

# 博士論文

## ソフトウェア要求仕様書の品質モデル設計方法論と 自動車ソフトウェアシステムへの応用に関する研究

D2017SE002 蛸島 昭之

指導教員 青山 幹雄

2018年2月

南山大学大学院 理工学研究科 ソフトウェア工学専攻

---

## A Quality Model Design Methodology of Software Requirements Specifications and Its Applications to Automotive Software Systems

D2017SE002 Akiyuki Takoshima

Supervisor Mikio Aoyama

February 2018

Graduate Program in Software Engineering  
Graduate School of Science and Engineering  
Nanzan University

## 要約

ソフトウェア技術の発達により、ソフトウェアに対する社会からの要求が高度化かつ多様化している。自動車ソフトウェア分野では、ソフトウェアの利用目的がハードウェアの制御からソフトウェアでのみ実現できる付加価値の創造へと変化しつつある。そのため、自動車ソフトウェアに対する要求もこれまで以上に高度化かつ多様化している。多くのステークホルダが関わる自動車ソフトウェアの開発では、ステークホルダ間で多様な要求を誤りなく文書化し、共有するためにはソフトウェア要求仕様書（SRS：Software Requirements Specification）の品質が他のドメイン以上に重要となる。

本研究では、ソフトウェア要求仕様書の品質モデルを設計するため、SRS 構成モデルの設計方法、SRS 品質モデルの設計方法、SRS インスペクションの設計方法を提案する。

SRS 構成モデルは、要求の多様性を生み出すドメイン特性に対するステークホルダの関心事に基づいて設計する方法を提案する。さらに、提案方法を実際の自動車ドメインに適用し、自動車ソフトウェアに適した SRS 構成モデルを設計する。

SRS 品質モデルは、自然言語で記述された SRS の読者であるステークホルダのパースペクティブ（立場）における語用論に基づき品質特性を定義することで、これまでの方法を拡張した方法を提案する。SRS 品質モデルでは、定義した品質特性を語用論的な特徴から、SRS の表現の正当性確認と内容の妥当性確認を独立したインスペクションとして実施可能となるよう層別する方法を提案する。

提案方法を用いて設計した SRS 品質モデルに基づき正当性確認と妥当性確認のプロセスを段階的に行う二段階インスペクションプロセスを提案する。正当性確認プロセスを効率的に実施する方法として第三者によるインスペクションの方法を設計する方法を提案する。

提案方法を、実際の自動車ソフトウェアシステムの SRS に適用し、SRS 品質改善と手戻り工数削減について有用性を評価した。その評価に基づき、SRS 品質の設計と向上に対する本研究の技術的貢献について議論する。

以上から、本研究の成果は、ソフトウェア工学の中で、特に、要求工学の発展に次の 3 つの点で貢献するものと言える。

- (1) 従来、一般的に研究されてこなかったドメイン固有の要求仕様書の要求項目をステークホルダ関心事に基づき導出する方法を提案し、ドメイン固有要求仕様書の設計を可能とした。
- (2) 要求仕様書の品質モデルをその対象とする読者のパースペクティブに基づき設計する方法を拡張し、自動車ソフトウェア要求仕様書を対象とする品質モデルを定義した。
- (3) 本研究で定義した要求仕様書の品質モデルに基づき、要求仕様書インスペクション設計方法を設計し、自動車ソフトウェアの異なる製品ドメインへ適用し、その有効性を明らかにした。

## Abstract

As software technologies evolve, the requirements from the society are becoming more and more sophisticated and diverse. In the automotive industry, the purpose of software is shifting to the creation of added values that can be realized only by software. Requirements for automotive software, therefore, are becoming more diverse than ever, like other domains, since new features such as Internet connection services and advanced driver-assistance systems appear. Since the automotive industry has traditionally been organized in a highly vertical manner, automotive software development involves many different stakeholders. Software requirements specification, SRS, plays a significant role in correctly documentation and communication of those diverse requirements among the stakeholders.

In this thesis, the author proposes an SRS construction model design method, an SRS quality model design method, and an SRS inspection design method, in order to design the quality model of SRSs.

The author proposes a method to design SRS construction models based on the stakeholders' concerns with domain characteristics that produce the diversity of requirements. An SRS construction model for automotive software is designed by applying the proposed method.

The thesis proposes an extended method to design a set of quality characteristics based on the understandings by reading the contents from the perspectives of the stakeholders, i.e. the readers. The SRS quality model categorizes the defined quality characteristics, from the linguistic properties, into two groups: the ones that should be measured in the verification process, and the others that should be measured in the validation process.

The author proposes a two-stage inspection method, consisting of clearly separated verification and validation processes, which is based on the proposed SRS quality model. A third-party inspection method is designed as an efficient implementation of the verification process.

The author applied the proposed methods to the actual automotive SRSs and evaluated their effectiveness. The technical contribution of this research work to the design and improvement of SRS quality is discussed based on the evaluation.

# 目次

1	はじめに.....	8
1.1	研究の背景.....	8
1.1.1	自動車ソフトウェアに対する要求の高度化と多様化.....	8
1.1.2	自動車ソフトウェア要求仕様書の品質向上要求の高まり.....	8
1.2	研究の目的.....	9
1.3	論文の構成.....	10
2	研究課題.....	12
2.1	ステークホルダの関心事に基づく SRS 構成モデルの設計方法.....	12
2.2	パースペクティブに基づく SRS 品質モデルの設計方法.....	12
2.3	SRS 品質モデルに基づく自動車 SRS のインスペクション方法.....	12
3	関連研究.....	13
3.1	自動車要求工学.....	13
3.1.1	自動車ソフトウェア開発における要求工学.....	13
3.1.1.1	自動車要求工学の位置づけ.....	13
3.1.1.2	自動車要求工学の課題.....	13
3.1.2	自動車要求工学の実践.....	15
3.1.3	自動車 SRS に関する研究.....	18
3.1.3.1	要求仕様の記述.....	18
3.1.3.2	要求カテゴリ分類の自動化.....	18
3.1.3.3	品質特性と欠陥の分類.....	19
3.2	ソフトウェア要求仕様書の構成モデル.....	19
3.3	ソフトウェア要求仕様書の品質特性.....	21
3.4	ソフトウェア要求仕様書のインスペクション.....	22
3.4.1	インスペクション方法とそのバリエーション.....	22
3.4.2	ドキュメント読解技術と PBR.....	23
3.4.3	インスペクション実施上の課題.....	23
3.4.4	インスペクションの設計技術.....	23
3.4.4.1	RISDM のインスペクションモデル.....	23
3.4.4.2	RISDM によるインスペクションシステムのデザイン.....	24
3.4.4.3	RISDM による RIS のデザイン方法.....	25
3.5	関連研究のまとめ.....	28
3.5.1	自動車要求工学.....	28
3.5.2	SRS 構成モデル.....	28
3.5.3	SRS 品質モデル.....	28
3.5.4	SRS のインスペクション.....	28
4	アプローチ.....	29
4.1	SRS 品質保証のフレームワーク.....	29
4.2	アプローチの全体像.....	29
4.3	ステークホルダの関心事に基づく SRS 構成モデルの設計.....	30

4.4	パースペクティブに基づく SRS 品質モデルの設計.....	30
4.5	SRS 品質モデルに基づく自動車 SRS インスペクション方法.....	30
<b>5</b>	<b>ステークホルダの関心事に基づく SRS 構成モデルの設計.....</b>	<b>31</b>
5.1	SRS のメタモデル.....	31
5.2	ステークホルダの関心事に基づくドメイン固有 SRS 構成モデルの設計プロセス.....	31
5.3	参照 SRS 構成モデルの選択.....	34
5.4	ドメイン特性の展開.....	35
5.4.1	製品ドメインの定義.....	35
5.4.2	ドメイン特性展開表の作成.....	35
5.5	ステークホルダのパースペクティブの展開.....	37
5.5.1	ステークホルダの特定と抽象化.....	37
5.5.2	パースペクティブ展開表の作成.....	37
5.6	ステークホルダのドメイン特性に関する関心事の展開.....	38
5.7	ドメイン特性に関する関心事から SRS 要求項目への展開.....	40
5.8	参照 SRS 構成モデルの拡張.....	41
<b>6</b>	<b>パースペクティブに基づく SRS 品質モデルの設計.....</b>	<b>43</b>
6.1	SRS 品質モデル定義の目的.....	43
6.2	プラグマティック品質特性の必要性.....	43
6.3	プラグマティック品質特性の導出プロセス.....	44
6.3.1	参照 SRS 品質特性の定義.....	44
6.3.2	ステークホルダの特定とパースペクティブの決定.....	45
6.3.3	プラグマティック品質特性の定義.....	45
6.4	SRS の正当性確認と妥当性確認.....	48
<b>7</b>	<b>SRS 品質モデルに基づく自動車 SRS インスペクション方法.....</b>	<b>50</b>
7.1	二段階インスペクションプロセスの提案.....	50
7.1.1	第三者インスペクタの定義.....	50
7.1.2	二段階インスペクションのプロセス.....	50
7.2	第三者インスペクション方法.....	52
7.2.1	第三者インスペクションのメタモデル.....	52
7.2.2	SRS 品質スコアの測定方法.....	52
7.2.2.1	SRS 構成モデル.....	52
7.2.2.2	SRS 品質特性.....	52
7.2.2.3	質問セット.....	53
7.2.2.4	インスペクションマトリクス.....	55
7.2.2.5	変換マトリクス.....	56
7.2.2.6	インスペクションガイドライン.....	56
7.2.2.7	アセスメントレポート.....	56
7.2.3	品質スコアの可視化方法.....	57
7.2.3.1	品質の評価方法と可視化方法.....	57
7.2.3.2	非正規化スコア.....	57
7.2.3.3	正規化スコア.....	57
7.2.3.4	非正規化スコアと正規化スコアの利用方法.....	59

7.2.3.5	要求項目ヒストグラム.....	59
7.2.3.6	スコアヒートマップ.....	59
7.2.4	品質スコアの計算方法.....	60
7.2.4.1	非正規化スコアの計算方法.....	60
7.2.4.1.1	品質副特性ビューと品質特性ビュー.....	60
7.2.4.1.2	構造ビュー.....	60
7.2.4.2	正規化スコアの計算方法.....	61
7.2.4.2.1	品質副特性ビューと品質特性ビュー.....	61
7.2.4.2.2	構造ビュー.....	61
<b>8</b>	<b>提案方法の実際の SRS への適用.....</b>	<b>64</b>
8.1	多様な自動車 SRS への適用性.....	64
8.1.1	適用のコンテキスト.....	64
8.1.2	データセット.....	64
8.1.3	検証結果.....	65
8.2	SRS 品質改善効果の検証.....	66
8.2.1	適用例 1 (ボディー製品ソフトウェア) .....	66
8.2.1.1	検証方法.....	66
8.2.1.2	検証結果.....	66
8.2.2	適用例 2 (パワートレイン製品ソフトウェア) .....	68
8.2.2.1	検証方法.....	68
8.2.2.2	検証結果.....	68
8.2.3	SRS 品質改善効果のまとめ.....	69
8.3	手戻工数削減効果の検証.....	69
8.3.1	検証方法.....	69
8.3.2	検証結果.....	70
<b>9</b>	<b>研究課題に対する評価.....</b>	<b>71</b>
9.1	ステークホルダの関心事に基づく SRS 構成モデルの設計方法.....	71
9.2	パースペクティブに基づく SRS 品質モデルの設計方法.....	71
9.3	SRS 品質モデルに基づく自動車 SRS の第三者インスペクション方法.....	71
<b>10</b>	<b>考察.....</b>	<b>72</b>
10.1	ステークホルダの関心事に基づく SRS 構成モデルの設計方法.....	72
10.2	パースペクティブに基づく SRS 品質モデルの設計方法.....	72
10.3	SRS 品質モデルに基づく自動車 SRS の第三者インスペクション方法.....	72
<b>11</b>	<b>今後の課題.....</b>	<b>73</b>
11.1	ステークホルダの変化への対応.....	73
11.2	インスペクションの対象範囲拡大と独立性の向上.....	73
11.3	設計方法論の他分野への適用.....	73
<b>12</b>	<b>まとめ.....</b>	<b>74</b>
12.1	研究の背景と課題.....	74
12.2	課題解決.....	74
12.3	本研究の貢献.....	74

参考文献..... 76  
研究業績..... 80

# 1 はじめに

## 1.1 研究の背景

### 1.1.1 自動車ソフトウェアに対する要求の高度化と多様化

自動車ソフトウェアは自動車の基本機能である「走る、曲がる、止まる」に関するアナログ制御をデジタル制御へ置き換えることで発展してきたため、自動車ソフトウェアに対する要求はデジタル制御に関する機能要求と品質要求に限定されていた。しかし、近年ではそれらの3つの基本機能に加えて、インターネット上のサービスや車外インフラと「つながる」機能が4つ目の基本機能として普及しつつある。これは、自動車におけるソフトウェアの役割が、これまでの制御のデジタル化から、ソフトウェアでしか実現できない付加価値の創造へとシフトしつつあることを意味する。

自動車におけるソフトウェアの適用領域が拡大することで、自動車ソフトウェアに対する要求が多様化している。例えば、インターネットへの接続によって新たに発生するセキュリティ要求やプライバシー要求が挙げられる。無線通信によるソフトウェア機能のアップデート（SOTA : Software Update Over-The-Air）や自動運転などの発展によって自動車ソフトウェアに対する要求はさらに多様性が高まっていくと予想される。

### 1.1.2 自動車ソフトウェア要求仕様書の品質向上要求の高まり

自動車の機能はモデルチェンジの度に増加する。近年では、それらの新機能の多くはエレクトロニクス技術によって実現されるが、なかでもとりわけソフトウェアが重要な役割を担っている。現在の自動車では、全機能のおよそ半数がソフトウェアによって実現されるまでになっているが、今後もその比率がさらに増加していくことは確実である。このよう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倍の割合で増加している [12]。

このように大規模で複雑な自動車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: Requirements Engineering) の活用が不可欠である [17]。なかでもソフトウェア要求仕様書 (SRS : Software Requirements Specification) のインスペクションがソフトウェアの品質、開発コスト、期間を満たす上で効果があることが知られている [38]。自動車産業は伝統的に垂直統合型の構造をとってきたので [53]、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。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 SRS が極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 [68] では SRS 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活動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 の品質向上に繋がっていないとの報告もある [16]。

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要因として、自動車ソフトウェアの多様性がある。Pretschner らは自動車ソフトウェア工学の調査論文において、製品ドメインをインフォテイメント系、ボディー/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 [53]。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技術が求められるので、異なる組織で担務する開発形態が一般的である。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、開発組織を越えて統一したインスペクションの方法が構築されていない。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。

## 1.2 研究の目的

研究の背景でも述べたように、自動車における新機能の多くはソフトウェアによって実現されており、高度化する多様な要求を扱うことが求められる。ステークホルダ間で多様な要求を誤りなく伝達、共有するためには SRS の品質が重要となる。

国際規格などが提供する SRS 構成モデルは汎用的なため、多様な要求の表現には十分ではない。既存研究によって提案された SRS 品質モデルでは SRS の読者が正しく要求を理解できることを保証できない。さらに、体系化されていないインスペクションでは SRS が構成モデルに従い記述されていることと、記述された内容が SRS 品質モデルの定める品質特性を満たしているかを評価するには不十分である。

以上の理由から、下記の 3 点を本研究の目的とする。

- (1) 多様な要求を表現できる SRS 構成モデルの設計方法を提案し、SRS の品質向上を図る。
- (2) ステークホルダのパースペクティブに基づく SRS 品質モデルを設計し、SRS 読者にとっての理解しやすさの観点から SRS 品質を測定することを可能とし、SRS の品質向上を図る。
- (3) SRS が SRS 構成モデルに基づき記述され、SRS 品質モデルが定める品質特性を満たしているかを評価する体系的なインスペクション方法を設計することで、SRS 品質の測定を可能とする。

## 1.3 論文の構成

本論文の構成を以下に述べる。図 1-1 は論文の構成を示し、図中の矢印は前の章で示された結果を引き継いでいることを意味する。

2 章では、本研究の課題について述べる。

3 章では、研究課題に関連する研究について議論する。

4 章では、本研究の課題を解決するためのアプローチについて述べる。

5 章では、ステークホルダの関心事に基づく SRS 構成モデルの設計について述べる。5 章は (1) 要求項目のメタモデル、(2) SRS 構成モデルの設計方法、(3) 自動車ソフトウェアに適した SRS 構成モデルの 3 つの話題から構成される。

(1) では、多様な要求の構造を明らかにするため、SRS のメタモデルを提案する。(2) では、提案する要求項目のメタモデルに基づき体系的に SRS 構成モデルを設計する方法を提案する。(3) では、提案方法を適用し自動車ソフトウェアに適した SRS 構成モデルを設計する。

6 章では、パースペクティブに基づく SRS 品質モデルの設計について述べる。6 章は (1) ステークホルダのパースペクティブを考慮した品質特性、要求の正当性確認と妥当性確認を層別した SRS 品質モデルの 2 つの話題から構成される。

(1) では、参照する SRS 品質特性に対しステークホルダのパースペクティブを考慮したプラグマティック品質特性を設計する。(2) では、設計した品質特性を語用論的な特性から、SRS の表現の正当性確認と内容の妥当性確認を独立したインスペクションとして実施可能となるよう層別した SRS 品質モデルを設計する。

7 章では、SRS 品質モデルに基づく自動車 SRS インスペクション方法について述べる。7 章は (1) 二段階インスペクションプロセスと、(2) 第三者インスペクション方法の 2 つの話題から構成される。

(1) では、提案する SRS 品質モデルに基づき SRS の表現の正当性確認と内容の妥当性確認を効率的に実施するための二段階インスペクションプロセスを提案する。(2) では、二段階インスペクションプロセスのうち、SRS の表現の正当性確認を体系的に実施するための方法として第三者によるインスペクション方法を設計する。

8 章では、6 章と 7 章で行った提案に対する適用し、提案方法の多様な自動車 SRS への適用性、SRS 品質スコア改善の効果、手戻り工数削減の効果を評価する。

9 章では、8 章での評価に基づき研究成果の研究課題に対する評価について述べ、10 章で関連研究に対する考察を行う。

11 章では、今後の課題を提示し、12 章で本研究のまとめについて述べる。

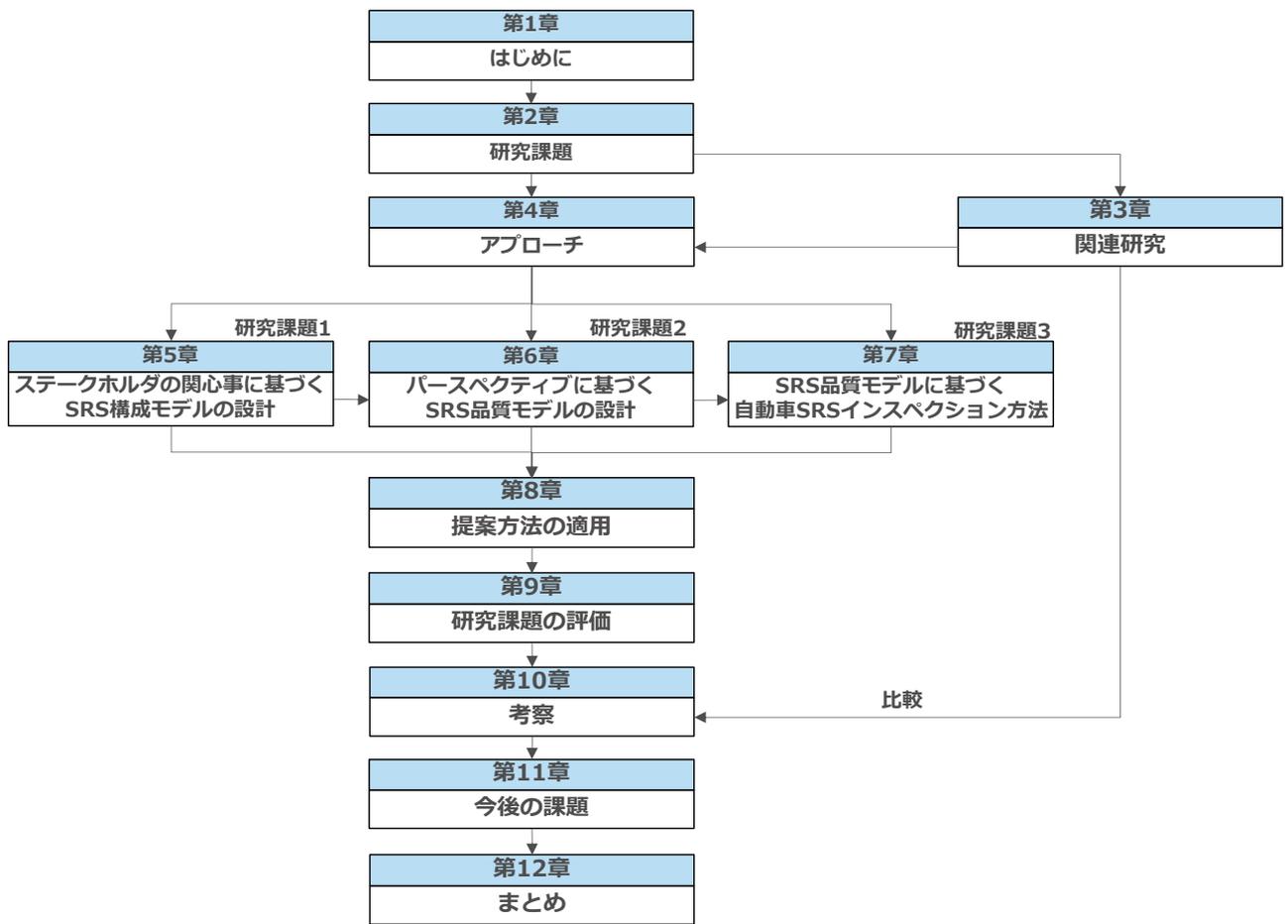


図 1-1 論文の構成

## 2 研究課題

### 2.1 ステークホルダの関心事に基づく SRS 構成モデルの設計方法

SRS の品質を確保するには、SRS に記述されるべき内容の構造、すなわち目次構成、が SRS の作成に先立ち事前に定義されている必要がある。多くの文献によって SRS 構成モデルが提案されているが、それらは汎用的で特定のドメインでそのまま利用するには不適切であるか、特定のドメイン用途に特化しているため他のドメインへは適用できないなどの問題がある。そのため、あるドメインに特有の要求を表現できる SRS 構成モデルを設計することが課題となっている。また、自動車ソフトウェアのように多くのステークホルダが関わるドメインでは、SRS 構成モデルは多様な要求を表現できなければならない。この多様な要求を表現するための要求項目のメタモデルを定義し、体系的にドメイン固有 SRS 構成モデルを設計する方法を提案する。

