

要求開発アライアンス: Requirement Development Alliance

<http://www.openthology.org>

Openthology Version 1.0 の文書は [クリエイティブ・コモンズ-帰属-2.5 \(CC-by-2.5\)](https://creativecommons.org/licenses/by/2.5/) で提供され、営利目的で利用することが可能です。

Openthology Ver0.6 から Ver1.0 へ ～何が変わったのか～

Version 1.0 β1

■このファイルについて

本ファイルは、要求開発アライアンスの策定する要求開発方法論 Openthology ドキュメントの一部です。最新のファイルは、<http://www.openthology.org> より取得することができます。

■フィードバック

要求開発方法論 Openthology は常に利用者のフィードバックを受け付けます。誤り・改善点・アイデア・コメントは、info@openthology.org まで気軽にお寄せください。

■要求開発アライアンスについて

2005年3月発足。企業や組織のIT化についての共通課題を、業務の可視化によって解決することをテーマとして結成した団体で、ユーザ企業やシステム開発関連企業あわせて50社以上、170名以上が参画しています。目下の活動は、要求開発方法論 Openthology の策定であり、Openthology のドキュメントは、要求開発アライアンスのホームページ(<http://www.openthology.org>)からダウンロードできます。Openthology は、ソフトウェア開発が始まるまでに行なうべきシステム要求の作成過程の進め方、手法、成果物などを規定したもので、現在もバージョンアップを進めています。

1. 概要

この文書は、要求開発方法論 Openthology Version 1.0 と、2005 年 1 月に公開された、Version 0.6 の違いについて説明します。すでに Version 0.6 を知っている読者にとっては、変更点を把握し、Version 1.0 の理解を容易にすることができます。また、本文書はできるだけ「なぜ変更したのか(変更の背景)」を説明しようとしています。変更の背景を知る事により、Version 1.0 の理解の一助になれば幸いです。



Openthology Version 1.0 で加えた変更点は、大きく二つに分類できます。

「理論から実践へ」

Openthology Version 0.6 は、要求開発における方法論の基礎であり、「なぜ要求開発が必要なのか(WHY)」「要求開発では何をすべきなのか(WHAT)」が論点の中心でした。しかし、この時点では、実際のプロジェクトに適用する際に「いつ(WHEN)」「誰が(WHO)」「どうやるのか(HOW)」については詳細が未確定であり、誰もが容易に要求開発チームを組成したり、プロジェクトに適用できるものではありませんでした。

そこで、Openthology Version 1.0 では、「フェーズ分割」「仮説検証型 PDCA サイクルの導入」「PDCA プロセスキャビネット」「プロセスセル」などの概念を導入することによって、実践可能な方法論として具体化／洗練を行ないました。

「手法のレベルアップ」

Openthology はモデリングによるビジネスの可視化を重視しています(⇒コア文書【コンセプト編】参照)。しかし、システム開発とは異なる「ビジネスのモデリング」については難易度の高いものがあり、誰もが容易にモデリングできない場合があります。

