OPENTHOLOGY

BDA コンセプトに基づく OpenThology要求開発方法論

Ver 0.6
2005年1月25日

OPENTHOLOGY

ドキュメント構成

- Openthology要求開発方法論
  - コア　[Openthologyコアverx.x.ppt]
  - プロセス詳細説明書　[OpenThologyコア（プロセス詳細説明）verx.x.ppt]
- Openthologyガイドライン
  - ファイル名　[Openthologyガイドラインverx.x.ppt]
- バージョンについて
  - 当面の間すべてのファイルに同じバージョン(現在0.6)を使います。今後、ガイドラインが分割される場合、ガイドラインごとにバージョン管理を行います。
- 本資料の利用について
  - 下記のクリエイティブコモンライセンスに従っていれば、商用にて無償利用が可能です。
    - http://www.openthology.org/download.htm
- 資料の読み方および印刷について
  - 現バージョンは、パワーポイントのノート部にて説明をしていますので、そちらもご覧ください。また印刷時は「印刷対象のノート」を利用すると説明が印刷できます。

OPENTHOLOGY

改訂記録

リリース日	version	内容	改訂部分	担当
2004年6月23日	Ver 0.1	ビジネスモデリング研究会に初期配布	全体をざっくり作成	萩本
2004年7月23日	Ver 0.2	ビジネスモデリング研究会に2回目配布	プロセス詳細説明追加。組織モデルをここに吸収。解説をノート部分に挿入	萩本
2004年8月6日	Ver 0.3	ビジネスモデリング研究会に提供	プロセス詳細説明追加。ノートに説明追加。清水建設安井チームによる開発タイプ図の追加。 enThology全体構成説明資料追加、コアとプラグインにファイル分割	萩本
2004年9月30日	Ver 0.4	ビジネスモデリング研究会に提供	動機編を豆蔵山岸さんの資料に差し替え、永和システムマネジメントの平鍋さんの「使われない、使えないシステム」のページをマージ。 BDA宣言に豆蔵羽生田さん案を部分採用。 豆蔵内で発足したenThology開発チーム（矢崎さん、神崎さん、萬本さん、岡村さん、岸さん、瀬谷さん）の意見を参考にコア概念モデルを修正し、メタモデルを定義。 プロセス詳細説明編を別ファイル化。	萩本
2004年11月17日	Ver 0.5	ビジネスモデリング研究会による拡張	名称をOpenThologyに変更。（清水建設安井さんからロゴを提供していただきました） Thanks！ プロセス詳細編全般について、清水建設の安井さん、野田さん、武井さんと豆蔵開発メンバーの考えを導入。 ジャステックの佐々木さんのデシジョン分析図を導入。 主にプロセス詳細編の改訂。	萩本
2005年1月22日	Ver 0.6	一般公開	公開に向けて、ノート部分を使つた解説を補強	萩本

OPENTHOLOGY

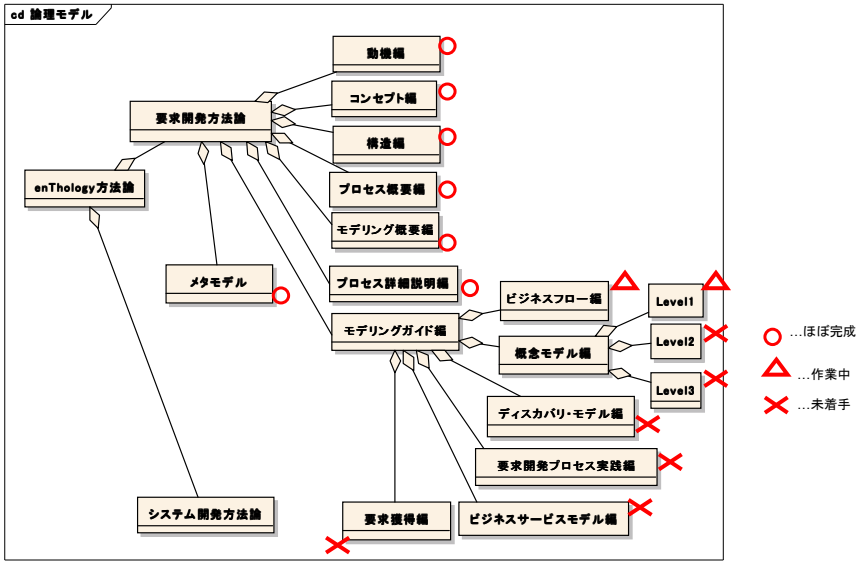
Specal Thanks!

- アイデアを提供していただいた方々へ
  - 様々な資料を提供し、研究会の土台を築いていただいた、、日本総研の細川さん、ウルシステムの河野さん、豆蔵の山岸さん、羽生田さん
  - 要求開発をプロジェクトで採用していただいたTSLとプレアードのみなさん
  - Ver0.4にて多くの助言をいただいた豆蔵のenThology開発Groupのみなさんと
  - Ver0.5にて多くの助言をいただいた清水建設の安井さん、野田さん、武井さん
  - Ver0.6にてアクティビティと成果物およびロールの対応表を作成していただいた、東洋エンジニアリング依田さん
  - そして様々なアドバイスをいただいた要求開発アライアンスのみなさん

  – これからも、Openthologyは業界オープンな要求開発方法論として、要求開発アライアンスメンバーのナレッジを結集させつつ、発展することを願っております。どうぞこれからも積極的な参加をお願いします。