以上の理由から、本研究では下記の 3 点を具体的な研究課題とする。

#### (1) 要求項目メタモデルの提案

開発対象ソフトウェアのステークホルダがドメインに対して持つ関心事に基づき要求項目のメタモデルを定義し、多様な要求と SRS 構造の対応関係を明らかにする。

#### (2) ステークホルダの関心事に基づくドメイン固有 SRS 構成モデル設計方法の提案

要求項目メタモデルの構造に基づき、体系的に SRS 構成モデルを設計する方法を提案する。

#### (3) 自動車ドメイン固有 SRS 構成モデルの提案

提案方法に基づき、自動車ソフトウェアの多様なステークホルダの要求を表現できる SRS 構成モデルを提案する。

### 2.2 パースペクティブに基づく SRS 品質モデルの設計方法

SRS の品質を向上するには、品質を測定するための品質特性が必要である。自然言語で記述された SRS に対する内容の理解はステークホルダによって異なり、ステークホルダのパースペクティブに基づき品質特性を定義することが課題である。

ソフトウェア開発成果物の品質を確認する手段として正当性確認と妥当性確認 (V&V : Verification and Validation) の考え方が広く受け入れられている。しかしながら、SRS に対する正当性確認と妥当性確認に関する従来の研究は形式仕様言語によって記述された要求仕様が主な対象であり [10]、本研究が対象とする自然言語で記述された SRS の品質確認に適用する方法は未確立である。SRS の正当性確認と妥当性確認を行うための SRS 品質モデルを設計することが課題である。

以上の理由から、本研究では下記の 2 点を具体的な研究課題とする。

#### (1) ステークホルダのパースペクティブを考慮した品質特性の提案

自動車ソフトウェアのステークホルダのパースペクティブにおける語用論に基づいた品質特性としてプラグマティック品質特性 (PQC) を提案する。

#### (2) SRS の表現の正当性確認と内容の妥当性確認を層別した SRS 品質モデルの提案

定義した個々のプラグマティック品質特性を語用論的な特徴から、SRS の表現の正当性確認と内容の妥当性確認を独立したインスペクションとして実施可能となるよう層別した SRS 品質モデルを提案する。

### 2.3 SRS 品質モデルに基づく自動車 SRS のインスペクション方法

SRS の正当性確認と妥当性確認を行うためのインスペクション方法を設計するのが課題である。

上記の理由から、本研究では下記の 2 点を具体的な研究課題とする。

#### (1) 二段階インスペクションプロセスの提案

SRS の体系的な V&V 実施方法として二段階のインスペクションプロセスを提案する。

#### (2) 第三者インスペクション方法の確立

SRS の正当性確認を実施する最適な方法として第三者インスペクションを提案する。

## 3 関連研究

### 3.1 自動車要求工学

#### 3.1.1 自動車ソフトウェア開発における要求工学

##### 3.1.1.1 自動車要求工学の位置づけ

要求工学の研究領域の一つとして、自動車ソフトウェア開発での要求工学における取り組みを明らかにするため、関連研究の調査を行った。

2000年代に入り、自動車ソフトウェアの規模と複雑度の増大により多くの問題が顕在化する中で、一部の自動車メーカーやサプライヤの間で要求工学の重要性が認識されるようになった。WeberらはDaimlerChrysler（現Daimler AG）がそれらの問題に対処するために立ち上げた要求工学研究プロジェクトの活動成果として自動車要求工学について報告している[70]。このWeberらの発表を契機に、自動車ソフトウェア開発分野でも要求工学の重要性が広く認識されるようになった。2004年には、国際要求工学学会の併設ワークショップとしてAutomotive Requirements Engineering Workshop（AuRE 2004）[4]が開催されている。その後も、欧州の自動車メーカー、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。

自動車要求工学という研究領域が正確に定義されているわけではないが、Houdekらは自動車要求工学では大きく分けて、コンセプト要求、システム要求、コンポーネント要求の3つを扱うとしている[19]。それら3つの要求に対する説明を表3-1に示す。本研究では、自動車要求工学の定義としてHoudekらの定義を採用する。

表 3-1 自動車要求工学が取り扱う要求領域[19]

要求領域	説明
コンセプト要求	ビジョンから始まり、製品のイノベーションが形成される。ビジョンはユーザ観点（ユーザ要望）と技術観点（可能性）の両面から詳細化される。プロトタイプが重要な役割を果たし、ここでは「古典的な」要求工学の成果物は作成されない。代わりに、プレゼンテーション、図、ミッションステートメント、高レベルのユーザストーリー、またはペルソナが利用される。
システム要求	要求はシステムレベルの機能と個別のコンポーネントへの割り当てとして文書化される。例えば、アダプティブクルーズコントロールの仕様書は130ページである。システム要求からは、要求のパッケージが導出され、コンポーネントへと引き渡される。
コンポーネント要求	大半のコンポーネントは完成車メーカーではなくサプライヤによって開発される。そのため、開発契約の一部として明示的な仕様が必要となる。製品要求に加え、コンポーネント要求には物流、品質管理、開発責任などのプロセス要求が含まれる。コンポーネント要求をまとめる際には、担当エンジニアは引き渡されたシステム要求パッケージを統合する必要がある。異なるシステムからの要求が相反する場合、担当エンジニアは不明点の確認と調停を行わなければならない。

##### 3.1.1.2 自動車要求工学の課題

これまでの自動車要求工学の研究から、自動車要求工学に特有の課題、および一般の要求工学と共通してはいるが自動車要求工学にとって重要な課題が明らかになっている。以下では、それらの課題について述べる。

###### (1) 自動車要求工学プロセス上の課題

自動車メーカー（OEM と呼ばれる）と多層のサプライヤ（Tier N サプライヤ）からなる垂直統合的な産業構造に起因する自動車要求工学プロセスにおけるコミュニケーションギャップに関する研究がある[2]。Allman

らは、組織構造とコミュニケーションに着目し、自動車メーカ 1 社とサプライヤ 1 社のエンジニア 15 人に対し 14 回の半構造化インタビューを実施した結果、現在の自動車要求工学において、次の 7 つの問題が識別されたとしている。

- 1) 製品知識の不足
- 2) コンテキスト知識の不足
- 3) 抽象レベルの非連続性
- 4) コミュニケーションとフィードバックのチャネルの不足
- 5) 異なる分野にまたがる理解の不足
- 6) 不明確な責任と境界
- 7) 要求を理解しメンテナンスするためのリソースの不足

## (2) 要求記述と管理の課題

Pretschner らは自動車システム開発において今後の取り組みが期待されるソフトウェア工学の研究領域を提案している[53]。その中で、自動車要求工学が解決すべき課題として、Allman らも指摘している自動車メーカとサプライヤ間のコミュニケーションギャップに加え、次の 3 つの課題を指摘している。

### 1) 抽象度の異なる要求を扱う必要性

Houdek の定義にもあるように、自動車要求工学では異なる抽象度の要求を扱う必要があり、そのこと自体が課題となる。本来は高い抽象度で記述すべき要求に抽象度の低い詳細な要求が紛れ込むと、設計者に対し本来は課すはずではない制約を課し、選択可能な設計空間を狭めてしまうことになる。異なる抽象レベルを適切に扱える要求モデルを解明する必要がある。

### 2) 機能要求に加え多様な非機能要求を含む品質要求を扱う必要性

機能要求と品質要求の両方を考慮しながら、要求工程から設計工程へと段階的に進んでいく体系的な方法は依然として大部分が未解決のままである。ドメイン固有の設計と分析のパターンを獲得するのが有望な研究の方向性である。

### 3) 多様性を扱う必要性

自動車は国と地域ごとの法律の違い、グレード間での差別化などのため多くのバリエーションを持ち、それらの差別化の多くはソフトウェアによって実現される。そのため自動車ソフトウェア開発ではソフトウェアプロダクトライン (SPL : Software Product Line) の考え方が導入されている[67]。自動車要求工学では、SPL 開発が取り扱う大量のバリエーションを要求レベルで管理するための手法の研究が必要である。

## (3) 問題領域と解決領域の分離

その他の自動車要求工学上の課題として、Weber らは要求 (問題領域) に留まらず、実装方法 (解決領域) に踏み込むことが必要な状況があることを指摘している[70]。企業情報システムなどの要求定義では、アプリケーションとプラットフォームとの独立性が比較的高いため、実装方法の指定により過度な制約を課さないことが推奨されている。一方、自動車ソフトウェアの開発では異なるサプライヤが開発する複数のコンポーネントの仕様が互いに整合することを保証するなどの目的で、要求のなかに具体的な実装方法を含めることが必要となる場合がある[70]。さらに、Fanmuy らによる自動車関連企業を含む産業界での要求工学実態調査によると、解決方法が要求のなかに紛れ込むことは最も頻繁に発生する要求欠陥であると報告されている[16][17]。

## (4) 課題解決に向けた取り組み

現在の自動車産業界の要求工学プロセスは適切に定義されていないことが多いため、要求の獲得、仕様化、品質保証といった活動はアドホックに行われている[7]。そのため、欧州ではそのような未解決の課題を解決するため、組込みソフトウェアシステム、とりわけ自動車ソフトウェア、における体系的な要求工学とマネジメントのための検証された実施可能なガイドラインを開発することを目的に REMsES (Requirements

Engineering and Management of software-intensive Embedded Systems) [55]プロジェクトが実施され、その成果が報告されている[7].

#### (5) 自動車ソフトウェア開発の発展によって新たに発生する課題

これまで述べた課題に加え、今後の自動車ソフトウェア開発の発展によって新たな課題が発生すると予想される。自動車の新たな発展のトレンドを Mercedes-Benz はその頭文字を取って CASE (Connected, Autonomous, Shared & Service, and Electric Drive) と表現している。Houdek は CASE の発展により発生すると予想される新たな自動車要求工学上の課題を表 3-2 にまとめている[19]。電動運転 (Electric Drive) の領域では従来の要求工学技術が引き続き利用できるとしている。一方、ネットワーク接続性 (Connectivity) とシェアードサービス (Shared & Service) の領域ではバッテリー充電状態のモニタや現在位置の取得などの目的で Web を介した車両へのアクセスが行われるようになるため、IT 分野で培われた Web サービス開発に関する要求工学技術が自動車要求工学にも必要となる。自動運転 (Autonomous) の領域では、事前には想定しきれない走行コンテキストに対し適切な振る舞いを仕様化する必要がある。従来の要求工学技術ではこの問題を解決できないため、ゴールベース要求工学の拡張が必要となるが、これは自動車要求工学のみならず、要求工学全体においても新たな課題となる。

表 3-2 CASE の発展により発生する要求工学上の課題[19]

発展領域	自動車要求工学で重要度が増す技術	要求工学全体で新たに発生する課題
Electric Drive	なし	なし
Connectivity	(webサービス開発に関する) IT分野の要求工学技術	なし
Shared & Service		
Autonomous	ゴールベース要求工学	

### 3.1.2 自動車要求工学の実践

自動車要求工学の実践を明らかにするため関連研究の調査を行った。Fanmuy と Foughali は航空宇宙、自動車、エネルギーの大手企業 8 社、およびコンサルティング企業と学術機関を対象にした産業界における要求工学実践状況のサーベイ結果を報告している[17]。サーベイでは参加者へのアンケートにより、表 3-3 に示す 13 の観点についての実態調査が行われた。

サーベイに参加したすべての産業に共通する特徴として、要求量の増大がある。それらの要求の管理には Microsoft Office (Word と Excel) が最も多く利用され、次いで IBM DOORS の利用が多い。要求は多くの場合に自然言語で記述され、形式仕様言語やモデルの利用は限定的である。要求工学プラクティスの実践状況は企業間のみならず、同一企業内においてもプロジェクトごとに異なっている。違いが大きいプラクティスとしては以下が挙げられている。

- 1) 要求の背景の記録
- 2) 要求の優先順位付け
- 3) 要求のバージョン管理
- 4) 要求仕様書テンプレートの適用
- 5) 要求品質ルール適用

現在の実践状況から、要求の複雑さの解消が主要な未解決課題である。また、仕様書の作成に時間がかかりすぎていることも課題である。要求に関してもっとも頻繁に発生する誤りとして以下が挙げられている。

- 1) 解決方法の内容で要求を表現する
- 2) 曖昧な要求

- 3) 一貫性の欠如
- 4) 完全性の欠如
- 5) 単一の要求に複数の要求が含まれる
- 6) 不正確な要求
- 7) 検証不可能な要求

表 3-3 サーベイ観点[17]

1. Needs, requirements	
	Definition of needs, requirements
	Identification and versioning of requirements
	Number of requirements
	Prioritization of needs, requirements
	Elicitation techniques for needs
	Stakeholders implicated in the collection of needs
	Efficiency of the elicitation of needs
	Requirements management
	Quality rules for requirements
	Specification templates
	Formatting of specification documents
	Capitalization of the justification towards needs, requirements
	Requirements engineering tools
2. Design of solution	
	Derivation of requirements
	Systems hierachy
	Requirements allocation
	System analysis
3. Verification and validation of requirements	
	Most common defects
	Verification/validation of requirements by inspections
	Verification/validation of requirements by the use of models
	Traceability of requirements and tests
	IVVQ improvements
4. Management of changes to requirements	
	Change management
	Change management improvements
5. Configuration management	
	Configuration management
	Tools for configuration management
	Improvement of configuration management
6. Risks management	
7. Requirements management	
8. Customers/suppliers coordination	
	Maturity of suppliers in requirements engineering
	Exchanges/Communication between customers/suppliers
9. Inter-project capitalization	
	Re-use of requirements
	Improvement in the reuse of previous requirements
10. Benefits of a requirements engineering approach	
11. Limits of a requirements engineering approach	
12. Main areas of improvement of requirements engineering	
13. Maturity in requirements engineering	

### 3.1.3 自動車 SRS に関する研究

SRS の品質に関する研究のうち自動車ソフトウェアの SRS の関連研究の調査を行った。関連研究は、次の 3 分野に大別される。

- (1) 要求仕様の記述
- (2) 要求カテゴリの分類
- (3) 品質特性と欠陥の分類

#### 3.1.3.1 要求仕様の記述

Aceituna は組込みシステムの SRS 作成者が考慮すべき関心事 (Concern) として下記の 6 つを定義し、各関心事に対するチェックリストを提案している[1]。

- (1) 利用時のコンテキスト (Operational context)
- (2) 事前事後条件 (Pre and post-condition)
- (3) ステークホルダのビューポイント (Stakeholder viewpoints)
- (4) トレーサビリティ (Traceability)
- (5) 時相的な性質 (Temporal characteristics)
- (6) 変更要求 (Changing requirements)

Post らは BOSCH が開発する自動車電子部品のリアルタイム要求に関する一連の研究を行っている。文献[49]ではリアルタイム要求が違反してはならない特性として *rt-consistency* を定義している。文献[51][52]では Konrad と Cheng によって提案された制限付き英文法[33]の自動車ソフトウェア要求への適用性を評価し、3 つの記述パターンを追加することで自動車の振る舞い要求の記述へ適用可能なことを示している。文献[48]ではパターンシステムに則り自然言語で記述された半形式的な要求を自動でリアルタイム論理式へ変換し、プロパティ違反の自動チェックを行うツールチェーンを提案している。チェック対象のプロパティは一貫性に加え、著者による先行研究の中で定義された *rt-consistency*[49]と *vacuity*[50]の 3 つである。提案するツールチェーンを BOSCH の複数の自動車ソフトウェア開発プロジェクトに適用したフィージビリティスタディの結果、リアルタイム要求の形式化に必要な工数増加は許容可能な範囲であり、分析アルゴリズムの計算量は実現可能なオーダーであった。ツールチェーンの利用による効果には仕様不備の検出および仕様不備が存在しないことの形式的な証明が挙げられている。

#### 3.1.3.2 要求カテゴリ分類の自動化

自動車 SRS は大規模で複雑なためレビューによって品質を確保するには多大な工数を要する。とりわけ SRS 内の複数の章節や複数の関連するドキュメントに分散した要求の中から一貫性と完全性に関する欠陥を検出するのは困難である。Ott は要求のカテゴリを自動で分類するアルゴリズムとツールを開発し Mercedes-Benz に適用した結果、レビュープロセスの改善効果を報告している[44]。一方、課題としては精度の高い分類器 (classifier) の生成には質の高い学習データが必要であることが挙げられている。その後、Ott と Raschke によって分類の自動化による効果の追試が行われている。この結果から、要求カテゴリを事前に分類しておくことで、レビューによる一貫性と完全性の違反の検出効率が高まることが確認されている[46]。

Knauss と Ott は要求カテゴリの完全な自動分類と半自動分類の効率比較を行っている[31]。完全自動のアプローチは要求カテゴリの分類自体は高速に実行できるが誤判定率が高く、その後に人手による確認作業に多くの工数を要するため、分類速度自体は若干劣る半自動アプローチの方がトータルでは効率が高いと結論付けている。

要求カテゴリ分類の自動化にはレビュー効率の向上以外にも多くの応用が考えられる。その一つとして、Ott と Houdek は、複雑度が増す法規と自動車ソフトウェア要求との間の一貫性の検証に要求カテゴリの自動分類技術を適用することを提案している[45]。適用結果から、手で要求カテゴリを分類した一つの SRS を学習デ

ータとして利用することで、十分な精度の分類器が生成できることが報告されている。

### 3.1.3.3 品質特性と欠陥の分類

Ott らは Mercedes-Benz での乗用車向けボディー製品の 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 [43]。

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、品質特性のうち正当性と完全性の欠如が要求欠陥の 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 [36]。

自然言語による要求仕様記述に関する多くの文献では受動態 (passive voice) と弱い言葉 (weak word) の使用が要求の読みにくさと曖昧性を引き起こすとされている。Mercedes-Benz での乗用車ソフトウェア開発環境においては受動態と弱い言葉が頻繁に使われているため、Krisch らはそれらが実際に問題を引き起こすのかを実証的に調査した結果を報告している [34]。調査結果によると、処理の主体であるアクタの記述が欠けている場合であっても受動態の利用による問題の発生はほぼ皆無であった。一方、弱い言葉を含む要求のうち約 12% は問題があるとの結論が得られている。

自動車 SRS に関し、要求仕様の記述、要求カテゴリの分類、品質特性と欠陥分類の分野で多くの研究がなされていることが分かった。しかし、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。

## 3.2 ソフトウェア要求仕様書の構成モデル

ソフトウェア要求仕様書 (SRS) の構成モデルとしてもっとも多く参照されているものの 1 つに国際規格の IEEE 830-1998 [21] がある。しかし、IEEE 830 はすでに廃版となっており後継規格の ISO/IEEE 29148-2011 によって置き換えられている [27]。SRS 構成モデルを定めた組織標準として、欧州宇宙機関 (European Space Agency) の定めた ESA software engineering standards (ESAPSS-05-0 Issue 2) [13]、アメリカ航空宇宙局 (NASA: National Aeronautics and Space Administration) が定めた NASA Software Documentation Standard (NASA-Std-2100-91) [41]、アメリカ国防総省 (United States Department of Defense) が定めた Software Requirements Specification (DI-IPSC-81433A) [11]、などがある。

標準規格以外では、Robertson らがコンサルティング経験から得た知見をもとに Volere Requirements Specification Template を提案している [56][57]。Wiegers らも著書 [71] の中で SRS のテンプレートを提案している。また、国内では情報サービス産業協会 (JISA) の要求工学 WG により編纂された要求工学知識体系 (REBOK) が SRS テンプレートを提案している [29]。著者によるこれらの SRS 構成モデル間の対応関係を表 3-4 に示す。

国際規格や組織標準として提供されている複数のソフトウェア要求仕様書テンプレートの中から、自組織の標準テンプレートを作成するための適切な参照テンプレートを選択する基準を定義し、5 つのテンプレートに対する評価を行った研究 [18] や、医療機器ソフトウェアに特化したソフトウェア要求仕様書 [69] の提案など、要求仕様書構成モデルの研究がある。しかし、参照テンプレートを拡張し、自動車などの特定ドメインに適した SRS 構成モデルを設計する方法は確立していない。

