

要求発展型開発 WG 2011 年度 活動報告書

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**

はじめに

IPA/SEC では、日々変化する社会環境やビジネスニーズ等に柔軟に対応するための情報システム構築技術の適用に関する指針づくりに向けた調査を実施し、要求の変化に対応する情報システムの構築に関する技術課題を収集し、ステークホルダ要求の観点などの切り口から情報システム構築技術の体系化を行い、その結果を報告書としてまとめました。

本活動は、「2011 年度 システムズエンジニアリング実践拠点事業」として、調査について株式会社三菱総合研究所に委託して実施、その結果をもとにエンタプライズ系ソフトウェア開発力強化推進委員会要求発展型開発ワーキンググループにより、実施しました。

要求発展型開発 WG 活動報告

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

【活動報告書】

独立行政法人情報処理推進機構

Copyright© Information-Technology Promotion Agency, Japan. All Rights Reserved 2012

目次

I.	背景と目的.....	1
II.	要求の変化.....	2
III.	活動概要.....	4
	1. 2010年度の取り組み.....	4
	2. 2011年度の活動概要.....	7
	3. WG 検討概要.....	10
IV.	各技術分野・知識体系の整理.....	12
	1. プロセス整理.....	13
	2. ステークホルダー整理.....	14
	3. 要求の変化とステークホルダーの関係整理.....	22
V.	技術課題の整理.....	23
	1. 技術課題抽出の留意点.....	23
	2. 技術課題について.....	24
	3. 技術課題の評価軸の検討.....	26
	4. 技術課題抽出.....	27
	5. 技術課題のまとめ.....	32
VI.	まとめ.....	33
	1. 評価軸ごとの寄与度.....	33
	2. 技術課題の総合評価.....	38
VII.	今後の課題など.....	40
	参考文献.....	41

付録： 委託調査で取り上げた技術課題の一覧	42
1. ビジネス IT オペレーションにおける技術課題	43
超上流工程分析・評価(システムズエンジニアリング)	44
2. 利用・運用における技術課題	49
運用技術	50
3. 開発における技術課題	53
上流工程における横断的連携 とシステム検証技術	55
上流工程における横断的連携とシステム検証技術 - 上流工程開発力強化 -	57
上流工程における横断的連携とシステム検証技術 - ロバストネスの確保 -	58
要求管理 - 要求の明確化 -	59
要求管理 - スコープ決め -	59
要求管理 - トレーサビリティ管理/ベースライン管理 -	61
要求管理 - コントロールケースの活用 -	62
システム構成関連技術(部品化・連携) - 構造の明確 -	63
システム構成関連技術(部品化・連携) - 部品化 -	64
システム構成関連技術(部品化・連携) - クラウド活用 -	65
安全と情報セキュリティ - テスト網羅性の保証 -	66
安全と情報セキュリティ - 不測の管理・仮定の管理 -	67
安全と情報セキュリティ - 運用性・可用性・アシュアランスの評価 -	68
安全と情報セキュリティ - 共通基盤としての保証 -	69
安全と情報セキュリティ - 他者に求められる保証 -	70
非ウォーターフォール型開発技術 - 要求の明確化 -	71
非ウォーターフォール型開発技術 - 構成管理 -	73
非ウォーターフォール型開発技術 - 短期間での開発 -	75
非ウォーターフォール型開発技術 - 反復による開発 -	77
非ウォーターフォール型開発技術 - 品質管理 -	79

I. 背景と目的

現代社会において、消費者ニーズやビジネスニーズは、刻々と変化していくものであり、そのスピードも増している。したがって、社会で活用される情報システムにもその変化への迅速な追従が求められている。それらの変化は、ステークホルダーの関与、時間の経過によって、更なる要求の変化となり、対応する情報システムの構築は、複雑化を増している。

また、その変化に対応しなければ、現代社会の情報システムは成立しない。

従来のように情報システムの構築を行った場合、ビジネススピードに対応ができず、歪みが生じシステム運用部門による個別対応やイレギュラー処理が多発し、人為的ミスが生じる可能性が高まる。

情報システムの構築においては、開発プロセスの移行に伴い変化するステークホルダーの要求をはじめ、消費者ニーズ、ビジネスニーズという根本的な要求は、日々変化していくことが前提である。IPA/SEC エンタプライズ系ソフトウェア開発力強化推進委員会要求発展型開発ワーキンググループ（以下、要求発展型開発WG）では、このように捉えた上で、それらに対応する開発技術を『要求発展型開発』と定義した。

『要求発展型開発』においては、ただ単に開発段階で要求の変化に対応するというのではなく、ビジネス要求やシステム運用、ビジネス（サービス）運用で発生する要求など、情報システム構築の全プロセス、ステークホルダー、アーキテクチャなど、全要素を網羅することを目指し、いかなる環境においても対応可能なシステム構築を対象とする。

II. 要求の変化

要求の変化は様々な要因により発生する。それにより、社会、事業経営を支える情報システムに対する要求も変化し、それに対応していかなければならない。

また、関係するステークホルダーの変化、実現するまでのプロセスによる、要求の発展についても考慮が必要である。

ただし、ここでは、要求の変化や要求が発展していくことを検討範囲としており、バグ改修等の不具合の修正を対象としているものではない。

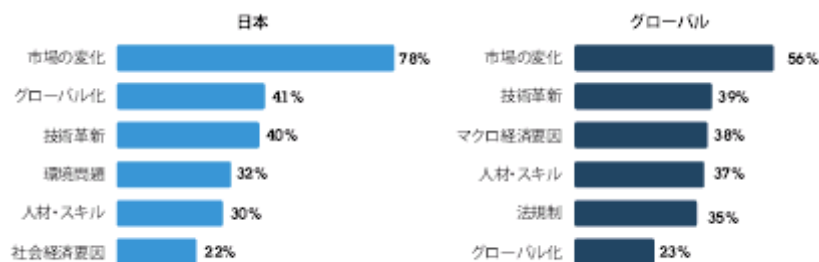
要求が変化し、要求発展を検討していく上で、必要だと思われる要素について、以下の3つを挙げる。それらは、要求変化の直接的要素でもあり、要求変化・発展に対応するための要素でもある。

① 環境の変化と持続可能な社会や経営

現代社会にとって、IT 活用はなくてはならないものとなっていることは当然のことながら、その重要性は年々高まっている。特に情報システムにおいては、企業の業務システムや制御システムのみならず、CPS (Cyber Physical Systems) の発展で組込みシステムにおいても活用される機会が増えてきている。またスマートフォン等の組込みシステムから気楽に情報システムがアクセス可能となり、国民からもますます身近なものとなってきている。

このような状況のもと、進化・発展を続ける利用環境の多様化への対応や、加速する情報の伝達スピード等の要因によるビジネス市場への対応は、日々変化し、情報システムは、それらへ追従を迫られている。

2010年に日本IBMが発表している「IBM Global CEO Study 2010 Japan Report」でも、経営者が考える自社に最も大きな影響を与える外部要因として「市場の変化」を挙げている(図II-1)。



出典: 日本 IBM: Global CEO Study 2010 Japan Report

図 II-1 今後の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(sustainability: サステナビリティ)

サステナビリティ(持続可能性)とは、もともとは地球環境、自然環境との共生をテーマとして提唱されていた言葉である。たとえば、将来枯渇が予想される限りある資源に対し、代替エネルギーの利用や再利用、再生エネルギーなどの利用により、人間社会を現在だけではなく、将来にわたって持続可能なものとするというものである。

このサステナビリティが、最近では、ビジネスシーンにおいても使われている。この場合には、市場環境や社会環境の変化などに対応し、現在のニーズを満たすとともに予測される将来を見据えた社会システムや事業を持続的に運営していくと言ったこととなる。

これらの社会システムや事業を運営していく上で、情報システムにおいても、持続可能性に配慮した構築が必要となる。様々な環境の変化に対応するために、要求の変化に対応していく情報システムの構築方法は、一般的なシステム構築とは開発方針が異なることになる可能性がある。例えばアジリティの高いシステムの構築に際してはすでに、重要視する開発技術に明白な差異があると考えられる。環境の変化、すなわち要求の変化が激しい時代の情報システム構築技術の在り方を世に示していく必要がある。

③ 要求の逸脱(システムの複雑化)

情報システムの利用用途の拡大、利用環境の多様化など様々な要因により、システムは複雑化を増し、システム間の連携も様々である。また、そのシステムの連携は、ニーズの変化により、日々変化している。そのため、障害発生時には、原因究明に困難を極め、思わぬところに落とし穴が存在している場合がある。

障害を発生させないことを前提としたシステム構築を行った場合においても、前述のとおり、ニーズは日々変化しており、それに伴うシステム対応を求められ、システムは巨大化と共に複雑さを増しているため、絶対に障害が発生しない確証のある構築は、極めて困難ということとなる。

そのため、障害の発生に対応するためにアシュアランスを確保するための技術が必要となる。

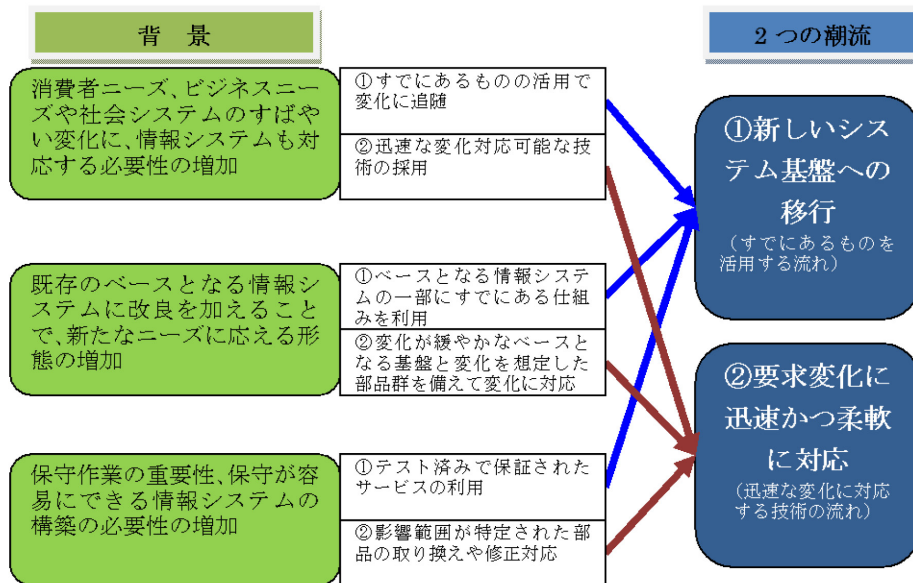
III. 活動概要

1. 2010 年度の取り組み

保守容易性(機能拡張性、性能拡張性など広義の意味)の高い既存の情報システムがベースとなる必要があるとして、IPA/SEC では、2010 年度「保守性の高い情報システムの構築技術に関する調査」(以降、「2010 年度調査」という。)を実施した。

2010 年度調査においては、2つの注目すべき保守容易性に関連したシステム構築の潮流が明らかになった。1 つは、クラウドコンピューティング環境など新しいシステム基盤への移行による情報システムの構築であり、極力「作らない」ことで提供される新しいサービスに移行することに重点がある。もう 1 つは、要求の変化に迅速かつ柔軟に対応可能な情報システムの構築であり、新しいサービスを状況の変化に即して「素早く開発・移行できる」ことに重点がある。

図 III-1 保守容易性に関連した 2 つの潮流



① 新しいシステム基盤への移行

利用者側の立場で、すでにあるものを活用する流れとみることができる。この流れには、開発や保守をアウトソーシングする外部人材の活用がある。保守容易性を高めるために、その外部人材等のリソースが構築してテストを行い、動作確認した外部ソフトウェア製品の活用につながる。この外部ソフトウェア製品を活用する流れから、社内にシステム基盤も持たずに外部基盤の活用によりサービスの提供を受ける形態へとつながる。SaaS などクラウドコンピューティングに代表される外部基盤の活用が新しいシステム基盤への移行である。

② 要求変化に迅速かつ柔軟に対応

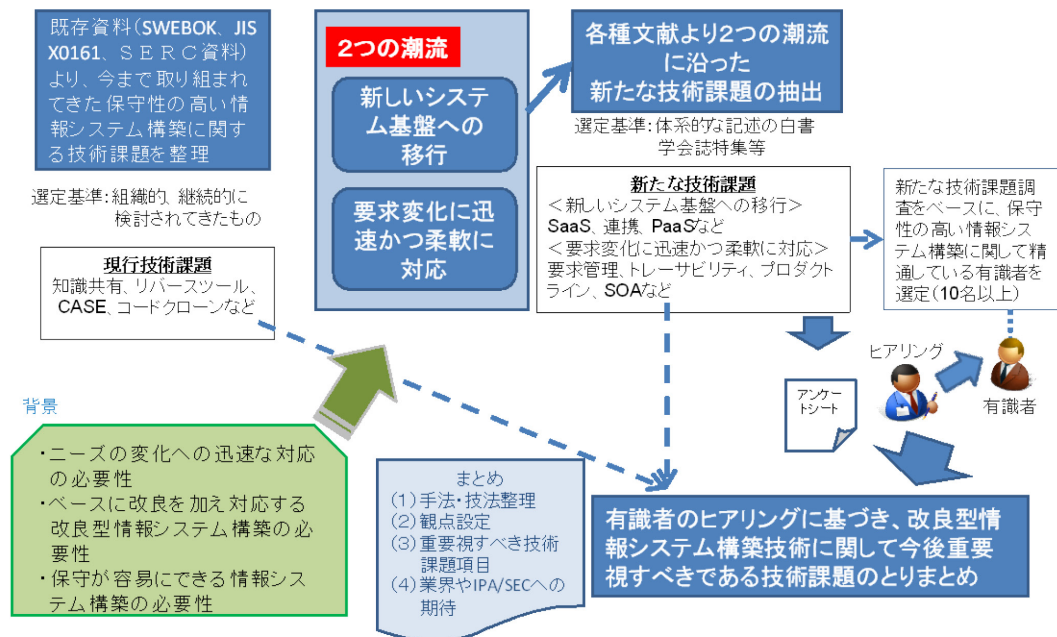
ソフトウェア部品の活用を行うなど情報システム構築期間を短縮することでその変化に追従する方法である。社内の独自の経営戦略を素早く実現し、競争優位を確立するためにも重要である。

この変化に対応するためには、要求把握の充実、迅速かつ柔軟なシステム開発が重要であり、一貫した工程があることで、実現可能性が高まる。

これらの2つの潮流で活用される技術課題項目、特に上流工程での技術課題項目を中心に、保守性の高い情報システム構築技術に関する技術課題項目の抽出を行い、さらに今後深掘すべき技術課題の洗い出しを行うため、以下の3点について、調査・検討等を行った。

- ・ 既存システムをベースとした改良型情報システム構築のためになる重要となる技術課題項目の検討・調査
- ・ 情報システムの一部機能の新しいシステム基盤への移行や要求の変化に迅速かつ柔軟に対応可能な情報システムの実現のために重要となる技術課題項目の検討・調査
- ・ 技術課題項目全体の整理と特徴付け

図 III-2 2010 年度調査の実施概要



これら2つの潮流について、①外部の基盤の活用、②要求把握の充実、③迅速かつ柔軟なシステム開発、④一貫した工程の4つの観点のもと、保守性の高い情報システム構築技術について、有識者にヒアリング調査を実施した。

図 III-3 今後重視すべき技術課題(有識者ヒアリング調査結果)

有識者の専門領域	観点	(要求定義フェーズ) 要求管理	(設計・構築フェーズ) 部品化・連携	(テストフェーズ) ソフトウェアアシュアランス	(その他)
1	ソフトウェアエンジニアリング全般	要求の明確化 要求の可視化	「構造の明確化と定義に基づく部品化」 「分解と再結合」、 「System to System Integration」	ソフトウェアアシュアランス	
2	ビジネスシステム開発支援	トレーサビリティ管理			
		要求管理	SOA	回帰テスト、テスト自動化	
3	ビジネスシステム開発支援	プロダクトライン			
		スコープ決め			
			モデルとビューの分離、オブジェクト指向、 差分開発、アプリケーションフレームワーク		
4	ビジネスシステム保守開発	「コーディングレベルでの仕様記述」 「業務担当と仕様記述者の連携」		「業務担当による受入テスト」「実データケースによるテスト網羅性」	
5	クラウドコンピューティング技術	「要求仕様は顧客と一緒に作る」	SOA、SaaS、PaaS、連携	セキュリティ認証	ガバナンス、アジャイル開発
6	クラウドコンピューティング基盤構築	業界特化	連携、相互運用性、SaaS、マッシュアップ、 データ移行、SOA、Webサービス	品質保証、検証・監査、運用管理 セキュリティ、仮想化、マルチテナント	
7	インフラ基盤開発	スコープ決め、プラットフォームアプローチ			
		マネジメント技術、サービス提供戦略	オブジェクト指向、コンポーネント、差分開発		
8	組込システム開発	プロダクトライン			
		ドメイン分析、概念モデリング、変更管理、派生開発			
9	組込システム開発	形式仕様記述言語			
		変更管理			
			再利用、フレームワーク、部品化、部品探索	テスト網羅性証明	
		影響範囲の特定		影響範囲の特定	
10	保守技術	不測事態への対応	再利用	テスト網羅性証明、回帰テスト、テスト自動化	要員スキル、デザインレビュー、アジャイル開発
11	保守技術	仮定の管理		クラウドコンピューティングの機密性	「ソフトウェアの進化プロセス」システム維持可能性

これらのヒアリングの結果、【要求管理】、【部品化・連携】、【ソフトウェアのアシュアランス】の3つの観点に多くの技術課題が関連付けられることが分かった。

この3つの観点に対し、「技術課題概要」、「課題解決効果」、「課題解決に向けた留意事項・課題」についてまとめを行った。

① 要求管理

- a) 要求の明確化 b) スコープ決め c) トレーサビリティ管理

② 部品化・連携

- a) 構造の明確化 b) 部品化 c) 連携

③ ソフトウェアのアシュアランス・品質保証

- a) テスト網羅性の保証 b) 共通基盤としての保証 c) 他者に求める保証

2. 2011 年度の活動概要

本年度の目的

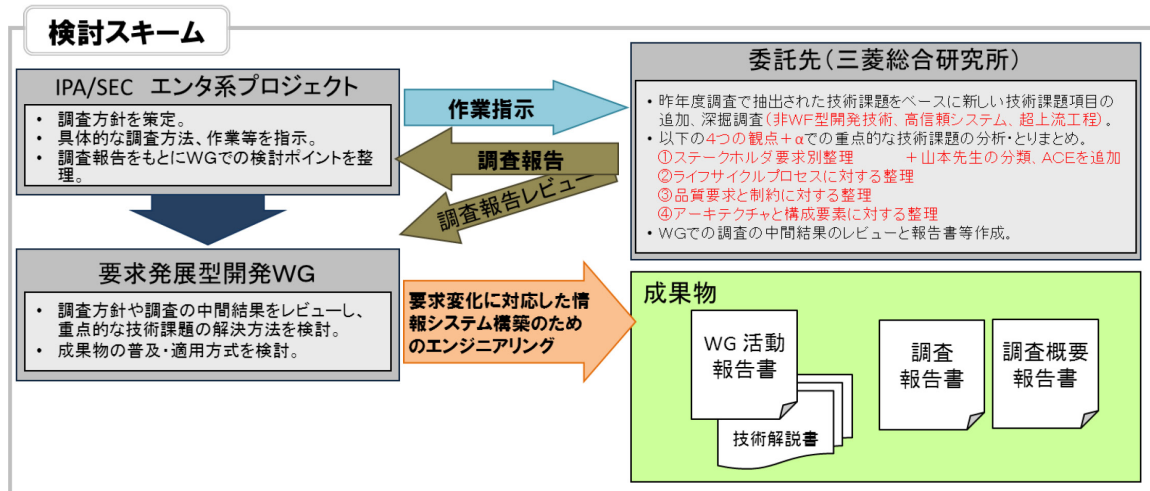
IPA/SEC では、前述のように 2010 年度調査を実施し、クラウドコンピューティング環境など新しいシステム基盤への移行や要求の変化に迅速かつ柔軟に対応可能な情報システム構築を中心に必要となる技術課題項目の抽出を行い、今後深掘すべき技術課題の洗い出しを行った。

