

記述漏れと曖昧な表記の防止を目的とした
要件定義書の第三者品質スコアリングに向けた試み

**Toward A Scoring Method of Document Quality for
Prevention of Incomplete or Ambiguous Requirements Specification**

株式会社 NTT データ 技術開発本部
NTT DATA Corporation Research and Development Headquarters

○大杉 直樹	並川 顕 ¹⁾	小橋 哲郎 ²⁾
重木 昭信 ³⁾	木谷 強 ⁴⁾	山本 修一郎 ⁵⁾
○Naoki Ohsugi	Akira Namikawa ¹⁾	Tetsuo Kobashi ²⁾
Akinobu Shigeki ³⁾	Tsuyoshi Kitani ⁴⁾	Shuichiro Yamamoto ⁵⁾

Abstract This paper proposes a scoring method of document quality for prevention of incomplete or ambiguous System (or Software) Requirements Specification (SRS). The proposed method quantifies degree of completeness and unambiguity of a SRS document as a score on a 100-point scale. Smaller score indicates incomplete and/or ambiguous SRS so that makes the development stakeholders aware of the project's risky condition. It provides an opportunity to discuss more clear requirements and prevent inaccurate estimation. In addition, this paper explains a case study for verifying usefulness of the proposed method and reliability of the resulted scores. In the case study, the project managers' comments are explained as well as the scoring results of the 10 system developments.

1. はじめに

要件定義書はシステム開発における重要な成果物である。顧客とベンダは、開発するシステムの仕様（システムを導入する業務、求められる機能や性能、既存システムや連携システムとのインタフェース、保守や運用におけるサービスレベルや作業内容など）を検討し、合意した内容を要件定義書に記述する。ベンダは、記述内容に基づいて実現方法を検討し、プロジェクト計画を作成する。要員やコストを見積もって開発体制を整備し、詳細設計など具体的な開発を開始する[1]。

要件定義書に記述漏れや曖昧な記述があると、顧客の理解とベンダの理解に大きな差異が生じ、見積り誤りや後工程での再作業・追作業が発生する[2]。ベンダが、顧客が望むシステムの仕様の一部を見落としのまま、もしくは内容を誤解したまま作業を進めてしまい、見込んでいなかった機能を作りこむための追作業、誤って作りこまれた機能やアーキテクチャを修正するための再作業が必要となる。開発スケジュールの進捗が遅延すると共に、要員をより長期に渡って確保するためにコストも膨らんでしまう[6]。

本稿では、要件定義書の記述漏れと曖昧な記述を防止するため、要件定義書の品質を得点化する手

株式会社 NTT データ 技術開発本部 ソフトウェア工学推進センター シニアエキスパート
Senior Expert, Center for Applied Software Engineering, R&D Headquarters, NTT DATA Corporation
〒135-8671 東京都江東区豊洲 3-3-9 豊洲センタービルアネックス Tel: 050-5546-2303
Toyosu Center Bldg. Annex, 3-9, Toyosu 3-chome, Koto-Ku, Tokyo 135-8671 Japan Tel: +81 50 5546 2303

- 1) 株式会社 NTT データ 技術開発本部 ソフトウェア工学推進センター
Center for Applied Software Engineering, R&D Headquarters, NTT DATA Corporation
- 2) 株式会社 NTT データ 技術開発本部 ソフトウェア工学推進センター 部長
Senior Manager, Center for Applied Software Engineering, R&D Headquarters, NTT DATA Corporation
- 3) 株式会社 NTT データ 顧問
Adviser, NTT DATA Corporation
- 4) 株式会社 NTT データ 技術開発本部 副本部長
Deputy Senior Executive Manager, R&D Headquarters, NTT DATA Corporation
- 5) 株式会社 NTT データ 技術開発本部 システム科学研究所 所長
Senior Executive Manager, Research Institute for System Science, R&D Hdqrs., NTT DATA corporation

法(要件定義書品質スコアリング)を提案する。提案手法では、開発プロジェクト外部の人間(第三者)が手順書に従って記述漏れや曖昧な記述がないことを系統的に確認し、要件定義書の品質を表すスコアを算出する。第三者が要件定義書の品質の低さ(つまり、記述漏れや曖昧な記述)を指摘することで、顧客とベンダに見積り誤りや再作業・追作業回避に向けた行動をとる契機を与えられる。