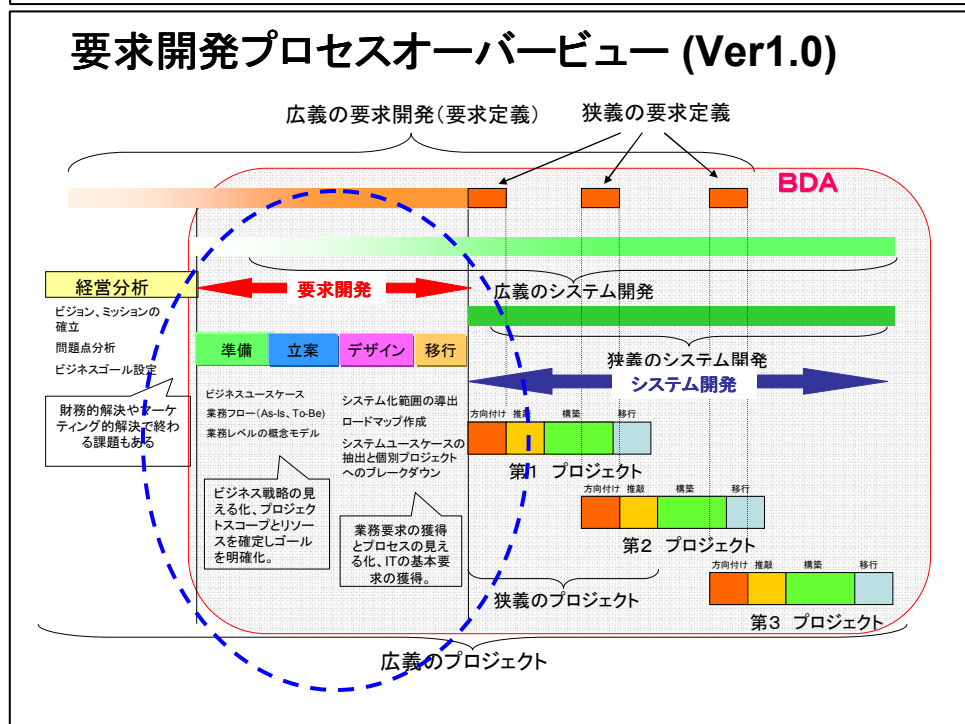
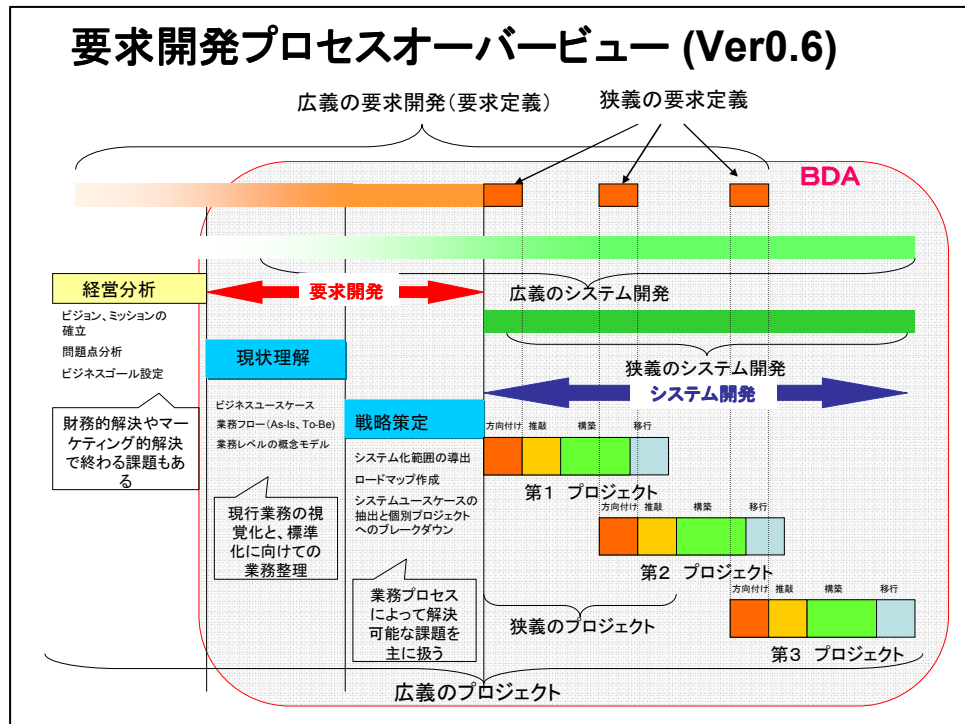
Openthology Version 1.0 では、ビジョンとゴールの見える化(BSC 戦略マップ モデリング)の手法を追加するとともに、概念モデリングの手法として「TFP 分割手法」を追加し、より容易にモデリングを可能としました。

それぞれの変更の詳細は、次項で個別に説明を行ないます。

2. 理論から実践へ

概要で述べたとおり、要求開発方法論 Openthology の Version0.6 は、要求開発における方法論の基礎であり、「なぜ要求開発が必要なのか(WHY)」「要求開発では何をすべきなのか(WHAT)」が論点の中心であり、「いつ(WHEN)」「誰が(WHO)」「どうやるのか(HOW)」についてあまり触れていませんでした。

