

自動車ソフトウェア要求仕様書の 第三者インスペクション方法の提案と適用評価

蛸島 昭之^{1,a)} 青山 幹雄²

受付日 2016年8月10日, 採録日 2017年1月10日

概要: 自動車ソフトウェアの規模と複雑度の増加, ならびに安全性や信頼性への高い要求にともない, ソフトウェア要求仕様書 (SRS) の重要度が増している. しかし, 自動車ソフトウェアの SRS は多様性が高いため, 体系的なインスペクションの適用が困難である. その結果, インスペクションの効果がインスペクタの経験とドメイン知識に依存している. 本稿では, ドメイン知識を必要としない第三者インスペクション方法を提案する. 提案方法では基準となる参照 SRS と自動車ソフトウェア向けの品質特性を定義し, 第三者インスペクションによって SRS のドキュメント品質を定量的に測定し評価する方法を提案する. 提案方法を 4 つの製品カテゴリから収集した実際の SRS に対して適用し, 品質スコアによる SRS 品質の定量的評価の有効性を示した. また, 自動車ソフトウェアの SRS の記述における多様なバリエーション要求と抽象度の混在の問題を明らかにした.

キーワード: 要求工学, ソフトウェア要求仕様書, インスペクション, 品質保証, 自動車ソフトウェア

A Third-party Inspection Method for Software Requirements Specifications and Its Evaluation with Multi-domain Automotive Products

AKIYUKI TAKOSHIMA^{1,a)} MIKIO AOYAMA²

Received: August 10, 2016, Accepted: January 10, 2017

Abstract: Automotive software is facing the increase of size and complexity, and strenuous demands for safety and reliability. Software requirements specification (SRS) is consequently gaining importance in automotive software development. However, the diversity in automotive SRSs is becoming an obstacle to apply systematic inspection methods. The performance of an inspection, as a result, largely depends upon the inspector's experience and domain knowledge. This paper proposes a third-party inspection method. We define a reference SRS and quality characteristics for automotive software in order to propose a third-party inspection method that can quantitatively measure the document quality of SRSs. We applied the proposed method to actual SRSs collected from four product domains, and demonstrated its effectiveness. Furthermore, we identified diverse variability and confusion of the abstraction levels of requirements in the SRSs from the inspection.

Keywords: requirements engineering, software requirements specification, inspection, quality assurance, automotive software.

1. はじめに

自動車におけるソフトウェアの重要性が増すにつれて, 自動車ソフトウェアの規模は年々増大し, その複雑度は 10 年ごとに 10 倍のペースで増加している [5], [9], [14]. あわ

¹ 株式会社デンソー
DENSO CORPORATION, Kariya, Aichi 448-8661, Japan

² 南山大学
Nanzan University, Nagoya, Aichi 466-8673, Japan

^{a)} akiyuki_takoshima@denso.co.jp

せて、安全性や信頼性に対する厳しい要求もある。このようなソフトウェアの開発を所定の予算と期間を満たしつつ完了するには要求工学 (RE) の活用が不可欠である [12]。なかでもソフトウェア要求仕様書 (SRS) のインスペクションがソフトウェアの品質、開発コスト、期間を満たすうえで効果があることが知られている [21]。自動車産業は伝統的に垂直統合型の構造をとってきたため [25]、製品開発には担当部署やサプライヤといった多くのステークホルダが関わる。そのため、異なるステークホルダ間で開発ゴールを共有する手段としても SRS がきわめて重要な役割を果たす。自動車ソフトウェア開発プロセスのフレームワークを定めた業界標準のプロセスモデルである Automotive SPICE [30] では SRS 品質確保のための検証を求めており、多くの自動車ソフトウェア開発組織では検証手段としてインスペクションを採用している。しかし、その実施方法は開発組織に任されており、インスペクションの実施が必ずしも SRS の品質向上につながっていないとの報告もある [11]。自動車ソフトウェア開発でインスペクションの適用を妨げる 2 つの要因がある。

1 つは自動車ソフトウェアのインスペクションにとって基準となる参照 SRS の構造と品質特性が確立していないことである。そして、もう 1 つは自動車ソフトウェアの多様性である。Pretschner らは自動車ソフトウェア工学のサーベイ論文において、製品ドメインをインフォテイメント系、ボディ/コンフォート系、セーフティ系、パワートレイン系、インフラ系の 5 つに分類している [25]。自動車ソフトウェア開発では、これらのドメインごとに異なる知識とスキルが求められるため、異なる組織で担務する開発形態が一般的である。第 1 著者の組織では前述の 5 つのドメインとは異なるが、パワートレイン事業部、電子事業部、熱事業部、情報安全事業部に分かれて製品開発を行っている [7]。異なる開発組織に属するプロジェクトでは独自に SRS の構成要素や品質評価の基準を定義しているため、開発組織を越えて全社的に統一したインスペクションの方法が構築されていない。そのため、インスペクションの成果はインスペクタの経験やドメイン知識に依存してしまう。さらに、豊富な経験と知識を有するドメイン専門家の人数は限られており、インスペクションの効率と品質保証が困難となっている。

2. 研究の枠組み

2.1 研究のコンテキスト

本研究は自動車部品サプライヤの開発組織において、自動車ソフトウェアシステムを対象とする。サプライヤは自動車メーカーから ECU (Electrical Control Unit) または複数 ECU を組み合わせたシステムの単位で製品開発を受注する。自動車メーカーでは複数のサプライヤから納品された ECU やソフトウェアシステムを統合して車両を開発す

る [17]。

サプライヤは自動車メーカーから製品全体に対する要求仕様書を受領し、それをもとにソフトウェア要求仕様書 (SRS)、ハードウェア要求仕様書 (HRS) を作成し製品を開発する。本稿では、そのうちの自然言語で記述された SRS を研究の対象とする。

SRS の大部分は自然言語によって記述されるが、自然言語で自動車システムのような複雑で動的なシステムの要求を記述すると無曖昧性、完全性、一貫性といった重要な記述上の特性を損なうことが知られている [33]。そのため、多くの SRS では必要に応じて UML、データフロー図、決定表などの各種図表を用いて補足している。パワートレインなどの連続制御系では、要求の一部が Simulink などのモデルで表現されることもある。記述言語は主に日本語と英語である。

2.2 本研究の目的

SRS の品質は要求定義工程以降の生産性と最終的なソフトウェアの品質に大きな影響を与える。そのため、体系的なインスペクションを実施し、ソフトウェア開発プロジェクトにおける SRS の品質を一定水準以上に引き上げることが望まれる。

体系的なインスペクションに関する先行研究として、IT システム開発を対象としたインスペクション設計方法が提案され、効果をあげている [28]。しかし、IT システムとは異なり多様な製品ドメインを持つ自動車ソフトウェアの開発に、既存のインスペクション設計方法をそのまま適用することはできない。そのため、本研究の目的は以下の 2 つである。

- 1) 自動車ソフトウェアの特性に適した体系的なインスペクション方法を定義する。
- 2) 製品ドメインの違いによる SRS の構造的な差異を扱う方法を定義する。

以上の目的から、本稿ではドメイン知識を持たない第三者でも製品領域を限定せずに実施可能な第三者インスペクション方法を提案する。

2.3 研究課題

提案方法の適用と評価を通して以下の研究課題への回答を試みる。

- RQ1. 本稿で提案する第三者インスペクション方法は SRS 品質の改善に有効か？
- RQ2. IEEE Std. 830-1998 で推奨されている目次項目は自動車ソフトウェアの SRS にとって適切か？
- RQ2.1 自動車ソフトウェアシステムの要求を記述するには不要な目次項目はあるか？
- RQ2.2 自動車ソフトウェアシステムの要求を記述するには不足している目次項目はあるか？

3. 関連研究

3.1 SRS の構造

SRS に関する国際規格には IEEE Std. 830-1998 [15] がある。そのほかに、アメリカ国防総省が定めた Software Requirements Specification (DI-IPSC-81433A) [8], NASA が定めた NASA Software Documentation Standard (NASA-Std-2100-91) [22] などがある。規格以外では、Robertson らがコンサルティング経験から得られた知見をもとに Volere Requirements Specification Template を提案している [26], [27]。Wieggers らも著書 [32] の中で SRS のテンプレートを提案している。しかし、自動車ソフトウェアの要求に対するこれらのテンプレートの適合性などを評価した研究はない。

