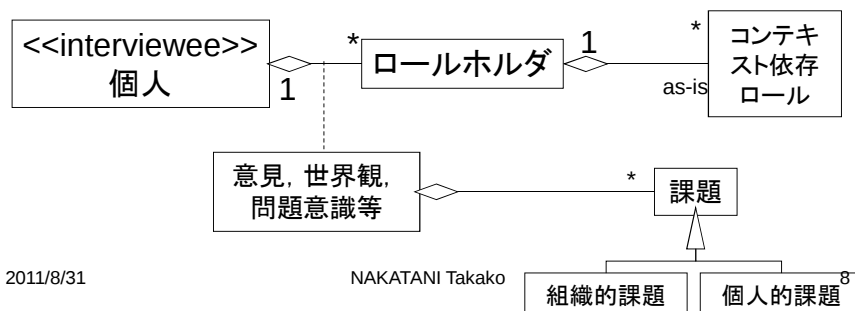
2011/8/31

NAKATANI Takako

7

見解の相違(の構造)とは？

- 責務が異なれば視点は異なる。
- 視点が異なれば課題の優先順位も異なる
- 課題の中には、個人的な問題、組織の責務がある

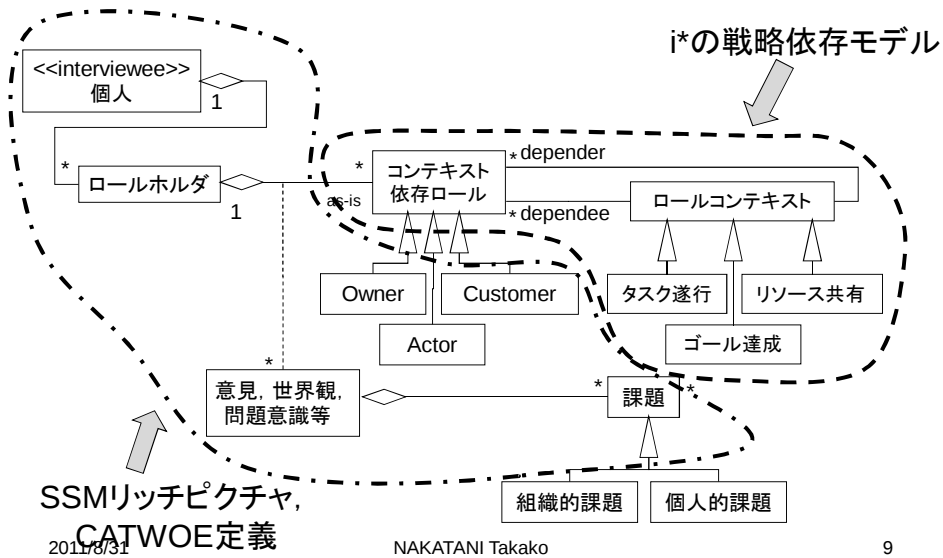


2011/8/31

NAKATANI Takako

8

個人が抱える課題の構造



課題と解の構造は 何によって表せるか

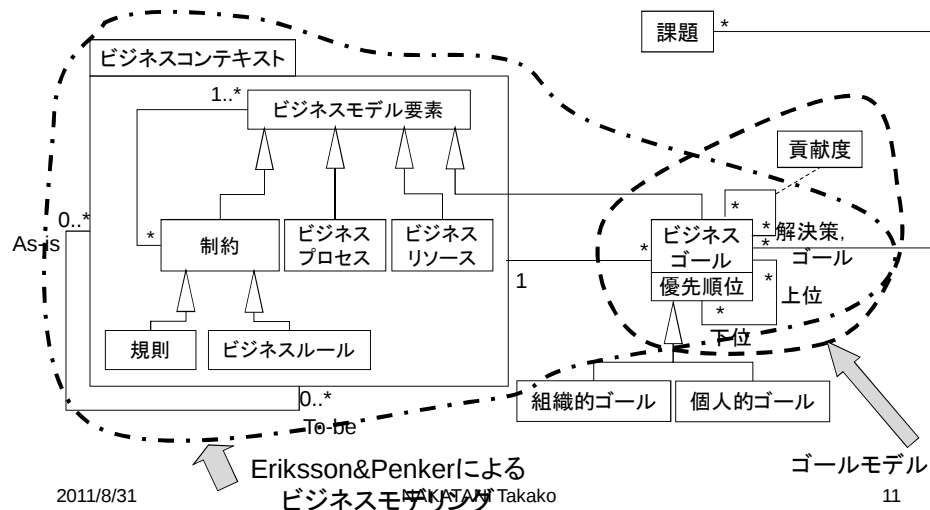
- ビジネスコンテキストによって、課題と解の構造を表す
 1. 課題→ゴールの設定→ゴール達成手段としての解
 2. ビジネスコンテキスト
 - ビジネスルール, ビジネスプロセス, ビジネスリソース
 - AsIsビジネスモデル＝課題のコンテキスト
 - ToBeビジネスモデル＝解のコンテキスト