OPENTHOLOGY

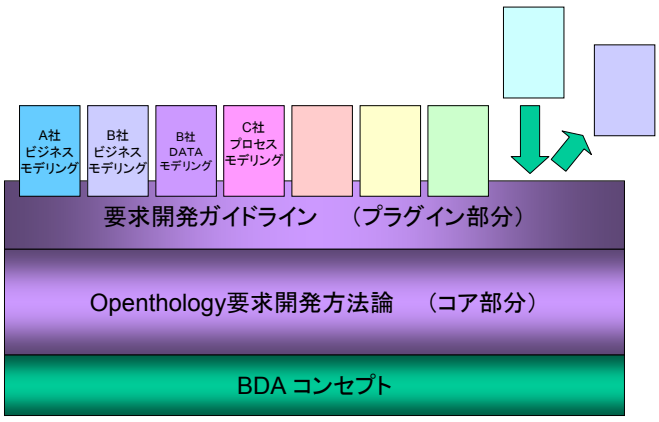
Openthology要求開発方法論の全体像



OPENTHOLOGY

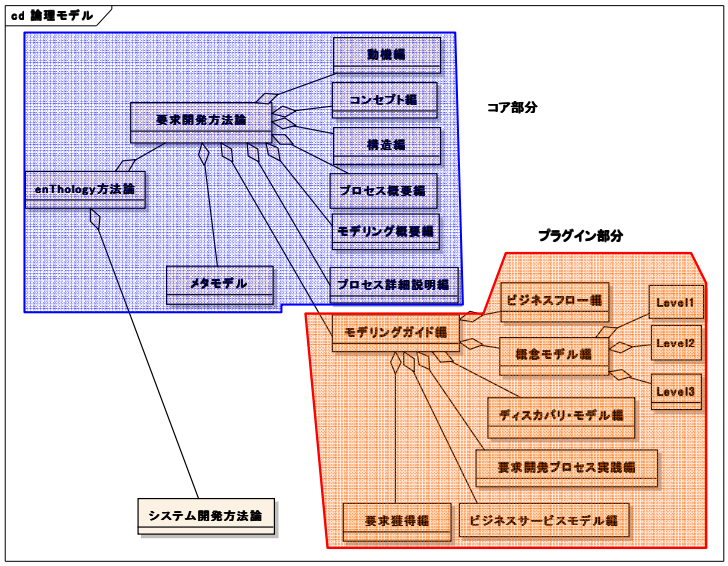
Openthology要求開発方法論 コアとプラグイン構成現構成

ガイドライン部分だけがプラグインできる



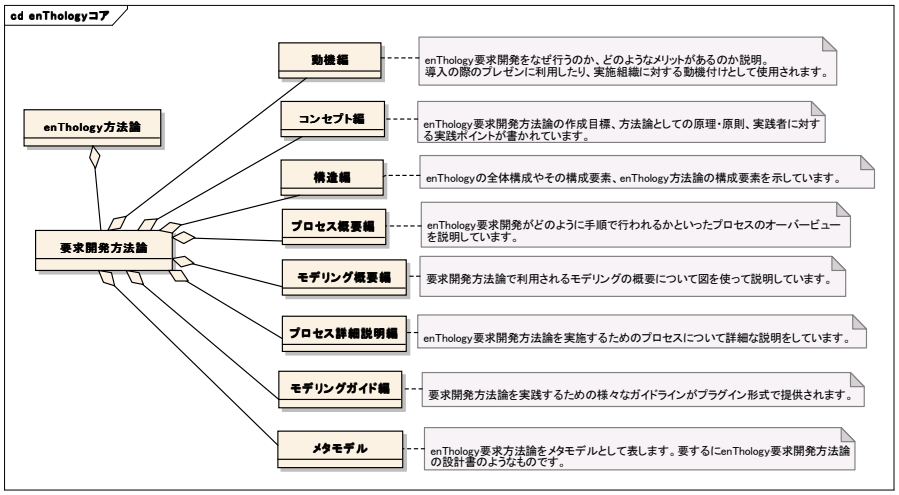
OPENTHOLOGY

Openthology要求開発方法論の全体像　コアとプラグイン（現在の案）



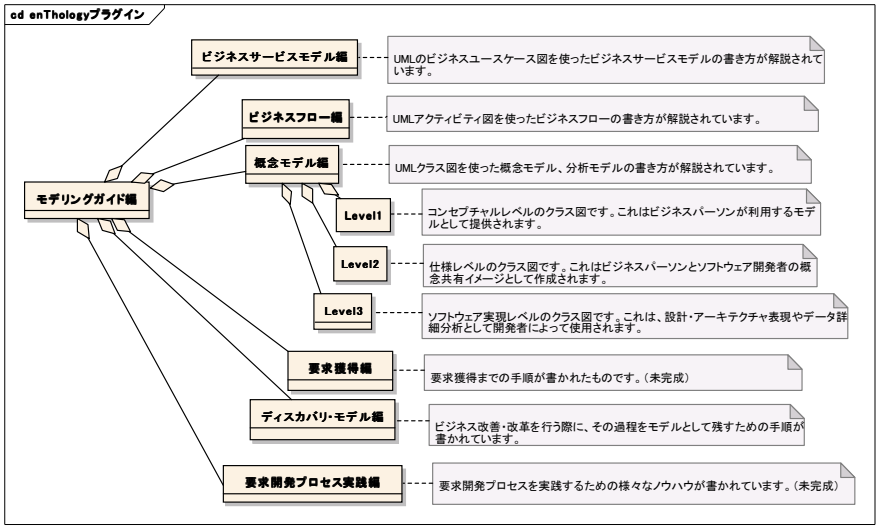
OPENTHOLOGY

Openthologyコア部分の説明



OPENTHOLOGY

Openthology　プラグイン



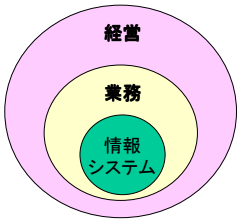
OPENTHOLOGY

動機編

OPENTHOLOGY

業務抜きのIT化はありえない

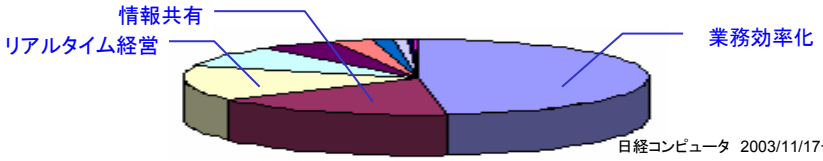
- 業務はトータルコーディネーション
  - 業務は、人、組織、装置、設備、情報システム、パートナなどのコラボレーションで構成される
  - 人間系と情報システムは互換性があるが、特性が違う（得意不得意がある）
  - 人間系の設計と情報システムの設計を片側ずつやっては最適化できない
- IT化は、経営課題の解決を上位目的としており、業務構造や業務プロセスの設計、再設計の文脈で企画される
  - 業務の設計から生まれるITの設計
  - 業務の見直しから生まれる情報システムの見直し



OPENTHOLOGY

今情報化投資のROIが注視されている

- 2000年以降、企業の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」

OPENTHOLOGY

どうIT化すればいいかわからない

業務効率化をはかる一般企業

IT化投資額は、**年間15兆円**

至るところに使えないシステムが・・・・・・

OPENTHOLOGY

ROIをクリアしないシステムの例

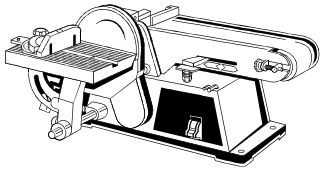
- 人の作業をただ電子化したようなシステム
  - やたら画面数が多く、複雑、拡張性がない
- 全体効率化に寄与しないシステム
  - ローカルなユーザの要求で作られたシステム
- 既存のシステムをストレートにリプレースしたシステム
  - 従来の人系とコンピュータ系の境界が固定化
  - システムに合わせた業務フロー
- 業務が変わってもどう変えればいいかわからないシステム
  - システム化されたときの意図が不明
  - 業務との対応関係が管理されていない

ビジネスドリブンでないシステム化

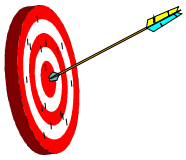
OPENTHOLOGY

ROIを高めるために

- 「システム開発」のコスト削減
  - ➡ 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、オフショア開発）



- 経営課題解決に合致するシステムのプランニング
  - ➡ 何を作るべきか
    - 業務に合わせたシステム要求
    - 適正なシステム化範囲
    - 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

OPENTHOLOGY

プロジェクト失敗の要因は、企画・要件定義

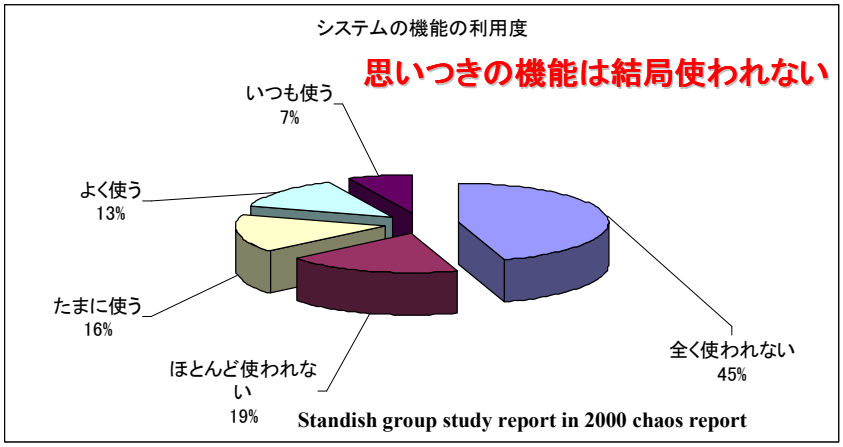
- 国内のシステム開発プロジェクトの成功率
  26.7%
- プロジェクトの遅れの原因
  - 要件定義が計画より長引いた…… 37.7%
  - 企画作業が長引いた …… 22.7%
- システム品質問題の原因
  - 要件定義が不十分 …… 35.9%
  - システムの企画が不十分 …… 18.7%
- 稼動システムの満足度
  - 「計画通り利用しているが不満足」‥ 24.3%

稼動しても役に立たない

日経コンピュータ 2003/11/17号より引用

OPENTHOLOGY

使われない、使えないシステム

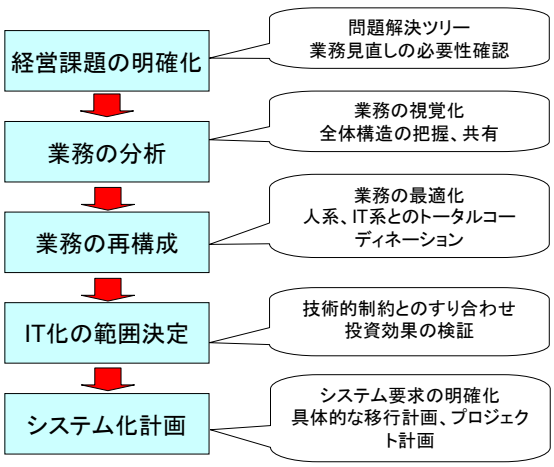


要求は業務からロジカルに導かれなければならない

OPENTHOLOGY

業務からロジカルにシステムの要求へ

- 解決すべき経営課題の設定が基本
- 課題解決のために業務から見直す
- ITは業務の構成要素として、総合的に見直す
- 結果として、ITがなすべきところが決まる



OPENTHOLOGY

経営とITを融合させる必要性

- ビジネスゴールに直結したシステム化をはかる
  - Business Driven（業務の視点から）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のリンクが管理されている
  - 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

トレーサビリティが確保され、管理されている

OPENTHOLOGY

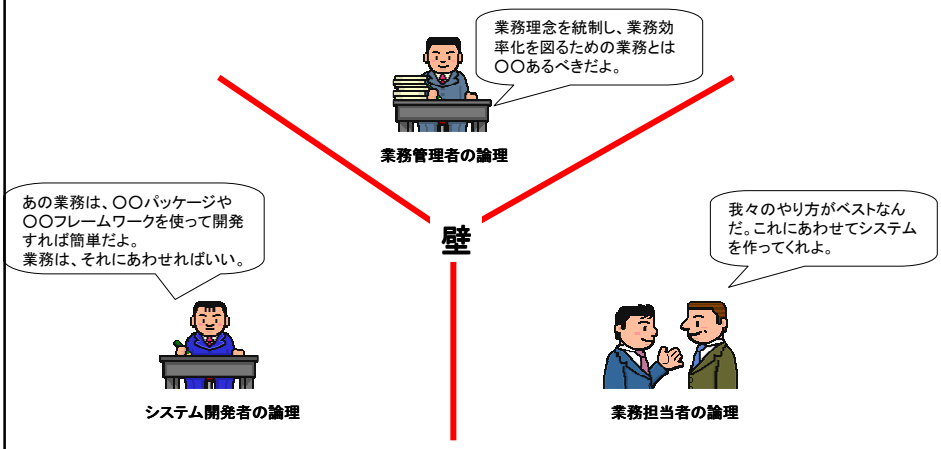
要求は「開発」するもの

- 「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
- ユーザからヒアリングした要求の実現が業務効率化に結びつくとは限らない。
  - ユーザの理解の範囲内で生まれた属人的なもの
  - 直感的、場当たり的であることが多い
- 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。