本年度(2011 年度)は、2010 年度調査で抽出された技術課題をベースとしてさらなる技術課題項目の追加、深掘調査を行い、ステークホルダー要求やライフサイクルプロセス、情報システムの品質要求と制約、アーキテクチャと構成要素に応じた情報システム構築技術の体系化を行った。

これにより、我が国の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した。

① ワーキンググループ検討と委託調査連携

調査・検討にあたっては、調査実施の委託先である株式会社三菱総合研究所に委託した調査とともに、IPA/SEC エンタプライズ系ソフトウェア開発力強化推進委員会内に設置した、「要求発展型開発 WG(ワーキンググループ)」での検討を進め、双方連携の上、実施した。



② WG活動

2011年11月より、WG会合を開催し、委託調査内容のレビューと共に検討を行った。

- 第1回開催： 2011年11月 8日
- 第2回開催： 2011年12月 15日
- 第3回開催： 2012年 1月 12日
- 第4回開催： 2012年 2月 14日
- 第5回開催： 2012年 3月 6日
- 第6回開催： 2012年 3月 13日

【要求発展型開発WG委員リスト】

敬称略

	氏名	所属
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 (NPO 法人 IT サービスマネジメントフォーラムジャパン)
委員	山本 久好	日本アイ・ビー・エム株式会社
委員	鷲崎 弘宜	学校法人早稲田大学

③ 2011年度 委託調査の要件

件名 : 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

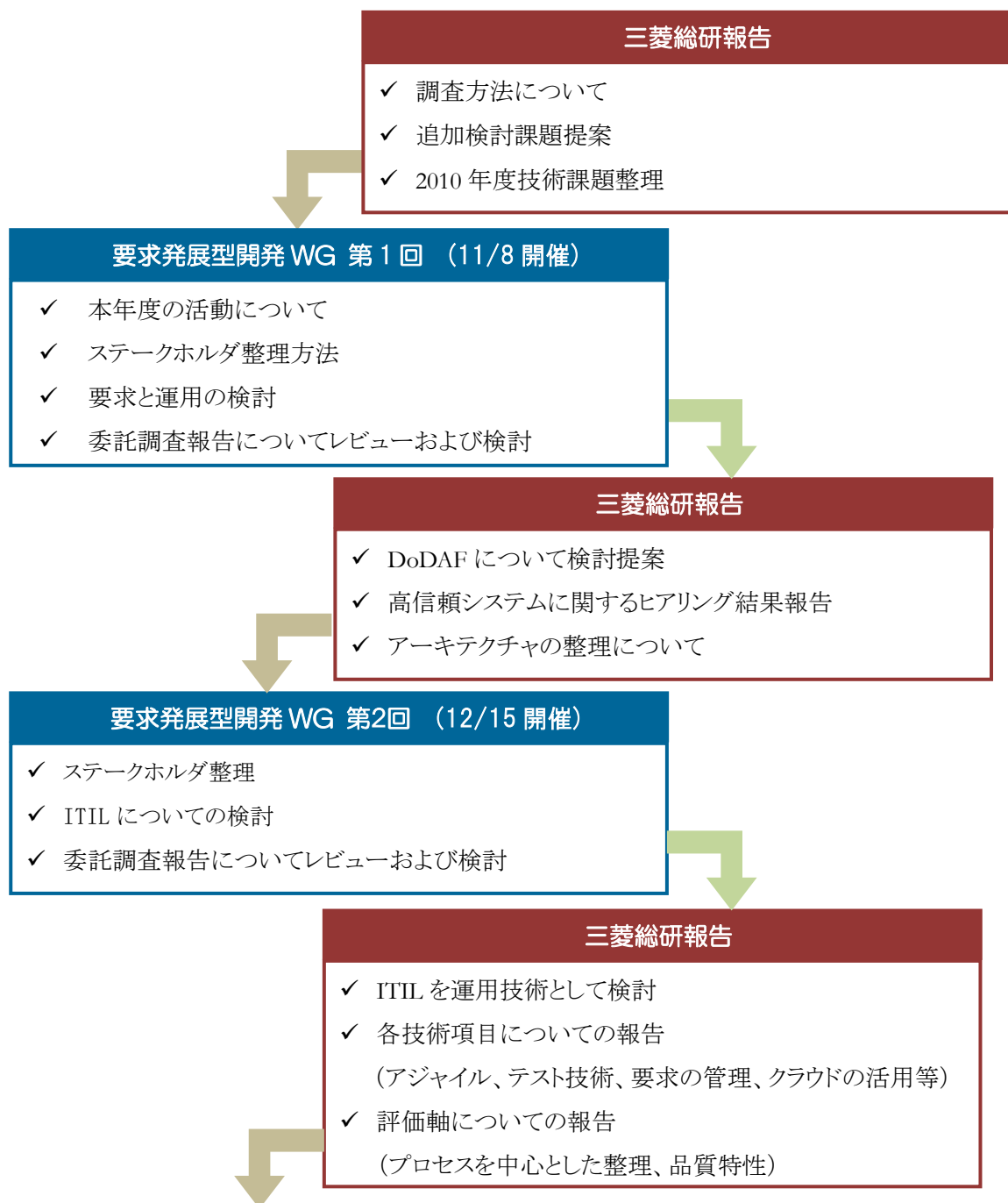
委託内容:

- (1) 新しい技術課題の追加調査
 - <調査項目>
 - ・ 技術課題概要
 - ・ 課題解決効果
 - ・ 課題解決に向けた留意点・課題

- (2) 技術課題項目の整理と深掘調査
 - ① ステークホルダー要求別整理
 - ② ライフサイクルプロセスに対する整理
 - ③ 品質要求と制約に対する整理
 - ④ アーキテクチャと構成要素に対する整理

3. WG検討概要

要求発展型開発 WG 会合の各回の主な議題および、三菱総合研究所の調査報告については、下記のとおりである。



要求発展型開発 WG 第 3 回 (1/12 開催)

- ✓ ステークホルダ定義の整理について
- ✓ プロセス整理について
- ✓ 要求例について
- ✓ 委託調査報告のレビューと検討

三菱総研報告

- ✓ 評価軸のまとめと技術マップ作成について
- ✓ 運用の定義について

要求発展型開発 WG 第 4 回 (2/14 開催)

- ✓ 委託調査報告のレビューと検討

要求発展型開発 WG 第 5 回 (3/6 開催)

- ✓ 要求発展型開発 WG 活動報告まとめ
- ✓ 検討課題整理

要求発展型開発 WG 第 6 回 (3/13 開催)

- ✓ 要求発展型開発 WG 活動報告まとめ

IV. 各技術分野・知識体系の整理

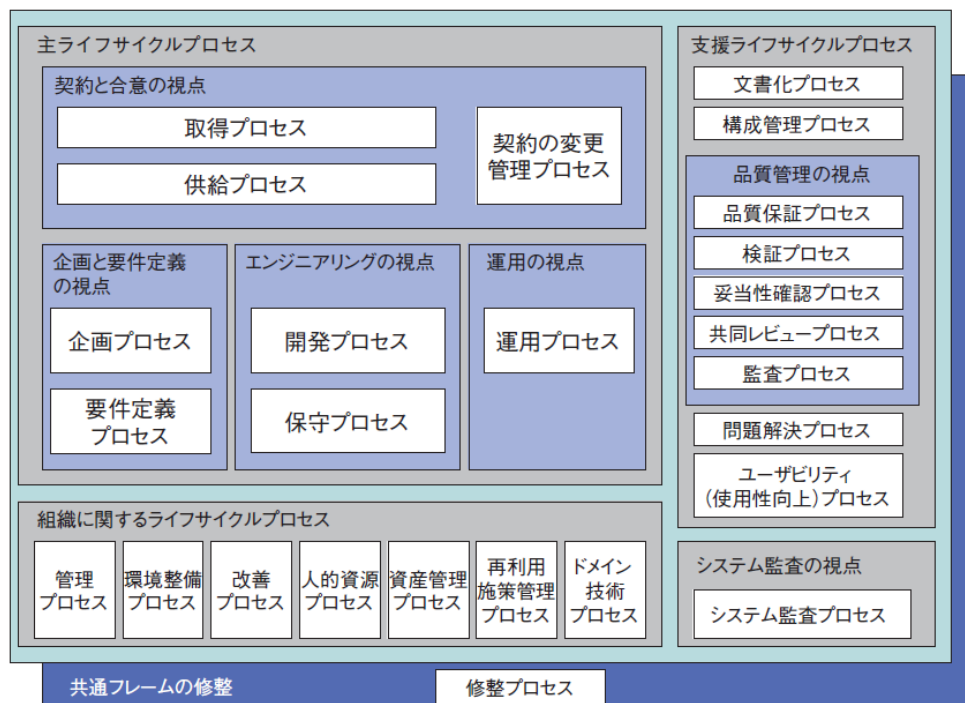
各技術分野／知識体系を整理するにあたり、技術分野／知識体系ごとに技術等の表現方法や守備範囲の違いにより、そのまま読み解くには困難であった。そのため、プロセスやステークホルダーなどの指し示す用語について、技術分野／知識体系間で、ある程度の共通化を図りつつ、それぞれの特性を読み解くとともに整理を行った。

対象とした技術分野／知識体系は以下のとおり。

- ① 共通フレーム 2007 第 2 版 「利害関係者の役割」
- ② ビジネスアナリシス知識体系ガイド(BABOK®ガイド) Version2.0
- ③ 要求工学知識体系(REBOK) 第 1 版
- ④ プロジェクトマネジメント知識体系ガイド(PMBOK®ガイド) 第 4 版
- ⑤ TOGAF Version 9
- ⑥ ITIL v3

また、要求についても、技術とのマッピングをするためには、体系化が必要であるという認識から、要求の目的として整理を行った。(要求の変化とステークホルダーの関係整理)

図 IV-1 共通フレームの基本構成



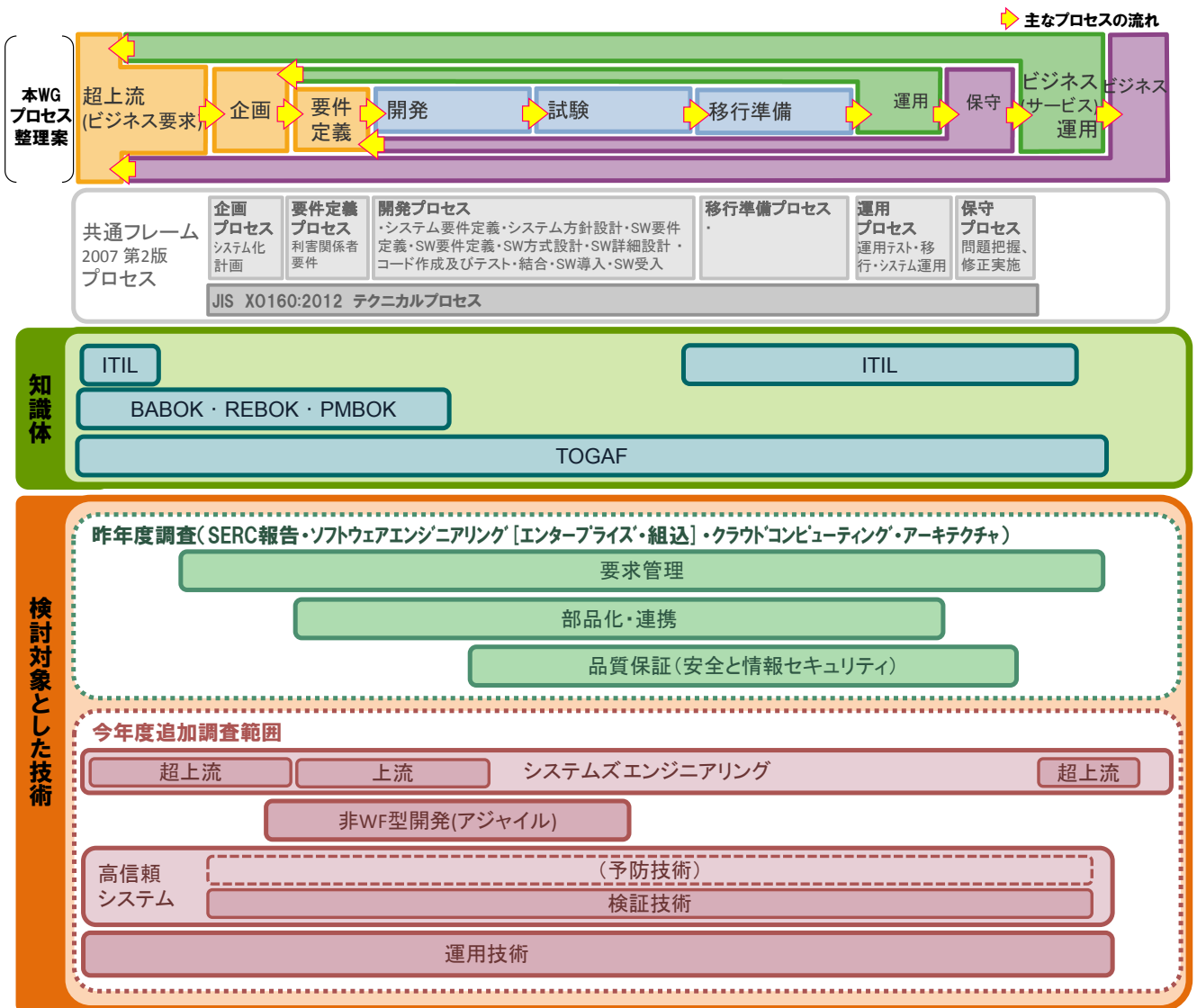
出典: 共通フレーム 2007 第 2 版

1. プロセス整理

要求発展型開発 WG では、要求の変化に対応する開発を検討するにあたり、まず、共通フレーム 2007 第 2 版のプロセスを中心に、ビジネス要求から運用までのビジネスサイクルのプロセスを改めて見直しを行った。その結果、当然のことながら、ビジネス運営を行うビジネス(サービス)運用/保守と超上流工程は、密接なつながりがあり、全てのプロセスが循環(ループ)していることが明らかとなった。

これらの基本的なプロセスの循環に対応する上で、2010 年度調査で洗い出された技術課題とともに必要とされる技術・知識体系の対応範囲を本 WG 内にて図 IV-2 のように整理し、委託調査の調査範囲、WG での検討範囲として、検討・調査を進めた。

図 IV-2 プロセス・技術/知識体系の整理



2. ステークホルダー整理

要求元となるステークホルダーについての検討にあたり、各技術分野／知識体系により、ステークホルダーの定義が異なり、同一線上での検討が困難であった。そのため、本 WG において、図 IV-3 に示すようにステークホルダーの大分類を定義し、それぞれの技術分野／知識体系で定義されているステークホルダーをそれにしたがって、それぞれの範囲やビューポイントを考慮し、整理した上での検討を進めることとした。

本 WG でステークホルダー大分類については、共通フレーム 2007 第 2 版の部署等／役割(ロール)の経営層、業務部門、情報システム部門、ベンダに不足要素を加えた、下記の 7 分類である。それぞれの技術／知識体系でのステークホルダーについては、次ページ以降を参照のこと。

要求発展型開発WGでのステークホルダー大分類(7分類)

- 外部 : 顧客、規制機関など
- 経営層 : 社長をはじめとする社内の経営層、株主など
- 管理部門 : 人事などの社内管理部門、労働組合など
- 業務部門 : 社内でのシステム利用者、サービス主管部門など
- 情報システム部門 : 情報システム部門、システム開発担当など
- システム運用部門 : システム運用・保守部門、ヘルプデスクなど
- ベンダ : ベンダ、システム開発パートナーなど

図 IV-3 ステークホルダー整理

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	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	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 プロセス生成物専門家 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 上級システムエンジニア 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				

① 共通フレーム 2007 第 2 版

共通フレームの場合には、「部署等／役割(ロール)」として、ステークホルダーを整理している。

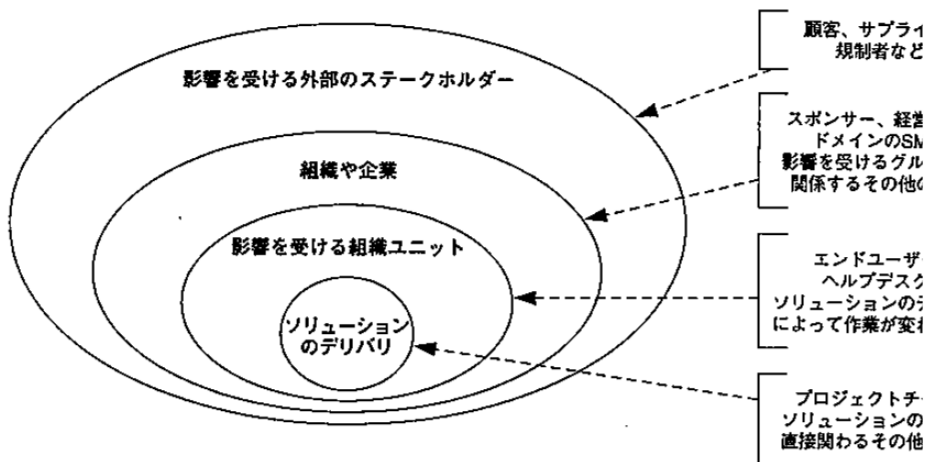
部署等／役割(ロール)		要件の定義内容		
経営層	社長	事業要件 定義	業務要件 定義	
	担当役員			
業務部門	部門長			
	業務推進担当			
	システム推進担当			
	関連会社			
情報システム 部門	部門長			システム 要件定義
	システム開発担当			
	システム子会社			
ベンダ	元請けベンダ			
	アウトソーサ			
	サブベンダ			

出典: 共通フレーム 2007 第 2 版 / IPA/SEC

ステークホルダー	役割等
経営層	社長、担当役員
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
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
ベンダ	元請けベンダ、アウトソーサ、サブベンダ

出典: 共通フレーム2007 第2版を基に作成

② ビジネスアナリシス知識体系 (BABOK) ガイド



出典:ビジネスアナリシス知識体系ガイド(BABOK®ガイド) Version2.0/IIBA®日本支部

ステークホルダー	役割等
ドメインの SME	部署や職務などとは分けて考える
エンドユーザー	部署や職務などとは分けて考える
実務の SME	プロジェクトのライブラリアン、変更管理担当者、構成管理担当者、ソリューションアーキテクト、開発者、データベースアーキテクト、情報アーキテクト、ユーザビリティアナリスト、トレーナー、組織変革コンサルタントなど
運用サポート	ヘルプデスク、ネットワーク技術者、リリースマネジャー
プロジェクトマネジャー	スクラムマスター、チームリーダー
サプライヤ	プロバイダ、コンサルタントなど
テスト担当者	品質保証アナリスト
規制者	行政府、規制担当機関、監査役
スポンサー	マネジャー、経営幹部、プロダクトマネジャー、プロセスオーナー

