

要求の変化に対応する
情報システム構築技術の適用に関する調査

概要調査報告書

2012年4月26日

独立行政法人情報処理推進機構
技術本部 ソフトウェア・エンジニアリング・センター

| | |
|--------------------------|----|
| 1. 目的と背景 | 3 |
| 2. 調査手順と調査方法 | 6 |
| 3. 新しい技術課題の追加調査 | 10 |
| 4. 技術課題項目の整理と深掘調査 | 25 |
| 5. 技術課題の分析 | 39 |
| 6. まとめ | 48 |
| 付録A. アーキテクチャフレームワーク | 55 |
| 付録B. DoDのケーパビリティアプローチ | 60 |
| 付録C. ソフトウェア工学分野の技術課題との比較 | 63 |

本報告書全般に関する注意

本報告書で活用している多くの用語について、日本語訳が定まっていないか、JISとして定義が進められているものなどがあり、日本語として訳することが不適切なものも多い。読み易さのために日本語訳を試みているが、ここで使われている日本語訳は本報告書限りとする。各種団体による標準化が待たれる。

1. 目的と背景

1. 目的と背景

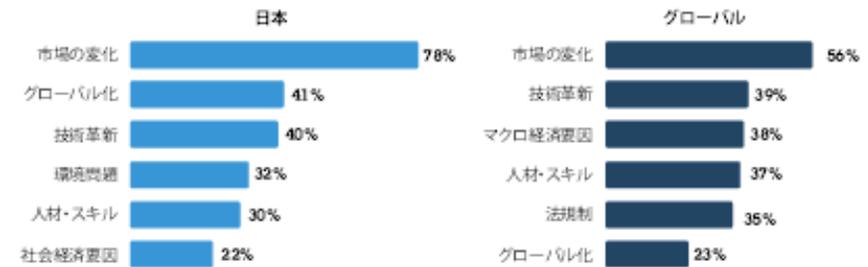
【目的】

- 「要求の変化」という課題にとって重要となる技術項目を調査し、技術課題をまとめる。
- 変化が激しい現代社会において、社会や企業を支える情報システムを構築するための重要な技術を整理し、提供することで、我が国の社会インフラや企業活動の強化に資すること、特にユーザ企業における情報システム構築の技術力、産業力強化に資する技術的な指針が示され、刻々と変化する社会や国民ニーズに対応可能なIT社会の実現に資することを目的とする。

【背景】

① 様々な環境の変化

- 情報システムは現代社会にとって必須
- 市場環境、自然環境、社会環境の変化が早い
- 環境の変化は、企業活動、社会活動に大きな影響
- 環境の変化は、情報システムに対する要求の変化として影響
- IT利活用を進めるためには**要求の変化に対応する情報システムの構築技術**が必要



出典：日本 IBM：Global CEO Study 2010 Japan Report

図 CEOが考える今後3年間で自社に大きな影響を与える外部要因

② 持続可能性と要求の変化

- 激しい変化が起こったとしても企業や社会インフラの事業経営を止めることはできない。
- これらの事業者をITで支え続けるために、**継続性の高い情報システムが求められる。**

③ 障害と要求の変化

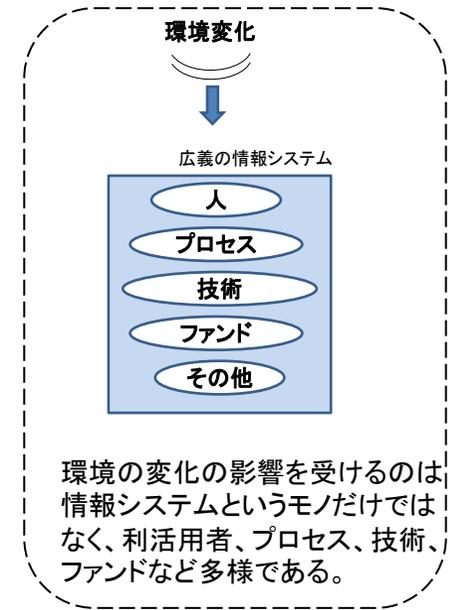
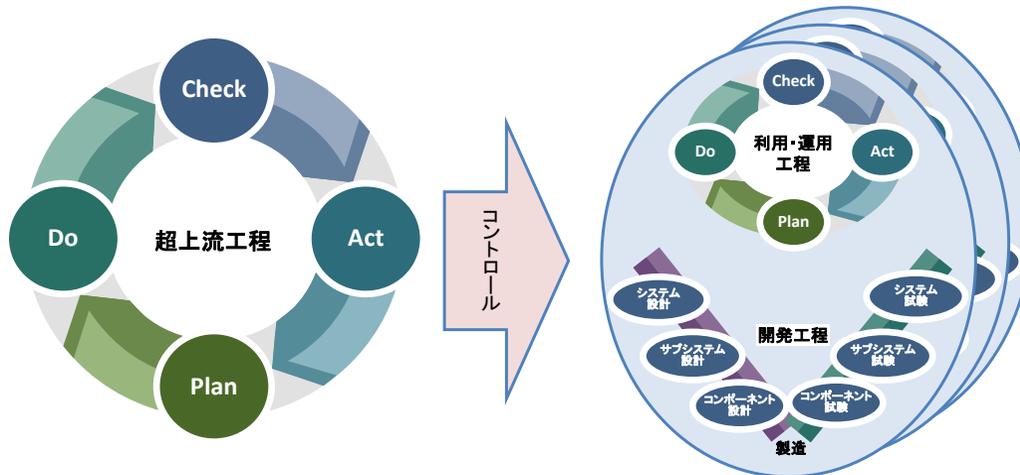
- 障害の発生は「要求通りにシステムが動かなくなった」こと
 - それを「要求逸脱」として認識、要求逸脱へは、**利用・運用面も含む対応**が必須
- ⇒ 「要求の変化」への対応は、**利用・運用技術も含めた広い技術での対応が必要**

1. 目的と背景

【情報システムの継続性の実現技術の課題解決に向けて】

要求逸脱の発生に対して、「利用・運用」と「開発」で情報システムの継続性を維持

- 「利用・運用」を含めた広義の情報システムは、「環境の変化」を受容
- 「計画的にコントロールする超上流工程」も含む、トータルなシステムにより実現
- 計画的な発想、利用・運用起点的な発想、継続開発的な発想が重要



【参考】
「ISO/IEC/IEEE29148:2011 Requirement Engineering」では、利用・運用起点的な考え方は ConOps (Concept of Operation) や OpsCon (Operational Concept) に基づくシステムの実現として認識

2. 調査手順と調査方法

2. 調査手順と調査方法: 調査項目

①新しい技術課題の追加調査

- システムズエンジニアリング(超上流工程分析評価技術を中心に) (文献・Web調査、ヒアリング調査)
- 高信頼システム(検証技術) (ヒアリング調査)
- 非ウォータフォール型開発技術 (文献・Web調査)

さらに2010年度調査結果についてフィージビリティを調査し、その結果を反映して再整理

②技術課題項目の整理と深掘調査

- 次の観点による2010年度調査結果も含めた技術課題の整理
 - ・ ライフサイクルプロセス、ステークホルダー要求
 - ・ 品質要求と制約(非機能要求)、アーキテクチャと構成要素

**本調査は、IPA/SEC内要求発展型開発WGによってレビューを受け、適宜方向性の改善を図った。
本報告書は改善された方向でまとめられた。**

2. 調査手順と調査方法: レビュー

WG委員一覧を以下に示す。

要求発展型開発ワーキンググループ: (敬称略、順不同)

| | |
|---------------|----------------------|
| 主査: 山本 修一郎 | 国立大学法人名古屋大学 |
| 委員: 赤埴 淳一 | 日本電信電話株式会社 |
| 委員: 秋山 浩一 | 富士ゼロックス株式会社 |
| 委員: 白坂 成功 | 学校法人慶應義塾大学 大学院 |
| 委員: 鈴木 三紀夫 | NPO法人ソフトウェアテスト技術振興協会 |
| 委員: 成瀬 泰生 | 富士通株式会社 |
| 委員: 山本 久好 | 日本アイ・ビー・エム株式会社 |
| 委員: 鷺崎 弘宣 | 学校法人早稲田大学 |
| IPA/SEC 山下 博之 | |
| IPA/SEC 吉元 一郎 | |
| IPA/SEC 岡山 歩 | |

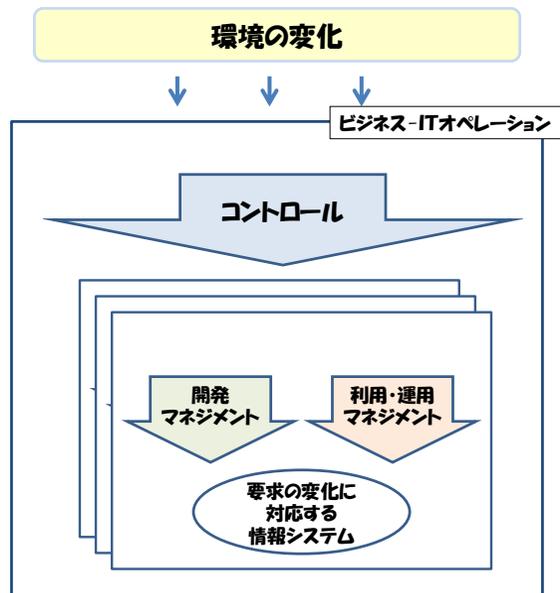
3. 新しい技術課題の追加調査

3. 新しい技術課題の追加調査:

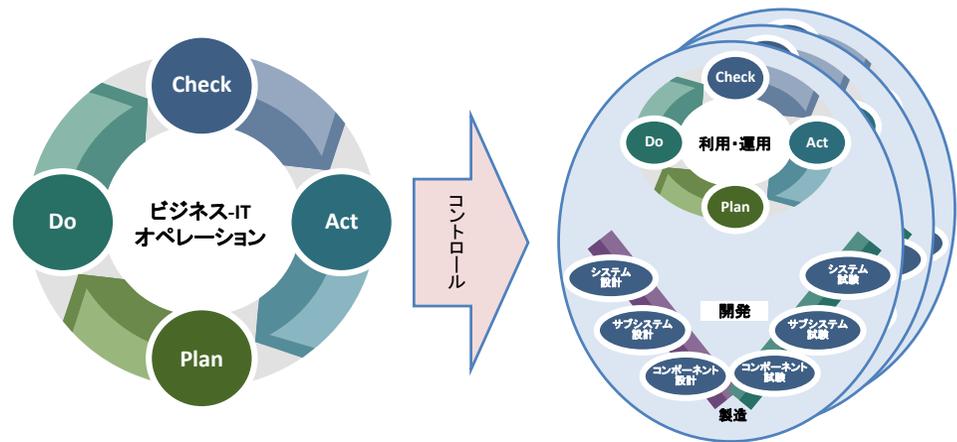
情報システムの継続性の実現技術の仮説

- 情報システムの継続性: 情報システム群全体をコントロールすることが特徴
- 本報告書では、これを**ビジネス - ITオペレーション**と呼ぶ

仮説設定: 「**ビジネス-ITオペレーション**が、要求の変化をコントロールする」



環境の変化を受け、要求の変化をコントロール



ビジネス環境の変化を受け、ビジネス-ITオペレーションにより、要求の変化をコントロール

3. 新しい技術課題の追加調査: 対象領域へのWG意見の反映

次のとおり、対象領域に運用技術を追加した。

- ① システムズエンジニアリング(超上流工程分析・評価技術を中心に)
- ② 高信頼システム(検証技術)
- ③ 非ウォーターフォール型開発技術
- ④ 運用技術(ITIL V3を対象に)

