

Title	ITサービス管理におけるDevOpsとITILに関する一考察
Author(s)	本田, 祐吉
Citation	年次学術大会講演要旨集, 30: 32-37
Issue Date	2015-10-10
Type	Conference Paper
Text version	publisher
URL	http://hdl.handle.net/10119/13219
Rights	本著作物は研究・技術計画学会の許可のもとに掲載するものです。This material is posted here with permission of the Japan Society for Science Policy and Research Management.
Description	一般講演要旨

IT サービス管理における DevOps と ITIL に関する一考察

○本田 祐吉 エヌアイシーネットシステム株式会社

1. はじめに

社会経済の市場環境変化が速い中で、IT を活用した顧客からの要求事項は、効率性、迅速性、経済性、安全性である。

IT サービスを提供する側としては、時間をかけて製品・サービスを企画・開発し、従来型のウォーターフォール的な開発・提供スタイルでは変化についていけない時代になってきている。

現在は、製品・サービスを迅速にリリースし、市場ニーズを把握しながら改善を重ねるアジャイル的な開発アプローチが注目されている。

IT システムの開発と運用面においては、これらに対応するために様々な取組みがなされており、DevOps (Dev:Development, Ops:Operations) は、その中でも特に注目されている考え方である。

本論文は、IT Service Management (ITSM) において、システム運用の視点から DevOps の現状と課題を捉え、IT 融合を考慮し IT Infrastructure Library (ITIL) のフレームワークと連携させることで ITSM の更なる充実を図る検討を行った。

IT 融合とは、IT とビジネス分野の融合により顧客や社会に新たな価値を生み出し、改善から革新的な変革までを含む幅広いイノベーションを創出することと整理されている。

結論的には、DevOps を推進し ITIL と連携させるためには新たなフレームワークの整備とそれを可能とさせるための IT 融合人材の育成が急務であることを明示し、円滑に解決を図るための施策に関して考察を行った。

2. DevOps の定義と現状

2.1. DevOps の定義

DevOps は、Velocity Conference 2008 で Patrick Dubois と Andrew Shaffer の間で論議されたことから広まったものである。DevOps の主旨は、開発 (Dev) チームと運用 (Ops) チームの間に生産性の高いコミュニケーションとコラボレーション文化を育むという最も基本的な考え方に則って IT サービスを提供することである。

開発チームと運用チームとの継続的なチームワークの実践を通じてより良い品質のサービスを迅速かつ確実に構築し、安定したサービスとして提供し続けることを促進するものと言い換えることができる。

2.2. DevOps の概要

IT 業界における DevOps の捉え方は、システム開発に重きを置き、ウォーターフォール型 (WF) からアジャイル型 (Agile) に移行し、さらにリーン手法を用いて開発を進めるソフトウェア開発ライフサイクル (SDLC) として扱う傾向が強い。これは本来の DevOps の定義から外れた動きであり、Dev だけが前面にでて ITSM 上で重要なプロセスである Ops のことがなござりにされている感がある。

IT の現場では、新機能や機能改善といったシステムの追加更新をミッションとする開発チームと、可能な限りシステムに変更を加えずに安定運用を図りたいとする運用チームは、互いに相反するミッションを持つ立場にあり、従来から摩擦が生じている状況にある。表 2.1 に開発と運用の比較を示す。

表 2.1 開発と運用の比較

特徴	DevOps	開発	運用
取組み姿勢		機能を作り上げる	安定したサービスの提供
対応期間		機能を構築する特定期間	サービスが継続する限り対応
対応範囲		担当する分野	ユーザに提供している全ての範囲
管理体制		プロジェクトマネジメント	IT Service Management (ITIL)
改善対応		迅速なPDCA	緩やかなPDCA
技術レベル		ある程度高い	普通
最近の傾向		アジャイル型開発	クラウド型運用

本来の DevOps の目標は、開発チームと運用チームの間に存在する摩擦、リスク、およびその他の制約を取り除き、顧客のビジネスが必要とする頻度と速度で、迅速かつ適切なシステムを構築し、リリース後は安定した運用サービスの提供を可能にすることにある。この点を改善しない限り DevOps は SDLC に閉じたものになり DevOps の発展と成功はありえない。