Version 0.6 における要求開発プロセスは、大きく分けて「現状理解フェーズ」と「戦略策定フェーズ」の2フェーズで要求開発プロセスを定義していました。しかし、この2つのフェーズでそれぞれのモデリングを並行して行い、要求を開発することは困難です。そこで、Version 1.0 では、フェーズを4つに分割し、段階的な詳細化を行なうことにより、より容易に要求開発プロジェクトを推進できるようにしました。



Version 1.0 で定義された 4 つのフェーズは、「準備」「立案」「デザイン」「シフト」であり、それぞれのプロセスで行なうべきことは、プロセスキャビネット(後述)として整理されました。

要求開発のフェーズ (Ver1.0)

準備

- スコープとリソースの決定
 - ・ビジネス戦略の把握とプロジェクト課題の見極め
 - ・必要人材のアサイン
 - ・要求開発プロジェクトゴールの策定

立案

- 概観の把握と可視化
 - ・プロジェクトゴールにスコープしたビジネス領域の可視化
 - ・ビジネス要求の抽出

デザイン

- ToBe業務の設計
 - ・ToBe業務の設計
 - ・IT基本要求の開発

シフト

- システム計画
 - ・IT基本要求の抽出
 - ・システム移行計画書の作成

加えて、Version 1.0 では、仮説検証型 PDCA 反復の導入が行われました。要求開発が取扱うのはビジネス・ビジネスモデルそのものであるため、(従来のシステム開発の企画段階・要求定義段階でよくあるように)「根拠なき計画の立案」が行なわれてしまったり、「リアル間のない計画」となってしまう可能性があります。そこで、「準備」「立案」「デザイン」「シフト」の 4 つのフェーズそれぞれに、Plan, Do, Check, Action の Disciplines(規律)を用い、より確実な要求開発の推進を可能としています。

仮説検証型PDCA反復サイクル(Ver1.0)

計画(PLAN)

Input
↓

- 何をもって“OK”とするか、という基準を決めること
 - ・各Phaseの「目的」達成のために何を行うのか、を「決める」
 - ・それは、何をもってOKとなる(する)のか、という達成基準を「決める」
 - ・これは「なにをやるのか」を確定することを意味する

実施(DO)

↑
評価

- 各PhaseのPLANを実現すること
 - ・PLANで決めたことを決めた基準に達するように行うこと
 - ・CHECKが出来るように行わなければ意味がない

検証(CHECK)

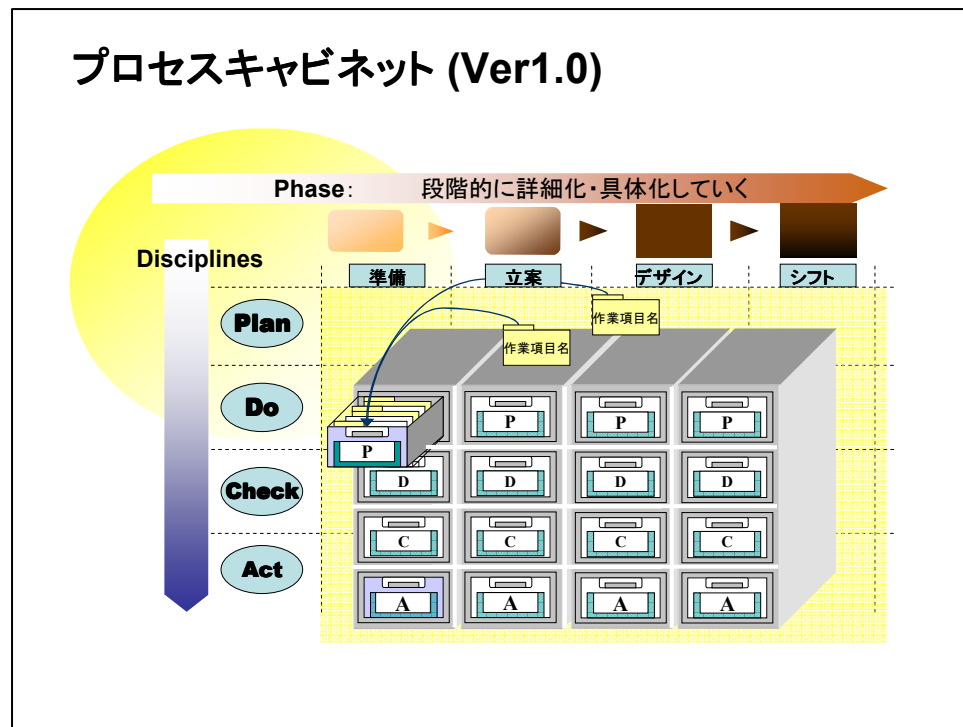
Input
↓

- DOの結果をPLANにもとづいて評価すること
 - ・PLANで計画された基準と方法によりDOの結果を評価する
 - ・「誰が」行うのか、を目的に応じて決めておく