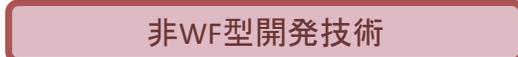
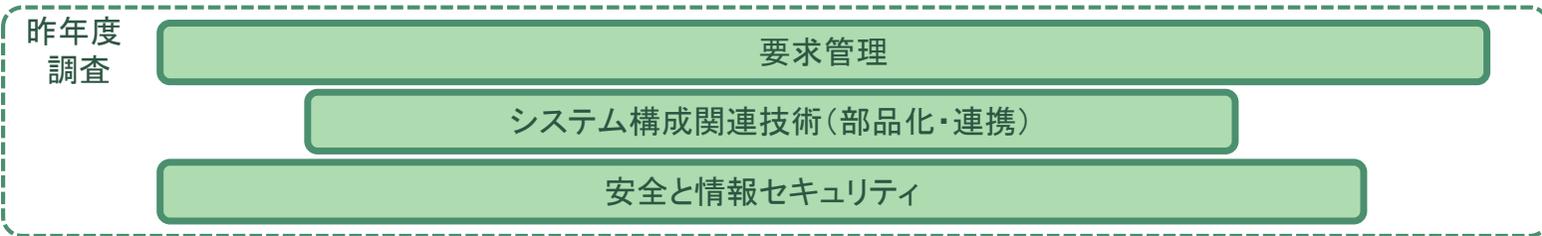
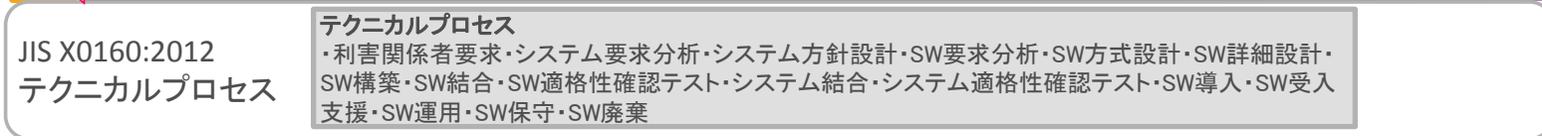
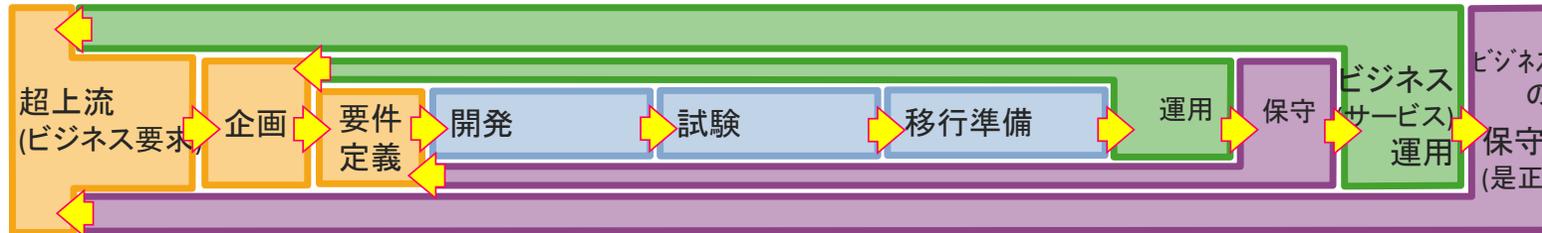
また技術課題としては取り上げなかったが、次のアーキテクチャフレームワークについて、WG意見に従ってまとめた。

- 経済産業省の EA (Enterprise Architecture)
- Zachman アーキテクチャフレームワーク
- TOGAF (The Open Group Architecture Framework) Version 9
- DoDAF (DoD Architecture Framework) 2.0

3. 新しい技術課題の追加調査: 追加技術課題

プロセス範囲と検討対象の技術/知識体系

▶ 主なプロセスの流れ



今回検討対象とした技術および知識体系

3. 新しい技術課題の追加調査: 追加技術課題の選定理由

- ① システムズエンジニアリング(超上流工程分析・評価技術を中心に)
 - ・ システム構築技術の国際標準
 - ・ 「要求の変化」が話題のひとつ(エンタプライズシステムズエンジニアリングと呼ばれる)

- ② 高信頼システム(検証技術)
 - ・ 高信頼システムを実現する技術として予防活動と検知活動
 - ・ 2010年度予防活動として形式手法を選定、残りの検知活動を2011年度調査

- ③ 非ウォーターフォール型開発技術
 - ・ アジリティの高いシステム構築等のための短期開発を実現する典型的な手法
 - ・ 2010年度調査に含まれず ⇒ 2011年度調査で追加

- ④ 運用技術(ITIL: Information Technology Infrastructure Libraryを対象に)
 - ・ 要求の変化への対応をするためには、運用と開発の双方の技術が必要、その運用部分
 - ・ WGの議論で要求の変化にとって運用技術の重要性を討議、技術項目に追加。
 - ・ 2011年度は、ITIL V3に限定

① ステークホルダー要求

システム要求の変更は、変更の可否判断のためには要求の因果関係をたどる必要があり、ステークホルダー分析が有効である。技術課題の性質を整理するために、各ステークホルダーの典型的な要求の変化に対する技術課題の貢献の可否を示すことができる可能性があるためである。

② ライフサイクルプロセス

具体的に何を実施するところで、技術課題が適合するののかについて整理することで、具体的な開発等での活用の特徴を表現できる可能性があるためである。

③ その他

IPA/SEC非機能要求とアーキテクチャ分析WGで活用した軸

- － 品質要求と制約に対する整理
- － アーキテクチャと構成要素に対する整理

3. 新しい技術課題の追加調査： 2011年度追加調査

■2010年度調査結果に含まれていない新しい技術課題分野・課題項目を抽出し、再整理。

| | | |
|----------|-------------------------------|-----------------------|
| 【2010年度】 | (1)要求管理 | [有識者ヒアリング調査] |
| | (2)システム構成関連技術(部品化・連携) | [有識者ヒアリング調査] |
| | (3)品質保証(「安全と情報セキュリティ」とタイトル変更) | [有識者ヒアリング調査] |
| 【2011年度】 | (4)システムズエンジニアリング(超上流を中心に) | [文献・Web調査、有識者ヒアリング調査] |
| | (5)高信頼システム(テスト検証技術) | [有識者ヒアリング調査] |
| | (6)非ウォーターフォール型開発技術 | [文献・Web調査] |
| | (7)運用技術 | [WG委員講演] |

技術課題を洗い出したところ、合計で10の技術課題を選出 ⇒ 2010年度調査を加え再整理

- システムズエンジニアリング
 - ・ 超上流工程分析評価技術
 - ・ 上流工程における横断的連携技術
- 検証技術
 - ・ 上流工程開発力強化技術
 - ・ ロバストネスの確保
- 非ウォーターフォール型開発技術
 - ・ 要求の明確化
 - ・ 構成管理
 - ・ 短期間での開発
 - ・ 反復による開発
 - ・ 品質管理
- 運用技術
 - ・ ITIL継続的改善

調査項目： 概要、課題解決効果、課題解決に向けた留意点・課題、情報リソース、2010年度調査視点によるフェージビリティ

3. 新しい技術課題の追加調査： 2010年度調査等を加えた 再整理結果

(1) 要求管理

| 分類 | 技術項目・関連プラクティス | 目的および補足 |
|-------------------------|--|--|
| 要求の明確化 | <ul style="list-style-type: none">• GSN (Goal Structuring Notation)• 形式手法• アーキテクチャ記述 | 情報システムを保守し、要求を明確化し、見えるような状態にしておき、変化に耐えるようにする必要がある。 |
| スコープ決め | <ul style="list-style-type: none">• 変動部分の明確化• 投資効果と市場と技術変化の見極め• ドメイン分析と概念モデリング | 保守性を高めるために、拡張ポイントを明確にするためのスコープ決めを行う必要がある。 |
| トレーサビリティ管理 /ベースライン管理 | <ul style="list-style-type: none">• RTVM (Requirement Traceability Verification Matrix) | 部分的に段階的詳細化で開発を実施するために、要求のベースラインを管理し、トレーサビリティを確保することが重要であるため、RTVMを活用する。 |
| コントロールケースの活用 | <ul style="list-style-type: none">• コントロールケース | 変化の要求記述法として将来要求等についての記法がある。 |

3. 新しい技術課題の追加調査： 2010年度調査等を加えた 再整理結果

(2) システム構成関連技術(部品化・連携)

| 分類 | 技術項目・関連プラクティス | 目的および補足 |
|--------|--|--|
| 構造の明確化 | <ul style="list-style-type: none">・アーキテクチャ分析・形式手法・プロダクトライン・プロトフォームアプローチ・スケルトンと部品 | 情報システムの保守性を良くするためには、ソフトウェアを分解可能な形で部品化しておく必要がある。そのためにはアーキテクチャ記述言語を活用して、構造の明確化と定義を行い、アーキテクチャを明確にする必要がある。 |
| 部品化 | <ul style="list-style-type: none">・SOA・オブジェクト指向・業種・業務特化 | 情報システムの保守性を良くするためには、ソフトウェアを分解可能な形で部品化しておく必要がある。 |
| クラウド活用 | <ul style="list-style-type: none">・性能および機能のスケーラビリティ | ビッグデータを活用したシステム等、システムを経済的に構築するために、データ処理のスケーラビリティや、機能の変化を十分に確保できるアーキテクチャで対応する必要がある。 |

3. 新しい技術課題の追加調査： 2010年度調査等を加えた再整理結果

(3) 安全と情報セキュリティ

| 分類 | 技術項目・関連プラクティス | 目的および補足 |
|--------------------|---|---|
| テスト網羅性の保証 | <ul style="list-style-type: none">・ アシユアランスケース・ トレーサビリティ・ プロダクトライン・ 形式手法・ テスト網羅性証明・ 回帰テスト・ テスト自動化 | ビジネス要求の変更を確実に反映させるために、技法等を活用することでテストケースの網羅性を保証し、テストを実施することが求められる。 |
| 不測の管理・仮定の管理 | <ul style="list-style-type: none">・ アシユアランスケース | 環境の変化から要求は絶えず進化するため、不測の事態を考慮し、ソフトウェアを進化させる必要がある。 |
| 運用性・可用性・アシユアランスの評価 | <ul style="list-style-type: none">・ アシユアランスケース | 要求の変化を確実に実装するために、それに対応した運用性、可用性やアシユアランス評価を確保する手順が構築できていることが必要である。 |
| 共通基盤としての保証 | <ul style="list-style-type: none">・ 情報セキュリティ | 要求が変化したシステムを安全に提供するためには、情報セキュリティが確保できる手順が構築できていることが求められる。 |
| 他者に求められる保証 | <ul style="list-style-type: none">・ SLA・ 情報セキュリティ保証 | 情報セキュリティが確保できることを保証するために、認証などの他者に求められる保証が必要となることがある。 |

3. 新しい技術課題の追加調査： 2010年度調査等を加えた再整理結果

(4) 超上流工程分析・評価(システムズエンジニアリング)

| 分類 | 技術項目・関連プラクティス | 個別技術項目の目的および補足 |
|------------|---|--|
| 超上流工程分析・評価 | <ul style="list-style-type: none"> システムシンキング System thinking | システムズアプローチのための思考法である。ものごとを多面的に分析するための考え方であり、多面性を表現する各観点から鳥瞰した内容の表現力、つまりモデリングもこの能力が必要である。 |
| | <ul style="list-style-type: none"> ケーパビリティに基づく開発の分析 Capability-Based Planning Analysis | 対象組織の目標達成に向けた、対象組織の情報システム群の高水準要求分析のことである。ニーズ、ケーパビリティ、目標達成可能性の認識と評価を実施、アーキテクチャモデリングと評価、統合的なポートフォリオマネジメント、同様に改善計画と実施、それに要求の変化に対応したケーパビリティの修正マネジメントを行う。 |
| | <ul style="list-style-type: none"> エンタプライズ有効性評価 Enterprise Evaluation and Assessment | エンタプライズシステムを含むシステム全体の有効性を評価する。システム全体のもつべき全体の統合の在り方を分析し、方向性を決定する。 |
| | <ul style="list-style-type: none"> 戦略的技術計画 Strategic Technology Planning | エンタプライズシステムが準拠すべき標準やパターンの最小セットを制定する。リスクを避け目標達成可能性が向上する技術戦略を策定する。 |
| | <ul style="list-style-type: none"> 技術および標準動向調査 Technology Planning | 今後の技術動向、標準動向を予測し、ロードマップを作成する。 |
| | <ul style="list-style-type: none"> ステークホルダー分析 Stakeholder analysis | 当該システムのステークホルダーが誰であり、どのような役割をもっているのかを分析する。 |
| | <ul style="list-style-type: none"> ConOpsの分析・定義 | システム全体としての 利用・運用の在り方と、それに従ったシステムのコンポーネントの利用・運用の在り方、相互依存関係などを分析して、システムの利用・運用の概念をまとめる。 |