3.2 SRS 品質特性と測定方法

SRS 品質特性として、IEEE Std. 830-1998 で定義された 8 つの品質特性が広く用いられている。しかし、IEEE Std. 830-1998 では、各品質特性を測定する方法は述べられていない。Davis らは 24 の SRS 品質特性と各特性を測定するメトリクスを提案している [6]。しかし、これらの測定手法は自然言語で記述された SRS をインスペクションするには具体性に欠け、製品開発にそのまま適用するには不十分である。

3.3 インスペクション方法

Fagan によるインスペクション (Fagan Inspection) [10] の提案以来、その改良に関する多くの提案がある [4]。

3.3.1 N 重インスペクション

N 重インスペクションとは複数の小人数チームでインスペクションを実施する方法である。1 つの大人数チームで実施するよりも多くの欠陥を検出できるという報告がある [20]。しかし、インスペクションの効果はインスペクタの専門知識に最も強く依存するという結果も得られている [16]。そのため製品ドメインによらず全社的に統一されたインスペクション方法としては不十分である。

3.3.2 段階的インスペクション

段階的インスペクション [18] は各段階で特定の特性、たとえば移植性、再利用性、保守性などに着目してインスペクションを行う方法である。しかし、ドメイン知識の有無で段階を分けることは想定されていない。また、その適用と効果の検証はソースコードに対してのみ行われており、要求インスペクションへは適用されていない。

3.3.3 インスペクションシステム設計手法

SRS の品質を定量的に測定し、異なるプロジェクト間で SRS 品質の相対比較を行えるインスペクション方法を設計するための方法論 (RISDM) が提案されている [28]。RISDM では標準 SRS およびそこからテラリングされた

SRS をインスペクションの対象としている。しかし、製品ドメインごとに形式の異なる多様な SRS をインスペクションするには RISDM 手法は十分ではない [29]。

3.4 自動車 SRS に関する研究

Weber らの発表した Daimler の自動車開発における要求工学に関する論文 [31] を契機に、自動車ソフトウェア開発分野でも要求工学の重要性が認識されるようになった。その後、欧州の自動車メーカ、主要サプライヤ、および大学を中心に自動車要求工学に関する研究が活発に行われてきた。自動車要求工学に関する研究のうち、SRS を対象とした研究には以下のものがある。

Ott は Daimler での乗用車向けボディ製品の SRS と関連仕様書の問題 5,999 件を分析し、4 階層の木構造からなる品質特性モデルを提案している [23]。

Langenfeld らは Bosch での約 5 年間にわたるハイブリッド車向け DC-DC コンバータ SRS の 588 個の欠陥について、IEEE Std. 830-1998 の品質特性と独自に定義した欠陥の発生源に基づき分析している。その結果、品質特性のうち正確性と完全性が要求欠陥の 61% を占め、それらに一貫性を加えた 3 つが最も修正コストが高いことが明らかになった [19]。

Aceituna は組込みシステムの SRS 作成者が考慮すべき実行時の状況 (Operational Context) など 6 つの関心事 (Concern) を定義し、各関心事に対するチェックリストを提案している [1]。

しかし、これらの研究では個々の機能要求と非機能要求の記述の評価が対象であり、本研究のように SRS 全体を対象としてドキュメント品質を定量評価する研究は行われていない。

4. アプローチ

4.1 SRS 品質モデル

SRS の品質は図 1 の UML クラス図に示すようにドキュメント品質と要求品質の 2 つの要素から構成される [2]。ドキュメント品質は SRS が満たすべき品質のうち記述に関する品質であり、一般にドメイン知識のない第三者が測定可能である。一方、要求品質は SRS の内容に関する品質であり、測定にドメイン知識が必要となる。要求品質からドキュメント品質への依存は、要求品質を高めるためには

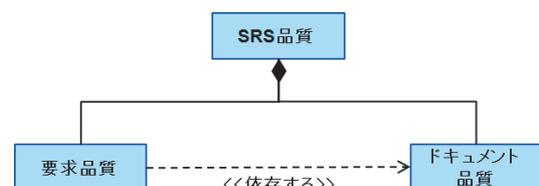


図 1 SRS 品質モデル

Fig. 1 SRS quality model.

ドキュメント品質の確保が前提となることを表している。

4.2 第三者インスペクション

全社のソフトウェア開発プロジェクトにおける SRS の品質を一定水準以上に引き上げるためには体系的なインスペクション方法が必要である。しかし、自動車ソフトウェア開発では、製品ドメインごとに異なるドメイン知識が必要となるため、インスペクション方法も製品ドメインの特性を反映させる必要がある。そこで、本稿では前述の SRS 品質モデルに着目し、第三者が測定に製品ドメインの知識を必要としないドキュメント品質のインスペクションを第三者インスペクション方法として体系化する。ドメイン知識を必要とする要求品質のインスペクションは本稿の対象外とし、従来どおりドメイン専門家がインスペクションを行うことを前提とする。

このように対象の製品ドメインを限定しない第三者インスペクション方法を構築することで、全社横断的に効率的なインスペクションの実施が可能となる。また、事前に十分なドキュメント品質を確保しておくことで、ソフトウェア開発チームとドメイン専門家が要求品質のインスペクションを効率的に行うことが可能となる。

5. 自動車 SRS の第三者インスペクション方法

5.1 第三者インスペクションの目的

第三者インスペクションの目的は要求品質インスペクションに先立ち、SRS がドキュメント品質を満たしていることを確認することである。

5.2 第三者インスペクションのプロセス

提案する第三者インスペクションのプロセスを図 2 に示す。このプロセスは、第三者によるドキュメント品質インスペクションとプロジェクトチームおよびドメイン専門家による要求品質インスペクションの 2 段階のインスペクションを想定しているが、本研究の対象は第三者インスペクションに限定する。

- (1) インスペクタによる SRS の受領
要求アナリストや開発チームからなるプロジェクトから独立したインスペクタが、要求アナリストが作成した SRS を受領する。
- (2) 第三者によるインスペクションの実施
インスペクタが第三者の立場で SRS のインスペクションを実施する。
- (3) アセスメントレポートの作成
第三者インスペクタはインスペクション結果から得られた品質スコアに基づきアセスメントレポートを作成する。
- (4) プロジェクトによる SRS とレポートの受領
プロジェクトは第三者インスペクタが作成したアセス

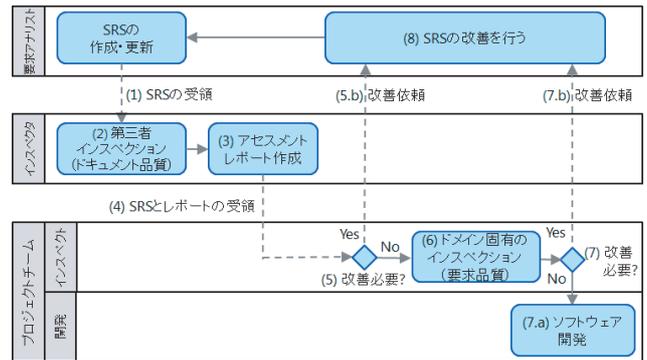


図 2 第三者インスペクションのプロセス

Fig. 2 Third-party inspection process.