2.3. Dev の現状

2.3.1. Dev の現場

Dev の現場における問題点として、一貫性のない仕組みや環境の中で開発や展開が行われることから、品質やテスト技法の低さが表面化している。また、IT 部署間のコミュニケーションと理解の欠如により、Service Level Agreement (SLA) 違反や頻繁な停止が発生し、システムの稼働を維持するために、無駄な時間とお金が費やされている。特にセキュリティ対策 (認証・暗号化・伝送

化など)や可用性対策(負荷分散・多重化・バックアップなど)といった非機能要件の検討は必須である。構築に際しては、要求仕様・開発期間・コストなどが優先されると、非機能要件が十分検討されないまま「開発しっぱなし」となり、最終的に運用に関わるコストの増大や安定稼働を低下させることに繋がる。

このような中で、市場のニーズに対応するために開発成果物のリリースサイクルを加速させ、かつ品質を高める取組みを進める DevOps は、IT 業界の生き残りの鍵になるものである。

2.3.2. ウォーターフォール型とアジャイル型

Dev の分野では WF 型から Agile 型への移行が進んでいると一般的に言われている。この流れは一部の条件下では正しい方法であるが、全ての場合に適切なものではない。図 2.1 は左側に WF 型、右側に Agile 型の開発工程の流れを示したもので、どちらも要件定義から始まってリリースで終わるサイクルになっている。

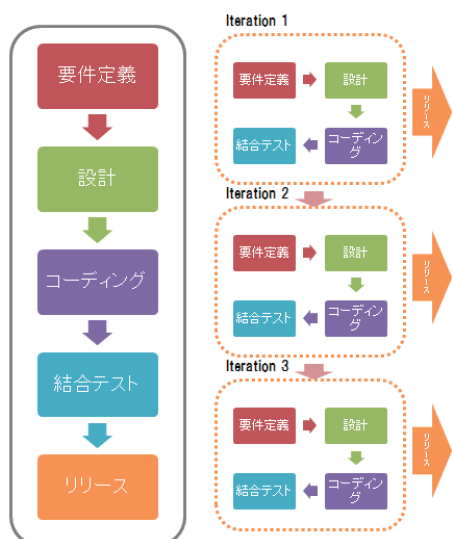


図 2.1 WF 型と Agile 型

WF 型は、システムの開発を「要件定義」「設計」「コーディング」「結合テスト」の工程に分け順に段階を経てリリースする方法で、開発着手前の段階で仕様など全ての重要な決定を行う。ひとつの工程が終わるごとにその工程での成果が文書化されるので管理がしやすいのが特徴である。

しかし仕様決定後に仕様内容に変更が生じた場合は、システム全体に変更の影響が生じることになり、これらに対応することが難しいという欠点がある。さらに、WF 型は、一般的に工期が長期間になることから、初期の段階で顧客の最終的な意見を全てに反映しにくいという欠点もある。

Agile 型は、ビジネス要件上の優先順位を考慮し、システム開発を部分的なパーツに分解して

「イテレーション」と呼ばれる期間ごとに開発し、アプリケーションを提供するまでのスピードを重視することが大きな特徴である。Agile 型では「変化に対応する」というメリットを発揮するために、結果的に WF 型のように予め決定した機能を全て作ることを目的とせず、部分最適を実現しながら全体最適に近づける取組み姿勢である。

従って、開発当初に決めた機能項目を捨てて対応するという行動様式が生じても不思議ではない。

表 2.2 に WF 型と Agile 型の特徴をまとめたものを示す。

表 2.2 WF 型と Agile 型の特徴

特徴	開発方式	ウォーターフォール	アジャイル
対象サービス		基幹系サービス	Web系サービス
要求内容		確実性	迅速性
実装までの期間		中長期	短期
変更の頻度		数か月～数年	日に数度
変更の規模		大きい	小さい
変更のリスク		大きい	小さい
展開のタイミング		定期的	定常的
市場との関連		タイムラグ	ニーズを作る
開発容易性		難	易
開発継続性		断続	連続
開発と運用の連携		分離	緊密
開発の姿勢		仕様決定されたものは全て開発	優先順位に基づく最適な取捨選択

2.3.3. WF 型と Agile 型の開発効率分析事例

日本情報システム・ユーザー協会（JUAS）は 2015 年 4 月 15 日、「ソフトウェアメトリクス調査 2015」を発表している。表 2.3 に、WF 型開発と Agile 開発の生産性、有意差の調査結果を示す。分析対象データは WF が 428 件、Agile が 37 件と、分析データ数が少ないが、全体的な傾向を掴むことができる。表中の「JFS」はシステムの規模を推定するために JUAS が独自に作成した指標で「画面数+帳票数×2/3」で計算されている。調査結果から以下のことが言える。