要件定義書のスコアは、「書くべき内容」が「曖昧でない書き方」で記述されているかを基準として、文書化された手順により算出される。「書くべき内容」と「曖昧でない書き方」は、良い要件定義書の作成について説明した IEEE Std 830-1998[5]に基づいて定義され、手順書ならびにスコア算出シートとして文書化されている。スコアリングの実施者は、手順書に従って要件定義書を確認した結果をスコア算出シートに記入することでスコアを自動的に算出できる。このため、システム開発の知識がある程度あれば、スコアリングにより要件定義書の評価する人間(評価者)の違いに関わらず、およそ同程度の所要時間でおよそ同程度のスコアを算出できると期待される。

本稿では、提案手法の信頼性と有効性を検証するためのケーススタディについても報告する。ケーススタディでは、ある企業におけるシステム開発 10 案件で作成された要件定義書に提案手法を適用した。提案手法によって算出したスコアの信頼性を検証するため、プロジェクト外部の第三者 4 人で 3 案件の要件定義書を個別にスコアリングし、4 人がそれぞれ算出したスコアや作業時間がおよそ同程度になることを確認した。また、提案手法の有効性を検証するため、各案件を担当するプロジェクトマネージャ(PM)から提案手法が「要件定義書の記述漏れと曖昧な記述の防止」に役立つかどうか定性的評価(コメント)を収集した。

以降、2章で関連研究、特に第三者による品質評価、並びに成果物の品質定量化に関する取り組みを紹介する。3章で要件定義書品質スコアリングの構成要素を説明する。4章で信頼性と有効性を検証するためのケーススタディについて報告し、5章でまとめと今後の課題について述べる。

2. 関連研究

開発当事者としての知識を持たない、プロジェクト外部の担当者(第三者)が成果物の品質を評価する取り組みは、レビューやインスペクションの事例として報告されている[9]。岡田ら[10]は、富士ゼロックス株式会社における第三者テスト部門の取り組みを紹介している。岡田らの報告によると、開発部門からの依頼に応じ、テストを専門とする部門の担当者が開発されたシステムと顧客要件の適合性をブラックボックステストによって検証する。本稿における提案手法でも、プロジェクト外部の専門担当者によるスコアリング実施を想定している。ただし、より上流の成果物である要件定義書の評価する点が岡田らの報告と異なる。開発の早期に問題を検出することで後工程での手戻りを防止し、スケジュールの遅延を低減できる[10]。

細川[4]は、日本 IBM 株式会社における第三者インスペクションの取り組みを紹介している。専門の担当者が、開発部門からの依頼に応じて要件定義書のインスペクションを実施する。これにより、第三者の視点から客観的に問題を指摘できるだけでなく、開発を担当する要員と並行して品質評価の作業を実施できる。さらに、専門の担当者に経験を蓄積することでインスペクション作業も効率化できる。本稿における提案手法では、要件定義書の品質を第三者が評価することに加え、結果をスコアとして定量化する。品質の高低をわかり易く表現することで、顧客とベンダ双方に品質改善に向けた行動を促すことができる。

成果物の品質を定量化する試みは、ソースコードの複雑さを定量化する研究で多く行なわれてきた。McCabeら[8]やHalstedは[3]、ソースコードを形式的に調査し、ソースコードの複雑さを系統的に計算する方法を提案した。例えば、McCabeはソースコード中の独立した経路の数を循環的複雑度(Cyclomatic Complexity)と名付け、複雑さを表す指標として提案した[8]。本稿の提案手法も要件定義書の記述内容を意味的に理解するのではなく、形式的に調査することでスコアリングを実施する。ただし、ソースコードより上流の成果物である要件定義書の品質を定量化することで、開発の早期に手戻り低減に向けた行動を促すことができる。

小池ら[7]は、レビューの質と価値を定量化する手法について提案した。小池らの手法では、事前配布資料の有無、レビュー参加者の人選などから「レビューの質」を、次工程への持ち越しを防止した不具合の規模と影響度から「レビューの価値」を定量化する。この手法によってレビューの質と価値を定量化し、文書の品質を間接的に評価できる。一方、本稿における提案手法は文書の品質を直接的に定量化する。小池らの手法と提案手法を併用することで、複数の観点から補完的に要件定義書の品質を評価できる。