メントレポートとともに要求アナリストの作成した SRS を受領する。

- (5) 第三者インスペクション結果の判断
 - 5.a) インスペクション結果が品質基準を満たす場合、プロジェクトチームはチーム内での SRS の要求品質インスペクションを計画する。
 - 5.b) インスペクション結果が品質基準を満たさない場合、要求アナリストへ SRS の改善を依頼する。
- (6) プロジェクトによるインスペクションの実施
プロジェクトを構成する管理者、開発者、テストなどがそれぞれの立場からドメイン知識を活用して SRS の要求品質インスペクションを実施する。必要に応じてプロジェクト外のドメイン専門家にインスペクションへの参加を依頼する。
- (7) 要求品質インスペクション結果の判断
 - 7.a) 要求品質インスペクション結果が品質基準を満たす場合、プロジェクトは SRS をベースライン登録しソフトウェアの開発に着手する。
 - 7.b) 要求品質インスペクション結果が品質基準を満たさない場合、要求品質インスペクションでの指摘事項をまとめ要求アナリストへ SRS の改善を依頼する。
- (8) SRS の改善
 - 8.a) 第三者インスペクション後に SRS 改善依頼を受けた場合、要求アナリストはアセスメントレポートを参考に SRS を改訂する。
 - 8.b) プロジェクトによる要求品質インスペクション後に SRS 改善依頼を受けた場合、要求アナリストは指摘に対して SRS を改訂する。

5.3 第三者インスペクションのメタモデル

第三者インスペクションのメタモデルを図 3 の UML クラス図に示す。SRS のドキュメント品質を定量的に評価するためには、SRS の構造を定義した参照 SRS とドキュメント品質特性が必要となる。参照 SRS の実体は SRS に含まれるべき要素とその構成を定めた SRS テンプレートである。インスペクションマトリクスは参照 SRS の目次項目

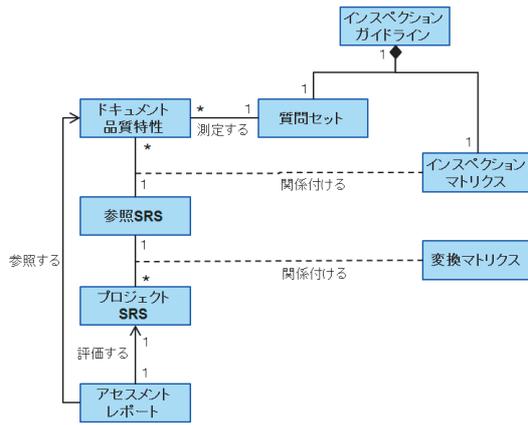


図 3 第三者インスペクションのメタモデル

Fig. 3 Third-party inspection meta-model.

と測定すべき品質特性を関連付ける。質問セットはインスペクタがSRSを読解しドキュメント品質特性を測定するための具体的な手段である。変換マトリクスはプロジェクトSRSの構成要素を参照SRSへ関連付ける。インスペクションマトリクスと質問セットの組合せをインスペクションガイドラインと呼び、第三者がプロジェクトSRSをインスペクションするための指針とする。アセスメントレポートはインスペクションから得られたSRS品質スコアを可視化した結果を提供する。

5.4 SRS品質スコアの測定方法

5.4.1 参照SRS

自動車ソフトウェアのSRSを記述するための適切な参照SRSはまだ提案されていない。本稿では、広く使われているSRSテンプレートの1つであるIEEE Std. 830-1998のSRSテンプレートを仮の参照SRSとする。この参照SRSに基づき、提案方法を実際の自動車ソフトウェアSRSへ適用し、評価することで、IEEE Std. 830-1998のSRSテンプレートが自動車ソフトウェアSRSのインスペクションのための参照SRSとして適切かを検証する。

5.4.2 SRS品質特性

IEEE Std. 830-1998 SRSテンプレートを参照SRSとするため、SRS品質特性の定義には、同じくIEEE Std. 830-1998で定義されている品質特性を参考とした。ただし、IEEE Std. 830-1998で定義されている8つの品質特性から重要度/安定度がランク付けされている(Ranked for importance and/or stability)を除外し、あらたに達成可能性(Achievable)を追加した8つの品質特性をSRS品質特性として定義した。ランク付けを除外したのは、自動車ソフトウェア開発では初期の要求スコープを開発途中で変更することはほとんど行われないためである[11]。逆に、達成可能性を追加した理由はIEEE Std. 830-1998で定義された品質特性には開発者のパースペクティブが不足しているためである。

表 1 SRS品質特性の割当て

Table 1 Allocation of SRS quality characteristics.

SRS品質特性	ドキュメント品質特性	要求品質特性
正当性	責任追跡性	正当性
無曖昧性	明確性	N/A
検証可能性		要求網羅性
完全性	記述網羅性	要求網羅性
無矛盾性	N/A	無矛盾性
変更容易性	変更容易性	N/A
追跡可能性	追跡可能性	N/A
達成可能性	N/A	達成可能性

5.4.3 ドキュメント品質特性と要求品質特性

第三者インスペクションを実施するためには、SRS品質モデルに基づきSRS品質特性をドキュメント品質特性と要求品質特性に割り当てる必要がある。あるSRS品質特性が単純にドキュメント品質特性か要求品質特性のどちらかへ割り当てられない場合、そのSRS品質特性を分解、または他のSRS品質特性と統合するなどして新たな品質特性を定義した後、ドキュメント品質特性か要求品質特性のどちらかへ割り当てる。本研究における、各SRS品質特性のドキュメント品質特性と要求品質特性への割当てを表1に示す。

5.4.4 インスペクションマトリクス

参照SRSのすべての要素がすべての品質副特性を満たす必要はない。不要な品質副特性の測定を避けインスペクションを効率化するため、参照SRSの目次項目と測定すべき品質副特性を関連付けるインスペクションマトリクスを定義する(表2)。マトリクス上で“X”が入っている個所が参照SRSに対するインスペクションポイントを表している。「1.1 Purpose(目的)」を例にとると、この節に対しては品質副特性C2-6、C3-3の評価が必要であることを意味する。この例では、インスペクションポイントは合計で149カ所である。本稿では、各インスペクションポイントで質問への答えがイエス(不備なし)であれば1点加点、答えがノー(不備あり)であれば加点しないというスコアリング方法を採用する。

5.4.5 質問セット

各インスペクションポイントにおいてドキュメント品質特性を測定するための具体的な方法としてドメイン知識がなくてもイエスまたはノーで客観的に回答できる質問(closed question)を定義する。ドキュメント品質特性は抽象度が高くそのままでは質問を定義できないため、抽象度の低い副特性へ分解する。各品質副特性に対応した15の質問(質問セット)を表3に示す。第三者インスペクタはSRSを読解しながら各品質副特性が満たされているかを確認するチェックリストとして質問セットを利用する。

表 3 質問セット
Table 3 Question set.

ID	品質特性	品質副特性	質問
C1-1	責任追跡性	開発目的	製品の開発目的が記述されているか？
C1-2		機能目的	機能の目的が記述されているか？
C2-1	明確性	非冗長	1つの文に複数の要求が記述されていないか？
C2-2		多義的でない	多義性を持つ用語が定義なしに記述されていないか？
C2-3		定量的	定量的に記述すべき箇所が定性的な記述になっていないか？
C2-4		あいまい性の回避	あいまい性の高い語句が使用されていないか？
C2-5		態	能動態で記述されているか？
C2-6		参照	外部文書を参照する場合は、参考文献欄に文書の場所を特定するのに十分な情報が記述されているか？
C3-1	記述網羅性	未決定事項	未決定事項には「解消方法」「解消期限」「責任者」が明記され「追跡のための番号」が振られているか？
C3-2		ラベル	図表にはタイトルが付けられ、ユニークな番号が振られているか？
C3-3		テンプレート	標準SRSで指定されている項目はすべて記述されているか？
C4-1	変更容易性	相互参照	他の要求を参照する場合、正しい要求への明示的な相互参照が記述されているか？
C4-2		検索可能性	目次と索引を作成しているか？
C5-1	追跡可能性	後方追跡性	上位要求とソフトウェア要求との間の関連を表すトレーサビリティマトリックスが作成されているか？
C5-2		前方追跡性	各要求にユニークなIDが振られているか？

表 2 インスペクションマトリクス
Table 2 Inspection matrix.

