

「要求開発」と「システム開発」

(株) 豆蔵

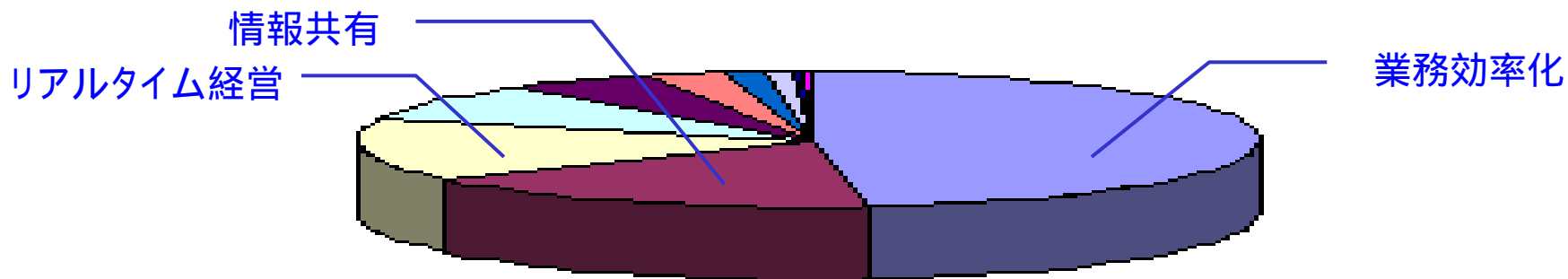
山岸 耕二

Agenda

- 情報化投資の効率化
- 「要求開発」という開発
- 「要求開発」の構成要素
- 「要求開発」の現状と推進体制
- まとめ

今情報化投資のROIが注視されている

- 情報化投資は絞られてきている
- 情報化投資の目的
 - 業務効率化によるコスト削減 …………… 48.2%
 - 情報共有などによる営業や販売の強化 …… 17.5%
 - 経営指標の早期把握やリアルタイム経営 …… 15.2%



日経コンピュータ 2003/11/17号より引用

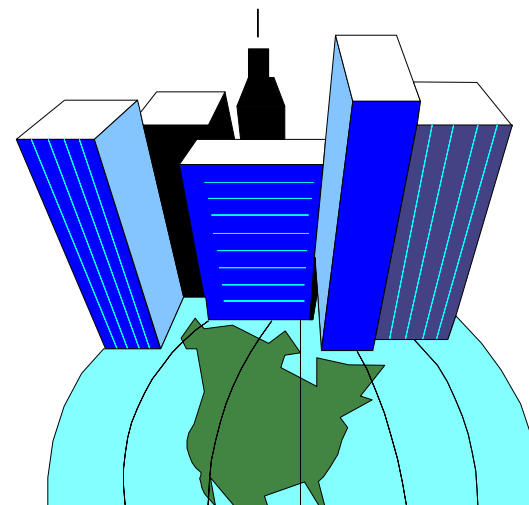
「莫大な情報化投資に対して、ITはいったい何をしてくれたのか」

「どんなシステムを作れば業務は効率化されるのか」

最近の開発のトレンド

基幹系システムの再構築

- システム化対象範囲が広がっている
 - 企業ポータル
 - SCM (Supply Chain Management) 構築
 - 企業、グループ統合
- 機能別システムから、全体最適化を目指す統合化
- 企業内を統一したアーキテクチャで再構築しようとしている
Enterprise Architecture
- 人、組織、装置、システムなどがトータルで業務を構成する
 - 業務をトータルコーディネーション
 - システムの境界の再定義が必要



プロジェクト失敗の要因は、企画・要件定義

- 国内のシステム開発プロジェクトの成功率 **26.7%**
- プロジェクトの遅れの原因
 - 要件定義が計画より長引いた…………… **37.7%**
 - 企画作業が長引いた…………… **22.7%**
- システム品質問題の原因
 - 要件定義が不十分…………… **35.9%**
 - システムの企画が不十分…………… **18.7%**
- 稼動システムの満足度
 - 「計画通り利用しているが不満足」… **24.3%**
稼動しても役に立たない!



ROIをクリアしないシステムの例

- 人の作業をただ電子化したようなシステム
 - やたら画面数が多く、複雑、拡張性がない
- 全体効率化に寄与しないシステム
 - ローカルなユーザの要求で作られたシステム
- 既存のシステムをストレートにリプレースしたシステム
 - 従来の人系とコンピュータ系の境界が固定化
 - システムに合わせた業務フロー
- 業務が変わってもどう変えればいいかわからないシステム
 - システム化されたときの意図が不明
 - 業務との対応関係が管理されていない