OPENTHOLOGY

なぜよいシステム開発ができないのか？

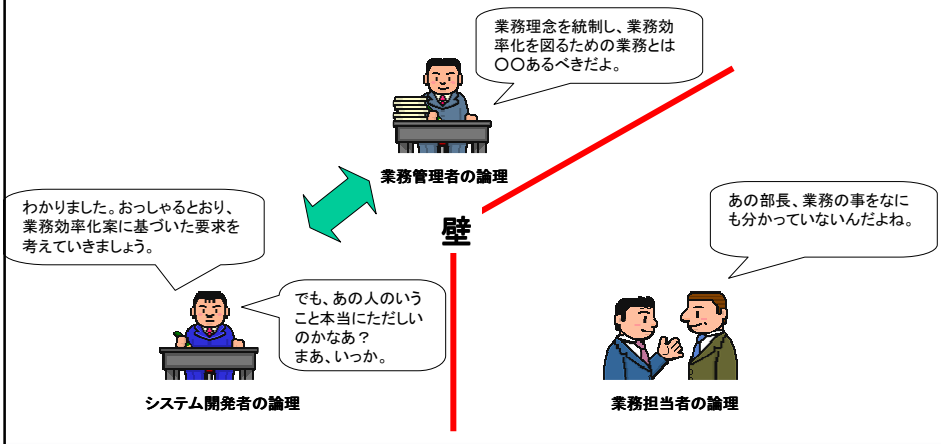
– 最悪のシナリオ



OPENTHOLOGY

それは、正しいシステム要求がだせないから

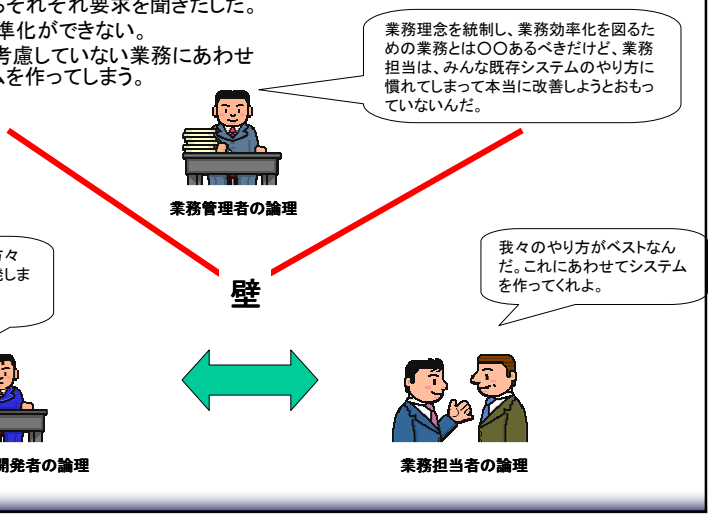
- **うまくいきそうで駄目なシナリオ（その1）**
  - トップ・業務管理者の考えどおりのシステムを作ったが、業務担当から大クレーム。
  - 結局使われないシステムになった。



OPENTHOLOGY

それは、正しいシステム要求がだせないから

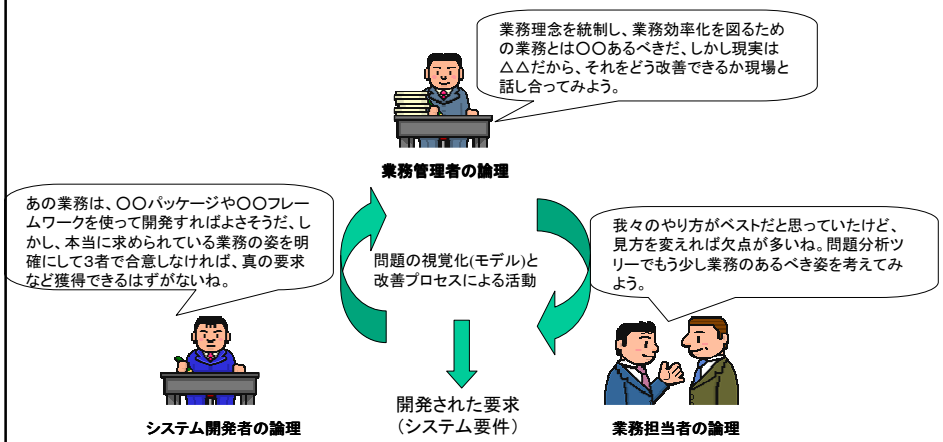
- **うまくいきそうで駄目なシナリオ（その2）**
  - 業務担当者からそれぞれ要求を聞きだした。
    - » 業務の標準化ができない。
    - » 効率化を考慮していない業務にあわせてシステムを作ってしまう。



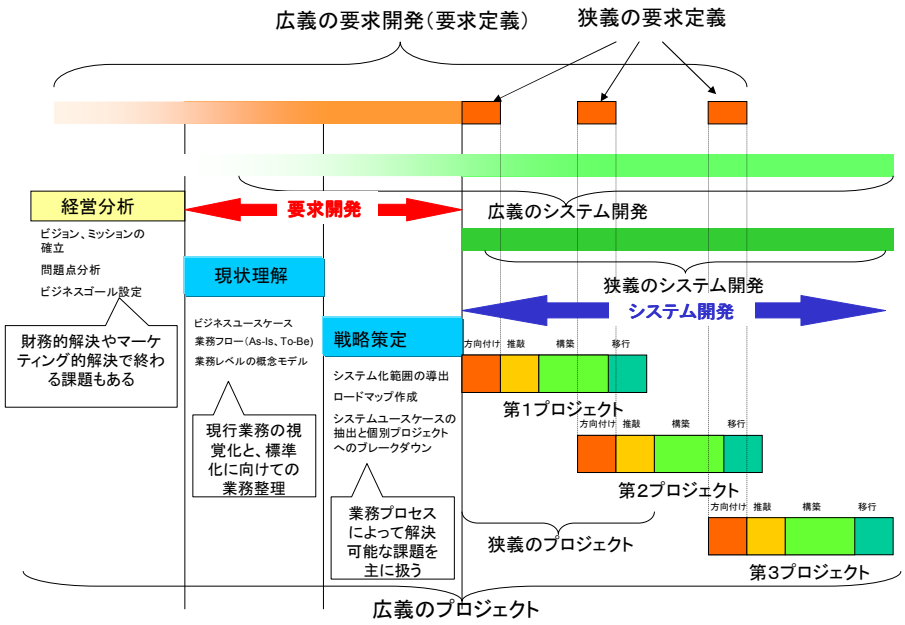
OPENTHOLOGY

要求開発とは

– 正しい要求の獲得と合意のための活動

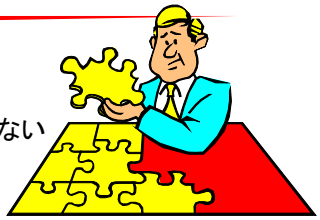


プロジェクトの全体像と要求開発

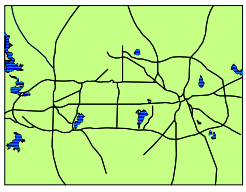


OPENTHOLOGY

業務がわからずに要求は作れない



- システム化対象は、より広範、より複雑に
  - SCM、企業・グループ統合、企業ポータル・・・
  - 全体の構造や流れが担当者の視野ではつかめない
  - 多くの立場のメンバーが関わっている
- 業務はトータルコーディネーション
  - 人、組織、装置、設備、情報システムなどが連携して業務が実現される
  - 全体が見えないと、情報システムが担うところや与える影響を見切れない
  - つなぎ合わせてみないと全体はわからない
- 全体の地図、今の構造、先の構造、途中の構造
  - As-is、To-Be、Realistic、Transient
  - 現実的なロードマップにしたがってRealisticに移行
  - 既存システム、新規システム、パッケージなど、トータルでコーディネーション



業務の視覚化はグラウンドデザインに必須

OPENTHOLOGY

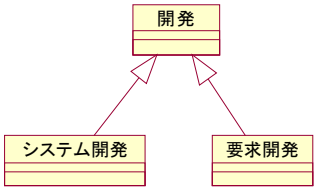
要求開発とは

- 経営上の目的を果たすために、業務の設計、あるいは再設計を行う中で、ビジネス上の要求と情報システムが担うべき要求を導き、定義する活動
  - 業務構造の視覚化、業務分析、業務設計、IT化のスコープ決定、システム化計画立案などの作業を含む
  - 要求開発のアウトプットが、後続のシステム開発プロジェクトのインプットとなる

