

サービス・プロジェクトの成功条件を探る —要求仕様書記述状況の分析—

板倉真由美

サービス・ビジネスはさまざまな産業・業界の中心に位置付けられている。IT産業も同様であり、多くのITベンダがサービス・ビジネスを事業の柱としつつある。ITサービスにおいては、顧客の業務ソフトウェアがプロジェクトの形態で開発されているが、プロジェクトの遅れや稼働後のトラブルなどを低減し、成功率を高めることが重要な課題である。本研究ではソフトウェア開発を題材に要求定義の成果物を分析し、最終的なプロジェクト成果との関連を探った。その結果、成否を左右する項目として“目的”が挙げられた。この項目はITに限らずサービス・ビジネスに対して普遍的な意味を持つものである。

キーワード：サービス、要求定義、プロジェクト

1. はじめに

情報システムは徐々に新たな社会インフラとして浸透し、知らないうちに生活や産業の基盤となっていた。その事実を人々が改めて認識したのはつい最近の大規模システム障害や、情報漏えいなどの事件が相次いで発生したときであろう。私たちが日常生活を過ごすためにITシステムの存在は不可欠であり、すでに生活基盤のひとつになっているのである。それゆえ、その安定性や信頼性は以前に増して強く求められている。近年、社会的な影響が及んだITシステムの問題を振り返ると、ソフトウェアが主要な原因となっていることに気づく。プログラムの動作自体の問題や、想定していた限界値を超えたときの動作の問題、個人情報保護の不備、新機能追加時の影響分析不足、ソフトウェアのテスト不足などである。

ところで、こうしたシステムの多くはITベンダがそれぞれの顧客の業務に合わせて開発することが多い。そのようなシステムは、ITベンダらが顧客からプロジェクトとして開発を受注することがほとんどである。つまり多くの業務アプリケーションはITベンダが提供するサービス・ビジネスによって開発されていることになる。

あらゆる産業のサービス化と同様にIT産業もまたサービス化が加速している。IT産業はメーカーがハー

ドウェア中心のビジネスを行っていた頃とは様変わりし、サービス・ビジネスを事業の柱とし始めている。かつてハードウェアの開発・製造は付加価値をつけた高品質の機器をできるだけ低コストで作ることで、高い競争力と利益を得てきた。ソフトウェア製品は、市場調査や現行製品の評価をもとにベンダが中心となって機能を定義し、コスト上・戦略上適切な開発および販売プランを立て、全体のバランスを考慮した価格を設定しビジネスを行ってきた。

ITサービスでは、業務パッケージ・ソフトウェアを使う場合であっても導入しただけで終わることは少ない。顧客の要望に沿った改善や、パッケージ・ソフトウェアでは足りない機能を作ることなど、付加価値を伴う作業が発生するのである。ITサービスによるソフトウェア開発とはプロジェクトという体制によってソフトウェアを作り上げて提供し、稼働させることと違ってよいだろう。

なお、ITサービスのソフトウェア開発と製品としてのソフトウェア開発は全く同じではなく、それぞれ特徴がある。特徴的な相違点をいくつか挙げてみる。

1. ITサービスにおける開発には顧客の複数のステークホルダ（意思決定者や利害関係者）が直接的継続的に関与する。
2. ITサービスにおける開発ではプロジェクトの終了間際まで要求の追加や変更がある。
3. ITサービスにおける開発では、要求の追加・変更がプロジェクト中にも発生するため、プロジェクト開始段階で精緻な見積りを行うことが難

いたくら まゆみ

日本アイ・ビー・エム(株) 東京基礎研究所
〒242-8502 大和市下鶴間 1623-14

しい。

4. IT サービスにおける開発では、顧客の既存の IT 環境も十分考慮したシステム設計・実装が求められる。

これらを一見してわかるように IT サービスでは顧客の要望・要求が重要なキーワードである。なぜならソフトウェア製品開発に必要な資金は IT ベンダ自身の投資によるものだが、サービスでは顧客から得る売り上げがすべてだからである。その違いは想像以上に大きい。

