

RISDM : ソフトウェア要求仕様書の インスペクションデザイン方法論の提案と適用評価

齋藤 忍^{†1†2} 竹内 睦貴^{†1} 山田 節夫^{†2} 青山 幹雄^{†3}

ソフトウェア要求仕様書(SRS)の品質はプロジェクトの成否の鍵である。従来、SRS の品質を評価・改善するためのインスペクションは提案されているが、その方法は経験や知識に依存しているため、インスペクションの品質保証が困難である。本稿ではインスペクションの体系的なデザイン方法論 RISDM (Requirements Inspection System Design Methodology)を提案する。提案方法論ではインスペクションのメタモデルからインスペクション方法を体系的に設計する。まず、SRS の満たすべき品質としてプラグマティック品質特性(PQC: Pragmatic Quality Characteristic)を定義し、それに対するSRSの読解技術としてPBR(Perspective Based Reading)に基づく質問セットの生成方法を提案する。また、質問セットをSRSに効果的に適用するために、SRSの構成要素ごとに質問セットを対応付けるインスペクションポイントセットの作成方法も提案する。RISDMによりデザインされたインスペクション方法を適用することにより、品質特性や構成要素毎にSRSの詳細な品質評価が可能となり、具体的なSRSの改善アドバイスが可能となった。提案方法論を2008年から企業内の第3者インスペクションの取り組みに適用し、そのデータの解析からRISDMでデザインしたSRSのPQCや読解技法は、プロジェクトのリスク予測やSRSの品質改善に有効であることを検証できた。

RISDM: A Requirements Inspection Systems Design Methodology and Its Evaluation Based on Practical Cases

SHINOBU SAITO^{†1†2} MUTSUKI TAKEUCHI^{†1} SETUO YAMADA^{†2}
MIKIO AOYAMA^{†3}

The quality of the SRS (Software Requirements Specification) is the key to the success of software development. The inspection for the verification and validation of SRS are widely practiced. However, the techniques of inspection are rather ad hoc. This article proposes RISDM (Requirements Inspection Systems Design Methodology) to design the RIS (Requirements Inspection System) to be conducted by a third-party inspection team. The RISDM includes a meta-model and design process of RIS, PQM (Pragmatic Quality Model) of SRS, and a technique to generate inspection question set based on the PQM and PBR (Perspective-Based Reading). We have been applying the RIS designed by the proposed RISDM to more than 140 projects of a wide variety of software systems in a company for five years. By analyzing the statistics, we discovered some key quality characteristics of SRS strongly correlate to the project cost and, and help to predict the risk.

1. はじめに

近年のソフトウェア開発ではプロジェクトメンバの専門職化が進み、要求定義工程と以降のソフトウェア開発工程では異なるチーム(会社・組織等)で実施されるケースが多くなっている。そのため、要求定義工程と後続のソフトウェア開発工程の間の橋渡しを担う成果物であるソフトウェア要求仕様書(SRS: Software Requirements Specification)の重要性は高まってきている。

SRSの品質はプロジェクトの成否と深く関係しており、その品質を確保する取り組みとして、従来の要求工学では要求の検証や妥当性確認(V&V: Verification and Validation)が一般的な活動として位置付けられる。特にSRSのインスペクションやレビューの技法は広く実践されている[2,4,22]。しかしながら、近年の調査で、多くの企業やプロジェクトではSRSに対するインスペクション方法を定めて

はいるが、ほとんどのケースでそれらが適切に適用されていないとの報告がされている[7,12]。

SRSのインスペクションが組織的に適用されていない原因の1つとして、SRSのインスペクション方法をデザインする体系的な方法論が未整備なことが挙げられる[29]。そのためプロジェクトではSRSのインスペクション(SRSの品質評価の仕組みやSRSを読むための方法等)を個別に定義・実践しなければならない。結果としてSRSのインスペクションの成否は経験者の知識やスキルに依存してしまう。

本稿では上述の課題解決のため、SRSのインスペクションをデザインするための方法論(RISDM: Requirements Inspection System Design Methodology)を提案する。RISDMではSRSの満たすべき品質としてプラグマティック品質特性(PQC: Pragmatic Quality Characteristic)を定義する。また、PBR(Perspective Based Reading)に基づくインスペクションの読解技術を定義する。本稿ではRISDMの有効性を示すため、RISDMの企業の実プロジェクトへの実践適用と評価を通して、以下の2つの仮説に対する検証を試みる。

(1) RISDMでデザインしたSRSのプラグマティック品質特性(PQC)は、開発工程のリスク評価に有用である。

†1 (株)NTT データ
NTT DATA Corporation
†2 日本電信電話株式会社
NTT Corporation
†3 南山大学
Nanzan University

(2) RISDM でデザインした PBR に基づく SRS の読解技術は、SRS の改善に有用なアドバイスが指摘できる。

本稿の構成は以下のとおりである。2 章では関連研究を紹介する。3 章では SRS のインスペクションデザイン方法論、RISDM の全体像を示す。4 章では RISDM による SRS のインスペクションのデザイン方法を記す。5 章では RISDM の企業での実践状況を報告し、6 章では RISDM のプロジェクト適用による評価と仮説検証の結果を記す。7 章で評価結果に基づく RISDM の有効性を考察し、8 章でまとめと今後の課題を述べる。

2. 関連研究

2.1 インスペクションの基本要素

ソフトウェア開発のためのインスペクション技法は厳密に定義され、広く認知もされている[10,24]。要求工学の分野でも要求のインスペクションやレビューを支援する研究は数多く存在する[1,2,4,13,16,17,18,25,29,30]。

Laitenberger ら[21]はソフトウェア開発のインスペクションを構成する基本要素として、プロセス、役割、プロダクト、読解技術、の 4 つを定義している。

2.2 SRS の品質モデル

SRS の標準ガイドラインとして IEEE Std.830 [15]が広く知られている。IEEE Std.830 では SRS が備えるべき 8 つの品質特性が定義されている。一方、Davis ら[6,14]は SRS の一般的品質のフレームワークを提案している。しかし、これらの品質特性は、大部分が自然言語で記載される SRS をインスペクションするには具体性に欠けている。従来の SRS の品質特性に特定の背景や視点を加味した具体的、かつ実効的な品質特性が必要である。

Krogstie ら[20]は SRS に関する品質特性の 1 つとして PQ (Pragmatic Quality)を提案している。PQ とは特定の読者にとっての理解に関する品質と定義される。例えば、SRS の PQ とは、システム開発における特定のアクタ(例：ユーザ、設計者、テスター)にとっての理解に関する品質となる。ソフトウェア開発における SRS は、要求定義や開発(アーキテクトや設計者)のチームに加えて、顧客や保守・運用チーム等の多数の読者が存在する。そのため、PQ は実践的な SRS の品質モデルを検討する上で有益な概念である。

Fabbrini ら[9]は、Krogstie ら[20]の品質モデルを拡張し、シンタクティック、ストラクチャル、セマンティック、プラグマティックの 4 つの品質タイプを定義している。しかしながら、既存研究では PQ の概念的な定義や他品質との関係性の説明に留まっている。PQ の具体的な定義や評価方法の提示には至っていない[27,28]。