表 2.3 WF 型開発と Agile 開発の生産性、有意差の調査結果

	WF	アジャイル	
件数	428	37	
総費用(万円)	累乗近似式	2913x ^{0.176}	340.70x ^{0.73}
	JFS1=100	6551	9826
工数(人月)	JFS1=200	7402	16298
	累乗近似式	3.69x ^{0.72}	6.63x ^{0.58}
工期(月)	JFS1=100	102	96
	JFS1=200	167	143
工期(月)	累乗近似式	4.62x ^{0.2}	4.72x ^{0.17}
	JFS1=100	12	10
JFS1=200	13	12	

・工数面では、WF に対し Agile が有利であり、小規模なプロジェクトで 6 ポイント、中大規模では 14 ポイント改善している。

・工期における大きな有意差はないが、WFの方が Agile より長い傾向を示している。

・総費用の面では条件の精査が必要であるが結果からは WFの方が Agile より有利である。

以上の結果から、WF と Agile を比較した場合の大きな優位差は、工数と工期に関して WF よりも Agile が優れていることが分かる。Agile の特徴であるビジネス要件に対するスピード感が優れている点が顧客に受け入れられている要因となっていることが分かる。

2.4. Ops の現状

1970 年代の金融系オンラインシステムの稼働、1990 年代のメインフレームからオープンアーキテクチャやインターネットの登場、2010 年代のクラウドとグローバル化の進展など IT システムは大きな変遷を経て今日に至っている。

Ops すなわちシステム運用は、DevOps の考え方が誕生する前から様々な取り組みを行いながらシステムの安定運用の改革を行ってきている。

2.4.1. Ops の現場

IT システム運用現場のミッションは、システム開発側から受入れたシステムを 24H365D 安定したサービスを提供することと、システム障害が発生した場合はビジネスに与える影響を最小限に抑える措置をとることにある。何も起こらない事が当たり前で、起きたら顧客からクレームを受けるのがシステム運用の実態である。

IT 業界では「開発半年、運用 10 年」と言われるようにシステム運用は顧客と付き合う期間が長いのが特徴である。また保守運用費と呼ばれるコストは、IT 総予算の 70%弱（2003 年当時）から 53%（2012 年）に下がっている調査結果が JUAS から報告されている。現時点でもシステム運用の分野が予算の 50%以上を占めていることから、顧客からは品質の向上とコスト削減が常に求められている。

これらの状況を改善するために 1980 年代に ITIL のフレームワークが英国で誕生し、現在ではグローバルスタンダードになり日本でも定着している。しかし、ITIL と DevOps との間での連携はなく、独立した状態の中でそれぞれが機能しているのが実態である。双方の良い面を活用し相乗効果が生まれにくい状況にあるのは、IT システム全体にとって損失である。この連携をどのようにして実現し推進するかが大きな課題である。

2.4.2. ITIL と DevOps の連携

ITSM の代表的な国際標準が ITIL であり、ITSM に関わる一連のベストプラクティスを体系化し



ている。1980 年代の ITIL V1 以降、内容の改定が進んでおり、ITIL V2 までは運用品質を高める機能が強調されていた。その後、ITIL V3 と最新版はサービスのライフサイクル

という視点に立ち、IT サービスに関わる戦略的思考を強化している。最新の ITIL 2011 edition は、「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」「継続的サービス改善」の 5 分野で構成されている。

ITIL は、ITSM を IT サービスのニーズの把握からシステム設計、開発・導入、運用までの流れをシームレスにするとともに、常に PDCA を回しながら改善を図るプロセスと全体のライフサイクルを重要視したフレームワークである。この体系化されたフレームワークの中に DevOps の考え方を融合することにより、システム設計、開発・導入の部分が強化され、IT サービスの価値創造がさらに高まる。

特に現在は技術面だけでなく顧客視点やサービス視点での変革が求められている。また、企業の IT システムを取り巻く環境も仮想化技術やクラウドサービスの登場により多様性、複雑性が増していることを考慮すると ITIL と DevOps の連携は、避けて通れない時期に来ている。