IEEE Std. 830-1998 目次項目	品質副特性									
	I-D	I-G	I-O	I-C	T	S	E	E	E	O
1. Introduction										
1.1 Purpose						X		X		
1.2 Scope	X							X		
1.3 Definitions, acronyms, and abbreviations						X		X		
1.4 References								X		
1.5 Overview						X		X		
2. Overall description										
2.1 Product perspective			X		X	X	X	X		
2.1.1 System interfaces			X		X	X	X	X		
2.1.2 User interfaces			X		X	X	X	X		
2.1.3 Hardware interfaces			X		X	X	X	X		
2.1.4 Software interfaces			X		X	X	X	X		
2.1.5 Communications interfaces			X		X	X	X	X		
2.1.6 Memory constraints			X	X	X	X	X	X		
2.1.7 Operations	X	X	X		X	X	X	X		
2.1.8 Site adaptation requirements	X	X	X		X	X	X	X		
2.2 Product functions			X		X	X	X	X		
2.3 User characteristics			X		X	X	X	X		
2.4 Constraints			X		X	X	X	X		
2.5 Assumptions and dependencies			X		X	X	X	X		
2.6 Apportioning of requirements			X		X	X	X	X		
3. Specific requirements				X		X	X	X		
3.1 External interfaces				X		X	X	X		
3.2 Functions										
3.2.1 System Feature										
3.2.1.1 Introduction/Purpose of feature	X	X			X	X	X			
3.2.1.2 Stimulus/Response sequence			X	X	X	X	X	X		
3.2.1.3 Associated functional requirements	X	X	X	X	X	X	X	X	X	X
3.3 Performance requirements	X	X	X	X	X	X	X	X	X	X
3.4 Logical database requirements	X	X	X	X	X	X	X	X	X	X
3.5 Design constraints			X		X	X	X	X		
3.6 Software system attributes	X	X	X	X	X	X	X	X	X	X
4. Supporting information										
4.1 Table of contents and index									X	
4.2 Appendixes										

インスペクションマトリクス

表 4 変換マトリクス
Table 4 Translation matrix.

プロジェクトSRS 目次項目	IEEE Std. 830-1998 目次項目									
	I-D	I-G	I-O	I-C	T	S	E	E	E	O
目次										X
変更履歴										
1. はじめに										
1.1 目的と適用範囲	X	X								
1.2 参考文献							X			
1.3 用語・略語							X			
1.4 I/Fとハードウェア前提条件									X	

変換マトリクス

5.4.6 変換マトリクス

プロジェクト SRS 間に存在する構成要素の差異を吸収し 統一的な手順でインスペクションを行うため、すべてのプロジェクト SRS の変換先として参照 SRS を定義する。参照 SRS とプロジェクト SRS の対応付けは、プロジェクト SRS の目次項目と参照 SRS の目次項目を対応付ける変換マトリクスを作成することで行う (表 4)。変換マトリクス上で“X”が入っている箇所が変換ポイントを表している。たとえば、プロジェクト SRS の「1.4 I/Fとハードウェア前提条件」は参照 SRS の「2.4 Hardware Constraints (ハードウェア制約)」と関連付けられる。

変換マトリクスは基本的に第三者インスペクタが作成す

る。プロジェクト SRS を作成した要求アナリストの協力が得られる場合、第三者インスペクタと要求アナリストがそれぞれ独立して変換マトリクスを作成し、変換結果の相違点について合意形成を行う。変換マトリクスの作成に際しては、記述ガイドラインと記述サンプルを参照し、プロジェクト SRS の目次項目ごとに該当する参照 SRS の目次項目を識別する。

5.4.7 インスペクションガイドライン

インスペクションマトリクスと質問セットの組合せをインスペクションガイドラインと呼ぶ。このインスペクションガイドラインと前述の変換マトリクスを用いることで、ドメイン知識を持たない第三者インスペクタでもプロジェクト SRS の各目次項目をどの品質副特性に対して測定すべきかを一意に特定できる。特定された品質特性の測定は質問への回答結果から機械的に行える。

5.4.8 アセスメントレポート

アセスメントレポートはインスペクションから得られた SRS 品質スコアを可視化した結果を要求アナリストとプロ

表 5 インспекション対象 SRS のデータセット

Table 5 Data set of inspected SRSs.

仕様書 ID	製品ドメイン	ページ数
SRS_A	インフラ	76
SRS_B	ボディ	34
SRS_C	ボディ	39
SRS_D	安全	41
SRS_E	コンフォート	53
SRS_F	安全	108
SRS_G	インフラ	19
合計		370

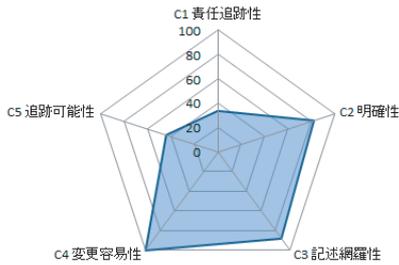


図 4 SRS_A の非正規化スコア (品質特性ビュー)

Fig. 4 Non-normalized quality characteristics view for SRS_A.

ジェクトチームへ提供する。アセスメントレポートの目的は要求アナリストとプロジェクトチームが SRS の記述上の欠陥を把握するのを支援し、SRS の改善を促すことである。

5.5 品質スコアの可視化方法

品質スコアは非正規化スコアと正規化スコアの 2 つの基準で評価する。非正規化スコアはプロジェクト SRS そのものを対象に採点したものである。一方、正規化スコアはそれぞれ異なる構成を持つプロジェクト SRS の目次項目を参照 SRS の目次項目へ対応付けてから採点したものである。また、品質スコアを異なる側面から分析するため、各スコア基準には構造と品質特性の 2 つのビューを定義する。品質スコア計算方法の詳細は付録 A.1 で説明する。

このほかに品質スコアの表現方法としてプロジェクト SRS 目次項目の参照 SRS 目次項目に対する網羅率を表す目次項目ヒストグラムと、品質特性および参照 SRS 目次項目との品質スコア分布を表すスコアヒートマップを定義した。

以下、表 5 で示したデータセットのうち SRS_A を例に、可視化された品質スコアからどのような知見が得られるかを議論する。

5.5.1 非正規化スコア

品質特性ビュー (図 4) および副特性ビュー (図 5) からは SRS 全体でどの品質特性を改善すべきかを評価できる。まず品質特性のレベルで見ると、SRS_A では主に責任追跡性 (C1) と追跡可能性 (C5) の 2 つの品質特性で改善が必要であることが分かる (図 4)。

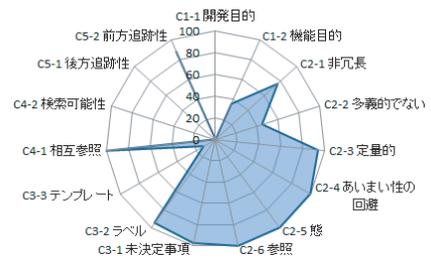


図 5 SRS_A の非正規化スコア (品質副特性ビュー)

Fig. 5 Non-normalized quality sub-characteristics view for SRS_A.

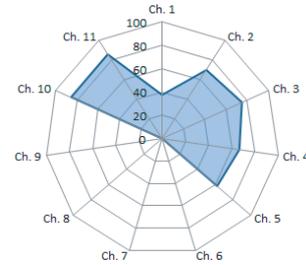


図 6 SRS_A の非正規化スコア (構造ビュー)

Fig. 6 Non-normalized structural view for SRS_A.

また、品質特性レベルでは 80% を超えている明確性 (C2) も副特性ごとに分析すると非冗長 (C2-1) と多義的でない (C2-2) ではまだ改善の余地があることが分かる (図 5)。

構造ビュー (図 6) からはプロジェクト SRS のスコアを章ごとに評価できるため、改善に着手すべき目次項目を特定する指針となる。SRS_A では、6 から 9 章がゼロ点となっている。これは、それらの章には SRS には本来記述すべきではない設計情報が記述されておりインспекションの対象とならなかったためである。6 から 8 章を除くと、1 章のスコアが他の章のスコアよりも相対的に低いことが分かる。

5.5.2 正規化スコア

正規化スコアはプロジェクト SRS の目次項目を参照 SRS の目次項目に対応付けた場合のスコアを算出したものである。正規化スコアも非正規化スコアと同様、品質特性ビュー (図 7) と品質副特性ビュー (図 8) および構造ビュー (図 9) から評価できる。正規化スコアの構造ビューからは、参照 SRS の目次項目を軸にプロジェクト SRS の品質を評価できる。SRS_A では参照 SRS の 1, 2, 4 章に相当する部分の品質スコアが 3 章に比べて相対的に低いことが分かる。