2.3 SRS の読解技術

SRS のインスペクションの特徴として、自然言語文書を読解する作業の割合が多いことが挙げられる[9,13]。この読解作業は多くの時間を要し、結果として読解の誤りも多く

なる。形式的な要求記述によるインスペクションも提案されているが[33]、依然としてインスペクションの効果は人(インスペクタ)が文書を読むという読解の作業に依存しており、読解には高度なスキルや知識が求められる[13]。しかし、高度なスキルを有するインスペクタは限られている。インスペクタの能力に依存しないインスペクションを可能とする実践的な SRS の読解技術が必要である。

インスペクションのための読解技術としてパースペクティブに基づく読解である PBR (Perspective-Based Reading)が挙げられる[31]。PBR では、特定のパースペクティブ(例：ユーザ、設計者、テストの立場)に基づき作成された質問セットを提供する。インスペクタは質問セットの観点から SRS を読解し、インスペクションを実行する。Shull ら[8]の PBR の研究では、異なるパースペクティブの視点から複眼的に SRS のインスペクションをおこなう。例えば、テストをパースペクティブとした場合、対象の SRS にはテストケースやテストデータを作成可能な情報が記載されているか?という観点からの質問セットが例示される。

しかし、既存研究[8]には PBR の肝要となる質問セットを作成する上での実践的な指針はない。質問セットの作成には読者(インスペクタ)の知識やスキルが必要になるため、PBR の実践に関する研究は限定的である。

2.4 SRS のインスペクションのインパクト

要求定義工程に要したコストと、プロジェクトの最終的な開発コストの関係性に関する調査[11]より、両者には負の相関があると報告されている。Damian ら[5]は商用システムの開発をする企業への調査で、要求定義工程が他工程に及ぼすインパクトの複雑性に関する研究を実施している。SRS のトレーサビリティ品質の低下が、プロジェクト全体の開発コスト増大に影響しているとの報告もある[26]。また、SRS のインスペクションの実施コストに対する費用対効果に関する研究もある [19,27]。しかしながら、上述の研究は SRS の品質の一側面と、後続工程への影響の関係を評価・考察しているに留まる。SRS の品質の包括的な評価と、そのプロジェクトへのインパクトに関する実証的な研究は十分とはいえない。

3. RISDM のフレームワーク

3.1 インスペクションのモデルとそのデザイン方法論

本研究でデザイン対象とするインスペクションは、Laitenberger ら[21]が提案するインスペクションの基本要素(プロセス、役割、プロダクト、読解技術)に、品質とレポートを加えた以下の 6 つの基本要素で構成するメタモデルで定義する(図 1 参照)。

- (1) 役割：インスペクションに関連する人や組織。
- (2) プロセス：インスペクションを実施する手順。
- (3) プロダクト：インスペクションの対象(例：SRS、設計書、ソースコード、テストケース)。

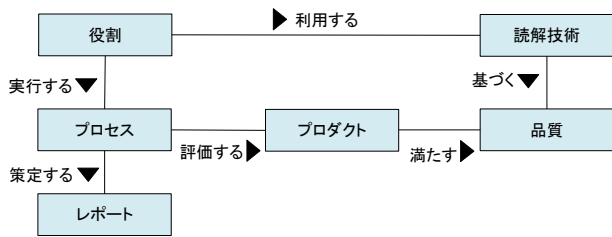


図 1 インスペクションメタモデル

Figure 1 Inspection Meta-Model

- (4) 品質：プロダクトが満たすべき品質特性。
- (5) 読解技術：プロダクトを読むための方法。
- (6) レポート：プロダクトの評価(インスペクション)結果。
 インスペクションのデザイン方法論とは、このメタモデルに基づき、その6つの基本要素を具体化することである。

3.2 RISDM による RIS のデザイン

提案する RISDM は、前述のメタモデルを入力として、プロダクトを SRS としたインスペクションをデザインする方法論である(図 2 参照)。なお、本稿では、前述のメタモデルの基本要素を具体化した SRS のインスペクションの活動を1つのシステムと捉え、要求インスペクションシステム(RIS: Requirements Inspection System)と呼ぶ。RIS は RIS デザイナが RISDM を用いてデザインする。

本稿では図 2 に示すように、プロジェクトの要求アナリストが作成した SRS を、プロジェクトから独立したインスペクタがインスペクションをおこなうアプローチを取り上げる[27,28]。即ち、本稿のデザイン対象の RIS は、前述の基本要素における役割(要求アナリスト、インスペクタ)とプロセス((1)SRS 受領, (2)インスペクション実行とリスク評価, (3)改善アドバイスの策定)は既に規定されている。そのため、本研究の RISDM はインスペクションの4つの基本要素(プロダクト、品質、読解技術、レポート)のデザイン方法論になる。

3.3 RISDM における RIS のデザインモデル

3.3.1 デザインモデルの構造

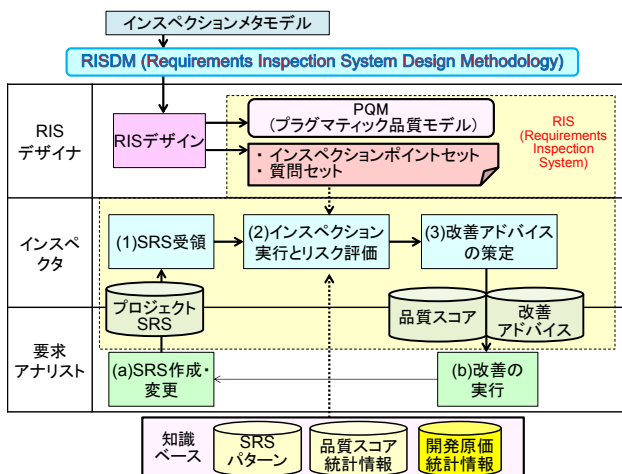


図 2 RISDM と要求インスペクションシステム(RIS)

Figure 2 RISDM and RIS for Third-Party Inspection

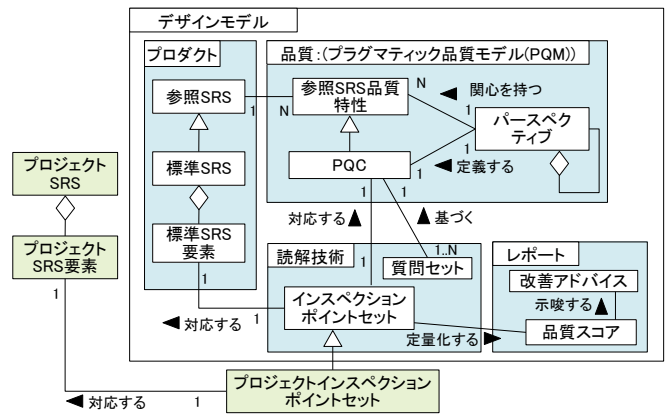


図 3 RIS のデザインモデルと関連要素

Figure 3 Design Model of RIS and Related Elements

図 3 に RISDM によるインスペクションのデザインモデルとその関連要素を示す。デザインモデルは RISDM が対象とするプロダクト、品質、読解技術、レポートとその構成要素の関係を表すメタモデルである。以降ではこれら4つの基本要素とその関連要素の詳細を説明する。