OPENTHOLOGY

「開発」とよぶ理由

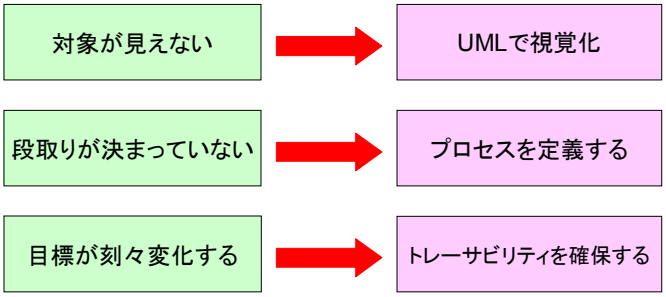
- 要求はもともとあるものではなく、業務を元に開発されるものである
- 「システム開発」と「要求開発」を同列に表すことができる
- システム開発と要求開発はどちらも「開発」なのでアナロジーで語ることができる。システム開発で培った技術体系が利用できる。
  - 開発プロセス
  - 開発方法論
  - モデル
  - ツール



OPENTHOLOGY

システム開発と要求開発のアナロジー

- 対処方法： 業務もソフトウエ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



OPENTHOLOGY

現場は業務改革の駆動力を持っている

- 現場は地上戦でホフク前進している
  - 自分の目の前しか見えていない
  - とりあえず個々の問題で手一杯
- 見えれば変わる
  - 違った視点で見れるようになれば全貌がわかる
  - 今どうなっているか、次にどうすればいいかわかれば、自分で変わる
- つかめれば合意できる
  - 全体構造が共通認識されれば、他の部署の立場がわかる
  - 互いの落としどころを理解できる
- 意欲がないわけではない
  - 全体が見えれば、どうあるべきかはわかる
  - どうすればいいかがわかれば、かわりたがる

視覚化の手立てと進め方があれば、ユーザー主導でことは進む

OPENTHOLOGY

要求開発方法論導入による効果

- 課題解決に直結するシステムの実現
  - 業務構造の確実な把握
  - 業務改善を合わせたシステム化
  - 業務から導かれたシステム要求
- 試行錯誤のないリズミカルなIT化のプランニング
  - 行うべきこと、手順が決まっている
  - 成果物が規定されている
  - 体制が決まっている
  - 必要な観点を一通り網羅して抜けのないプランへ
- システム開発以前に投資効果を検証
  - 業務のシミュレーション
  - ROIの試算
  - プロトタイピング

OPENTHOLOGY

コンセプト編

OPENTHOLOGY

Openthology方法論開発における目標

- ビジネスからITへと繋げる方法論を持つ
- 全てのフェーズでモデリングによる可視化を重視する
- 各フェーズの作業内容を明確にする
- 各フェーズの作業に関する目的・動機が明確であること
- フェーズ間のモデル遷移を明確にする
- ドメインの特徴に合わせた方法を選択可能
- メタモデルによる自己説明できる
- 分かりやすくシンプルである
- コンセプト重視：導入時にテーラリングがしやすい

OPENTHOLOGY

BDA宣言

- ITの視覚化はビジネスの視覚化から始める。
- IT要求はビジネス要求の中から獲得。
- ITはビジネスの手段であり，ITそのものに価値はない
- ITの可視化はモデルによるビジネスの可視化から始まる
- ビジネスとITの健全さは継続的な改善による。
- ビジネスとITをモデルによって視覚化・共有化する。

BDA：Business Driven Architecture™

「BDA™」は基本的に豆蔵にて商標登録していますが、これはあくまで、業界の中で末永く自由（オープン）にこの言葉を使っていただくため商標として仮押さえしたものです。ということですので、「Java™」のように、自社製品のような使い方をしない限り、自由に使っていただけます。

OPENTHOLOGY

Openthology
5つのプリンシプル

1. **ビジネス主導によるIT化**
   - 業務の姿にITを合わせること
2. **効果検証型プロセスの導入**
   - 最低1ヶ月に1回はモデリング効果を検証する
3. **検証可能なビジネスモデル**
   - 概念モデルをビジネスフローで検証する
   - ビジネスフローをプロトタイプで検証する
4. **ビジネス担当主導のビジネスモデリング重視**
   - 現場のビジネスナレッジを重視したボトムアップアプローチ
5. **フレキシブル・ビジネスモデリング**
   - スピーディかつ状況に応じてビジネスモデリングを行う

OPENTHOLOGY

Openthology
7つのプラクティス

- **勇気**
  - 問題を発言する勇気を持とう
- **オープン**
  - 業務問題点をオープンにする環境と人を形成
- **成功は失敗から**
  - 失敗を隠さない。業務問題を抱えていた部署が新たな改善策を提案する文化を築こう
- **スピーディなビジネス改善と公開**
  - 改善できるものは即改善、改善したら即公開
- **目的を理解したモデリング**
  - ビジネスモデリングはモデリングの目的を理解して初めて実践できる
- **モデリングの価値**
  - モデリングによる視覚化・共有化こそが、ビジネス改善の第一歩
- **ペアモデリング**
  - 常にモデルの共有化を図り、最低でもペアでモデリングすること