出典:ビジネスアナリシス知識体系ガイド(BABOK®ガイド) Version2.0 を基に作成

③ 要求工学知識体系 (REBOK)

REBOK では、ステークホルダーは、要求の関与者としており、要求アナリスト、要求工学の関与者等については、アクタとしてステークホルダーとは分けて定義している。



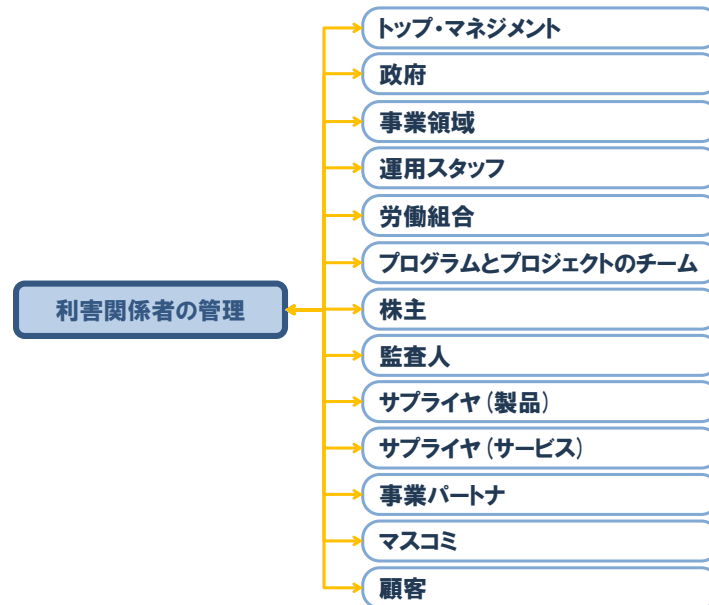
出典: 要求工学知識体系(REBOK) / 一般社団法人情報サービス産業協会(JISA)

ステークホルダー	役割等
ユーザ	経営者 (CIO)、エンドユーザ、プロジェクトマネジャー、システム責任者
ベンダ	ベンダ側経営者、プロジェクトマネジャー、上級システムエンジニア、ソフトウェア開発者

出典: 要求工学知識体系(REBOK)を基に作成

④ ITIL v3

図 IV-4 ITIL v3 考えられる利害関係者



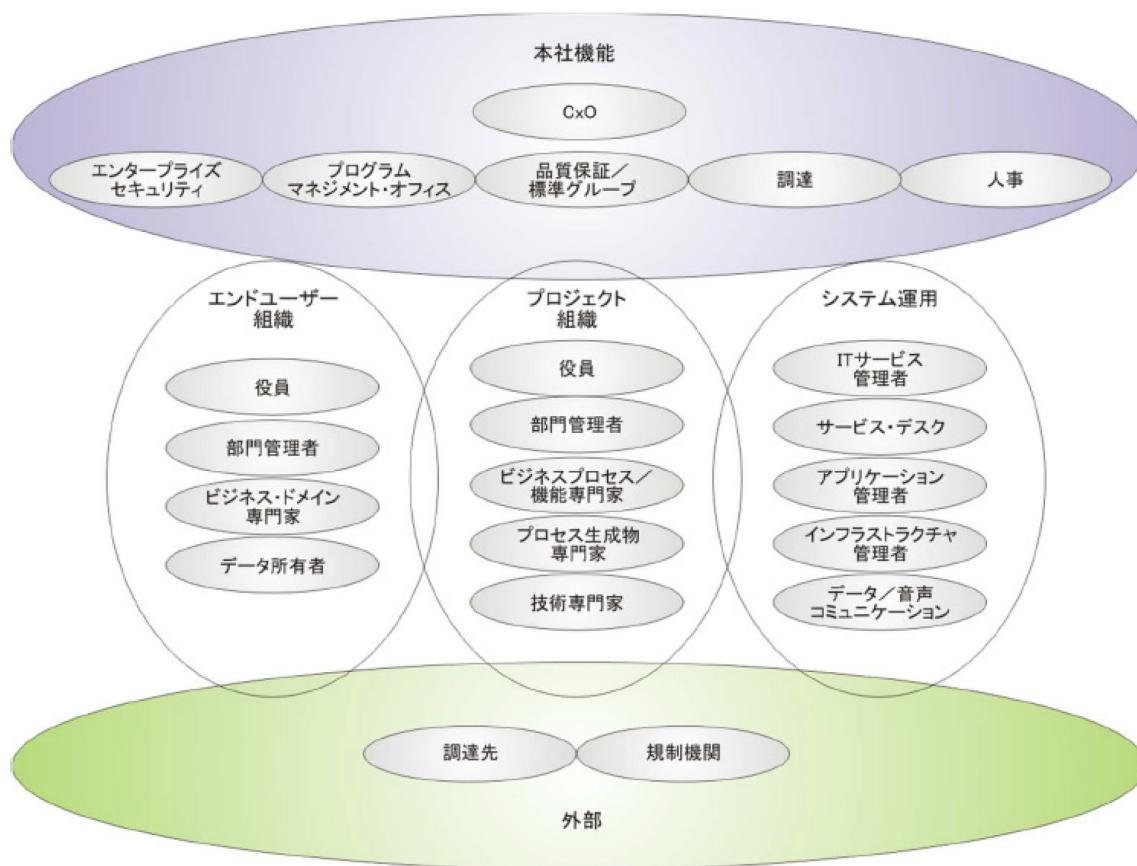
出典:ITIL V3 書籍 サービスランジション(日本語版)/itSMF Japan を基に作成

表 IV-1 影響を受ける可能性のある人々の例

<input type="checkbox"/> サービス変更(技術の一新など)の後援者
<input type="checkbox"/> サービス変更またはサービスランジションによって影響を受ける人々
<input type="checkbox"/> サービスのセットやサービス・パッケージに含まれる商品やサービスのサプライヤ
<input type="checkbox"/> 新規または変更されたサービスに関与するサービスマネジメント・チーム
<input type="checkbox"/> サービスランジションや、新規または変更されたサービスの影響を受ける顧客や消費者
<input type="checkbox"/> 関係管理スタッフ
<input type="checkbox"/> 内部監査や外部監査
<input type="checkbox"/> 情報セキュリティ
<input type="checkbox"/> 不正対策部門
<input type="checkbox"/> リスク監理
<input type="checkbox"/> 組織の株主、マネジメント、スタッフ
<input type="checkbox"/> 労働者グループ/労働組合
<input type="checkbox"/> 政治団体や規制機関
<input type="checkbox"/> 一般市民などの、より広範なコミュニティ
<input type="checkbox"/> サービス・ライフサイクル全体の中でプロジェクトを提供している、プロジェクト管理チームおよびプログラム管理チーム

出典:ITIL V3 書籍 サービスランジション(日本語版)/itSMF Japan を基に作成

⑤ TOGAF Version 9

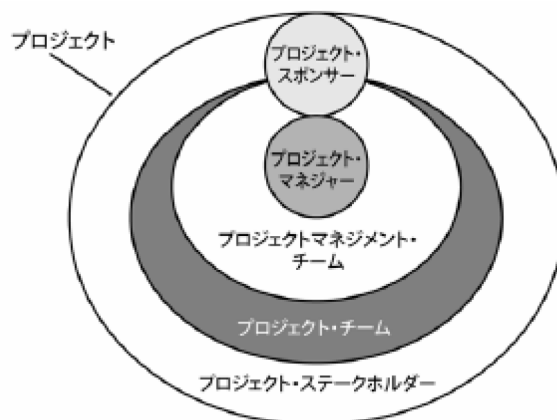


出典: TOGAF® Version 9 日本語訳版 / The Open Group Japan

ステークホルダー	役割等
本社組織	CxO、エンタープライズセキュリティ、プログラムマネジメントオフィス、品質保証/基準グループ、調達、人事
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
プロジェクト組織	役員、部門管理者、ビジネスプロセス/機能専門家、プロセス生成物専門家、技術専門家
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
外部	調達先、規制機関

出典: TOGAF® Version 9 日本語訳版を基に作成

⑥ プロジェクトマネジメント知識体系ガイド(PMBOK)

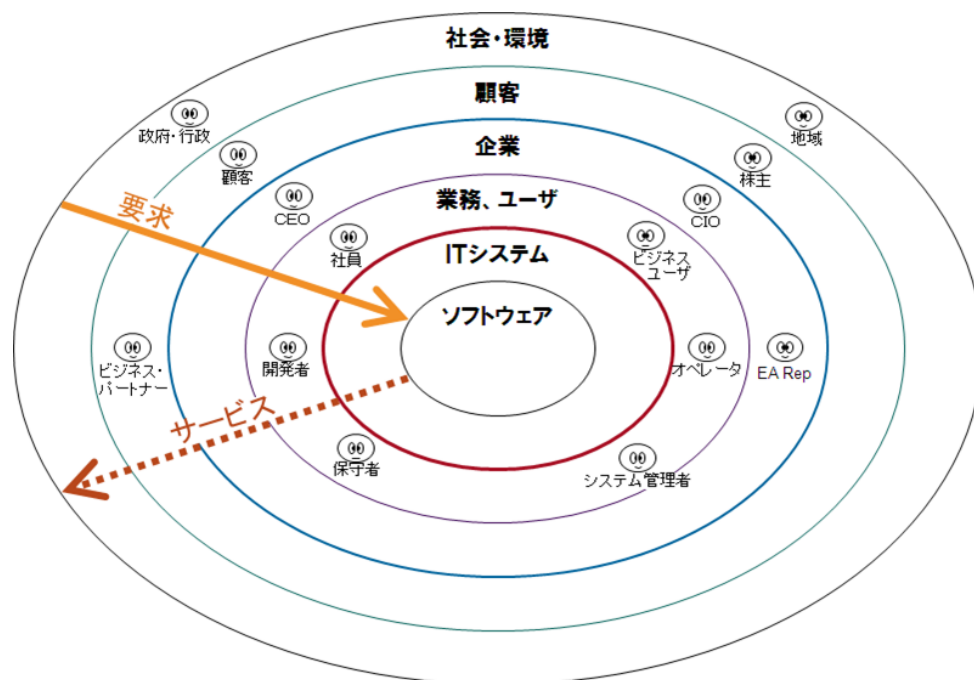


出典:プロジェクトマネジメント知識体系ガイド(PMBOK®ガイド)第4版/PMI 日本支部

ステークホルダー	役割等
プロジェクトマネジャー	プロジェクトの責任者
顧客 ユーザー	プロジェクトの成果物(製品やサービス)を使用する個人または組織
母体組織	プロジェクトの制作に関わるチームが所属する組織
プロジェクト チームメンバー	プロジェクトの制作に関わるグループ
プロジェクト マネジメントチーム	プロジェクトの制作に直接的に関わるチーム(メンバー)
プロジェクト スポンサー	プロジェクトに対して資金等資源を提供する個人または組織
影響力者	顧客組織または母体組織の立場により影響を及ぼす可能性のある個人またはグループ
PMO	母体組織内に設立される部署。 直接的または間接的にプロジェクトマネジメント・チームに携わる。
プロジェクト ステークホルダー	ポートフォリオマネジャー、プログラムマネジャー、プロジェクトマネジメントオフィス、運用マネジメント、機能マネジメント、ビジネスパートナー、顧客・ユーザ

出典:プロジェクトマネジメント知識体系ガイド(PMBOK®ガイド)第4版を基に作成

⑦ KIKUMA RING (参考)



出典：IT アーキテクチャサミット 2008，日本 IBM 榎原氏資料を一部改編

3. 要求の変化とステークホルダーの関係整理

基本的な要求の目的を整理し、それに関連するステークホルダーを関連付けることにより、IT に対する要求変化の基本・起点が定義、整理されると考え、本 WG では、いくつかの要求例に対し要求目的別に代表的なステークホルダーの関係を整理した。(図 IV-5 要求目的とステークホルダーの関係)

要求目的の整理にあたっては、IT に対する経営的課題、保守起点での要求、社内業務効率化の3点を課題と絞り、社団法人 日本情報システム・ユーザー協会(JUAS)で発表されている、以下のデータを使用し基本的な要求として検討した。

- IT に対する経営的課題 : 2011年3月発表の「企業 IT 動向調査 2011」内の IT 投資で解決したい中期的な経営課題
- 保守起点での要求 : 「ソフトウェアメトリクス調査 2011」保守作業の発生目的
- 社内業務効率化 : 「ソフトウェアメトリクス調査 2009」ITIL 適用採用目的

上記の要求を基本に、要求目的を11分類し、前述したステークホルダー7分類のうち、ベンダを除いた6分類に対し、関連づけを行った。整理した要求目的11分類は、以下のとおり。

- ・ 企業の社会的責任(CSR向上)
- ・ 法令順守・業界標準への準拠
- ・ 経営の透明性確保
- ・ 経営スピード向上
- ・ 市場シェア拡大
- ・ 顧客の確保、維持
- ・ 業務効率向上(省力化、業務コスト削減)
- ・ 業務スピード向上(リードタイム短縮等)
- ・ 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)

図 IV-5 要求目的とステークホルダーの関係

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	③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.違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・ミドル変更への対応 OS変更への対応						○	○

V. 技術課題の整理

有識者ヒアリングを中心に 2010 年度調査報告書および 2011 年度の委託調査において、技術課題の洗い出しを実施した。各技術課題は技術項目から抽出した。本章では、その概要を説明する。

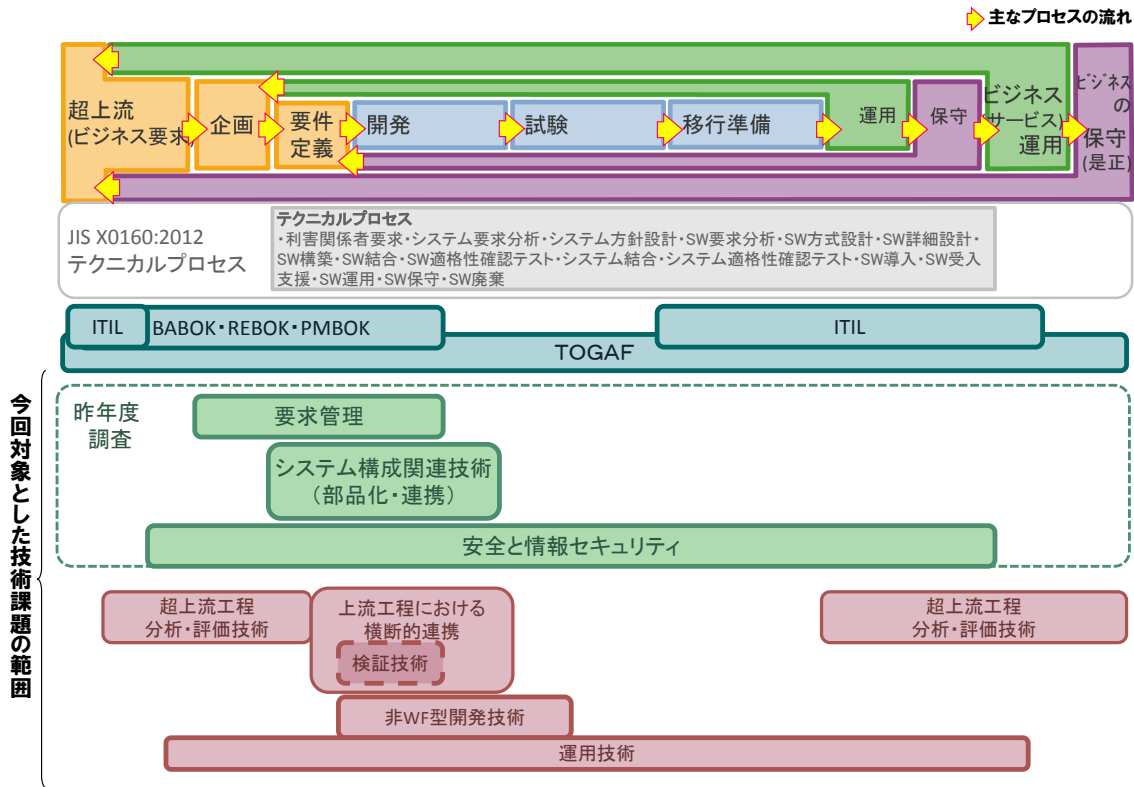
1. 技術課題抽出の留意点

有識者ヒアリング、文献・Web 調査および WG の議論を通して、技術課題をとりまとめたが、留意すべき点がある。この技術課題は網羅的にまとめている点ではないことである。例えば要求工学分野であれば、要求管理技術と並んで要求獲得技術が挙げられるが、本技術課題には要求獲得技術が取り上げられていない。これは要求獲得が「要求の変化」に対応するために重要ではないという意味ではないことに注意されたい。「要求の変化」に対応する技術を強調するために、「要求の変化」への対応に関係なく重要である要求獲得技術を敢えて取り上げていない。また検証技術についても非常に重要な位置にあるものの、「要求の変化」に特化したものを挙げづらいため、本調査で取り上げることが難しく、含んでいない。一方で検証技術が他工程に対して効果があるものや「要求の変化」に対する強さを検証する技法については、上流工程の技術として位置づけている。同様に他の技術が重要ではないということではなく、通常開発と比較してより重要な技術と考えたものを取り上げている。

2. 技術課題について

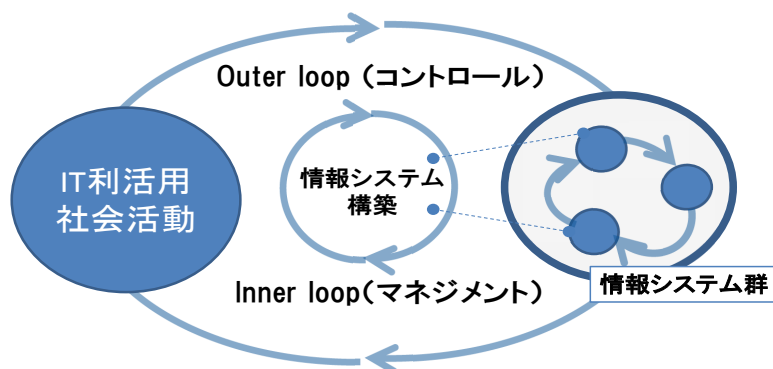
本章では、技術課題の概要を示す。なお、これらの技術のテクニカルプロセスを基準とした位置を図 V-1 示す。

図 V-1 プロセス範囲と取り上げた技術課題



要求の変化を前提とした情報システム構築は、トップダウン指向で構築する技術を必要としている。特に各情報システム構築における対応を Inner loop と呼ぶことがあり、その場合要求の変化の原因である環境の変化を受け止めるべく本報告書で活用しているビジネス-IT オペレーションである超上流工程分析評価技術の改善サイクルは、Outer loop と呼ばれる。

図 V-2 Outer Loop と Inner loop (システムズエンジニアリングの例)



出典: Hitchens, Derek, K.: Systems Engineering - A 21st Century Systems Methodology, John Wiley&Sons, 2007.