表 3-4 SRS 構成モデル間の対応関係

ISO/IEC/IEEE 29148:2011	REBOK	IEEE Std. 830-1998	NASA_DID_P200	DI-IPSC-81433A	ソフトウェア要求 第3版
1. Introduction	1. はじめに	1. Introduction	1.0 Introduction		1. はじめに
1.1 Purpose	1.1 ソフトウェアの目的			1.2 System overview	
1.2 Scope	1.2 ソフトウェアの適用範囲	1.2 Scope			1.3 プロジェクトスコープ
1.3 Product overview	2. 概要	2. Overall description			2. 概説
1.3.1 Product perspective	2.1 ソフトウェアの展望	2.1 Product Perspective			2.1 プロダクトの背景
		2.1.1 System interfaces			
		2.1.2 User interfaces			5.1 ユーザーインターフェイス
		2.1.3 Hardware interfaces			5.3 ハードウェアインターフェイス
		2.1.4 Software interfaces			5.2 ソフトウェアインターフェイス
		2.1.5 Communications interfaces			5.4 通信インターフェイス
		2.1.6 Memory constraints			
		2.1.7 Operations			
		2.1.8 Site adaptation requirements	5.6 Site Adaptation	3.6 Adaptation requirements	
1.3.2 Product functions	2.2 ソフトウェアの機能概要	2.2 Product functions			
1.3.3 User characteristics	2.3 ユーザ特性	2.3 User characteristics			
1.3.4 Limitations	2.4 制約	2.4 Constraints	5.5 Implementation Constraints		
? Apportioning of requirements		2.6 Apportioning of requirements	7.0 Partitioning for Phased Delivery	3.18 Precedence and criticality of requirements	
1.4 Definitions	1.3 ソフトウェアで用いている定義、用語、略語	1.3 Definitions, acronyms, and abbreviations	9.0 Glossary		付録A：用語集
2. References	1.4 ソフトウェアと関連する資料	1.4 References	2.0 Related Documentation	2. Referenced documents	1.4 参考文献
	1.5 ソフトウェアの概要	1.5 Overview		1.3 Document overview	
3. Specific requirements	3. 詳細要求	3. Specific Requirements			
3.1 External interfaces	3.2 外部インターフェイス要求	3.1 External interfaces	4.0 External Interface Requirements	3.3 CSCI external interface requirements	5. 外部インターフェイス要求
3.2 Functions	3.3 ソフトウェア機能	3.2 Functions		3.2 CSCI capability requirements	3. システムフィーチャ
3.3 Usability Requirements					
3.4 Performance requirements	3.4 性能要求	3.3 Performance requirements	5.2 Performance and Quality Engineering Requirements		6.2 性能
3.5 Logical database requirements	3.4 論理データベース要求	3.4 Logical database requirements		3.5 CSCI internal data requirements	
3.6 Design constraints	3.5 設計制約	3.5 Design constraints	5.5 Implementation Constraints	3.12 Design and implementation constraints	2.4 設計と実装の制約条件
		3.5.1 Standards compliance			
3.7 Software system attributes	3.4 ソフトウェア品質要求	3.6 Software system attributes		3.11 Software quality factors	6. 品質属性
		3.6.1 Reliability			
		3.6.2 Availability			
		3.6.3 Security	5.4 Security and Privacy Requirements	3.8 Security and privacy requirements	6.3 セキュリティ
		3.6.4 Maintainability			
		3.6.5 Portability			
3.8 Supporting information		4. Supporting information			
4. Verification				4. Qualification provisions	
5. Appendices		4.2 Appendixes	11.0 Appendices	Appendices	
5.1 Assumptions and dependencies	2.5 前提条件と依存	2.5 Assumptions and dependencies			2.5 前提条件と依存関係
5.2 Acronyms and abbreviations			8.0 Abbreviations and Acronyms		
		1.1 Purpose		1.3 Document overview	1.1 目的
		4.1 Table of contents and index			
				1. Scope	
				1.1 Identification	
					1.2 文書の表記法
	3.1 システム特性 (System mode)			3. Requirements	
				3.1 Required states and modes	
			3.0 Requirements Approach and Tradeoffs	3.2 CSCI Requirements	
			5.0 Requirements Specification	3.4 CSCI internal interface requirements	
			5.1 Process and Data Requirements		4. データ要求
					4.1 論理データモデル
					4.2 データディクショナリ
					4.3 レポート
					4.4 データの獲得、整合性、保持、廃棄
			5.3 Safety Requirements	3.7 Safety requirements	6.4 安全性
			5.7 Design Goals		2.2 ユーザクラスと特性
				3.9 CSCI environment requirements	2.3 稼働環境
				3.10 Computer resource requirements	6.1 ユーザリティ
				3.13 Personnel-related requirements	6.x [その他]
				3.14 Training-related requirements	7. 国際化要求と現地化要求
				3.15 Logistics-related requirements	8. その他の要求
				3.16 Other requirements	付録B：分析モデル
				3.17 Packaging requirements	
			6.0 Traceability to Parents Design	5. Requirements Traceability	
			10.0 Notes	6. Notes	

### 3.3 ソフトウェア要求仕様書の品質特性

SRS 品質特性として、IEEE 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。後継規格である ISO/IEC/IEEE 29148-2011 では IEEE 830-1998 で定義された品質特性の一部が新たな品質特性に置き換わっている。また、各品質特性が単一の要求の特性を表すのか、要求の集合の特性を表すのかが明示されるようになった。INCOSE (The International Council on Systems Engineering) は実務者向けの要求記述ガイド (Guide for Writing Requirements) [28] を発行しており、その中で網羅的な品質特性のリストを提示している。著者による IEEE 830-1998, ISO/IEC/IEEE 29148-2011, INCOSE 要求記述ガイド間での品質特性の対応関係を表 3-5 に示す。また、SRS の品質ではないが、SRS を基に作成されたソフトウェアプロダクトの品質特性に関する国際規格として ISO/IEC 25000 シリーズ (通称, SQuaRE) がある [25][26]。SRS では開発するソフトウェアに対する品質要求として SQuaRE で定義されたソフトウェア品質特性を参照することができる。

表 3-5 品質特性の対応関係

ISO 29148	IEEE 830	INCOSE
完全性 (個別) Complete	完全性 Complete	完全性 (個別) COMPLETE
完全性 (集合) Complete	完全性 Complete	完全性 (集合) COMPLETE
実現可能性 (個別) Feasible	N/A	実現可能性 (個別) FEASIBLE
実現可能性 (集合) Affordable	N/A	実現可能性 (集合) FEASIBLE
限定性 (集合) Bounded	正当性 Correct	N/A
必要性 (個別) Necessary	順位付け Ranked for importance and/or stability	必要性 NECESSARY
実装独立性 (個別) Implementation Free	N/A	N/A
追跡可能性 (個別) Traceable	追跡可能性 Traceable	N/A
無曖昧性 (個別) Unambiguous	無曖昧性 Unambiguous	無曖昧性 UNAMBIGUOUS
検証可能性 (個別) Verifiable	検証可能性 Verifiable	検証可能性 VERIFIABLE
一貫性 (個別) Consistent	一貫性 Consistent	N/A
一貫性 (集合) Consistent	N/A	一貫性① CONSISTENT
単独性 (個別) Singular	N/A	単独性 SINGULAR
N/A	変更容易性 Modifiable	N/A
N/A	N/A	理解可能性 COMPREHENSIBLE
N/A	N/A	抽象度の妥当性 APPROPRIATE
N/A	N/A	妥当性の検証可能性 ABLE TO BE VALIDATED
N/A	N/A	正確性 CORRECT
N/A	N/A	適合性 CONFORMING

上記の文献のように複数の品質特性を列挙した文献は多くあるが、それらは経験的に重要と思われる特性を列挙したものであり、有益ではあるが論理的な裏付けはなされていない。そのような批判から、Lindland らは言語学的概念 (linguistic concept) に着目し、要求仕様書品質を体系的に説明するためのフレームワークを提案している[37]。Lindland らによるフレームワークの提案と同時期に、Pohl は仕様化 (specification)、表現 (representation)、合意 (agreement) の 3 次元からなる要求工学プロセスのフレームワークを提案している[47]。Krogstie らは Lindland らと Pohl によって別々に提案されたフレームワークを統合することで新たな要求仕様書品質のフレームワークを提案している[35]。Fabbrini らは Krogstie らのフレームワークをさらに拡張することで、自然言語で記述された要求の品質モデルを提案している[14]。そのモデルの中では、シンタクティック品質、セマンティック品質、プラグマティック品質の三つの品質タイプと、品質タイプ間の関係が定義されている (図 3-1)。この順序関係は、組織内での要求品質の改善プログラムを展開する際はシンタクティック品質の改善から始め、より上位の品質へ向かって取り組んでいく必要があることを示唆している。そのような要求品質の改善プログラムを実施するには、品質特性の測定と評価が行えることが望ましい。しかしながら、既存研究では品質特性と品質タイプの概念的な定義に留まっており、具体的な品質特性の測定と評価方法の提示は依然未解決の課題である。



図 3-1 自然言語で記述された要求の品質モデル[14]

### 3.4 ソフトウェア要求仕様書のインスペクション

ソフトウェア開発成果物の品質を確認する手段としてインスペクションが広く用いられている。IEEE 1028-2008[23]では具体的な確認方法としてマネジメントレビュー、テクニカルレビュー、インスペクション、ウォークスルー、オーディットの 5 つが定義されている。IEEE 1028-2008 では品質確認の対象成果物として SRS を含む 39 項目が列挙されている。ISO/IEC 20246-2017[24]では IEEE 1028-2008 で定義された 5 つの方法のうちテクニカルレビュー、インスペクション、ウォークスルーについて、成果物やドキュメントの読解技術などについて解説している。IEEE 1028-2008 で定義された技術のなかではインスペクションに関する研究が最も活発に行われている。

#### 3.4.1 インスペクション方法とそのバリエーション

Fagan によるインスペクション (Fagan Inspection) [15]の提案以来、その改良に関する多くの提案がある [5]。代表的な拡張としては N 重インスペクションと段階的インスペクションが挙げられる。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研究がある [40]。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もっとも強く依存するという結果も得られている [30]。そのため複数ドメインにまたがってインスペクションする方法としては不十分である。

段階的インスペクション[32]は複数の段階に分けてインスペクションを行う方法である。各段階では特定の特性、例えば移植性、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。しかし、正当性確認と妥当性確認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。

### 3.4.2 ドキュメント読解技術とPBR

インスペクションを行うためにドキュメントを読解する技術についても研究が行われている。ISO/IEC 20246-2017ではアドホックレビュー、チェックリストベースレビュー、シナリオベースレビュー、パースペクティブベースリーディング (PBR)、ロールベースレビューの4つの読解技術が提示されている。これらの技術の中ではPBRがもっとも効率的かつ効果的に欠陥を検出できると報告されている[61]。

PBRは要求ドキュメントの読解技術としてBasiliらによって開発された[6]。PBRでは異なるパースペクティブ(立場)に基づいたチェック項目を用意し、各インスペクタはユーザ、設計者、テストなどのパースペクティブに立って対象のドキュメントを読解する。複数のインスペクタが異なる観点からレビューすることから、指摘の重複を減らしつつ網羅的な欠陥の検出が期待できる[74]。PBRの効果に対する追試が行われ、Basiliらの報告[6]と同等の効果が学生を対象とした実験で確認されている[8][39]。一方、パースペクティブの違いによる検出欠陥の数と種類に統計的に有意な差は認められないという追試結果もある[39]。このように追試結果に再現性がない要因として、自然言語で記述された要求ドキュメントの読解による欠陥の検出結果は、インスペクタ個人の能力に大きく依存するためと考えられる。

### 3.4.3 インスペクション実施上の課題

インスペクションは適切に実施されれば欠陥検出に大きな効果を発揮する。一方、産業界での実践状況調査では実施回数に対して十分な欠陥検出が行われていないという報告がある[16][17]。その調査によると、インスペクションが適切な効果をあげられていない原因として、事前に定められた要求品質の基準が正しく適用されている組織は全体の15%に過ぎないことが指摘されている。また、インスペクションの成果が経験豊富なインスペクタの参加の有無に大きく依存することも指摘されている。インスペクションが期待される効果をあげられていない別の要因として、国際規格で述べられているベストプラクティスは一般的すぎるため、実際のソフトウェア開発プロジェクトへは直接適用できず、対象とする問題に合わせた詳細化とテーラリングが必要となることを指摘する研究もある[60]。

### 3.4.4 インスペクションの設計技術

上記の課題を解決するため、SaitoらはSRSのインスペクションを設計するための方法論(RISDM: Requirements Inspection System Design Methodology)を提案している[59]。以下では、文献[59]に示されている方法論の要点について記述する。

#### 3.4.4.1 RISDMのインスペクションモデル

RISDMでデザインする対象のインスペクションは6つの基本要素からなる図3-2のメタモデルで定義される。インスペクションのデザインとは、このメタモデルに基づき、それら6つの構成要素を具体化することである。

- (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)
- (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

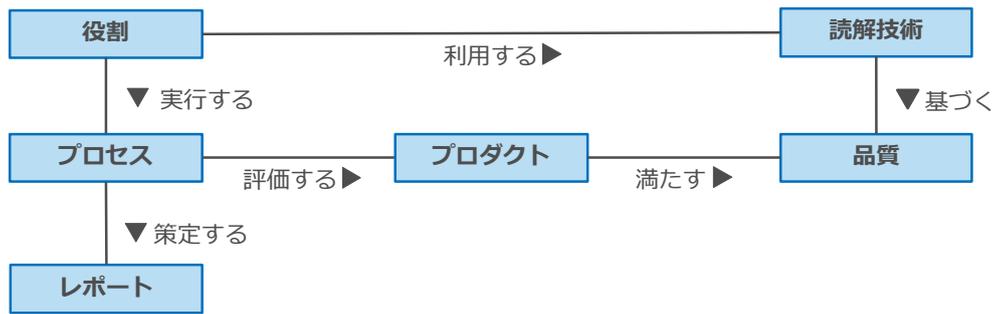


図 3-2 インスペクションメタモデル[59]

### 3.4.4.2 RISDMによるインスペクションシステムのデザイン

RISDM は図 3-2 のメタモデルにおけるプロダクトを 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。メタモデルから具体化された SRS のインスペクションの活動は要求インスペクションシステム (RIS : Requirements Inspection System) と呼ばれる。文献[59]では RIS をデザインする手順について解説されている。ただし、図 3-3 に示すように、ソフトウェア開発プロジェクトからは独立した第三者が SRS をインスペクションすることが想定されているため、基本要素の役割 (要求アナリストとインスペクタ) 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) は事前に規定されている。よって、4 つの基本要素 (プロダクト, 品質, 読解技術, レポート) のデザイン手順についてのみ解説が行われている。

図 3-4 は RISDM によるインスペクションデザインモデルとその関連要素を示している。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となっている。文献[59]では、それら 4 つの基本要素とその関連要素の詳細について述べられている。

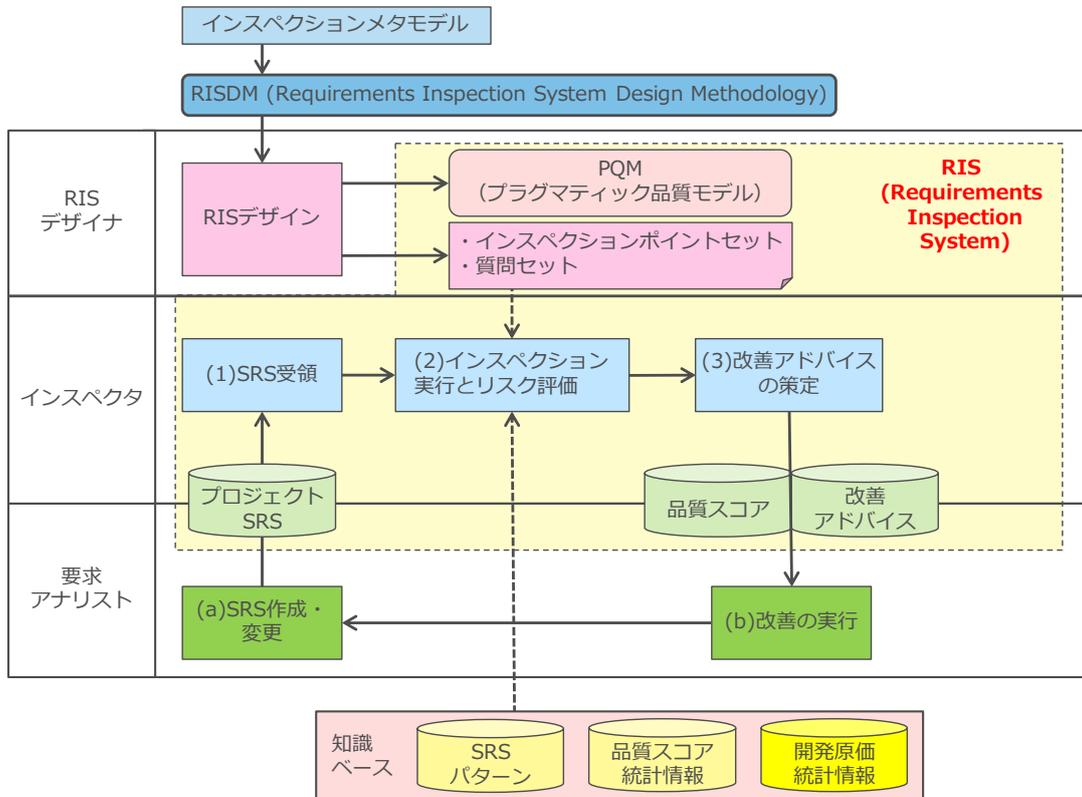


図 3-3 RISDM と要求インスペクションシステム (RIS) [59]

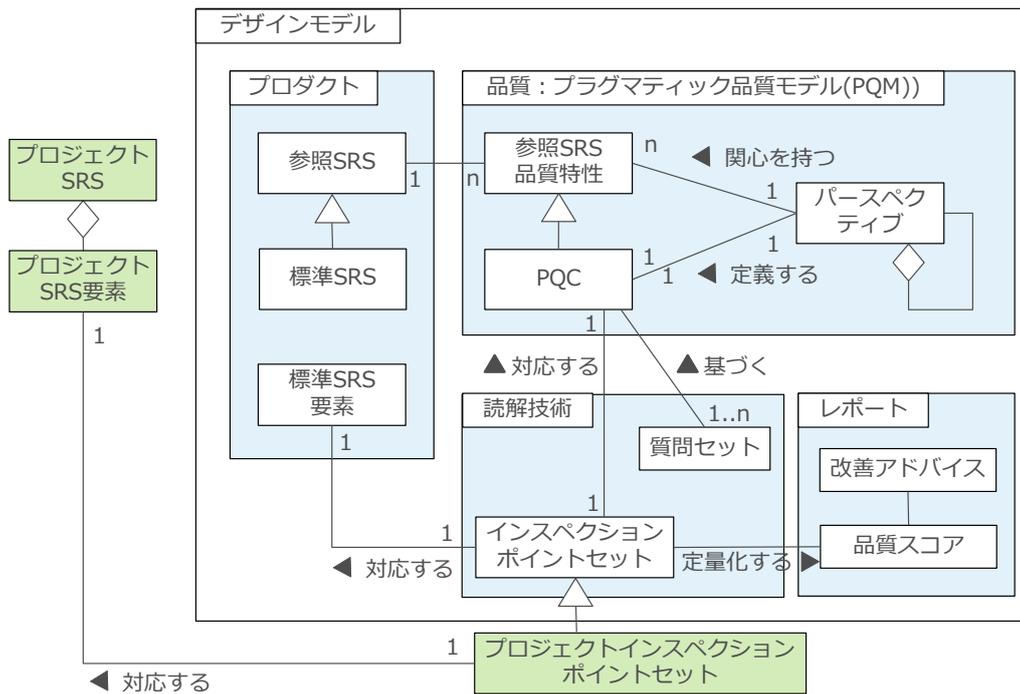


図 3-4 RIS のデザインモデルと関連要素[59]

### 3.4.4.3 RISDMによるRISのデザイン方法

RISDMによるインスペクションのデザインプロセスは6つのプロセスからなる。図3-5は各プロセスの入出力となる成果物を示している。以下では、各プロセスの詳細と出力成果物の例について述べる。

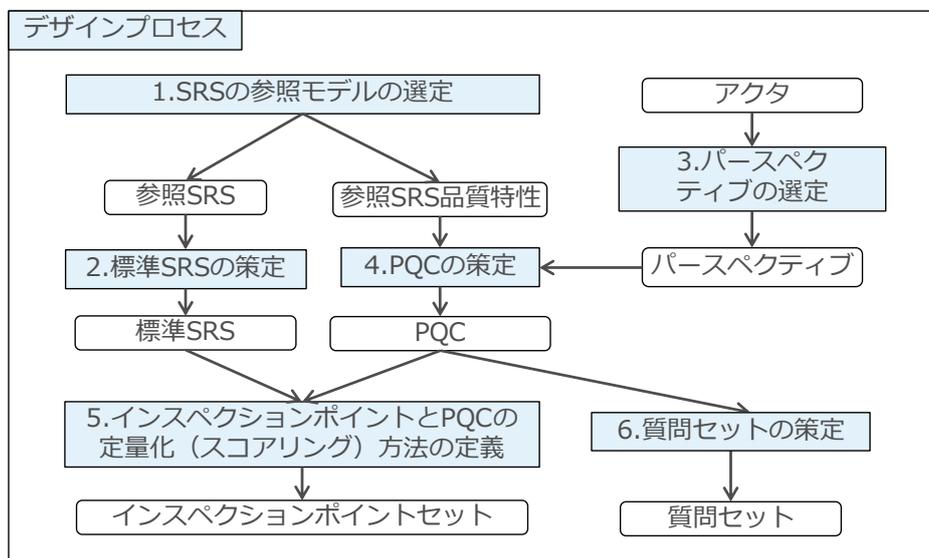


図 3-5 RIS のデザインプロセスと関連プロセス[59]

#### (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ための基準となるSRSの構造（参照SRS）と品質特性（参照SRS品質特性）である。参照モデルにはIEEE 830などを選定する。構造と品質特性にはそれぞれIEEE 830とISO/IEC 25010といった異なる規格を選定することでも

きるとされている。しかし、文献[59]では何を基準に参照モデルを選定すべきかという指針は示されていない。

(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。文献[59]では参照 SRS から標準 SRS を導出するための具体的な手順や指針は示されていない。

(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を二段階で実施する。初めに SRS の読者（アクタ）を定義し、次いで読者の SRS に対するパースペクティブを選定する。パースペクティブの内容は必要に応じて詳細化や分解を行ってもよい。文献[59]では4つのアクタとそこから導出されたパースペクティブが示されている。また、導出された4つのパースペクティブの詳細化と分解を行った例も示されている（図 3-6）。

(4) PQC の策定

前プロセスで選定された各パースペクティブに対してプラグマティック品質（PQC）を策定する。文献[59]で示された PQC の策定結果は図 3-6 に示す通りである。

(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

SRS のすべての構成要素に対してすべての PQC を評価（インспекション）することは効果的ではないため、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。文献[59]で示されたインспекションポイントセットの作成結果は図 3-7 の通りである。ここでは、“X”が SRS の各構成要素に対してインспекションを実施すべき PQC の対応関係を表している。

(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 が満たされているかを問いかける質問の形式で策定する。文献[59]で示された質問セットの策定結果は図 3-8 に示す通りである。

このように、RISDM では標準 SRS およびそこからテーラリングされたプロジェクト SRS をインспекションの対象としている。しかし、差分開発が主流の自動車ソフトウェア開発では、新製品の SRS が新規に作成されるのは稀である。新製品の SRS は、以前の製品の SRS に対し従来の機能要求からの変更点の反映と、新たな機能要求の追記によって作成されるのが一般的である。そのため、標準 SRS を更新することは容易ではない。このように多様な製品ドメインを持ち、製品ドメインごとに形式の異なる多様な自動車 SRS をインспекションするには RISDM 手法は十分ではない [64][65][66]。

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	の重 ラ要 度ク と付 安 け 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に記述されるすべての要求は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すべて存在する
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
	標準機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 3-6 パースペクティブとプラグマティック品質特性[59]