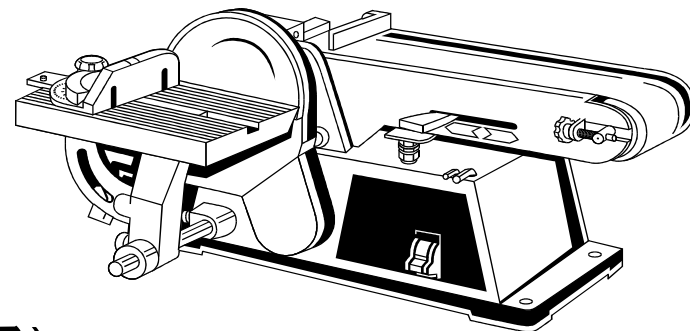


ROIを高めるために

- 従来は、「システム開発」のコスト削減がメインだった

➡ **いかに効率よく作るか**

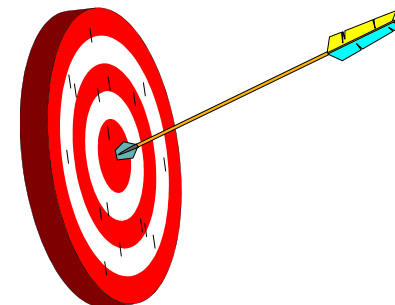
- システム開発プロセスの標準化
- 開発ツールの導入
- フレームワーク、コンポーネント再利用
- 低コストな労働力の利用(外注、オフショア)



- 今ビジネスゴールに合致するシステムのプランニングに

➡ **何を作るべきか**

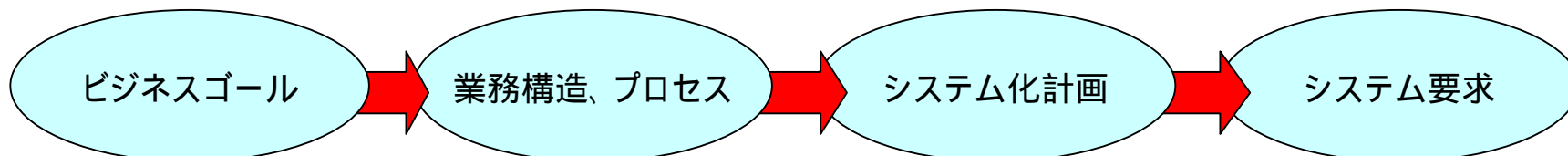
- 業務に合わせたシステム要求
- 適正なシステムスコープ
- 業務との整合性、トレーサビリティ



間違えたものを正しく作っても投資効果は得られない

業務からロジカルにシステム要求を導く

- ビジネスゴールに直結したシステム化をはかる
 - Business Drivenにシステム化をプランニングする
 - ビジネスゴールとシステムの要求がロジカルにつながっている
- ビジネスとシステム要求がリンクしている
 - ビジネスが変わったときに対応する要求が明確
 - ビジネス追尾型の情報化