例えばソフトウェア製品開発であれば、提供する機能は IT ベンダが主導権をもって選択することが可能である。もちろん顧客の要望によって取り込む機能やその時期が変わることはあるが、選択権はベンダにある。しかし IT サービス・プロジェクトでは、顧客側は「予算は早期確定。要件はなるべく後に確定」と考える傾向があり、なかなか要求仕様が確定しない[1]。また、筆者が 2000 年に行った調査では、要求定義以降プロジェクト終了までに要求の追加・変更の無かったプロジェクトは 0 件であった[3]。品質についてもソフトウェア製品開発の場合は、その製品に設定した品質目標を満たせば良いが、IT サービス・プロジェクトでは顧客ごと・プロジェクトごとにその品質目標が異なり、さらに、現システムに悪影響を与えないといった暗黙の条件までも満たす必要がある。

一言でまとめれば IT サービス・プロジェクトの成功とは、顧客から直接寄せられる期待や要求と、現実にプロジェクトを通じてできることのバランスを取り、かつ契約した期間内に予定した予算内で顧客が満足する品質のソフトウェアを完成させることなのである。

2. サービス・プロジェクト成功の意義

さて、そのような困難さを持つ IT サービス・プロジェクトの成功率はいったいどのくらいなのだろうか。これに即答するのは難しい。単に IT ベンダから開示されていないという理由だけではなく、成功・失敗の定義が共有・共通化されていないため一意にできないことや、調査でどのようなソフトウェアを対象にしたか等の条件によって結果が異なるからである。例えば、IPA SEC が 2005 年に IT ベンダを対象に行った調査[1]では、620 件のプロジェクト中「成功した」「概ね成功した」という回答が 90%以上にのぼっているが、2004 年の Standish Group の調査[2]では、プロジェクトの 29%が成功、53%が Challenge (遅延、予算

超過、機能不足など)、18%が失敗 (プロジェクトのキャンセル、納品したが使われない) となっている。[2]の調査結果からは、約 7 割のプロジェクトに何らかの問題があったことになる。

ソフトウェア開発プロジェクトの成功率を高めたいというのは IT ベンダの共通の願いであり、また、ソフトウェア開発委託者 (顧客) の願いでもある。IT ベンダにおいては、1 プロジェクトの失敗やトラブルがその社の利益を著しく侵食してしまうことも珍しくない。ひとたびプロジェクトが危機に瀕すると全社を動員してその修復にあたることはよく耳にするが、その人件費だけを考えても利益への直接的なインパクトであることは明らかである。さらにそのような事態を招いたことによって、顧客満足度が低下して将来のビジネスを失う、社会的に信用が低下する、従業員の精神的肉体的な負担も大きい、などマイナスの影響は計り知れない。顧客にとっても、本来稼動すべき日程を守れないことによるビジネス機会の喪失、低い品質によるストレスや業務の生産性低下、さらにシステム停止などが起きれば、社会的・国際的な信用問題ともなりうる。

多くの IT ベンダはこれらの問題に対応すべく、レビューやインスペクションなどの品質上の問題を早期に解決するアプローチを行っている。しかし、それにも関わらず失敗プロジェクトは簡単には減らない。また、多くの研究でプロジェクト・ライフサイクルの早期に問題を解決することが全体の問題解決コストを抑えられると指摘されているが、早期に問題を見つける方法は単純ではない。前述のレビューやインスペクションはすでに適用されており、ひたすらこのアプローチを強化するだけでは、プロジェクトチームの負担が増すだけで、成功率向上は望めないのである。