山本修一郎教授 (WG 主査) ホームページ資料を基に作成

今回取り上げた技術課題は、全体として図 V-3、4 のとおり全体をコントロール・モニタリングするビジネス-IT オペレーション、および利用・運用、さらに開発の 3 種類に分けることができる。情報システムの継続性確保のため、情報システム群全体について要求の変化をコントロールすることが特徴である。本報告書ではこれを「ビジネス-IT オペレーション」と呼ぶ。

以下、技術課題について説明する。

図 V-3 要求の変化に対応する技術の全体像(1)

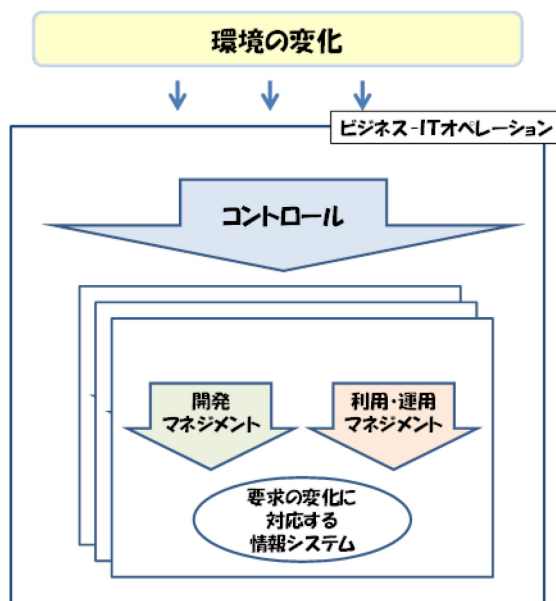
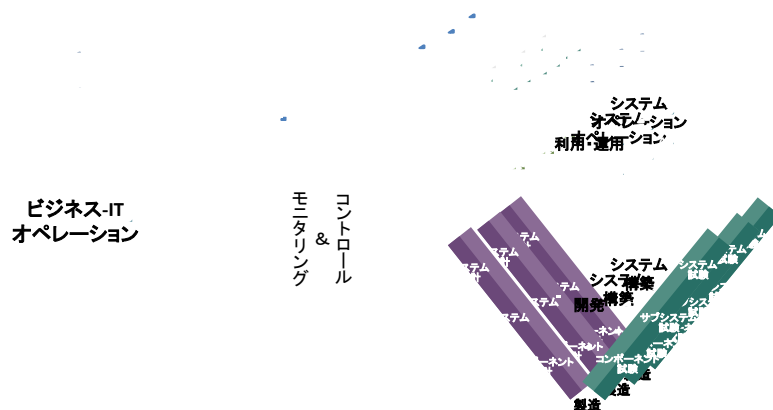


図 V-4 要求の変化に対応する技術の全体像(2)



3. 技術課題の評価軸の検討

技術課題の体系的な整理を目的に次の3つのカテゴリで評価軸を設定した。

- ① プロセスを中心とした評価
- ② ステークホルダー要求を中心とした評価
- ③ 品質特性を中心とした評価

① プロセスを中心とした評価

開発工程、超上流工程、運用工程、アーキテクチャフレームワークに分け、それぞれの工程内でのプロセスを整理し、評価軸として、各技術課題をプロット、整理を行う。

工程	整理プロセス
開発工程	開発プロセス標準ISO/IEC 12207とISO/IEC 15288プロセスを活用し、整理を行う。
超上流工程	要求の変化・発展を限定し、Enterprise Systems Engineering のプロセスを活用し、整理を行う。
運用工程	ITILを活用することとなった。各段階をプロセスと見立てて整理を行う。
アーキテクチャフレームワーク	TOGAF-ADM(アーキテクチャ開発プロセス)の各フェーズをプロセスと見立てて、整理を行う。

② ステークホルダー要求を中心とした評価

整理した11の要求目的とステークホルダーの関係をステークホルダー要求として、各技術課題をプロット、整理を行う。

③ 品質特性を中心とした評価

ISO25010 システム/ソフトウェア品質特性に対して、各技術課題をプロット、整理を行う。

4. 技術課題抽出

① 2010 年度調査の技術調査を元に再整理を実施

2010 年度調査は、情報システムの保守性を高める構築技術、移行性の確保に向けた構築技術として調査を実施した。概要は次のとおりである。

要求管理		
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。
システム構成関連技術(部品化・連携)		
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 ・ プロッ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	性能および機能のスケラビリティ	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。

安全と情報セキュリティ		
テスト網羅性の保証	<ul style="list-style-type: none"> ・アシュアランスケース ・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	アシュアランスケース	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・アシュアランスの評価	アシュアランスケース	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、可用性やアシュ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	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	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。

② 2011 年度追加した技術課題

2011 年度調査では、次の 4 つの分野を対象に技術課題を抽出した。

➤ システムズエンジニアリング

要求の変化に対応する技術であるエンタプライズシステムズエンジニアリングより技術課題を選定した。

➤ 高信頼システム

予防技術と検証技術があるが、予防技術については 2010 年度調査で実施されていたため、2011 年度調査では検証技術について技術課題を選定した。

➤ 非ウォーターフォール型開発技術

非ウォーターフォール型開発技術より技術課題を選定した。開発技術というよりプラクティスを中心であるが、技術課題に絞り込むため、敢えてプラクティスを除き、開発技術より選定した。

➤ 運用技術

ITIL の体系を対象に技術課題を選定した。

以下、それぞれについて概要を示す。

(a) システムズエンジニアリング

(a-1) コントロール・モニタリングにおける技術課題

超上流工程分析・評価	
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。
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。
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。

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。
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。
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。
ConOps の分析・定義	組織の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。

(a-2) 開発における技術課題

上流工程における横断的連携	
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。
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。
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

(b) 高信頼システム

上流工程、特に要求分析において有効な技術について、次のとおり選定した。

(b-1) 開発における技術課題

上流工程における連携技術		
	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。
	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。

(c) 非ウォーターフォール型開発技術

(c-1) 開発における技術課題

非ウォーターフォール型開発技術		
	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作成などが技術課題として挙げられる。
	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。

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しかえし頻繁にリリースし、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。
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が必要とする。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。
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。

(d) 運用技術

(d-1) 利用・運用における技術課題

運用技術	
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。

5. 技術課題のまとめ

2010年度調査の技術調査を元に再整理して得た技術課題と2011年度調査で選定した技術課題をまとめたものについては(課題名などを一部調整している)を付録-1に掲載した。

VI. まとめ

技術課題の整理結果に基づき、以下では、評価軸としての開発プロセス、超上流プロセス、運用プロセス、EA 開発プロセス(TOGAF ADM)、ステークホルダー要求(St 要求)、品質特性に対して、前述した技術課題の寄与度(貢献が期待される度合い)を総合的に評価する。

なお、委託調査報告では、技術課題の寄与度について、◎、○、△で評価していたが、WG 内で評価結果の見直しを行ったうえ、議論を単純化するため、それぞれ、5 点、3 点、1 点で再計算することとした。

1. 評価軸ごとの寄与度

① 開発プロセス標準に対する技術課題の寄与度

開発プロセス標準 ISO/IEC 12207 と ISO/IEC 15288 に対する技術課題の寄与度を 表 VI-1 にまとめた。同表では、共通部 (ISO/IEC 12207-15288) と固有部に分けて寄与度を提示している。

同表から分かるように、ISO/IEC 12207-15288 共通部分については、開発プロセスに対する超上流分析評価、上流横断連携、要求管理、部品化連携、安全とセキュリティが有効である。一方、運用技術や非ウォーターフォール型開発技術の開発プロセス標準に対する貢献度は低い。

しかし、ISO/IEC 12207-15288 のそれぞれの固有プロセスに対して、必ずしも本稿で提示した技術課題の寄与度が高いわけではない。また、個々の技術課題の寄与度が低いとしてもそれを総合的に活用することで全体として開発プロセスに貢献できる可能性もある。

次に、技術分類ごとに、開発プロセスへの寄与度を見ると、安全とセキュリティ技術の総合評価が最も高くなったことから、開発プロセス全体に貢献できる可能性が高いことが分かる。

ただし、今回の評価は定性的であり、より詳細な評価を実施することにより、評価精度を向上する必要がある。

表 VI-1 技術課題と開発プロセス

技術課題	ISO/IEC 12207-15288	ISO/IEC 12207 固有	ISO/IEC 15288 固有	総合評価
超上流分析評価	3.0	0.6	0.6	1.4
上流横断連携	4.5	1.9	1.4	2.6
要求管理	5	1.4		2.1
部品化連携	4	1.7	2.0	2.6
安全とセキュリティ	5	2.3	2.1	3.1
運用技術	1.5	1.3	1.7	1.5
非 WF 型	0.5	1.0	0	0.3
評価軸平均	3.4	1.5	1.1	1.9

② 超上流プロセスに対する技術課題の寄与度

超上流プロセスに対する技術課題の寄与度を表 VI-2 にまとめる。同表から分かるように、超上流プロセスに対しては超上流評価技術の寄与度が高い。これは超上流プロセスのために超上流分析評価技術が研究開発されているため、当然の結果である。また、運用技術が超上流分析評価技術について寄与度の総合評価が高くなった。これは運用技術が超上流プロセスでも重要な課題となっているためである。

一方で、上流横断連携、要求管理、部品化連携、安全とセキュリティ、非ウォーターフォール型開発技術などの超上流プロセスに対する寄与度の総合評価は極めて低い。これらの技術課題分野でも超上流プロセスに対する取組みが急務である。そうでないとすると、超上流プロセス技術や運用技術と、他の技術とを連携するための新たな方法論の登場が望まれる。

たとえば、今回の報告書では、安全とセキュリティの技術課題分類に中に、アシュアランスケース技術をまとめているが、アシュアランスケース技術は超上流や運用でも重要になるため、適切に組み合わせるための方法が必要である。

表 VI-2 技術課題と超上流プロセス

技術課題	戦略的技術計画	ケーパビリティに基づく開発分析	技術と標準動向調査	エンタープライズ有効性評価	ステークホルダー分析	総合評価
超上流分析評価	◎	◎	◎	◎	◎	5.0
上流横断連携	△	△		○	△	1.2
要求管理					◎	1.0
部品化連携			◎			1.0
安全とセキュリティ			△	△	△	0.6
運用技術	◎	◎	△	○	○	3.4
非 WF 型					△	0.2
評価軸平均	1.6	1.6	1.7	1.7	2.3	1.8

③ 運用プロセスに対する技術課題の寄与度

ITIL V3 の運用プロセスに対して技術課題分類の寄与度を表VI-3 にまとめる。同表から分かるように、当然のことだが、運用技術の寄与度が最も高い。次いで超上流分析評価技術、要求管理技術も運用プロセスに貢献できることが分かる。

逆に言えば、他の技術課題でも、今後、運用プロセスを考慮した新たな技術の発展が必要である。この理由は、従来の技術課題が、開発に焦点をあてているために、運用プロセスに起因する課題を見逃しているためであると考えられる。運用で発生する問題を解決するためには、運用プロセスに根差した技術開発が必要である。

運用プロセスごとに見ると、技術課題の寄与度が高いプロセスは継続的サービス改善とサービス

トランジションとなった。この理由は、この2つのプロセスが、開発と関係が深いためである。これに対してサービスストラテジやサービスデザインは、開発技術の対象とならないサービスを企画設計するプロセスであるため、現在の技術課題では不十分であることが分かる。今後、超上流分析評価技術でも運用技術との連携が求められることになる。

表 VI-3 技術課題と運用プロセス

技術課題	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	サービストランジション	サービスオペレーション	総合評価
超上流分析評価	◎	△	△	△	△	1.8
上流横断連携	△			○		0.8
要求管理	○		△	○	△	1.6
部品化連携	△			△		0.4
安全とセキュリティ	○	△			○	1.4
運用技術	◎	◎	◎	◎	◎	5
非 WF 型			△	○		0.6
評価軸平均	2.6	1.0	1.1	2.3	1.4	1.7

④ EA開発プロセスに対する技術課題の寄与度

TOGAF の ADM(アーキテクチャ開発プロセス)を構成するフェーズごとに、技術課題の寄与度を評価した結果を表VI-4 にまとめる。

表 VI-4 技術課題とEA 開発プロセス

技術課題	初期	ビジョン	BA	ISA	TA	ソリューション	移行計画	実装監督	変更管理	要求管理	総合評価
超上流分析評価	○	◎	◎	○	○	○	○	△	○	△	3
上流横断連携		△	◎			◎					1.1
要求管理				△	△	◎			△	○	1.1
部品化連携				◎	◎	◎	◎	◎	△		1.6
安全とセキュリティ	△		△	○	○	◎		○			1.6
運用技術			○	○	○	◎		○	○	◎	2.5
非 WF 型				△		◎				○	0.9
評価軸平均	0.6	0.9	2.0	2.3	2.1	4.7	1.1	1.9	1.1	1.7	1.7

この表から、超上流分析評価技術と運用技術が EA 開発プロセスとの寄与度が最も高いことが

分かる。一方で、技術課題が全体としてソリューションフェーズに対して最も寄与度が高く、次いで、情報システムアーキテクチャとテクノロジーアーキテクチャに対する寄与度が高くなっていることも分かる。この理由は、従来の技術課題がソリューション開発や情報システムアーキテクチャに焦点を当てているためである。

テクノロジーアーキテクチャ、ビジネスアーキテクチャ、実装監督、要求管理については技術課題の貢献も期待できるがまだ十分ではない。今後は、他のフェーズと同様に、新たな技術開発の取組みが必要である。

⑤ ステークホルダー要求に対する技術課題の寄与度

本報告書で整理したステークホルダー要求ごとに、技術課題の寄与度を評価した結果を表VI-5にまとめる。なお、同表では11個のステークホルダー要求を7項目に再分類している。具体的には、企業の社会的責任と法令遵守、経営の透明感確保と経営スピード向上、市場シェア拡大と顧客確保維持、業務効率向上と業務スピード向上について、ステークホルダーの要求対象が同じであるため、まとめた。

表 VI-5 技術課題とステークホルダー要求

技術課題	企業の社会的責任 法令遵守	経営の透明感確保 経営スピード向上	市場シェア拡大 顧客確保維持	業務効率向上 業務スピード向上	業務品質向上	製品サービス開発改善	ITインフラ変更対応	総合評価
超上流分析評価		△			◎			0.9
上流横断連携		△	△					0.3
要求管理					○		◎	1.1
部品化連携						◎	△	0.9
安全とセキュリティ	○							0.4
運用技術				◎			○	1.1
非WF型	△					◎		0.9
評価軸平均	0.6	0.3	0.1	0.7	1.1	1.4	1.3	0.8

同表から、ステークホルダー要求に対する技術課題の寄与度が低いことが分かる。要求変化の要因としてステークホルダーが重要であることは広く認識されていることから、今後、ステークホルダー分析技術を拡充していく必要がある。

⑥ 品質特性に対する技術課題の寄与度

ISO25010 システム/ソフトウェア品質特性に対する技術課題の寄与度を評価した結果を表VI-6にまとめる。

同表から品質特性に対しては、超上流分析評価技術の寄与度が高くなったが十分ということとはできないことが分かる。他の技術課題については、品質特性に対する寄与度が低いことから、品質特性を向上していくための新たな技術開発が望まれる。

表 VI-6 技術課題と品質特性

技術課題	ISO25010 システム/ソフトウェア品質特性	総合評価
超上流分析評価	2.5	同左
上流横断連携	1.2	同左
要求管理	0.8	同左
部品化連携	1.2	同左
安全とセキュリティ	0.8	同左
運用技術	1.5	同左
非 WF 型	0.9	同左
評価軸平均	1.2	同左

2. 技術課題の総合評価

① 評価

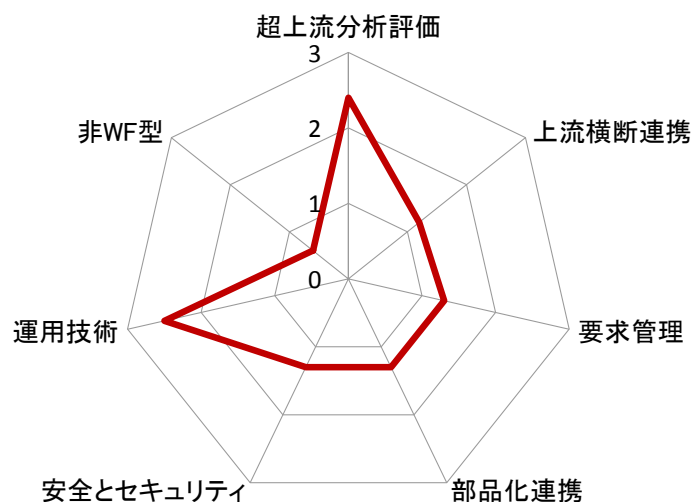
各技術課題の評価結果をまとめると表VI-7 のようになる。総合評価で 3 点以上の技術課題はない。このことから、各技術課題の高度化が必要であることが分かる。総合評価が 2 点台となったのは、超上流分析評価技術と運用技術である。この2つの技術に注目していく必要がある。

これに対して要求管理技術がこれらの技術に較べて低い寄与度になった理由は、現状の要求管理技術が、従来の開発プロセスにしか焦点をあてておらず開発プロセス以外に対して適応できていないためである。新たに、超上流プロセスや運用プロセス、そしてステークホルダー分析に対応できる要求管理技術の開発が強く望まれる。

表 VI-7 技術課題に対する総合評価

技術課題	開発プロセス	超上流プロセス	運用プロセス	EA 開発プロセス	St 要求	品質特性	総合評価
超上流分析評価	1.4	5.0	1.8	3	0.9	2.5	2.4
上流横断連携	2.6	1.2	0.8	1.1	0.3	1.2	1.2
要求管理	2.1	1.0	1.6	1.1	1.1	0.8	1.3
部品化連携	2.6	1.0	0.4	1.6	0.9	1.2	1.3
安全とセキュリティ	3.1	0.6	1.4	1.6	0.4	0.8	1.3
運用技術	1.5	3.4	5	2.5	1.1	1.5	2.5
非 WF 型	0.3	0.2	0.6	0.9	0.9	0.9	0.6
評価軸平均	1.9	1.8	1.7	1.7	0.8	1.2	1.5