3.3.2 プロダクト

RISDM では SRS のインスペクションをデザインする上で、RIS を導入する企業・組織における SRS の標準的な目次構成(標準 SRS)を策定する。はじめに標準 SRS の参考となる SRS の構造(参照 SRS)を選定する。代表的な参照 SRS としては IEEE Std.830 の SRS の構造がある。RISDM ではこの参照 SRS の構造に基づき、標準 SRS を策定する。詳細は後述するデザインプロセスで示す。

図 3 のプロダクトの構成要素は以下になる。

- (1) 参照 SRS：基準となる SRS の目次構成。
- (2) 標準 SRS：企業・組織の標準的な SRS の構造。参照 SRS の構造に基づき策定される。
- (3) 標準 SRS 要素：標準 SRS を構成する要素。SRS の各目次項目が各要素に対応する。

図 3 の左部にはプロジェクトの要求アナリストが作成する SRS(プロジェクト SRS)とその構成要素(プロジェクト SRS 要素)が記されている。前述の標準 SRS と標準 SRS 要素の組合せは、RIS を導入する企業・組織においては1つのみ存在するものである。一方で、プロジェクト SRS とプロジェクト SRS 要素の組合せは、プロジェクトが複数存在すればその分だけ作成されるものである。

3.3.3 品質：PQM (Pragmatic Quality Characteristics)

RISDM では品質のモデルを PQM(Pragmatic Quality Characteristics)と呼ぶ。図 3 に記すように、PQM は以下に示す3つの構成要素の関係を表すモデルである。

- (1) パースペクティブ：SRS の特定の読者における SRS に対する立場(パースペクティブ)。例えば、読者が「ソフトウェア技術者」の場合、SRS に対するパースペクティブは「(SRS の内容を)実装する」になる。
- (2) 参照 SRS 品質特性：基準となる SRS の品質特性。代

表的な参照 SRS 品質特性としては IEEE Std.830 の 7 つの品質特性がある。

- (3) PQC(プラグマティック品質特性)：読者のパースペクティブの観点における、SRS の理解し易さの程度に関する品質。参照 SRS 品質特性に基づき定義される。

上述のように RISDM では、パースペクティブの観点から、参照 SRS 品質特性を具体化し、PQC の定義をおこなう。詳細は後述するデザインプロセスで示す。

3.3.4 読解技術

RISDM では PBR のアプローチに基づき SRS を読むための方法を定義する。これによりインスペクタの SRS への適所・適切なインスペクションの実施を支援する。読解技術の構成要素は以下になる。

- (1) インスペクションポイントセット:PQC とそれに基づき評価すべき標準SRSの箇所(目次項目)との対応付けを示したもの。標準 SRS の各目次項目でどの PQC の評価(インスペクション)を実施すべきかを示す。
- (2) 質問セット:インスペクションポイントセットで示された箇所(目次項目)での評価観点を示したもの。内容は PQC に基づいた質問形式で記述される。

インスペクションポイントセットは標準 SRS に対するインスペクションの実施箇所を示したものである。そのため前項と同様に、RIS を導入する企業・組織においては 1 つのみ存在するものである。

もし、プロジェクトが作成した SRS(プロジェクト SRS)の構成が、標準 SRS の構成とは異なる場合には、インスペクションポイントセットは、プロジェクト SRS の構成に応じた修正が必要となる。図 3 の下部に記したプロジェクトインスペクションポイントセットは、修正後の実施箇所を記したものになる。プロダクトと同様に、プロジェクトインスペクションポイントセットは、プロジェクトが複数存在すればその分だけ作成される。

3.3.5 レポート

RISDM では、前項の読解技術に基づく SRS の評価結果の内容をレポートとして定義する。レポートは以下の 2 つの構成要素を含む。

- (1) 品質スコア:インスペクションポイントにおける評価結果を得点化し、SRS の品質を数値として表したものの。
- (2) 改善アドバイス:品質スコアの内容を受けて、SRS の改善に有用な情報(修正が必要な SRS の箇所、参考となる成果物サンプル)を記したものの。

レポートはインスペクション対象のプロジェクト SRS に対して 1 つ作成される。

3.4 知識ベース

図 2 の下部に記す知識ベースはインスペクションの基本要素ではないが、RIS の実行や SRS の改善に活用される。インスペクタが SRS のインスペクションやリスク評価の実行の際に参照する統計情報と、要求アナリストが SRS を

改善する際に参考となる SRS の成果物サンプル(SRS パターン)が、開発ライフサイクルを通じて蓄積される。

統計情報は、過去にインスペクションを実施した SRS の品質スコアの情報(品質スコア統計情報)と、当該 SRS に対応するプロジェクトの後続(ソフトウェア開発)工程におけるコストの計画・実績の情報(開発原価統計情報)の 2 つからなる。前者は SRS の品質評価時のベンチマークに、後者は開発工程でのコスト予実乖離のリスク予測に活用する。

4. RISDM による RIS のデザイン方法

図 4 に RISDM の前提、およびデザインプロセスの内容を記す。前述したように本稿では役割とプロセスはデザイン対象外(前提)となる。以降では前提の内容、および RISDM におけるプロダクト、品質、読解技術、レポートのデザイン方法を記す。

4.1 前提

本稿で取り上げるインスペクション方法は、プロジェクトの要求アナリストが作成した SRS(プロジェクト SRS)を、プロジェクトから独立したインスペクタがインスペクションを実行するものである[27,28]。図 4 に示すインスペクションプロセス(3 つのプロセス)の内容を以下に記す。

- (1) SRS の受領：プロジェクトの要求アナリストが作成した SRS を、インスペクタが受領する。
- (2) インスペクション実行とリスク評価：インスペクタが SRS のインスペクションを実行し、SRS の品質スコアと SRS パターンを含む改善アドバイスを記したインスペクションレポートを作成する。併せて、後続(ソフトウェア開発)工程のリスク評価もおこなう。この際、前述の知識ベースも参照される。
- (3) 改善アドバイスの策定：インスペクション結果に基づき、インスペクタは要求アナリストに対して SRS の品質改善のためのアドバイ