筆者は、2000 年に IT サービス・プロジェクトを対象に、失敗したプロジェクトと成功したプロジェクトの違いをさまざまな観点で調査した。その結果、初期に行われる要求定義 (Requirements Definition: RD) への取り組みが重要な要因のひとつであるという知見を得たが[3]、具体的に要求定義における何の要因がプロジェクトの成功に寄与するのかが明らかにできなかった。しかし、結果との因果関係が明確でなければ、要求定義で何に気をつければプロジェクトが成功するのかわからない。プロジェクトの初期に問題を発見すればその効果が大きいということはまた、初期に発見するのは非常に難しいということでもある。

3. 要求定義とプロジェクト結果の関連

ソフトウェア開発にはいくつかのよく知られた方法論があり、その方法論に適したプロセスがある。著名なものとしてはウォーターフォール型、スパイラル型、反復（イタラティブ）型などがある。いずれにおいても、要求定義は初期に行うプロセスとして位置づけられている。大西らは、要求定義とは「一般的にシステム開発の初期に実施する作業で、開発するソフトウェアに必要な機能や性能などをまとめる作業」と定義している[5]。要求定義によってまとめられたソフトウェアに必要な機能や性能は、なんらかのドキュメントにまとめられ、設計へと引き継がれていく。このドキュメントは、SRS（Software Requirements Specifications、日本語では要求仕様書）と呼ばれ、IEEEはSRSの標準[4]を定義している。この標準では良いSRSの満たすべき8つの品質特性を示している。

- | | |
|---|------------|
| 1. Correct | 妥当である |
| 2. Unambiguous | 非曖昧である |
| 3. Complete | 完全である |
| 4. Consistent | 無矛盾である |
| 5. Ranked for importance and/or stability | ランク付けされている |
| 6. Verifiable | 検証可能である |
| 7. Modifiable | 変更可能である |
| 8. Traceable | 追跡可能である |

これら8つの品質特性の他に[4]は、SRSで記述すべき項目とそれに沿ったテンプレートを提供し、SRSの項目ごとに何を記述すべきかを解説している。第1章が概要について、第2章が開発するソフトウェア全般について、第3章がすべての要求についての具体的な記述、第4章が補足である（図1）。

さて、要求定義がソフトウェア開発プロセスの最初に位置づけられ、その成果であるSRSが次のプロセスに引き継がれていく以上、SRSの品質が最終成果であるソフトウェア自身に何らかの影響を与えていることは間違いない。しかし、どんな要素が最終的に致命的な問題につながるのか、またその程度については把握されていない。明らかでない理由のひとつとして、要求定義から最終納品物のソフトウェア自身が完成する間の要因の多さが挙げられる。例えばステークホルダだけを考えても、プロジェクトマネージャを始め、ソフトウェアの骨格を決めるアーキテクト、使用される

- | |
|------------------------|
| 1. イントロダクション |
| 1.1 目的 |
| 1.2 範囲 |
| 1.3 用語・定義 |
| 1.4 参照 |
| 1.5 全体像 |
| 2. 概要説明 |
| 2.1 プロダクト概観 |
| 2.2 プロダクト機能 |
| 2.3 対象ユーザの特徴 |
| 2.4 開発の制約事項 |
| 2.5 要求の前提事項 |
| 2.6 要求のプライオリティ |
| 3. 詳細化された要求 |
| 3.1 外部インタフェース |
| 3.2 機能 |
| 3.3 パフォーマンス要件 |
| 3.4 論理DB要求 |
| 3.5 標準やハードウェア制限による設計制限 |
| 3.6 ソフトウェアシステムの属性 |
| 3.7 特定要求に関する詳細 |
| 3.8 追記 |
| 4. その他関連情報 |

図1 IEEE SRS テンプレートの構成

ツールや技術に特化したスペシャリスト、実装の中心的な役割を担うコーダやテスト、また忘れてはならないのが発注者側である顧客自身も経営者層から情報システム部門、エンドユーザ部門など数えきれないほど多彩である。これらステークホルダ以外にも予算・期間・社内標準・関連する法律などの各種制約事項が次々に挙がる。全部の要因を網羅した上でITサービス・プロジェクトをモデル化しようとしても、パラメータが多すぎるのである。