PQC		標準SRSの目次			
ID	名称	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	×			
C2	記述項目網羅性	×	×	×	×
C3	テンプレート使用	×	×	×	
C4	標準記法使用		×		
C5	用語定義				×
C6	識別子の付与	×	×		
C7	一意識別性	×	×		×

図 3-7 インспекションポイントセット[59]

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 3-8 質問セット[59]

## 3.5 関連研究のまとめ

### 3.5.1 自動車要求工学

自動車要求工学が取り扱う要求領域としてコンセプト要求，システム要求，コンポーネント要求の3つの領域があることが明らかになった。本研究では，そのうちのコンポーネント要求を主な対象とする。

自動車 SRS に関し，要求仕様の記述，要求カテゴリの分類，品質特性と欠陥分類の分野で多くの研究がなされていることが分かった。しかし，自動車 SRS が表現すべき要求を定めた記述モデル，自動車 SRS の品質を測定するためのモデル，そして作成された SRS がそれら2つのモデルに適合していることを評価するためのインスペクション方法が確立していないという問題がある。これらの問題の詳細と，問題を解決のための研究課題については以下の節で詳述する。

### 3.5.2 SRS 構成モデル

SRS 構成モデルに関する多くの提案がなされている。それら既存のモデルの中から特定ドメインに特化した SRS 構成モデルを設計する際の参照モデルを選択するため評価基準を提案する研究もある。しかし，参照モデルからドメイン固有の派生モデルを設計するための具体的な方法は確立していない。この問題を解決するため，本研究では 2.1 に挙げた研究課題に取り組む。

### 3.5.3 SRS 品質モデル

SRS の品質特性に関する多くの提案がなされている。それらの品質特性は経験に基づく提案であり，有益ではあるが論理的な裏付けがないとして，自然言語で記述された SRS の品質を体系的に説明するための品質モデルを提案する研究もある。しかし，既存研究では品質モデルの概念的な定義に留まっており，SRS の品質を定量的に評価できないという問題がある。また，既存の品質モデルでは正当性確認と妥当性確認の違いが考慮されていないという問題点もある。これらの問題を解決するため，本研究では 2.2 に挙げた研究課題に取り組む。

### 3.5.4 SRS のインスペクション

インスペクションの実施方法に関する多くの提案がなされている。しかし，ソフトウェア開発成果物の品質確認として確立した正当性確認と妥当性確認のフレームワークを SRS のインスペクションに適用するための方法が未確立という問題がある。

産業界でのインスペクション実践状況調査によると，実施工数に対し十分な欠陥検出が行われていないという報告がある。インスペクションが期待される効果をあげられていない原因として，国際規格で述べられているベストプラクティスは一般的すぎるため，実際のソフトウェア開発プロジェクトへ適用するには実用性に欠けることを指摘する研究がある。このような問題を解決するため，SRS のインスペクションを設計する方法論 (RISDM) が提案されている。しかし，企業情報システムの SRS を対象とする RISDM には，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しなければならない自動車ソフトウェアのような分野へそのまま適用することはできないという問題がある。

上記の問題を解決するため，本研究では 2.3 に挙げた研究課題に取り組む。

## 4 アプローチ

### 4.1 SRS 品質保証のフレームワーク

ソフトウェア開発成果物の一般的な品質保証では、作成された成果物（プロダクト）が事前に定義された品質特性や制約（プロパティ）を満たしているかを品質保証プロセスによって確認するアプローチが用いられる。同様のアプローチを SRS の品質保証に適用する場合、次にあげる 3 つの要素が必要となる。

- (1) SRS（プロダクト）が持つべき要求項目を定義した SRS 構成モデル
- (2) SRS が満たすべき品質特性（プロパティ）を定義した SRS 品質モデル
- (3) SRS のインスタンスが構成モデルと品質モデルに適合していることを確認できるインスペクション方法（プロセス）

これら 3 つの要素間の関係を SRS 品質保証のフレームワークとして図 4-1 に表す。モデルのレベルでは、SRS の構成要素と満たすべき品質特性がそれぞれ SRS 構成モデルと SRS 品質モデルによって定義される。ソフトウェア開発のために実際に作成される SRS は SRS 構成モデルのインスタンスと考えることができる。同様に、この SRS の品質は SRS 品質モデルのインスタンスと考えられる。インスペクションは、SRS インスタンスがこれら二つのモデルに適合しているかを確認するためのプロセスとなる。

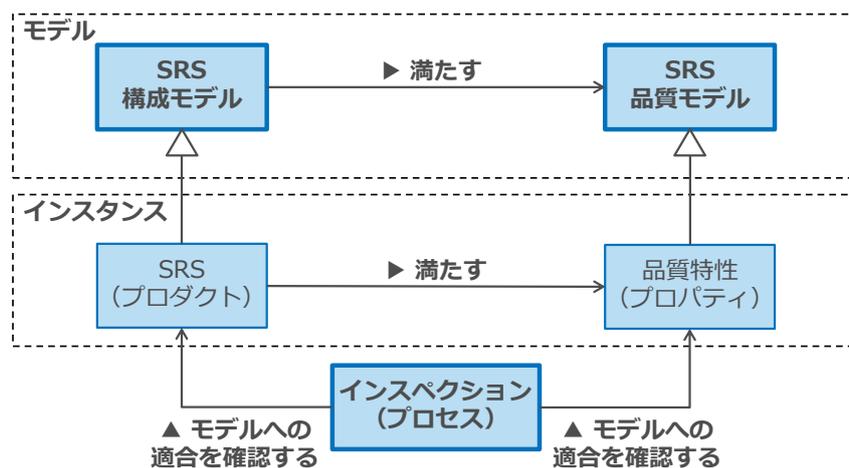


図 4-1 SRS 品質保証のフレームワーク

### 4.2 アプローチの全体像

上述の品質保証の考え方に基づき本研究のアプローチをゴール指向モデルで表現したのが図 4-2 である。はじめに、本研究の目的である SRS 品質モデルの設計をトップの戦略ゴールとして設定する。続いて、戦略ゴールを達成するための戦術ゴールには、SRS の要求項目が定義されていること、SRS が満たすべき品質基準が定義されていること、作成された SRS が要求項目と品質基準の定義を満たしているかを評価するための基準が定義されていることを設定する。これらの戦術ゴールを達成するための具体的なタスクとして、図 4-1 に示す SRS 構成モデル、SRS 品質モデル、SRS のインスペクションを設計する。これらのタスクを実施することで、最終の戦略ゴールである SRS 品質モデルの設計が達成される。以下では、各タスクの詳細について説明する。

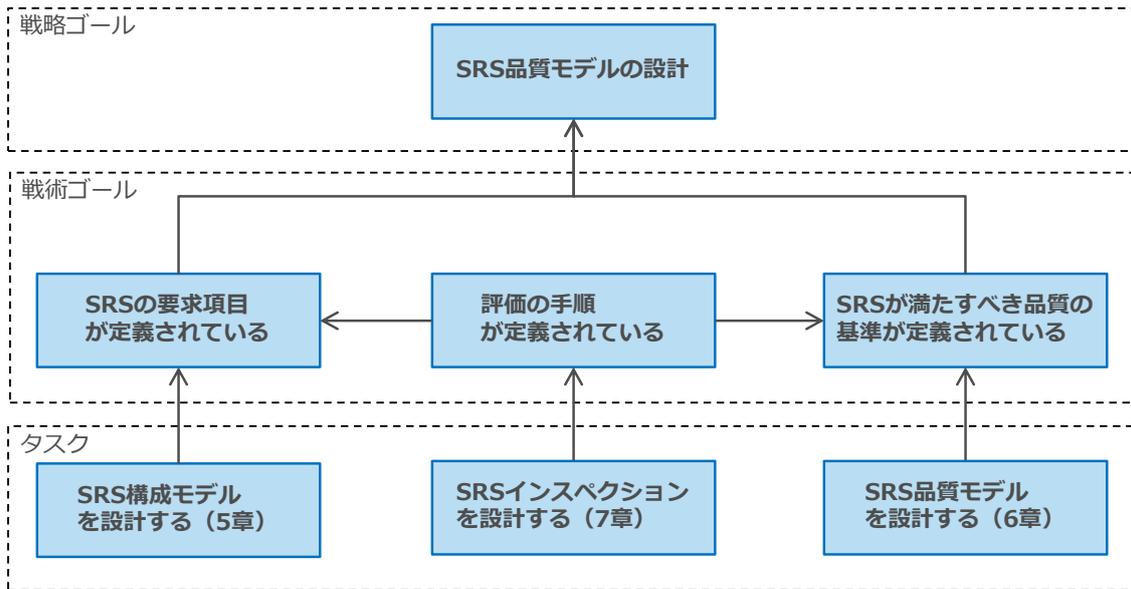


図 4-2 本研究の全体的なアプローチ

### 4.3 ステークホルダの関心事に基づく SRS 構成モデルの設計

ステークホルダはソフトウェアに対して各自のパースペクティブに起因する異なる関心事を持つ。それらの関心事の違いが要求の多様性を生み出す。したがって、多様な要求を表現できる SRS 構成モデルの設計にはステークホルダの関心事を捉えることが必要である。本研究では、ステークホルダのドメイン特性に関する関心事に着目した SRS 構成モデルの設計方法を提案する。また、提案手法を適用し、自動車ソフトウェアに適した SRS 構成モデルを構築する。

### 4.4 パースペクティブに基づく SRS 品質モデルの設計

言語学的には、SRS に記述された文章の構造と内容の正しさに加え、読者（ステークホルダ）にとっての記述内容の理解しやすさが重要である。本研究では、自動車 SRS を読解するステークホルダのパースペクティブ（立場）を考慮したプラグマティック品質特性を定義する。また、定義したプラグマティック品質特性を正当性確認と妥当性確認のどちらの対象とすべきかによって分類した SRS 品質モデルを構築する。

### 4.5 SRS 品質モデルに基づく自動車 SRS インスペクション方法

SRS が十分な品質を持つためには、次の二つの条件を満たす必要がある。

- (1) SRS 構成モデルに定義された要求項目がもれなく記述されていること
- (2) 記述された内容が SRS 品質モデルに定義されたプラグマティック品質特性を満たしていること

これら二つの要求を満たしていることを確認する方法としてインスペクションを利用した方法を設計する。研究課題でも述べた通り、SRS 品質モデルに基づく効率的なインスペクション方法として要求の正当性確認と妥当性確認からなる二段階インスペクションプロセスのアプローチをとる。

本研究では、この二段階のうち要求の正当性確認フェーズを主な研究対象とし、要求の正当性確認の効率的な実施方法として第三者インスペクションを設計する。第三者インスペクションの設計には関連研究で挙げた RISDM[59]を参照モデルとする。RISDM は企業情報システムの SRS を対象に開発された方法論である。そのため、そのままでは特性の違う自動車ソフトウェアの SRS をインスペクションすることはできない。本研究では、自動車 SRS のインスペクションに適合するための RISDM に対する拡張を提案する。

## 5 ステークホルダの関心事に基づく SRS 構成モデルの設計

### 5.1 SRS のメタモデル

開発対象ソフトウェアのステークホルダがドメインに対して持つ関心事に基づき、多様な要求と SRS 構造の対応関係を明らかにするため、要求項目のメタモデルを図 5-1 の通り定義する。

このメタモデルでは、あるドメインに特化した SRS 構成モデルはドメインに依存しない共通の要素（要求項目）に、ドメイン依存の要素（要求項目）を添加（mix-in）することで生成されることを意味する。製品ドメインに依存しない共通の要素は参照 SRS から継承する。一方、ドメインに依存する要素は開発対象ソフトウェアに対するステークホルダの関心事と関連付けられる。これは、ドメイン依存の要求項目の目的が、ステークホルダの関心事に関する要求表現するためである。ドメイン特性に関する関心事は、ステークホルダが自身のパースペクティブを通してドメイン特性を評価することで明らかになる。

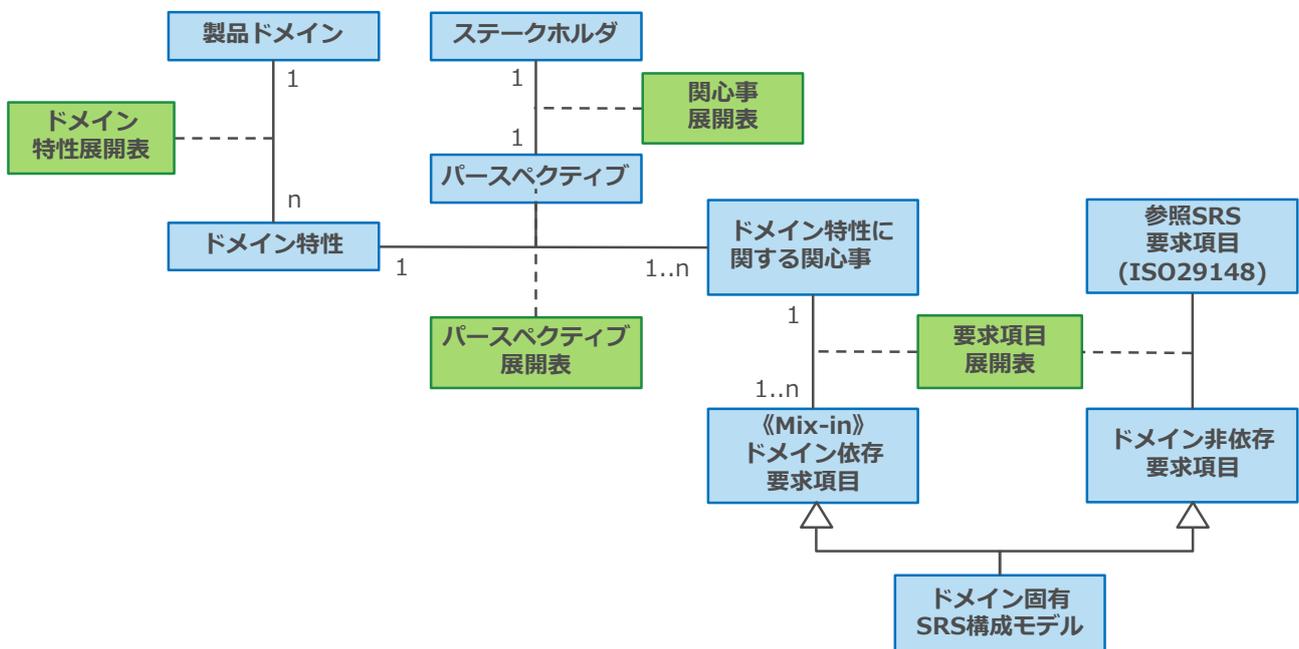


図 5-1 要求項目のメタモデル

### 5.2 ステークホルダの関心事に基づくドメイン固有 SRS 構成モデルの設計プロセス

#### (1) SRS 構成モデル展開表の関係モデル

ステークホルダの関心事からドメイン固有 SRS 構成モデルを導出する方法として、本研究では、品質機能展開（QFD：Quality Function Deployment）の考え方を導入する。

図 5-2 は図 5-1 のメタモデルに基づきステークホルダの関心事からドメイン固有 SRS 構成モデルを導出する 4 つの展開表間の関係を表している。本研究では、これをまとめて SRS 構成モデル展開表と呼ぶこととする。関心事展開表は、ドメイン特性展開表とパースペクティブ展開表の展開結果を入力として、ドメイン特性に関する関心事を展開する。要求項目展開表は、関心事展開表の展開結果を入力に、参照 SRS 要求項目と追加 SRS 要求項目を展開する。なお、各展開表の中の矢印（↓）は行列の展開する向きを示している。例えば、ドメイン特性展開表では製品ドメインからドメイン特性へ展開を行うことを意味する。

本研究では、SRS 構成モデル展開表によって図 5-1 に示したメタモデルの各クラスを具体化していくことで SRS 構成モデルを設計する。その詳細な手順については以降の節で説明する。

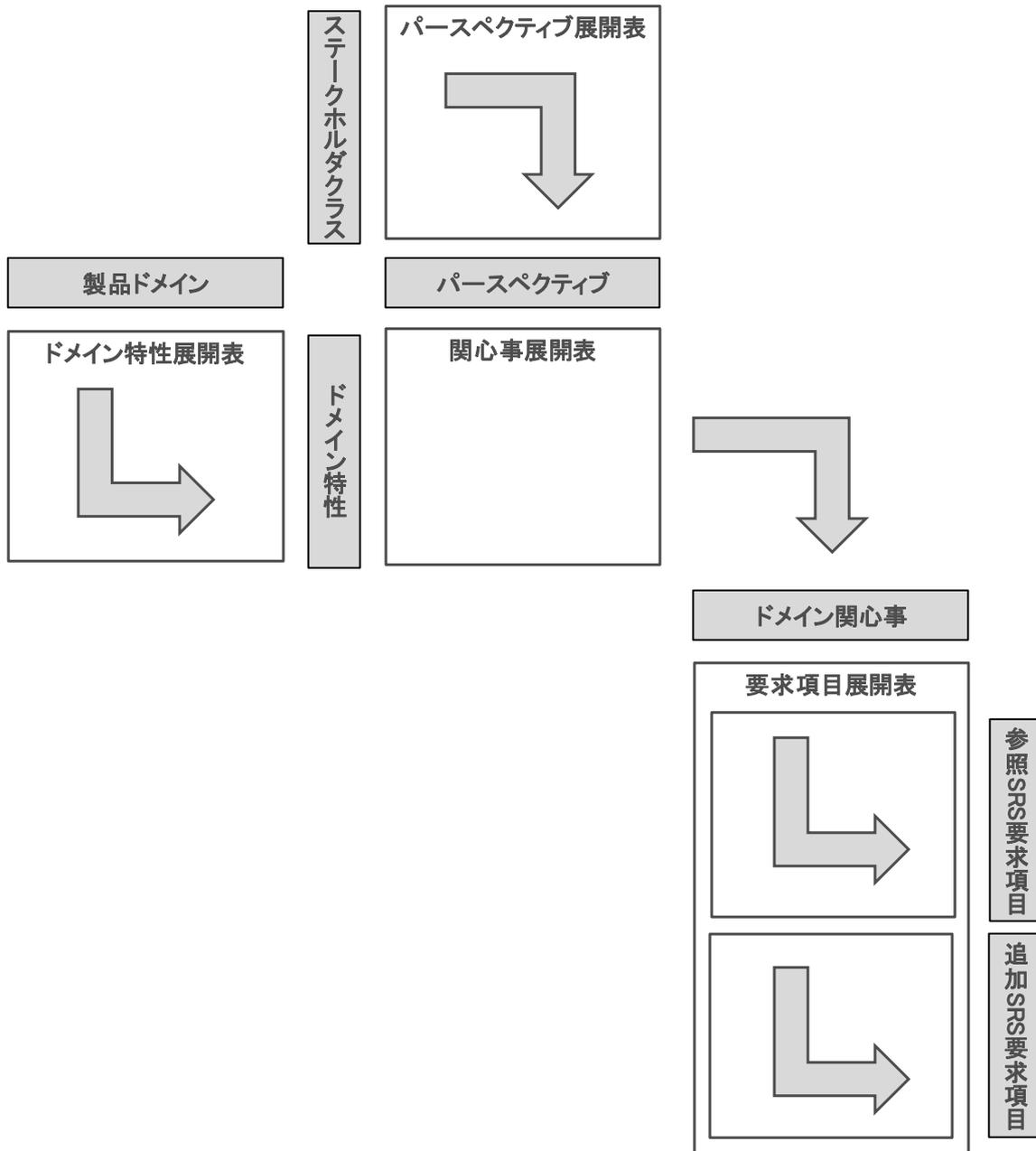


図 5-2 SRS 構成モデル展開表

## (2) SRS 構成モデルの設計プロセス

前節で示した SRS 構成モデル展開表に基づき、本研究では図 5-3 に示す次の 6 つのステップから構成される SRS 構成モデルの設計プロセスを提案する。

ステップ (1)：既存の SRS 構成モデルの中から、ドメイン固有の SRS 構成モデルを設計するための最適な参照 SRS 構成モデルを選択する。

ステップ (2)：複数の製品ドメインから各ドメインの特性を展開する。

ステップ (3)：ステークホルダが持つパースペクティブを展開する。

ステップ (4)：ステップ (1) とステップ (2) で展開したドメイン特性とパースペクティブから、ステークホルダがドメイン特性に関して持つ関心事を展開する。

ステップ (5)：まず初めにドメイン特性に関する関心事と参照 SRS 構成モデルの要求項目の対応付けを行う。続いて、参照 SRS 構成モデルとの対応関係が見つからなかったドメイン特性に関する関

心事に対して、それらの関心事に関する要求を表現するための要求項目を定義する。  
ステップ (6)：新たに定義した要求項目を追加することで参照 SRS 構成モデルを拡張し、ドメイン固有の SRS 構成モデルを設計する。