3. 新しい技術課題の追加調査： 2010年度調査等を加えた 再整理結果

(5) 上流工程における横断的連携

システムズエンジニアリングにおける方法論のうち、システムズエンジニアリングハンドブックに挙げられている Cross-Cutting Technical Methodsという技術課題となる3つの方法論に加え、検証技術の2つの技術課題を加えた。

| 分類 | 技術項目・関連プラクティス | 個別技術項目の目的および補足 |
|---------------|---|---|
| 上流工程における横断的連携 | <ul style="list-style-type: none"> モデリング・シミュレーション・プロトタイピング Modeling、Simulation、Prototyping | システムをモデル、設計、要求の三要素で特徴づける。モデリングはシステムの振舞や機能性、物理的な状況など特徴を表現するための道具立てである。また数学モデルまで拡張し、実際にコンピュータ上等でモデルを動かして評価するのがシミュレーションである。評価目的に限定された仕様で実現してみることをプロトタイピングと呼ばれる。 |
| | <ul style="list-style-type: none"> 機能に基づくSE手法活用 Functions-Based Systems Engineering Method | 機能要求の分析と洗い出しを行う。 【結果】振舞、コンテキスト、コントロールフロー、データフロー、データディクショナリ、ER、Functional Flow Block、Models、Simulation Results、Integrated Definition for Functional Modelingの各図を得る。 |
| | <ul style="list-style-type: none"> オブジェクト指向SE手法活用 Object-Oriented Systems Engineering Method | システムズエンジニアリングのプロセスをOMGが中心となってSysMLで記述する方法論を用意している。アシュアランスについても記述する方法論の研究も進んでいる。アクティビティとしては次のとおり。 Analyze Needs, Define Systems Requirements, Define Logical Architecture, Synthesize Allocated Architectures, Optimize and Evaluation Alternatives, Validate and Verify System |
| | <ul style="list-style-type: none"> 上流工程開発力強化 | 状態遷移系テスト、W字型開発などで早期リリースするために、開発の手戻りを削減する。そのために上流工程の開発力を強化する。 |
| | <ul style="list-style-type: none"> ロバストネスの確保 | 環境の違い等の少々の変化による改修を避けるために、堅牢性を確保したソフトウェアの実現を確認するロバストネステストを実施する。 |

3. 新しい技術課題の追加調査： 2010年度調査等を加えた 再整理結果

(6) 非ウォーターフォール型開発技術

2009年度にIPAが実施した非ウォーターフォール型開発技術の調査における技術資料等から技術課題をピックアップ

① 要求の明確化

| 分類 | 技術項目・関連プラクティス | 目的および補足 |
|-------------------------|--|--|
| 非ウォーターフォール型開発における要求の明確化 | <ul style="list-style-type: none">・ 機能仕様定義・ パフォーマンス仕様定義・ プロトタイピング・ 実現可能性調査・ ビジネス調査 | 顧客のニーズに従った商品を適時に提供するため、システムを絶えず進化させるための前提として要求の変化を明確化する必要がある。そのため、要求を明確化する技術を必要とする。 要求の明確化には、機能仕様・パフォーマンス仕様の定義や、ビジネス要件やシステムがサポートするビジネスプロセスの範囲を明確化した上でのプロトタイプの実成などが技術課題として挙げられる。 |

② 構成管理

| 分類 | 技術項目・関連プラクティス | 目的および補足 |
|-----------------------|---|---|
| 非ウォーターフォール型開発における構成管理 | <ul style="list-style-type: none">・ コーディング規約・ 所有権管理・ トレーサビリティ管理・ 継続的統合 | 短期間の開発サイクルを回す非ウォーターフォール型開発手法では、日々変化する要求の変化に対応するために、開発工程における多くの管理技術や開発技術がある。各ステークホルダー要求の変更に 対応し、頻繁に修正を入れつつ管理を行うためには、コーディング規約や所有権の管理、トレーサビリティ、ビルド自動化や自動テストといった継続的な統合にかかわる技術などが重要となる。 |

3. 新しい技術課題の追加調査： 2010年度調査等を加えた 再整理結果

③ 短期間での開発

| 分類 | 技術項目・関連プラクティス | 目的および補足 |
|---------|--|---|
| 短期間での開発 | <ul style="list-style-type: none">・機能毎の開発・コード自動生成・シミュレーション・テスト駆動開発・シンプルな設計のための技術 | 非ウォーターフォール型開発手法では日々変化する要求の変化に対応するために、短い開発工程を繰り返す頻りにリリースし、ステークホルダーとともに評価と計画の見直しを実施する。短い開発工程を実現するための技術として、コード自動生成やシミュレーション、テスト駆動開発などが挙げられる。 |

④ 反復による開発

| 分類 | 技術項目・関連プラクティス | 目的および補足 |
|---------|--|---|
| 反復による開発 | <ul style="list-style-type: none">・リファクタリング・イテレーション計画・スプリントレビュー・頻繁なリリース管理 | 非ウォーターフォール型開発手法では開発サイクルが繰り返されるため、繰り返しのサイクル(イテレーション)を管理しつつ、ステークホルダーからの要求を適切に取り込んでいくための技術を必要とする。繰り返しの中で要求を取り込むためには、ユーザからのフィードバックを含めたスプリントレビューやサイクル開始前のスプリント計画が重要であり、プロジェクトの管理にはリファクタリングや、タイムボックス等を利用した頻繁なリリースの管理が挙げられる。 |

3. 新しい技術課題の追加調査： 2010年度調査等を加えた 再整理結果

⑤ 品質管理

| 分類 | 技術項目・関連プラクティス | 目的および補足 |
|-----------------------|--|---|
| 非ウォーターフォール型開発における品質管理 | <ul style="list-style-type: none">ソフトウェアインスペクションペアプログラミング | 品質の低下と開発期間の延長を防ぐために、ソフトウェアインスペクションを早期に実施する。 |

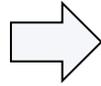
(7) 運用技術

継続的サービス改善は、超上流工程から運用工程まで結ぶ改善サイクルである。

| 分類 | 技術項目 | 目的および補足 |
|------|---|--|
| 運用技術 | <ul style="list-style-type: none">ITIL継続的改善 | 超上流工程から運用工程までITサービス開発を含む改善サイクルをもった利用・運用技術集である。 |

3. 新しい技術課題の追加調査：再整理結果の分析

SWEBOKの知識項目
から技術課題を見る



- 現時点で超上流工程の技術は適用外
- 非ウォーターフォール型開発技術がソフトウェア構築に対応

| 技術課題(分類) | ◎: 技術課題すべてに○ | | | | | | ○: 技術課題の一部に○ | | | | |
|---------------------------|--------------|------------|------|------------|-------------|------|-----------------|------|------|------|------|
| | 超上流工程分析・評価 | 上流工程の横断的連携 | 要求管理 | システム構成関連技術 | 安全と情報セキュリティ | 運用技術 | 非ウォーターフォール型開発技術 | | | | |
| | | | | | | | 要求の明確化 | 構成管理 | 短期開発 | 反復開発 | 品質管理 |
| SWEBOK知識項目 (一部システムに拡張) | | | | | | | | | | | |
| ソフトウェア要求 | | | ◎ | ○ | ○ | | ○ | | | | |
| ソフトウェア設計 | | | | ◎ | ○ | | | | ○ | | |
| ソフトウェア構築 | | | | ○ | ○ | | ○ | ○ | ○ | ○ | ○ |
| ソフトウェアテスト | | | | | ○ | | | | | | |
| ソフトウェア維持 | | | | | ○ | ◎ | | | | | |
| ソフトウェア構成管理 | | | | | | | | ○ | | | |
| ソフトウェア工学プロセス | | | | | | | | | | | |
| ソフトウェア工学ツールと方法 | | ◎ | ○ | | ○ | | | ○ | ○ | ○ | |
| ソフトウェア品質 | | | | ○ | | | | | | | ○ |
| 以下はSWEBOK範囲外 | | | | | | | | | | | |
| 超上流工程技術 | ◎ | | | | | ◎ | | | | | |

注) 注視点:

4. 技術課題項目の整理と深掘調査 － ライフサイクルプロセス －

4. 技術課題項目の整理と深掘調査:

(整理軸についての検討と整理)

WG意見に従い、次の内容で調査を実施

● 技術課題項目の整理と深掘調査

次の集約した観点による技術課題整理

- ライフサイクルプロセス
 - ✓ 超上流工程(ESE)
 - ✓ システム構築(ISO/IEC12207(JIS X0160)、ISO/IEC15288)
 - ✓ システム利用・運用(ITIL(ISO/IEC20000))
 - ✓ アーキテクチャ開発(TOGAF-ADM)

• ステークホルダー要求

今回の技術の分類としてではなく、具体的なシステムが設定できる場合に意義のある方法となるため、整理の例として適用した。

- その他
 - ✓ 品質要求(ISO/IEC25010)と制約(非機能要求)
 - ✓ アーキテクチャと構成要素

4. 技術課題項目の整理と深掘調査:

(整理軸についての検討と整理、開発プロセス(1/2))

- ソフトウェア方式設計までの上流工程のプロセスに関係する技術が多い

| | | △ ひとつ以上が関係 | ○ 過半数が関係 | ◎ 全技術課題が関係 | | | | | | | | | | | | | | | | | | | | |
|------------------------|--|---|----------|------------|--------|----------|--------------|------------|------------|----------|----------|----------------|-------------|----------|----------|----------|--------|--------|--------|-----------|--------|--------|--------|---|
| | | ●: ISO/IEC12207+15288, ○: ISO/IEC12207, ■: ISO/IEC15288 | | | | | | | | | | | | | | | | | | | | | | |
| プロセス | | ● | ● | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ■ | ■ | ■ | ■ | ■ | ■ | |
| | | 利害関係者要求定義 | システム要求分析 | システム方式設計 | システム実装 | ソフトウェア実装 | ソフトウェア要求事項分析 | ソフトウェア方式設計 | ソフトウェア詳細設計 | ソフトウェア構築 | ソフトウェア結合 | ソフトウェア適格性確認テスト | ソフトウェア受入れ支援 | ソフトウェア運用 | ソフトウェア保守 | ソフトウェア廃棄 | システム結合 | システム検証 | システム移行 | システム適格性確認 | システム運用 | システム保守 | システム廃棄 | |
| 超上流工程分析・評価 | | ◎ | ○ | ○ | △ | △ | △ | △ | △ | △ | △ | △ | △ | △ | △ | △ | △ | △ | △ | △ | | | | |
| 上流工程における横断的連携 | | ○ | ◎ | ◎ | ◎ | ◎ | ◎ | △ | △ | ○ | ○ | | | | | | ○ | ○ | △ | ○ | | | | |
| 要求管理 | | ◎ | ◎ | ◎ | ◎ | ◎ | ◎ | △ | △ | | | | | | | | | | | | | | | |
| システム構成関連技術 (部品化・連携) | | △ | ◎ | ◎ | ◎ | ◎ | ◎ | | | | | △ | △ | △ | | △ | △ | ○ | ○ | ○ | ○ | | △ | △ |
| 安全と情報セキュリティ | | ◎ | ◎ | ◎ | ◎ | ◎ | ◎ | ○ | ○ | △ | ○ | | | △ | △ | △ | △ | ○ | ○ | ○ | ○ | △ | △ | |
| 運用技術 | | ○ | ○ | | | | | | | | ○ | ○ | ○ | ○ | ○ | | | ○ | | ○ | ○ | ○ | ○ | |

注) 注視点:

(プロセス名は、JIS X0160:2012より)

4. 技術課題項目の整理と深掘調査:

(整理軸についての検討と整理、開発プロセス(2/2))

- 非ウォーターフォール型開発技術は、ソフトウェア実装にほぼ集中

| | | △ ひとつ以上が関係 ○ 過半数が関係 ◎ 全技術課題が関係 | | | | | | | | | | | | | | | | | | | | | | | |
|-----------------|---------|--|----------|----------|--------|----------|--------------|------------|------------|----------|----------|----------------|----------|-------------|----------|----------|----------|--------|--------|--------|-----------|--------|--------|--------|--|
| | | ●: ISO/IEC12207+15288, ○ISO/IEC12207, ■: ISO/IEC15288 | | | | | | | | | | | | | | | | | | | | | | | |
| | | ● | ● | ● | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ◎ | ■ | ■ | ■ | ■ | ■ | ■ | |
| プロセス | | 利害関係者要求定義 | システム要求分析 | システム方式設計 | システム実装 | ソフトウェア実装 | ソフトウェア要求事項分析 | ソフトウェア方式設計 | ソフトウェア詳細設計 | ソフトウェア構築 | ソフトウェア結合 | ソフトウェア適格性確認テスト | ソフトウェア導入 | ソフトウェア受入れ支援 | ソフトウェア運用 | ソフトウェア保守 | ソフトウェア廃棄 | システム結合 | システム検証 | システム移行 | システム適格性確認 | システム運用 | システム保守 | システム廃棄 | |
| 非ウォーターフォール型開発技術 | 要求の明確化 | ○ | ○ | △ | ○ | ◎ | ◎ | ○ | △ | ○ | | | | | | | | | | | | | | | |
| | 構成管理 | | | | | ◎ | △ | △ | △ | ◎ | | | | | | | | | | | | | | | |
| | 短期間での開発 | | | | | ◎ | ○ | ○ | ○ | ○ | △ | | | | | | | | | | | | | | |
| | 反復による開発 | | | | | ◎ | △ | △ | △ | ○ | | | | | | | | | | | | △ | | | |
| | 品質管理 | | | | | ◎ | | | △ | ◎ | | | | | | | | | | | | | | | |

注)注視点:

(プロセス名は、JIS X0160:2012より)

4. 技術課題項目の整理と深掘調査:

(整理軸についての検討と整理、超上流工程プロセス)

- 超上流工程の技術力で全体をコントロール
- 運用技術の継続的改善技術が超上流工程と運用工程を結ぶ技術
- 運用技術は上流工程以下の技術ともつながっているため、さらに超上流と上流以下を結ぶ技術

| | | 超上流工程分析・評価のみ: | | | | |
|---------------------|------------------|---|---|--|--|--|
| | | <input type="checkbox"/> 関連プロセス | <input checked="" type="checkbox"/> 対応プロセス | | | |
| | | <input checked="" type="checkbox"/> ひとつ以上が関係 | <input checked="" type="checkbox"/> 過半数が関係 | <input checked="" type="checkbox"/> 全技術課題が関係 | | |
| | プロセス | 戦略的技術計画 Strategic Technical Planning | ケーパビリティに 基づく開発の分析 Capability- Based Planning Analysis | 技術と標準動向調査 Technology and Standards Planning | エンタプライズ有効 性評価 Enterprise Evaluation and Assessment | ステークホルダ 分析 Stakeholder analysis |
| 超上流工程分析・評価 | システムシンキング | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| | ケーパビリティに基づく開発の分析 | | <input checked="" type="checkbox"/> | | | |
| | エンタプライズ有効性評価 | | | | <input checked="" type="checkbox"/> | |
| | 戦略的技術計画 | <input checked="" type="checkbox"/> | | | | |
| | 技術および標準化動向調査 | | | <input checked="" type="checkbox"/> | | |
| | ステークホルダ分析 | | | | | <input checked="" type="checkbox"/> |
| | ConOpsの分析/定義 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 上流工程における横断的連携 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 要求管理 | | | | | | <input checked="" type="checkbox"/> |
| システム構成関連技術(部品化・連携) | | | | <input checked="" type="checkbox"/> | | |
| 安全と情報セキュリティ | | | | <input checked="" type="checkbox"/> | | <input checked="" type="checkbox"/> |
| 運用技術 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 非ウォーターフォール型 開発技術 | 要求の明確化 | | | | | |
| | 構成管理 | | | | | |
| | 短期間での開発 | | | | | |
| | 反復による開発 | | | | | |
| | 品質管理 | | | | | |

注)注視点:

4. 技術課題項目の整理と深掘調査:

(整理軸についての検討と整理、運用プロセス)

- 継続的サービス改善プロセスとサービスオペレーションプロセスに集中
- 継続的サービス改善プロセスが超上流工程技術との強い関係性

| | | <input type="checkbox"/> △ ひとつ以上が関係 | <input type="checkbox"/> ○ 過半数が関係 | <input checked="" type="checkbox"/> ◎ 全技術課題が関係 | | |
|---------------------|---------|-------------------------------------|-----------------------------------|--|-------------|--------------|
| ITIL V3 フェーズ | | 継続的サービス改善 | サービス・ストラテジ | サービス・デザイン | サービス・トラジション | サービス・オペレーション |
| 超上流工程分析・評価 | | ◎ | △ | △ | △ | △ |
| 上流工程における横断的連携 | | △ | | | | △ |
| 要求管理 | | ○ | | | | △ |
| システム構成関連技術(部品化・連携) | | △ | | | | |
| 安全と情報セキュリティ | | ○ | △ | | | ○ |
| 運用技術 | | ◎ | ◎ | ◎ | ◎ | ◎ |
| 非ウォーターフォール型 開発技術 | 要求の明確化 | △ | | △ | | △ |
| | 構成管理 | | | | | |
| | 短期間での開発 | △ | | △ | | △ |
| | 反復による開発 | △ | | △ | | △ |
| | 品質管理 | | | | | |

注)注視点: 

4. 技術課題項目の整理と深掘調査:

(整理軸についての検討と整理、アーキテクチャ開発手法プロセス)

- アーキテクチャ開発手法のプロセスに多くの技術が貢献の可能性
- 逆にアーキテクチャ開発手法は出来上がった手法のため、超上流工程プロセスの評価・分析手法の整合性について検討が必要

| | | <input type="checkbox"/> ひとつ以上が関係 | <input type="checkbox"/> 過半数が関係 | <input checked="" type="checkbox"/> 全技術課題が関係 | | | | | | | | | |
|---------------------|---------|-------------------------------------|-------------------------------------|--|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|--------------------------|
| TOGAF-ADM | | 初期フェーズ | アーキテクチャビジョン | ビジネスアーキテクチャ | 情報システムアーキテクチャ | 技術アーキテクチャ | ソリューション | 移行計画 | 実装監督 | アーキテクチャ変更管理 | 要求管理 | | |
| 超上流工程分析・評価 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 上流工程における横断的連携 | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 要求管理 | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| システム構成関連技術(部品化・連携) | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 安全と情報セキュリティ | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 運用技術 | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 非ウォーターフォール型 開発技術 | 要求の明確化 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| | 構成管理 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| | 短期間での開発 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| | 反復による開発 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 品質管理 | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

注)注視点: 

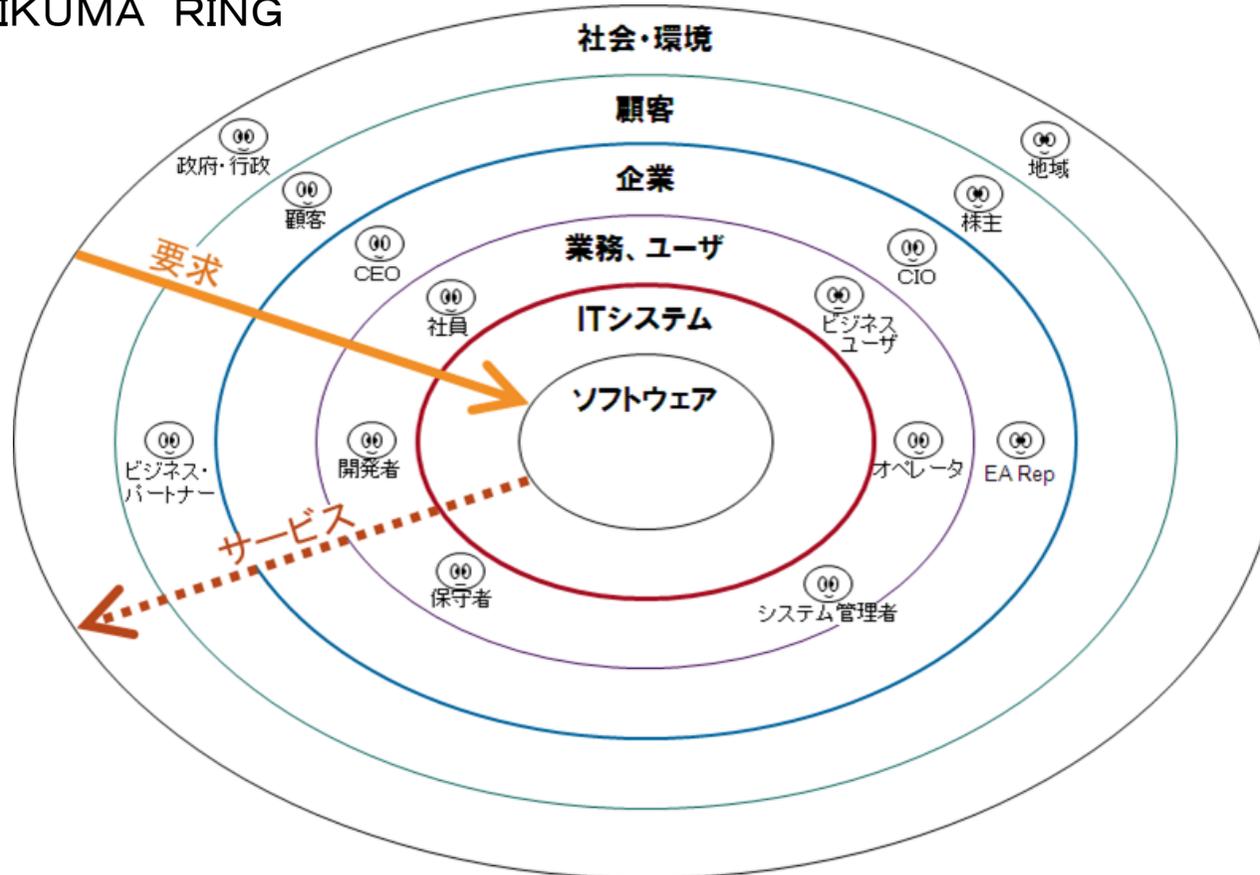
4. 技術課題項目の整理と深掘調査 － ステークホルダー要求 －

4. 技術課題項目の整理と深掘調査:

(整理軸についての検討と整理、ステークホルダー要求)

• 様々なステークホルダーがあり、階層的な整理が有効

KIKUMA RING



ITアーキテクチャサミット 2008, 日本IBM榊原氏資料を一部改編

4. 技術課題項目の整理と深掘調査:

(ステークホルダー(用語の違い))

- 「ステークホルダー要求」は情報システムの変化の主要原因として認識、しかし標準間で相違点が多い
- 複数の標準を併用する場合が増加しており、併用する際はこのような比較表が必要であり重要

【共通フレームの分類をベースに各知識体系、フレームワーク間の関係を比較】

| ステークホルダー 大分類 | 共通フレーム | | BABOK | | REBOK | | ITIL | TOGAF | | PMBOK | KIKUMA RING |
|-----------------|-------------------|-----------------------------------|---------------------------|-------------------------------------|----------|--|---|--------------|--|---|---------------------------------|
| | 部署等 / 役割 (ロール) | | ステークホルダー | | ステークホルダー | | 利害関係者 | ステークホルダー | | ステークホルダー | ステークホルダー |
| | 分類 | 対象 | 分類 | 対象 | 分類 | 対象 | 対象 | 分類 | 対象 | 対象 | 対象 |
| 外部 | | | 影響を受ける外部 | 顧客 規制者 サプライヤ | | | 事業顧客/外部顧客 マスコミ 政府/規制機関 労働組合 株主 事業パートナー | 外部 | 規制機関 調達先 | 顧客 | 政府行政 地域 顧客 サプライヤ 株主 |
| 経営層 | 経営層 | 社長 担当役員 | 組織や企業 | スポンサー 経営幹部 | ユーザ | 経営者(CIO) | トップ・マネジメント | 本社機能 | CxO | スポンサー | CEO,CIO, EA Rep |
| 管理部門 | | | | | | | 監査人 | 本社機能 | エンタプライズセキュリティ プログラムマネジメント 品質保証/標準グループ 調達/人事 | 影響力者 | |
| 業務部門 | 業務部門 | 部門長 業務推進担当 システム推進担当 関連会社 | 組織や企業 影響を受ける 組織ユニット | ドメインのSME ビジネスアナリスト エンドユーザー | ユーザ | エンドユーザー | 内部顧客 ユーザ 事業領域/事業部門 | エンドユーザ 組織 | 役員 部門管理者 ビジネス・ドメイン専門家 データ所有者 | ユーザー | 社員 ビジネスユーザ |
| 情報システム 部門 | 情報システム部門 | 部門長 システム開発担当 システム子会社 | ソリューションの デリバリー | プロジェクトマネ ジャー 実装のSME テスト担当者 | ユーザ | プロジェクトマネ ジャー システム責任者 | 内部サービスプロバイ ダ プログラムとプロジェクト のチーム | プロジェクト 組織 | 役員 部門管理者 ビジネス・プロセス専門家 プロセス生成物専門家 技術専門家 | プロジェクト・マネージャー 母体組織 プロジェクト・チーム・メンバー プロジェクトマネジメント・チーム PMO | 開発者 保守者 システム管理者 |
| システム 運用部門 | | | 影響を受ける 組織ユニット | 運用サポート ヘルプデスク | | | 運用スタッフ | システム 運用 | ITサービス管理者 サービス・デスク アプリケーション管理者 インフラストラクチャ管理者 データ/音声コミュニケーション | | オペレータ |
| ベンダ | ベンダ | 元請けベンダ アウトソーサ サブベンダ | | | ベンダ | ベンダ側経営者 プロジェクトマネ ジャー 上級システムエン 지니어 ソフトウェア開発者 | サプライヤ(製品) サプライヤ(サービス) サードパーティー | | | | |

注) ITILの情報はITIL V3(2007年版)のコア書籍STのp.174図5.5および図5.6からの引用である

4. 技術課題項目の整理と深掘調査:

(要求の変化とステークホルダーの関係例)