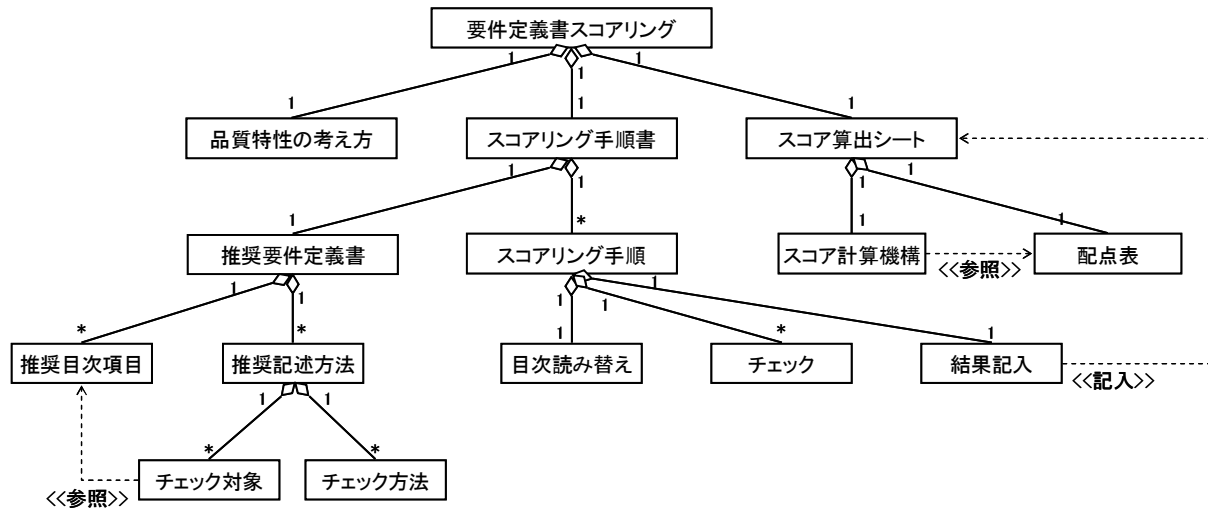


図 1. 要件定義スコアリングの構成要素

3. 要件定義書品質スコアリング

3.1 構成要素

図 1 に要件定義書スコアリングの構成要素を UML のクラス図として記す。要件定義書スコアリングを構成する各要素を四角形で表し、要素間の関係、特に集約関係を直線で記している。菱形の印を記した方が親となる要素であり、もう一方の先が親を構成する子要素である。親と子が 1 対 1 の関係にある場合は直線の両端に 1 を、1 対多の関係にある場合は、1 と * 印を各々記した。例えば、図中の「スコアリング手順」は「目次読み替え」、「チェック」、「結果記入」から構成される。「スコアリング手順」と「目次読み替え」、「チェック」、「結果記入」は各々 1 対 1、1 対多、1 対 1 の関係であることを示す。

以降本章では、「要件定義書スコアリング」の構成要素である「品質特性の考え方」、「スコアリング手順書」、「スコア算出シート」について、各々の子要素も含めて説明する。

3.2 品質特性の考え方

要件定義スコアリングでは要件定義書の品質を表 1 に示す 4 つの特性として捉える。これらの特性は、良い要件定義書の作成について述べた IEEE Std. 830-1998[5] (以降、IEEE830 と略記する) で説明されている品質特性を基に、株式会社 NTT データ (以降、当社と略記する) におけるプロジェクト管理の有識者から得た意見に基づいて定義したものである。その際、「第三者が客観的に評価可能であること」という観点で概念を整理・統合して新たに定義し、さらに定義した内容を実務者が連想し易いように名称を変更した。表の 1 列目に要件定義スコアリングにおける各品質特性の名称、2 列目に各特性の定義、3 列目に各特性の起源となる IEEE830 の品質特性を記す。

表 2 に IEEE830 の品質特性と、要件定義スコアリングでそれらをどのように取り扱うかを示す。表の 1 列目に IEEE830 における各品質特性の名称、2 列目にその概要、3 列目に要件定義スコアリングにおける取り扱いを示す。なお、IEEE830 の品質特性は原文どおりの英語表記とした。