5.5.3 非正規化スコアと正規化スコアの使用方法

(1) 非正規化スコアの使用方法

非正規化スコアは、たとえば以下の場合に利用する。

- 同一の目次項目を持つプロジェクト SRS 間のドキュメント品質の比較
- 同一プロジェクト SRS の改訂前後でのドキュメント品質の比較

6. 適用

提案方法による自動車 SRS のインスペクションは 2015 年から適用を開始し、7つの SRS に対して提案方法によるインスペクションを実施した。適用対象の選定に際しては、データセットの多様性を確保するため複数の製品ドメインから SRS を収集した。

6.1 データセット

適用を行った7つの SRS が対象とする製品のドメインとページ数を表 5 に示す。すべての SRS は日本語で記述されている (SRS_F のみ英語も併記)。これらの SRS は第 1 著者の所属する組織で開発した6つの製品 (SRS_A から SRS_F) と、これから開発を行う製品 (SRS_G) の SRS である。各製品の要求仕様書は SRS に加えてハードウェア要求仕様書など複数のドキュメントから構成される。本稿の対象は、そのうちの SRS である。しかし、ソフトウェアに関する要求仕様書の名称は開発プロジェクトごとに異なっているため、適用に際しては各ドキュメントの内容と、開発プロセス内でのドキュメントの位置づけから IEEE Std. 830-1998 が定義する SRS に近いドキュメントを選択した。

複数の ECU で構成されているシステム製品の場合は ECU ごとに仕様書が作成されるため、システム製品からは主要な機能を担う ECU を 1 つ選択し、その ECU の SRS をインスペクションした。

1 つの ECU でも機能ごとに複数の SRS へ分冊される製品もある。これは、自動車ソフトウェアの特徴である仕向けや車両グレードごとの多様なカスタマイズへ対応するためである。SRS が分冊されている場合は、SRS の集まり全体を 1 つの SRS として評価を行った。

7. 評価

7.1 提案手法の効果検証 (RQ1)

7.1.1 検証目的

本稿で提案する第三者インスペクションによって得られる SRS 品質スコアが要求アナリストによるプロジェクト SRS の改善に対し有効であるかを検証する。

7.1.2 検証方法

表 5 に示したデータセットのうち SRS_A と SRS_G は同一の開発チームにより作成された。このチームは SRS 品質の低さを課題としてとらえており、新たに開発する製品 G では、SRS 品質の改善活動に自発的に取り組むこととなった。そこで、既存製品の要求仕様書 (SRS_A) に対し本稿で提案するインスペクション方法から得られた品質スコア (5.5 節参照) に基づき新たな製品の要求仕様書 (SRS_G) を作成した。そのため、SRS_G は SRS_A の版を改訂したものではなく、異なる製品向けに作成された新規の要求仕様書である。これら 2 つ要求仕様書 (SRS_A と SRS_G) の

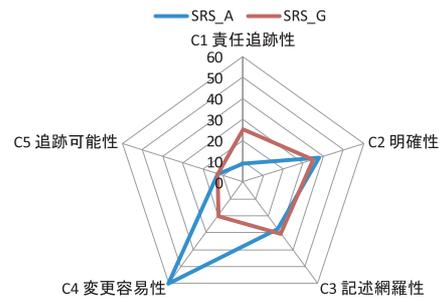


図 12 SRS 品質の比較 (品質特性ビュー)

Fig. 12 Relative comparison in quality characteristics view.

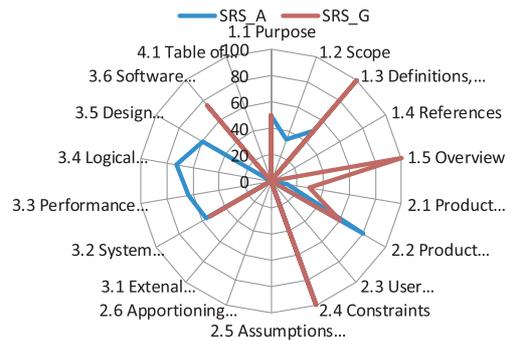


図 13 SRS 品質の比較 (構造ビュー)

Fig. 13 Relative comparison in structural view.

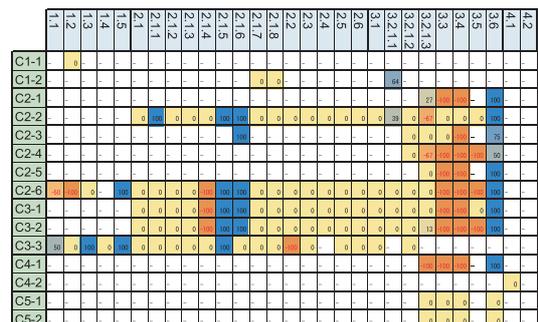


図 14 SRS_A 品質の比較 (スコアヒートマップ)

Fig. 14 Relative comparison in score heat map.

品質スコアを比較し、SRS_G の品質スコアが SRS_A よりも改善されているかを確認する。

7.1.3 検証結果

SRS_A と SRS_G の品質スコアを品質特性ビューと構造ビューで比較した結果をそれぞれ図 12 と図 13 に示す。

SRS_A と SRS_G は異なる構造を持つため、図 12 と図 13 の比較結果はともに正規化スコアによるものである。図 12 の品質特性ビューからは、SRS_G では SRS_A に比べて責任追跡性 (C1) のスコアが向上しているが、明確性 (C2)、記述網羅性 (C3)、追跡可能性 (C5) の 3 特性ではほぼ変化がなく、変更容易性 (C4) では逆にスコアが低下していることが分かる。次に図 13 の構造ビューによる比較を見ると、品質特性の場合と同様にスコアが向上している参照 SRS 目次項目もあれば低下している項目もある。

続いてスコアヒートマップ (図 14) により SRS_A か

ら SRS_G への品質スコアの増減を大域的に可視化する。図 14 は各インスペクションポイントに対し SRS_G の品質スコアから SRS_A の品質スコアを減じた結果を表している。このスコアヒートマップからは、SRS 全体で品質スコアが増減した品質特性はなく、品質スコアの変動は主に目次項目の増減に起因していることが読み取れる。

これらの分析結果から、SRS_G が SRS_A よりも改善されたとはいえない。これは、先行研究 [28] でも示されているように、1 回の SRS の改訂では必ずしも十分な改善が達成されなためと考えられる。本研究では開発チームが 2 回目以降の改訂を行うところまで追跡調査を行うことができなかった。しかし、従来の主観による判断とは異なり、品質スコアを用いた客観的な判断のもとでは、SRS の改訂を重ねることでその品質が向上すると考えられる。そのため、本稿で提案する第三者インスペクション方法から得られる品質スコアは SRS の改訂結果を評価し、十分な改善が達成したことを確認するために有効であるといえる。

7.2 参照 SRS の妥当性検証：不要項目の検証 (RQ2.1)

7.2.1 検証目的

参照 SRS に自動車ソフトウェアにとっては重要ではない目次項目が含まれていると、参照 SRS からプロジェクト SRS を作成する際に本来は必要ではない目次項目を記述する工数と、その目次項目をインスペクションする工数で二重の無駄が発生する。また、参照 SRS をもとに作成されていないプロジェクト SRS をインスペクションする場合、プロジェクト SRS にとっては不要な参照 SRS 目次項目に対して減点されてしまい、正規化スコアが本来より低く評価されるという問題もある。そのため、参照 SRS が自動車ソフトウェアにとって不要な目次項目を含まないことは重要である。そこで、自動車ソフトウェアの要求を記述するうえで IEEE Std. 830-1998 SRS テンプレートの目次項目には不要な要素がないかを検証する。

7.2.2 検証方法

7 つの SRS それぞれに作成された目次項目ヒストグラムをもとに作成した箱ひげ図 (図 15) から、SRS 記述量の分布を分析する。タイトル名の右端に * の付いている目次項目 (3.1, 3.2, 3.3, 3.4, 3.6) は一般的に仕様書のページ数が増えるほど出現数が多くなる。そのため、それらの目次項目の記述量の評価は、式 (1) で定義される記述密度を用いることとした。

$$\text{記述密度} = \frac{\text{目次項目の出現数}}{\text{目次項目のページ数}} \quad (1)$$

7.2.3 検証結果

正規化された項目を除くと、最小値が 1 未満の項目は少なくとも 1 つの SRS がその項目を含んでいないことを意味する。そのなかでも、中央値が 1 未満のものはデータセットすべての SRS で記述量が少ないことを表している。こ