対応(ACT)

- 出来ることを「すぐに」やること
 - ・CHECKでの非適合事項の是正だけがACTではない
 - ・その時点で「やったほうがよい」ことは、その時点で行う
 - ・その時点でできないことは、次フェーズ(イテレーション)のPLANにまわす

この4つのフェーズ、4つの Disciplines(規律)はマトリクスを構成しています。これを、Openthologyでは「プロセスキャビネット」と呼び、それぞれのマトリクスごとに、方法論やモデル・ノウハウを整理しています。



要求開発プロジェクトを計画・実行する際には、それぞれのプロセスキャビネットにある作業項目を抽出し、WBSを作成することによって、容易に要求開発を行うことができるようになりました。

プロセスキャビネットの導入で、より具体的な要求開発プロセスを組み立てることが可能となりましたが、ここでひとつ問題が発生しました。プロセスキャビネットを効果的に利用するためには、PDCAのセットとしてアクティビティを取り出し、アクティビティ間のインプット/アウトプットを考慮してWBSを組成する必要がありますが、これは・・・「けっこう面倒！」なのです。そこで、要求開発アライアンスでは標準的なPDCAアクティビティを「プロセスセル」という形で提供することにしました(次頁の図を参照)。

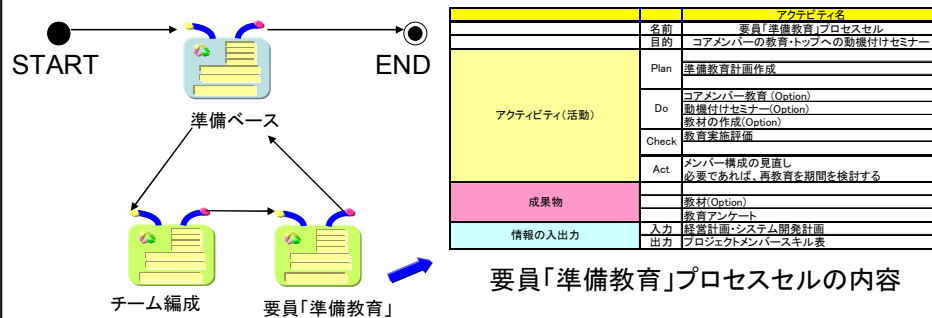
結果として、Openthologyでは大きく2種類の利用方法を提供します。もしも利用者が既に自社の標準的な開発標準を持っている場合には、必要なアクティビティのみをプロセスキャビネットから取り出し、プロジェクトに組み込む事ができます。また、プロジェクトの特性によって標準的なプロセスセルを利用できない場合にも、プロセスキャビネットをベースに独自のWBSを作成することができます。

また、標準的な要求開発プロジェクトを行なうのであれば、プロセスキャビネットではなく、標準プロセスセルを元にWBSを組成することができます。

プロセスセル (Ver1.0)