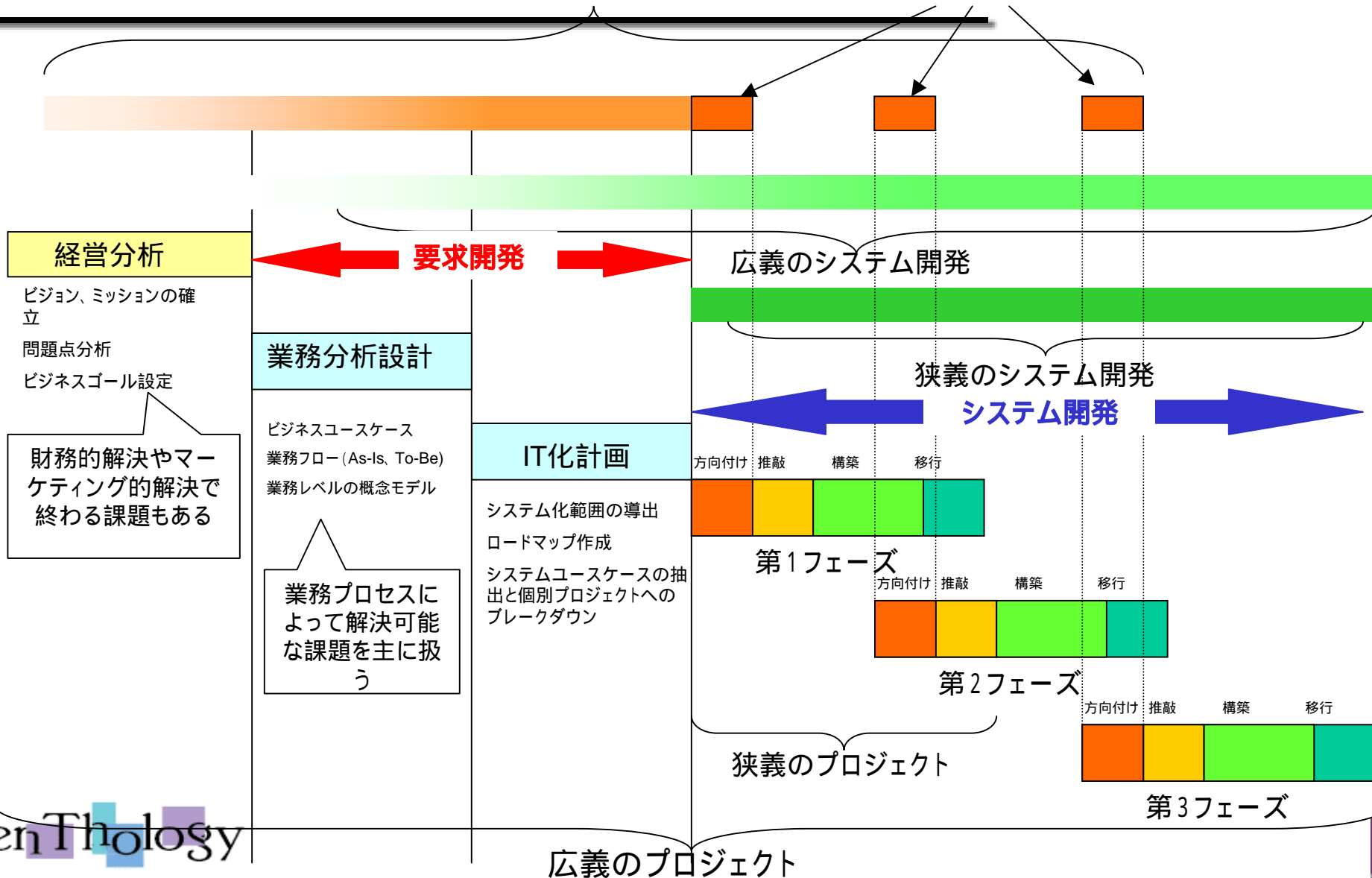
ビジネス環境が変わると**業務形態**が変わり、**システム要求**が変わる

「要求開発」という開発

プロジェクトの全体像

広義の要求開発 (要求定義)

狭義の要求定義



従来の「要求定義」

- 広く要求定義などと呼ばれてきた
- 各システム開発の初期作業と混乱してきた
- 実施者は、主にIT推進プロジェクトや事業企画、情報システム部
- 限られたSIベンダー、コンサル会社が手を伸ばしていた
- SIをとりたいベンダーがおまけにこの領域のコンサルもただでやっていた
- 決まった進め方や成果物の定義など、標準的なものはなかった。
- DOAに基づく手法は企業内のデータモデリングの観点からそれなりの方法論をもっていたが、DBの設計的な視点が強い
- ERPベンダーはフィット&ギャップの名の下に独自のやりかたで業務分析をやってきた

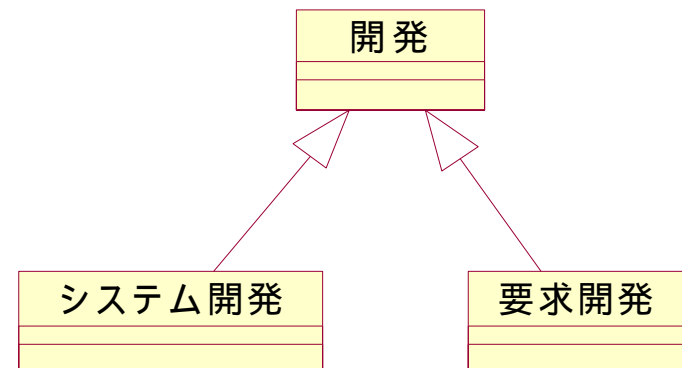
要求は「開発」するもの

- 「要求分析」、「要求定義」などは、要求がすでに存在しているという前提に立っている
- ユーザはユーザなりの要求をもっているが、ユーザの理解の範囲内で生まれたもので、それが業務効率化に結びつくものとは限らない。
- ユーザからヒアリングした要求は、直感的、場当たりのであることが多い。
- 要求は、業務を分析することによって開発される。ロジカルに導かれる必要がある。

「開発」とよぶ理由

- 要求はもともとあるものではなく、業務を元に開発されるものである
- 「システム開発」と「要求開発」を同列に表すことができる
- システム開発と要求開発はどちらも「開発」なのでアナロジーで語ることができる。システム開発で培った技術体系が利用できる。