以降の節では、自動車ソフトウェア向け SRS 構成モデルの設計を例に各ステップの詳細について説明する。

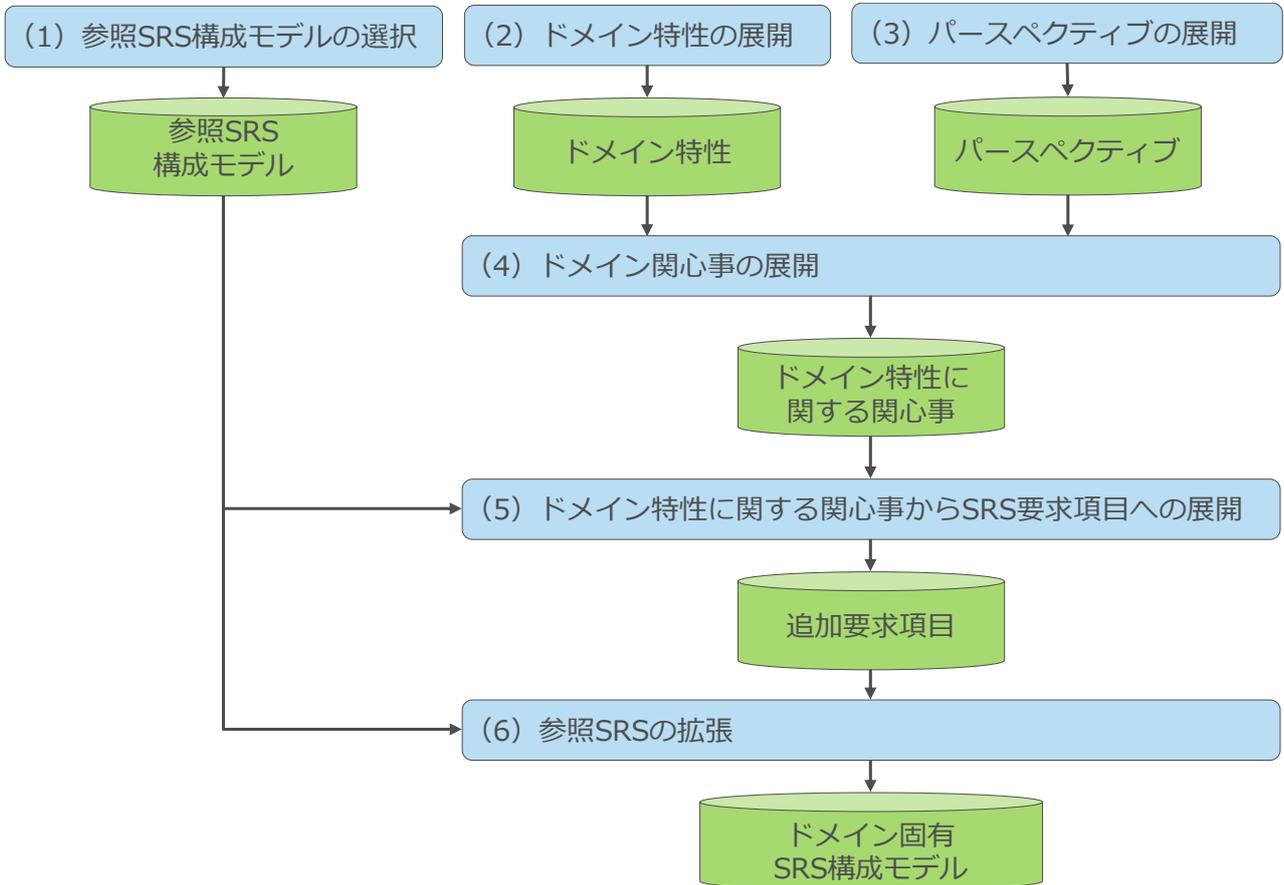


図 5-3 SRS 構成モデルの設計プロセス

### 5.3 参照 SRS 構成モデルの選択

要求仕様書モデルのすべてを独自に定義するには工数を要するので、自組織に適した既存の SRS 構成モデルを選定し利用する方法を提案する。ベースとなる SRS 構成モデル（以下、参照 SRS 構成モデルと呼ぶ）の選定には[18]で提案されている評価基準を利用する。[18]による 5 つの SRS に対する評価結果を表 5-1 にまとめる。

表 5-1 SRS の評価基準と評価結果

評価基準	IEEE 830-1984	ESA PSS-05-0	NASA DID-P200	JPL D-4005	DOD MIL-STD-498
独立性 Independence	Medium	Medium	High	High	High
統合性 Integration	Mdeium	Medium	High	High	High
正確性 Precision	High	Low	High	High	High
一般性 Generality	Not for large project	Not for large projects	Not for small projects	Suits all projects	Not for small projects
構成 Organization	High	Low	Low	Medium	Medium to high
項目の完全性 Cotent completeness	High	Medium	High	High	High
ビューの完全性 View completeness	Satisfiers all users	Satisfies only testers	Satisfies design and testers	Satisfies all users	Satisfies all users
変更容易性 Modifiability	Adequate	Inadequate	Inadequate	Adequate	Adequate

自動車ソフトウェア開発では下記の理由から「一般性」、「ビューの完全性」、「変更容易性」の 3 つの評価基準が重要となる。

- (1) 一般性: 規模の異なる大小のプロジェクトが存在することから、評価基準「一般性」がすべてのプロジェクトに適合する (Suits all projects) と評価されていること
- (2) ビューの完全性: 多様なステークホルダが SRS の読者となることから、評価基準「ビューの完全性」がすべてのユーザ (のニーズ) を満足する (Satisfies all users) と評価されていること
- (3) 変更容易性: 派生開発が主流のため変更容易性が重要であることから、評価基準「変更容易性」が十分である (Adequate) と評価されていること

表 5-1 の中で三つの評価基準をすべて満たすのは JPL D-4005 のみであるが、この規格は現在一般には公開されていない。JPL D-4005 に次いで自動車ソフトウェアへの適合度が高いのは「一般性」と「変更容易性」の評価基準を満たす IEEE 830-1998 と DOD MIL-STD-498 の二つである。自動車ソフトウェア開発では国際規格への準拠が強く求められるので、最終的に IEEE 830-1998 が参照 SRS 構成モデルとして最適であると結論した。しかし、IEEE 830 はすでに廃版となり後継規格の ISO/IEC/IEEE 29148-2011[27]で置き換えられている。ISO 29148 が提案する SRS 構造は、細部の違いはあるが基本的に IEEE 830 を踏襲しているので、本研究では ISO 29148 で定義されている SRS の構造を参照 SRS 構成モデルとした。

## 5.4ドメイン特性の展開

### 5.4.1 製品ドメインの定義

自動車を構成するカーエレクトロニクス製品は多様であり、複数の製品ドメインに分類することができる。文献により製品ドメインの分け方には差異があり、広く合意された分類は存在しない。文献[42][53][62][63]が定義する製品ドメインを比較したところ、表 5-2 に示す対応関係が得られた。

表 5-2 参考文献で定義された製品ドメインの対応関係

		文献			
		Automotive Software Engineering	Software Engineering for Automotive Systems	Automotive Software Architecture	Software Engineering for Embedded Systems
製品ドメイン	Electronic Systems of the Powertrain		Power train and chassis control	Powertrain	Powertrain and Transmission
	Electronics of the Chassis				Chassis and Safety
	Body Electronics		Body/comfort	Chassis and body	Body
	Comfort and Convenience				Chassis and Safety
	Passive Safety		Safety Electronics		
	Multimedia Systems		Multimedia, telematics, and HMI	N/A	Infotainment and Telematics
	Distributed and Networked Electronic Systems		N/A	Active Safety	Driver Assistance
	N/A		Infrastructure	Electronic systems	N/A

本研究では表 5-2 の対応関係を参考に、製品ドメインを下記の 5 つに分類した。

- (1) パワトレ&シャシー
- (2) ボディー
- (3) アクティブセイフティー
- (4) インフォテインメント
- (5) 通信インフラ

これら 5 つの製品ドメインのうち、パワトレ&シャシードメインはパワートレインとシャシー、ボディードメインは利便快適とパッシブセイフティーをそれぞれサブドメインに持つ。各製品ドメインの説明と製品例を表 5-3 に示す。

表 5-3 カーエレクトロニクス製品ドメイン

製品ドメイン		説明	製品例
(1) パワトレ&シャシー	パワートレイン	車両の駆動力に関する製品	エンジン制御ユニット
	シャシー	車両の走行制御に関する製品	電動パワーステアリング
(2) ボディー	利便快適	車室内の利便性と快適性に関する製品	カーエアコン
	パッシブセイフティー	衝突時の乗員の安全を確保するための製品	エアバッグ
(3) アクティブセイフティー		衝突事故を回避または損害を軽減するための製品	自動ブレーキ
(4) インフォテインメント		乗員へ情報と娯楽を提供するための製品	カーナビゲーション
(5) 通信インフラ		車両と車外のサービスを接続するための製品	データ通信モジュール

### 5.4.2 ドメイン特性展開表の作成

上記で定義した製品ドメインを入力にドメイン特性の展開を行う。本研究で調査した範囲内では、カーエレクトロニクス製品ドメインによる特性を網羅的に扱った文献は存在しないため、製品ドメインの定義で参照した文献[42][53][62][63]を参考に、製品ドメインからドメイン特性への展開を行った(表 5-4)。参考のため、表 5-4 には展開されたドメイン特性と ISO 25010[25]で定義された品質特性との対応関係も併せて表記している。

表 5-4 ドメイン特性展開表

ISO25010への対応付け	ドメイン特性	製品ドメイン						
		パワトレ&シャシー		ボディー		アクティブセイフティー	インフォテインメント	通信インフラ
		パワートレイン	シャシー	利便快適	パッシブセイフティー			
N/A	制御アルゴリズム (連続制御)	○	○			○		
N/A	離散イベント処理 (離散制御)			○	○			
N/A	データ処理					○	○	○
相互運用性	車内ネットワーク上での相互運用性	○	○	○	○	○	○	○
	車外システムとの相互運用性						○	○
時間効率性	ハードリアルタイム	○	○		○	○		○
	ソフトリアルタイム			○			○	○
健康・安全リスク緩和性	安全性要求	○	○		○	○		
セキュリティ	セキュリティ要求						○	○
N/A	プライバシー要求						○	○
信頼性	信頼性要求	○	○		○	○		○
可用性	可用性要求	○	○		○	○		
N/A	法規遵守	○	○		○	○		○
使用性	ユーザインタフェース (メカニカル)			○			○	
	ユーザインタフェース (GUI)						○	
適応性	適合 (キャリブレーション)	○	○			○		
	可変性	○	○	○	○	○	○	○

## 5.5 ステークホルダのパーспекティブの展開

### 5.5.1 ステークホルダの特定と抽象化

SRS が要求を定義する対象のソフトウェアに関わりを持つステークホルダを特定する。ステークホルダの特定方法にはオニオンモデル[2]やペルソナ[54]など多くの方法がある。

自動車ソフトウェアの開発では、3年から5年程度の長い開発ライフサイクルを通して多くのステークホルダが関わるため企業情報システムの開発に利用される方法をそのまま適用することは困難である。本研究では、簡略化した自動車開発のライフサイクルに沿って主要なステークホルダを特定する方法をとる。特定したステークホルダは、同種の関心事を持つグループをステークホルダクラスとしてまとめることで抽象化する。ステークホルダの特定と抽象化の結果を表 5-5 に示す。

表 5-5 自動車ソフトウェアのステークホルダ

開発ライフサイクル	ステークホルダ	ステークホルダクラス
製品企画	完成車メーカー	取得者（顧客）
	上位Tierサプライヤ	
製品開発	プロジェクトマネージャ	プロジェクトマネージャ
	要求エンジニア	要求エンジニア
	システム開発者	開発者
	ソフトウェア開発者	
	ハードウェア開発者	
	ビルド担当者	
	システムテスタ	テスタ
	ソフトウェアテスタ	
	ハードウェアテスタ	
		ユーザマニュアル作成者
サービスマニュアル作成者		
量産	生産技術者	製造部門
	品質保証技術者	
販売	自動車セールス	自動車セールス
利用	整備士	整備士
	自動車オーナー	ユーザ
	商用車両運転手	
	品質管理技術者	品質管理部門

### 5.5.2 パーспекティブ展開表の作成

ステークホルダクラスからパーспекティブを展開する。パーспекティブとは、ある特定の読者が SRS に対して取る立場である。そのため、異なるステークホルダは SRS に対して異なるパーспекティブを持つ。例えば、取得者は SRS に記述される「要求を出す（要求する）」というパーспекティブを持つ。また、表 5-5 に挙げたステークホルダクラスのうち整備士のように直接は SRS を参照しないステークホルダクラスもある。しかし、整備士はテクニカルライタが SRS を入力に作成したサービスマニュアルを参照して車両の点検や修理を行う。そのため、整備士は SRS を間接的に参照していると解釈し、「点検・修理する」を整備士のパーспекティブとして設定した。各ステークホルダクラスに対して主要なパーспекティブ（1次パーспекティブ）と2次的なパーспекティブを展開した結果を表 5-6 に示す。

表 5-6 パースペクティブ展開表

		パースペクティブ										
		要求する	管理する	変更する	設計・実装する	テストする	説明する	製造する	販売する	点検・修理する	利用する	品質管理する
ステークホルダクラス	取得者（顧客）	P	S			S					S	
	プロジェクトマネージャ	S	P									
	要求エンジニア			P							S	
	開発者				P	S					S	
	テスト					P					S	
	テクニカルライター						P					
	製造部門					S		P				
	自動車セールス								P		S	
	整備士									P		
	ユーザ										P	
	品質管理部門											P

凡例

P：1次パースペクティブ S：2次パースペクティブ

## 5.6 ステークホルダのドメイン特性に関する関心事の展開

これまで展開したドメイン特性とステークホルダクラスのパースペクティブから、ステークホルダクラスが持つ自動車製品のドメイン特性に関する関心事を展開する。関心事の展開には各ステークホルダの1次パースペクティブのみを利用する。ドメイン特性とステークホルダクラスのパースペクティブからドメイン特性に関する関心事を展開する理由は、ステークホルダが自身のパースペクティブを通してドメイン特性を評価することで、そのドメインに関する関心事が明確になるためである。そのため、パースペクティブの異なるステークホルダ同士では関心事も異なる。

関心事を展開するためのガイドとして自動車ソフトウェアのような組込みソフトウェアでは[1]の関心事チェックリスト、企業情報システムソフトウェアでは Zachman フレームワーク[73]のような視点をガイドとして利用することを提案する。例えば[1]には時間的な特性 (Temporal characteristics) に関するチェックリストとして以下の質問が提案されている。

- (1) センサと対応するアクチュエータの間に遅延はあるか？
- (2) フィードバックループには応答時間の制約はあるか？
- (3) 通信チャンネルには応答時間の制約はあるか？

開発者のパースペクティブ「設計・実装する」からドメイン特性「ハードリアルタイム」を評価するために上記のチェックリスト (2) や (3) を参照することで、「応答時間」というドメイン関心事を抽出することがガイドされる。同様にすべてのドメイン特性に対してドメイン関心事を展開した結果を表 5-7 に示す。

表 5-7 ドメイン関心事展開表

		ドメイン関心事																																				
		応答時間	スループット	スケラビリティ	予測可能性	ハードウェアリソース要求	ピーク負荷時の振る舞い	システムのモード	事前事後条件	通信プロトコル	データフォーマット	脅威	セキュリティメカニズム	検出と回復	コーディング規約	検証方法	障害モード	フェイルセーフ	安全規格	データの回復	MTBF	診断情報の記録	フェイルオーバーシヨナル	個人情報の保護	個人情報の消去	法規の順守	入力頻度	ノイズ	国際化	適合パラメータ	適合範囲	パラメータ間依存関係	可変点	バリエーション	バリエーション選択基準			
ド メ イ ン 特 性	制御アルゴリズム (連続制御)	○			○																																	
	離散イベント処理 (離散制御)						○	○																														
	データ処理		○																																			
	車内ネットワーク上での相互運用性			○						○	○																											
	車外システムとの相互運用性			○						○	○																											
	ハードリアルタイム	○			○	○																																
	ソフトリアルタイム	○																																				
	安全性要求															○	○	○	○					○														
	セキュリティ要求													○	○	○	○																					
	プライバシー要求																								○	○												
	信頼性要求																					○	○	○														
	可用性要求																						○	○														
	法規遵守																										○											
ユーザインタフェース (メカニカル)																										○	○											
ユーザインタフェース (GUI)																												○										
適合 (キャリブレーション)																													○	○	○							
可変性																																		○	○	○		
パ ー ス バ ク テ ィ ブ	取得者 (顧客)							○		○	○								○			○	○	○	○													
	プロジェクトマネージャ														○											○												
	要求エンジニア																																					
	開発者	○	○	○	○	○	○	○	○	○	○	○	○	○												○	○							○	○	○	○	
	テスト	○				○	○			○	○			○		○																						
	テクニカルライタ																											○										
	製造部門										○	○																		○	○	○						
	自動車セールス																																					
	整備士																							○														
	ユーザ																																					
品質管理部門																																						



## 5.8 参照 SRS 構成モデルの拡張

SRS 構成モデル設計プロセスの最終ステップでは、ステップ (5) で展開した追加要求項目を参照 SRS 構成モデルへ追加 (mix-in) することで拡張し、新たな SRS 構成モデルを設計する。(1) から (6) のステップから成る SRS 構成モデルの設計プロセスに従い設計した自動車ソフトウェアの SRS 構成モデルを図 5-4 に示す。

図 5-4 では、設計した SRS 構成モデルをクラス図で表現している。SRS クラス以外の個々のクラスは要求項目に相当し、SRS クラスは他のクラスのコンポジションとして表現される。白い背景色のクラスが参照 SRS 構成モデル (ISO 29148) から継承した要求項目、青い背景色のクラスが参照 SRS 構成モデルへ追加した要求項目をそれぞれ表している。

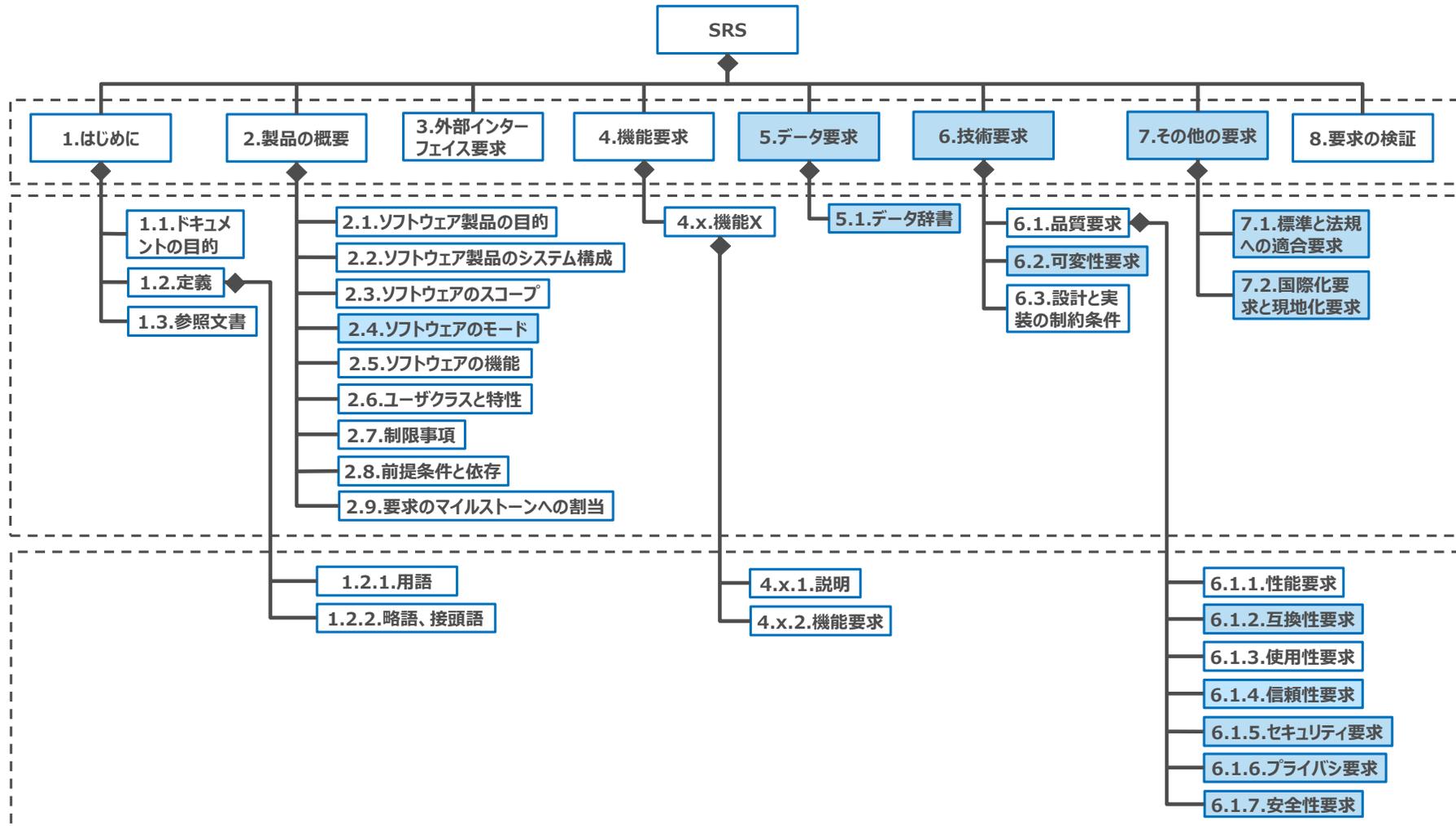


図 5-4 自動車ドメイン固有 SRS 構成モデル

## 6 パースペクティブに基づく SRS 品質モデルの設計

### 6.1 SRS 品質モデル定義の目的

SRS 品質モデルを定義する目的は以下の二つである。

(1) 読者（ステークホルダ）のパースペクティブの導入

SRS 品質の確認は SRS の読者が正しく内容を理解できる程度を表すプラグマティック品質のレベルで行う必要がある。そこで、SRS 読者のパースペクティブに基づき SRS 品質を定義することで、単なる概念であるプラグマティック品質をプラグマティック品質特性（PQC）として具体化する。

(2) SRS の V&V（Verification and Validation）フレームワークの確立

IEEE 830-1998 をはじめ、SRS 品質モデルを提案する多くの文献がある。しかし、従来の SRS 品質モデルでは正当性確認と妥当性確認の観点が分離されておらず、各品質特性を正当性確認と妥当性確認どちらのプロセスに適用すべきかが明確ではないという課題がある。本研究では、具体化した PQC を正当性確認と妥当性確認で確認すべき品質項目として、次の二つに分離する。

1) 表現品質: 正当性確認の確認項目

2) 内容品質: 妥当性確認の確認項目

この分類により、体系的な SRS の V&V 活動が可能になると期待できる。

### 6.2 プラグマティック品質特性の必要性

関連研究 3.3 で述べた通り、自然言語で記述された要求の品質モデルが提案されている（図 6-1）。そのモデルでは、言語論に基づきシンタクティック、セマンティック、プラグマティックの 3 つの品質タイプが定義されている。それぞれの品質タイプの意味は以下の通りである。

1) シンタクティック品質

要求を記述する文章の文法的な正しさの程度に関する品質

2) セマンティック品質

記述された要求の正しさの程度に関する品質

3) プラグマティック品質

読者のパースペクティブから見た要求の理解しやすさの程度に関する品質

図 6-1 は上記の 3 つの品質タイプ間の依存関係を表している。シンタクティック品質が満たされていない、すなわち文法的に正しくない文章では読者が要求を正しく解釈できないため、セマンティック品質とパースペクティブ品質も満たされない。シンタクティック品質が十分でもセマンティック品質が不十分、すなわち文法的には正しいが意味的には誤った要求は、読者のパースペクティブに関わらず正しく理解することができないためプラグマティック品質を満たすことができない。さらに、シンタクティック品質とセマンティック品質の両方が満たされている、すなわち文法的にも意味的にも正しい要求記述であっても、その要求を必要とする読者のパースペクティブからは正しく理解できない場合、プラグマティック品質を満たすことができない。