3. ITIL と DevOps の融合

ITIL と DevOps の融合といっても、単純に接点のプロセス部分を共有化することで解決するものではない。

開発と運用の分離と連携というトレードオフの課題を解決する必要がある。具体的な事項としては、システム開発と運用プロセスに関連する組織の明確な分離を行い、相互牽制体制を構築することが必須事項である。

また、開発と運用の物理的な環境を分離し相互に影響を及ぼさないようにし、リスクの最小化を図ることが求められる。特に問題となるのは、ユーザの個人情報など重要データに直接アクセスできる制限と管理を確実に実施することである。

このように分離すべきものと連携すべきものを整理した上で、開発と運用の壁を取り除き新たな ITSM の体系を作ることが重要である。

開発と運用の全体的なプロセスは図 3.1 に示す通りである。それぞれの開発と運用プロセスにおいて、効率性、迅速性、正確性、安全性、コスト削減を図るように取組んでいるが、2 つのプロセスが接続する部分（受渡と受入）で、課題が発生

しているのが一般的である。

ここ数年 DevOps が注目され 1 日に何度もリリースを行うサービスが出てきている。Dev と Ops の接点に当たるプロセスは ITIL の中で「変更管理」「リリース管理」であり、図 3.1 に示すプロセスの流れの中で、受渡・受入の部分に大きな課題がある。

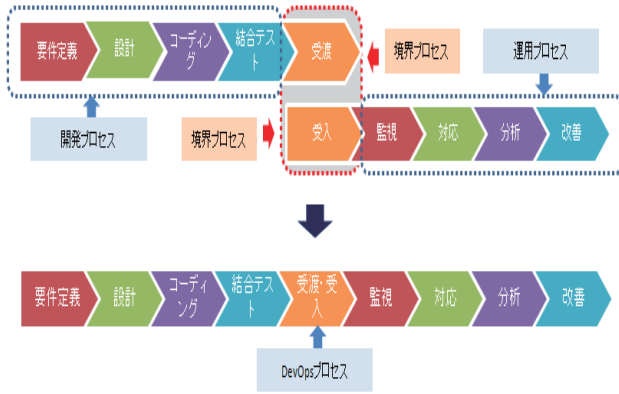


図 3.1. ITIL と DevOps の融合

これらの対策として、開発部門で対応しているテストや本番展開（デプロイメント）の自動化ツールを積極的に使用することと、運用部門においても各種自動化ツール等を利用することが求められている。これにより ITSM の各種プロセスにおいて、プロセスの手順や手続きなどの処理の自動化と高度化を実現できる。

さらに、ITIL 2011 の知見を取り入れてプロセスの成熟度を向上することで、サービスの価値向上に大きく貢献できると見込まれる。表 3.1 は ITIL の 5 つの分野の概要と、DevOps と関連がある項目を整理したものである。特に重要な事項には◎をまた考慮すべき事項には△を付した。

表 3.1 ITIL と DevOps の関連項目

フェーズ	ITIL		DevOps
	分類	ポイント	
サービスストラテジ 【サービスマネジメント戦略】	IT サービス財務管理プロセス	資金確保、高い ROI の実現	
	需要管理プロセス	ユーザーの需要に合わせた調整	△
	サービスポートフォリオ管理プロセス	サービスの強み弱みの明確化	
サービスデザイン 【サービスの設計と調整】	事業関係管理プロセス	顧客のニーズ把握と顧客との関係維持	
	デザインコーディネーション	サービスデザイン全体の調整	△
	サービスカタログ管理	サービスの情報を管理	
	サービスレベル管理	SLA を満たす	△
	キャパシティ管理	キャパシティ要件を満たす	△
サービストランジション 【新規または変更したサービスの運用移行】	可用性管理	可用性要件を満たす	△
	IT サービス継続性管理	リスクを管理し、最低限のサービスを保証	
	情報セキュリティ管理	機密性、完全性、可用性を確保	△
	サプライヤ管理	サプライヤの管理	
サービスオペレーション 【サービス運用管理実施】	移行の計画立案およびサポート	移行計画どリソース確保	△
	変更管理	変更実施の手順確立	◎
	リリース管理および展開管理	変更実施	◎
	サービス資産管理および構成管理	サービス資産、構成を正確に保つ	△
	ナレッジ管理	ノウハウの共有	
継続的サービス改善	組織・役割・機能	サービスデスク、技術管理、IT 運用管理、アプリケーション管理	△
	イベント管理	サービス監視と対応	
	インシデント管理	サービス中断や品質低下の事象に対する早期解決、暫定対応を優先	
	問題管理	根本原因の特定、解決	△
	要求表現	バグレポートなどの要求を処理	
アクセス管理	アクセス権限設定		
		技術、プロセス、サービスの PDCA をまわす	△