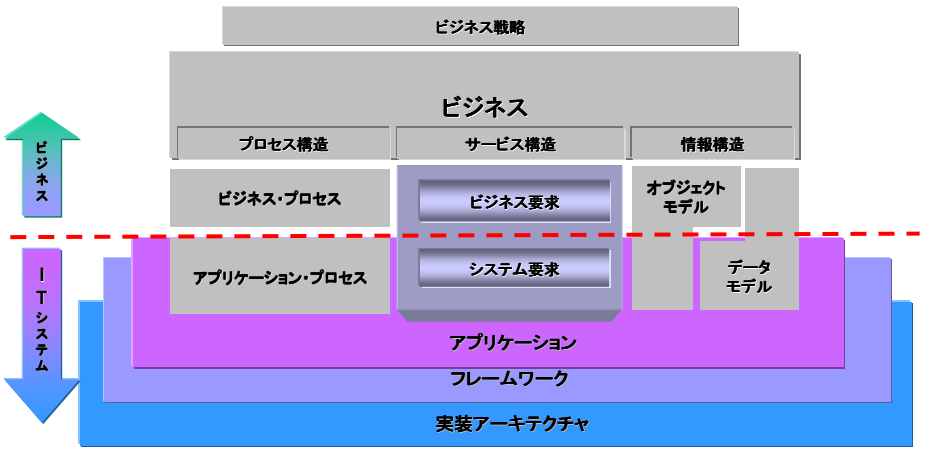
OPENTHOLOGY

構造編

enThology

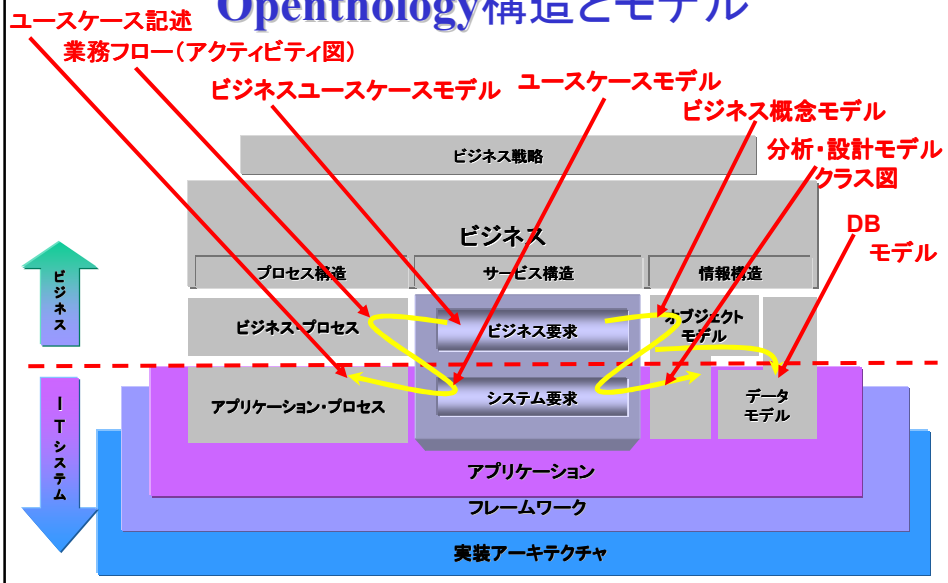
OPENTHOLOGY

Openthology構造



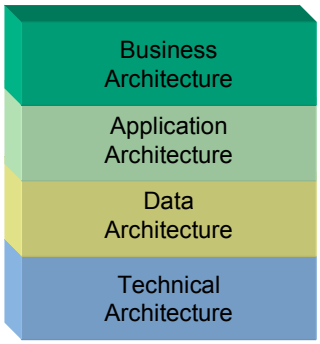
OPENTHOLOGY

Openthology構造とモデル



OPENTHOLOGY

参考（EA構造）



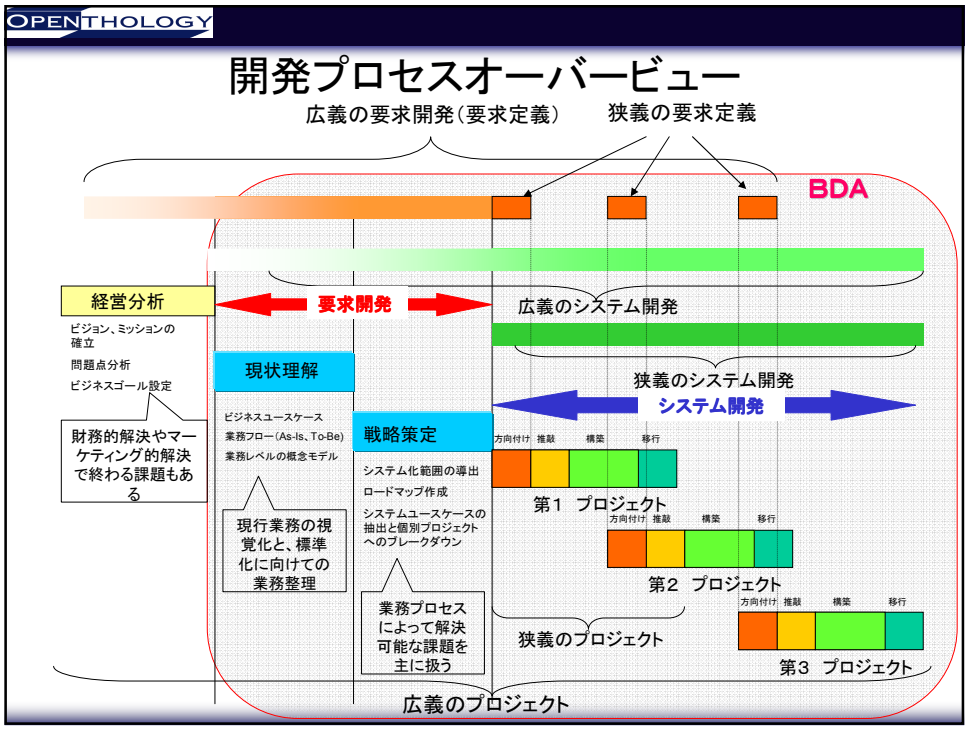
OPENTHOLOGY

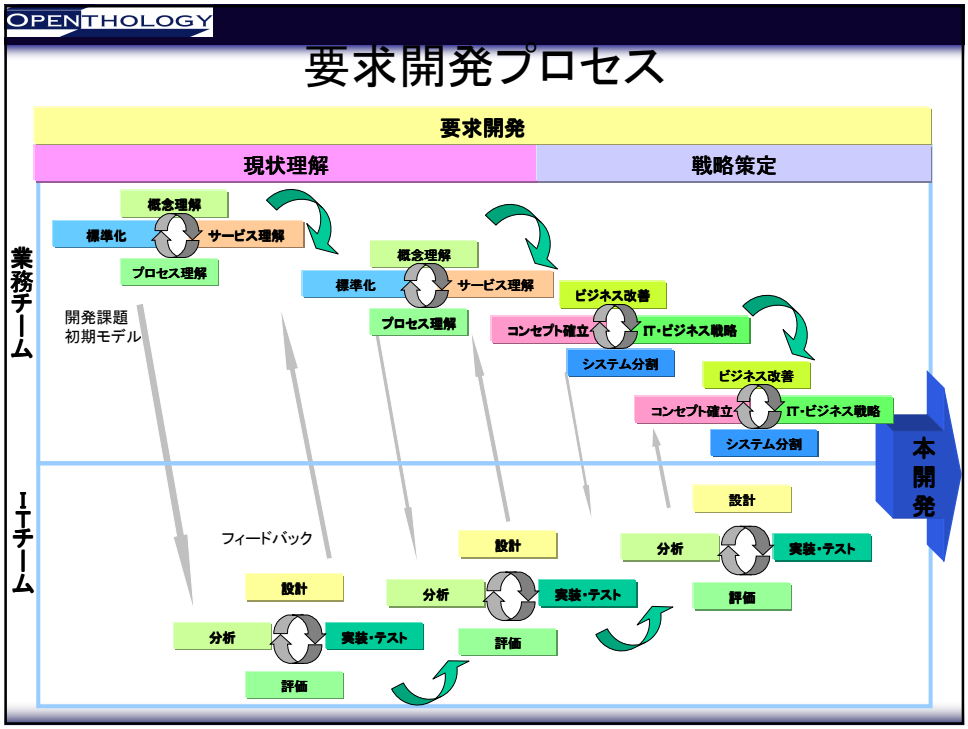
Openthology構成要素の説明

- メソッド(Openthology Method)
  - UMLベースのプロセスとモデルを含んだ要求開発、システム開発の方法論を提供
- プロセス(Openthology Process)
  - 要求開発からシステム開発までの一貫したフェーズ組立戦略や開発ドキュメントセットを提供
- モデル（Openthology Model）
  - フェーズの中で使用するUMLベースのモデルを定義
- 表記法(Notation　UML+α)
  - ビジネスITの視覚化のための表記法をUMLベースで提供。ビジネスモデリングの中ではUML+αを表記法として提供
- フレームワーク/ツール(Framework/Tools)
  - システム開発に利用できる標準的なフレームワークとツールを提供

OPENTHOLOGY

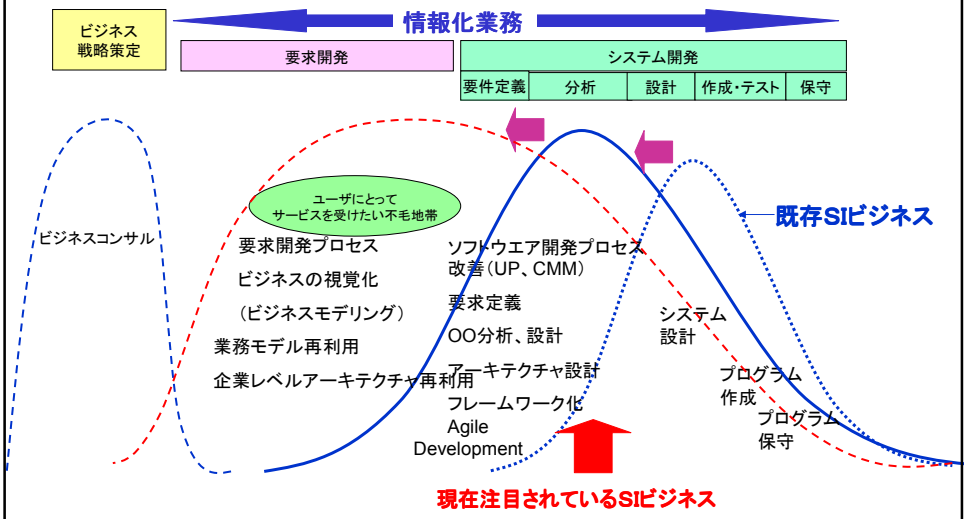
プロセス概要編


OPENTHOLOGY
開発プロセスオーバービュー
広義の要求開発(要求定義)
狭義の要求定義
BDA
経営分析
ビジョン、ミッションの確立
問題点分析
ビジネスゴール設定
財務的解決やマーケティング的解決で終わる課題もある
要求開発
広義のシステム開発
現状理解
ビジネスユースケース
業務フロー(As-Is、To-Be)
業務レベルの概念モデル
現行業務の視覚化と、標準化に向けての業務整理
戦略策定
システム化範囲の導出
ロードマップ作成
システムユースケースの抽出と個別プロジェクトへのブレークダウン
業務プロセスによって解決可能な課題を主に扱う
狭義のシステム開発
システム開発
方向付け
推敲
構築
移行
第1 プロジェクト
第2 プロジェクト
第3 プロジェクト
狭義のプロジェクト
広義のプロジェクト


OPENTHOLOGY
要求開発プロセス
要求開発
現状理解
戦略策定
業務チーム
ITチーム
概念理解
標準化
サービス理解
プロセス理解
開発課題
初期モデル
ビジネス改善
コンセプト確立
IT・ビジネス戦略
システム分割
本開発
フィードバック
設計
分析
実装・テスト
評価