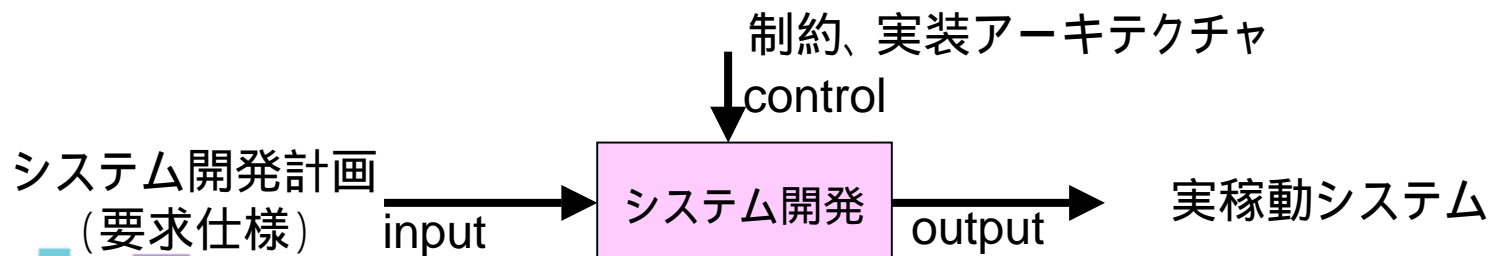
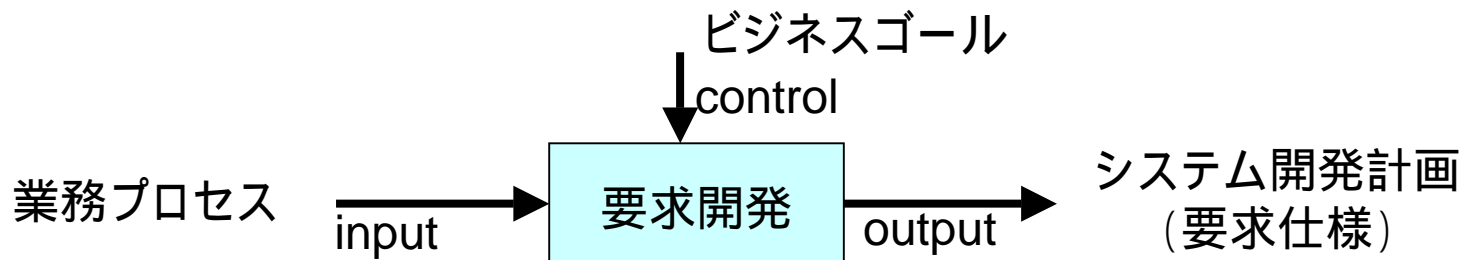
- 開発プロセス
- 開発方法論
- モデル
- ツール



システム開発と要求開発のアナロジー

● 入出力

- システム開発: **要求仕様**を入力に、技術的制約や実装アーキテクチャに基づき**実稼動システム**を出力する
- 要求開発: **現状業務、想定業務構造**を入力に、ビジネスゴールに基づき**要求仕様**を出力する



システム開発と要求開発のアナロジー

● モデル

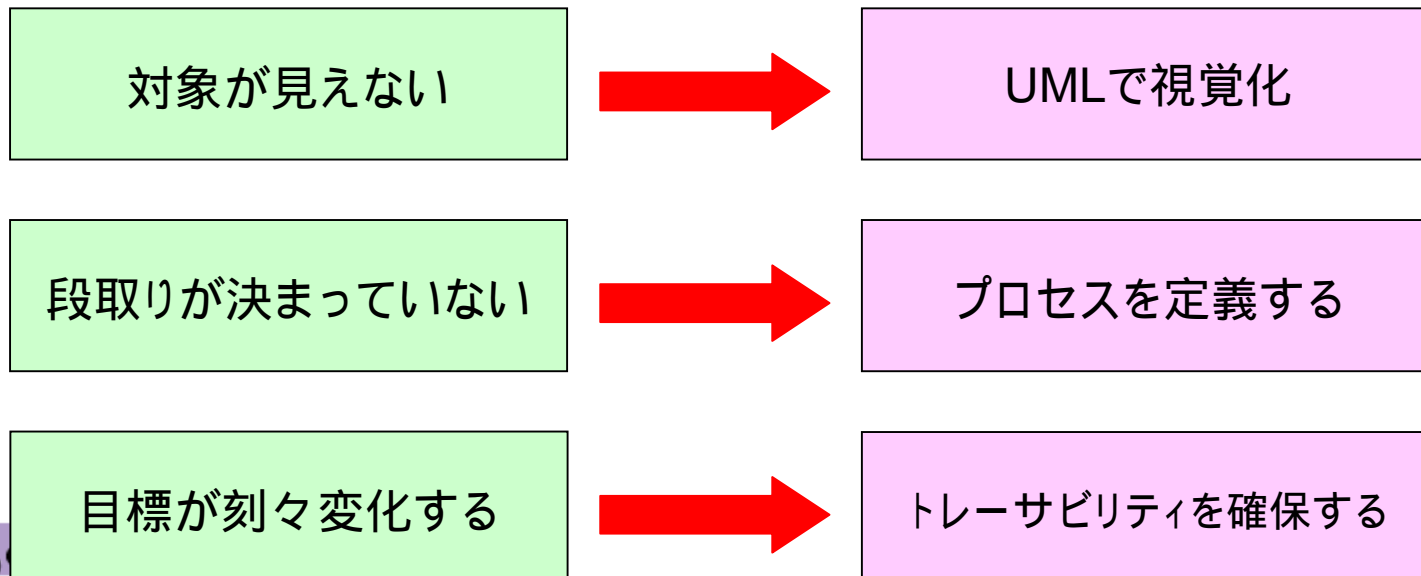
- システム： オブジェクトのコラボレーションによってサービスを提供する
- 業務： ビジネスリソース(人、組織、装置、ソフトウェアシステムなど)のコラボレーションによってサービスを提供する