- プロセスキャビネットからPDCAの単位に活動をまとめ成果物と一緒にまとめパターン化したもの

例) 準備フェーズのプロセスセル



3. 手法のレベルアップ

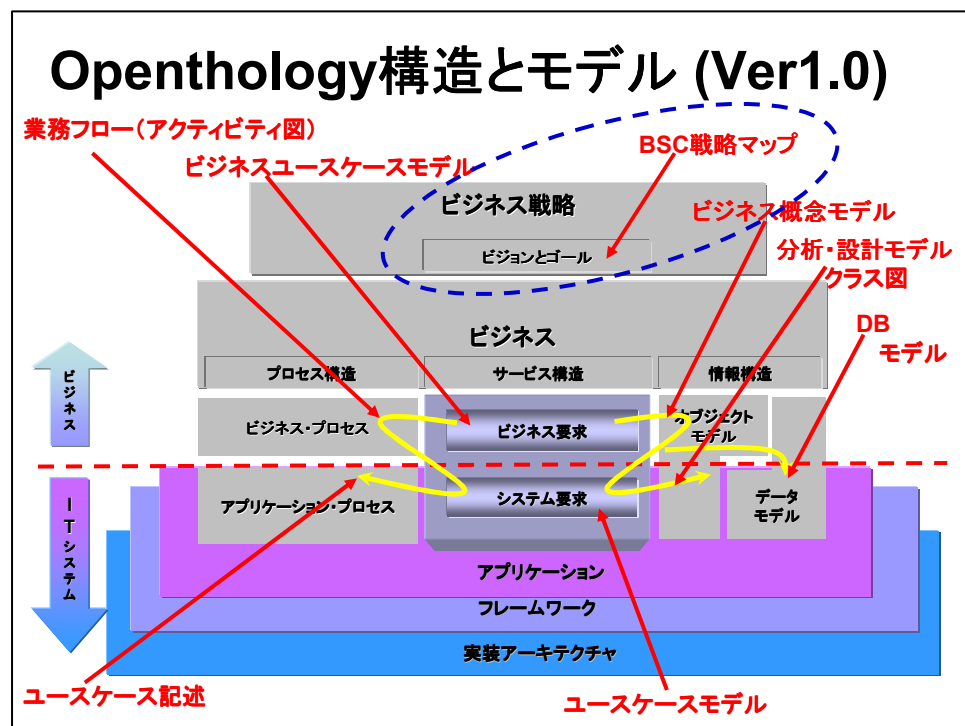
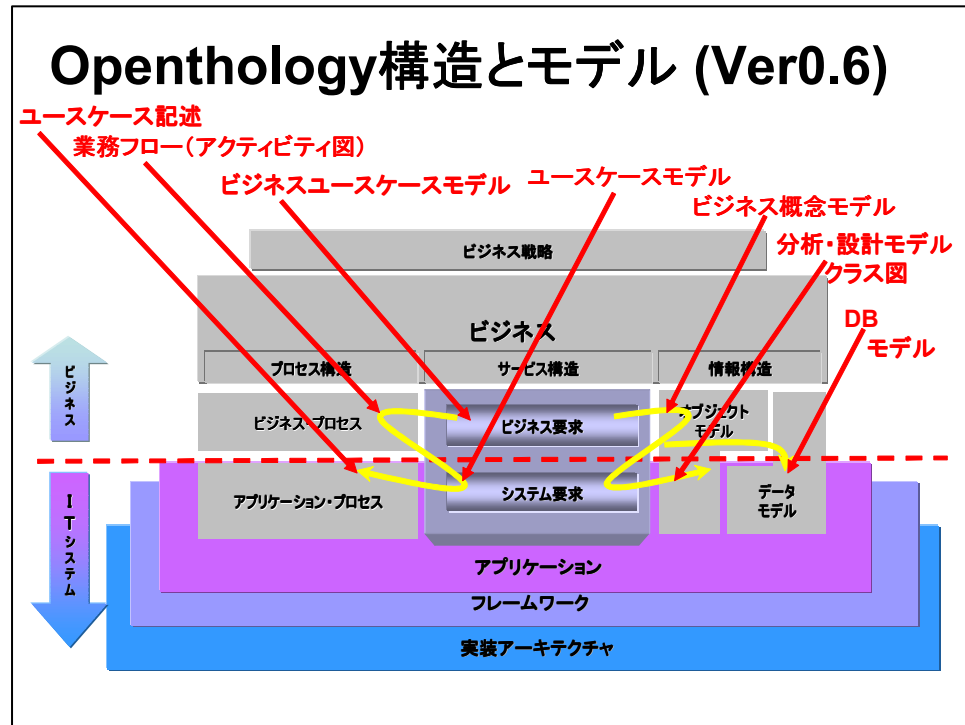
ビジョンとゴールの見える化

Version 0.6 では、そもそものビジネス戦略は要求開発の外部からのインプットでした。しかし、実態としては、ビジネス戦略が曖昧であるために、要求開発のインプットとし難く、結果として根拠の無い計画や戦略を選択してしまうリスクがありました。そこで、戦略を実行するためのモデルとして、BSC 戦略マップが追加されました。