■JUAS調査におけるIT投資、保守作業、ITIL適用理由を参考に要求目的例を作成。

| 要求目的例 (IT投資効果) | ①JUAS企業IT動向調査2011 | ②ソフトウェアメトリクス調査2011 | ②ソフトウェアメトリクス調査2009 | ステークホルダー大分類 | | | | | |
|---------------------------------------|--|--------------------------------------|-------------------------------------|-------------|-----|------|------|-------|------|
| | Q3-1.IT投資で解決したい中期的な経営課題 | Q3.2.保守作業の発生目的 | Q6.2 ITIL適用活用目的 | 外部 | 経営層 | 管理部門 | 業務部門 | 情シス部門 | 運用部門 |
| 企業の社会的責任 (CSR向上) | 13.企業としての社会的責任の履行 (セキュリティ確保、個人情報の保護等) | | | ○ | ○ | ○ | | | |
| 法令順守・業界標準への準拠 | | 制度ルール変化 | 4.遵法あるいは認証取得 | ○ | ○ | | ○ | ○ | |
| 経営の透明性確保 | 12.経営の透明性の確保 (内部統制、システム監査への対応等) | | 3.IT統制,ITガバナンスの強化 | | ○ | ○ | | | |
| 経営スピード向上 | 1.迅速な業績把握、情報把握 (リアルタイム経営) | 経営目標変化 | 1.経営あるいは事業からの要求 | | ○ | ○ | ○ | | |
| 市場シェア拡大 | 3.グローバル化への対応 10.ビジネスモデルの変革 | | 2.グローバル標準の採用 | | ○ | | ○ | | |
| 顧客の確保、維持 | 2.顧客重視の経営 11.営業力の強化 | | | | ○ | | ○ | | |
| 業務効率向上 (省力化、業務コスト削減) | 4.社内コミュニケーションの強化 6.IT開発・運用のコスト削減 7.業務プロセスの効率化 (省力化、業務コスト削減) | 業務方法変化 ユーザビリティ変化 担当者要望 | 8.ITサービス管理の効率化 9.ITサービスコストの削減 | | ○ | ○ | ○ | ○ | ○ |
| 業務スピード向上 (リードタイム短縮等) | 5.企業間(グループ、業界、取引先間)の情報連携 8.業務プロセスのスピードアップ (リードタイム短縮等) | | 6.部門間連携の強化 | | ○ | | ○ | | |
| 業務品質・精度の向上 | 9.業務プロセスの質・精度の向上 (ミス、欠品削減等) | システムバグ | 5.プロセス、ルールの明確化 7.ITサービスの品質・信頼性向上 | | | | ○ | ○ | ○ |
| 製品・サービスの開発、改善 | | | 10.IT運用に関する特定の問題解決 | | | | | | ○ |
| ITインフラ変更への対応 (製品保守、データ量対応、最新技術の活用) | | データ量の変化 ハード・モデル変更への対応 OS変更への対応 | | | | | | ○ | ○ |

4. 技術課題項目の整理と深掘調査:

(要求の変化と技術課題)

- ステークホルダー要求と技術課題を関係づけることを例示、さらに品質特性で分類
- 品質特性による整理は例示
- 具体的なシステム開発事例に適用することで特徴を表現

例

| 要求(変化の原因) | ステークホルダー | 技術課題 | |
|---------------------------------------|----------|--------------|-------------------------|
| 企業の社会的責任(CSR向上) | システム部門 | 品質保証 | 共通基盤としての保証 |
| 法令順守・業界標準への準拠 | システム部門 | 品質保証 | 他者に求められる保証 |
| 経営の透明性確保 | 経営企画部門 | ESE | ケーパビリティに基づく開発に関する分析 |
| 経営スピード向上 顧客の確保、維持 | システム部門 | 上流工程 | 機能に基づくSE手法活用 |
| 市場シェア拡大 | システム部門 | システム 構成技術 | 構造の明確化 |
| 業務効率向上(省力化、業務コスト削減) | システム部門 | 運用技術 | ITIL活用 |
| 業務品質・制度の向上 | 経営企画部門 | ESE | ステークホルダー分析 |
| 製品・サービスの開発、改善 | 経営企画部門 | ESE | ConOpsの分析/定義 |
| ITインフラ変更への対応(製品保守、 データ量対応、最新技術の活用) | システム部門 | 要求管理 | コントロールケースによる 非機能要求記述 |

4. 技術課題項目の整理と深掘調査:

(ステークホルダー要求: 品質要求と制約(非機能要求))

- 同様に具体的なシステム開発に対して、その特徴を抽出する手段のひとつとして活用
- 以下に例示(具体的なシステム事例ではない)

△ ひとつ以上が関係 ○ 過半数が関係 ◎ 全技術課題が関係

例

| 技術課題 | ステークホルダー要求 | | 要求分類 | | | | | EA | ISO25010システム/ソフトウェア品質特性 | | | | | | | | | | | 制約 | | |
|-----------------|--|--------------------------------|------------|------|----------|-----------|-------|--|-------------------------|-------|-----|-----|-----|--------|-----|-----|-----|-----|-----|----|--------|------------|
| | 要求例 | ステークホルダー | アプリケーション機能 | 技術機能 | ハードウェア機能 | アーキテクチャ要求 | IT外要求 | ③ ① ④ ② ① ④ ② ① ③ ④ | 機能適合性 | 性能効率性 | 互換性 | 使用性 | 信頼性 | セキュリティ | 保守性 | 移植性 | 有効性 | 効率性 | 満足性 | | リスク回避性 | コンテクトカバレッジ |
| 超上流工程分析・評価 | モデリングを活用した迅速な開発 企業の社会的責任(CSR向上) 業務品質・精度の向上 製品・サービスの開発、改善 ITインフラ変更への対応(製品保守、データ量対応、最新技術の活用) | 経営企画部門 システム部門 | | | | | | ① ④ | ○ | ○ | △ | ○ | ○ | ○ | ○ | △ | ○ | △ | ○ | ◎ | | ○ |
| 上流工程における横断的連携 | 経営スピード向上 顧客の確保、維持 迅速な開発、顧客確保、維持 業務効率向上(省力化、業務コスト削減) 業務品質・精度の向上 | システム部門 | ◎ | △ | ○ | ○ | | ① ③ | ◎ | ○ | △ | | | | ◎ | △ | | | | | | |
| 要求管理 | 市場シェア拡大 経営スピード向上 顧客の確保、維持 ITインフラ変更への対応(製品保守、データ量対応、最新技術の活用) | サービス提供企業 サービス主管部門 システム部門 | ○ | △ | △ | ○ | △ | ① ③ | ○ | △ | | | △ | △ | ○ | △ | | | | | △ | |
| システム構成関連技術 | 市場シェア拡大 経営スピード向上 | システム部門 | ◎ | | | ◎ | | ③ ④ | ○ | △ | △ | | | ○ | ◎ | △ | | | | | △ | ○ |
| 安全と情報セキュリティ | 企業の社会的責任(CSR向上) 法令順守・業界標準への準拠 | サービス提供企業 システム部門 | ○ | △ | △ | ○ | ○ | ① ③ ④ | △ | | | | ○ | ○ | △ | | | | | △ | △ | △ |
| 運用技術 | 経営スピード向上 | システム部門 | ◎ | | | | ◎ | ① | | | | | | | | | ◎ | ◎ | ◎ | ◎ | | |
| 非ウォーターフォール型開発技術 | 業務品質・精度の向上 製品・サービスの開発、改善 | システム部門 | ◎ | | | | | ④ | ◎ | △ | | △ | △ | | △ | | | | ○ | | | |

4. 技術課題項目の整理と深掘調査:

(ステークホルダー要求: アーキテクチャと構成要素)

- 同様に具体的なシステム開発に対して、その特徴を抽出する手段のひとつとして活用
- 以下に例示(具体的なシステム事例ではない)

(次の情報を利用: IBM: Patterns for e-business (<http://www.ibm.com>) <https://www.ibm.com/developerworks/patterns/>)

△ ひとつ以上が関係 ○ 過半数が関係 ⊙ 全技術課題が関係

| 例 | 技術課題 | 要求例 | 静的構造 | | | | 動的振舞 | | | | 要求分類 | | | | |
|---|-----------------|---|-----------|-------------------|------|---------------|------------|------------|-------------|---------------|----------------|------|----------|-----------|-------|
| | | | ステークホルダ要求 | アプリケーションとプレゼン層の分離 | 直接接続 | ハブ・アンド・スポーク接続 | 更新データのハブ配置 | プレゼン層の同期統合 | プレゼン層の非同期統合 | アプリケーションの同期統合 | アプリケーションの非同期統合 | 技術機能 | ハードウェア機能 | アーキテクチャ要求 | IT外要求 |
| | 超上流工程 | e-business向けシステムの効率的構築 TCOの最小化と保守性の向上 バックエンドアプリケーションの統合化 迅速なビジネス開始 複数のデリバリー・チャネルの統合 顧客から見たビューの統一 | ⊙ | △ | ○ | △ | ○ | ○ | ○ | △ | | | | | |
| | 上流工程連携 | バックエンドアプリケーションの統合化 迅速なビジネス開始 ツール活用による開発コスト削減と早期開発 | ⊙ | △ | ⊙ | ⊙ | ○ | ⊙ | ○ | ⊙ | ⊙ | △ | ○ | ○ | △ |
| | 要求管理 | バックエンドアプリケーションの統合化 保守性の向上 一括カスタマイズ | ⊙ | △ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ⊙ | ○ |
| | システム構成技術 | 保守性の向上 ITインフラ変更への対応(製品保守、データ量対応、最新技術の活用) | ⊙ | | ○ | ○ | ○ | ○ | ○ | ⊙ | | | | ⊙ | |
| | 安全と情報セキュリティ | 法令順守・業界標準への準拠 企業の社会的責任(CSR向上) | ⊙ | | ⊙ | ⊙ | ⊙ | ○ | ○ | ○ | △ | △ | ○ | ○ | |
| | 運用技術 | 経営スピード向上 | ⊙ | | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | | | | | ⊙ |
| | 非ウォーターフォール型開発技術 | 業務品質・精度の向上 製品・サービスの開発、改善 | ⊙ | △ | ○ | ○ | ○ | ○ | △ | △ | ⊙ | | | | |

5. 技術課題項目の分析

● 超上流工程プロセスによる分析結果より

- 全体をコントロールする技術である超上流工程の技術力強化は、IT技術の効果的利用による産業力強化に貢献
- 運用技術の技術課題「継続的改善」は超上流工程と一体化できる可能性
- 運用技術の技術課題「継続的改善」が、超上流工程と上流工程を中心とした開発プロセスをつなぐ役割を担って「ビジネス-ITオペレーション」を実現

● アーキテクチャ開発手法プロセスによる分析結果より

- アーキテクチャ開発手法のプロセスは、それ自身全体体系として存在
- 本来目的との適合性・整合性を確認しながら、アーキテクチャ開発手法の発展をウォッチすることが必要

● 開発プロセスによる分析結果より

- 開発技術の観点からは上流工程の技術が重要
- このことは信頼性の高い情報システムの構築のケースと類似
- 非ウォーターフォール型開発技術はソフトウェア構築に集中して活用する技術

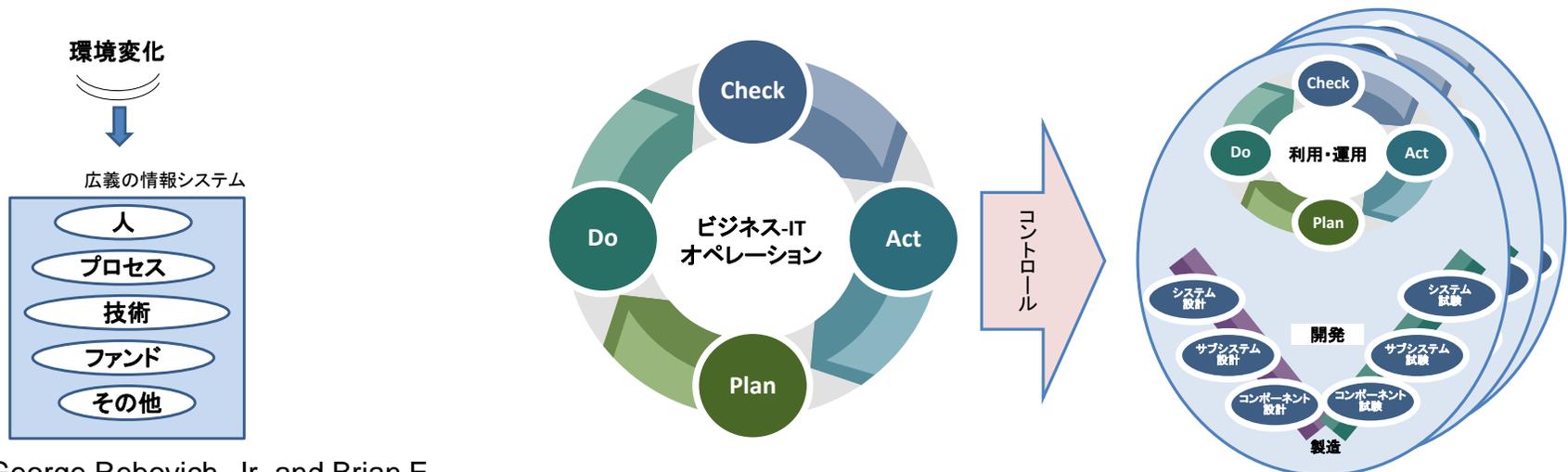
● アーキテクチャ要求による分析結果より

- ステークホルダーの扱いの観点から、運用について重点を置いていない標準類あり
- PMBOKのようなポートフォリオマネージャ、プログラムマネージャを定義している標準類と、今回洗い出した超上流工程の技術課題との関係性については検討の余地があり、特に要求管理者としての位置づけの有無にも留意が必要