OPENTHOLOGY

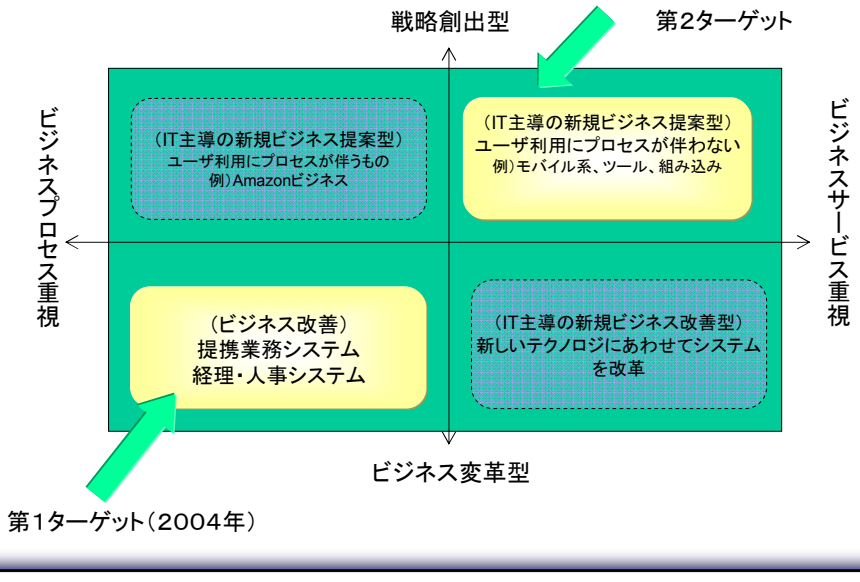
変化するビジネスコア



システム開発への実践的工学技術の適用

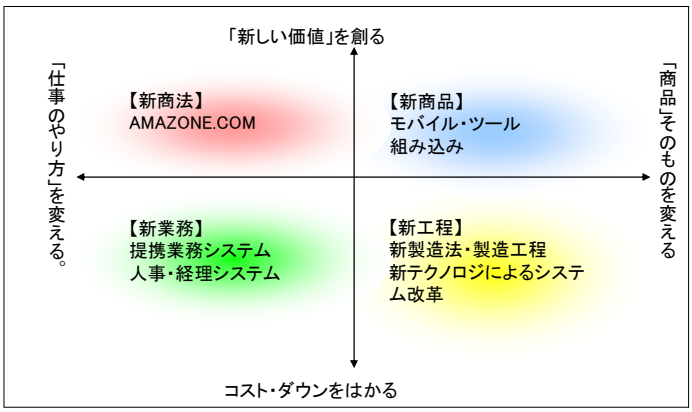
OPENTHOLOGY

開発タイプの分類



OPENTHOLOGY

開発タイプの分類



OPENTHOLOGY

開発タイプにあわせた プロセスとメソッドの基本

1. 戦略創出型

新規ビジネスを立ち上げる時、いままでのビジネスを参考にすることができないタイプ。
一般的にこのタイプはITをビジネスアイデアのコアとすることが多い。
このタイプでは**現状理解フェーズの作業内容は、市場・ニーズ分析**に限定されることになる。要求開発は、**IT化戦略分析フェーズ重視**で進む事になる。

2. ビジネス変革型

既存の業務のコアコンピタンスを時代に合わせて再定義し、業務を変革することで企業価値を高める活動の延長で、ITの変革を行う。
このタイプでは現状分析フェーズの作業内容は、**現状業務の理解、標準化**を行う。そしてIT化戦略分析フェーズでは、現状からビジネス全体最適された**理想の姿への変革がテーマ**となる。

3. ビジネスプロセス重視

ビジネスを行う上でプロセス構造が重視されるもの。一般的にこの領域だけに存在するビジネスは少なくビジネスサービスを踏まえた上で業務フローを中心とした**ビジネスプロセス寄りのモデル化**が必要とされる。

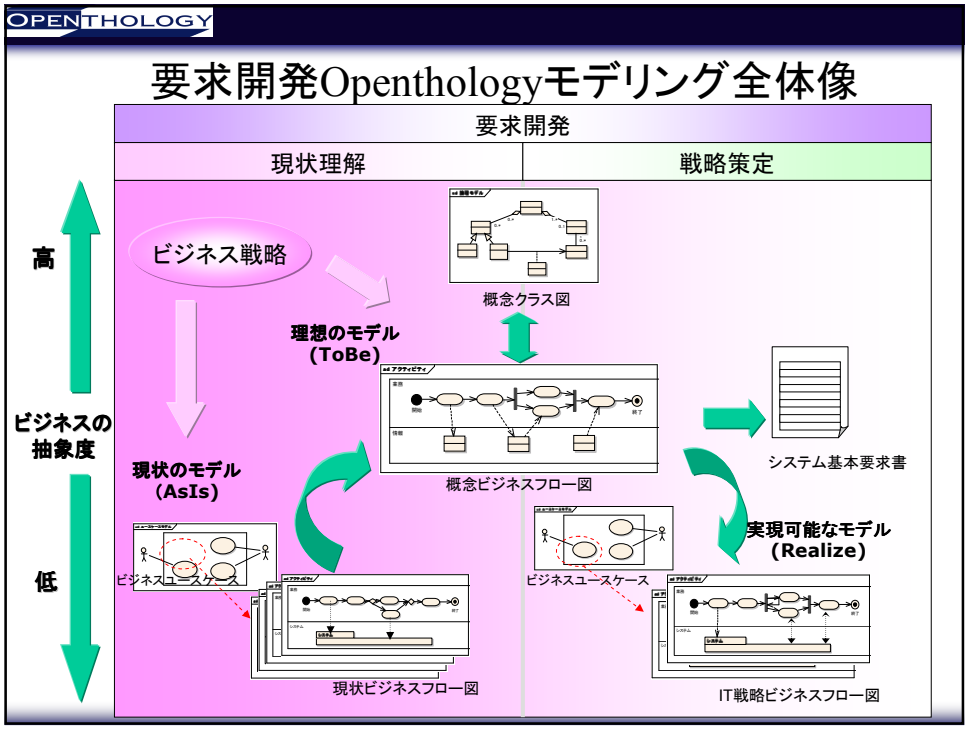
4. ビジネスサービス重視

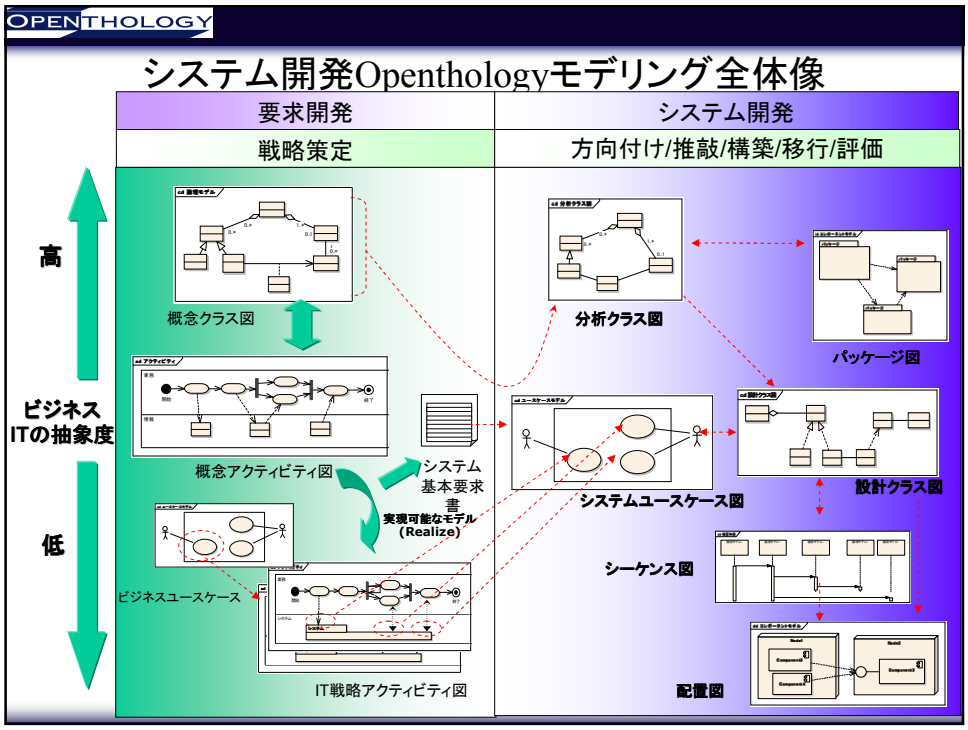
ビジネスを行う上で、ビジネス実行者のプロセスを限定しないもの。つまりサービスを提供するがその使い方の手順を限定しないもの。**コンシューマー向けサービスなどが対象となる**。この場合、業務フローなどのモデル化が意味を成さないケースが多く、**ビジネスユースケースを中心としたサービス構造**を捉えるためのモデル化が必要とされる。