図 VI-1 技術課題の必要性評価

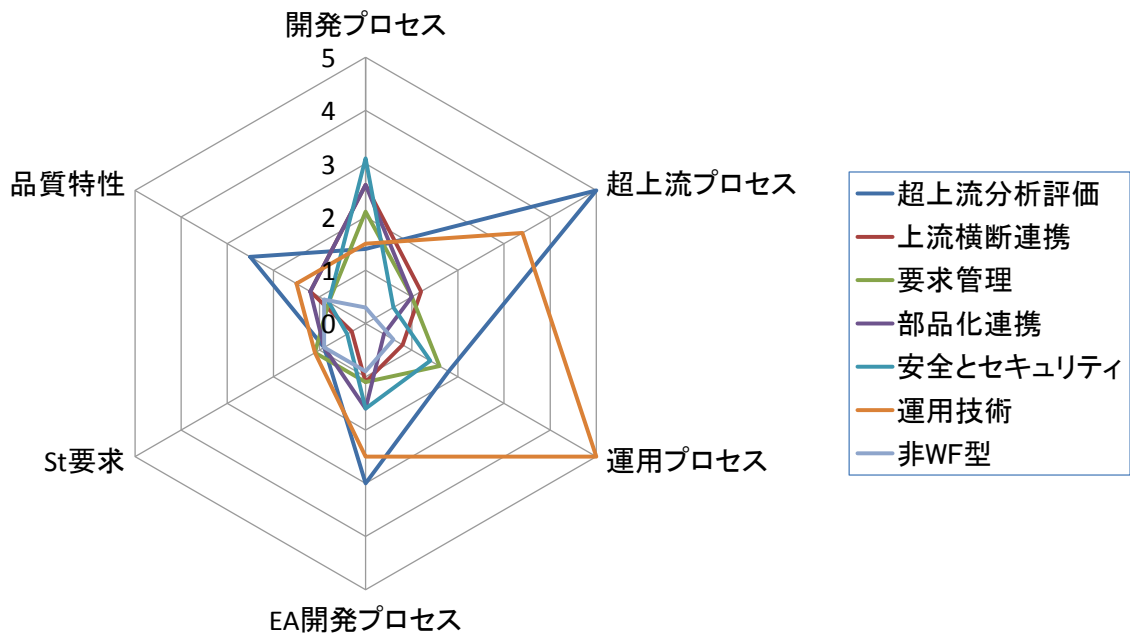


② 分析

図VI-1 に各技術課題を評価軸で 5 段階評価した結果を示す。この図では、超上流分析評価技術と運用技術が超上流プロセスと運用プロセスに直接的に関係することから評価が高くなっている。またこれらの技術とプロセスが相互に関連することも図から良く理解できる。しかし、これらの技術はまだ成熟している訳ではない。今回の評価点が高くなっているのは、この2つの技術に対する期待度の高さを示していると言えよう。また、EA 開発プロセスに対してもこの2つの技術課題の寄与度が相対的に高いことも重要である。この理由は、超上流や運用では特定のシステムやサービスだけを対象にするのではなく、多くのシステムを同時並行的に扱う必要があるためである。

なお、開発プロセスに対しては、超上流プロセスや運用プロセスのように適合する技術課題がなかったが、平均的にすべての技術課題が貢献できることも分かる。

図 VI-2 技術課題の寄与度評価



ステークホルダー要求分析に対しては、重要であるにもかかわらず十分な技術課題の開発ができていない結果となった。本報告書でも示している通り、多数の標準においてステークホルダー分析の重要性が指摘されている。しかし、標準間でもステークホルダーの定義に差異がある。今後、ステークホルダーに対するメタモデルを開発することなどにより標準間の相互連携も必要になると思われる。

VII. 今後の課題など

本 WG での検討結果から、昨今の刻々と変化していくビジネスニーズ、利用環境の変化などに伴う要求の変化に対応する開発については、超上流工程および、ビジネス・サービスの利用・運営を含めた運用工程を軸に検討していく必要がある。

また、現在確立されている技術は、特定工程のソフトウェア開発を成立させるための技術である傾向が見られるが、複雑化するシステム、ビジネスニーズに対応するためには、超上流工程から運用工程まで全てのプロセスで発生するステークホルダー要求を一貫して捉える必要がある。

最後に、『要求発展型開発』に関する今後の更なる検討課題として、以下を挙げる。

◆ 要求変化・発展に対応可能な、超上流プロセス、運用プロセスの技術の確立

従来技術では、超上流プロセスとビジネス(サービス)運用プロセスに対し、対応が不十分であり、ビジネス要求(Outer Loop)に対応する技術の確立が必要。

➤ 運用プロセスに直結した超上流プロセスの検討

ビジネス(サービス)は、開始させること自体が目的ではなく、継続して利用・活用することが目的。また、ひとつの情報システム・ソフトウェア運用だけではなく、情報システム群など既存資源の利用・運用への影響を考慮した検討が必要。超上流プロセスを検討する上では、運用プロセスを考慮した検討が必要。

➤ 超上流プロセス～運用プロセスのトレーサビリティ技術開発

要求管理・分析をはじめとする、ビジネス要求の実現において、ソフトウェア開発での要求管理にとどまらず、超上流プロセスと運用プロセスの追跡性向上を可能とする技術開発が必要。

◆ 全てのプロセスで発生する要求管理・分析技術の確立

開発プロセス以外でも、全てのプロセスで、ステークホルダー要求管理・分析、ケーパビリティ分析技術の確立が必要。

参考文献

- BABOK®翻訳プロジェクト IIBA®日本支部. (2009). ビジネスアナリシス知識体系ガイド (BABOK®ガイド) Version 2.0. IIBA®日本支部.
- HitchinsKDerek,. (2007). Systems Engineering :A 21st Century Systems Methodology. WILY.
- IBM. (2010). IBM Global CEO Study 2010 Japan Report.
- IPA/SEC. (2007). 共通フレーム 2007 第2版. オーム社.
- JapanitSMF. (2011). ITILv3 コア書籍. 特定非営利活動法人 itSMF Japan.
- JapanOpen GroupThe. (2010). TOGAF Version 9. The Open Group Japan.
- Project Management InstituteInc. (2008). プロジェクトマネジメント知識体系ガイド (PMBOK®ガイド) 第4版. Project Management Institute, Inc.
- 一般社団法人 日本情報システム・ユーザー協会(JUAS). (2009). ソフトウェアメトリクス調査 2009.
- 一般社団法人 日本情報システム・ユーザー協会(JUAS). (2011). ソフトウェアメトリクス調査 2011.
- 情報サービス産業協会 (JISA) 一般社団法人. (2010). 要求工学知識体系 (Requirements Engineering Body Of Knowledge:REBOK. 近代科学社.

付録： 委託調査で取り上げた技術課題の一覧

有識者ヒアリングを中心に 2010 年度調査報告書および 2011 年度の委託調査において、技術課題の洗い出しを実施した。各技術課題は技術項目から抽出した。ここでは技術課題に対する課題解決効果、課題解決に向けた留意点・課題等について一覧として挙げる。

ここで注意すべき点は、網羅的にまとめている点ではないことである。既に確立されている技術、「要求の変化」に特化しない技術については、挙げていない。通常開発と比較した場合、要求の変化により対応する技術について、重要技術として取り上げている。

以下、技術課題の一覧を示すが、これらの技術のテクニカルプロセスを基準とした位置を下図に示す。

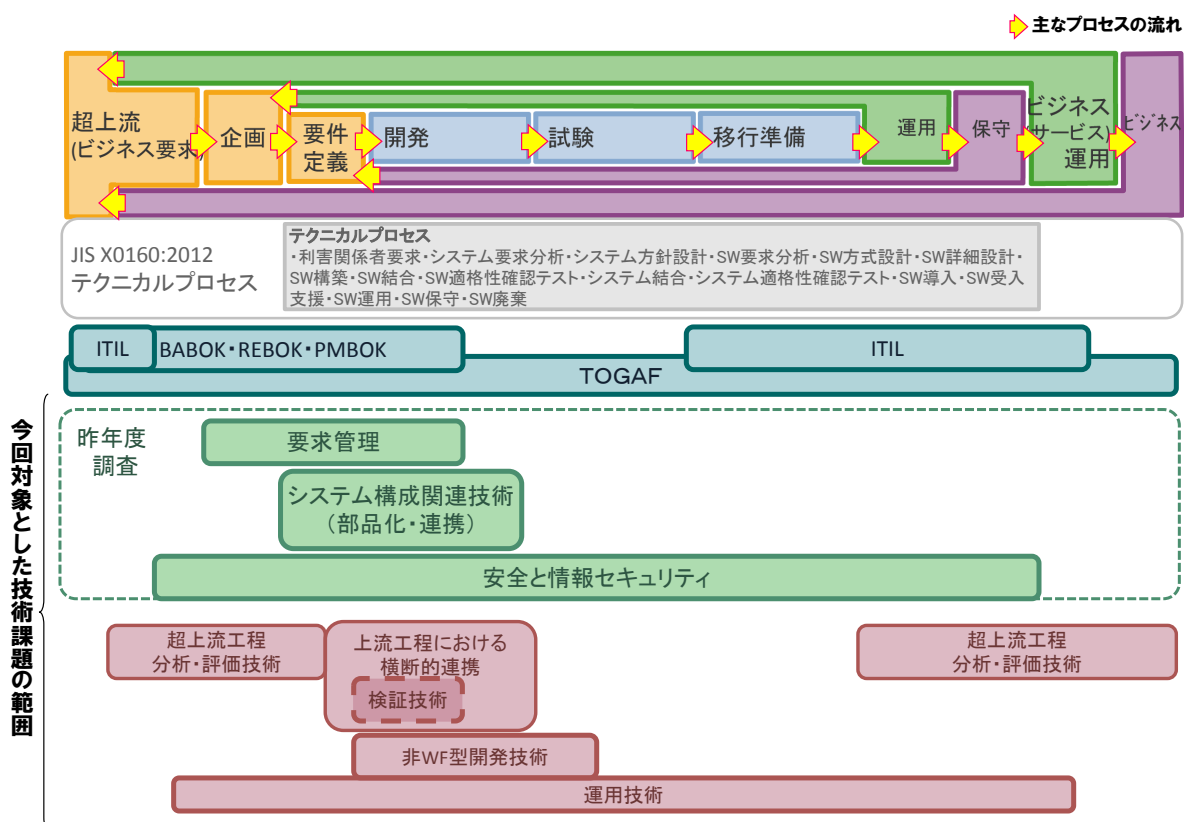
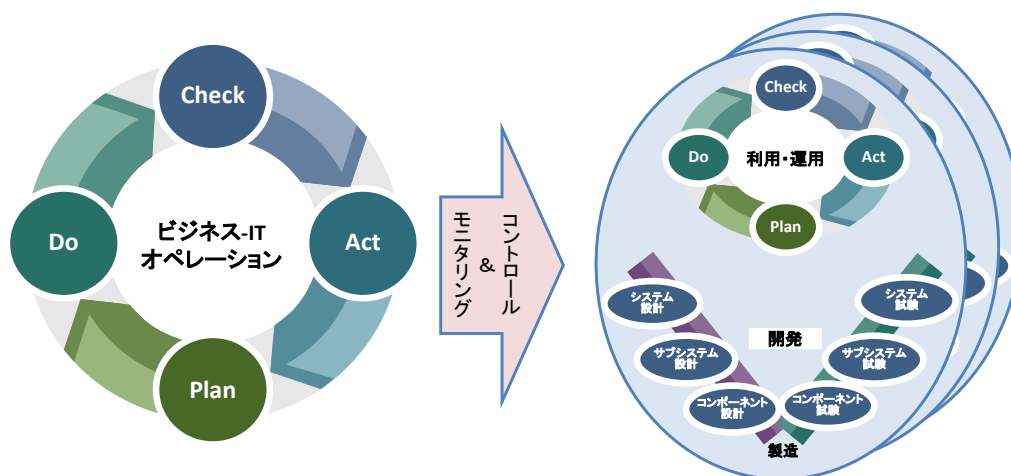


図 プロセス範囲と取り上げた技術課題

今回取り上げた技術課題は、全体として次図のとおり全体をコントロール・モニタリングするシステムITオペレーション、および利用・運用、さらに開発の3種類に分けることができる。以下、技術課題について説明する。

図 付録-1 要求の変化に対応する技術の全体像



1. ビジネスITオペレーションにおける技術課題

今回取り上げた超上流工程分析・評価技術(超上流プロセスにおけるシステムズエンジニアリングの工程分析・評価技術)は、システム構築および運用をコントロールする技術として、ビジネス運用も一部含んだ位置となっている。技術課題の一覧は次のとおりである。

超上流工程分析・評価(システムズエンジニアリング)	
	①システムシンキング (System thinking)
	②ケーパビリティに基づく開発の分析 (Capability-Based Planning Analysis)
	③エンタプライズ有効性評価 (Enterprise Evaluation and Assessment)
	④戦略的技術計画 (Strategic Technology Planning)
	⑤技術および標準動向調査 (Technology Planning)
	⑥ステークホルダー分析 (Stakeholder analysis)
	⑦ConOps の分析・定義

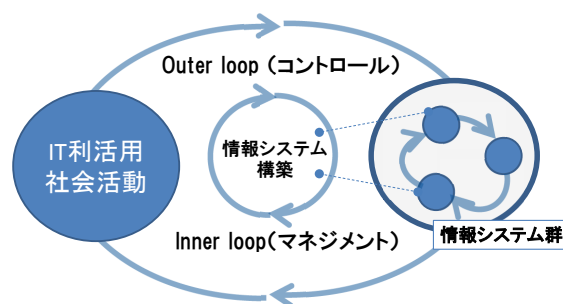
以下、技術課題を説明する。

技術課題： **超上流工程分析・評価(システムズエンジニアリング)**

技術概要

システム開発のための「コントロール」に関する技術である。

要求の変化を前提とした情報システム構築は、トップダウン指向で構築する技術を必要としている。特に各システム構築(System of Systems を含む)における対応を **Inner loop** と呼ぶことがあり、その場合要求の変化の原因である環境の変化を受け止めるべく本報告書で活用している用語ビジネス-IT オペレーションである超上流工程分析評価技術の改善サイクルは、**Outer loop** と呼ばれる。Outer loop を構成する技術は、必要でかつ重要であるが困難な技術課題である。



原出典：Hitchins, Derek, K.: Systems Engineering – A 21st Century Systems Methodology, John Wiley&Sons, 2007.

山本修一郎主査 ホームページ資料を基に作成

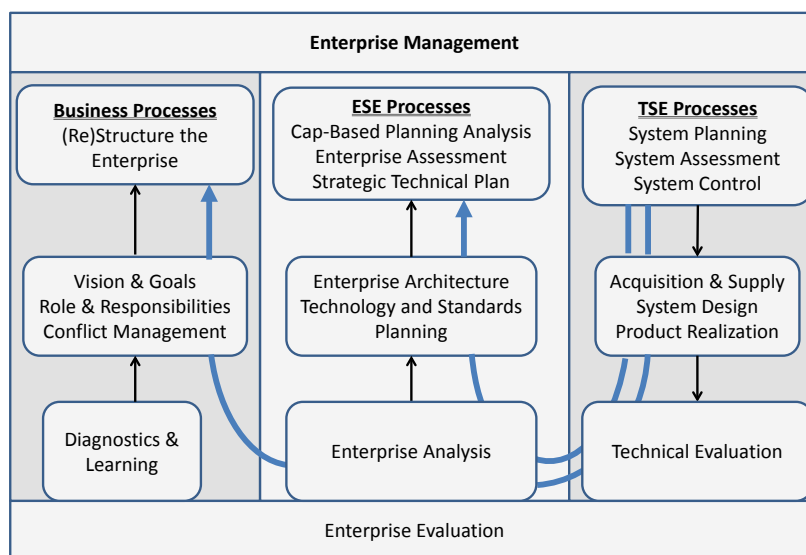
図 Outer Loop と Inner loop (システムズエンジニアリングの例)

この図では、問題スペースとしての IT 利活用社会活動が環境の変化に伴って変化し、求めたものをひとつの情報システムではなく、利用・運用も含めた情報システム群としての影響と考慮して対応し、対応した情報システム群を IT 利活用社会活動が利用していくという意味である。

システムズエンジニアリングにおいて、エンタプライズシステムズエンジニアリング (ESE) が当該内容を担っている。ESE では、要求の変化の原因である環境の変化に対応した情報システム構築に資することも含むケーパビリティを定義し、ケーパビリティの下、各システムのシステム化計画が策定される。

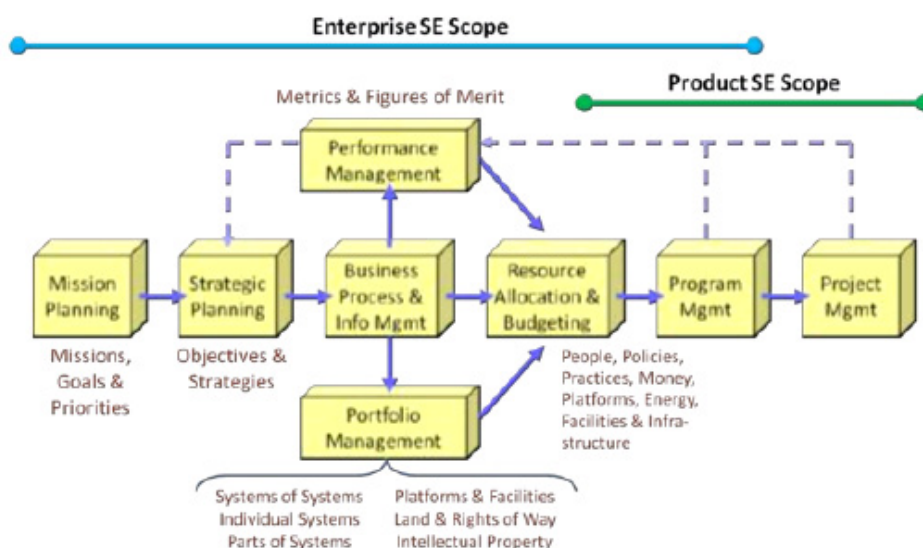
また通常の変動についてはケーパビリティ定義の範疇での要求の変化となるが、想定外のようなケーパビリティを定義していない範囲の要求の変更が生ずる場合は、outer loop の機能としてケーパビリティ自身の改善サイクルを働かせ、各システムの改善も図る。次図に既存システムズエンジニアリング (Traditional Systems Engineering) プロセスやビジネスプロセスとの関係性を示した。ビジネスマネジメントとシステム構築の間、システム化計画を拡張したイメージでのプロセスとして位置づけられている。また、ESE をビジネスマネジメントの支援という視点からまとめたのが、次々図である。ポートフォリオマネジメントなど事業におけるシステム全域が対象となっている。なお、既存システムズエンジニアリングプロセスは、各自活用しているシステム開発プロセスと置き換えても同様に考えても問題ない。

技術課題: **超上流工程分析・評価(システムズエンジニアリング)**



出典:George Rebovich, Jr. and Brian E. White: ENTERPRISE SYSTEMS ENGINEERING
Advances in the Theory and Practice, CRC Press, 2010. Figure1.4 を基に作成

図 ESE と既存開発との関係性例



出典:INCOSE:The Guide to the Systems Engineering Body of Knowledge (SEBOK) Version
0.5

図 関係アクティビティ

技術課題： **超上流工程分析・評価(システムズエンジニアリング)**

技術項目①	システムシンキング (System thinking)
<p>システムズアプローチのための思考法である。ものごとを多面的な分析を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、つまりモデリング能力と考えて良い。要求の変化もモデルを通して考えていかないと影響範囲の分析が難しくなる。</p> <p>他方、MITのNancy Leveson教授の分析によると統合システムの安全性、すなわち単体システムでは全く問題がないものに結合したシステム(System of Systems)にすることで、安全性を損なうケースがあることがわかっており(IPA/SECの2010年度のモデリング技術応用WGにおいても検討されていた)、問題解決のために必要な技術としてシステムシンキングが挙げられている。そこで、技術項目としては異論があるかもしれないが、システムシンキングはシステムズエンジニアリング分野の全域にも求められている最重要項目であるため、敢えてここで挙げることにした。</p>	
技術項目②	ケーパビリティに基づく開発の分析 Capability-Based Planning Analysis
<p>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。ステークホルダー要求の変化に対応した開発を許すことができるシステムを導入するために、そのシステムに必要なとする能力を事前に把握するために全体プログラムの分析を行うと言える。次の3つのケーパビリティがあり、それらを分析する。</p> <ul style="list-style-type: none"> ・システムケーパビリティ(system capability) システムを実現する直接的な能力のことである。これを欠くとシステム構築に不安が残る。また要求の変化に対応したシステム構築能力も必要となる。 ・オペレーショナルケーパビリティ(operational capability) ビジネス運用も含めた利用・運用に関する能力のことである。組織としてのサービスを実施する能力となる。システムを使ってビジネスを成功させる能力だけでなく、変化に対応する能力、要求逸脱への対応能力も求められる。 ・組織ケーパビリティ(organizational capability) 上述を備えた組織能力のことである。組織能力は組織の構成員がシステムを実現そして利用・運用し、ビジネス的な成功を収めるためのコンピテンシーを必要としている。組織ケーパビリティに欠けると様々な問題を発生すること可能性がある。情報システムが「動かない」原因のひとつでもある。 	