5. 技術課題の分析

(超上流工程技術の展開に向けた分析)

- 環境の変化の影響を受けるのは人(ステークホルダー)だけではなく、プロセス、技術、ファンドまで(報告書では広義の情報システムと呼んだ)
- 総合してIT利活用社会に起こる環境の変化に呼応して要求の変化が起こり、情報システム群に対する要求の変化をコントロールしながら適用(outer loop)
- 一方で、情報システム構築は変化に対応して継続開発(inner loop)
- 全体を分析



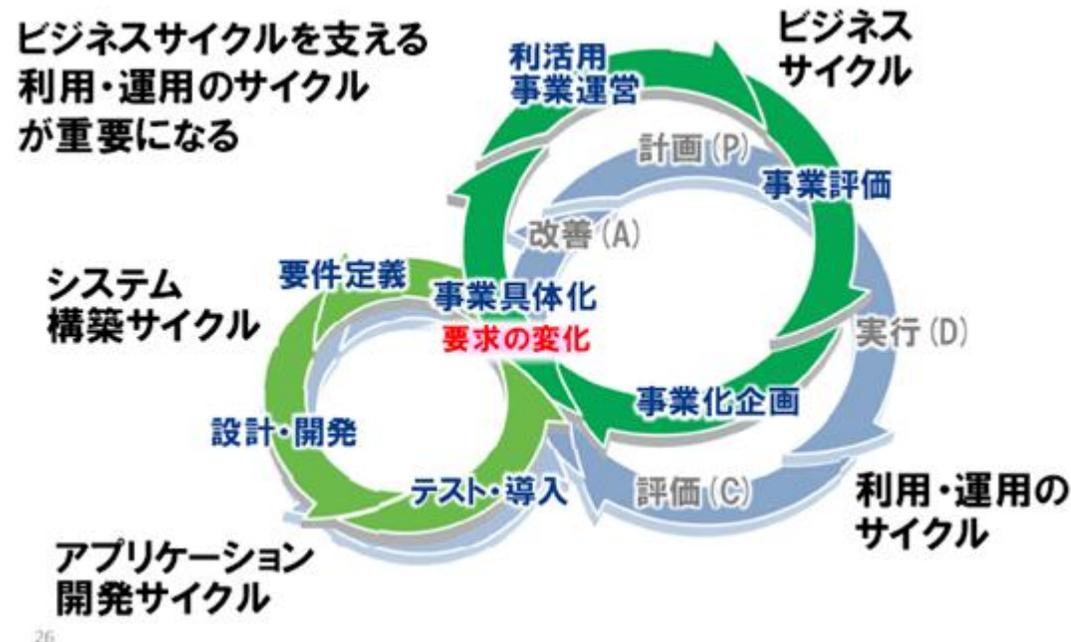
参考: George Rebovich, Jr. and Brian E. White: ENTERPRISE SYSTEMS ENGINEERING Advances in the Theory and Practice, CRC Press, 2010.

ビジネス-ITオペレーションは、情報システム群をコントロール

5. 技術課題の分析

(要求の変化を前提とするシステム像の分析)

- 例えば下図では、要求の変化を前提とするシステムは、ビジネスサイクルと利用・運用サイクル、システム構築・アプリケーション開発サイクルの三位一体で実現
- 超上流工程であるビジネスサイクルの一部で全体をコントロールする
- 超上流工程での分析や意思決定のためにはケーパビリティの分析が非常に重要

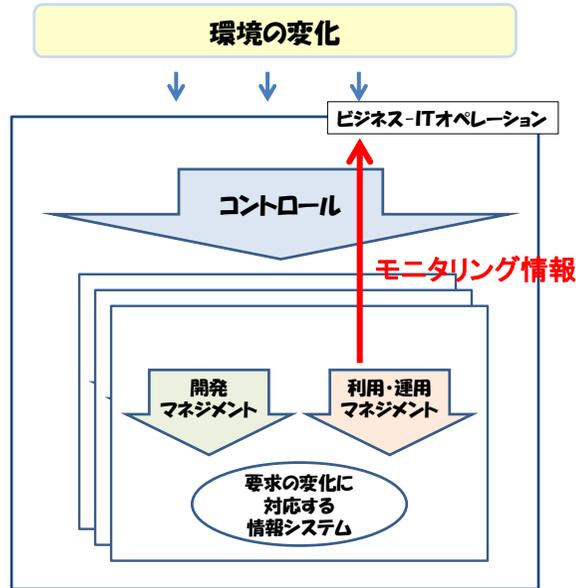


成瀬WG委員提供資料(富士通「FUJITSU SI」のコンセプト 2win cycle model)を基に一部修正

5. 技術課題の分析

(超上流工程分析評価技術)

- 超上流工程は、情報システム群の要求の変化をコントロールするのが役割



outer loop として関連組織の
情報システム群からの情報と
ビジネスモニタリング結果を分析

- 「ビジネス-ITオペレーション」は、運用技術の継続的サービス改善を介して、上流工程以降や利用・運用側と接続
- ビジネス側を通して環境の変化をモニタリング
- 利用・運用側を通してシステムの状況をウォッチし、環境の変化をモニタリングし、その結果を「ビジネス-ITオペレーション」で分析

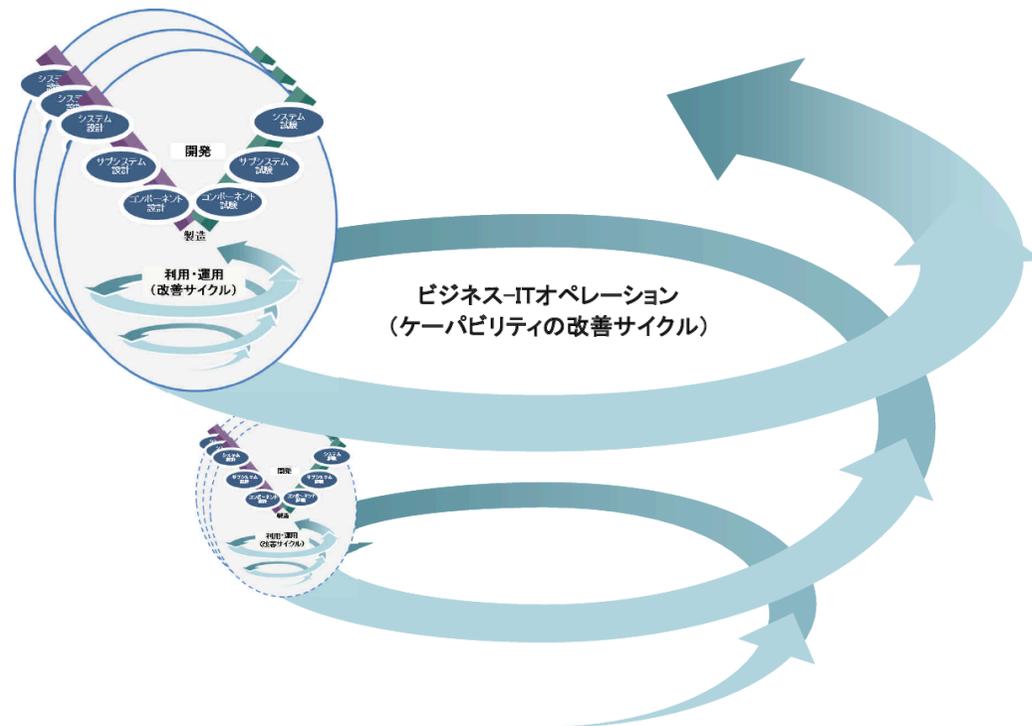
要求の変化に関する超上流工程の活動例

- 環境の変化をモニタリングし、要求の変化を分析、そして組織・システム・オペレーションの各ケーパビリティ(能力)と照合して変化の範囲がその能力の範疇にあるかを確認
- 能力を超えていた場合は、ケーパビリティの改善活動
- その結果を利用・運用プロセスの継続的改善活動を通して、各システムに伝達して各システムの要求の変化に対応した活動を実施
- 利用・運用プロセスからは、逆にシステムを通して得たモニタリング結果を超上流工程のプロセスへ送信

(outer loopの改善サイクル)

- ケーパビリティを中心とした改善サイクル(outer loop)とサービスとシステムの改善サイクル(inner loop)
- システム(利用・運用)のinner loop の改善サイクルよりモニタリング情報を、outer loop が獲得
- inner loop から得たモニタリング情報等を活用して、outer loopの改善サイクルにコントロールされて、inner loopの改善サイクルをマネジメント

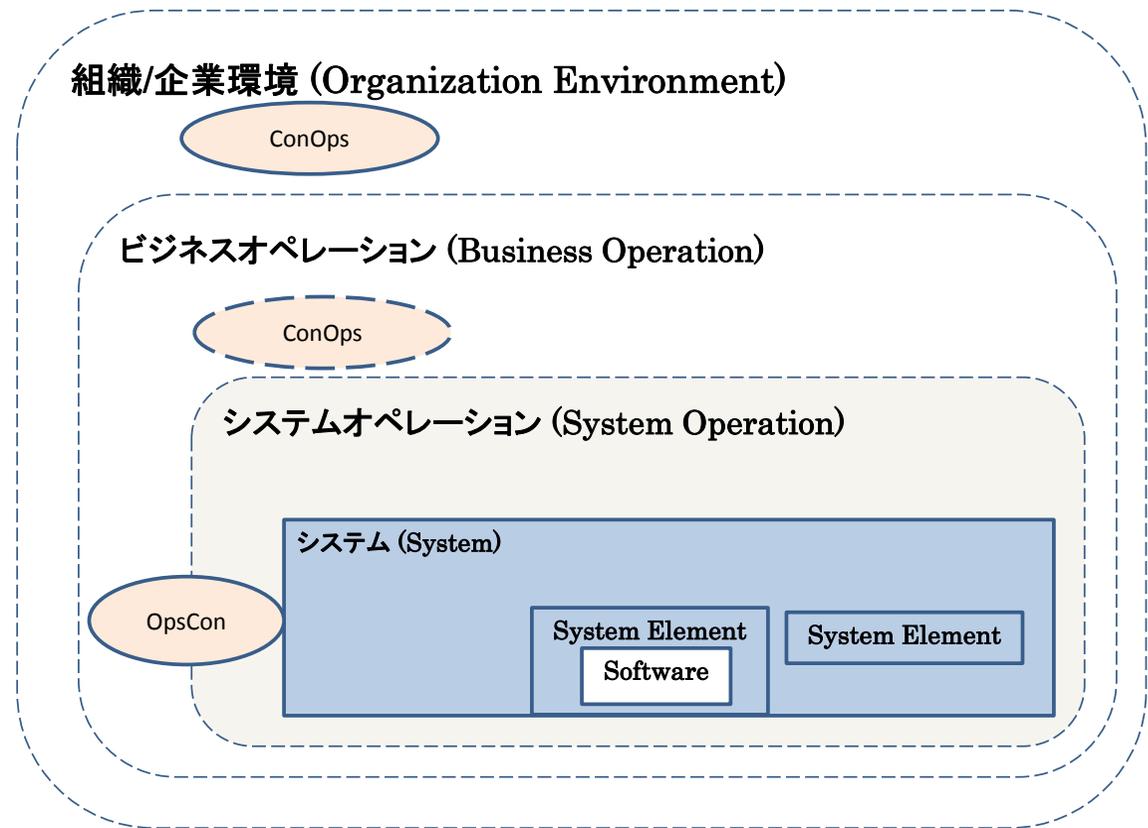
利用・運用を通じたモニタリング結果に基づくコントロール



5. 技術課題の分析

(要求の変化を前提とするシステム像の分析)

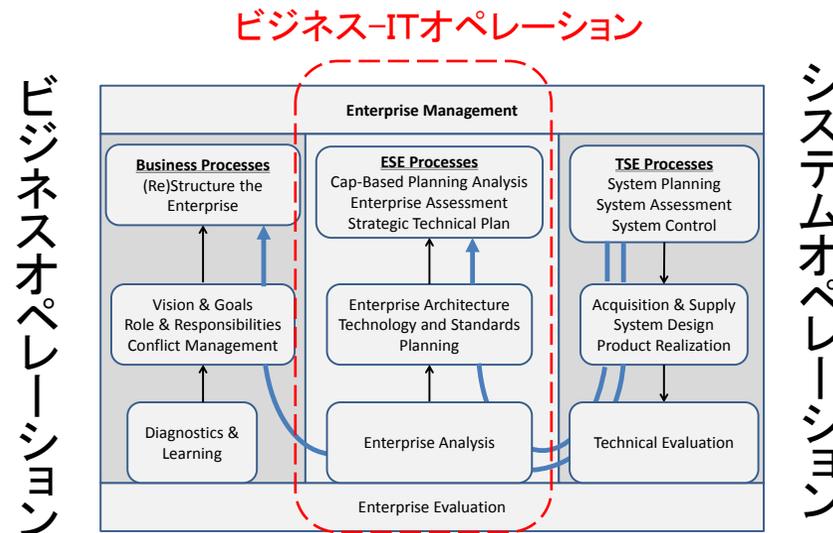
- 右図のフレームでの開発をイメージ
 - すべてがConOps起点である。
 - システムオペレーションはビジネスオペレーションがConOpsに基づいて要求の変化に対応
 - システム構築とオペレーションの構築の双方がConOps(もしくはOpsCon)の下で実施される。



参考: ISO/IEC/IEEE29148:2011 Systems and software engineering – Life cycle processes – Requirements engineering (Annex A System operational concept, Annex B Concept of operations)

要求の変化に対応する超上流工程に関する課題:

- 要求の変化をコントロールするのは、ビジネス部門だけでもシステム部門だけでも困難
- 要求を管理できる共通部門や委員会を設置するなどの対応が必要



参考: George Rebovich, Jr. and Brian E. White: ENTERPRISE SYSTEMS ENGINEERING Advances in the Theory and Practice, CRC Press, 2010.