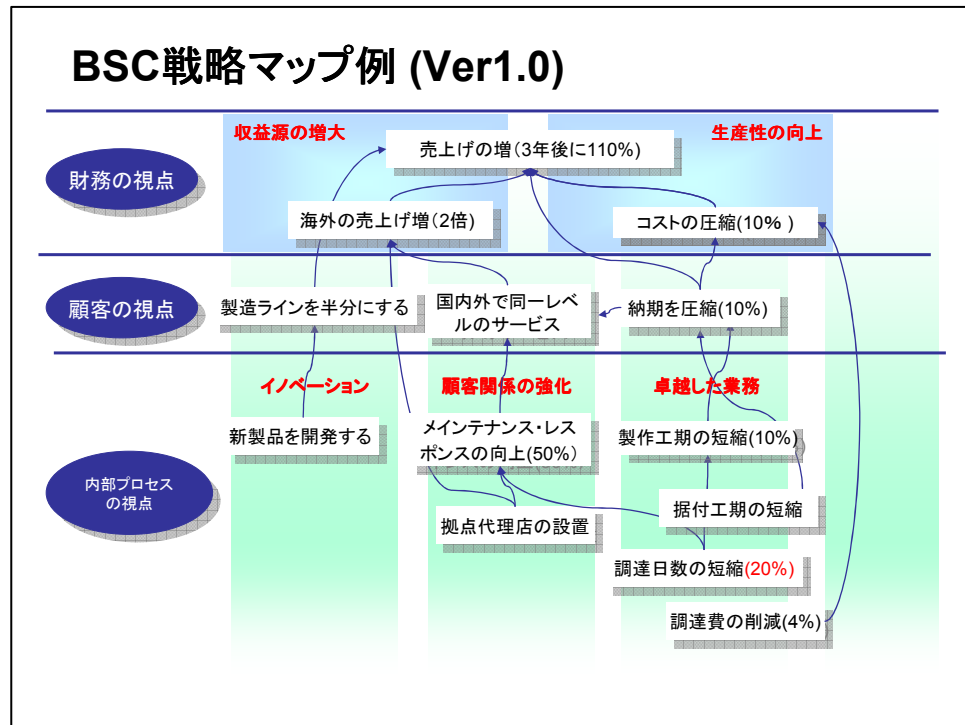
戦略を実行するために (Ver1.0)

- Openthologyの提言
 - BSC戦略マップの利用 (戦略の見える化)
 - しかし「最初からBSCを利用している」企業ばかりとは限らない
 - そこで、既存の経営計画などを逆に「あてはめて」いく
 - 「戦略のリバース」と呼ぶ所以は「あてはめ」にある
 - IT貢献度マップ
 - ITシステムと「戦略」の因果関係を明らかにする
 - ゴール記述書
 - システム開発のゴール(目標)を明確にしてリストする