こうした要素が要求定義作業の後にも多数出現することを承知の上で、筆者は関係者とともに実際のデータを用いてこの問題に取り組んでみた。この試みでは最初から精緻なモデルを構築することを目的にはしない。まず、最終的な問題につながる要因の候補を統計的に把握して、ソフトウェア開発プロジェクトの初期に予防策を施すことが目的である。モデルの精緻化はそれから良い。まずは何らかの関連性を示すことから始め、収集データを増やしていけば、結果的に精緻化が可能になるからである。

本研究に関連する研究としては、情報処理学会ソフトウェア工学研究会要求工学ワーキンググループ[6]（2005年まで活動）にて、SRSの品質特性と、ISO/

IEC 9126[7]および JIS X 9126[8]のソフトウェア製品の品質特性の関連を評価するという試みがある。[6]では要求工学の第一人者らが評価を行うことで、SRSの品質特性と、ソフトウェア製品の特定の性質の関連をみつけようとしている。また少し傾向は違うが、実プロジェクトの定性的あるいは定量的なデータを基に実証性を重視した研究を行い、ソフトウェアの生産性や信頼性向上を目指すエンピリカル・ソフトウェア工学がある。日本の代表的なプロジェクトである EASE[9]では産学共同で研究が進められており、現在は実装における生産性や見積り関連技術が中心であるが、ソフトウェア開発プロセス全般を対象とした取り組みを行っている。

4. 分析対象データ

本研究で分析の対象としたデータは、実際のソフトウェア開発で作成された SRS と、そのプロジェクトの結果である。ちなみに SRS は物理的にひとつの文書であることは稀で、通常は要求定義で作成される複数文書の総称である。

SRSの品質については、専任の品質レビューチームが評価した結果を用いた。レビューチームは、成果物の品質を6段階の数値で評価している。数値が大きくなるほど高い評価であり、漏れが少ない・整合性が取れている・意味が明確である、といった基準を満たすものが高品質な仕様書といえる。反対に低い評価は、そもそもその項目が書かれていない、あるいは記述として意味を成さない、という場合である。この分析ではプロジェクトで作成した SRS 自体を直接分析するのではなく、SRS に対するレビュー項目を IEEE SRS テンプレートの項目とマッピングしてから、各プロジェクトの SRS 評価と関連付けた。マッピングにより、各プロジェクトの固有な章立てにとらわれない分析を行うことができた。

なお、当研究で対象としたプロジェクトは、業務アプリケーションで顧客の要求を基本として開発が進められる IT サービスである。なお、プロジェクトの規模は一定の範囲内にあり、極端に大きいあるいは小さいプロジェクトはなく、規模の点で外れ値に相当するデータは含まれていない。また、最終プロジェクト結果は正常と問題の2カテゴリに分類した。問題プロジェクトの定義は、プロジェクトの3要素である QCD (Q: Quality 品質, C: Cost 費用, D: Delivery 納期)のうち、予定していた費用を超過したケースと予

定していた期日に納品できなかったケースを指す。

4.1 IEEE SRS テンプレート

IEEE SRS[4]は、全5章から構成されており、第5章で良い SRS の構成について述べられている。そこには SRS 自体の構成や、それぞれの章や節に記述すべき内容が解説されている。図1に、筆者が適語訳した5章の SRS を示す (なお、節以下は省略した)。

なお、前述したように SRS レビュー項目と SRS テンプレート (図1) をマッピングした結果、現在の SRS レビュー項目にテンプレートの項目 2.3 に相当する内容が無かったため、以降の分析からは項目 2.3 が欠落している。

5. 正常プロジェクトと問題プロジェクトの SRS 評価平均値の比較

最初に正常プロジェクトと問題プロジェクトの平均値による比較を行った。

IEEE SRS テンプレートにマッチングさせた SRS レビュー評価を、正常プロジェクトと問題プロジェクトの2つのカテゴリに分類し、それぞれの平均値をレーダーチャートに表した (図2)。レーダーチャート上の 1.1, 1.2 といった数字は SRS テンプレートの項目であり、実線は正常プロジェクトの平均値で破線が問題プロジェクトの平均値である。