2011/8/31

NAKATANI Takako

10

課題，解，ビジネスコンテキストの構造



手法の統合と補完

- 既存手法のメタモデルと開発したい手法のメタモデルの比較
 - 充当と補完
- メタモデルの妥当性確認は，事例適用によって行う
- 既存手法との競合は発生しない
 - 適用可能な場所に既存手法を適用すればよい
 - RODANの主張は成果物のメタモデルのみ

RODANの現状

- メタモデルはほぼ完成
 - 既存手法に比較して、視野が広い.
 - 個人のロール, ロールの依存関係から
 - ビジネスモデリング, および
 - 課題発見, 課題分析までを含む
- スコープ外
 - ステークホルダ間の競合状態のモデル
- 未完
 - 個人の主張の合理性を示すモデルの善悪を判断するのではなく, 了解(=理解したと納得)することが重要(このプロセスは未完)
 - 競合, 競合の解消は折衝プロセスで行う

2011/8/31

NAKATANI Takako

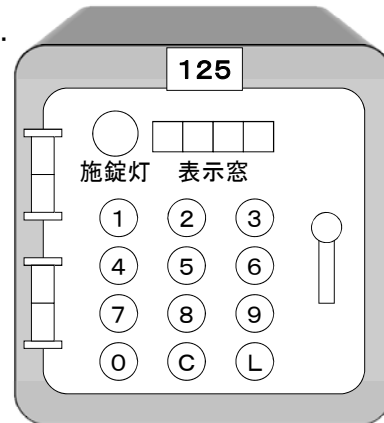
13

金庫問題

山崎利治: “DW2005のための問題”, ソフトウェア技術者協会, SEAMAIL, Vol.14, No.15(01/2005), pp.51--52.

ご利用に際して

- あなたの物置番号は125です。いま扉は閉まっていて施錠してあります(施錠灯がついています)。
- 物置番号の書かれた物置の扉にはパネルがあり、そこに、施錠灯、0から9までの押ボタン、C、Lの押ボタン、10進数字4桁の液晶表示窓、90度回転する取手が付いています。



2011/8/31

NAKATANI Takako

15

- 扉の開けかた
 1. 数字ボタンを1234(施錠鍵といいます)と左から順に押してください。
 2. 押した数字を表示窓で確認してください。それが1234であれば施錠灯が消えます(解錠)。
 3. そこで扉の取手を時計回りに90度廻して手前に引けば扉が開きます。
 4. 数字ボタンを押し間違えたときは、Cボタンを押して、始めからやり直してください(Cボタンを押すと数字窓の数字が消えます)。
- 扉の閉めかたと施錠
 1. 扉を閉め取っ手を逆時計回りに90度廻します(機械的な施錠)。
 2. Lボタンを押します(電子的な施錠)。
 3. ここで施錠灯が点きます(直前に扉を開けたときのあるいは、次項によって変更した鍵による施錠)。
 4. その後は解錠手続きをしない限り、取手を廻しても引いても扉は開かないはずです。

2011/8/31

NAKATANI Takako

16

宿泊客

理想的な宿泊客 A氏

物忘れがちな宿泊客 B氏

不注意な宿泊客 C氏

短気な宿泊客 D氏

貴重品

宿泊客の貴重品が安全に宿泊客によって保管されている

貴重品を各部屋の金庫に保管する

宿泊客の貴重品が安全にホテル従業員によって保管されている

貴重品をホテルの金庫に保管する

宿泊客の満足度を得る

宿泊客自身が貴重品を保管できるように、各部屋に金庫を提供する

宿泊客の貴重品はいつでも自由に出し入れ可能で、安全に保管されている

従業員U氏

保身的ホテル従業員SA氏

献身的ホテル従業員SB氏

支配人GM氏

2011/8/31

NAKATANI Takako

17

所有者別ゴールモデル

凡例

- ゴール
- ※要求
- 達成手段
- 貢献 (→)
- 依存 (→)
- ▲ and 依存
- △ or 依存
- 貢献度 +: 促進, -: 阻害
- not -: 阻害はない
- not+: 促進はしない