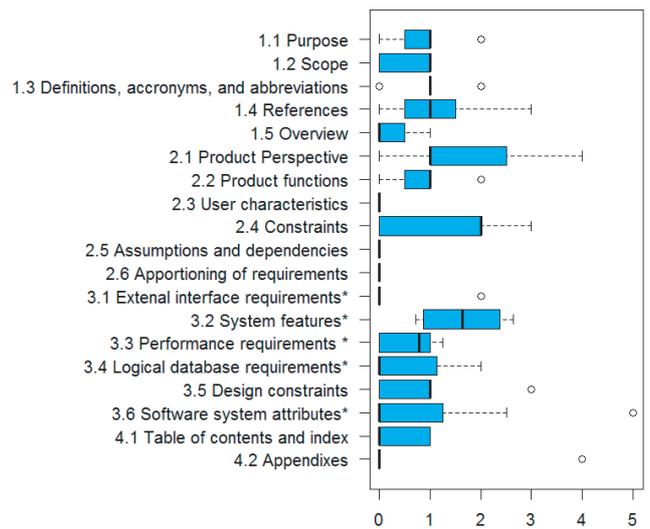


図 15 SRS 記述量の分布

Fig. 15 Distribution in amount of SRS description.

の結果から、すべての SRS で十分な記述が行われている項目はないことが分かる。そこで、1 章と 2 章について十分な記述が行われていない理由について考察する。

1 章の記述が少ないのは、ECU 単品で受注した製品の場合には、システム全体の目的といった 1 章に書かれるべき情報はあまり重要性ではないと多くのプロジェクトでは認識している可能性がある。

2 章については、すべての仕様書でまったく記述のなかった項目について考察する。これらの項目に関する記述がない理由は自動車ソフトウェアシステムの開発形態に起因する可能性がある。

まず、「2.3 User characteristics (ユーザ特性)」について考察する。自動車ソフトウェアではユーザインタフェースを持つものはナビゲーションソフトウェアなどに限定されるため、この項目の出現頻度が低くなったと考えられる。今回の評価サンプルの中にはユーザインタフェースを持つ製品がなかった。

次に「2.5 Assumptions and dependencies (前提と依存関係)」と「2.6 Apportioning of requirements (要求の割当て)」について考察する。自動車ソフトウェアの開発では、試作フェーズを繰り返した後に最終的な量産ソフトウェアを完成させる。試作フェーズごとに自動車メーカーから新たな要求が変更要求仕様書として提供される [17]。このように試作フェーズごとの要求の追加と変更が前提で開発が行われていることが、これらの項目が記述されない要因とも考えられる。また、自動車ソフトウェアの開発では、現時点では IT システムほど要求の変動が大きくないことも要因として考えられる。

データセットに対する SRS 記述量の分布分析から、「ユーザ特性」、「前提と依存関係」、「要求の割当て」の 3 つの目次項目に対する記述量が少ないことが確認された。これら

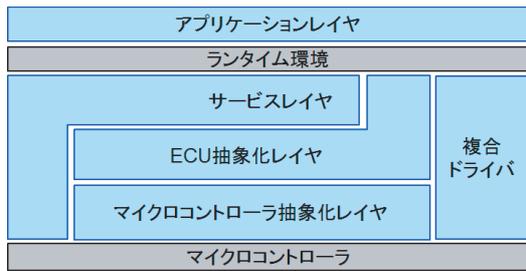


図 16 AUTOSAR 階層ソフトウェアアーキテクチャ [3]
 Fig. 16 AUTOSAR layered software architecture [3].

の目次項目に関して、より多くの SRS を対象に記述量を調査するとともに、記述量が少ない原因を検証する必要がある。

7.3 参照 SRS の妥当性検証：不足項目の検証 (RQ2.2)

7.3.1 検証目的

自動車ソフトウェアの SRS を記述するうえで IEEE Std. 830-1998 SRS テンプレートの目次項目には不足している要素がないかを検証する。

7.3.2 検証方法

IEEE Std. 830-1998 を参照 SRS として 7 つのプロジェクト SRS に対する変換マトリクスを作成した。その際に、参照 SRS の目次項目に適切な対応先が見つからなかった要素についてその原因を分析する。

7.3.3 検証結果

自動車ソフトウェアの特徴のうち IEEE Std. 830-1998 では適切に扱うのが難しい 2 つの特徴を特定した。

(1) 多様なバリエーション要求

インスペクションした 7 つの SRS のうち新規開発品向けの SRS.G を除く 6 つの SRS には特定の車種や仕向けのみで有効な要求が含まれていた。これは、自動車ソフトウェア開発には車種ごとのグレード間での差別化などのために非常に多くのバリエーションが要求されるためである。典型的なプレミアム車種では約 80 のバリエーションを持つ [25]。このように多様なバリエーションを扱う必要があるため、近年の自動車ソフトウェア開発ではプロダクトライン開発が取り入れられつつある。しかし IEEE Std. 830-1998 ではソフトウェアプロダクトライン開発にとって重要な共通性 (commonality) と可変性 (variability) の扱いが考慮されていない。

(2) 抽象度の異なる要求

製品の機能要求はすべて参照 SRS の「3.2 System Features (機能)」に属する。しかし、本研究でインスペクションした SRS には、システム全体の振舞いを表した抽象的な要求から、設計の領域まで踏み込んだ詳細な要求まで様々な抽象度の要求が存在した。一般的な自動車ソフトウェアは図 16 に示すような階層アーキテクチャを持ち、COTS (Commercial Off-The-Shelf) 製品を利用する場合を除く

と、すべてのレイヤが開発対象となる。そのため、抽象度の高いアプリケーションレイヤに関しては SRS が設計情報を含まないようにする必要がある [6], [15]。一方、抽象度の低いドライバレイヤでは SRS の記述も設計まで踏み込む必要がある [31]。したがって、自動車ソフトウェアの SRS では高レベルの要求から低レベルの要求まで抽象度の異なる要求を単一製品の SRS の中で扱う必要がある。しかし、IEEE Std. 830-1998 の目次項目と記述ガイドラインは抽象度の異なる要求を適切に構造化するには不十分である。

8. 考察

8.1 第三者インスペクションの有効性と適用性

IT システムの SRS インスペクションを対象とした先行研究 [28] では、ドメイン知識を有さない第三者でもすべての品質特性を測定している。しかし、自動車ソフトウェアは、それ自体に多様なドメインを内包し、インスペクションにはそれらのドメインごとに固有の知識が必要となる。そのため、IT システムの SRS インスペクション方法をそのまま適用することは困難である。そこで、本研究では SRS 品質モデルに基づきインスペクションをドメイン知識が必要な領域と不要な領域に分け、ドメイン知識を必要としない領域をインスペクションする具体的な方法として第三者インスペクション方法を定義した。実際の SRS に対して提案方法を適用し、品質スコアによる SRS 品質の定量的評価の有効性を確認した。

また、Pretschner らによる分類のうちボディ/コンフォート系、セーフティ系、インフラ系には提案手法が適用可能であることを確認した。今回のデータセットには含まれていないが、ボディ/コンフォート系と SRS の構造および記述内容が類似しているインフォテインメント系の SRS に対しても提案方法は適用可能であると推測できる。一方、パワートレイン系の SRS は記述内容の大部分が制御仕様に関するため、Simulink などのモデルによる記述の割合が多く、自然言語で記述された SRS を対象とする本提案方法には適していないと考えられる。

ドメイン知識がインスペクションに必要となることは自動車ソフトウェアに限らず、組み込みソフトウェア全般にいえることである。そのため、多くの組み込みソフトウェア開発組織では、本稿が提案する第三者インスペクションの導入により 4.2 節で述べた効果を得ることができると考えられる。

8.2 自動車 SRS の特徴とインスペクション

実際の 7 つの自動車製品ソフトウェア SRS の記述量を分析し、IEEE Std. 830-1998 SRS テンプレートの目次項目のうち自動車ソフトウェアの要求には適合しない可能性のある目次項目として「ユーザ特性」、「前提と依存関係」、