● 取組み方

- どちらも見えない複雑な系を相手にする
 - ▶ 問題を明確にする
 - ▶ 対象をモデルによって視覚化する
- システムモデリングとビジネスモデリング

システム開発と要求開発のアナロジー

- 対処方法：業務もソフトウェアシステムも目に見えない複雑な系
 - 対象が見えない
 - 段取りに決まり手がない
 - 目標は刻々変化する



システム開発と要求開発の相違

● スコープ

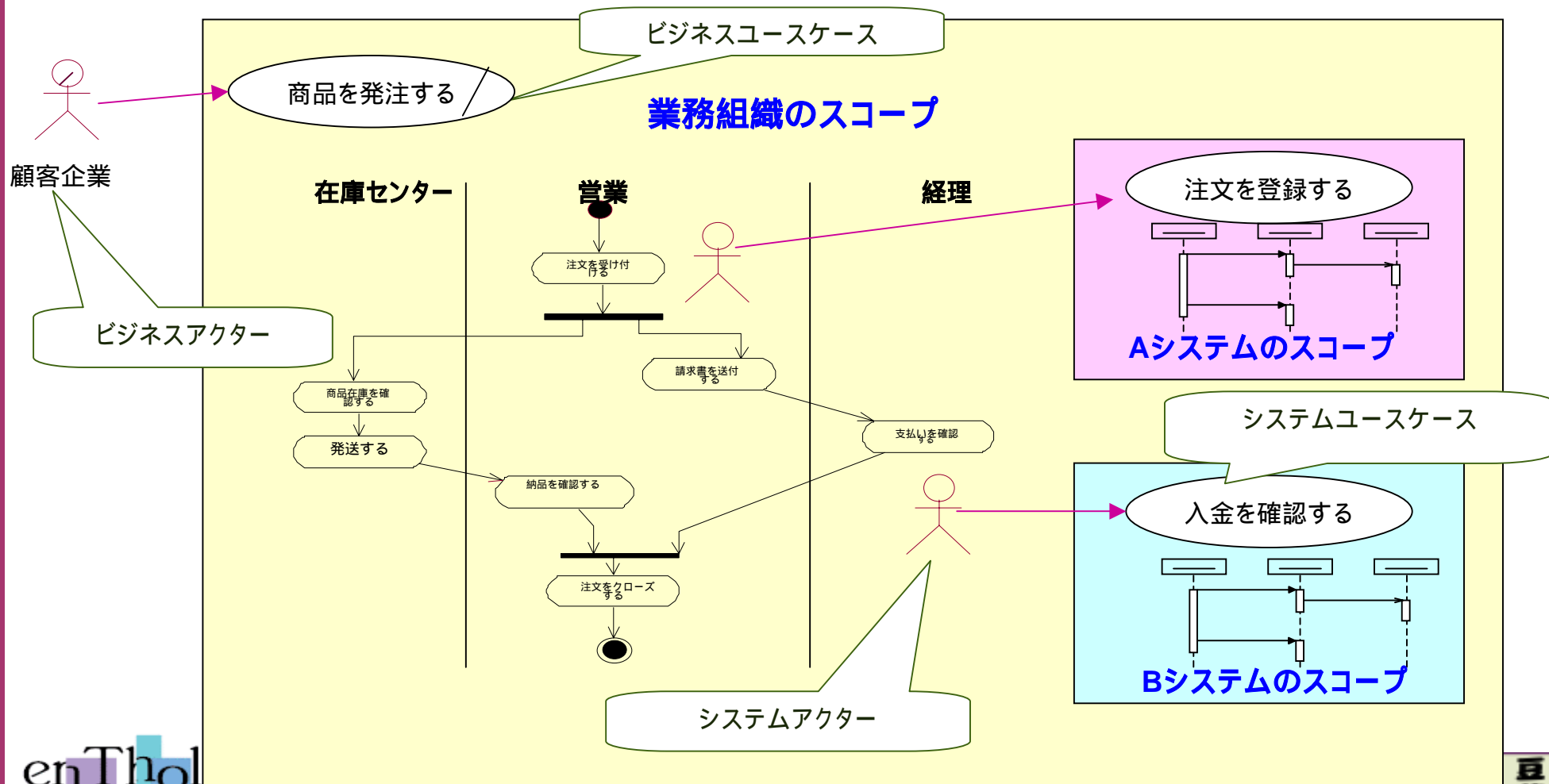
- システム開発： コンピュータシステム周辺
- 要求開発： 業務領域周辺

● 対象

- システム開発： ハードウェア、ソフトウェア
 - ▶ 比較的ロジカルな世界
- 要求開発： 業務
 - ▶ 人間系、一見こてこて、理不尽、ポリティカル

ビジネスとシステムの階層構造

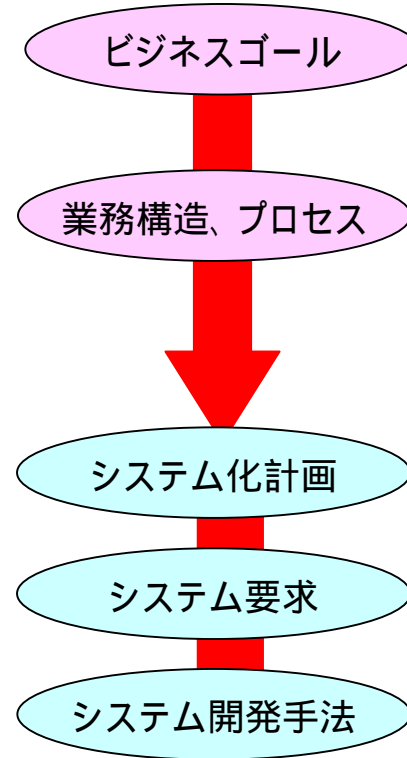
- システムは、業務のサブシステム(階層的)