IEEE830 における Verifiability について、Unambiguity の担保により Verifiability を向上できると考え、提案手法における「明確性」として概念を統合した。同様に、IEEE830 における Traceability の担保により Consistency と Modifiability を向上できると考え、提案手法における「追跡可能性」として概念を統合した。また IEEE830 における Ranked for importance and/or stability について、実務的には要件定義書に記述されることが多いが、本来はプロジェクト計画書や設計書などを作成する際、顧客が求める「要件」とは別の観点から議論する事項とし、要件定義書スコアリングの対象外とした。

3.3 スコアリング手順書

図 1 に示すように「スコアリング手順書」は、「推奨要件定義書」と「推奨記法」を記述したドキュメントである。「推奨要件定義書」は理想的な (100 点満点の) 要件定義書を記したものであり、書くべき内容

表 1. 要件定義書スコアリングの品質特性と、その起源となる IEEE Std. 830-1998 の品質特性

要件定義スコアリングの品質特性	定義	起源となる IEEE Std. 830-1998 の品質特性
合目的性	開発するシステムの目的が明らかであり、全ての要件はその目的に適っている。	Correctness
記述項目網羅性	要件定義書に書くべき項目が全て記載されている。	Completeness
明確性	全ての記述内容の意味が明確である。	Unambiguity, Verifiability
追跡可能性	各記述内容の起源(提案書や企画書などの記述内容、要件定義書内の他の記述内容など)が明確であり、かつ、各記述内容が後工程で参照しやすいよう容易に識別できるように記述されている。	Consistency, Modifiability, Traceability

表 2. IEEE Std. 830-1998 の品質特性と、要件定義書スコアリングにおける取り扱い

IEEE830 の品質特性	概要説明	要件定義スコアリングにおける取り扱い
Correctness	要件に誤りがなく、全ての要件は開発するシステムの満たすべき要件である。	「合目的性」として名称および概念を変更した。
Unambiguity	記述されている全ての要件について、一通りの解釈しかできない。	「明確性」として名称を変更した。概念は変更なし。
Completeness	要件定義書に書くべき項目が記載されている。	「記述項目網羅性」として名称を変更した。概念は変更なし。
Consistency	各要件がお互いに矛盾しておらず、一貫している。	「追跡可能性」として概念を Traceability へ統合した
Ranked for importance and/or stability	各要件について重要度や安定度(要件が変更される可能性の低さ)の観点で重み付けがされている。	顧客が求める「要件」とは別の観点で区別して議論する事項として対象外とした。
Verifiability	現実的なコストの範囲内でシステムが要件を満たすかどうかをチェックできる。	「明確性」として概念を Unambiguity へ統合した。
Modifiability	要件の変更に対して、容易かつ完全で一貫性を保って修正可能な構成である。	「追跡可能性」として概念を Traceability へ統合した。
Traceability	各要件の起源が明確であり、かつ、各要件が将来の開発ドキュメントから容易に参照できる。	「追跡可能性」として名称および概念を流用した。

を説明した「推奨目次項目」と、曖昧でない書き方を説明した「推奨記述方法」から構成される。さらに、各「推奨記述方法」に対して1つ以上の「チェック方法」と「チェック対象」を規定し、推奨要件定義書からの乖離の程度からスコアを算出する。算出されるスコアの値域は[0.00, 100.00]であり、スコアが大きい程、推奨要件定義書との乖離が少ない、つまり要件定義書の品質が高いことを示す。「スコアリング手順」では、スコアリングを「目次読み替え」、「チェック」、「結果記入」という手順として説明している。「結果記入」の説明において「スコア算出シート」を参照し、その記入手順を説明している。

表 3 に推奨目次項目を示す。表の1列目、2列目に「章」と「節」の名称、3列目に各章の内容を示す。例えば、「1章. 本要件定義書について」という章は、1.1から1.4まで4つの節から構成され、要件定義書というドキュメントそのものについての説明を記述する。当社業務ノウハウが含まれるため機密保持の観点から表 3 では割愛したが、「推奨目次項目」では節よりもう一段階詳細な「項」までを規定している。実務的に開発プロジェクトで作成される要件定義書は各々様式が異なるため、「推奨目次項目」どおりには記述されていない。このため、図 1 の「スコアリング手順」で示すように、最初に「目次読み替え」を行い、表 3 に示す目次項目に読み替えた上で「チェック」や「結果記入」を行なう。