図 6-1 自然言語で記述された要求の品質モデル[14]（再掲）

以上の理由から、SRS 読者のパースペクティブを考慮したプラグマティック品質の導入が必要である。しかし、プラグマティック品質自体は単なる概念であるため、SRS 品質の評価へ適用するには品質特性として具体化する必要がある。

### 6.3 プラグマティック品質特性の導出プロセス

プラグマティック品質特性は基準となる参照 SRS 品質特性に対する語用論的特性をステークホルダのパースペクティブから理解のしやすさとして定義する。そのため、プラグマティック品質特性は図 6-2 の導出関係に基づき、4つのステップから構成される以下プロセスに従い導出する。

- (1) プラグマティック品質特性を導出する基準となる参照 SRS 品質特性の定義
- (2) ステークホルダの特定とパースペクティブの決定
- (3) プラグマティック品質特性の定義
- (4) プラグマティック品質特性への名称の付与

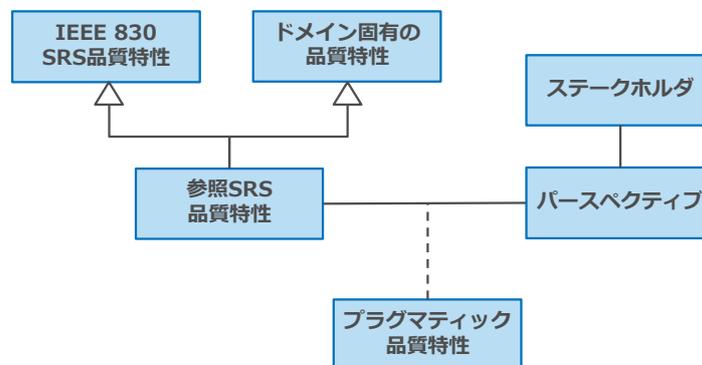


図 6-2 プラグマティック品質特性の導出関係

#### 6.3.1 参照 SRS 品質特性の定義

プラグマティック品質特性を導出する際に基準となる参照 SRS 品質特性を定義する。本研究では、IEEE 830-1998 で定義された SRS 品質特性を参考に、自動車ソフトウェアの特性を考慮した新たな SRS 品質特性を定義し、それを参照 SRS 品質特性とした。具体的には、IEEE 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、新たに達成可能性 (Achievable) を追加した。「重要度/安定度がランク付けされている」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 [16]。これに対して、達成可能性を追加した理由は IEEE 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。また、IEEE 830-1998 では曖昧な要求は検証不可能である [21] と説明されているように、検証可能性は無曖昧性に大きく依存している。そこで、これら 2 つの品質特性を統合し明確性という新たな品質特性を定義した。最終的に表 6-1 に示す 7 つの品質特性を自動車ソフトウェア向けの参照 SRS 品質特性として定義した。

表 6-1 IEEE 830-1998 から導出した参照 SRS 品質特性

IEEE 830-1998 SRS品質特性	参照SRS品質特性
正当性	正当性
無曖昧性	明確性
検証可能性	
完全性	完全性
一貫性	一貫性
変更容易性	変更容易性
追跡可能性	追跡可能性
優先度と安定性のランク付け	N/A
N/A	達成可能性

### 6.3.2 ステークホルダの特定とパースペクティブの決定

パースペクティブの定義に先立ち、パースペクティブを所有するステークホルダを特定する。本研究では、5.5 節で挙げたステークホルダクラスのなかから、直接 SRS を参照するステークホルダクラスのみを選択する。

続いて、各ステークホルダクラスの SRS に対する関わり方、すなわちパースペクティブを決定する。選択されたステークホルダクラスとそのパースペクティブの一覧を表 6-2 に示す。

表 6-2 SRS を直接参照するステークホルダクラス

ステークホルダ	ステークホルダクラス	パースペクティブ
完成車メーカー	取得者（顧客）	要求する
上位Tierサプライヤ		
プロジェクトマネージャ	プロジェクトマネージャ	管理する
要求エンジニア	要求エンジニア	変更する
システム開発者	開発者	設計・実装する
ソフトウェア開発者		
ハードウェア開発者		
ビルド担当者		
システムテスト	テスト	テストする
ソフトウェアテスト		
ハードウェアテスト		
ユーザマニュアル作成者	テクニカルライター	説明する
サービスマニュアル作成者		

### 6.3.3 プラグマティック品質特性の定義

6.3.2 で導出したパースペクティブに対しプラグマティック品質特性（PQC : Pragmatic Quality Characteristics）を定義する。PQC の定義は 4 つのステップからなるプロセスに沿って行う。以下ではプロジェクトマネージャクラスを例に各ステップの詳細を説明する。

#### (1) パースペクティブの詳細化

6.3 で導出したパースペクティブを詳細化する。必要であれば一つのパースペクティブを複数のパースペクティブへ分解する。本研究では、パースペクティブを 2 段階目（レベル 2）まで詳細化する。

プロジェクトマネージャクラスのレベル 1 のパースペクティブは「管理する」である。このパースペクティブの分解と詳細化を行い「必要な成果物が作成されていることを保証する」、「正しい成果物が作成されていることを保証する」という 2 つのパースペクティブ（レベル 2）を得た。

(2) 参照 SRS 品質特性の対応付け

詳細化された（レベル 2 の）各パースペクティブに対し、最も関連性の高い参照 SRS 品質特性を選択する。

プロジェクトマネージャクラスのパースペクティブ「必要な成果物が作成されていることを保証する」に対して満たすべき品質特性として、要求と要求に基づき作成された成果物の間の対応関係が追跡できることを意味する「追跡可能性」を選択した。

(3) PQC の定義

選択した参照 SRS 品質特性に対し、SRS 読者の理解のしやすさを測る具体的な基準として PQC を定義する。

追跡可能性の意味（要求と要求に基づき作成された成果物の間の対応関係が追跡できること）をプロジェクトマネージャにとって理解しやすくするために SRS が備えるべき特性として「双方向トレーサビリティが確立されている」を定義した。

(4) PQC 名称の付与

最後に、PQC の定義に対し理解しやすい特性名を付与する。「双方向トレーサビリティが確立されている」という特性に対しては、対応する参照 SRS 品質特性の追跡可能性がすでに定義内容を容易に理解できる特性名となっているため、この例では対応する PQC も同じく「追跡可能性」という名称を付与した。対応する参照 SRS 品質特性の名称からは PQC の定義内容が容易に想像できない場合は別の名称を付与する。例えば、プロジェクトマネージャクラスのもう 1 つのパースペクティブ「正しい成果物が作成されていることを保証する」も参照 SRS 品質特性の追跡可能性に対応しているが、こちらは PQC の定義をよりよく表現する「外部一貫性」という名称を付与した。

以上の手順に従い、表 6-2 に示したすべてのパースペクティブに対し、パースペクティブの詳細化（ステップ 1）、参照 SRS 品質特性への対応付け（ステップ 2）、PQC の定義（ステップ 3）、PQC 名称の付与（ステップ 4）を行った結果を表 6-3 に示す。

表 6-3 パースペクティブとプラグマティック品質特性

ステークホルダクラス	パースペクティブ		参照SRS品質特性							プラグマティック品質特性 (PQC)	
	レベル1	レベル2	正当性	明確性	完全性	一貫性	変更容易性	追跡可能性	実現可能性	特性名	定義
取得者	要求する	すべての要求が獲得されていることを確認する			×					Requirements completeness 要求網羅性	SRSには製品の成立に必要な要求がすべて記述されている
		不要な要求が追加されていないことを確認する	×							Correctness 正当性	SRSに記述された要求はすべて製品の成立に必要なものである
		要求が正しく理解されていることを確認する	×							Accountability 責任追跡性	要求の必要性に対する説明責任が果たされている
プロジェクトマネージャ	管理する	必要な成果物が作成されていることを保証する						×		Traceable 追跡可能性	双方向トレーサビリティが確立されている
		正しい成果物が作成されていることを保証する						×		external consistency 外部一貫性	トレーサビリティにより内容や意味が正しく継承されている
要求エンジニア	変更する	要求仕様書を正しく変更する					×			Modifiable 変更容易性	SRSは変更が容易な構成になっている
設計者 テスト	設計する 実装する テストする	要求に矛盾のないことを確認する				×				Internal consistency 内部一貫性	SRS内に記述された要求が互いに矛盾していない
		要求の実現性を判断する							×	Feasible 実現可能性	記述された要求はすべて現実的な制約の範囲内で実現できる
テクニカルライター	説明する	要求を一意に解釈する	×							Definiteness 明確性	すべての読者が一意に内容を解釈できるよう明確に記述されている
		設計・実装に必要な情報をすべて入手する 検証に必要な情報をすべて入手する マニュアル作成に必要な情報をすべて入手する			×					Descriptive completeness 記述網羅性	SRSに記述すべき内容がすべて存在する

## 6.4 SRS の正当性確認と妥当性確認

ソフトウェア開発成果物の正しさを確認する手段として正当性確認と妥当性確認 (V&V: Verification and Validation) の考え方が広く受け入れられている。ソフトウェア V&V のプロセスとアクティビティは国際規格の IEEE 1012-2004 に定義がある [22]。しかし、SRS に対する正当性確認と妥当性確認の方法はいまだ確立していない。限定的な取り組みとして形式仕様言語によって記述された要求仕様に対するシミュレーションやモデル検査による正当性確認が行われている [10]。しかし、機能要求以外の品質要求や制約といったいわゆる非機能要求、要求の根拠や開発の目的といった背景情報は形式仕様言語では表現できない。また、自然言語で記述された SRS に対する正当性確認と妥当性確認に関する研究は少ない。

要求工学知識体系 (REBOK) [29] では、要求の正当性確認と妥当性確認はそれぞれ以下のように定義されている。

(1) 要求の正当性確認

要求仕様書が構造的かつ意味的な特性に照らして正しいことを確認すること

(2) 要求の妥当性確認

要求仕様書がステークホルダの所期の要求を満たしていることを確認すること

上記の要求の正当性確認の定義のうち、構造的な正しさはシンタクティック品質に、意味的な正しさはセマンティック品質にそれぞれ対応する。一方、要求の妥当性確認の「ステークホルダの所期の要求を満たしている」という定義はプラグマティック品質に対応する。実際には、プロジェクトで定義された用語の利用といった正当性確認プロセス内で確認可能なプラグマティック品質も一部存在する。一方、ステークホルダのパースペクティブに関わりなく、意味的な正しさ (セマンティック品質) のみから妥当性を確認できる要求も存在する。以上から、シンタクティック品質、セマンティック品質、プラグマティック品質と正当性確認および妥当性確認との対応関係を整理した結果を表 6-4 に示す。この対応関係は常に成立する厳密なものではないが、要求の正当性確認と妥当性確認のプロセスで評価すべき品質特性の指針と成り得る。

表 6-4 正当性確認および妥当性確認プロセスで評価できる品質タイプ

	正当性確認	妥当性確認
プラグマティック品質	△	○
セマンティック品質	○	△
シンタクティック品質	○	×

凡例 ○：評価可能、△：一部評価可能、×：評価不可能

6.3.3 ではステークホルダクラスのパースペクティブに基づいて品質特性を定義したため、すべての品質特性をプラグマティック品質特性 (PQC) と呼んでいる。しかし、それらの PQC はシンタクティック品質とセマンティック品質を内包している。そこで、表 6-4 の対応関係に従い 6.3.3 で定義した PQC を正当性確認プロセスで評価すべきものと妥当性確認プロセスで評価すべきものに分類する。また、正当性確認プロセスで評価する PQC の集合を表現品質、妥当性確認プロセスで評価する PQC の集合を内容品質と呼ぶこととする。表現品質と内容品質の意味はそれぞれ以下のように定義する。

(1) 表現品質 (Representation Quality)

SRS に記述された文章の構造的な正しさと、意味的な正しさの程度に関する品質

(2) 内容品質 (Contents Quality)

(システム要求などの) 上位要求に照らして、SRS に記述された要求がそれらの要求を満たしている程度に関する品質

上記の定義に従い，各 PQC の表現品質と内容品質への割り当て結果を表 6-5 に示す。

表 6-5 PQC の表現品質と内容品質への割り当て

		正 当 性	明 確 性	完 全 性	一 貫 性	変 更 容 易 性	追 跡 可 能 性	実 現 可 能 性	表 現 品 質	内 容 品 質
P Q C	要求網羅性			×						×
	正当性	×								×
	責任追跡性	×							×	
	追跡可能性						×		×	
	外部一貫性						×			×
	変更容易性					×			×	
	内部一貫性				×					×
	実現可能性							×		×
	明確性		×						×	
	記述網羅性			×					×	

これまでに導出した SRS 品質モデルの構造を UML のクラス図で表現したものが図 6-3 である。このように品質タイプを分類することで，SRS の正当性確認と妥当性確認が効体系的に行えるようになると期待できる。



図 6-3 クラス図で表現した SRS 品質モデル

## 7 SRS 品質モデルに基づく自動車 SRS インスペクション方法

### 7.1 二段階インスペクションプロセスの提案

SRS 品質モデル (図 6-3) で示したように、内容品質を確保するには、先行して表現品質の確保が必要である。従来のインスペクションでは、これら二つの品質が区別されることなく同時に評価されていたため、内容品質の確保が効率的に行われていなかった。本研究では、1章で設計した SRS 品質モデルに基づき、SRS 表現の正当性確認 (verification) と内容の妥当性確認 (validation) を段階的に行う二段階のインスペクションプロセスを提案する。このように、正当性確認のプロセスで評価すべき品質特性と、妥当性確認のプロセスで評価すべき品質特性を明確に分離した SRS 品質モデルに基づき段階的なインスペクションを設計することで、SRS の正当性確認と妥当性確認、いわゆる V&V 活動を体系的に実施できる。

SRS V&V の効率的な実施方法として、二段階インスペクションの第 1 フェーズ (正当性確認プロセス) は第三者インスペクタが、第 2 フェーズ (妥当性確認プロセス) は開発プロジェクトチームがそれぞれ担務するプロセスを提案する。正当性確認プロセスを第三者インスペクタが担当することで、インスペクタの割り当てが容易となり、インスペクタの稼働率向上と負荷の平準化が期待できる。一方、多忙な開発チームとドメイン専門家は正当性確認済みの要求を対象にするため、妥当性確認の効率化が期待できる。これらの期待効果から、総合的なインスペクションコストの低減が期待できる。

本研究では、SRS V&V の 2 つのプロセスのうち、第三者インスペクタによる要求の正当性確認プロセスを主な研究対象とする。自動車 SRS 向け第三者インスペクションの設計は、企業情報システムの SRS を対象に開発されたインスペクション設計方法論である RISDM 手法を拡張することで行う。

#### 7.1.1 第三者インスペクタの定義

本研究における第三者インスペクタとは、対象 SRS が定義するソフトウェアの開発に直接関与していない「第三者」のことである。Saito らの研究における第三者インスペクション[58]の考えを基礎としている。

自動車ソフトウェア SRS をインスペクションする第三者インスペクタには、SRS が対象とする特定製品ドメインの知識は必要とされないが、自動車ソフトウェア全般に関する知識が求められる。例えば、エアコン制御ソフトウェアに関する SRS をインスペクションする場合、冷媒サイクルに関する知識は必要ではないが、シリアル通信、割込み処理といった製品ドメインによらず自動車ソフトウェア全般に広く利用される技術については理解していることを想定している。

#### 7.1.2 二段階インスペクションのプロセス

正当性確認プロセスである第三者による表現品質インスペクションと、妥当性確認プロセスであるプロジェクトチームおよびドメイン専門家による内容品質インスペクションから構成される二段階インスペクションのプロセスを図 7-1 に示す。

- (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 の内容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 の内容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) 内容品質インスペクション結果の判断
  - 7.a) 内容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) 内容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、内容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内容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。

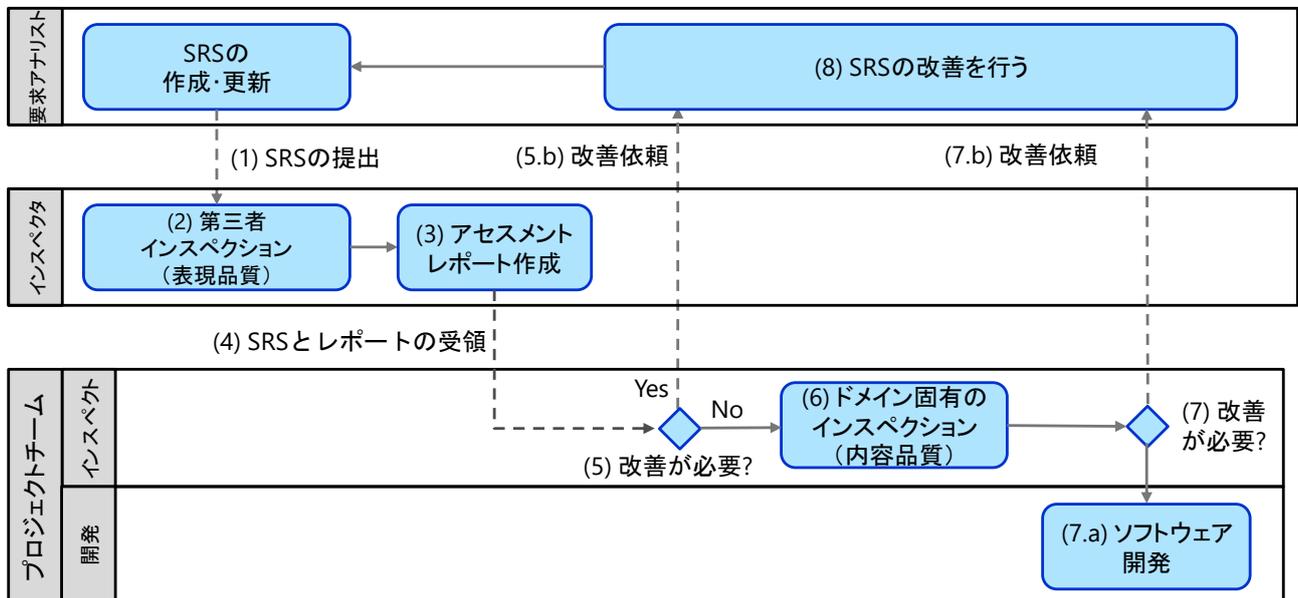


図 7-1 二段階インスペクションのプロセス

## 7.2 第三者インスペクション方法

### 7.2.1 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 7-2 の UML クラス図に示す。SRS の表現品質を定量的に評価するためには、SRS の構造を定義した SRS 構成モデルと表現品質特性が必要となる。SRS 構成モデルは 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 SRS 構成モデルの要求項目と測定すべき表現品質特性を関連付ける。質問セットはインスペクタが SRS を読解し表現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 SRS 構成モデルへ関連付ける。インスペクションマトリクスと質問セットの組み合わ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。

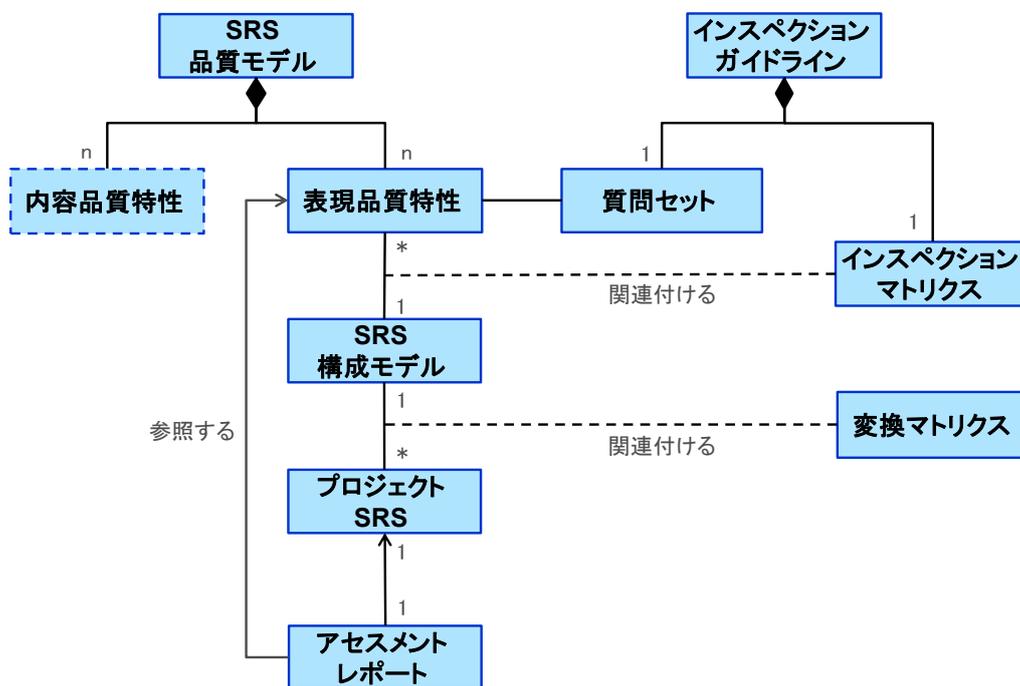


図 7-2 第三者インスペクションのメタモデル

### 7.2.2 SRS 品質スコアの測定方法

#### 7.2.2.1 SRS 構成モデル

自動車ソフトウェアの SRS を記述するための参照モデルとして、5 章で提案した SRS 構成モデルが利用できる。ただし、本稿の以降の提案方法の説明と適用評価では IEEE 830-1998 を SRS 構成モデルとして利用する。

#### 7.2.2.2 SRS 品質特性

自動車 SRS の正当性確認を行うための品質特性として、6 章で提案した SRS 品質モデルの表現品質特性を利用する。ソフトウェア品質保証活動における正当性確認とは「正しく製品を開発していること」の確認という説明がよくなされる。しかし、このような抽象的な概念では実際の正当性確認活動を行うには不十分である。そのため、本研究では 6 章で提案した表現品質特性の副特性への分解と詳細化を行う。各表現品質特性とその副特性を表 7-1 に示す。さらに、次節で説明する質問を各副特性へ関連付けることで SRS 内の文章が「正しく」表現されていることが確認できる。

表 7-1 表現品質特性と表現品質副特性

ID	表現品質特性	表現品質副特性
C1-1	Accountability 責任追跡性	Project objective 開発目的
C1-2		Purpose of feature 機能目的
C2-1	Definiteness 明確性	Non-redundant 非冗長
C2-2		Non-equivocal 多義的でない
C2-3		Quantative 定量的
C2-4		Freedom from ambiguity あいまい性の回避
C2-5		Voice 態
C2-6		Reference 参照
C3-1	Descriptive completeness 記述網羅性	TBD 未決定事項
C3-2		Label ラベル
C3-3		Template テンプレート
C4-1	Modifiable 変更容易性	Cross reference 相互参照
C4-2		Searchable 検索可能性
C5-1	Traceable 追跡可能性	Backward traceability 後方追跡性
C5-2		Forward traceability 前方追跡性

### 7.2.2.3 質問セット

表現品質副特性を測定するための具体的な方法として、各品質特性に関連するパースペクティブの所有者であるステークホルダでは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。各品質副特性に対応した 15 の質問からなる質問セットを表 7-2 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 7-2 質問セット

ID	品質特性	副特性	質問
C1-1	Accountability	Project objective 開発目的	製品の開発目的が記述されているか？ The objective of developing the software is stated?
C1-2	責任追跡性	Purpose of feature 機能目的	機能の目的が記述されているか？ The purpose of the function is stated?
C2-1	Definiteness 明確性	Non-redundant 非冗長	1つの文に複数の要求が記載されていないか？ Multiple requirements are not included in a single sentence?
C2-2		Non-equivocal 多義的でない	多義性を持つ用語が定義なしに使われていないか？ Any equivocal terms are used without defining its meanings?
C2-3		Quantative 定量的	定量的に記述すべき個所が定性的な記述になっていないか？ Qualitative expression is not used where quantitative expression
C2-4		Freedom from ambiguity あいまい性の回避	あいまい性の高い表現が使われていないか？ Ambiguous expressions are not used?
C2-5		Voice 態	能動態で記述されているか？ The sentence is written in the active voice?
C2-6		Reference 参照	外部文書を参照する場合は、参考文書欄に文書の場所を特定するのに十分な情報が記載されているか？ If external documents are referred, reference section provides enough information to locate the documents?
C3-1	Descriptive completeness 記述網羅性	TBD 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？ Each TBD is accompanied by a) How to resolve b) Due date c) Responsible person d) Tracking ID
C3-2		Label 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？ A unique number is assigned to figures and tables?
C3-3		Template 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？ All elements required by the standard SRS are included?
C4-1	Modifiable 変更容易性	Cross reference 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行われているか？ Cross-reference is provided to the appropriate requirement when a
C4-2		Searchable 検索可能性	目次と索引を作成しているか？ Table of contents and index are created?
C5-1	Traceable 追跡可能性	Backward traceability 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？ 上位要求との関連付けがされているか？（関連はSRS自体に記述せずトレーサビリティマトリックスを用いてもよい） Traceability matrix that specifies the link between high-level requirements and software requirements?
C5-2		Forward traceability 前方追跡性	各要求にユニークなIDが振られているか Each requirement has a unique identification?