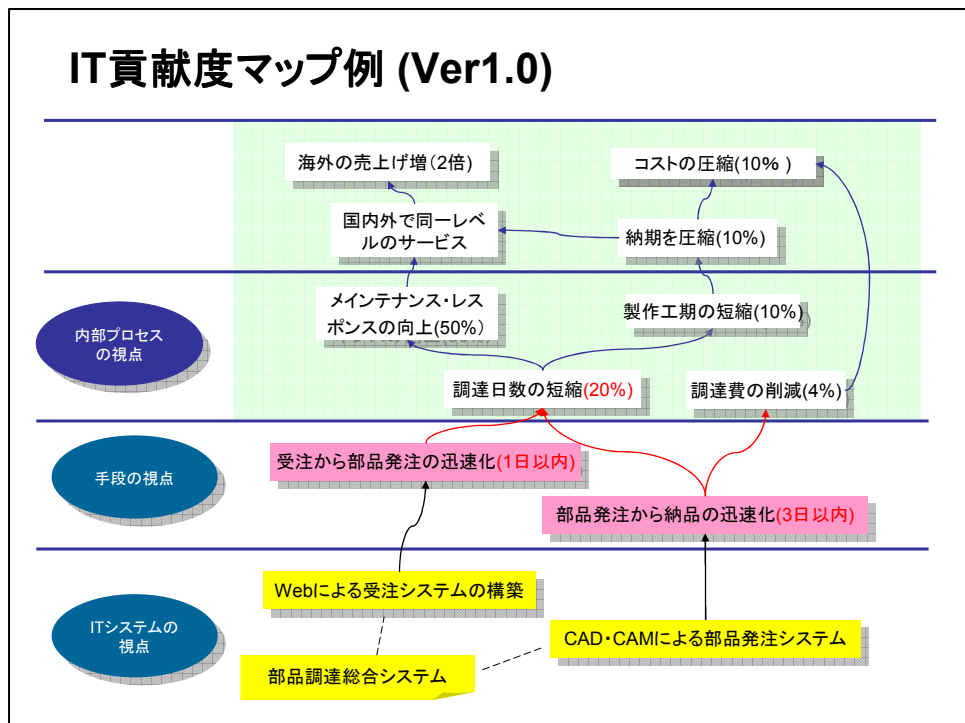
この違いは、次の Openthology 構造図により明確となります。Ver 0.6 ではビジネスそのものを分析/モデリング対象としていましたが、Ver 1.0 ではビジネス戦略の分析/モデル化もスコープの範囲と捕らえています。なお、誤解されやすいのですが、Openthology はビジネス戦略そのものの立案を目的とするわけではありません。既に存在するビジネス戦略(経営計画や企画案など)を分析・分解し、要求開発におけるゴールを明確化するためにモデリングします(よって、「戦略のリバース」と呼ぶことがあります)。



BSC 戦略マップは、ビジネス戦略を「財務の視点」「顧客の視点」「ビジネスプロセスの視点」「学習と成長の視点」で捉えて図表を記載し、関連性を明確化します。



IT 貢献度マップは、BSC 戦略マップの拡張で、具体的なビジネス戦略に IT システムがどう貢献するかを明確にし、ゴールと達成目標を明確化するものです。

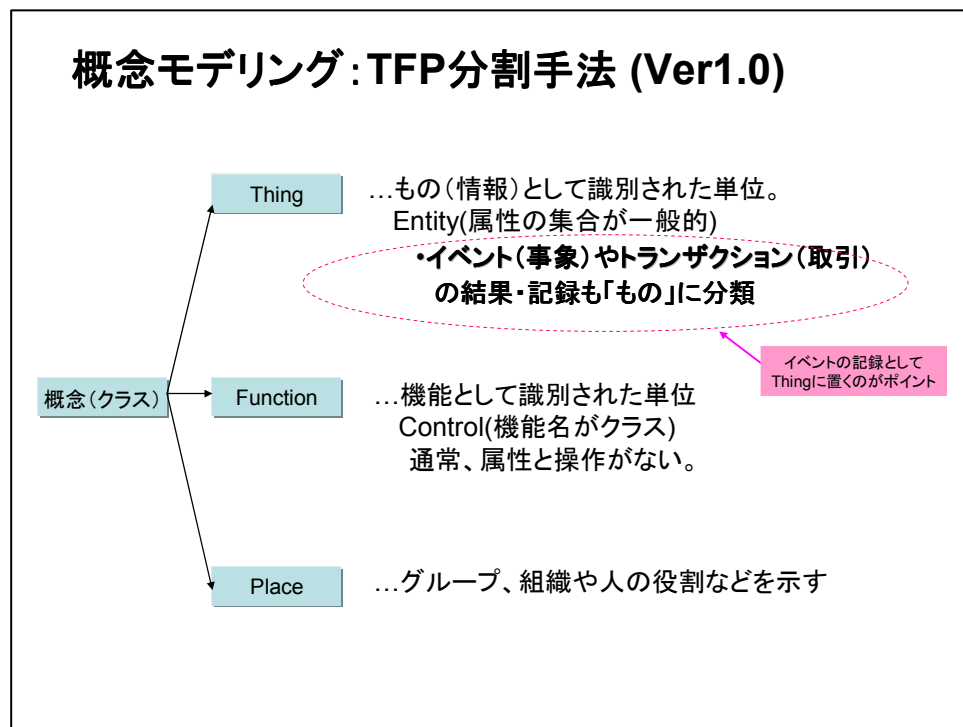


各モデルの詳細については、モデリング概要またはモデリングガイドラインを参照してください。

TFP 分割手法(概念モデルの強化)