この分析で以下の2点が観察された。

1. 正常プロジェクトは SRS の記述について問題プロジェクトよりも高い評価を得ている。しかし項目 3.2 と項目 2.1 は例外で、問題プロジェクトのほうが高い評価だった。
2. 正常プロジェクトと問題プロジェクトで顕著な

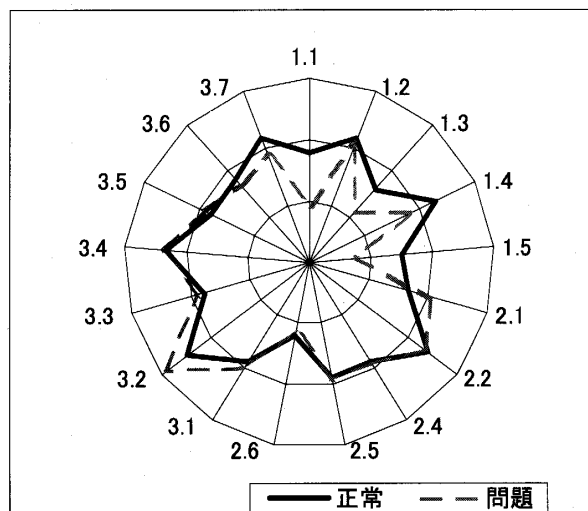


図2 プロジェクト結果と SRS 記述状況

差のある項目は1.1, 1.3, 1.5である。

上記の2つの観察結果に挙げたSRS項目は1.1, 1.3, 1.5, 2.1, 3.2である。これらの項目が何を記述しているかを簡単に述べる。

項目1.1, 1.3, 1.5は、図1にあるように[1. イントロダクション]の中の節である。項目1.1は目的およびこの仕様書が誰に読まれるべきかを記述するところである。項目1.3は略語の正式名を含む用語の定義をするところであり、今後プロジェクトで作成するドキュメントのことも意識している。項目1.5は、このSRSの構成や何が書かれているかを概説する部分である。また、項目2.1はこのプロジェクトで作成されるソフトウェアが関連するシステムや、関連するソフトウェアとのインタフェースの取り方や、既存システムの一部なのかといった情報が含まれる。項目3.2はこのソフトウェアが提供する機能について、入力値や動作の一連の流れなどを含めて記述する部分である。SRSテンプレートは、第2章、第3章と進むにつれて詳細度が上がる構成になっている。つまり、平均値による比較から、正常プロジェクトは概要部分をよく記述しており、問題プロジェクトは概要を書いていないが、詳細項目はよく記述しているということがわかる。

果たしてこの結果はレビューアの知識と一致しているのだろうか？ ベテランのレビューアとこの結果について会話をしたところ、明確に今まで意識したわけではなかったが、最初から機能についての記述が詳細化されていても、系統立てて整理されていないタイプのプロジェクトが後で問題を起こしやすいという認識を持っていることがわかった。

6. 統計手法を適用した分析

平均値比較で観察された事象を統計的に検証するために、プロジェクトの結果を目的変数とし、SRSの各項目を説明変数として分散分析を行った。分析に使用したソフトウェアは、株式会社日本科学技術研修所(JUSE)作成の統計解析パッケージ・ソフトウェアJUSE-MAである。また、6節の分析に際しては[10][11]を参考にした。

目的変数であるプロジェクト結果は、例えば「正常=Y/N」という質的変数である。説明変数はSRSの全17項目の成果物レビュー評価値で、これらを量的変数として扱った。前述したように、レビューの評価値は厳密な意味での量的変数とすることは問題がある

ので、分析の目的を傾向の把握に限定して、厳密な境界値を重視しない方針をとった。

なお、統計的な差異の判断基準である有意の基準値は以下の通りである。

等分散で5%有意 (F分布, 両側検定)