### 7.2.2.4 インспекションマトリクス

一般に、SRS 構成モデル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спекションを効率化するため、参照 SRS の要求項目と測定すべき品質副特性を関連付けるインспекションマトリクスを定義する (表 7-3)。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спек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 の評価が必要なことを意味する。この例では、インспекションポイントは合計で 149 か所である。本研究では、各インспек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。

表 7-3 インспекションマトリクス

SRS構成モデル要求項目	品質副特性														
	C1-1	C1-2	C2-1	C2-2	C2-3	C2-4	C2-5	C2-6	C3-1	C3-2	C3-3	C4-1	C4-2	C5-1	C5-2
1. Introduction															
1.1 Purpose								X			X				
1.2 Scope	X							X			X				
1.3 Definitions, accronyms, 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 intrfaces				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				
2.1.8 Site adaptation requirements		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					
2.5 Assumptions and dependencies				X				X	X	X	X				
2.6 Apportioning of requirements				X				X	X	X	X				
3. Specific requirements															
3.1 Extenal interfaces				X				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	X				
3.2.1.3 Associated functional requirements			X	X	X	X	X	X	X	X					
3.3 Performance requirements			X	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	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	X
4. Supporting information													X		
4.1 Table of contents and index															
4.2 Appendixes															

### 7.2.2.5 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 の構成要素を SRS 構成モデルの構成要素へマッピングする。参照 SRS とプロジェクト SRS の対応付けは、プロジェクト SRS の要求項目と SRS 構成モデルの要求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 7-4）。変換マトリクス上で“X”が入っている個所が変換ポイントを表している。例えば、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 の要求項目ごとに該当する SRS 構成モデルの要求項目を識別する。

表 7-4 変換マトリクス

プロジェクトSRS 要求項目	SRS構成モデル要求項目																			
	4.2 Appendixes	4.1 Table of contents and index	3.6 Software system attributes	3.5 Design constraints	3.4 Logical database requirements	3.3 Performance requirements	3.2 Functions	3.1 External interfaces	2.6 Apportioning of requirements	2.5 Assumptions and dependencies	2.4 Constraints	2.3 User characteristics	2.2 Product functions	2.1 Product perspective	1.5 Overview	1.4 References	1.3 Definitions, acronyms, and abbreviations	1.2 Scope	1.1 Purpose	
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 参照文書																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

### 7.2.2.6 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み合わ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 の各要求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。

### 7.2.2.7 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 の表現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。

### 7.2.3 品質スコアの可視化方法

#### 7.2.3.1 品質の評価方法と可視化方法

品質スコアは非正規化スコアと正規化スコアの二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 の要求項目を SRS 構成モデルの要求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二つのビューを定義する。SRS 品質の評価基準とビューの全体像を図 7-3 に示す。品質スコア計算方法の詳細は 7.2.4 で説明する。

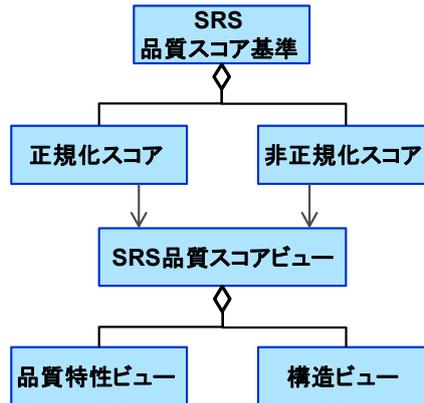


図 7-3 SRS 品質の評価基準とビューの全体像

この他に品質スコアの表現方法としてプロジェクト SRS 要求項目の SRS 構成モデルに対する網羅率を表す要求項目ヒストグラムと、品質特性および SRS 構成モデルの要求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。

以降の節では、8 提案方法の実際の SRS への適用の表 8-1 で示す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。

#### 7.2.3.2 非正規化スコア

品質特性ビュー (図 7-4) および副特性ビュー (図 7-5) 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わかる (図 7-4)。

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わかる (図 7-5)。

構造ビュー (図 7-6) 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、改善に着手すべき要求項目を特定する指針となる。SRS\_A では、6 から 8 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スペク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よりも相対的にスコアが低いことがわかる。

#### 7.2.3.3 正規化スコア

正規化スコアはプロジェクト SRS の要求項目を SRS 構成モデルの要求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 7-7) と品質副特性ビュー (図 7-8) および構造ビュー (図 7-9) から評価できる。正規化スコアの構造ビューからは、SRS 構成モデルの要求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 SRS 構成モデルの 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わかる。

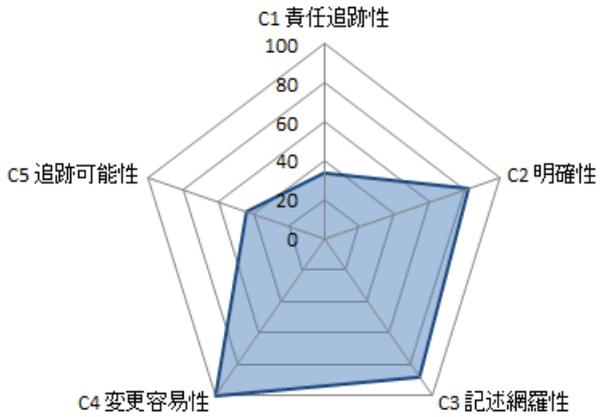


図 7-4 SRS\_A 非正規化スコア (品質特性ビュー)

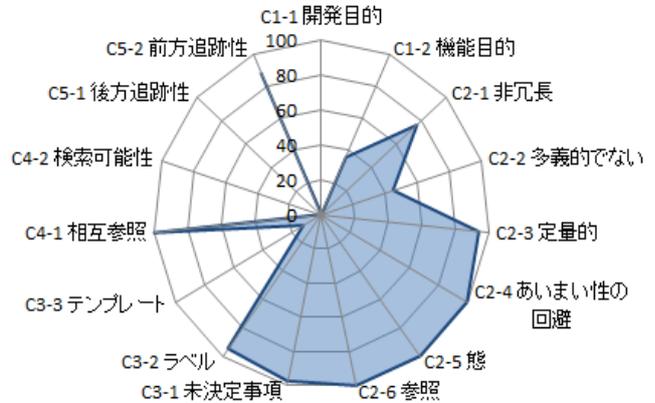


図 7-5 SRS\_A 非正規化スコア (品質副特性ビュー)

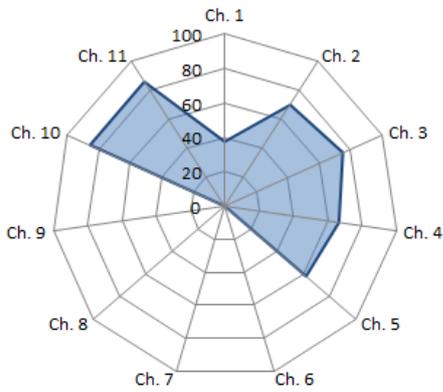


図 7-6 SRS\_A 非正規化スコア (構造ビュー)

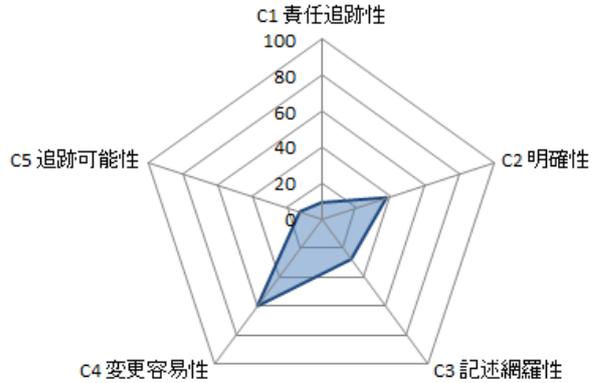


図 7-7 SRS\_A 正規化スコア (品質特性ビュー)

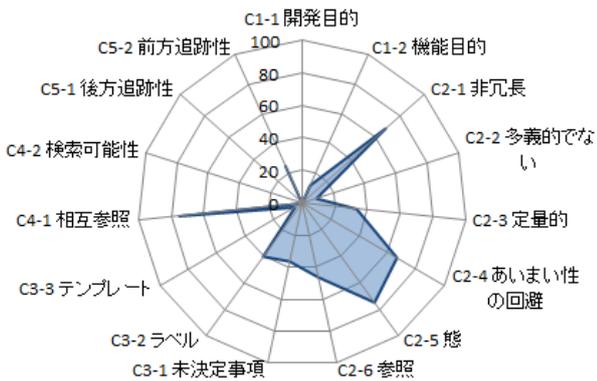


図 7-8 SRS\_A 正規化スコア (品質副特性ビュー)

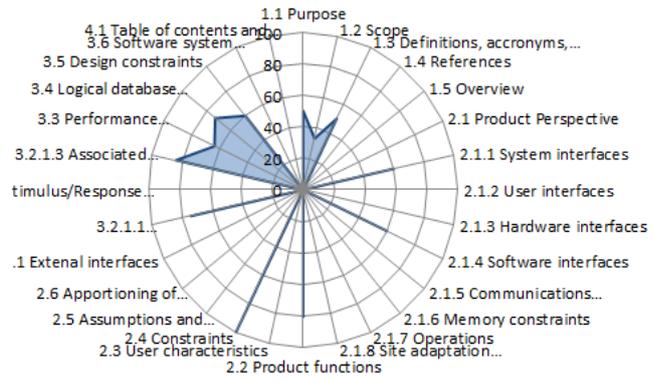


図 7-9 SRS\_A 正規化スコア (構造ビュー)

#### 7.2.3.4 非正規化スコアと正規化スコアの利用方法

##### (1) 非正規化スコアの利用方法

非正規化スコアは、例えば以下の場合に利用する。

- 1) 同一の要求項目を持つプロジェクト SRS 間の表現品質の比較
- 2) 同一プロジェクト SRS の改訂前後での表現品質の比較

##### (2) 正規化スコアの利用方法

一方、正規化スコアは、例えば以下の場合に利用する。

- 1) 異なる要求項目を持つプロジェクト SRS 間の表現品質の比較
- 2) 自動車 SRS にとってあるべき要求項目 (SRS 構成モデル) に対する充足度の評価

#### 7.2.3.5 要求項目ヒストグラム

SRS 構成モデルとは構造 (目次構成) が異なるプロジェクト SRS を評価するには変換マトリクスを作成する。変換マトリクスに出現する“X”の数を SRS 構成モデルの要求項目ごとに集計したのが要求項目ヒストグラム (図 7-10) である。正規化スコアからは SRS 構成モデルのどの要求項目に相当する部分のスコアが低いかが把握できる。しかし、スコアが低い原因が不十分な記述内容にあるのか、その SRS 構成モデルの要求項目に相当する要求項目がプロジェクト SRS には含まれていないのかを判別することはできない。要求項目ヒストグラムを用いることでその判断が可能となる。

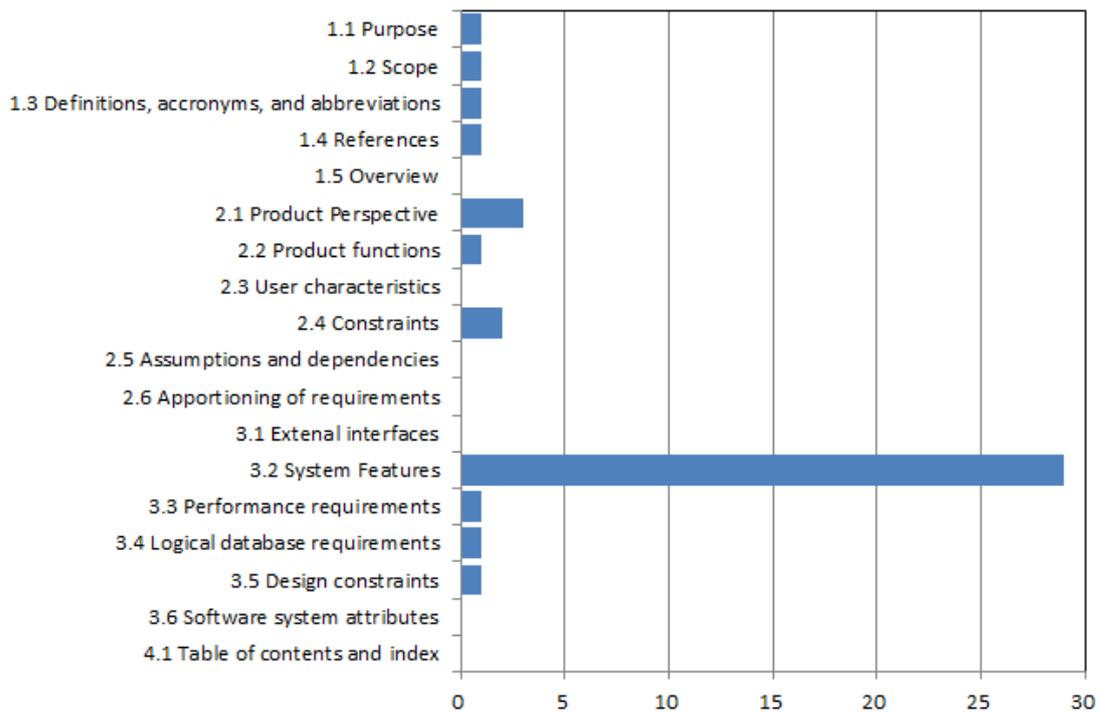


図 7-10 SRS\_A の要求項目ヒストグラム

#### 7.2.3.6 スコアヒートマップ

スコアヒートマップ (図 7-11) はインスペクションポイント全体での品質スコアの分布を表す。スコアヒートマップからは、正規化スコアの構造ビューと品質特性ビューからでは読み取れない、ある要求項目内での品質特性ごとのスコアの分布と、逆にある品質特性内での要求項目ごとの分布が評価できる。スコアヒートマップの各セルの数字はそのインスペクションポイントの品質スコアである。インスペクションの対象ではないセルには数字ではなくハイフンが示されている。

	1.1	1.2	1.3	1.4	1.5	2.1	2.1.1	2.1.2	2.1.3	2.1.4	2.1.5	2.1.6	2.1.7	2.1.8	2.2	2.3	2.4	2.5	2.6	3.1	3.2.1.1	3.2.1.2	3.2.1.3	3.3	3.4	3.5	3.6	4.1	4.2
C1-1	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
C1-2	-	-	-	-	-	-	-	-	-	-	-	0	0	-	-	-	-	-	-	-	-	36	-	-	-	-	-	-	-
C2-1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	73	100	100	-	0	-	
C2-2	-	-	-	-	-	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	36	0	67	0	0	0	0	0
C2-3	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	0	100	0	100	-	0	
C2-4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	100	100	100	100	0	
C2-5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	100	100	100	100	-	0	
C2-6	100	100	100	-	0	0	100	0	0	100	0	0	0	0	100	0	100	0	0	0	0	100	0	100	100	100	100	0	0
C3-1	-	-	-	-	-	0	100	0	0	100	0	0	0	0	100	0	100	0	0	0	0	100	0	100	100	100	0	0	0
C3-2	-	-	-	-	-	0	100	0	0	100	0	0	0	0	100	0	100	0	0	0	0	100	0	87	100	100	100	0	0
C3-3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	-	-	-	-	-	-	-	-	-	-	-	-	-
C4-1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	100	100	100	-	0	-	
C4-2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	
C5-1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	-	0	-	
C5-2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	100	0	0	-	0	-	

図 7-11 SRS\_A のスコアヒートマップ

## 7.2.4 品質スコアの計算方法

表 8-1 に示すデータセットから 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 7-5 の見方について説明する。表 7-5 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 7-3 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) の要求項目に関するインスペクション結果を表し、各列が表現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。

### 7.2.4.1 非正規化スコアの計算方法

#### 7.2.4.1.1 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 (7.1) を用いて計算する。

$$\text{品質スコア} = \frac{\text{●の数}}{\text{○の数} + \text{●の数}} \times 100 [\%] \quad (7.1)$$

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 (7.1) 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次のようになる。

$$5 / (10+5) = 33 [\%]$$

#### 7.2.4.1.2 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 (7.1) 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 (要求項目単位) となる。要求項目 1.1 を例にとると、品質スコアは次のように計算される。

$$1 / (1+1) = 50 [\%]$$

さらに、1章全体の品質スコアは次のよう計算される。

$$3 / (5+3) = 38 [\%]$$

## 7.2.4.2 正規化スコアの計算方法

### 7.2.4.2.1 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спекションポイントごとの得点の計算

あるインспек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спекションポイントに関連する SRS 構成モデル要求項目に対応するプロジェクト SRS 要求項目をすべて抽出する。続いて抽出したプロジェクト SRS 要求項目のインспекション結果のうち対象の品質副特性のみに対して式 (7.1) を適用し品質スコアを計算する。

機能目的 (C1-2) の場合、関連する SRS 構成モデル要求項目は「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 要求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 SRS 構成モデル要求項目に関するインспек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 要求項目が対応していることがわかる。それら14のプロジェクト SRS 要求項目に対して式 (7.1) を適用すると、このインспекションポイント (表現品質副特性 (C2-1) と SRS 構成モデル要求項目 (3.2.1.1) の交点) における品質スコアは次のように計算される。

$$5 / (9 + 5) = 36 [\%]$$

#### (2) 表現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спекションポイントごとのスコアが求まった。最終的な表現品質副特性の品質スコアはそれらインспекションポイントごとのスコアに以下の式 (7.2) を用いて計算する。

$$\text{品質スコア} = \frac{\sum \text{インспекションポイントでのスコア}}{\text{インспекションポイント数}} \times 100 [\%] \quad (7.2)$$

このようにインспекションポイントでのスコアの合計をインспекションポイントの数で割っているのは、関連するインспек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 (7.2) を適用すると品質スコアは次のようになる。

$$(0+0+36) / 3 = 12 [\%]$$

品質特性ビューの品質スコアも同じく式 (7.2) 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次のように計算される。

$$(0+0+0+36) / 4 = 9 [\%]$$

#### 7.2.4.2.2 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 要求項目に対応するプロジェクト SRS 要求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

要求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 SRS 構成モデル要求項目に関連する表現品質副特性に対して式 (7.1) を適用し品質スコアを計算する。