「要求の割当て」を特定した。一方、自動車ソフトウェアの要求にとっては重要であるが IEEE Std. 830-1998 の SRS テンプレートと記述ガイドラインでは適切に扱うことのできない項目として次の2つの特徴を特定した。

(1) 多様なバリエーション要求

自動車ソフトウェアをはじめプロダクトラインを採用する組み込みソフトウェアではバリエーション要求が多いことが指摘されている [24]。しかし、実際の SRS をインスペクションした結果、自動車ソフトウェアの SRS ではバリエーション要求の記述が体系化されていないことが確認された。そのため、フィーチャモデルのようなバリエーションを表現するモデルの自動車 SRS への適切な導入方法と、それらに対するインスペクション方法に関する研究が必要である。

(2) 抽象度の異なる要求

実際の SRS をインスペクションした結果、自動車ソフトウェアの SRS には抽象度の異なる要求が混在していることが明らかになった。このような課題に対して、要求の抽象度を階層化するモデルとして Requirements Abstraction Model (RAM) が提案されている [13]。しかし、階層化された要求をどのように記述するかまでは提案がなされていない。そのため、抽象度に応じて階層化された自動車ソフトウェア要求を記述するための SRS 構造と要求の抽象度に応じたインスペクション方法の研究が必要である。

自動車ソフトウェアの要求に最適な SRS の構造に関する提案はない。本研究により得られた知見は今後のこの分野での研究に対する1つの指針となりうる。

9. まとめ

本稿では第三者インスペクションの具体的な方法を提案し、その効果を評価するため7つの SRS に対して提案方法を適用した。

第三者インスペクションから得られた品質スコアは SRS 品質に対して客観的な視点を与えるため、SRS の改訂が実際に SRS 品質の改善につながったかを判断するために有効であることが確認できた。

7つの SRS に対する記述密度の分析からは、IEEE Std. 830-1998 SRS テンプレートのうち3つの目次項目は自動車ソフトウェアの SRS では記述量が少なく、その原因を検証する必要があることが明らかになった。また、IEEE Std. 830-1998 SRS テンプレートは自動車ソフトウェアの特徴である多様なバリエーション要求、抽象度の異なる要求を適切に構造化して扱うためには不十分であるという知見も得た。

本稿では、SRS 品質モデルのうちドキュメント品質特性を測定し評価するための第三者インスペクション方法を定義するにとどまったが、要求品質特性についても実用的なインスペクション方法を検討し、SRS 品質特性全体の測定と評価を行える包括的なインスペクションシステムの開発

にも取り組む予定である。

謝辞 (一社)情報サービス産業協会 (JISA) 要求インスペクション設計研究会の斎藤忍委員はじめ、ご討議いただいた委員各位に感謝します。

参考文献

- [1] Aceituna, D.: Survey of Concerns in Embedded Systems Requirements Engineering, *SAE Int'l J. Passenger Cars-Electronic and Electrical Systems*, Vol.7, No.1, pp.1-7 (2014).
- [2] 青山幹雄, 中根拓也: ReqQA: ソフトウェア要求仕様書品質解析ツールの提案と評価, *情報処理学会論文誌*, Vol.57, No.2, pp.694-706 (2016).
- [3] AUTOSAR Consortium: AUTOSAR - Layered Software Architecture, AUTOSAR Consortium, Technical Report Release 4.2.2 (2015).
- [4] Aurum, A., Petersson, H. and Wohlin, C.: State-of-the-Art: Software Inspections after 25 Years, *J. Software Testing Verification and Reliability*, Vol.12, No.3, pp.133-154 (2002).
- [5] Braun, P. et al.: Guiding Requirements Engineering for Software-Intensive Embedded Systems in the Automotive Industry, *Computer Science - Research and Development*, Vol.29, No.1, pp.21-43, Springer (2014).
- [6] Davis, A. et al.: Identifying and Measuring Quality in a Software Requirements Specification, *Proc. 1st Int'l Software Metrics Symposium*, pp.141-152, IEEE Computer Society (1993).
- [7] DENSO CORPORATION, available from (<http://www.denso.co.jp/ja/>).
- [8] DoD, DI-IPSC-81433A: Data Item Description: Software Requirements Specification (1999).
- [9] Ebert, C. and Jones, C.: Embedded Software: Facts, Figures, and Future, *IEEE Computer*, Vol.42, No.4, pp.42-52 (2009).
- [10] Fagan, M.: Design and Code Inspections to Reduce Errors in Program Development, *IBM Systems Journal*, Vol.15, No.3, pp.182-211 (1976).
- [11] Fanmuy, G. et al.: Requirements Verification in the Industry, *Proc. CSDM 2011*, Springer, pp.145-160 (2011).
- [12] Fanmuy, G. and Foughali, G.: A Survey on Industrial Practices in Requirements Engineering, *INCOSE International Symposium*, Vol.22, No.1, pp.1021-1040 (2012).
- [13] Gorschek, T. and Wohlin, C.: Requirements Abstraction Model, *Requirements Engineering J.*, Vol.11, No.1, pp.79-101 (2006).
- [14] Grimm, K.: Software Technology in an Automotive Company - Major Challenges, *Proc. ICSE 2003*, pp.498-503, IEEE Computer Society (2003).
- [15] IEEE, Std. 830-1998: IEEE Recommended Practice for Software Requirements Specifications, IEEE Computer Society (1998).
- [16] Kantorowitz, E., Guttman, A. and Arzi, L.: The Performance of the N-Fold Requirement Inspection Method, *J. Requirements Engineering*, Vol.2, No.3, pp.152-164 (1997).
- [17] Krisch, J. and Houdek, F.: The Myth of Bad Passive Voice and Weak Words: An Empirical Investigation in the Automotive Industry, *Proc. RE 2015*, pp.344-351, IEEE Computer Society (2015).
- [18] Knight, J.C. and Myers, E.A.: An Improved Inspection Technique, *CACM*, Vol.36, No.11, pp.51-61 (1993).
- [19] Langenfeld, V. et al.: Requirements Defects over a

Project Lifetime: An Empirical Analysis of Defect Data from 5-Year Automotive Project at Bosch, *Proc. REFSQ 2016*, LNCS Vol.9619, pp.145-160, Springer (2016).

- [20] Martin, J. and Tsai, W.T.: N-Fold Inspection: A Requirements Analysis Technique, *CACM*, Vol.33, No.2, pp.225-232 (1990).
- [21] McConnell, S.: The Best Influences on Software Engineering, *IEEE Software*, Vol.17, No.1, pp.10-17 (2000).
- [22] NASA: NASA-STD-2100-91, NASA Software Documentation Standard, NASA Headquarters Software Engineering Program (1991).
- [23] Ott, D.: Defects in Natural Languages Requirements Specifications at Mercedes Benz, *Proc. RE 2012*, pp.291-296, IEEE (2012).
- [24] Pohl, K.: Requirements Engineering, Springer (2010).
- [25] Pretschner, A. et al.: Software Engineering for Automotive Systems: A Roadmap, *Proc. FOSE 2007 (Future of Software Engineering)*, pp.55-71, IEEE Computer Society (2007).
- [26] Robertson, J. and Robertson, S.: Volere Requirements Specification Template Edition 17, Atlantic Systems Guild (2014).
- [27] Robertson, S. and Robertson, J.: *Mastering the Requirements Process*, 2nd ed., Addison-Wesley (2006).
- [28] Saito, S., Takeuchi, M., Yamada, S. and Aoyama, M.: RISDM: A Requirements Inspection Systems Design Methodology, *Proc. RE 2014*, pp.223-232, IEEE Computer Society (2014).
- [29] Takoshima, A. and Aoyama, M.: Assessing the Quality of Software Requirements Specifications for Automotive Software Systems, *Proc. APSEC 2015*, pp.393-400, IEEE CPS (2015).
- [30] VDA QMC Working Group 13/Automotive SIG: Automotive SPICE Process Assessment Model V. 3.0 (July 2015), available from <http://www.automotivespice.com/>.
- [31] Weber, M. and Weisbrod, J.: Requirements Engineering in Automotive Development: Experiences and Challenges, *IEEE Software*, Vol.20, No.1, pp.16-24 (2003).
- [32] Wiegers, K. and Beatty, J.: *Software Requirements*, 3rd ed., Microsoft Press (2013).
- [33] Wilson, W.: Writing Effective Natural Language Requirements Specifications, *J. Defense Software Engineering*, pp.16-19 (1999).