表 3. 推奨目次項目の章・節とその内容

章	節	内容
1 章. 本要件定義書について	1.1. 要件定義書の目的	要件定義書というドキュメントそのものを説明する。ドキュメントの内容、読み方、関係する文書などを説明する。
	1.2. 要件定義書の想定読者	
	1.3. 要件定義書の構成	
	1.4. 参考文書	
2 章. システム開発概要	2.1. システム化の目的	開発するシステムを取り巻く諸要因(当該プロジェクトの上位目的や関連する周辺業務など)や当該システムの概要を説明する。
	2.2. 業務概要とシステム化の範囲	
	2.3. 制約事項	
	2.4. 用語集	
3 章. 要件に変更を与える事項・未確定事項	3.1. 要件に変更を与える事項	要件定義完了時に想定されるリスク要因を説明する。
	3.2. 未確定事項	
4 章. 機能要件	4.1. 業務フロー定義	業務の流れ、業務における機能の働きと詳細、機能の背景にあるデータモデルを詳細に説明する。
	4.2. 機能定義	
	4.3. データモデル定義	
5 章. 非機能要件	5.1. 品質要件	性能、信頼性、セキュリティなどの品質に加え、ネットワークなどのシステム基盤、運用、保守、移行などについて詳細に説明する。
	5.2. システム基盤構成要素要件	
	5.3. システム基盤機能要件	
	5.4. システムライフサイクル要件	
	5.5. 移行要件	

表 4. 推奨記述方法とチェック方法・チェック対象、および加点対象の品質特性(抜粋)

推奨記述方法	チェック方法	チェック対象	品質特性
「システム化の目的」を詳細に記述すること。	1. チェック対象に含まれる文の数を(句点をデリミタとして)数える。 2. 一文に2つ以上の内容が混合して記述してある文の数を数える。 3. 上記1と2を比較し、2が少ないほど加点する。	「2.1. システム化の目的」における「プロジェクトの目的」と「目的達成のための施策内容」について記述した項	合目的性
推奨する記述内容を充実させること。	1. 各目次に対応する内容が存在するかを確認する。存在すれば大きく加点する。 2. 対応する内容がない場合「当該プロジェクトで記述不要」、「現段階で未確定」、「他文書を参照」いずれか記述があるか確認する。それら記述の有無に応じて加点する。	全て目次項目	記述項目網羅性
明確性を向上させる推奨記法(UMLなど)を利用すること。	1. チェック対象に対応する内容が、(別途選定し、規定した)推奨する図で記述されていれば大きく加点する。 2. 図ではなく表や箇条書きとして記述されていれば、小さく加点する。 3. 自然言語で記述されていれば加点しない。	「2.2. 業務概要とシステム化の範囲」、「4.1. 業務フロー定義」、「4.3. データモデル定義」で、図として記述することを推奨する項	明確性
他の箇所を参照するとき、参照先を探しやすい形式で記述すること。	1. チェック対象に参照先があるか確認する。 2. 参照先がある場合、①参照先箇所が一意に特定できる、②参照先ページ範囲がわかる、③参照先文書がわかる、④参照先文書すらわからない、いずれかを確認する。 3. ①に近いほど大きく加点する。④は加点しない。	全目次項目(1章以外)	追跡可能性

品質特性	品質特性1				品質特性2				...	品質特性x			
推奨記法 推奨目次項目	推奨記述 方法 1-1	推奨記述 方法 1-2	...	推奨記述 方法 1- m_1	推奨記述 方法 2-1	推奨記述 方法 2-2	...	推奨記述 方法 2- m_2	...	推奨記述 方法 x-1	推奨記述 方法 x-2	...	推奨記述 方法 x- m_x
推奨目次項目i	チェック 結果 1-1-i	チェック 結果 1-2-i	...	チェック 結果 1- m_1 -i	チェック 結果 2-1-i	チェック 結果 2-2-i	...	チェック 結果 2- m_2 -i	...	チェック 結果 x-1-i	チェック 結果 x-2-i	...	チェック 結果 x- m_x -i
推奨目次項目ii	チェック 結果 1-1-ii	チェック 結果 1-2-ii	...	チェック 結果 1- m_1 -ii	チェック 結果 2-1-ii	チェック 結果 2-2-ii	...	チェック 結果 2- m_2 -ii	...	チェック 結果 x-1-ii	チェック 結果 x-2-ii	...	チェック 結果 x- m_x -ii
⋮	⋮	⋮	...	⋮	⋮	⋮	...	⋮	...	⋮	⋮	...	⋮
推奨目次項目y	チェック 結果 1-1-y	チェック 結果 1-2-y	...	チェック 結果 1- m_1 -y	チェック 結果 2-1-y	チェック 結果 2-2-y	...	チェック 結果 2- m_2 -y	...	チェック 結果 x-1-y	チェック 結果 x-2-y	...	チェック 結果 x- m_x -y