技術課題: **超上流工程分析・評価(システムズエンジニアリング)**

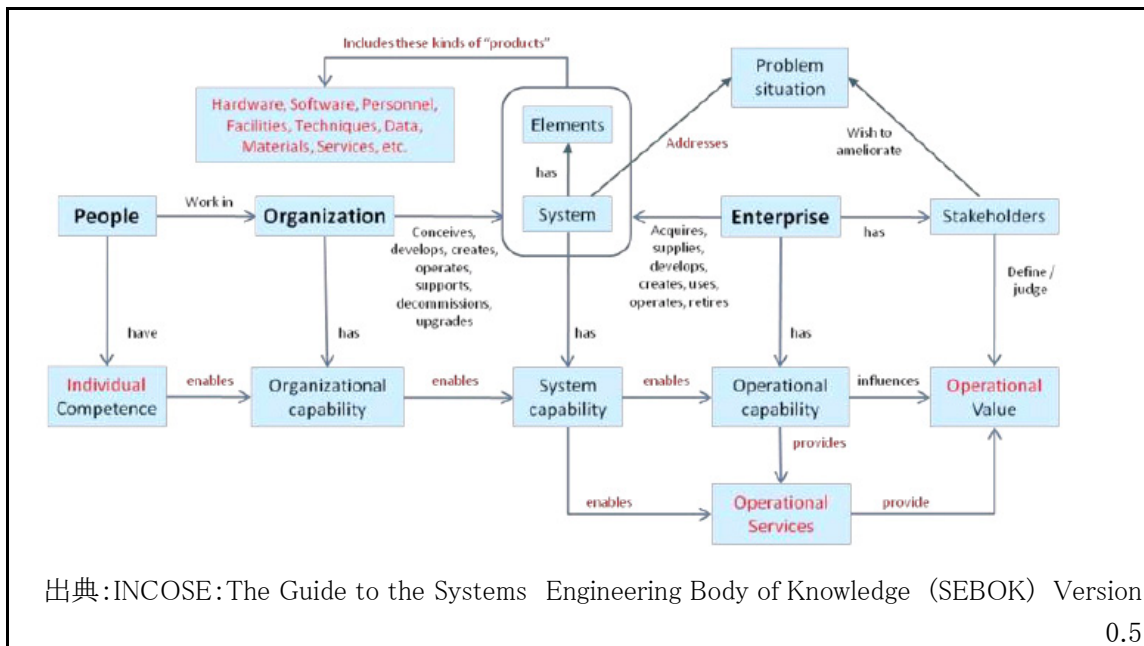


図 3つのケーパビリティの関係例

技術項目③	エンタプライズ有効性評価 (Enterprise Evaluation and Assessment)
<p>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。</p> <p>ここでは、組織ビジョンの実現に向けた進捗度を測定する。測定プログラムを確立し、意図通りに戦略や実装が働いているかを計測する。リスクや目標の達成度、課題の診断等を行う他、感度分析では組織としての要求の変化への対応に関する堅牢性とアジリティの度合いを確認する。シミュレーションを含む分析を実施し、その結果を早めにシステムやビジネスに反映する。</p>	
技術項目④	戦略的技術計画 (Strategic Technology Planning)
<p>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、目標達成可能性を向上させる技術戦略を策定する。戦略的な技術的計画の目的は、情報システム群を活用する組織全体の技術戦略を確立することである。組織の構成メンバを考慮しつつ、標準的な技術利用と新技術利用のバランスをとる。このために技術と標準化調査で策定したロードマップを活用する。</p> <p>このロードマップに組織のもつケーパビリティをマッピングし、整合性と相乗性を確認する。整合性と相乗効果がとれないことは、技術戦略上の回避すべきリスクや追求する可能性との調整を必要とする。結果は、プログラムやプロジェクトの実装ガイドラインの観点でまとめられ、活用される。</p>	
技術項目⑤	技術および標準動向調査 (Technology Planning)
<p>環境からエンタプライズに導入すべき技術の兆候を調査する。標準的技術や新技術を含む今後の技術動向および標準化動向を調査し、組織としてのロードマップを策定する。要求の変化に対応するための情報を提供する。</p>	

技術課題： **超上流工程分析・評価(システムズエンジニアリング)**

技術項目⑥	ステークホルダー分析 (Stakeholder analysis)
<p>ビジネスプロセスと開発プロセスに登場するステークホルダー間の公平性を評価する。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。要求を変化させる原因となるメンバを明確にする必要がある。</p>	
技術項目⑦	ConOps の分析・定義
<p>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。企業や事業レベルでのビジネスレベルで策定されたものを ConOps と呼び、特定の情報システム(System of Systems を含む場合もある)についてを OpsCon (Operational Concept)と呼ぶことがある。成果物については様々な呼ばれ方をするが、概念的な要件を定義したドキュメントであり、ビジネス(運用も含む)で活用するシステムの要件をまとめるためには、非常に重要な書類である。要求の変化への対応も ConOps、OpsCon を活用する。</p>	
課題解決効果	<p>システムシンキングは、非常に重要な概念であり、多面的な視点設定を行い、分析することで、対象となるシステム像を洗い出す。特にモデルを用いる開発技術のためには、この技術を習得する必要がある。それ以外の技術項目は、全項目を挙げて、超上流工程としてエンタプライズシステムズエンジニアリングによる解決を目指す。特に(1-2-2)から(1-2-6)については、文献[1]において、著者らが有効であるとして選択した項目である。宇宙・防衛関連では実績がある手法であり、新しいシステム構築技法への道を開くことが可能となる。</p>
課題解決に向けた留意点、課題	<p>防衛関係では実績や分析が進んでいるが、民生分野では実績があるとはいえない。特にコスト面を下げるためのテーラリングが必要である。他方、一番の懸案事項は、この分野を担える人材が不足していることである。我が国の場合、コンポーネント開発は超一流と言われているが、一方でシステムシンキングができる人材が文系・理系を問わず求められる。</p>
情報のリソース	<ul style="list-style-type: none"> ・ INCOSE: Systems Engineering - Systems Engineering Handbook, 2010. ・ George Rebovich, Jr. and Brian E. White: ENTERPRISE SYSTEMS ENGINEERING Advances in the Theory and Practice, CRC Press, 2010. ・ Nancy Leveson: Engineering a Safer World Systems Thinking Applied to Safety, The MIT Press, 2011. ・ http://www.sebokwiki.org/index.php/Guide to the Systems Engineering Body of Knowledge %28SEBoK%29 v. 0.5 (The Guide to the Systems Engineering Body of Knowledge (SEBoK) Version 0.5) ・ ISO/IEC/IEEE29148:2011 Systems and software engineering – Life cycle processes – Requirements engineering.

2. 利用・運用における技術課題

ここでは運用技術として ITIL に注目した。運用技術に関しては、実際には継続的改善技術のみが関係していることから、継続的改善に関わる広い範囲の工程に位置付けている。

運用技術	
	①ITIL継続的改善

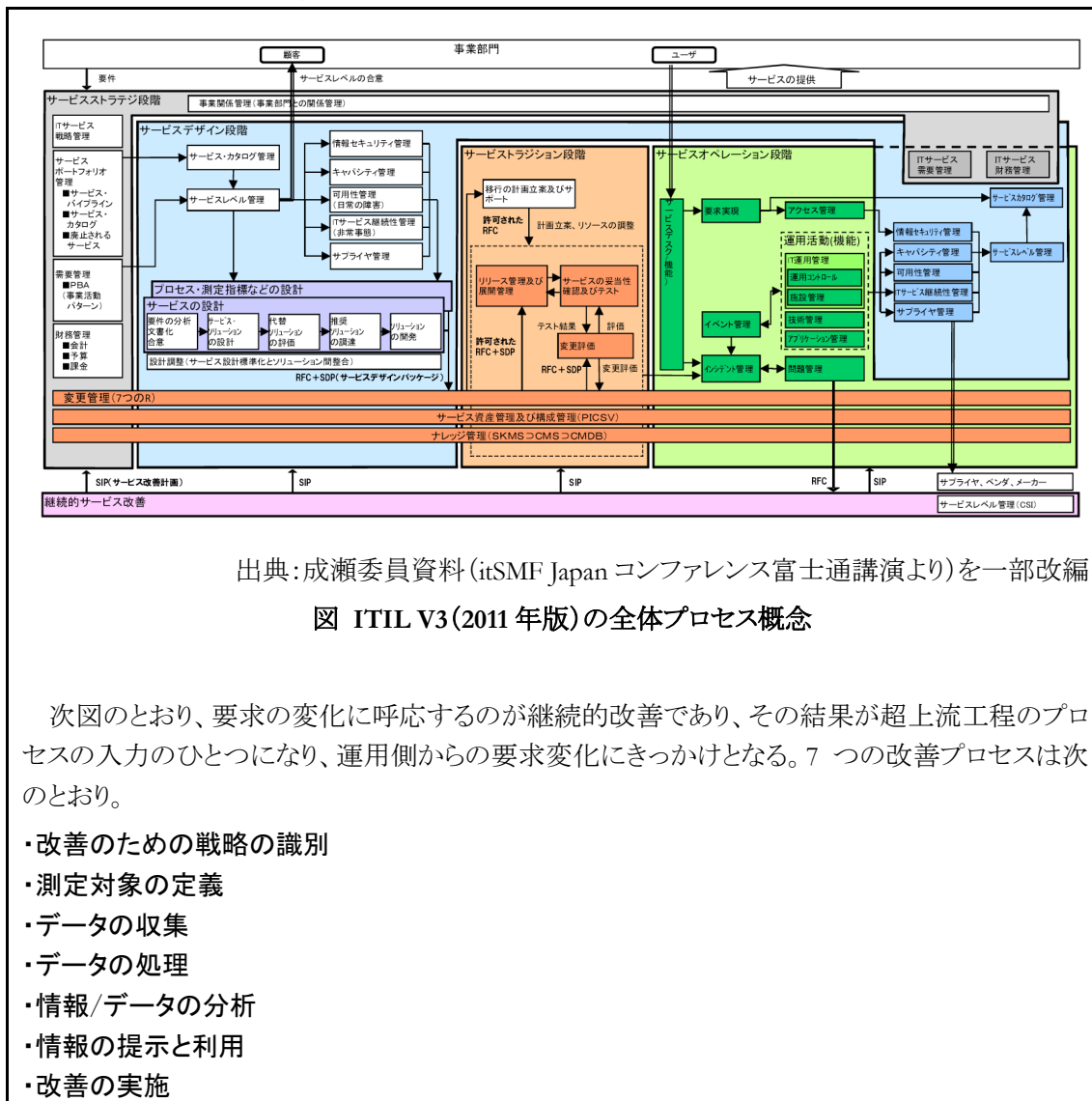
以下、ここで取り上げた技術課題を説明する。

技術課題: **運用技術**

技術概要	<p>要求の変化に対応するシステム構築技術は、要求管理者の下、継続的にシステムが開発されるイメージであり、その前提がITを使ったサービス(ここではITサービスと呼ぶ)の運営が実施されていることである。ビジネス活動の一部としてITサービスの運用があり、ビジネスとITサービスの両方のニーズから情報システムが構築され、要求の変化はビジネスおよびITサービスの運用に関わるステークホルダーの環境変化の気づきから発生すると考えられる。また要求の変化の結果によって変化した情報システムの受け取り手のひとりが運営先であり、運営面のITサービスの運用技術(特にマネジメント技術)がシステム開発と比較して前面にくる。</p>
技術項目①	<p>ITIL(Information Technology Infrastructure Library)は、ITサービスマネジメントのベストプラクティスをまとめたフレームワークである。詳細は日本の代表機関であるitSMF(特定非営利活動法人ITサービスマネジメントフォーラムジャパン)のホームページを参照されたいが、組織全体に展開するメリットとして、</p> <ul style="list-style-type: none"> ・ITサービスに対するユーザと顧客の満足度の向上 ・サービス可用性の向上、事業の利益と収益の増加 ・やり直しと損失時間の削減、リソースの管理と利用の改善によるコスト削減 ・新しい製品やサービスの市場投入までの時間の短縮 ・意思決定の改善とリスクの最適化 <p>が挙げられている。ITILは第2版まではITシステムの運用面に注目されていたが、ITIL第3版ではビジネスとの関わりが協調されており、ビジネス面からの要求の変化、ITサービス面からの要求の変化の両面性を持っている技術である(次図を参照)。</p> <p>次図のとおり、ステークホルダー管理を行い、運用でわかった事項を適切なステークホルダーに届けることになる。継続的サービス改善によって、各段階へSIP(サービス改善計画)が与えられる。事業部門からの要求はサービスストラテジ段階へ送られ、逆にサービスが提供される。</p>

¹ <http://www.itsmf-japan.org/itil/>

技術課題: **運用技術**



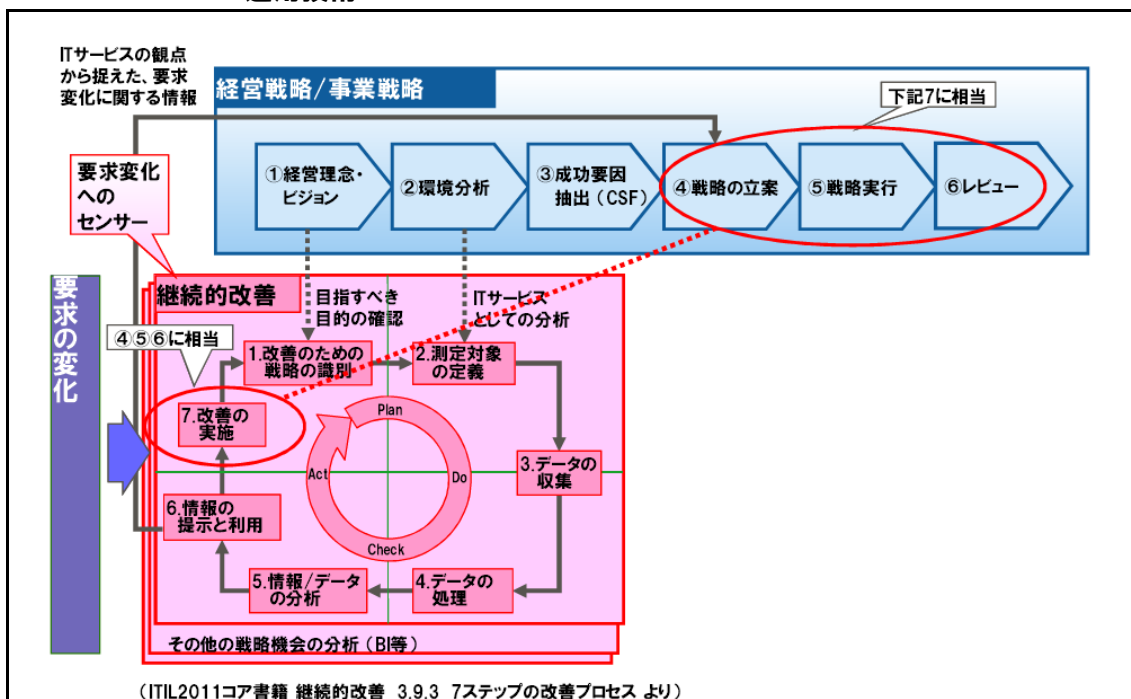
出典: 成瀬委員資料 (itSMF Japan コンファレンス富士通講演より) を一部改編

図 ITIL V3 (2011 年版) の全体プロセス概念

次図のとおり、要求の変化に呼応するのが継続的改善であり、その結果が超上流工程のプロセスの入力のひとつになり、運用側からの要求変化にきっかけとなる。7 つの改善プロセスは次のとおり。

- ・改善のための戦略の識別
- ・測定対象の定義
- ・データの収集
- ・データの処理
- ・情報/データの分析
- ・情報の提示と利用
- ・改善の実施

技術課題： **運用技術**



出典:成瀬委員資料(itSMF Japan コンファレンス富士通講演より)を一部改編

図 IT サービスの観点による要求変化への対応

課題解決効果

IT サービスというどうしてもシステム開発の陰のような存在として見る向きもあるが、本課題では大きく前面に出てビジネスマネジメントと連携してITシステムをコントロールする役割を期待する。

課題解決に向けた留意点、課題

IT サービスは、ビジネス要求の一環として運用要求を策定するが、一方でビジネスと同様にシステム開発側の分析結果の影響から、ビジネスと同様に運用要求の変更(開発側より詳細な情報の提供を受けることも含む)をする必要な局面も出てくる改善サイクルの存在を理解している必要がある。

また人材的に運用要求の分析と実際の運用実施の両面ができる技術者が少ないことが課題である

情報のリソース

- ・ WG 資料
- ・ itSMF (特定非営利活動法人 IT サービスマネジメントフォーラムジャパン) のホームページ (<http://www.itsmf-japan.org/itil/>)

3. 開発における技術課題

要求管理については上流工程、システム構成関連技術、上流工程における横断的連携(上流工程における横断的連携とシステム検証技術)は、アーキテクチャ設計までを範囲としている。一方で非ウォーターフォール型開発技術はソフトウェア構築を範囲としている。

上流工程における横断的連携とシステム検証技術	
	①モデリング、シミュレーション、プロトタイプング (Modeling、Simulation、Prototyping)
	②機能に基づく SE 手法活用 (Functions-Based Systems Engineering Method)
	③オブジェクト指向 SE 手法活用 (Object-Oriented Systems Engineering Method)
上流工程開発力強化	
	①状態遷移系テスト
	②W 字型開発
ロバストネスの確保	
	①ロバストネスの確保のためのテスト技法
要求管理	
要求の明確化	
	①Goal Structuring Notation
	②形式手法
	③アーキテクチャ記述
スコープ決め	
	①変動部分の明確化
	②投資効果と市場と技術変化の見極め
	③ドメイン分析と概念モデリング
トレーサビリティ管理 / ベースライン管理	
	①Requirements and Traceability and Verification Matrix
コントロールケースの活用	
	①コントロールケース
システム構成関連技術(部品化・連携)	
構造の明確化	
	①アーキテクチャ分析
	②形式手法
	③プロダクトライン
	④プラットフォームアプローチ
	⑤スケルトンと部品
部品化	
	①SOA
	②オブジェクト指向
	③業種・業務特化
クラウド活用	
	①性能及び機能のスケーラビリティ
安全と情報セキュリティ(品質保証)	