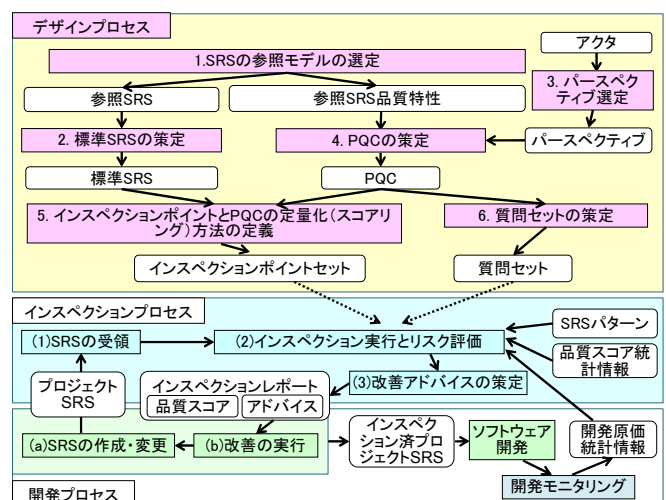


図 4 RIS のデザインプロセスと関連プロセス
 Figure 4 Design Process for RIS and Related Process

スの策定をおこなう。

インスペクションプロセスは(プロジェクト SRS の受領ごとに)繰り返し実行される。要求アナリストは上述のインスペクションプロセスの結果を受けて SRS の品質改善を実行する。必要に応じて改善後 SRS のインスペクションも再度おこなわれる。

SRS が改善された後、後続のソフトウェア開発が実行される。RIS では SRS のインスペクション後も継続してプロジェクトの開発モニタリングをおこなう。開発工程の原価の統計情報を蓄積し、SRS のリスク評価に活用する。

4.2 RISDM による RIS のデザインプロセス

4.2.1 デザインプロセスの構造

図 4 の上部に RISDM によるインスペクションのデザインプロセスを示す。6 つのプロセスと各プロセスの入出力の成果物を記している。

デザインプロセスでは、SRS の品質のモデル(PQM)と、それに基づく SRS を読むための方法 (インスペクションポイントセットと質問セット)が定義される。なお、デザインプロセスは RIS のデザイン時に 1 回のみ実行される。その上で、定義された PQM や SRS を読むための方法を参照し

表 1 標準 SRS の目次項目

Table 1 Table of Contents of Standard SRS

章	節
1. 本要求定義書について	1.1. 要求定義書の目的
	1.2. 要求定義書の想定読者
	1.3. 要求定義書の構成
	1.4. 参考文献
2. システム開発概要	2.1. システム化の目的
	2.2. 業務概要とシステム化の範囲
	2.3. 制約事項
	2.4. 用語定義
3. 要求に変更を与える事項・未確定事項	3.1. 要求に変更を与える事項
	3.2. 未確定事項
4. 機能要求	4.1. 業務フロー定義
	4.2. 機能定義
	4.3. データモデル定義
5. 非機能要求	5.1. 非機能要求グレード表
	5.2. システムアーキテクチャ要求
	5.3. 移行要求
	5.4. サービス提供要求

表 2 パースペクティブとプラグマティック品質特性

Table 2 Perspectives and PQC

パースペクティブ		参照 SRS 品質特性							PQC (プラグマティック品質特性)		
レベル 1	レベル 2	正確性	完全性	の ランク付 重要度と安定度	非曖昧性	検証可能性	追跡可能性	変更可能性	ID	名称	定義
要求する	ビジネス・システム要求を顧客ニーズに対応させる	X							C1	合目的性	SRS に記述される全ての要求は 1 つ以上のプロジェクト目的に対応付けがされている。
管理する	全ての成果物の進捗状況を管理する		X	X					C2	記述項目網羅性	SRS に記述すべき内容が全て存在する。
実装する	テンプレートに基づき設計する				X	X			C3	テンプレート使用	テンプレートを使用している。
	標準記法に基づいて設計する				X	X			C4	標準記法使用	標準(記述)記法を使用している。
	標準用語を利用する				X				C5	用語定義	用語とその定義が作成されている。
アーキテクチャに割り当てる	成果物の全体構成を把握する							X	C6	識別子の付与	成果物の一覧化と識別子が付与されている。
	成果物間の関連を特定する						X	X	C7	一意識別性	全ての成果物は一意に識別できる。

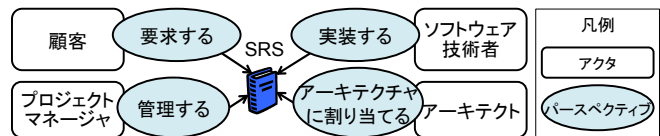


図 5 アクタと導出されたパースペクティブ

Figure 5 Actors and Perspectives

て、前述のインスペクションプロセスが実行される。

4.2.2 デザインプロセスの手順

RISDM によるインスペクションのデザインプロセス(6 つのプロセス)の詳細を、次章で述べる企業に RISDM を実践した際のケースを用いて示していく。

(1) SRS の参照モデルの選定

SRS の参照モデルとは、インスペクションをデザインする上で基準となる SRS の構造(参照 SRS)と品質特性(参照 SRS 品質特性)である。本ケースでは IEEE Std.830 で定義される構造と品質特性の内容を SRS の参照モデルとした。なお、参照モデルとしては IEEE Std.830 以外のものや、更に構造と品質特性でそれぞれ異なるものを選定することも可能である。例えば、構造は IEEE Std.830 を、品質特性は ISO/IEC 9126 や ISO/IEC 25010 を参照モデルとして選定することも RISDM では可能としている。

(2) 標準 SRS の策定

選定された参照 SRS に基づき、RIS を導入する企業・組織における標準的な SRS の構造(標準 SRS)を策定する。本ケースでは IEEE Std.830 の構造を踏まえ、表 1 に示すような標準 SRS を策定した。標準 SRS は章・節・項の 3 階層の目次項目で定義される(表 1 は節レベルまで記載)。標準 SRS は上述の目次項目と、関連する成果物テンプレートも含まれる。例えば、表 1 の「4.2 機能定義」には機能定義書や画面定義書のテンプレートを含む。

(3) パースペクティブの選定

SRS に対するパースペクティブの選定は 2 段階で実施する。初めに、SRS の読者(アクタ)定義する。次に、アクタの SRS に対するパースペクティブを選定する。必要に応じ

表3 インспекションポイントセット

Table 3 Inspection Point Set

PQC	ID	名称	標準 SRS の目次			
			2.1 システム化の 目的	2.2 業務概要とシ ステム化の範 囲	2.3 制約事項	2.4 用語定義
C1		合目的性	X			
C2		記述項目網羅性	X	X	X	X
C3		テンプレート使用	X	X	X	
C4		標準記法使用		X		
C5		用語定義				X
C6		識別子の付与	X	X	X	
C7		一意識別性	X	X	X	X

てパースペクティブの内容は詳細化や分解がされる。図 5 に本ケースで定義された4つのアクタ(顧客, プロジェクトマネージャ, ソフトウェア技術者, アーキテクト)と, 各アクタの SRS に対するパースペクティブ(レベル 1)を示す。本ケースではこれら4つのパースペクティブの詳細化と分解を実施し, 合計7つのパースペクティブ(レベル 2)を定めた。表 2 のパースペクティブの2つの列に, レベル 1 とレベル 2 のパースペクティブの内容をそれぞれ示す。