3.1. ITIL と DevOps の融合における課題

開発と運用が協力して共通のビジネスゴール

ルを目指す考え方は、積極的に推進すべきものであり、そのためにも ITIL と DevOps の融合は必要である。すでに ITIL は ITSM のフレームワークとして体系付けられているので、DevOps の考え方と開発のプロセスを ITIL に取り込み、統合的なフレームワークとして再構成すべきである。

ITIL と DevOps の融合における課題として、以下の 3 点が挙げられる。ただし、開発ならびに運用に関する技術面の課題は独立して解決できるものであることから、今回は敢えて除いた。重要な課題としては、(1) 仕組みとガバナンスの変革、(2) 見える化と自動化、(3) 新たなフレームワークを支える人材がある。

ITIL が現在の枠組みが完成するまでに 30 年以上を要している。それに比べ DevOps はまだ 10 年弱であることから、完全な融合までには時間を要するものと想定される。IT 業界の発展と IT を活用している産業界の更なる発展を考えた場合には、今回提案した ITIL と DevOps の融合は必須であると考えられる。

3.1.1 仕組みとガバナンスの変革

海外では自社でシステム開発するのが一般的であるが、日本では、主に SIer がシステム開発を請負、そして完成品を納品するビジネスモデルである。このようなビジネス環境の中で、開発と運用が協力し、共通のビジネスゴールへと向かう DevOps を推進することは難しい。成功させる要因としては、SIer だけでなくシステム開発を依頼する顧客とさらに運用する側の 3 者間の仕組みの設定と意識を変える必要がある。

また、一般に企業におけるシステムのガバナンスは、運用と開発を分離し、運用の中でも限られた人だけが本番系システムに関与できるようにすることで本番系システムの安全と安定を確保している。ガバナンスを維持する上で分離が必須の条件である仕組みを、如何にして ITIL と DevOps を融合させるのかに関して、IT 業界内でノウハウを蓄積し、最適な解を見出す必要がある。

3.1.2 見える化と自動化

ITIL と DevOps の融合において重要なのは、ビジネスゴールの方向にプロジェクト全体が向かっている状況を関係者全員が把握できる何らかの指標を設定することである。ITIL では KPI (Key Performance Indicator) の指標を個々のプロセスで設定し、状況を把握しながら改善に役立てている。

この仕組みを DevOps に適用し、見える化を図ることが重要である。

DevOps の関連事項として継続的デリバリーやイ

インフラ構築の自動化をどのように実現するかが直面する課題である。しかしこの部分が自動化ツールの普及などで一般化してくると、次はビジネスゴールの見える化を図るために何を測定し、共有するかがポイントとなる。また、そのための指標と効果を測定するメトリクスの導入が必須となる。

さらにこれらに関する目標を決めるには、システム開発のスタート時点で、顧客、開発者、運用者の3者による意識合わせが特に重要である。最終的にそれらに基づいたメトリクスを策定し、進捗管理を行う必要がある。

3.1.3 新たなフレームワークを支える人材

DevOps は開発と運用の間に生産性の高いコミュニケーションとコラボレーション文化を育む環境作りが趣旨であり、開発と運用のギャップを埋めることが求められる。

したがって、DevOps を推進するために期待される人材としては、ソフトウェア開発とシステム運用の両方の経験があり、対人スキルに関する経験とコラボレーションの手法による仕事の経験があり、ビジネスとテクノロジーのニーズの変化に対応する意欲がある人と言える。

上記に示した必須スキルの中で最も重要なのは、開発や運用に関わる技術スキルよりも、人間関係とコミュニケーションのスキルが最も必要とされている。

一般的に研究・技術開発に従事する人材に必要とされるスキルは、専門知識に加え創造力が求められる。

しかし、DevOps や ITIL のような実社会で即ビジネスに繋がる分野では、新たな発見をするあるいは創造する力よりは、いまある技術を活用しながら如何にしてビジネスに役立てるかを考えることが重要である。即ち技術の応用・適用・調整の分野に長けている人材が必要である。