ビジネスとシステムの階層構造

- ビジネスゴール(何のために: Why)
- ビジネスユースケース(何をする: What)
- 業務の設計: 実現方法(どうやって: How)

- システムの目的(何のために: Why)
- システムユースケース(何をする: What)
- システム設計: 実現方法(どうやって: How)



システムのWhyとWhatを開発する

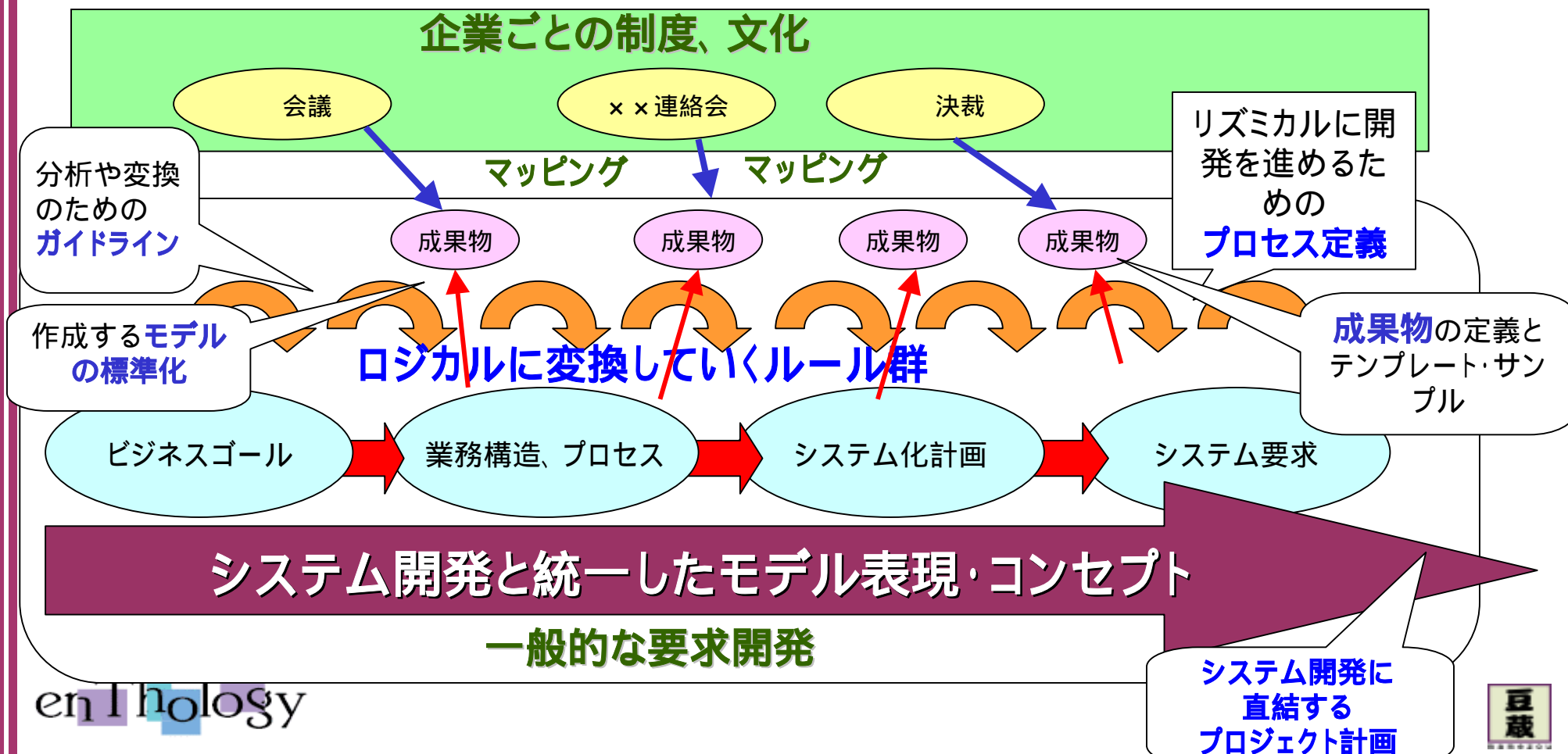
なぜ要求開発が重要か

- 要求開発はすべての企業が避けて通れない
 - いまや情報化業務は、すべての企業の中核業務
 - 情報化投資は肥大化している。他業務のBPRは進めているが、情報化業務のBPRは進んでいない
 - 「システム開発」をアウトソースしても、「要求開発」をアウトソースするのは非常に困難
- システム開発はオフショアへの流れ
 - 要求開発とシステム開発が明確に分離できればオフショアのコントロールも効きやすくなる
 - 要求開発は、オフショア化は困難。もっとも付加価値の高い開発部分となる