(4) PQC の策定

プラグマティック品質特性(PQC)の策定は, 前プロセスで選定されたパースペクティブ単位でおこなう。

表 2 には, 本ケースのパースペクティブから PQC の定義までの過程を記している。1 行目を例にとると, アクタ「顧客」から導出したパースペクティブ(レベル 1)は「要求する」が記された。この内容を詳細化し, パースペクティブ(レベル 2)は「ビジネス・システム要求を顧客ニーズに対応させる」となった。次にこのパースペクティブの内容に基づき, SRS に求められる参照 SRS 品質特性を選択した。本例のパースペクティブの内容から, SRS の要求が顧客ニーズを「正しく」反映しているという意味である「正確性」を選択した(正確性の列に“X”を表記)。最後に, 選択した参照 SRS 品質特性の内容を, 読者(アクタ)の理解のし易さという観点から具体化し PQC を定義する。本例では, 正確性の内容(SRS の要求が顧客ニーズを「正しく」反映している)を顧客(アクタ)に理解し易くするためには, どのような特性を SRS が備えるべきかを検討した。その結果として, 「SRS に記述される全ての要求は1つ以上のプロジェクト目的に対応付けがされている」という特性を導出し, これ

を PQC 「C1: 合目的性」として定義した。同様にして, 残り6つのパースペクティブのそれぞれに対応する PQC を定義していった。

(5) インспекションポイントと PQC の定量化(スコアリング)方法の定義

定義した全ての PQC に関する評価(インспекション)を, SRS の全ての構成要素に実施することは必ずしも効果的とはいえない。PQC に基づくインспекションを SRS の適切な箇所で行うため, 標準 SRS の目次(構成要素)と PQC を対応付けたインспекションポイントセットを作成する(表 3 参照)。表中の各升目は, 少なくとも1つ以上のインспекションを実施すべき場合は“X”と表記される。例えば「2.1 システム化の目的」の要素(節)では, C4 と C5 を除く5つの PQC に基づく評価(インспекション)を実施することを意味する。本表を参照することでインспекタは SRS のどの箇所に何の PQC に基づく評価(インспекション)を実施すべきかを把握することができる。本ケースでは SRS のインспекションポイントが合計 196 個となった。

併せて, インспекタがレポートを作成するために, PQC の評価結果の定量化方法の検討もおこなう。本ケースでは, 各インспекションポイントで不備がなければ(加点(+1), 不備が発見されれば加点しない(0 点))というスコアリング方法を採用した。その上で, スコアリングの合計値をインспекションポイント総数(196)で除算し, PQC の評価結果を 100 点満点で数値化した SRS の「品質スコア」を定義した。即ち, 品質スコアは SRS に不備が検出されなかった割合を示す。

(6) 質問セットの策定

インспекションポイントで確認すべき内容を, 定義した PQC の内容を満たしているか否かの問い掛け(質問)の形式で策定する。本ケースで作成した各 PQC に対応した7つの質問(質問セット)を表 4 に示す。各行に PQC を評価するための質問が記載されている。1 行目には「C1: 合目的性」を評価するために「プロジェクト SRS のビジネス・システム要求はプロジェクトの目的に対応付けて記載されているか?」という質問が記されている。

前項のインспекションポイントセットと本項の質問セットを合わせた2つのセットを, RISMD では読解技術と呼ぶ。インспекタはこれら2つのセットを参照すること

表4 質問セット

Table 4 Question Set

PQC		質問セット	インспекション ポイント数
ID	名称		
C1	合目的性	プロジェクト SRS のビジネス・システム要求はプロジェクトの目的に対応付けて記載されているか?	3
C2	記述項目網羅性	プロジェクト SRS の要素は標準 SRS に対応しているか?	54
C3	テンプレート使用	プロジェクト SRS の成果物は標準 SRS のなかで定められているテンプレートを用いて記載されているか?	36
C4	標準記法使用	プロジェクト SRS の成果物は標準 SRS のなかで定められている標準(記述)記法で記載されているか?	6
C5	用語定義	プロジェクト SRS の用語集は作成されているか?	3
C6	識別子の付与	プロジェクト SRS の成果物や特定の要素は識別子が付与されて表で一覧化されているか?	48
C7	一意識別性	プロジェクト SRS の成果物や特定の要素は識別子を用いて一意に特定できるか?	46
合計			196

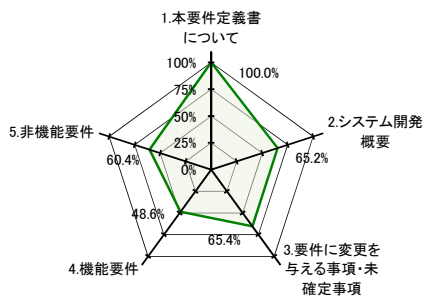


図6 SRSの目次(章)単位の品質スコアの可視化
 Figure 6 Visualization on PQC based on TOC of SRS

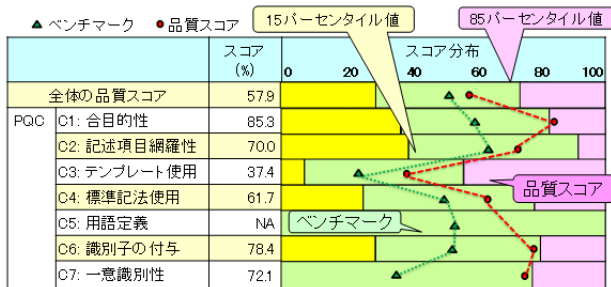


図7 品質(PQC)単位のベンチマーク診断図
 Figure 7 Benchmark Diagnostic Diagram

で, SRS のどの箇所を何の PQC に基づき評価すべきかを正確に把握でき, 更に質問に対する回答結果(Yes or No)から, 不備の有無(+1 or 0)を客観的に判断することが可能となる. 例えば, 標準 SRS には表 1 に記したレベル(章・節)の更なる下のレベル(項)まで定義されており, 合計 54 項が存在する. このことから C2(記述項目網羅性)には合計 54 個のインスペクションポイントが設定されている. もし SRS に全ての目次項目が記されていれば 54 点となり, 記述項目網羅性は満点 100%となる.

図 6 は読解技術を SRS に適用し, SRS の品質を定量化した結果を可視化した例である. SRS の各章におけるインスペクションポイントの合計値を 100%として, それに対する品質スコアの比率をレーダーチャートで表している. これにより, SRS の各章の相対的な品質が評価でき, SRS の記述を充実化すべき箇所が把握できる. 例えば図の例では「4. 機能要求」の品質スコアが 48.6%と相対的に低いこと

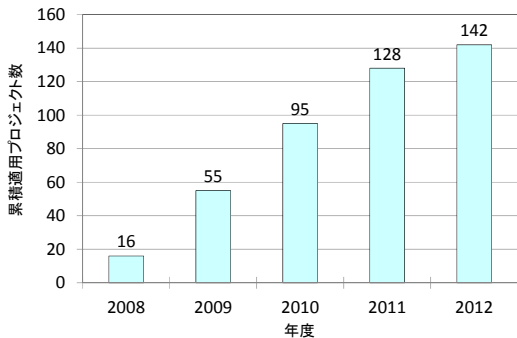


図8 累積適用プロジェクト数
 Figure 8 Accumulated Number of Project Inspected

がわかる.