6. まとめ

6. まとめ(今後に向けた課題について(1/5))

「要求の変化」という課題にとって重要となる技術項目を調査し、技術課題をまとめた。
以下、課題をまとめる。

● 超上流の活動について

- 本報告書で「ビジネス-ITオペレーション」と名付けた「要求の変化に係る超上流を中心とした活動(outer loop と呼んだ)」は、その在り方が課題となるケースがある。
- 活動の位置づけがビジネスからシステムまで及ぶために、ある意味企業の共通部門的な活動位置を設ける必要がある。さらに検討が必要である。
- 変化する要求の管理者もしくは管理するシステムの位置づけも同様な問題となる。
- ケーパビリティ分析を訴求することで、実現した情報システムと組織能力とのミスマッチを早急に避けるようにすべきである。また、当該技術を担う技術者の育成が求められる。
- PMBOKのポートフォリオマネージャやプログラムマネージャのBOKと当該技術との関係性を明らかにする必要がある。

6. まとめ(今後に向けた課題について(2/5))

● 運用技術と超上流技術

- 運用技術の継続的サービス改善は超上流工程と開発工程を結ぶ技術として位置づけることが可能である。
- つなぐのが役目であるので適切と考えるが、一方で、具体的な情報のやりとりのモデルについては今後の検討課題である。
- 超上流工程からトップダウン的に情報システムの継続性と発展をコントロールするという仮説は、運用技術によりボトムアップアプローチでの情報システム状況のモニタリング情報を超上流工程へ渡すことができることを確認したことで妥当性が得られたと考える。従って、トップダウンかボトムアップかの選択ではなく、両者の統合が重要になる。
- ビジネスも含む運用要求の分析と運用実施の両面ができる技術者育成が求められる。

6. まとめ(今後に向けた課題について(3/5))

● 要求の変化と様々な階層性について

- ステークホルダー要求の解説の際に掲載したKIKUMA RINGの階層性や、TOGAFのビジネスアーキテクチャ、情報アーキテクチャ、技術アーキテクチャ階層がある。このような階層性がなぜ必要とされるか、それらがどのように異なり、要求の変化の観点からどのような共通性があるかを今後明らかにする必要がある。
- 論文「Israel Navarro, Nancy Leveson, Kristina Lunqvist, Semantic decoupling: reducing the impact of requirements change, Requirements Engineering, Springer, 03 June 2010」にみられる管理、目的、設計原則、アーキテクチャ、設計仕様、物理仕様、運用という7階層に対して、サービス環境、サービス、運用者、サービス、コンポーネント、サービス確認でどのような「サービス要求水準」が必要になるかを次ページの表で整理している。このような表で要求変化を特定し、影響範囲を限定できると考えているようである。どこで変化が発生するかを識別するために、このような階層性が利用できると考える。

6. まとめ(今後に向けた課題について(4/5))

● 要求の変化と様々な階層性について(続き)

| 水準 | 分類 | サービス環境 | サービス運用者 | サービスコンポネント | サービス確認 |
|----|---------|---------|---------|------------|---------|
| 0 | 管理 | 管理計画 | | | 点検結果 |
| 1 | 目的 | 前提, 制約 | 責任分担 | 要求, 制約 | 充足性評価 |
| 2 | 設計原則 | インタフェース | 作業分担 | 機能分担 | 分担性評価 |
| 3 | アーキテクチャ | 環境モデル | 運用モデル | 機能モデル | モデル確認 |
| 4 | 設計仕様 | I/F設計 | HCI設計 | サービス仕様 | サービステスト |
| 5 | 物理仕様 | | 物理設計 | コード | テスト |
| 6 | 運用 | 監査手順 | 運用手順 | 障害報告 | 運用性確認 |

Israel Navarro, Nancy Leveson, Kristina Lunqvist: Semantic decoupling: reducing the impact of requirements change, Requirements Engineering, Springer, 03 June 2010.
の表を一部修正

- 変化対応技術ごとに、前提とする階層性が異なる可能性があり、今後、変化対応階層のメタモデルを明らかにするとともに、相互関係を整理することにより、変化対応技術の統合化が期待される。

6. まとめ(今後に向けた課題について(5/5))

● その他

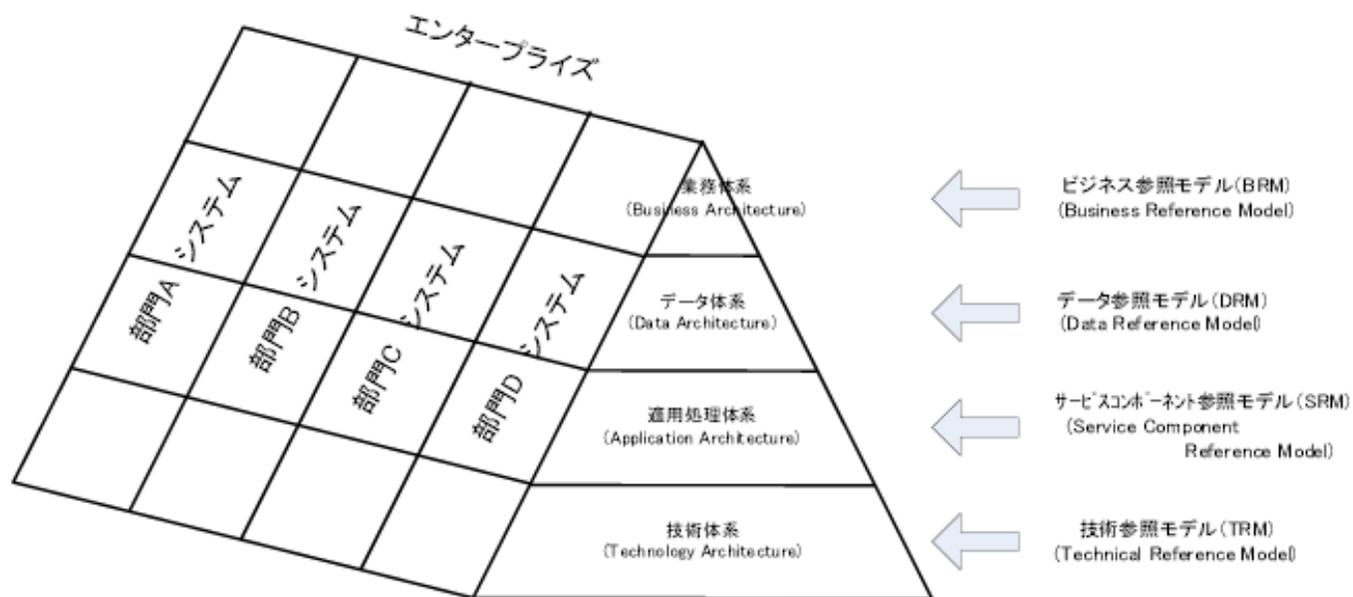
- システムズエンジニアリングの技術に限定すると、宇宙・軍事関係の事例に基づくものを多く採用している。我が国では民生関係にどう適用していくのか、コスト面、人材面を含めて検討が必要である。
- ビジネス-ITオペレーション視点での全体活動の事例の収集が望まれる。
- 本技術課題は部分的には情報システム構築のための実践活動が含まれていると考える。
- システムズエンジニアリングの事例は宇宙・軍事関係に限定されているので、民生用情報システム開発への適用性を明らかにする必要がある。また一方でTOGAFや当該テーマに適合する運用が為されているITIL V3等についても含めて、全体事例としてまとめ、さらにその相互関係を明らかにすることで、当該技術を訴求することが望まれる。

付録

次のアーキテクチャフレームワークについて説明する。

- 経済産業省のEA
- Zachmanのアーキテクチャフレームワーク
- TOGAF
- DODAF

(1) 経済産業省EAフレームワーク



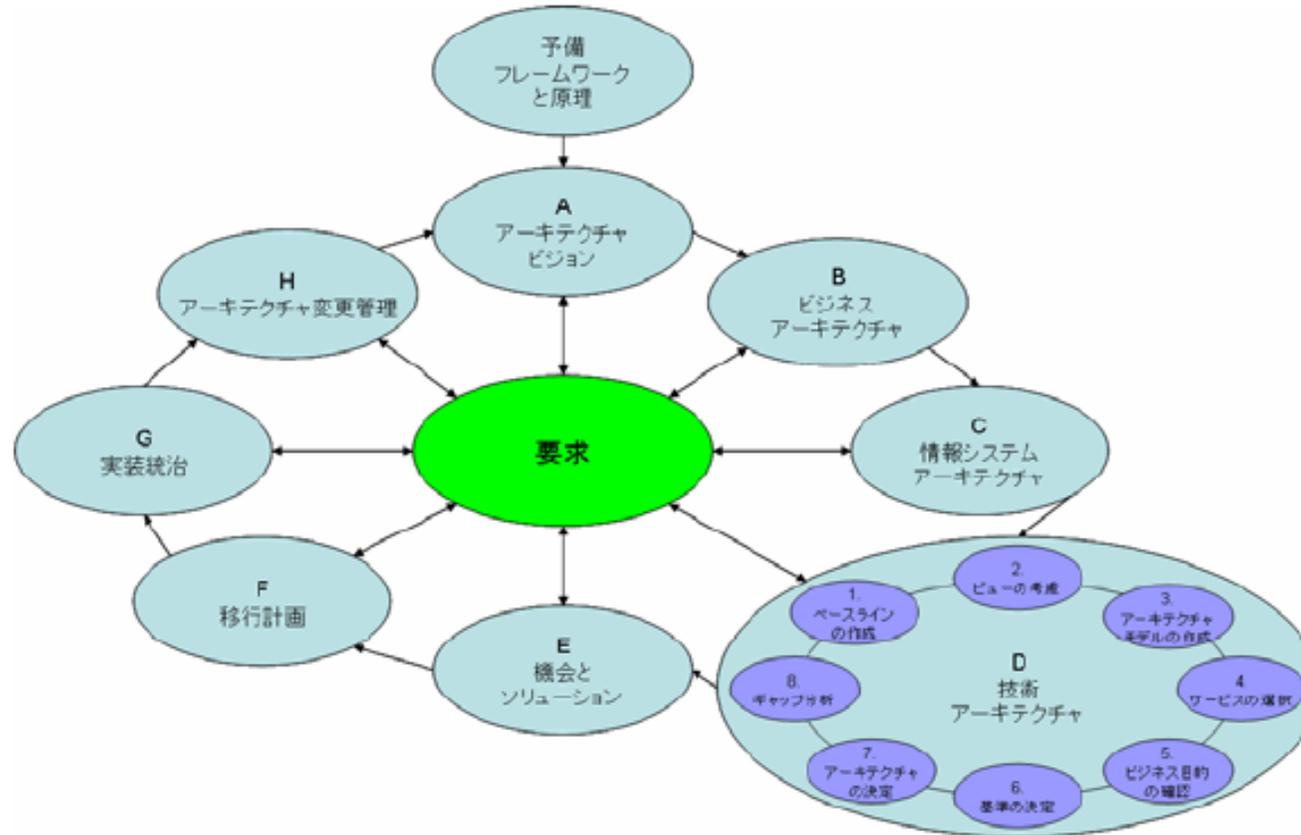
出典： 参照アーキテクチャ調査報告, IPA, 2005.
(日経BP: 業務・システム最適化計画参照モデル活用ガイド, 2006)

(2) Zachman フレームワーク

| | what -data (データ) | how -function (機能) | where -network (場所) | who -people (組織・人) | when -time (時間) | why -motivation (動機) |
|-------------------------------|------------------------|--------------------------|---------------------------|--------------------------|-----------------------|----------------------------|
| Planner (計画立案者) | ビジネス重要 項目の列挙 | ビジネス プロセスの列挙 | ビジネス物流 拠点の列挙 | ビジネス重要な 組織の列挙 | 周期・行事・重 要イベント列挙 | ビジネスの 目標と戦略 |
| Owner (利用者) | 意味的モデル | ビジネス プロセスモデル | 拠点間 物流システム | ワークフロー モデル | マスター スケジュール | ビジネス計画 |
| Designer (設計者) | 論理データ モデル | アプリケーショ ン構造 | 分散システム 構造 | HMI構造 | 処理順序構造 | ビジネスルール モデル |
| Builder (開発者) | 物理データ モデル | システム設計 | テクノロジー構造 | 表現構造 | 制御構造 | ルール設計 |
| Subcontractor (作業員) | データ定義 | プログラム | ネットワーク構 造 | セキュリティ 構造 | 処理タイミング | ルール定義 |
| FunctioningSystem (稼動システム) | データ | 機能 | ネットワーク | 組織 | スケジュール | 戦略 |

出典：参照アーキテクチャ調査報告, IPA, 2005.