テスト網羅性の保証	
	①アシュアランスケース
	②トレーサビリティ
	③プロダクトライン
	④形式手法
	⑤テスト網羅性証明
	⑥回帰テスト
	⑦テスト自動化
不測の管理・仮定の管理	
	①アシュアランスケース
運用性・可用性・アシュアランスの評価	
	①アシュアランスケース
共通基盤としての保証	
	①情報セキュリティ
他者に求められる保証	
	①情報セキュリティ保証
非ウォーターフォール型開発技術	
要求の明確化	
	①機能仕様の定義
	②パフォーマンス仕様の定義
	③プロトタイピング
	④実現可能性調査
	⑤ビジネス調査
構成管理	
	①コーディング規約
	②所有権管理
	③トレーサビリティ管理
	④継続的統合
短期間での開発	
	①機能毎の開発
	②コード自動生成
	③シミュレーション
	④テスト駆動開発
	⑤シンプルな設計のための技術
反復による開発	
	①リファクタリング
	②イテレーション計画
	③スプリントレビュー
	④頻繁なリリース管理
品質管理	
	①ソフトウェアインスペクション
	②ペアプログラミング

以下、技術課題について説明する。

技術課題項目： **上流工程における横断的連携 とシステム検証技術**

技術概要	<p>システムズエンジニアリングは、方法論、プラクティスとして非常に詳細なところまで定まってお り、ツールによる支援が望まれる分野でもある。要求管理など、ツール化が進んでいるところもあ るが、一貫した開発の環境も見られるようになった。構築に関係する部分だけではあるが、それ らの方法論となる技術を掲載した。</p>
技術項目①	<p>モデリング、シミュレーション、プロトタイピング (Modeling, Simulation, Prototyping)</p> <p>システムズエンジニアリングでは、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。 一方で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。最近ではHILS (Hardware In the Loop Simulation)、SILS(Software In the Loop Simulation)、MILS (Model In the Loop Simulation)、OILS(Operator In the Loop Simulation)など 多様なシミュレーション環境が活用される。他方、モデル段階が多いが、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が、前者 2 つと相まって、目標とするシステムの分析や 実現のベースに用いられる。</p> <p>システムズエンジニアリングに限らず、ソフトウェアエンジニアリングでも重要な技術である。変 化した要求への対応について上流工程段階で設計検証することは非常に重要である。基本的 な手法として位置づける。</p>
技術項目②	<p>機能ベースの SE 手法活用 (Functions-Based Systems Engineering Method)</p> <p>機能要求の分析と洗い出しを行う。機能要求、性能要求、計画決定要求、仕様および標準化 要求、アーキテクチャ概念や ConOps さらに制約などの情報を基に、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 の各図を得ることができる。</p>
技術項目③	<p>オブジェクト指向 SE 手法活用 (Object-Oriented Systems Engineering Method)</p> <p>システムズエンジニアリングのプロセスを OMG(Object Management Group)が中心となってア ーキテクチャ記述言語であるSysMLで記述する方法論を用意している。リアルタイム系の場合 リ アルタイムアーキテクチャ記述言語であるMarteも活用することがある。なお、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。</p> <p>関係アクティビティとしては次のとおり。</p> <ul style="list-style-type: none"> ▪Analyze Needs, ▪Define Systems Requirements, ▪Define Logical Architecture ▪Synthesize Allocated Architectures ▪Optimize and Evaluation Alternatives ▪Validate and Verify System

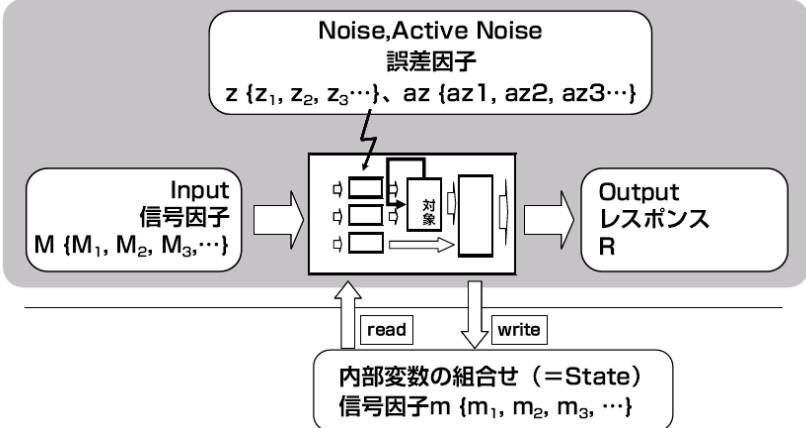
技術課題項目： **上流工程における横断的連携 とシステム検証技術**

課題解決効果	ツール化が可能のため、早期開発およびその保守が効率的に実施できる。要求の変化のビジネス面も含む影響度分析から構築まで、要求の変化に追従したシステムの実現に大きく寄与できる。
課題解決に向けた留意点、課題	ビジネス-IT オペレーションという観点不足しており、モノを作るという考え方である。モニタリングなどをもっと前面に出す必要がある。
情報のリソース	・ INCOSE: Systems Engineering - Systems Engineering Handbook, 2010.

技術課題： **上流工程における横断的連携とシステム検証技術**
－ 上流工程開発力強化 －

技術概要	<p>テスト項目作りは、要件定義書や、仕様書などをベースに作られることから、上流工程成果物の評価手法のひとつとしても考えることができる。検証できない要求や、多くのテスト項目を組み合わせないとテスト項目が作れない仕様書などをみつけるために良いとされている。要求の変化に合わせて早期の短期的な開発を、検証技術により実現する。</p>
技術項目①	<p>上流工程開発力強化（状態遷移系テスト）</p> <p>状態遷移が要求分析で定義できている場合、要求の変化の影響により状態遷移に対応することで、全体システムで支障がないことを確認することは障害発生抑制になる。また状態遷移は要求逸脱という動的な変化に対する、障害発生抑制に上流工程において有用である。</p>
技術項目②	<p>上流工程開発力強化（W字型開発）</p> <p>一般のV字型開発ではテスト設計が後段にくるが、テスト設計を開発前段に移すことで、品質リスクを軽減する。例えば次の効果が考えられる。</p> <ul style="list-style-type: none"> ・テストという別の側面から要求を確認することで、要件の矛盾点や抜け・漏れを防止できる可能性がある。 ・要件とテスト項目の間でのトレーサビリティが確保できる。 <p>テストの困難さを上流工程で把握することで、良質な設計内容の実現や、テスト作業の早期化ができる可能性がある。</p>
課題解決効果	<p>変化した要求を上流工程の早いうちに問題性を除去しておくことは非常に重要なことである。検知活動ではあるが、予防対策として非常に効果があると考えられる。</p>
課題解決に向けた留意点、課題	<p>人材的に要求を作る技術者とテスト項目を作る技術者が重なることが少ない、つまり両方のスキルをもっている技術者が稀有であることが課題である。</p>
情報のリソース	<ul style="list-style-type: none"> ・有識者ヒアリング ・IPA/SEC 編：高信頼化ソフトウェアのための開発手法ガイドブック，2011.

技術課題： **上流工程における横断的連携とシステム検証技術**
－ ロバストネスの確保 －

技術概要	
<p>要求の変化や要求逸脱に強いシステムはある一定のロバストネス(堅牢性)が確保されているシステムと言われる。ハードウェアを含むシステムではロバストネスの大小など重要項目となる。当該開発では変更に強い、別の言い方をすればノイズ成分がある入力に強いコンポーネントを実現するための検証技術に注目した。</p>	
技術項目	ロバストネスの確保 (ロバストネスの確保のためのテスト技法)
<p>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についてのテストを実施する。テスト工程に合わせて組み合わせるべき入力因子について、市場条件となる要因を誤差因子として積極的に取り込んだテストのことである。誤差因子という負荷をかけたテスト(図参照)で、様々な条件下で安定して動作するロバストネスを確保することができる。</p>	
 <p>The diagram, titled 'Test Model (Ralph Chart)', illustrates the flow of information and noise in a system. At the top, a box labeled 'Noise, Active Noise 誤差因子' contains the sets $z \{z_1, z_2, z_3, \dots\}$ and $az \{az_1, az_2, az_3, \dots\}$. A lightning bolt symbol indicates that this noise is applied to the system. On the left, an 'Input 信号因子' box contains $M \{M_1, M_2, M_3, \dots\}$. An arrow points from the input to a central box labeled '対象' (Target), which contains a simplified block diagram of a system. An arrow points from the target to an 'Output レスポンス' box containing R. Below the target box, a 'read' arrow points up and a 'write' arrow points down to a box labeled '内部変数の組合せ (= State) 信号因子 $m \{m_1, m_2, m_3, \dots\}$'. The entire process is enclosed in a grey rounded rectangle.</p>	
<p>出典:IPA/SEC 編:高信頼化ソフトウェアのための開発手法ガイドブック, 2011.</p>	
<p>図 テストモデル(ラルフチャート)</p>	
課題解決効果	
<p>ソフトウェアであっても場面によってはロバストネスを確保することが重要となることがある。要求逸脱に対するアシュアランスケースによる対応にも影響がある考え方である。</p>	
課題解決に向けた留意点、課題	
<p>情報システム開発での適用箇所についての知見を収集する必要がある。</p>	
情報のリソース	
<ul style="list-style-type: none"> ・ 有識者ヒアリング ・ IPA/SEC 編： 高信頼化ソフトウェアのための開発手法ガイドブック 	

技術課題： **要求管理 - 要求の明確化 -**

技術概要	<p>要求の変化に対応するためには、『要求の明確化』が求められる。要求の明確化のために、業務担当と仕様記述者の連携、コーディングレベルの仕様記述が必要となる。他方、『ドメイン分析と概念モデリング』は、固定部分と変動部分の要求を明確化するために必要な技術課題である。</p>
技術項目①	Goal Structuring Notation(GSN)
	<p>アシュアランスケースは要求の変化のうち要求逸脱をはじめとする様々な場面で有効である。要求の明確化の一環で記法の重要性が説かれている。アシュアランスケースを明確に記述するためには GSN を用いる。GSN はゴール指向の表記のひとつで、主張、立証、証拠を順々にツリー上に記述する。文脈、仮定、正当化、戦略、選択に関する記述を付加することで、内容が明確になる。</p>
技術項目②	形式手法
	<p>形式手法には様々な手法があるが、プログラムや機能仕様を、論理系をベースにした記法に従って記述することで、静的検証可能なプログラムや仕様を作成する手法である。</p> <p>手法の用いる論理系によって一般に用途が異なる。また、論理系の知識やスキルがない普通の技術者が使えるよう、様々な工夫がなされている。ここでは論理系に基づいた要求に対応する仕様を作成するために重要な技術として挙げられている。</p>
技術項目③	アーキテクチャ記述
	<p>要求仕様から詳細設計を行い、プログラムモジュールを作成するのが、ソフトウェア開発であると考えている技術者は多い。機能仕様や単体ソフトウェア/システムならそれでも可能であるが、実際には複合化した構造のソフトウェア/システムも多い。いわゆる非機能要求を満たすことを考える場合、コンポーネント(最小単位のシステム構造)間の静的構造と動的振舞いを作成して解決することになる。アーキテクチャ記述は、そのための記法であり、複合システムを記述する方法である。要求の変化の方向がわかっている場合は、それに強い構造を定義する必要がある。</p>
課題解決効果	
	<p>要求仕様が明確になり、下流工程の設計に無理なく引き継ぐことで、手戻りをなくし効率化が図れる。作成された文書の可読性を高めることで、要求仕様作成作業が容易になる。</p>
課題解決に向けた留意点、課題	
	<p>全体として教育・訓練が必要である。GSN は、それを使って保証することを記述すること自体が困難であり、訓練が必要である。形式手法は、論理系の理解と記述の教育、アーキテクチャ記述はアーキテクチャ記述言語などの手法・技法の普及がまず必要である。</p>
情報のリソース	
	<ul style="list-style-type: none"> ・有識者ヒアリング

技術課題： **要求管理 - スコープ決め -**

技術課題: **要求管理 - 要求の明確化 -**

技術概要	
	<p>スコープ決めは要求変化への対応手段であり、要求変化への対応が要求管理そのものと考え、変動部分を設定することである。</p>
技術項目①	変動部分の明確化
	<p>計画的に要求を変化させることが可能な部分を設定するために、変動部分を明確化する技術である。</p>
技術項目②	投資効果と市場と技術変化の見極め
	<p>要求の変化への対応に対する費用対効果の見極めをする技術である。</p>
技術項目③	ドメイン分析と概念モデリング
	<p>具体的な手法のひとつとして、問題の領域の要求の特徴を分析し、さらに顧客の要求をモデル化する。既存の要求を概念モデリングで整理し、新たな要求との違いを見出すことで、最小限の手間での開発を実施する。</p>
課題解決効果	
	<p>システムの枠組みや構造化、そして変動部分を明確にすることで、修正範囲を特定でき、修正のための設計作業が明確になる。また、プログラム変更に際しても、変更部分に焦点を絞って素早く修正を行うことができ、開発工程の効率化が図れる。</p>
課題解決に向けた留意点、課題	
	<p>要求の変化をとらえるためには、ドメインの知識がないと一般的に難しい。ドメイン分析と概念モデリングの考え方が重要な課題である。</p>
情報のリソース	
	<p>・有識者ヒアリング</p>

技術課題： **要求管理 - トレーサビリティ管理/ベースライン管理 -**

技術概要	<p>ビジネス要求とシステム要求、そしてアーキテクチャ、プログラムモジュール、さらにテストケースまでのトレーサビリティが求められている。特に認証の取得が必要なケースはトレーサビリティを求められることが少なくない。非ウォーターフォール型開発のアジャイル型開発のように、比較的小規模なソフトウェア開発に向けた手法を使った、全体システムの開発では、要求の詳細化が均一でなくなる。そのため、複合化した段階的詳細化の管理をする機能であるベースライン管理は重要な技術である。</p>
技術項目	RTVM
<p>Requirements and Traceability and Verification Matrix のことであり、要求と検証に至るトレーサビリティを確保するための行列のことである。実際にはこれを行列としではなく、ツールの機能として実現することで大規模なシステムでの利用が可能となっている。</p>	
課題解決効果	<p>要求定義書や仕様書、変更履歴、テストや障害の記録、ソースコードなどを相互に関連付けることで、仕様変更や欠陥がどのドキュメントに影響しているかの追跡が容易になり、要求を的確に情報システム構築に反映することができる。この結果、開発の手戻りがなくなり、効率化が図れる。また、要求の変化に対して影響する範囲を特定できるため、テスト範囲も明確になる。</p>
課題解決に向けた留意点、課題	<p>特に、非機能要求に対するトレーサビリティを支援する方法論は十分とは言えない状況である。</p>
情報のリソース	
<ul style="list-style-type: none"> ・有識者ヒアリング 	

技術課題： **要求管理 - コントロールケースの活用 -**

技術概要	<p>将来要求がある場合やミスユースケースの発生、さらにSLAが守られなくなった場合の行動パターン、システムの状態により稼働状況を変更する場合などの対応方法を記述することで、要求の変化への対応を行う。</p>
技術項目①	コントロールケース
	<p>詳細は文献[9]を参照されたい。想定する要求の変化を記述するためのメタモデル(もしくは表記方法)である。将来要求を盛り込む場合などに有効である</p>
課題解決効果	<p>許容する要求の範囲とその際の挙動を、アシュアランスケースを使わずに記述できる。</p>
課題解決に向けた留意点、課題	<p>一般的な実績が少ないため、実践力を高め、実績を増やす必要がある。</p>
情報のリソース	<p>・WGでの議論</p>

技術課題: **システム構成関連技術(部品化・連携) - 構造の明確 -**

技術概要	
	部品化は構造の明確化が前提となり、またモデルとビューの分離が部品化のための構造の明確化につながる。フレームワークが部品構造を明確化するために重要である。
技術項目①	構造の明確化(アーキテクチャ分析)
	要求の変化の方向について調査を行い、それに耐えうるアーキテクチャを設定できるか分析し、設計に活かす技術である。
技術項目②	構造の明確化(形式手法)
	要求を変化させる場所を特定し易くするために、矛盾点を除去しておきたい。そこで、形式手法を活用することで該当箇所を限定し、構造の明確化する。
技術項目③	構造の明確化(プロダクトライン)
	プロダクトに要求の一部の違いを反映し、効率的な開発を進めるためには構造の明確化が必要となっている。その実現のための技術がプロダクトラインの技術である。
技術項目④	構造の明確化(プラットフォームアプローチ)
	要求の変化する箇所を明確化し、概念モデルを適用し易くするためには部品活用の枠組みであるプラットフォームを活用した開発が必要とされている。ただし、実際には仕様のズレが存在する場合へのマージンを確保していることなど、前述のロバストネスの確保などが重要関連技術となる。
技術項目⑤	構造の明確化(スケルトンと部品)
	部品を作成するための仕様の枠組みのことである。想定する品質特性の違いには注意が必要である。
課題解決効果	
	変化しないところをフレームワークとしてとらえ、変化するところを、修正したり入れ替えたりする単位として部品化することで、保守の範囲を局所化できる。
課題解決に向けた留意点、課題	
	標準化が一番の課題である。
情報のリソース	
	・有識者ヒアリング

技術課題: **システム構成関連技術(部品化・連携) - 部品化 -**

技術概要	
	コンポーネント化がシステムを分解する上で重要であり、部品探索により部品を見つけやすくすることで、部品の活用が容易になる。アーキテクチャの基本パターンとしての部品化はアーキテクチャ設計する際に非常便利である。
技術項目①	部品化(SOA)
	Service Oriented Architecture のことであり、サービス起点での組み立て方法である。部品は良いが、全体を見るのが苦手なところの克服がポイントである。クラウドコンピューティングの一部では、SOA により組み立てる場合がある。
技術項目②	部品化(オブジェクト指向)
	ひとつのカプセル化の手法であり、部品化の方法として用いられている。 操作方法も同時にカプセル化していることからモジュール性の高い部品化を実現できる。
技術項目③	部品化(業種・業務特化)
	実際に利用する場合には、汎用の部品は基本的なものを除いて役立つものが少ない。そのため、業種や業務ごとに部品提供されることが利用頻度の向上につながる。
課題解決効果	
	既存部品の再利用、SaaS・オープンソースソフトウェアなど外部の部品の活用により、テスト作業を含めた作業期間の短縮ができる。
課題解決に向けた留意点、課題	
	標準化が課題である。また、非機能要求など品質が異なるものを部品として使うことに伴う品質低下の危険性があるため、ロバストネスを保った部品化が求められる。
情報のリソース	
	・有識者ヒアリング

技術課題: **システム構成関連技術(部品化・連携) - クラウド活用 -**

技術概要	クラウドコンピューティングサービスを活用した情報システムの構築を行い、要求の変化に従って、サービスを変更しながら利用することが可能になる。
技術項目①	クラウド活用 (性能及び機能のスケーラビリティ)
	クラウドコンピューティングでは繁忙期や閑散期での要求の違いに対処した効率的な利用を可能とするために、性能と機能についてのスケーラビリティを確保することが重要である。
課題解決効果	システム間、部品間の連携が容易になる。
課題解決に向けた留意点、課題	1社だけでなく、複数社のクラウドコンピューティングサービスや自社のサービスとの連携技術が重要となる。
情報のリソース	・有識者ヒアリング