図 2. スコア算出シートの様式

表 4 に推奨記述方法とチェック方法・チェック対象、および加点対象の品質特性を示す。表の 1 列目に推奨記述方法、2 列目に推奨記述方法で記述されていることをチェックする方法、3 列目にチェック対象、4 列目にチェック結果に基づいて加点する品質特性を示す。当社業務ノウハウが含まれるため機密保持の観点から、チェック方法の一部を抜粋して説明する。例えば、「明確性を向上させる推奨記法 (UML など) を利用していること」という推奨記述方法で要件定義書が記述されているかどうかを、「2.2. 業務概要とシステム化の範囲」などで図として記述することを推奨する項が「推奨する図で記述されているかどうか」をチェックして確認する。チェック結果に基づき「明確性」のスコアを加点する。

3.4 スコア算出シート

図 2 にスコア算出シートの様式を示す。スコア算出シートは、行方向に各推奨目次項目、列方向に各推奨記述方法を配置したスプレッドシートとして実装されている。行と列が交わる各セルには、推奨記述方法で各目次項目が記述されているかどうかを表 4 に示す方法で、チェックして結果を入力する。チェック結果に基づいてセル毎のスコアが自動的に算出された後、行方向にセル毎のスコアを総計したものが各目次項目のスコア、列方向に総計したものが各推奨記法のスコアとして算出される。さらに、対応する推奨記法のスコアを総計したものが各品質特性のスコアとして算出され、全品質特性のスコアを総計したものが最終的な総計スコアとなる。

図 1 に示すように、スコア算出シートは「配点表」と「スコア計算機構」を内包している。「配点表」は各チェック結果のスコアに対する重みを設定した内部計算用の表である。配点表で定義された配点に基づいて、「スコア計算機構」が各品質特性、および総計スコアを自動的に算出する。「チェック方法」や「チェック対象」の詳細な規定に加え、スコアを自動計算することで評価者の主観や計算誤りによるスコアへの影響を低減する。これにより、評価者が異なる場合でも、同程度のスコアを算出できると期待される。

4. ケーススタディ

提案手法によって算出したスコアの信頼性、および、提案手法の有効性を検証するため、ある企業におけるシステム開発 10 案件で作成された要件定義書を用いたケーススタディを実施した。まず、提案手法によって算出したスコアの信頼性を検証するため、プロジェクト外部の第三者が要件定義書を客観的に評価できるかどうかを確認した。4 人の評価者が 3 案件の要件定義書を個別にスコアリングし、4 人が各々算出したスコアと作業に要した時間がどの程度ばらつくかを調査した。また、提案手法の有効性を検証するため、10 案件各々に対するスコアリング結果を各案件の PM に提示し、「要件定義書の記述漏れと曖昧な記述の防止」に役立つかどうか定性的な評価(コメント)を収集した。

スコアリングを実施した 4 人の評価者(以降、評価者 1~4 と呼ぶ)は皆システム開発の基礎知識を備えていたが、業務知識や経験は大きく異なっていた。評価者 1 は要件定義についての業務知識と経験が豊富なコンサルタントであり、評価者 2~4 は組込みソフトウェア開発の実務者だった。評価者 1 と 2 は共に