図 7 は SRS の品質スコアと過去の全プロジェクトの SRS の品質スコアの平均値(ベンチマーク)との相対評価を可視化したベンチマーク診断図である. ベンチマークは, 3 章で述べた, RIS の知識ベースの品質スコア統計情報より算出される. 図に示すように, 全プロジェクトにおける対象 SRS の 15 パーセンタイル値と 85 パーセンタイル値も示すことで, SRS の相対的な品質レベルを把握できる.

5. 実践

RSIDM 適用によりデザインされた SRS のインスペクションは 2008 年から試行を開始し, 2010 年から本格展開をしている. 適用プロジェクトは年々増加し, 2012 年度末には累計で 140 件を超えている(図 8 参照). 適用プロジェクトの約 25%は開発規模が 500 人月を超えており, 大規模プロジェクトにも数多く適用されている[23,27,28].

6. 評価

前章の実践適用から得られた統計情報に基づく評価により, 1 章で記した 2 つの仮説の検証をおこなう. 仮説検証を通じて RISDM の有効性を示す.

6.1 評価 1: PQC によるリスク予測

6.1.1 目的

SRS の品質スコアが, プロジェクトの後続(ソフトウェア開発)工程のリスク予測に有効なことを示す. 本評価では, リスクを開発コストの計画値に対する実績値の乖離と捉える. PQC の品質スコアが, 開発コストの予実乖離を予測する評価指標と成り得るかを評価する.

6.1.2 方法

表 5 に開発コストの計画・実績の情報(開発原価統計情報)を収集した 7 プロジェクトのプロファイル(業種・開発種別・SRS ページ数)を記す.

ここで, 開発コストの予実乖離の度合いを示す指標として, コスト予実乖離率を以下の式で定義する.

$$\text{コスト予実乖離率} = \frac{|\text{実績値} - \text{計画値}| [\text{金額}]}{|\text{計画値}| [\text{金額}]}$$

計画値は, プロジェクトが SRS を策定した後に後続のソ

表 5 プロジェクト一覧

Table 5 Project Profile for Analyzing Cost Overrun

ID	業種	開発種別	SRS ページ数
1	金融	機能追加&更改	545
2	公共	新規	401
3	製造	新規	666
4	公共	新規	172
5	金融	機能追加&更改	145
6	流通	新規	300
7	サービス	新規	71

表 6 相関係数と P 値

Table 6 Correlation Coefficient and P value

PQC	項目	相関係数	P 値
C1	合目的性	0.516	0.236
C2	記述項目網羅性	-0.793	0.033
C3	テンプレート使用	-0.226	0.626
C4	標準記法使用	-0.219	0.638
C5	用語定義	-0.013	0.978
C6	識別子付与	-0.408	0.364
C7	一意識別性	-0.794	0.033

ソフトウェア開発工程に対して見積もりをした金額である。実績値は、顧客へのデリバリ後に確定した開発コストの金額である。

その上で本評価では、上述のコスト予実乖離率と、品質スコア統計情報から収集した7つのPQCの品質スコアとの相関分析を実施した。

6.1.3 結果(仮説の検証)

表 6 に相関分析の結果として、コスト予実乖離率と 7 つの PQC の品質スコアにおける相関係数と P 値を記す。C2 と C7 の相関係数の値は-0.5 以下となり負の相関があることを示している。即ち、C2 と C7 の品質スコアが低い場合、プロジェクトの開発コストの予実乖離が大きくなるといえる。特に両者の P 値(<0.05)より、C2 と C7 の品質スコアとコスト予実乖離率の間には 5 %の有意水準での相関が確認できた。

7プロジェクトの C2 と C7 の品質スコアの値と、コスト予実乖離率の分布を図9に表す。横軸(コスト乖離率)は25%を境に分割している[3]。C2 と C7 の回帰直線はほぼ等しい形状であるため、2つの回帰直線と25%の交点では、C2 と C7 の品質スコアはそれぞれ 59.5%と 63.0%とほぼ同じ値(≒60.0%)となる。そこで縦軸(品質スコア)を 60.0%の値で分割すると、図9は4つの象限に分割できる。本図で示すように、プロジェクトは左上と右下の2つの象限にのみ分布していることが分かる。左上の象限には、品質スコアが高く、且つコスト乖離率が小さいプロジェクトが分布している。一方で右下の象限には、品質スコアが低く、且つコスト

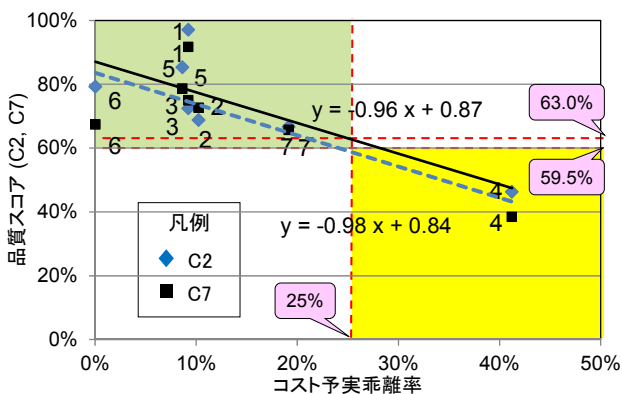


図9 品質スコア (C2, C7) とコスト予実乖離率
 Figure 9 PQC Score (C2, C7) and CDR

表 7 プロジェクト一覧

Table 7 Project Profile for Analyzing PQC Improvement

ID	業種	開発種別	SRS ページ数
A	金融	新規	191
B	製造	新規	289
C	金融	機能追加	550
D	サービス	更改	343
E	金融	更改	415
F	金融	新規	267

ト乖離率が大きいプロジェクトが分布している。

以上より、SRS の PQC における記述項目網羅性(C2)と一意識別性(C7)の品質スコアが低い(60.0%を下回る)と、開発コストの予実乖離のリスクが高くなるということが統計的に明らかになった。これにより仮説 1)の検証を達成した。

6.2 評価 2 : SRS 品質の納期遅延への影響評価

6.2.1 目的

RISDM でデザインした読解技術の適用で指摘されたレポートの内容(改善アドバイス)が、プロジェクトの要求アナリストがおこなう SRS の改善に有効なことを示す。

6.2.2 方法

本評価では表 7 に記す 6 プロジェクトを対象に、インスペクションとフィードバック実施(品質スコアと改善アドバイスの提供)後の、要求アナリストによる SRS の改善状況のモニタリングをおこなった。

6.2.3 結果(仮説の検証)

図 10 は、モニタリングした 6 プロジェクトにおける、PQC 全体(C1~C7)の品質スコア(実線グラフ)と、前節の評価でリスク評価に有効な指標とされた C2 と C7 の品質スコアの平均値(点線グラフ)の時系列な推移を示している。プロジェクト A は SRS の改善を 2 回実施しており、他のプロジェクトは 1 回実施している。