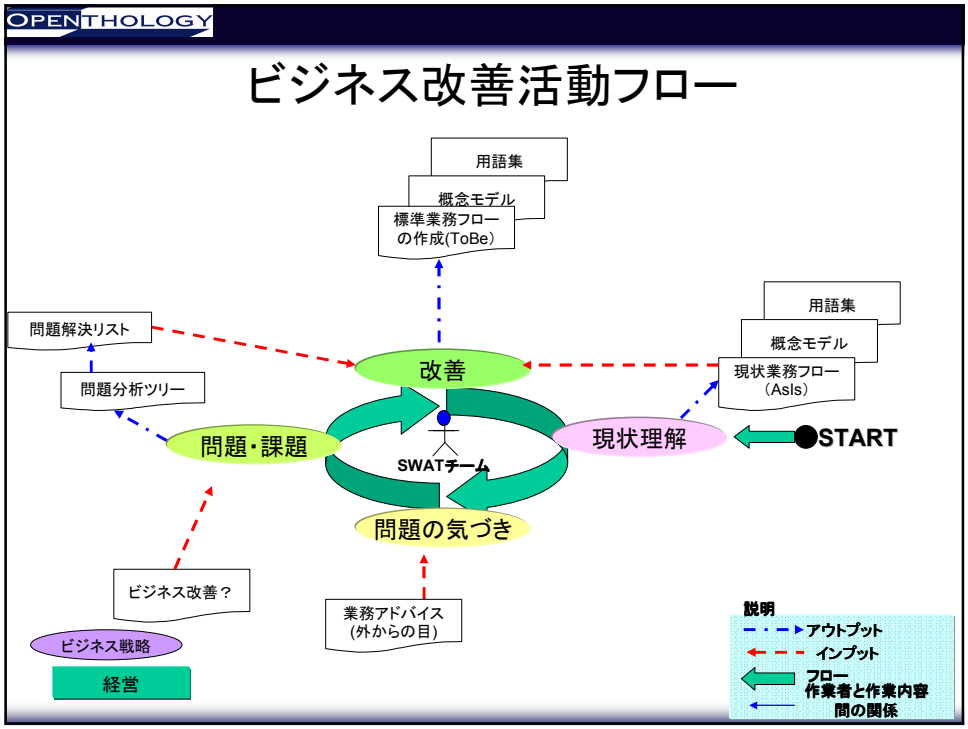
OPENTHOLOGY

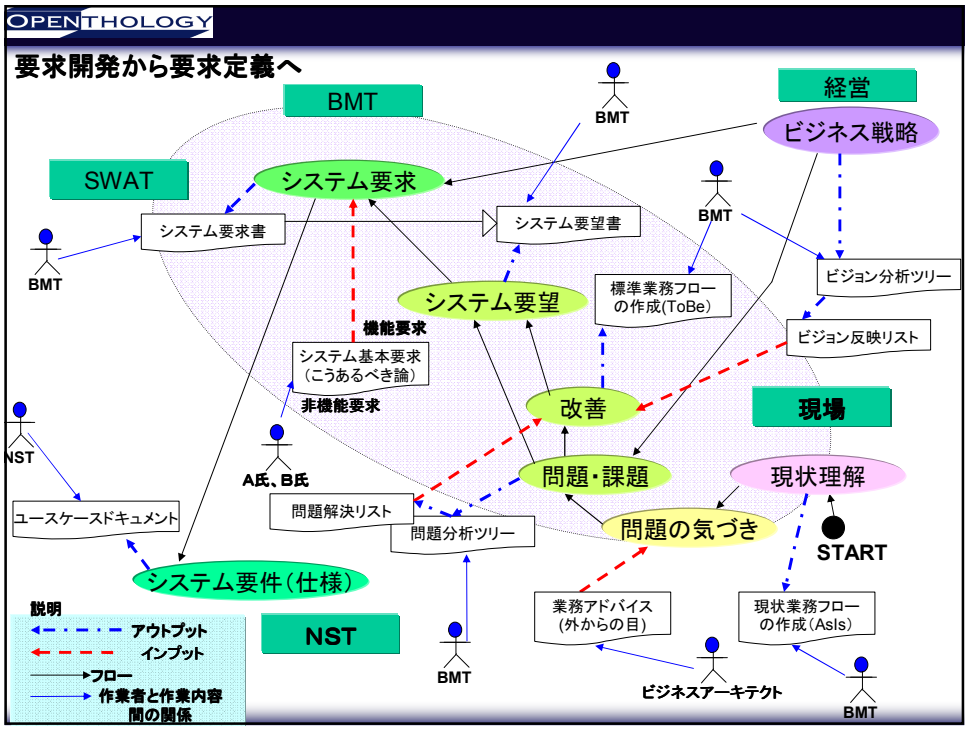
モデリング概要編

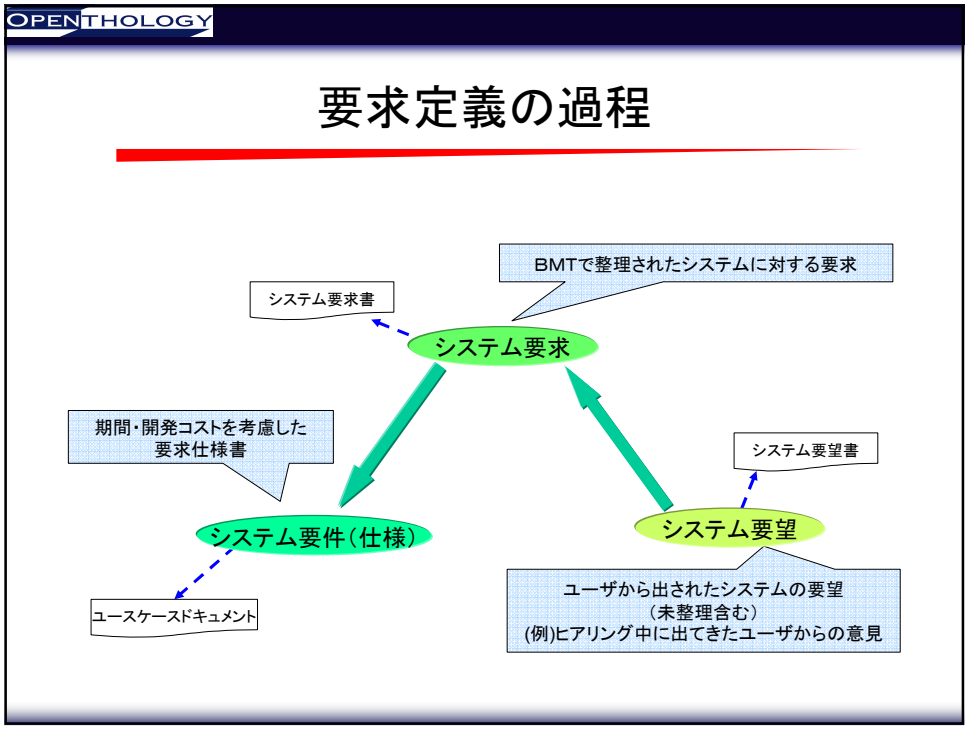
enThology




OPENTHOLOGY
システム開発Openthologyモデリング全体像
要求開発
システム開発
戦略策定
方向付け/推敲/構築/移行/評価
高
ビジネス
ITの抽象度
低
概念クラス図
概念アクティビティ図
システム
基本要求
書
実現可能なモデル
(Realize)
ビジネスユースケース
IT戦略アクティビティ図
分析クラス図
パッケージ図
システムユースケース図
設計クラス図
シーケンス図
配置図


OPENTHOLOGY
ビジネス改善活動フロー
用語集
概念モデル
標準業務フローの作成(ToBe)
問題解決リスト
問題分析ツリー
改善
問題・課題
SWATチーム
現状理解
START
用語集
概念モデル
現状業務フロー(AsIs)
問題の気づき
ビジネス改善？
業務アドバイス(外からの目)
ビジネス戦略
経営
説明
アウトプット
インプット
フロー
作業者と作業内容
間の関係


OPENTHOLOGY
要求開発から要求定義へ
BMT
BMT
経営
ビジネス戦略
SWAT
システム要求
システム要求書
システム要望書
BMT
BMT
ビジョン分析ツリー
システム要望
標準業務フローの作成(ToBe)
機能要求
ビジョン反映リスト
システム基本要求（こうあるべき論）
非機能要求
改善
現場
NST
A氏、B氏
問題・課題
現状理解
ユースケースドキュメント
問題解決リスト
問題分析ツリー
問題の気づき
START
システム要件（仕様）
説明
アウトプット
インプット
フロー
作業者と作業内容間の関係
NST
BMT
業務アドバイス（外からの目）
現状業務フローの作成（AsIs）
ビジネスアーキテクト
BMT

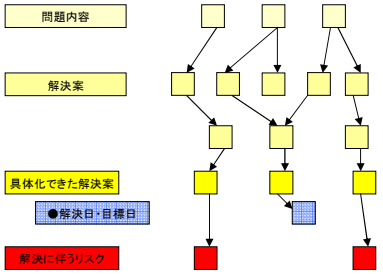

OPENTHOLOGY
要求定義の過程
BMTで整理されたシステムに対する要求
システム要求書
システム要求
期間・開発コストを考慮した要求仕様書
システム要望書
システム要件（仕様）
システム要望
ユースケースドキュメント
ユーザから出されたシステムの要望
（未整理含む）
(例)ヒアリング中に出てきたユーザからの意見

OPENTHOLOGY

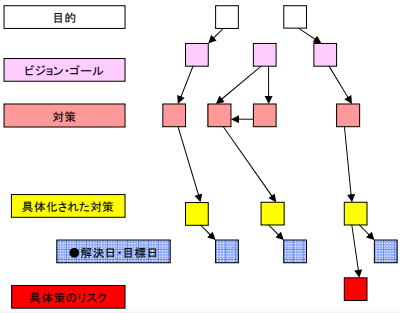
ビジネス改善アプローチ

トップと現場からの声をビジネス構造に反映するために
準備： 15cm四方の磁気カード、ホワイトボード
方法： ビジネス改善チームメンバーが話し合いながら下記のツリーを作成する。