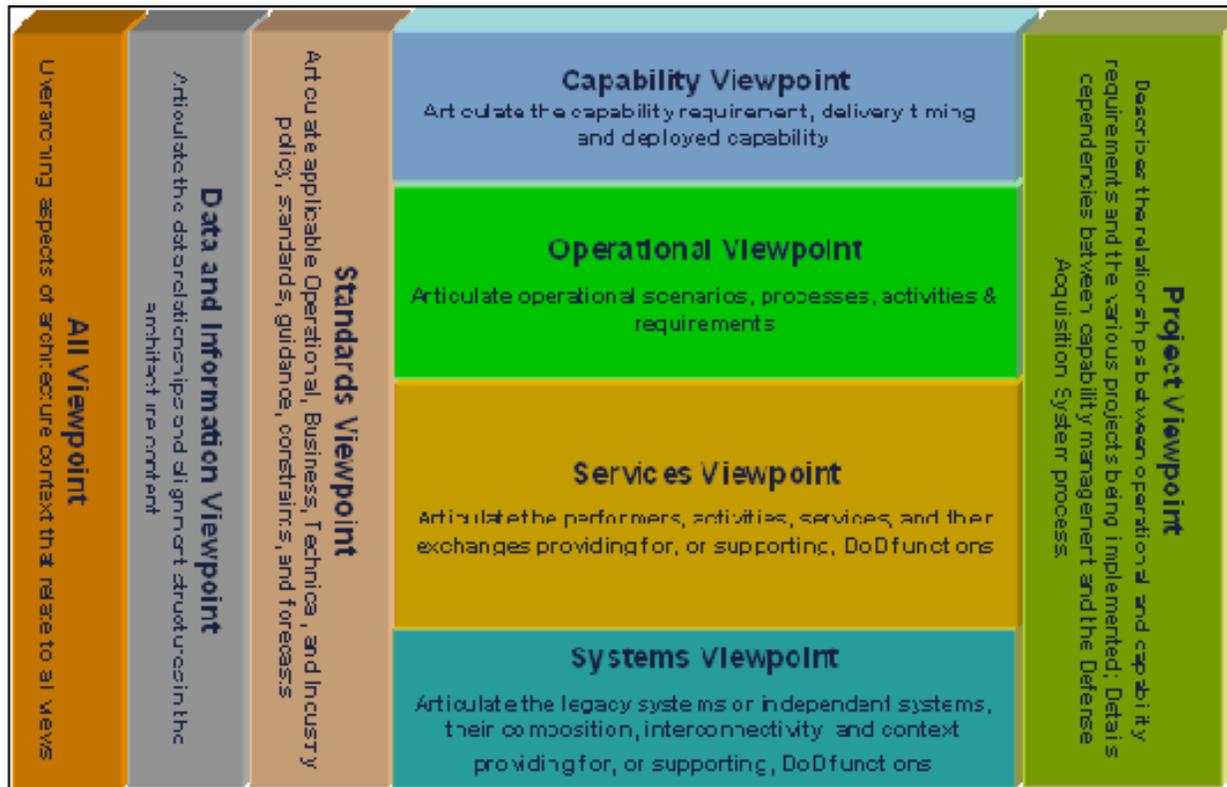
(3) TOGAF



出典： 参照アーキテクチャ調査報告, IPA, 2005.

(4) DoDAF

DoDAFのビューポイントを設定し、そこから見えるモデルを構築する



Viewpoint の一覧

- All Viewpoint
- Capability Viewpoint
- Data and Information Viewpoint
- **Operational Viewpoint**
- Project Viewpoint
- Services Viewpoint
- Standards Viewpoint
- Systems Viewpoint

参考: <http://dodcio.defense.gov/sites/dodaf20/>

ケーパビリティベースアプローチの内容 (出典: 防衛研究所紀要第11巻第3号(2009年3月))

(1) 問題点の所在

冷戦時の脅威は明確に特定されており、シナリオの数も限られていたが、冷戦後、脅威が多様化し、不透明感が増大した。そのため、これまでのように脅威を特定し、数少ないシナリオを取り上げて検討するだけでは不十分である。

(2) シナリオ検討のあり方

これまでのシナリオは、数少ないシナリオを呼称で分類している一次元のシナリオリスト (Name Level Scenario) であった。しかしながら、脅威の多様化などにより、一次元のシナリオリストだけでは十分にシナリオの内容を把握できない。そこで、考えられうる懸念 (plausible worries) をすべて一次元のシナリオリストに整理した上で、次のような観点から、多次元のシナリオ空間 (Scenario Space) に位置づける。

- 政治軍事的側面(どのように戦争が始まるのか、どの国が同盟関係にあるか等)
- 目標と戦略(目指すべき政治的・軍事的目標は何か)
- 戦力(戦力の規模、特徴、名目上の能力)
- 戦力の効率性(錬度、士気等)
- 環境条件(地形、気候等)
- その他の前提(部隊の機動力等)

(3) 分析のあり方

シナリオ空間から、必要となる任務要求 (Mission Needs) を洗い出し、その任務要求を満たすケーパビリティの選択肢についてシミュレーションを用いた探求的分析 (exploratory analysis) により分析し、評価する。

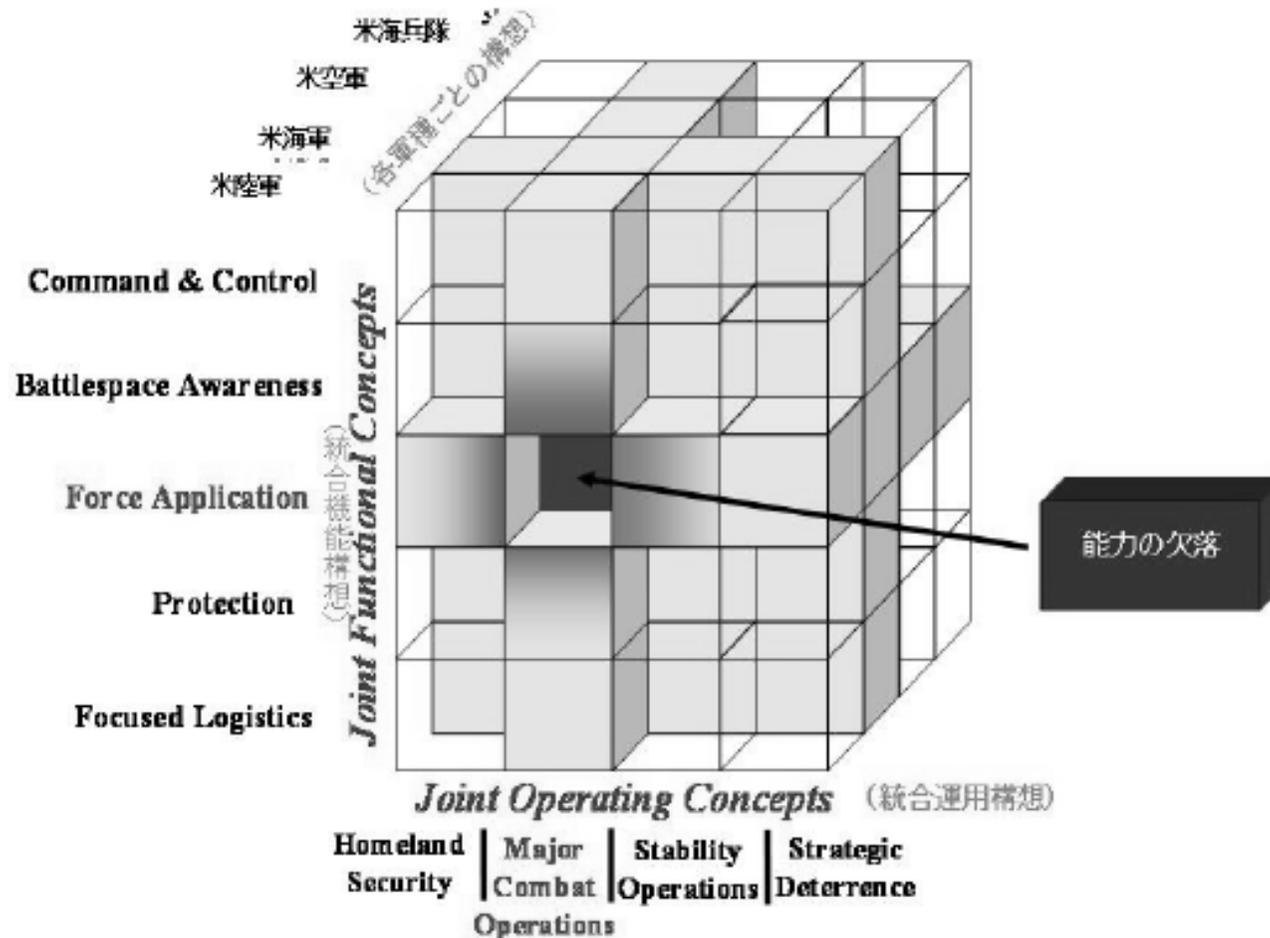
このケーパビリティベースアプローチに関する考え方は、ランド研究所のポール・デビス (Paul Davis) により整理されたものである。

Paul K. Davis, “Analytic Architecture for Capabilities-Based Planning Mission-System Analysis, and Transformation,” RAND, 2002, pp.23-25. 米国の防衛改革の構造と展望

(4) 計画のあり方

ソフトウェアの部品のように、各種軍事能力を部品化し、積木 (Building Block) 状に組み立てる。それはちょうど次ページの図のようなイメージである。そして、これら各種軍事能力の部品を事態対処に最適な形に組み合わせ、あらゆる事態に対処可能なケーパビリティを経済的に確保する。

部品とは、システムエンジニアリングの分野でオブジェクト指向分析技術を使用して作成されるプログラムの部品を想定していると思われる。



軍事能力(Capability)部品化のイメージ

出典：防衛研究所紀要第11巻第3号(2009年3月)

ICSE2010, 2011における技術課題の比較を行った。

- ソフトウェア工学分野の技術課題の洗い出しをICSE2010、ICSE2011 の資料を参考に行った。
- 運用技術と開発技術を関係づける考え方はソフトウェア工学分野としては欠如している。推進が必要。

付録C ソフトウェア工学分野の技術課題との比較

| キーワード | 技術課題 |
|----------------------|---|
| Testing | parameterized unit testing, software test automation, combinatorial test design, fault-proneness of aspect-oriented programs |
| MDE | lacks support for the modeling of architectural design rules |
| Architecture | architecture-based analysis, creating a systematic approach to designing domain specific distributed, real-time and embedded (DRE) software from software architectural design patterns |
| Software Evolution | software evolution analysis, impact analysis, change impact analysis, variability analysis, systematic change management, automated traceability, dynamic reconfiguration |
| Performance Analysis | performance prediction , performance modeling |
| グローバル | (多拠点での開発用の)assessment framework, cooperation across different countries, times zones, and cultures. |
| SPL | Symbolic model checking of software product lines |
| Human Factor | Personality factor conscientiousness on the effectiveness of Pair Programming, self-organizing teams, understanding "people" factors, factors that impact the integration experience of newcomers |
| カイゼン、見積 | process simulation, hybrid estimation(HyDEEP), maximum likelihood estimation |