本図より、4 プロジェクト(A, B, E, F)は PQC 全体の品質スコアが増加しており、SRS の改善が実施されたことが分かる。一方で、他の 2 プロジェクト(C, D)は変化がない。しかし、2 プロジェクトの C2 と C7 の品質スコアの平均値(C27 と D27 の点線グラフ)を見ると、どちらも SRS の初版の段階で 60 %を超えている。これは前節の評価によれば、C2 と C7 の品質スコアが 60%を下回っていないため、開発コストの予実乖離のリスクが低いことを意味する。そのため、要求アナリストはこれ以上の SRS の改善は必要ないと判断したとも考えられる。前述の 4 プロジェクトも、改善実施後は C2 と C7 の品質スコアの平均値が 60%を超えている。これら 4 プロジェクトも同様な判断をした結果、改善を完了したとも考えられる。

図 11 は、プロジェクト A における SRS の PQC 単位での改善推移をモニタリングした結果である。前述したようにプロジェクト A では 2 回の改善を実施している。最初のインスペクションの実施後の改善(1 回目)では、改善アドバイ

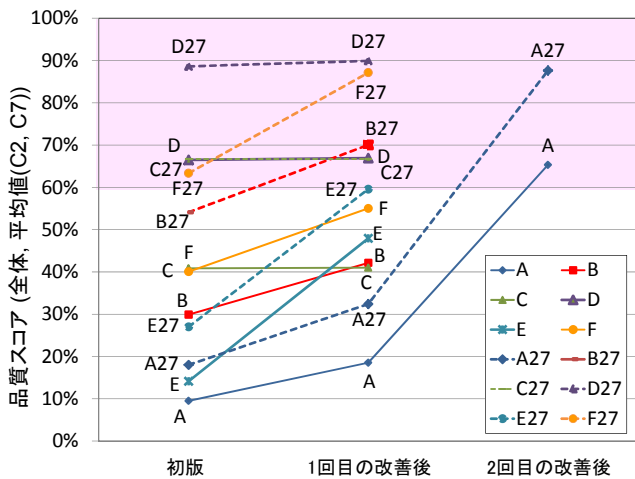


図 10 品質スコアの段階的改善(全プロジェクト)
 Figure 10 Incremental Improvements of OS

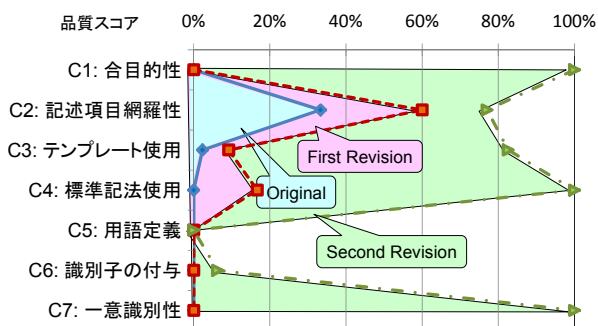


図 11 品質スコアの段階的改善の内訳(プロジェクト A)
 Figure 11 Incremental Improvements of QS at Project A

スに基づき、C2の品質が大きく改善されている。その上で再度のインスペクション適用後の2回目では、C1とC2とC7の品質が改善されている。1回目の改善ではSRSの記述項目が、2回目の改善ではSRSの記述内容が充実したと解釈できる。

以上より、インスペクタが作成したレポートにおいて指摘された内容(改善アドバイス)は、プロジェクトの要求アナリストのSRS改善のアクションにつながっていることが観察された。これにより仮説2)の検証を達成した。

7. 考察

7.1 PBRとプラグマティック品質特性の統合

RISDMではインスペクションにおける主要な読解技術であるPBRに対して、プラグマティック品質特性(PQC)の観点を統合してSRSのインスペクションをデザインした。SRSのPQCに基づき作成されたインスペクションポイントセットと質問セットを参照することで、たとえプロジェクトのドメイン知識が乏しい第3者(インスペクタ)でもSRSのインスペクションが実行できるようになる。これにより、大規模なソフトウェア開発プロジェクトのSRSに対しても、インスペクタによる組織的なインスペクションの実践が可

能になった。従来、プラグマティック品質特性とPBRはそれぞれ独立に研究されていたが、RISDMでは両者を適切に統合することで、実践的且つ有用的なインスペクションのデザインが可能であることを示した。

7.2 SRS品質の後工程へのインパクトの予測

本稿の評価結果より、SRSに対するPQCの品質スコアの値から、後続の開発工程の開発コストの予実乖離が予測可能であることが実証できた。本稿の評価結果より、記述項目網羅性(C2)と一意識別性(C7)の品質が低いSRSのプロジェクトでは、後続の開発工程での開発コストが計画と乖離するリスクが高くなることが分かった。一方で、他の品質特性は、予実乖離のリスクへの影響が少ないことも明らかになった。これまでも企業のソフトウェア開発プロジェクトにおける成否の重要要因として、SRSの品質特性は指摘されていた[5]。しかしながら、企業の実プロジェクトの開発コスト統計情報に基づく、SRSの品質スコアを活用した後工程へのインパクト(開発コストの予実乖離)の予測に関する実証的な研究はされていない。本研究の評価結果は重要な成果であるといえる。

7.3 RISDMの適用性

開発工程の成果物(例:ソースコード、テストケース)へのインスペクションと比較して、SRSに対するインスペクションでは、1章でも述べたように背景情報や記述スタイルが多様なことから組織的な実践が非常に難しいとされてきた[21,29]。ほとんどのプロジェクトでは有識者の知識・スキルに依存したインスペクションに留まり、プロジェクト共通に適用可能なSRSのインスペクションの定義は困難であった。本稿で紹介したRISDMの実践では、1企業の取り組みではあるが、定義したPQCとそれに基づく第3者によるインスペクションが、規模や業種、開発種別等が異なる140超のプロジェクトに適用された実績がある。このことから、RISDMは様々なプロジェクトのSRSに適用できる方法論であるといえる。

7.4 SRSのインスペクションの方法論工学

RISDMはインスペクションの方法論ではなく、インスペクションをデザインするための方法論である。Method Engineering(方法論工学)の領域では[32]、方法論の構築手法に関する研究がなされているが、設計や実装に関する方法論が議論の中心であり、要求定義におけるSRSのインスペクションの方法論に関する議論はほとんどされていない。更に、本稿で議論したような実践的なデザインのモデルやプロセス、および適用評価に関する研究報告は存在しない。また、方法論工学では、開発環境や利用状況に合わせた既存方法論の再利用・再構築も重要な研究テーマである[32]。本研究のRISDMのデザインモデルやデザインプロセスはソフトウェア開発の業種(ドメイン)、開発種別やスケール(開発規模)には依存しない方法論である。そのため、本稿の実践で紹介したエンタープライズ系ソフトウェア開発に留