表 A.1 SRS 品質スコアの算出方法

Table A.1 SRS quality score calculation.

付 録

A.1 SRS 品質スコアの計算方法

本付録では表 5 で示したデータセットから SRS_A を例に非正規化スコアと正規化スコアの計算方法について説明する。まず、表 A.1 の見方について説明する。表 A.1 中の青枠で囲まれた部分はインスペクションマトリクスであり、表 2 を時計回りに 90° 回転させたものである。その下の赤枠で囲まれた部分が変換マトリクスである。その右の緑枠で囲まれた部分が SRS_A のインスペクション結果である。インスペクション結果は各行がプロジェクト SRS (SRS_A) の目次項目に関するインスペクション結果を表し、各列がドキュメント品質副特性に関するインスペクション結果を表している。インスペクション結果を表す 3

つの記号のうち●は品質副特性が満たされていること、○は品質副特性が満たされていないこと、ハイフン (-) はインスペクション対象外であることをそれぞれ示している。

A.1.1 非正規化スコアの計算方法

A.1.1.1 品質副特性ビューと品質特性ビュー

非正規化スコアにおける品質副特性ビューの品質スコアはインスペクション結果の各列に対し以下の式 (A.1) を用いて計算する。

$$\text{品質スコア} = \frac{\text{●の数}}{\text{○の数} + \text{●の数}} \times 100[\%] \quad (\text{A.1})$$

機能目的 (C1-2) を例にとると、品質スコアは $5/(9+5) = 36\%$ となる。品質特性ビューの品質スコアも同じく式 (A.1) を用いて計算されるが、計算の対象がインスペクション結果の単一の列ではなく、品質特性を構成するすべての品質副特性の列になる。責任追跡性 (C1) を例にとると、品質スコアの計算対象は開発目的 (C1-1) と機能目的 (C1-2) の2つの列になる。そのため責任追跡性の品質スコアは $5/(10+5) = 33\%$ となる。

A.1.1.2 構造ビュー

構造ビューの品質スコアも品質特性ビューの品質スコアと同じく式 (A.1) を用いて計算されるが、計算の対象がインスペクション結果の列方向 (品質特性単位) ではなく行方向 (目次項目単位) となる。目次項目 1.1 を例にとると、品質スコアは $1/(1+1) = 50\%$ となる。1章全体の品質スコアは $3/(5+3) = 38\%$ となる。

A.1.2 正規化スコアの計算方法

A.1.2.1 品質副特性ビューと品質特性ビュー

機能目的 (C1-2) を例に正規化スコアにおける品質副特性ビューの品質スコアの計算方法を説明する。品質スコアの計算は以下の手順で行う。

(1) インスペクションポイントごとの得点の計算

あるインスペクションポイントの得点を計算するには、変換マトリクスからそのインスペクションポイントに関連する参照 SRS 目次項目に対応するプロジェクト SRS 目次項目をすべて抽出する。続いて抽出したプロジェクト SRS 目次項目のインスペクション結果のうち対象の品質副特性のみに対して式 (A.1) を適用して品質スコアを計算する。

機能目的 (C1-2) の場合、関連する参照 SRS 目次項目は「2.1.7 Operations」, 「2.1.8 Site adaptation requirements」, 「3.2.1.1 Introduction/Purpose of feature」の3つである。そのうち「2.1.7 Operations」, 「2.1.8 Site adaptation requirements」は対応するプロジェクト SRS 目次項目が存在しないことが変換マトリクスから確認できる。そのため、これら2つの参照 SRS 目次項目に関するインスペクションポイントでのスコアは0%となる。残る「3.2.1.1 Introduction/Purpose of feature」に関しては5.1.1から

11.3 まで14のプロジェクト SRS 目次項目が対応していることが分かる。それら14のプロジェクト SRS 目次項目に対して式 (A.1) を適用すると、このインスペクションポイント (ドキュメント品質副特性 (C2-1) と参照 SRS 目次項目 (3.2.1.1) の交点) における品質スコアは $5/(9+5) = 36\%$ となる。

(2) ドキュメント品質副特性に対する品質スコアの計算

手順 (1) から、あるドキュメント品質副特性に関連したインスペクションポイントごとのスコアが求まった。最終的なドキュメント品質副特性の品質スコアはそれらインスペクションポイントごとのスコアに以下の式 (A.2) を用いて計算する。

$$\text{品質スコア} = \frac{\sum \text{インスペクションポイントでのスコア}}{\text{インスペクションポイント数}} \times 100[\%] \quad (\text{A.2})$$

このようにインスペクションポイントでのスコアの合計をインスペクションポイントの数で割っているのは、関連するインスペクションポイントの数によらず品質副特性の品質スコア上限を100%とするためである。機能目的 (C1-2) に式 (A.2) を適用すると品質スコアは $(0+0+36)/3 = 12\%$ となる。

品質特性ビューの品質スコアも同じく式 (A.2) を用いて計算されるが、計算の対象が単一の品質副特性ではなく、品質特性を構成するすべての品質副特性となる。責任追跡性 (C1) を例にとると、品質スコアの計算対象は開発目的 (C1-1) と機能目的 (C1-2) の2つの品質副特性となるため責任追跡性の品質スコアは $(0+0+0+36)/4 = 9\%$ となる。

A.1.2.2 構造ビュー

正規化スコアの構造ビューに対する品質スコアを計算するにはまず変換マトリクスから参照 SRS 目次項目に対応するプロジェクト SRS 目次項目をすべて抽出する。続いて、抽出されたプロジェクト SRS 目次項目に対するインスペクション結果のうち、インスペクションマトリクスから品質スコア計算対象の参照 SRS 目次項目に関連するドキュメント品質副特性に対して式 (A.1) を適用して品質スコアを計算する。

上記の手順を参照 SRS 目次項目「2.1.4 Software interfaces」に適用すると、プロジェクト SRS 目次項目のうち「4.3」と「4.4」が対応目次項目として抽出される。続いてインスペクションマトリクスを確認すると参照 SRS 目次項目「2.1.4 Software interfaces」に関連するインスペクションポイントは多義的でない (C2-2), 参照 (C2-6), 未決定事項 (C3-1), ラベル (C3-2), テンプレート (C3-3) の5つドキュメント品質副特性に対して定義されていることが分かる。そのため、プロジェクト SRS 目次項目のうち「4.3」と「4.4」に対するインスペクション結果のうち、これら5つの品質副特性に限定して式 (A.1) を適用する。そ

の結果, 参照 SRS 目次項目「2.1.4 Software interfaces」の構造ビュー品質スコアは $6/(4+6) = 60\%$ となる.



蛸島 昭之 (正会員)

2004 年 University of Mary Washington Computer Science 学科卒業. 2005 年アルプス電気株式会社入社. キーレスエントリーシステムやタイヤ空気圧監視システムのファームウェア開発に従事. 2009 年株式会社デンソー入社. エアコン関連製品のソフトウェア開発に従事した後, 電子基盤システム開発部にて全社のソフトウェア開発を支援する基盤技術の研究開発を行っている.



青山 幹雄 (正会員)

1980 年岡山大学大学院工学研究科修士課程修了. 同年富士通株式会社入社. 大規模ソフトウェア開発とプロジェクト管理, ソフトウェア工学の実践に従事. 1986~88 年米国イリノイ大学客員研究員. 1995 年 4 月~2001 年 3 月新潟工科大学情報電子工学科教授. 2001 年より南山大学数理情報学部情報通信学科教授, 2009 年より情報理工学部ソフトウェア工学科教授. 博士 (工学). クラウドコンピューティング, サービス指向アーキテクチャ, 自動車組込みソフトウェア等を対象として, 要求工学, ソフトウェアアーキテクチャ技術, ソフトウェア進化の研究・開発と教育・人材育成に取り組む. 著書『要求工学知識体系』(2011 年刊: 共著) ほか多数. IEEE Software, IEEE Transactions on Services Computing 等の編集委員, 本会理事を歴任. 1993 年情報処理学会研究賞受賞. 日本ソフトウェア科学会, 自動車技術会, IEEE, ACM 各会員.