表 5. 評価者 4 人による 3 案件の要件定義書に対するスコアリング結果

	スコア(点)			作業時間(時間)		
	案件 A	案件 B	案件 C	案件 A	案件 B	案件 C
評価者 1	35.59	60.73	※ ² <u>83.87</u>	4.67	4.50	<u>8.91</u>
評価者 2	30.32	52.22	76.63	6.25	13.43	7.50
評価者 3	23.66	51.75	70.92	9.77	19.43	8.50
評価者 4	19.57	57.33	77.14	5.55	15.62	7.60
平均	27.29	55.51	77.14	6.56	13.25	8.13
標準偏差	7.09	4.30	5.30	2.23	6.34	0.69
変動係数※ ¹	<u>26.00%</u>	<u>7.75%</u>	<u>6.87%</u>	<u>34.08%</u>	<u>47.83%</u>	<u>8.49%</u>
最大値	35.59	60.73	83.87	9.77	19.43	8.91
最小値	19.57	51.75	70.92	4.67	4.50	7.50
最大値と最小値の差	<u>16.03</u>	<u>8.98</u>	<u>12.95</u>	<u>5.10</u>	<u>14.93</u>	<u>1.41</u>

※1 変動係数は標準偏差を平均値で除算した値である。平均値に対する値の相対的なばらつきを大きさを表す。

※2 本文で引用して説明している数値を太字と下線で強調した。

6年以上の開発経験がある一方、評価者 3 と 4 は入社 2 年目未満で開発経験に乏しかった。

表 5 に 4 人の評価者による 3 案件(案件 A~C)の要件定義書に対するスコアリング結果を示す。表の 1 行目~4 行目に各評価者の各案件に対するスコアと作業に要した時間を、5 行目以降に平均値や標準偏差などの統計値を示す。例えば、評価者 1 は案件 C のスコアリング作業に 8.91 時間を要し、83.87 点のスコアを算出した。スコアの値域は[0.00, 100.00]であり、値が大きい程要件定義書の品質が高いことを示す。また、作業時間は「要件定義書の内容確認」から「スコアを算出」し終える(ただし、スコアはスコア算出シートによって自動的に算出されるため、実際には評価者が全てのチェック結果を記入し終える)までの時間である。打ち合わせ、日程調整、要件定義書の電子ファイルをやり取りする時間は含まない。

変動係数(標準偏差を平均値で除算した値)、並びに、最大値と最小値の差に着目して、評価者が異なる場合のスコアのばらつきを考察する。スコアのばらつきは案件 A で最も大きく、標準偏差は 7.09、変動係数は 26.00%、最高点と最低点の差は 16.03 点だった。このことから、ばらつきが最も大きな場合でも、平均値の前後±7.09 点程度(平均値±26%程度)の個人差だったことがわかる。評価者の知識や経験が異なるのに対し、スコアはそれほど大きくはばらついていない。このことは、提案手法によってある程度客観的に要件定義書の評価できることのひとつの裏付けとなる。

次に、提案手法の有用性について確認するため「要件定義書の記述漏れと曖昧な記述の防止」に役立つかどうか、10 案件の PM から収集した定性的評価(コメント)を収集した。以下に収集したコメントを箇条書きで記す。各案件について PM と 3 回の打ち合わせを行い、スコアリング手法の目的と内容と進め方の説明、スコアリング作業の中間報告、最終結果報告とコメントの収集を実施した。10 案件でスコアリングの対象とした要件定義書は平均 A4 判 239 ページであり、担当者が作業に要した時間は平均 5.99 時間であった。メールや情報交換ツールを使つての要件定義書電子データのやり取り、打ち合わせの日程調整、打ち合わせ実施を含めると、各案件に約 10 営業日程度を要した。

- 記述・検討漏れのチェックとして有効(特に未確定事項と非機能要件についての指摘)。自分達の様式にない記述内容をカバーしている。
- 依頼後、3 営業日以内に結果が得られればお客様レビュー前に修正可能となる。
- 新規開発案件や更改案件では漏れなく要件を検討しなければならないため、特に有効である。
- 要件定義書への具体的修正案を記した一覧が欲しい。どこをどのように修正すればよいかをすぐに分かれば要件定義書へ反映しやすい。
- 結果を持ち帰って検討することで、要件定義書の問題を発見できた。
- 結果として出た点数が直感に合っている。
- 要件定義書スコアリングは主観的には有効だと感じる。ただし、スコアとプロジェクト成否の相関を調べ