まらず、組込みソフト開発やパッケージソフト開発の SRS への適用も可能な再利用・再構成が可能な方法論であるといえる。

8. まとめ

本稿ではソフトウェア要求仕様書(SRS)の品質向上を目指し、SRS のインスペクションのデザイン方法論の RISDM を提案した。SRS の第 3 者インスペクションのアプローチを取り上げ、提案方法論を適用して SRS のプラグマティック品質モデル(PQM)と PBR(Perspective-Based Reading)に基づく SRS の読解技法をデザインした。その上で、企業内の実プロジェクトへの RISDM の実践適用と評価をおこなった。結果として、RISDM でデザインした SRS の PQ や読解技法は、プロジェクトのリスク予測や SRS の品質改善に有用であることが検証できた。

第 3 者インスペクションの実践と評価の取り組みは今後も継続していく予定である。インスペクションの実施内容、および要求アナリストへの改善アドバイスの精度向上を進めていく。さらに、SRS の品質と開発コストの統計情報の蓄積も継続的に進めて、統計情報に基づく多角的なプロジェクトのリスク評価・予測にも取り組む予定である。

謝辞 本研究にご支援頂いている木谷強氏、端山毅氏、小橋哲郎氏、平岡正寿氏、藤貫美佐氏、並川顕氏、大杉直樹氏、山本英之氏、ならびに、関係各位に感謝致します。

参考文献

- 1) I. F. Alexander and R. Stevens, Writing Better Requirements, Addison-Wesley, 2002.
- 2) B. W. Boehm, "Verifying and Validating Software Requirements and Specifications," IEEE Software, Vol. 1, No. 1, Jan./Feb. 1984, pp. 75-88.
- 3) B. W. Boehm, et al., Software Cost Estimation with COCOMO II, Prentice Hall, 2000.
- 4) B. H. C. Cheng and J. M. Atlee, "Research Directions in Requirements Engineering," FOSE '07, IEEE Computer Society, May 2007, pp. 285-303.
- 5) D. Damian and J. Chisan, "An Empirical Study of the Complex Relationships between Requirements Engineering Processes and Other Processes that Lead to Payoffs in Productivity, Quality, and Risk Management," IEEE Trans. Software Engineering, Vol. 32, No. 7, Jul. 2006, pp. 433-453.
- 6) A. Davis, et al., "Identifying and Measuring Quality in a Software Requirements Specification," Proc. of the First Int'l Software Metrics Symposium, IEEE Computer Society, May 1993, pp. 141-152.
- 7) C. Denger and T. Olsson, "Quality Assurance in Requirements Engineering," in Engineering and Managing Software Requirements, A. Aurum and C. Wohlin, Eds. Springer, 2005, pp. 163-185.
- 8) C. Denger and F. Shull, "A Practical Approach for Quality-Driven Inspections," IEEE Software, Vol. 24, No.2, Mar./Apr. 2007, pp. 79-86.
- 9) F. Fabbrini, et al., "Achieving Quality in Natural Language Requirements," Proc. of 11th Int'l Software Quality Week Conference (QW'98), May 1998, 17 pages.
- 10) M. E. Fagan, "Advances in Software Inspections," IEEE Trans. Software Engineering, Vol. SE-12, No. 7, Jul. 1986, pp. 744-751.
- 11) K. Fosberg and H. Mooz, "System Engineering Overview," R. Thayer and M. Dorfman (eds.), Software Requirements Engineering, IEEE, 2000, pp. 44-72.
- 12) G. Fanmuy, et al., "Requirements Verification in the Industry," Proc. of CSDM 2011, Springer, 2012, pp. 145-160.
- 13) G. Génova, et al., "A Framework to Measure and Improve the Quality of Textual Requirements," Requirements Engineering J., Vol. 18, No. 1, Mar. 2013, pp. 25-41.
- 14) T. Gorschek and A. M. Davis, "Requirements Engineering: In Search of the Dependent Variables, Information and Software Technology, Vol. 50, No.1-2, Jan. 2008, pp. 67-75.
- 15) IEEE Std. 830-1998, Recommended Practice for Software Requirements Specifications, IEEE, 1998.
- 16) IEEE Std. 1028-1997, Standard for Software Reviews, IEEE, 1997.
- 17) JISA REBOK 企画 WG(編), 要求工学知識体系 (REBOK), 第 1 版, 近代科学社, 2011.
- 18) E. Kamsties, et al., "Detecting Ambiguities in Requirements Documents Using Inspections," Proc. of WISE'01, 2001, pp. 68-80.
- 19) E. Knauss, et a., "Investigating the Impact of Software Requirements Specification Quality on Project Success," Proc. of PROFES '09 LNBI Vol. 32, Jun. 2009, pp. 28-42.
- 20) J. Krogstie, et al., "Towards a Deeper Understanding of Quality in Requirements Engineering," Proc. of CAiSE '95, LNCS Vol. 932, Springer, 1995, pp. 82-95.
- 21) O. Laitenberger and J.-M. DeBaud, "An Encompassing Life-Cycle Centric Survey of Software Inspection," J. of Systems and Software, Vol. 50, No. 1, Jan. 2000, pp. 5-31.
- 22) C. J. Neill and P. A. Laplante, "Requirements Engineering: The State of the Practice," IEEE Software, Vol. 20, No. 6, Nov./Dec. 2003, pp. 40-45.
- 23) NTT DATA, <http://www.nttdata.com/global/en/>.
- 24) K. Pohl, Requirements Engineering, Springer, 2010.
- 25) A. Porter and L. G. Votta, "An Experiment to Assess Different Defect Detection Methods for Software Requirements Inspections," Proc. of ICSE 1994, IEEE Computer Society, pp. 103-112.
- 26) S. Saito, et al., "Empirical Analysis of the Impact of Requirements Traceability Quality to the Productivity of Enterprise Applications Development," Proc. of APSEC 2012, IEEE Computer Society, Dec. 2012, pp. 330-333.
- 27) S. Saito, et al., "Requirements Clinic: Third Party Inspection Methodology and Practice for Improving the Quality of Software Requirements Specifications," Proc. of RE 2013, IEEE Computer Society, Jul. 2013, pp. 290-295
- 28) 斎藤 忍 他, ソフトウェア要求仕様書の第 3 者インスペクション方法論とその実践評価, SES2013 論文集, Sep. 2013, pp. 1-8.
- 29) F. Salger, et al., "An Integrated Quality Assurance Framework for Specifying Business Information Systems," Proc. of CAiSE Forum 2009, Jun. 2009, pp. 25-30.
- 30) F. Salger, Requirements Reviews Revisited: Residual Challenges and Open Research Questions, Proc. of RE 2013, IEEE Computer Society, Jul. 2013, pp. 250-255.
- 31) F. Shull, et al., "How Perspective-Based Reading Can Improve Requirements Inspections," IEEE Computer, Vol. 33, No. 7, Jul. 2000, pp. 73-79.
- 32) K. Slooten and S. Brinkkemper, A method engineering approach to information systems development, In Information System Development Process, pp.167-186, North-Holland, 1993.
- 33) I. Sommerville and P. Sawyer, Requirements Engineering: A Good Practice Guide, Wiley, 1997.