■ボトムアップ（問題分析ツリー）
ビジネス遂行者主導のビジネス改善



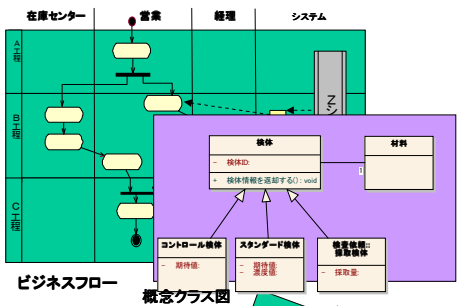
■トップダウン（ビジョン分析ツリー）
経営層からの要望・ビジネスビジョン・戦略の反映



OPENTHOLOGY

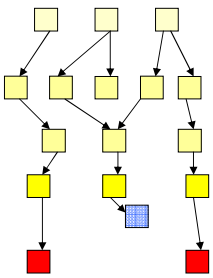
ビジネス改善アプローチ

■面の分析



ビジネスフロー　概念クラス図

■点の分析



■点と面の融合

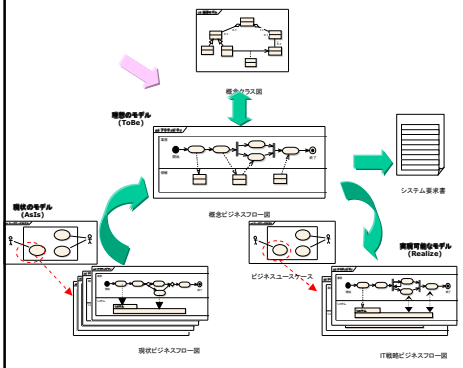
●ビジネスフロー図
・問題が解決した後の姿がビジネスフローとして表現されているか
・問題解決ポリシーがビジネスフローに反映されているか
・問題解決後の具体的なシナリオがビジネスシナリオに記述されているか
●概念クラス図
問題解決に必要とされる新たな概念構造が概念クラス図に表現されているか

OPENTHOLOGY

ビジネス改善モデリング

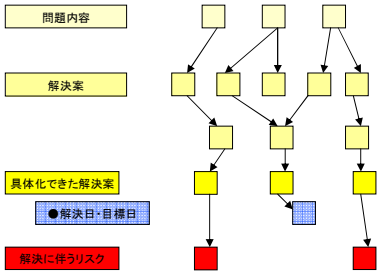
スケッチ・モデル（UML）

現状理解・戦略理解・開発モデルと、それぞれの作業過程の視覚化・共有化のための写像。写像した図から新たな問題や発見がある。



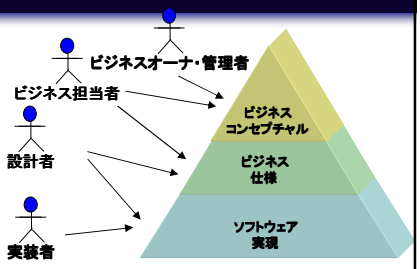
ディスカバリ・モデル

現状理解から戦略理解へ、どのように変えるのかといった発見過程を捉え、記録しておくモデル。



OPENTHOLOGY

情報構造モデル（クラス図）



- **3段階に抽象度を分類**
  - なぜ
    - **ユーザナレッジを無理なく吸収できる**
    - **コンセプチャルレベルを持つことで、概念構造を分かりやすいレベルで視覚化・合意できる**
  - **Level1**
    - レベル…コンセプチャル
    - 用途　…現状分析モデル
    - 説明
      - 汎化、関連、集約などの関係を使って用語語彙の関係意味構造を表したもの。
  - **Level2**
    - レベル…仕様
    - 用途　…現状分析モデル、戦略分析モデル、システム分析モデル
    - 説明
      - クラスの責務を表現するために必要とされるキーや主な属性を割り付ける。また、責務を表す操作も微量ながら割り付ける。
  - **Level3**
    - レベル…実現
    - 用途　…システム設計モデル
    - 説明
      - データモデル
        - » データベースの論理設計につなげるため、主キー、副キーの明確化、すべての属性の洗い出しを行う。
      - 設計モデル
        - » 属性と操作について、実装アーキテクチャを意識した構造として割り付ける。また、実装アーキテクチャを意識した最適なクラス構造に変更する。

OPENTHOLOGY

OpenThology方法論
メタモデル・概念モデル

フェィズ

- 要求開発ステージ
  - 現状理解
  - 戦略策定
- システム開発ステージ（RUPの場合）
  - 方向付け
  - 推敲
  - 作成
  - 移行

OPENTHOLOGY

ディシプリン

- 要求開発
  - 計画
  - 現状分析
  - ビジネス改善
  - IT戦略設計
  - システム計画
- システム開発
  - 分析
  - アーキテクチャ設計
  - 設計
  - 実装
  - テスト

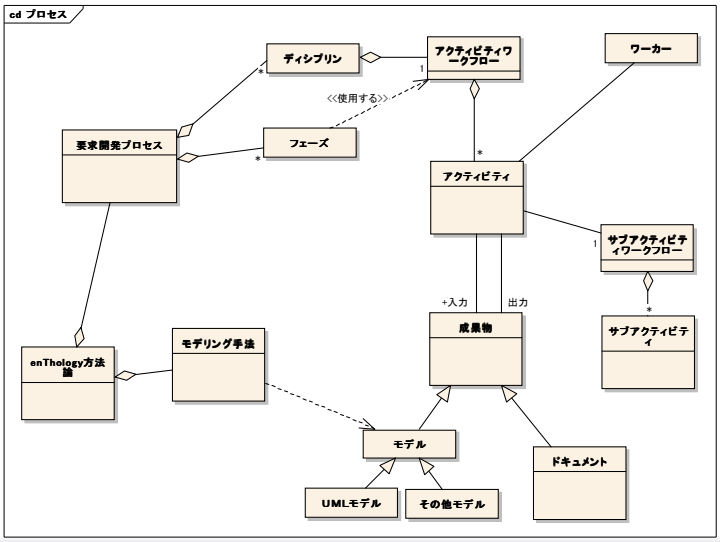
OPENTHOLOGY

モデル

- 要求開発
  - 現状分析モデル
  - 戦略分析モデル
- システム開発
  - ユースケース（要求）モデル
  - 分析モデル
  - アーキテクチャ設計モデル
  - 設計モデル
  - 実装モデル

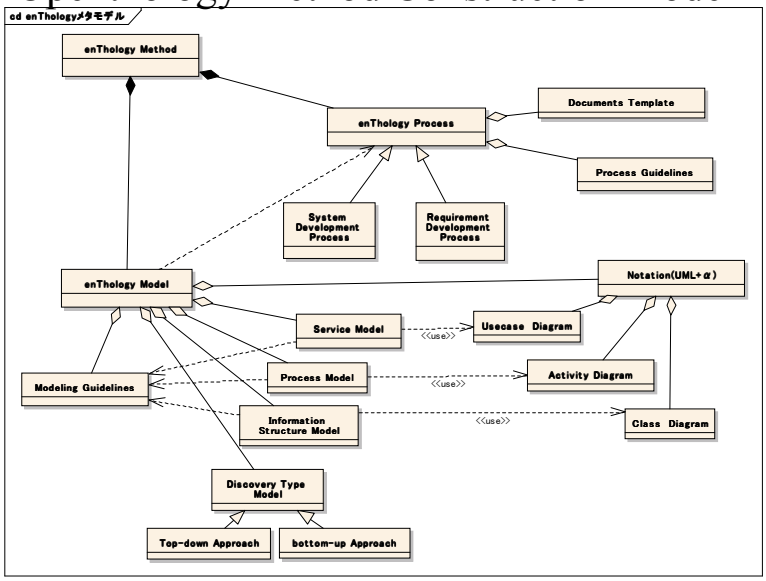
OPENTHOLOGY

Openthologyプロセス　メタモデル



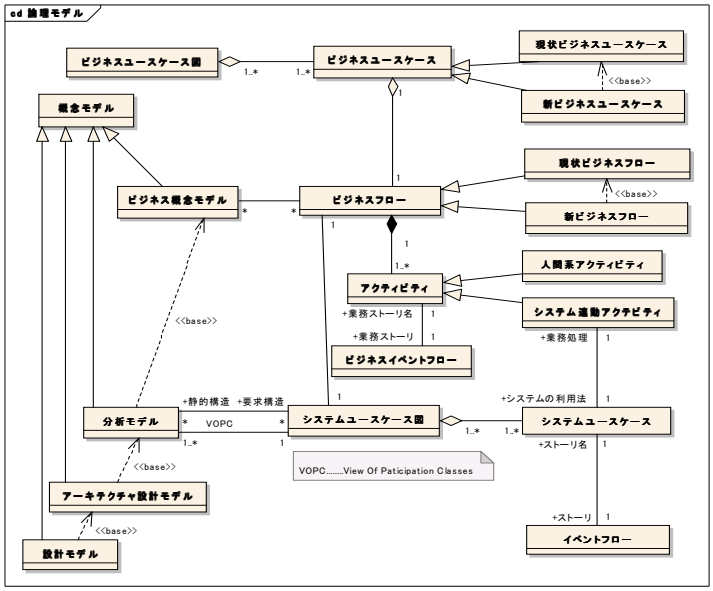
OPENTHOLOGY

Openthology Method Construction Model



OPENTHOLOGY

Openthology Method　model (Modeling Layer)



OPENTHOLOGY

終わり