るなど、提案手法の有効性を定量的に示した結果も必要だと感じる。

- 要件定義書から外した部分が記述不足と指摘されていた。要件定義書に書くべきことがわかった。
- スコアリングの指摘と同じ内容(機能要件の一部)をお客様からも指摘された。
- 続けて他の要件定義書に対するスコアリングも依頼したい。

ケーススタディの結果、提案手法が「要件定義書の記述漏れと曖昧な記述の防止」に役立つことを示すコメントを多く収集できた。一方で、「具体的な修正箇所・修正方法を示した一覧が必要」、「提案手法の有効性を定量的に示した結果も必要」など、提案手法の有用性を向上するための課題も得られた。

5. おわりに

本稿では、要件定義書の記述漏れと曖昧な記述の防止を目的とし、第三者が系統的手順に基づいて要件定義書の品質をスコアリングする手法を提案した。プロジェクト外部の第三者4人で3案件の要件定義書を個別にスコアリングし、提案手法によってある程度客観的に要件定義書を評価できることを確認した。また、10案件のPMから定性的な評価を収集し、提案手法の有効性を確認した。

今後の課題として、提案手法の定量的評価が挙げられる。著者らは、提案手法で算出したスコアと「記述漏れ」や「曖昧な記述」の関係分析を検討している。「記述漏れ」や「曖昧な記述」に起因する後工程での要件定義書変更(仕様変更)回数、手戻り工数、作業期間やコストの予実乖離などをデータとして収集し、スコアとの相関関係を分析する。スコアが高いほど仕様変更回数や手戻り工数が小さければ、提案手法が「要件定義書の記述漏れと曖昧な記述の防止」に有効であることを定量的に確認できる。

また、ケーススタディで得た知見に基づく提案手法の改善も課題である。実務で作成された要件定義書の内容、現場の技術者や有識者、スコアリングを実施した評価者の意見に基づき、推奨目次項目や推奨記述方法を修正する。さらに、要件定義書のスコアから、要件定義以降の工程における問題とその発生確率の予測を可能とするため、収集したデータの分析、チェック方法や配点の調整を行なってゆく。

6. 謝辞

要件定義書の品質評価にあたり、日本アイ・ビー・エム株式会社 板倉(鎌田)真由美氏からご助言をいただきました。また本稿の執筆にあたり、国立大学法人奈良先端科学技術大学院大学 森崎修司先生からご助言をいただきました。厚く御礼申し上げます。

7. 参考文献

- [1] 独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター編, “IT ユーザとベンダのための定量的見積りの勧め ～見積り精度を向上する重要ポイント～,” オーム社, 2005.
- [2] 独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター編, “経営者が参画する要求品質の確保 ～超上流から攻める IT 化の勘どころ～,” オーム社, 2005.
- [3] Halsted, M.H., “Elements of Software science,” Elsevier, New York, 1977.
- [4] 細川 宣啓, “第三者インスペクションによる品質検査と欠陥予防,” 情報処理, vol.50, no.5, pp.405-409, 2009.
- [5] IEEE-SA Standards Board, “IEEE Recommended Practice for Software Requirements Specifications,” Software Engineering Standards Committee of the IEEE Computer Society, IEEE Std 830-1998, 1998.
- [6] Kamata, M. I., Tamai, T., “How Does Requirements Quality Relate to Project Success or Failure?,” In *Proc. of the 15th IEEE Int'l. Requirements Engineering Conf. (RE2007)*, pp.69-78, 2007.
- [7] 小池 利和, 相澤 武, 中川 忠彦, 藤巻 昇, 渡井 克己, 山下 英祐, 奥村 学, “レビューの質と価値の定量化の提案,” 第27回ソフトウェア品質シンポジウム予稿集, 2008.
- [8] McCabe, Thomas J., and Butler, Charles W., “Design Complexity Measurement and Testing,” *Communications of the ACM*, Vol.32, No.12, pp.1415-1425, 1989.
- [9] 森崎 修司, “ソフトウェアインスペクションの動向,” 情報処理, vol.50, no.5, pp.377-384, 2009.
- [10] 岡田 俊明, 國部 宗紀, 杉山 篤志, “第三者テスト部門による設計品質向上とテスト生産性向上への取り組み,” SPI Japan 2007 –ソフトウェアプロセス改善カンファレンス 2007– 予稿集, 2007.