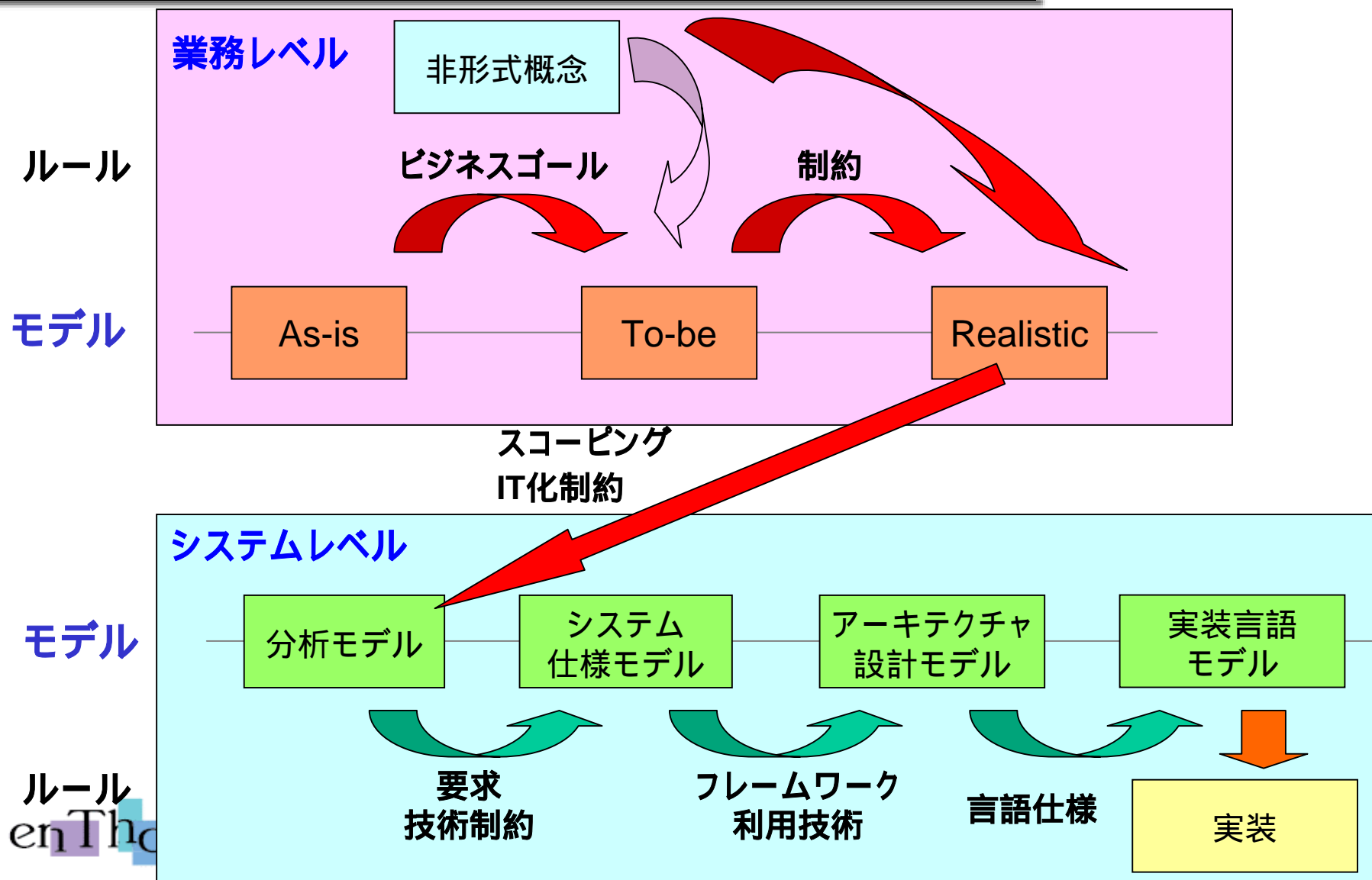
「要求開発」の構成要素

全体像

- 要求開発では、企業の制度や文化によるマイルストーンとマッピングが必要



開発をモデルの変換過程としてとらえる



「要求開発」の最終成果物

- 業務とトレーサビリティがあるシステム化の仕様(システム化プロジェクト計画)
 - システム化の仕様であって、個別システムの仕様ではない
- 現状から目標とするシステム導入までのロードマップ
 - 複数のシステム開発プロジェクトで実現
 - 既存システムから段階的に移行する計画
- システム全体の基本アーキテクチャ
- システムの要求仕様
 - 各システムへの割り振り
- 予算、スケジュール

ビジネス(プロセス)モデリング

- UMLをベースに業務の可視化を行う
 - 業務の全体構造をとらえる
- UMLを使う理由
 - 「システム開発」と統一したコンセプト、統一したモデル
 - ▶ システム開発はUML利用が一般的になっている
 - 様々なビューで複雑な対象をモデル化
 - ▶ 複雑な系を表せる表現力
 - 世界標準のモデリング言語
 - ▶ 方言や様々なバリエーションが発生しにくい
 - ▶ グローバルにコミュニケーションできる