支配人GM氏

悪意のある従業員U氏

ホテル従業員SS氏, SD氏

宿泊客の貴重品が安全に保管されている

宿泊客が部屋の金庫を利用している

宿泊客の貴重品を取得する

宿泊客の部屋に迅速に進入できる

貴重品を迅速に入手する

従業員の特権を使う

宿泊客に部屋の金庫の利用を依頼する

宿泊客に金庫の利用方法を伝える

宿泊客に金庫を利用しない場合のリスクを伝える

従業員が想定する宿泊客

宿泊客の貴重品の保管方法でも自由に出し入れ可能である

誤った使い方をしても、金庫の貴重品が安全に手元に戻る

誰にでも使いやすい金庫である

信頼性の高い金庫である

安全性の高い金庫である

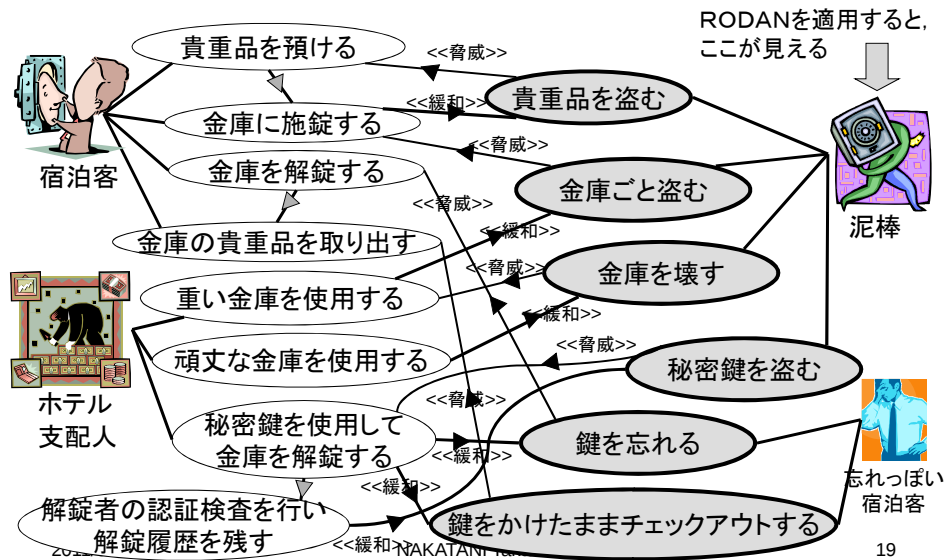
各部屋に金庫を設置する

2011/8/31

NAKATANI Takako

18

金庫のMisuse Caseの例



まとめ

- 要求分析手法としてRODANの開発過程を紹介
 - メタモデル構築→既存手法の割り当て→補完
- RODANのプロセスを紹介
 - 信念対立×→モデル化による判断停止→相互了解
 - 定義された要求は、再度、現実世界へ差し戻し、妥当性の確認を行う。(これは、要求の確認という意味ではない。)
- 最後に:
 - システム開発に必要な妥当な要求とは、ステークホルダー間の信念対立を解消しなければ得ることはできないであろう

参考文献

- Checkland, Peter and Scholes, Jim, Soft Systems Methodology in Action, John Wiley & Sons, 1991.
- Eriksson, Hans-Erik and Penker, Magnus, Business Modeling with UML, John Wiley & Sons, 2000.
- 池田 清彦, 構造主義科学論の冒険, 毎日新聞社, 1990.
- 中谷 多哉子, 藤野晃延, "RODANのためのビジネスゴール分析手法の提案", ソフトウェア工学の基礎ワークショップ2005, pp.129--134, 2005.
- NAKATANI, Takako and FUJINO, Terunobu, "Role and Owner Based Business Domain Analysis", Proc. of the Asia Pacific Software Engineering Conference'2005 (APSEC2005), pp.130--137 (2005).
- 西條 剛央, 構造構成主義とは何か～次世代人間科学の原理～, 北大路書房, 2005.
- Yu, Eric S. K., "Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering", Proc. of the 3rd International Symposium On Requirements Engineering (RE'97), pp.226--235.

— 禁 無 断 転 載 —

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

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2-45 タイム24ビル17F

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