等分散で1%有意 (F分布, 両側検定)

平均値の差で5%有意 (t分布, 両側検定)

平均値の差で1%有意 (t分布, 両側検定)

6.1 正常プロジェクト

分散分析で目的変数に指定したのは、プロジェクト結果「正常=Y」である。SRSの各項目を説明変数とし、正常プロジェクトを目的変数に指定した結果、特定項目で有意な差があればその項目が問題発生に關与している可能性が高い。

分析の結果、正常プロジェクトと問題プロジェクトで有意な統計的差異があったものは以下の通りである。

- ・項目1.1 目的。正常プロジェクトか問題プロジェクトかによる差が表1, 表2に示すように等分散で片側確率5%有意であった。
- ・項目2.1 プロダクト概観。正常か問題プロジェクトかによる差が表3, 表4に示すように等分散で片側確率5%有意であった。

6.2 AIDによる分析結果

次に、AID (Automatic Interaction Detector, 多段層別分析)を行った。AIDは判別分析に対応する解析手法で、どの説明変数とその目的変数に最も寄与

表1 項目1.1についての分散分析結果 (正常)

| | 正常=N | 正常=Y |
|------|------|------|
| 正常=N | — | 21.0 |
| 正常=Y | 1.4 | — |

表2 項目1.1についての検定結果 (F分布・両側検定)

| | | 片側確率 | 検定結果 |
|------|------|------|------|
| 正常=N | 正常=Y | 1.4 | 5%有意 |

表3 項目2.1についての分散分析結果 (正常)

| | 正常=N | 正常=Y |
|------|------|------|
| 正常=N | — | 22.0 |
| 正常=Y | 2.2 | — |

表4 項目2.1についての検定結果 (F分布・両側検定)

| | | 片側確率 | 検定結果 |
|------|------|------|------|
| 正常=N | 正常=Y | 2.2 | 5%有意 |

するかを多段階に二分割を行って層別する分析手法である。AIDで分散分析と同じサンプルを分析し、正常プロジェクトに至るルートを発見することが目的であった。

層別では停止ルールなどを設定することができるが、この分析は特に工夫も無く終了した。最も決定的な説明変数として挙げたのは、項目1.1 目的。であり、その他の項目の寄与は比較にならないくらい小さかった。

7. 分析結果について

以上の分析の結果、読み取れることを以下に挙げる。

- ・項目1.1 目的。については、正常プロジェクトと問題があったプロジェクトで有意な差があった。AIDにおいても最も正常プロジェクトに影響する項目として挙げた。
- ・項目2.1 プロダクト概観については、正常プロジェクトと問題があったプロジェクトでは分散分析上、有意な差があった

興味深いのは、項目1.1が強く正常プロジェクトに影響しているという点である。

これについて少し筆者の考えを述べる。なぜ目的が欠落したSRSを書くプロジェクトは問題になりやすいのか？ 平均値による比較では、項目3.2 機能の評価は問題のあるプロジェクトのほうが高かった。この事実と目的を書かないプロジェクトを併せて考えてみたい。顧客は時として期待や希望を交えて、欲しい機能をベンダに伝える。それに対してベンダはプロジェクトマネジメントの観点から要求の順位付けをして、期間と品質と予算の3つの柱のバランスを取ってソフトウェアを設計し開発する。しかしこのとき、開発するソフトウェアの目的を理解していないとどうなるであろう？ 目的を理解しないと、要求の取捨選択の基準が曖昧になり、結果として機能の記述に走り、矛盾した要求を実現しようとしたり、重要度の高い機能が欠落するなどの状態になったりするのではないだろうか。

以上は筆者の推測であるが、統計解析の結果を再度現実の問題プロジェクト一つ一つに当てはめて検証し、真の原因を追究することが必要である。原因がわからないと対策は立てられないからである。