代表的なUML利用例

- ユースケース図
 - ビジネスユースケース、システムユースケース
- アクティビティ図
 - 業務フロー
- クラス図
 - 業務に現れる概念の関係を表す
- シーケンス図
 - ビジネスプロトコルを表現する (利用頻度 小)

モデルレベルの標準化

- 各アクティビティで作成されるモデルは適切なレベルでなければならない
 - 視点
 - スコープ
 - 抽象度
 - 詳細度、粒度
- レベルを標準化できればモデルを集積できる
 - レファレンスモデルの開発
 - カテゴリーズ、モデル再利用、パターン化

その他

- ROIの評価方法
 - 情報化の投資効果をシミュレートする
- 問題解決、合意形成手法
 - SSM、KJ法などヒューリスティックな手法
- Workshopのファシリテーション手法
 - クロスファンクションチームによるプロジェクト推進体制

「要求開発」で大切なこと

- 業務から要求が導かれる(ビジネスドリブンであること)こと
- 業務と要求のトレーサビリティが確立されること
- 一貫したコンセプトやモデルに基づいてIT化が行われること
 - 要求開発とシステム開発のモデルや道具立てが共通
- 標準化がなされている
 - プロセス、用語、成果物(モデル)
 - モデルのレベルが整理されている

「要求開発」の現状と推進体制

ビジネスモデリングを標榜するグループ

- 名称: ビジネスモデリング研究会
- 発足: 2003年5月
- 主要構成メンバー (50音順)
 - ウルシステムズ 平澤氏、河野氏、石川氏
 - オージス総研 山口氏、佐川氏
 - TIS 熊谷氏
 - 東洋エンジニアリング 依田氏(シナジー研究所)、中下氏、他
 - 清水建設 安井氏、武井氏、野田氏
 - サントリー 片山氏、村林氏
 - 日本IBM 藤井氏、渡辺氏
 - 日本総研 細川氏、鈴村氏、他
 - 豆蔵 羽生田、萩本、他

総勢36名

ビジネスモデリング研究会の趣旨

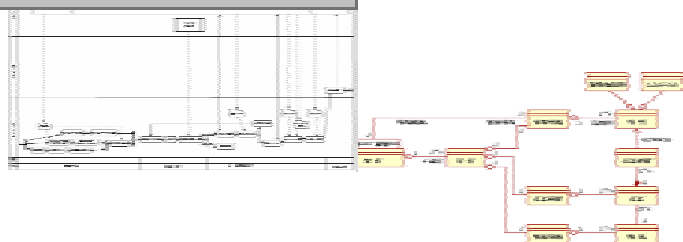
- システム開発に至るビジネスモデリングの領域について、定義、活動、成果物などのコンセンサスを形成し、「(システムの)要求開発」なる分野を確立する。
- 2004夏に「ビジネスモデリングプロセス」のドラフトリリースを目指して活動中。

豆蔵における活動

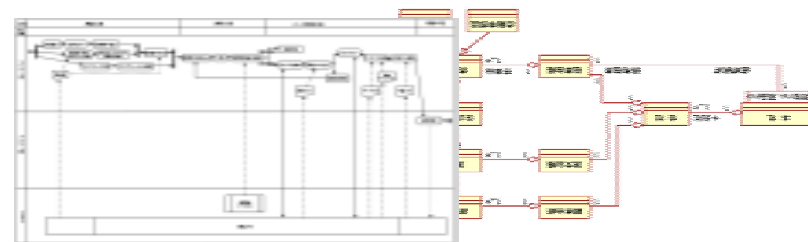
- BDA(Business Driven Architecture)のコンセプトに基づき「要求開発方法論」を策定中。
- すでに案件ベースで事例を蓄積。

標準業務フロー

標準業務フローの作成
本質部分に的を絞った抽象化
(工程の定義等の見直し)



個別業務フローの作成
各部門に利用できる程度に具体化する(入出力など)



オフショア開発手法への展開

- オフショア開発での失敗事例が多い
 - 「要求開発」、「システム開発」の区別なく、方法論未確立のまま、日本的開発をまとめて海外へ
 - コミュニケーションギャップ、常識の認識ミス、受け渡し方がアドホック
- オフショア開発のための開発手法が必要
 - 「要求開発」のアウトプット
 - 「システム開発」時のモニタリングポイント
 - ▶ レビュードキュメント定義、フィードバック形式
 - 受け入れ時の成果物
 - 成果物の受け入れテスト

まとめ

- 企業の情報化投資のROIを高めるには、「要求開発」をシステムティックに行い、経営とITを直結させる必要がある。
- 「要求開発」のコア技術として、UMLベースのビジネスモデリング、要求開発プロセスの確立が重要である。
- 「要求開発」の開発方法論を確立する動きが始まっており、「システム開発」方法論以上に企業に付加価値を提供するポテンシャルをもつ

「要求開発」と「システム開発」 おわり