技術課題: **安全と情報セキュリティ - テスト網羅性の保証 -**

技術概要	
	<p>回帰テストやテスト自動化などは、テスト網羅性の保証手段として重要である。さらに安全性を確保していることを保証するためにアシュアランスケースを記述することも重要である。</p>
技術項目①	アシュアランスケース
	<p>ゴール指向の記法の GSN を用いて、エビデンスから安全性を確保していることを説明するために記述するものである。</p>
技術項目②	トレーサビリティ
	<p>認証関係で仕様通り確実に実装ができていることを示すため、トレーサビリティの結果を提示し、それに対するテスト網羅性を示すことで全体を保証する。</p>
技術項目③	プロダクトライン
	<p>プロダクトラインは安全性を保つために粒度を明確化する際の支援をする。</p>
技術項目④	形式手法
	<p>段階的詳細化による手法により要求の漏れを減らし、テスト網羅性と相まって全体を保証する。</p>
技術項目⑤	テスト網羅性証明
	<p>テストケースが充足していることを示す。実験計画法等の手法により余計なテストケースを削減する。</p>
技術項目⑥	回帰テスト
	<p>要求の変更に対して、影響がないことを確認するためのテストである。</p>
技術項目⑦	テスト自動化
	<p>繰り返しテストを実施するために、単体テストについては自動化されていることも多い。</p>
課題解決効果	
	<p>要求が明確になっていけば、もしくは、影響範囲を特定できるような工夫が設計工程でできていけば、回帰テストを実施する範囲が明確になり、テスト作業を効率化できる。また、要求が明確になっていることで、テストの網羅性や粒度を明確に定めることができ、テスト作業の実効性が高まる。テスト自動化は繰り返し実施されるテスト作業自体の効率化につながる。</p>
課題解決に向けた留意点、課題	
	<p>認証関係でテスト網羅性を求められることも多くなっているが、一方で情報システムでは膨大なテストを実施することで、保証できると誤解している場合がある。重要な局面についてのテスト網羅性を、効率的に確保することが重要である。</p>
情報のリソース	
	<p>・有識者ヒアリング</p>

技術課題: **安全と情報セキュリティ – 不測の管理・仮定の管理 –**

技術概要	<p>不測の事態の発生に対する対処方法は重要であり、そのために絶えず要求を変化させることは重要である。また現実問題として情報システムに実装されているのはモデルであるため、時間の経過とともに要求の逸脱が大きくなる可能性がある。仮定を管理しておくことは不測の事態への対応のためにも重要である。</p>
技術項目	アシュアランスケース
	<p>次の使い方として有効な技術である。アシュアランスケースを記述し、不測への対応の保証、要求の逸脱への対処などすべてアシュアランスケースの記述と連携したシステム構築・運用が重要である。</p>
課題解決効果	<p>不測の事態の発生時・発生前の対処方法を事前に洗い出すことでシステム障害の発生を抑えるものである。</p>
課題解決に向けた留意点、課題	<p>実際にアシュアランスケースを記述するスキルが問題になる。</p>
情報のリソース	<p>・有識者ヒアリング</p>

技術課題: **安全と情報セキュリティ**
－ 運用性・可用性・アシュアランスの評価 －

技術概要	<p>運用性、可用性、アシュアランスを確保するためにこれらの状況を評価し、それに従って場合によっては、必要な要求を変化させることがある。</p>
技術項目①	<p>アシュアランスケース</p> <p>逸脱などの状況をモニタリングする仕組みを取り入れて評価し、必要とあれば要求を変化させる対応させる記述を行う。</p>
課題解決効果	<p>モニタリングにより、障害の発生を事前に抑える仕組みは重要である。</p>
課題解決に向けた留意点、課題	<p>アシュアランスケースを記述する能力をもった人材が必要である。</p>
情報のリソース	<p>・有識者ヒアリング</p>

技術課題： **安全と情報セキュリティ - 共通基盤としての保証 -**

技術概要	<p>仮想化は共通基盤として重要な技術であり、マルチテナントにより利用者側に共通サービスとしての提供が可能となるが、そのための情報セキュリティを共通基盤として確保することは重要である。</p>
技術項目	<p>情報セキュリティ</p> <p>要求の変化に対して、運用面も含めて、共通基盤として脆弱性対策のなされた基盤を提供することは重要である。</p>
課題解決効果	<p>サービス提供側が、サービスレベル、品質、セキュリティの保証を SLA で明確にすることで、サービス利用側でのテスト確認項目が限定される。</p>
課題解決に向けた留意点、課題	<p>運用面での解決が重要である。基盤だけ情報セキュリティを確保しても、運用面に課題が残ると意味が無い。</p>
情報のリソース	<p>・有識者ヒアリング</p>

技術課題: **安全と情報セキュリティ - 他者に求められる保証 -**

技術概要	
	情報セキュリティ認証により、情報セキュリティ機能の適切性や確実性を第三者が評価し、認識することができる。検証・監査により各種ガイドラインに沿った運用手続きが行われていることの第三者による評価がなされ、改善見直しにつながる。
技術項目①	SLA
	サービス品質を利用側、提供側で合意しておくことで保証ができる部分がある。そのために SLA を取り交わす。
技術項目②	情報セキュリティ保証
	情報セキュリティに関する認証を活用して情報セキュリティを確保する。
課題解決効果	
	セキュリティ管理体制の充分性などに関し他者の保証が得られることで、利用者は安心してサービスを利用できる。また、SLA などの対象項目やレベルについて提供サービス種類毎に標準が示されることで、サービス選定時の基準となり、利用者の導入負荷を軽減できる。
課題解決に向けた留意点、課題	
	認証と現実運用の相違点がないよう確認と見極めが重要である。
情報のリソース	
	・有識者ヒアリング

技術課題： **非ウォーターフォール型開発技術 - 要求の明確化 -**

技術概要	<p>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。</p> <p>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を作成などが技術課題として挙げられる。</p>
技術項目①	<p>機能仕様の定義</p> <p>非ウォーターフォール型開発手法では、少なくとも次のイテレーションで実現する機能については明確に定義する必要がある。Planguage を使って機能要求仕様を記述することも考えられる。</p>
技術項目②	<p>パフォーマンス仕様の定義</p> <p>システムパフォーマンスとは、システムがどの程度うまく動くか、どんな利益があるか、環境にどう影響するか、などである。パフォーマンス仕様をインクリメンタルに記述し、改良していくことが必要である。パフォーマンス属性は、具体的には次の3種類に分類される。</p> <ul style="list-style-type: none"> ・品質 - 信頼性、使いやすさなど ・作業能力 ・リソース節約
技術項目③	<p>プロトタイピング</p> <p>要求の明確化を行うには、プロトタイピングが有効である。プロトタイプを多用する DSDM (Dynamic System Development Method) では、システムの評価を行うためのビジネス・プロトタイプ、ユーザビリティ・プロトタイプ、性能/容量プロトタイプ、機能/設計プロトタイプなどが用いられる。</p>
技術項目④	<p>実現可能性調査</p> <p>ビジネス要件を技術的に実現可能かどうか確認し、技術的な解決方法の候補を洗い出すとともに、大まかな時間とコストの見積もりを行う。ビジネス分析者、エンドユーザ、技術者、ユーザ側の上級管理者で行う必要がある。成果物としては、計画概要書などが挙げられる。</p>
技術項目⑤	<p>ビジネス調査</p> <p>システムがサポートするビジネスプロセスの範囲を確定し、要件に優先順位を付けることが重要である。今後の開発の技術的な基盤をさらに明確にし、非機能的な要件を確定するための調査でもある。成果物としては、ビジネス分野定義書、優先度の付いた要件の一覧、システムアーキテクチャ定義書、プロトタイプ概要計画書などが挙げられる。</p>
課題解決効果	<p>機能仕様の定義やパフォーマンス仕様の定義を実施することで要求を明確化していくことが可能である。また、DSDMのような開発手法では、要求の明確化にプロトタイプが多用される。実現可能性調査やビジネス調査を実施することで抽象度の高い処理要件と情報要件を識別し、その要件に基づいて、プロトタイプを構築しコミュニケーションを密に行いながら設計・構築を進め</p>

技術課題: **非ウォーターフォール型開発技術 - 要求の明確化 -**

る。

課題解決に向けた留意点、課題

非ウォーターフォール型開発手法は、要求が日々変化することを前提とした手法であり、ここで示した技術課題だけでなく、イテレーションなどの開発手法全体を通して、イテレーション毎の要求を明確化していることに注意が必要である。

情報のリソース

- ・Tom Gilb, Competitive Engineering: A Handbook For Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage, (Butterworth-Heinemann, 2005)
- ・DSDM CONSORTIUM, DSDM Consortium - Enabling Business Agility, <http://www.dsdm.org/>

技術課題: **非ウォーターフォール型開発技術 - 構成管理 -**

技術概要	<p>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マネジメント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、頻繁に修正をいれつつ、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。</p>
技術項目①	<p>コーディング規約</p> <p>障害発生の抑制に 대응するために、コードの品質を一定に保つ必要があり、コーディング規約を導入する必要がある。</p>
技術項目②	<p>所有権管理</p> <p>短期間の開発サイクルをチームで回すためには、チーム全体が共同で全てのコードに責任を持つ必要があり、同時に各クラスに関しても責任を持つプログラマを決める必要がある。所有者は個人かサブチームかチーム全員かなどが考えられるが、明確に定めておくことが重要である。</p>
技術項目③	<p>トレーサビリティ管理</p> <p>全ての製品(ドキュメント、ソフトウェア、テストコードなど)の進化を管理するため、全ての製品の変更を保存しておかなければならない。プロジェクトライフサイクル中に行われた変更はすべて元に戻せるようにしておく必要がある。</p>
技術項目④	<p>継続的統合</p> <p>非ウォーターフォール型開発手法では、継続的な統合が重要である。Scrum では、少なくとも 1 日 1 回、プロジェクトのチェックインされたコードをすべて統合することが求められている。継続的統合には、ビルド自動化や自動テストなどのツールの活用も必須である。</p>
課題解決効果	<p>短期間の開発サイクルを回すためには、コードを共有することが必須であり、所有権の管理が重要である。クラス毎の所有権の管理、作業成果物の所有権の管理、さらにはユーザへの引渡し権限なども管理することにより、開発速度が向上する。また、コードを共同所有する中で、コーディング規約、トレーサビリティに加え、ビルド自動化や自動テストにより継続して統合することがコードのデグレードの防止に繋がる。</p>
課題解決に向けた留意点、課題	<p>要求の変化に対応するために、顧客に常に見せることができる状態にしておくという意味でも継続的な統合はメリットがある。構成管理自体は非ウォーターフォール型開発手法にかかわらず、重要なことではあるが、継続的な統合のための技術は同開発手法では特に重要である。</p>
情報のリソース	<ul style="list-style-type: none"> ・Craig Larman, Agile and Iterative Development: A Manager's Guide, (Addison-Wesley Professional, 2003) [訳: 児高慎治郎ら, 初めてのアジャイル開発 ~スクラム、XP、UP、Evo で学ぶ反復型開発の進め方~, (日経 BP 社, 2004)] ・Prentice Hall Stephen R Palmer, John Mac Felsing, A Practical Guide to

技術課題: **非ウォーターフォール型開発技術 - 構成管理 -**

Feature-Driven Development, (Prentice Hall, 2002)

- Alistair Cockburn, Crystal Clear: A Human-Powered Methodology for Small Teams, (Addison-Wesley Professional, 2004)
- DSDM CONSORTIUM, DSDM Consortium - Enabling Business Agility, <http://www.dsdm.org/>

技術課題項目： **非ウォーターフォール型開発技術 - 短期間での開発 -**

技術概要	<p>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し頻繁にリリースし、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。</p>
技術項目①	機能毎の開発
<p>非ウォーターフォール型開発手法では既存のソフトウェアコンポーネントを活用することにより短期間での開発を実現する。そのためにも機能毎に開発する必要があり、FDD (Feature Driven Development) では2週間以内に実装できない機能はより小さな部分機能に分割される。</p>	
技術項目②	コード自動生成
<p>短期での開発には、自動コード生成アプリケーションの利用も考えられる。</p>	
技術項目③	シミュレーション
<p>変更内容を複数の開発関係者で素早く確認するために、シミュレーションを活用する。モデリングや影響評価が手法として挙げられる。</p>	
技術項目④	テスト駆動開発
<p>テストを書き、そのテストをパスするように目的のコードを記述するというテスト駆動開発は、短期間で実装と検証を繰り返すものでもあり、短期間での開発に大きなメリットがある。</p>	
技術項目⑤	シンプルな設計のための技術
<p>シンプルな設計は短期の開発には必須であり、そのための技術も多い。システムのメタファやテスト測定基準の要求仕様への記述、オープンエンドのアーキテクチャ、実行可能なスケルトンの構築、インクリメンタルなアーキテクチャの再設計などが挙げられる。</p>	
課題解決効果	<p>機能毎の開発やコード自動生成、シンプルな設計等の技術を活用することにより、開発速度が向上し、短期間での開発が可能となる。</p>
課題解決に向けた留意点、課題	<p>短期間での開発に必要な機能毎の開発やシンプルな設計のための技術には、非ウォーターフォール型開発手法でも多くの手法があり、開発案件に応じて適切な手法を選択する必要がある。</p>
情報のリソース	<ul style="list-style-type: none"> ・ Prentice Hall Stephen R Palmer, John Mac Felsing, A Practical Guide to Feature-Driven Development, (Prentice Hall, 2002) ・ Roger S. Pressman, Software Engineering: A Practitioner's Approach, (McGraw Hill Higher Education, 2004) [訳: 西康晴ら, 実践ソフトウェアエンジニアリング-ソフトウェアプロフェッショナルのための基本知識-, (日科技連出版社, 2005)] ・ Tom Gilb, Competitive Engineering: A Handbook For Systems Engineering,

技術課題項目： **非ウォーターフォール型開発技術 - 短期間での開発 -**

Requirements Engineering, and Software Engineering Using Planguage,
(Butterworth-Heinemann, 2005)

- Craig Larman, Agile and Iterative Development: A Manager's Guide,
(Addison-Wesley Professional, 2003) [訳: 児高慎治郎ら, 初めてのアジャイル開発
~スクラム、XP、UP、Evo で学ぶ反復型開発の進め方~, (日経 BP 社, 2004)]
- Alistair Cockburn, Crystal Clear: A Human-Powered Methodology for Small
Teams, (Addison-Wesley Professional, 2004)

技術課題項目： **非ウォーターフォール型開発技術 - 反復による開発 -**

技術概要	<p>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。</p>
技術項目①	<p>リファクタリング</p> <p>リファクタリングは反復による開発では重要であり、全てのテストが通ることを確認しながら、きめ細かいコードや大規模な設計要素を単純化していく必要がある。</p>
技術項目②	<p>イテレーション計画</p> <p>イテレーション計画を立てることで、イテレーションの度に要求を明確化していく。利害関係者があつまって、優先順位を付け直し、今回のイテレーションの目標を決定するタスクと、チームとプロダクトオーナーのみで集まって要求をどう実現するか熟考し、目標を達成するためのタスクを列挙するタスクなどが含まれる。</p>
技術項目③	<p>スプリントレビュー</p> <p>各イテレーションの最後に、レビューミーティングを行うことで要求を明確化し、イテレーションの意義を高める。チーム、プロダクトオーナー、その他の利害関係者により実施し、システムの機能、設計、長所/短所、チームの作業、今後問題が発生しそうな箇所について明確にする。</p>
技術項目④	<p>頻繁なリリース管理</p> <p>反復による開発における頻繁なリリースのための管理技術としては、スプリントバックロググラフの作成、スタンドアップミーティング、バーンダウンチャートによる可視化、タイムボックスなどが挙げられる。</p>
課題解決効果	<p>リファクタリングをし、設計やコードを単純化しつつ、イテレーション計画やスプリントレビューを実施することで、繰り返しのサイクルを管理しつつ、ステークホルダーからの要求を適切に取り込んでいくことが可能となる。</p>
課題解決に向けた留意点、課題	<p>頻繁なリリースの管理には、多くの可視化手法やタイムボックス等のプラクティスがあるものの、活用には熟練した技術を必要とするものが多い。</p>
情報のリソース	<ul style="list-style-type: none"> ・ Craig Larman, Agile and Iterative Development: A Manager's Guide, (Addison-Wesley Professional, 2003) [訳: 児高慎治郎ら, 初めてのアジャイル開発~スクラム、XP、UP、Evo で学ぶ反復型開発の進め方~, (日経 BP 社, 2004)] ・ James A. Highsmith, Adaptive Software Development: A Collaborative Approach to Managing Complex Systems, (Dorset House, 1999) [訳: 山岸 耕二ら, 適応型ソフトウェア開発-変化とスピードに挑むプロジェクトマネジメント, (翔泳社,

技術課題項目： **非ウォーターフォール型開発技術 - 反復による開発 -**

2003)]"

- ・ DSDM CONSORTIUM, DSDM Consortium - Enabling Business Agility, <http://www.dsdm.org/>
- ・ Alistair Cockburn, Crystal Clear: A Human-Powered Methodology for Small Teams, (Addison-Wesley Professional, 2004)

技術課題項目： **非ウォーターフォール型開発技術 - 品質管理 -**

技術概要	<p>非ウォーターフォール型開発手法では、早期開発による品質の低下、人に依存した品質のばらつきを防ぐための技術が必要であり、ソフトウェアインスペクションやペアプログラミングといった技術が利用されている。</p>
技術項目①	ソフトウェアインスペクション
	<p>リファクタリングは反復による開発では重要であり、全てのテストが通ることを確認しながら、きめ細かいコードや大規模な設計要素を単純化していく必要がある。</p>
技術項目②	ペアプログラミング
	<p>イテレーション計画を立てることで、イテレーションの度に要求を明確化していく。利害関係者があつまって、優先順位を付け直し、今回のイテレーションの目標を決定するタスクと、チームとプロダクトオーナーのみで集まって要求をどう実現するか熟考し、目標を達成するためのタスクを列挙するタスクなどが含まれる。</p>
課題解決効果	
	<p>インスペクションによって欠陥を発見するとともに、チームとしてのコラボレーション能力を高めることが重要である。ペアプログラミングはエクストリームプログラミング(XP)などで多く用いられる手法であり、品質向上に加え、開発効率の向上や教育効果なども期待される。</p>
課題解決に向けた留意点、課題	
	<p>ペアプログラミングは、開発チームの技術者に高いスキルを要求することやペアの相性の問題などがあり、活用には注意が必要である。</p>
情報のリソース	
	<ul style="list-style-type: none"> ・ Prentice Hall Stephen R Palmer, John Mac Felsing, A Practical Guide to Feature-Driven Development, (Prentice Hall, 2002) ・ James A. Highsmith, Adaptive Software Development: A Collaborative Approach to Managing Complex Systems, (Dorset House, 1999) [訳: 山岸 耕二ら, 適応型ソフトウェア開発-変化とスピードに挑むプロジェクトマネジメント, (翔泳社, 2003)] ・ Tom Gilb, Competitive Engineering: A Handbook For Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage, (Butterworth-Heinemann, 2005)" ・ Craig Larman, Agile and Iterative Development: A Manager's Guide, (Addison-Wesley Professional, 2003) [訳: 児高慎治郎ら, 初めてのアジャイル開発~スクラム、XP、UP、Evo で学ぶ反復型開発の進め方~, (日経 BP 社, 2004)] ・ Alistair Cockburn, Crystal Clear: A Human-Powered Methodology for Small Teams, (Addison-Wesley Professional, 2004)"