8. 考慮点

本研究における考慮点を以下に挙げる。

- ・レビュー者個々人の能力のばらつき
レビュー者は専任でプロジェクトの成果物レビューを行っており、ツールやチェックリスト等により、個人差を最小化している。すでに数百を超える品質点検を実施してきたチームによる評価であり、評価品質のばらつきは小さいと言って良い。
- ・SRSの評価数値について
SRSはある時点に、あらかじめ定められた判断基準によってレビューアが判断した相対的な値である。本研究ではこの評価値を統計処理の際に数値として扱った。ただし傾向を把握することに限定した用途であり、厳密な数値を求めることを目的としていない。
- ・適用プロジェクトの制限
本研究はある限定された範囲内のデータを用いており、当分析の結果がすべてのソフトウェア開発プロジェクトに当てはまるということではない。

9. まとめ

サービス・ビジネスは、多くのITベンダにとってビジネスの柱である。顧客に対してはより価値あるサービスを提供し、業界に対しては競争力があり、かつ適正な利益を生み出すという努力が常に続けられている。ここにはIT産業に限定しないサービス・ビジネスの本質がみえる。

本研究における分析の結果、サービス・プロジェクトの成功要因として“目的”という普遍的な項目が得られた。この知見は他のサービス・ビジネスにも有効ではないだろうか。サービス・プロジェクトでは、顧客との継続的で直接的なコミュニケーションを通じて要求を獲得して仕様化し、顧客の要求に応じた成果を作り上げていた。サービスとは提供者と受益者の間に生じる時限的で無形の価値であるという本質を考えたとき、本研究の結論がシステム的な要素ではなく“目的”という普遍的な項目に帰結したことは非常に示唆的である。目的が無い、もしくはステークホルダが目的を理解していないサービスやサービス・プロジェクトは、IT分野に限らず失敗する傾向を有するのではないだろうか。これは検証する価値がありそうである。

最後に、ますます大きな責任を担っていくITシステム・プロジェクトの成功率を高めるためには、ソフトウェア工学やプロジェクトマネジメントと同様に、サービスの本質に踏み込んだ研究が不可欠である。今後も筆者は実データを基としてソフトウェアの成否に

つながる成功/失敗のメカニズムを研究し、より高品質な IT サービスの推進に貢献していきたい。

参考文献

- [1] 独立行政法人 情報処理推進機構 (IPA) ソフトウェア・エンジニアリング・センター (SEC) 著作監修, 『ソフトウェア開発データ白書 2005』, 日経 BP 社, 60, 2005
- [2] The Standish Group, 2004 Third Quarter Research Report CHAOS demographics http://www.standish-group.com/sample_research/PDFpages/q3-spotlight.pdf
- [3] Kamata, Mayumi I., and Yoshizawa, Tadashi, "Key Factors for Managing Small Scale Software Projects," Proceedings of World Multiconference on Systemics, Cybernetics and Informatics, Volume I, ISBN 980-07-6687-1, SCI 2000, 47-52, 2000
- [4] Software Engineering Standards Committee of the IEEE Computer Society, IEEE Std 830-1998, "IEEE Recommended Practice for Software Requirements Specifications," ISBN 0-7381-0332-2, 1998
- [5] 大西 淳, 郷 健太郎, 『ソフトウェアテクノロジー 9 要求工学』, 共立出版, 2001
- [6] 要求工学ワーキンググループホームページ, <http://www.selab.is.ritsumei.ac.jp/~ohnishi/RE/rewg.html>
- [7] ISO/IEC 9126, ISO and IEC, "Software Engineering-Product Quality," 2001
- [8] JIS X 0129, 日本規格協会, 『日本規格協会ソフトウェア製品の評価—品質特性及びその利用要領』, 2003
- [9] Empirical Approach to Software Engineering project, <http://www.empirical.jp/top.html>
- [10] 森口繁一編, 『新編 統計的手法 改定版』日本規格協会, 1996
- [11] 吉澤 正, 『統計処理 情報処理入門コース 8』, 岩波書店, 1995