Openthology では、ビジネスドメインの分析整理に、オブジェクト(クラス)モデルを重視しています。しかし、従来のオブジェクト(クラス)モデルは、概念の中に機能的なクラスが存在すると、機能の中に関連が隠されてしまったり、機能の入出力関係と、「もの」と「もの」との関係が混在することになり非常に曖昧になるという問題がありました。

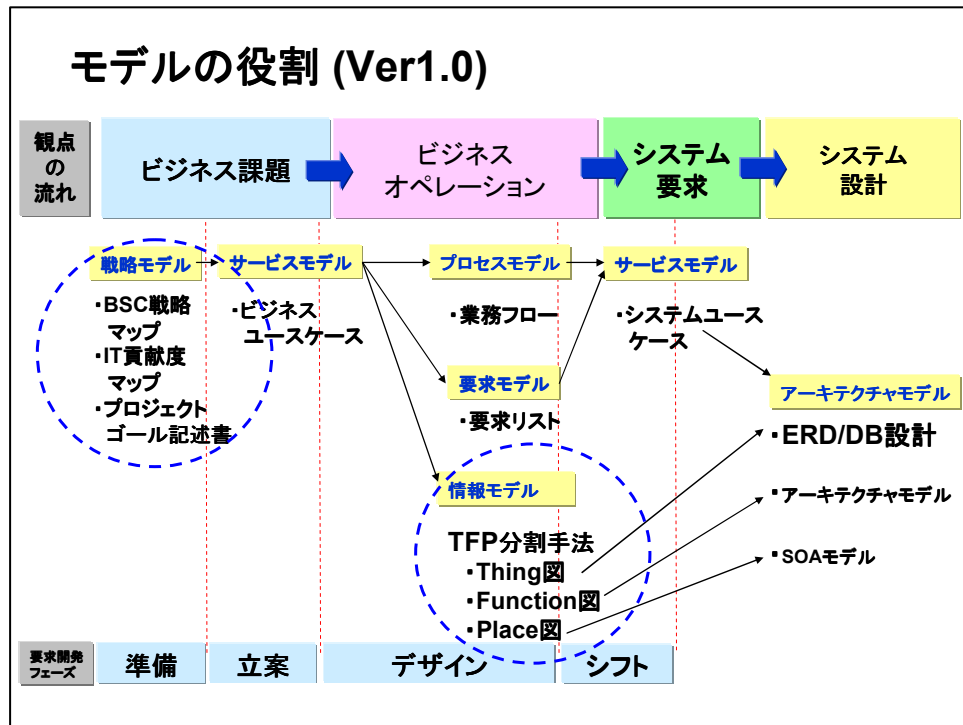
そこで、Version 1.0 ではこの問題を解決するために、TFP 分割手法を導入しました。概念(クラス)は、Thing, Function, Place という分類しやすい 3 つの属性に分け、結果として概念整理を容易にすることができました。



詳細については、モデリング概要またはモデリングガイドラインを参照してください。

Version 1.0 におけるモデルの役割

最後に、Version 1.0 において追加された 4 つのフェーズと、追加された 2 つを含むモデルの役割の関係を示します。フェーズが進むにあたって、観点が変わるとともに段階的にモデルも詳細化していく形となります。



以上が主な Openthology Ver 0.6 と Ver 1.0 の相違点です。

なお、本資料の作成にあたっては、以下の資料を参考としました。各資料を作成いただいた皆様に感謝いたします。

- ・ Openthology コア Ver 0.6 「BDA コンセプトに基づく Openthology 要求開発方法論」
- ・ Openthology Ver 1.0 Ignition Now! (安井理事:2005/10 要求開発アライアンス定例会発表資料)
- ・ Openthology Ver 1.0 Ignition Now! (萩本理事:2005/10 要求開発アライアンス定例会発表資料)
- ・ 要求開発～価値ある要求を導き出すプロセスとモデリング～(日経 BP 社/要求開発アライアンス)
- ・ 要求開発(Openthology)入門 (萩本理事/(株)豆蔵様のトレーニングテキスト)