特に ITIL と DevOps の融合の新たな取り組みを作り出すには、開発と運用の分野間の調整能力と課題解決能力を備えた IT 融合人材が望まれる。

この点に関しては、IT 関連産業の枠を超え、他産業・分野との融合によってイノベーションを起こし、新たなサービスを創造する役割を担う IT 融合人材に関して独立行政法人情報処理推進機構 (IPA) を中心として定義され、検討が進んでいる。

IT融合人材とは、IT融合により価値を創造し、イノベーションを創出する人材を指している。イノベーションの創出において、多様な専門性を持った人材が協働しながら組織として活動することを通じて変革をもたらすものである。IPAから

「IT融合人材育成における組織能力評価指標（成熟度モデル）活用事例」という具体的な事例が示されている。

4. ITIL と DevOps の融合のための施策

ITIL と DevOps の現状と融合に関して述べたが、具体的に進めるためには以下に示す施策を実施する必要がある。

(1) ITIL のフレームワークの中に DevOps の考え方をマージし、開発と運用を統合し本来あるべき ITSM の体系を新たに構築する。これを実現するためには、SIer とシステム運用会社のメンバーで構成されるグループを設立し、詳細に検討する必要がある。活性化を図るためには経済産業省が主導をとり IT 業界をまとめ、新たなフレームワークを作り上げて世界に向けて発信することが重要である。

(2) 顧客、開発者、運用者の3者によるボードを設立し、共通のビジネスゴールを定めることと、3者の関係を円滑に維持し、ゴールに向けた共通意識が育まれる環境を作る。また、新たなフレームワークの評価測定を行うためにも、メトリクスの制定が必要である。ITIL の KPI を参考にすることと、開発で使用されている評価方法を再検討し、開発分野での新たなメトリクスを策定することが求められる。

(3) 新たな ITSM の仕組みを策定する際に必要となる人材は、ソフトウェア開発とシステム運用の両方の経験があるのが望ましいが、最も重要なのは、開発と運用の間にある壁を崩し、新たな風を吹き込むことを可能とする者である。技術スキルの他に関連分野間の課題調整を行うコーディネーターとコミュニケーション力が要求される。

新たな視点としては、顧客のビジネス内容を十分に理解し、ビジネス目標を達成できるように IT でサポートすることを併せて考えることが特に必要である。IPA で定義、検討されている IT 融合人材の育成と実践での活用が重要である。

(4) ITIL と DevOps の融合に際しては、開発と運用の接点となる ITIL 上の「変更管理」と「リリース管理」のプロセスの中で、受渡・受入の部分に注力し、双方で調整・整備することで暫定的な融合を図ることができる。

この実績を通じて ITIL と DevOps の融合の本格的な次のステップへ進むべきである。

以上の対応を可能な限り早期に着手し、実行に

移すことが日本の IT 業界の発展に結びつくもの
と考える。

5. おわりに

IT サービス管理における DevOps と ITIL の現
状と課題に関して概要を示した。

日本の IT 業界は海外と比べてビジネスモデル
が異なることから DevOps と ITIL を融合させる
のは難しい点がある。しかし、IT 産業と IT を活
用したビジネスを展開する産業界の適切な発展
を考えた場合には、DevOps と ITIL すなわち開
発と運用の融合は避けて通れないものであるこ
とから、これを推進するための施策に関して案を
提示した。

DevOps が単なる言葉だけで終わらないよう
にするため、今後とも検討を進め実際の現場で融合
が進むように取組む予定である。

参考文献

- [1]Jan van Bon、ITILRV3 ファンデーション Van
Haren Publishing ISBN978-90-8753-061-7、
2007 年
- [2]itSMF Japan、IT サービスマネジメント事例
に学ぶ実践の秘訣 ISBN978-4-7981-3256-3
翔泳社 2013 年
- [3]斎藤昌義、システムインテグレーション崩壊
～これから Sier はどう生きればいいのか？～
ISBN978-4-7741-6542-4、技術評論社 2014 年
- [4] 日本情報システム・ユーザー協会 (JUAS)「ソ
フトウェアメトリックス調査 2015」 2015 年
- [5] I T サービスマネジメント育成ハンドブック
独立行政法人 情報処理推進機構 (IPA) 2007 年
- [6] 「IT 融合人材育成における組織能力評価指標
(成熟度モデル) 活用事例」 IPA 2015 年
- [7]本田祐吉、「IT サービスマネジメントのサー
ビス品質ならびに人材面に関する現状と課題」、
第 23 回年次学術大会、研究・技術計画学会、2008