上記の手順を参照 SRS 要求項目「2.1.4 Software interfaces」に適用すると、プロジェクト SRS 要求項目のうち「4.3」と「4.4」が対応要求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 SRS 構成モデル要求項目「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 つ表現品質副特性に対して定義されていることがわかる。そのため、プロジェクト SRS 要求項目のうち「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 (7.1) を適用する。その結果、SRS 構成モデル要求項目「2.1.4 Software interfaces」の構造ビュー品質スコアは次のように計算される。

$$6 / (4 + 6) = 60 [\%]$$

表 7-5 SRS 品質スコアの算出方法

SRS記述モデル目次項目		インスペクションマトリクス		変換マトリクス		インスペクション結果	
1	1.1 機能要件	X		X			
1	1.2 性能要件		X		X		
1	1.3 運用要件			X			
1	1.4 制約要件				X		
2	2.1 機能要件	X	X	X	X	X	X
2	2.2 性能要件	X	X	X	X	X	X
2	2.3 運用要件	X	X	X	X	X	X
2	2.4 制約要件	X	X	X	X	X	X
3	3.1 機能要件	X	X	X	X	X	X
3	3.2 性能要件	X	X	X	X	X	X
3	3.3 運用要件	X	X	X	X	X	X
3	3.4 制約要件	X	X	X	X	X	X
4	4.1 機能要件	X	X	X	X	X	X
4	4.2 性能要件	X	X	X	X	X	X
4	4.3 運用要件	X	X	X	X	X	X
4	4.4 制約要件	X	X	X	X	X	X
5	5.1 機能要件	X	X	X	X	X	X
5	5.2 性能要件	X	X	X	X	X	X
5	5.3 運用要件	X	X	X	X	X	X
5	5.4 制約要件	X	X	X	X	X	X
6	6.1 機能要件	X	X	X	X	X	X
6	6.2 性能要件	X	X	X	X	X	X
6	6.3 運用要件	X	X	X	X	X	X
6	6.4 制約要件	X	X	X	X	X	X
7	7.1 機能要件	X	X	X	X	X	X
7	7.2 性能要件	X	X	X	X	X	X
7	7.3 運用要件	X	X	X	X	X	X
7	7.4 制約要件	X	X	X	X	X	X
8	8.1 機能要件	X	X	X	X	X	X
8	8.2 性能要件	X	X	X	X	X	X
8	8.3 運用要件	X	X	X	X	X	X
8	8.4 制約要件	X	X	X	X	X	X
9	9.1 機能要件	X	X	X	X	X	X
9	9.2 性能要件	X	X	X	X	X	X
9	9.3 運用要件	X	X	X	X	X	X
9	9.4 制約要件	X	X	X	X	X	X
10	10.1 機能要件	X	X	X	X	X	X
10	10.2 性能要件	X	X	X	X	X	X
10	10.3 運用要件	X	X	X	X	X	X
10	10.4 制約要件	X	X	X	X	X	X
11	11.1 機能要件	X	X	X	X	X	X
11	11.2 性能要件	X	X	X	X	X	X
11	11.3 運用要件	X	X	X	X	X	X
11	11.4 制約要件	X	X	X	X	X	X

## 8 提案方法の実際の SRS への適用

### 8.1 多様な自動車 SRS への適用性

製品ドメインによって特性の異なる多様な自動車 SRS に対し、提案方法を適用することでドメイン横断的にインスペクションが実施できるかを検証した。データセットの多様性を確保し検証の妥当性を高めるため、5つの製品ドメインから8つのSRSを採取し提案方法によるインスペクションを実施した。

#### 8.1.1 適用のコンテキスト

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[34]。

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。本研究では、そのうちの自然言語で記述された SRS を適用の対象とする。

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 SRS 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[72]。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。記述言語は日本語と英語である。

#### 8.1.2 データセット

適用を行った8つのSRSが対象とする製品ドメインとページ数を表8-1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は著者の所属する会社組織で開発した8つの製品(SRS\_AからSRS\_F)と、これから開発を行う製品(SRS\_GとSRS\_H)のSRSである。このうちSRS\_AとSRS\_Gのみが同一の開発組織によって作成されたSRSである。その他のSRSはすべて異なる開発組織によって作成された。異なる開発組織はそれぞれ独自のSRSテンプレートを定義している。また、同一組織内で作成されたSRS\_AとSRS\_Gも異なるテンプレートを基に作成されている。そのため、データセットに含まれる8つのSRSはすべて異なる構造(目次構成)を持っている。

表 8-1 インスペクション対象 SRS のデータセット

要求仕様書ID	製品ドメイン	ページ数
SRS_A	インフラ	76
SRS_B	ボディー	34
SRS_C	ボディー	39
SRS_D	アクティブセイフティー	41
SRS_E	利便快適	53
SRS_F	アクティブセイフティー	108
SRS_G	通信インフラ	19
SRS_H	パワートレイン	39

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。本研究の対象は、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。

### 8.1.3 検証結果

データセット（表 8-1）に挙げた 8 つの自動車製品の SRS に対し提案方法によるインスペクションを実施した結果を図 8-1 と図 8-2 に示す。8.1.2 でも述べたように、データセットに含まれる 8 つの SRS はすべて異なる構造を持つ。しかし、個別の構造を持つプロジェクト SRS を、変換マトリクスを利用して共通の SRS 構成モデルへマッピングすることで、共通のインスペクションガイドライン（7.2.2.6 参照）に沿ってインスペクションが可能となることが確認された。

SRS 品質スコアの評価基準として正規化スコアを定義したことにより、図 8-1 と図 8-2 に示すように構造の異なる SRS 同士の品質スコアを比較することができる。このように異なる開発組織間で SRS 品質スコアを比較できるため、正規化スコアの導入により、品質スコアの高い SRS を作成している開発組織が実施しているベストプラクティスの他の開発組織への横展開や、すべての開発組織で共通の SRS 品質基準を設定するといった活動が可能となる。

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設計することで、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った。また、事前に十分な表現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内容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。

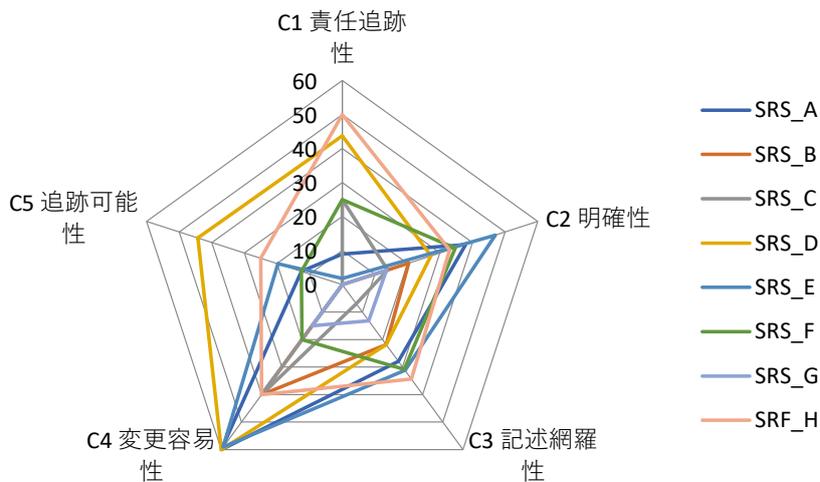


図 8-1 SRS 正規化スコア（品質特性ビュー）の比較

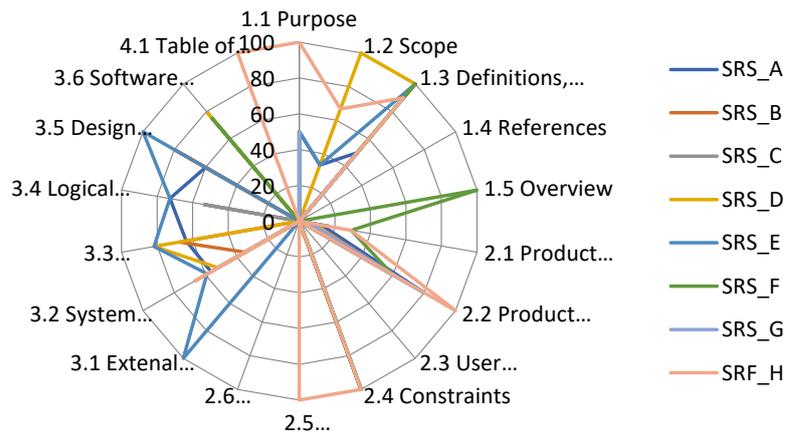


図 8-2 正規化スコア（構造ビュー）の比較

## 8.2 SRS 品質改善効果の検証

### 8.2.1 適用例1(ボディー製品ソフトウェア)

#### 8.2.1.1 検証方法

表 8-1 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) に対し本研究で提案するインスペクションによって作成されるアセスメントレポート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。インスペクション対象 SRS (SRS\_A と SRS\_G) の間の関係を図 8-3 に示す。

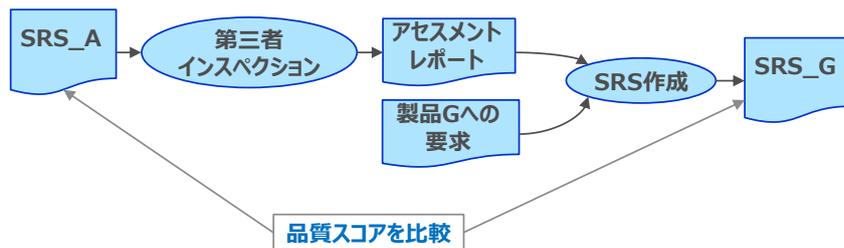


図 8-3 インスペクション対象 SRS 間の関係

#### 8.2.1.2 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 8-4 と図 8-5 に示す。

SRS\_A と SRS\_G は異なる構造を持つため、図 8-4 と図 8-5 の比較結果はともに正規化スコアによるものである。図 8-4 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わかる。次に図 8-5 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 SRS 構成モデルの要求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 8-6）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 8-6 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要求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言えない。これは、開発組織が SRS\_G の作成と併せて SRS 構造の見直しも同時に行ったため、プロジェクトの標準 SRS テンプレートの品質が未成熟なことが一因と考えられる。また、先行研究[59]では、1 回の SRS の改訂では必ずしも十分な改善が達成されないという調査結果も報告されている。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、本研究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、所期の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。

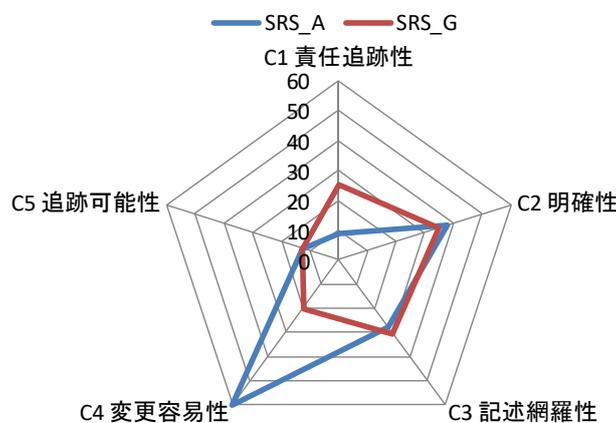


図 8-4 SRS 品質の比較（品質特性ビュー）

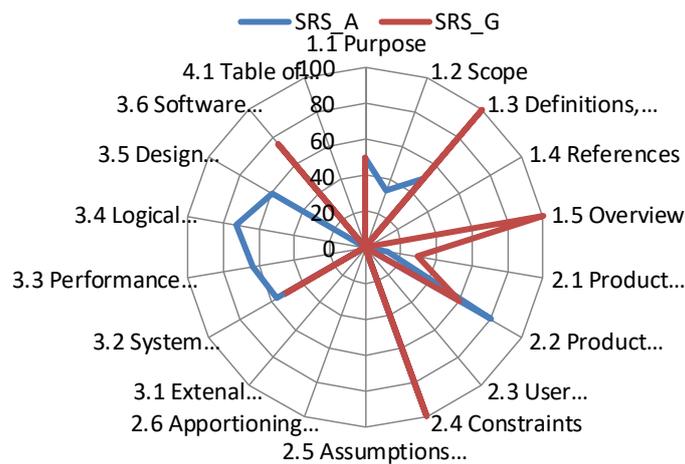


図 8-5 SRS 品質の比較（構造ビュー）

	1.1	1.2	1.3	1.4	1.5	2.1	2.1.1	2.1.2	2.1.3	2.1.4	2.1.5	2.1.6	2.1.7	2.1.8	2.2	2.3	2.4	2.5	2.6	3.1	3.2.1.1	3.2.1.2	3.2.1.3	3.3	3.4	3.5	3.6	4.1	4.2	
C1-1		0																												
C1-2													0	0								75								
C2-1																								27	100	100				
C2-2						0	100	0	0	0	100	100	0	0	0	0	0	0	0	0	0	39	0	-67	0	0	0	100		
C2-3												100											0	0	0	100		75		
C2-4																							0	-67	100	100	100		50	
C2-5																							0	100	100	100		100		
C2-6	-50	-100	0		100	0	0	0	0	100	100	100	0	0	0	0	0	0	0	0	0	0	0	0	100	100	100		100	
C3-1						0	0	0	0	100	100	100	0	0	0	0	0	0	0	0	0	0	0	0	100	100	0		100	
C3-2						0	0	0	0	100	100	100	0	0	0	0	0	0	0	0	0	0	0	0	100	100	0		100	
C3-3	50	0	100	0	100	0	0	0	0	0	100	0	0	0	100	0							0	13	100	100	100		100	
C4-1																								100	100	100		100		
C4-2																												0		
C5-1																								0	0	0		0		
C5-2																								0	0	0		0		

図 8-6 SRS 品質の比較 (スコアヒートマップ)

## 8.2.2 適用例 2(パワートレイン製品ソフトウェア)

### 8.2.2.1 検証方法

本研究で提案する第三者インスペクション方法の SRS 品質改善への効果を以下の手順で実施する。

- (1) SRS の第三者インスペクション実施
- (2) SRS 作成者へのアセスメントレポートの提供
- (3) SRS 作成者による SRS の改訂
- (4) 改訂された SRS の第三者インスペクション実施
- (5) 改訂前後での SRS 品質スコアの比較

### 8.2.2.2 検証結果

図 8-7 に示すようにすべての品質副特性で品質スコアが向上した。また、SRS 作成者へ実施したアンケートでは、以下の肯定的なコメントが得られた。

- 1) 優れている点と改善点が把握できたため、非常に役に立った
- 2) 第三者インスペクションにより要求仕様書内にあいまいな要求が残っていることに気付けた
- 3) 品質特性のスコアは概ね体感と一致しており、大きな乖離は感じなかった

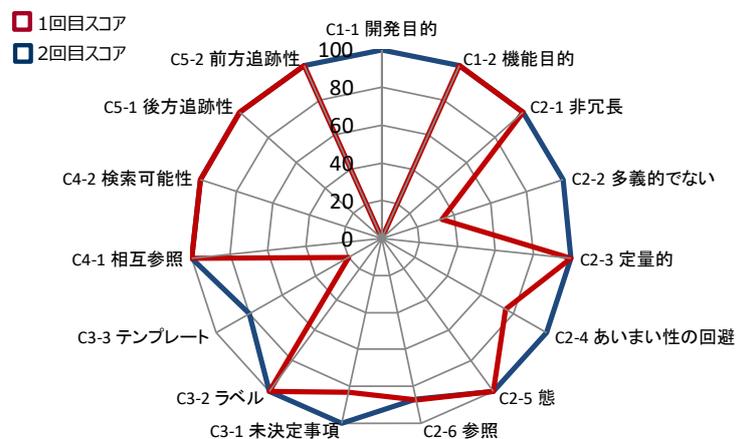


図 8-7 品質副特性毎の非正規スコア改善度 (実測値)

図 8-8 は、SRS の作成者に改訂前の SRS 品質スコアを基に予測してもらった改訂後の SRS 品質スコア予測値と実際のインスペクションによって得られた品質スコア実測値の比較結果である。比較結果を見ると、C3-1（未決定事項）と C3-3（テンプレート）に 20%程度の誤差はあるが、SRS 作者の予測値とインスペクション結果値は非常によく合致している。この結果は上記 3) のコメントと一致している。このように SRS 作者の体感とインスペクションによって得られた実際のスコアに高い相関があることは、提案手法による SRS 品質のスコアリング方法の妥当性を裏付ける一つの根拠といえる。

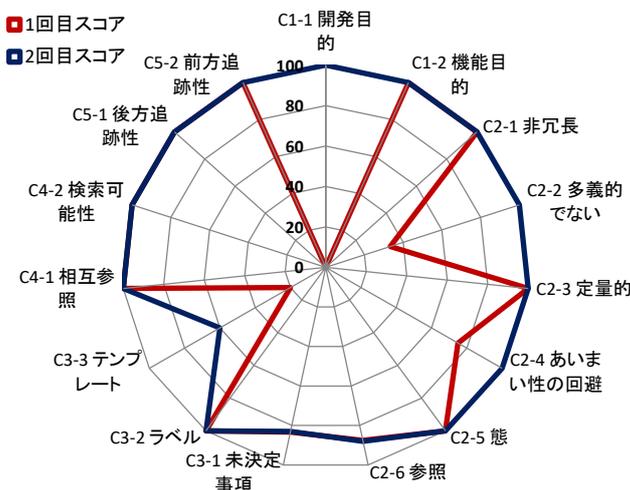


図 8-8 品質副特性毎の非正規スコア改善度（SRS 作者による予測値）

### 8.2.3 SRS 品質改善効果のまとめ

適用例 1 と 2 の結果から、第三者インスペクションによる SRS 品質の改善効果について以下のことが言える。

#### (1) SRS 品質に対する定量的な判断基準の提供

第三者インスペクションによって作成されるアセスメントレポートは、改訂による SRS 品質改善の結果を客観的に判断するための基準を提供する。改訂による SRS 品質の変化を従来の SRS 作成者とレビューアの主観に基づく曖昧な基準ではなく、SRS 品質スコアという客観的な基準で判断できるため、改訂により目標通りの品質改善が達成できたかを明確に判断できる。また、適用例 1 のように所望の改善効果が得られなかった場合も、再度の改訂により改善すべき品質特性と要求項目が把握できる。そのため、本研究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、所期の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。

#### (2) SRS 品質の向上

第三者インスペクションによって作成されるアセスメントレポートは SRS 改善のためのアドバイスを提供する。そのため、適用例 1 のように 1 度の改訂では十分な品質向上が達成できないことはあるが、SRS 品質スコアに基づく客観的な判断のもとでは、SRS の改訂を繰り返すことでその品質が向上すると考えられる。

## 8.3 手戻工数削減効果の検証

### 8.3.1 検証方法

開発完了した製品の中から要求品質の不足により設計手戻りが発生したものを対象に二段階インスペクションによる手戻り工数削減効果を試算した。この製品開発では設計担当者と要求定義担当者との間で SRS の疑問点に対する問い合わせと回答のやり取り（QA）が 175 件行われた。そのうち 50 件により要求の問題が明らかとなり、120 時間の設計の手戻りが発生した。

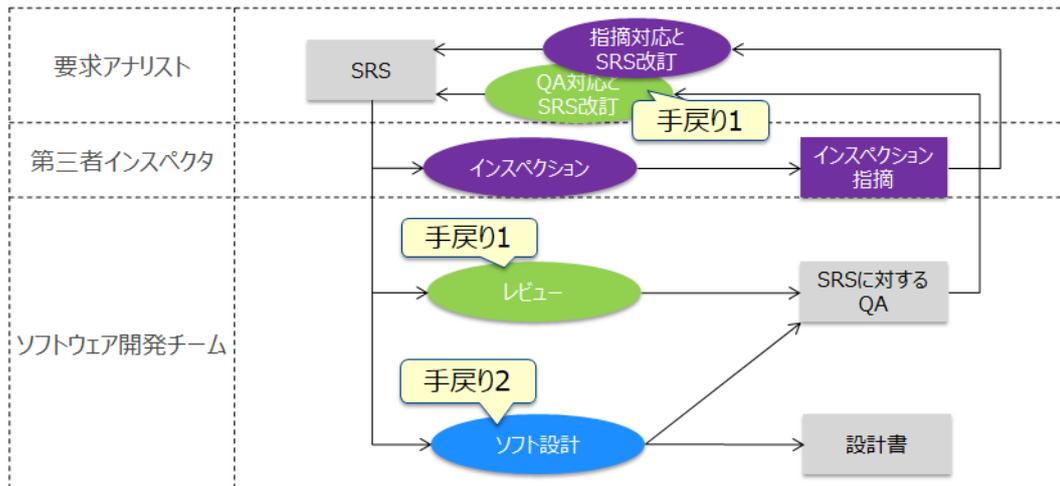


図 8-9 手戻工数削減効果の検証手順

### 8.3.2 検証結果

この SRS に対し、図 8-9 に示すように設計担当者によるレビューに先立ち、第三者インスペクションを実施した場合の工数削減効果を以下の式 (8.1) から算出した。

$$\text{工数削減効果} = \frac{\text{インスペクションで指摘できた QA 数}}{\text{開発時に発生した QA の総数}} \times 100 [\%] \quad (8.1)$$

第三者インスペクションにより指摘できた QA は全体の 23.4%であった。設計手戻りに至った QA に限定すると指摘率は 20.0%であった。そこから、QA1 件あたりの手戻り工数を 2 時間、すべての設計手戻り作業は同じ工数を要したと仮定し、工数削減効果を計算すると 106 時間となる。一方、第三者インスペクションの工数は 58 時間であった。工数削減効果を評価するために、ROI(Return On Investment) を式 (8.2) で定義する。

$$\text{ROI} = \frac{\text{削減工数} - \text{インスペクション工数}}{\text{インスペクション工数}} \times 100 [\%] \quad (8.2)$$

式 (8.2) よりインスペクションの費用対効果 (ROI) を計算すると次のようになる。

$$(106 - 54) \div 54 \times 100 = 82.8 [\%]$$

これは、二段階のインスペクションを実施することで、要求分析工程と設計工程での総工数を低減できることを示している。



図 8-10 第三者インスペクションによる工数削減効果

## 9 研究課題に対する評価

### 9.1 ステークホルダの関心事に基づく SRS 構成モデルの設計方法

- (1) 要求項目メタモデルの提案  
ある製品ドメインに特化した SRS のメタモデルを定義することで、多様なドメイン要求の構造を明らかにした。
- (2) SRS 構成モデル設計方法の提案  
メタモデルの要素間の対応関係を展開表として表し、展開表同士の導出関係を関係モデルとして定義することで、関係モデルに基づいたプロセスに沿って体系的にドメイン固有の SRS 構成モデルを設計することが可能となった。
- (3) 自動車ソフトウェアに適した SRS 構成モデルの提案  
提案する SRS 構成モデル設計方法に沿って多様な要求を持つ自動車ソフトウェアのための SRS 構成モデルを設計した。また、提案方法をイテレーティブに適用することで、今後も複雑さと多様性を増すと予想される自動車ソフトウェアの要求に対し、それらを表現できるよう SRS 構成モデルを継続的に発展させていくことができる。

### 9.2 パースペクティブに基づく SRS 品質モデルの設計方法

- (1) 自動車ソフトウェア参照 SRS 品質モデルの提案  
国際規格 IEEE 830-1998 が定義する SRS 品質特性を基に、自動車ソフトウェアの特性を考慮した参照 SRS 品質特性を定義したことで、明確に定義された基準に基づき自動車 SRS の品質を測定することが可能となった。
- (2) プラグマティック品質特性 (PQC) の提案  
プラグマティック品質という抽象的な概念をプラグマティック品質特性として具体化したことで、多様な自動車ソフトウェアのステークホルダにとっての SRS の理解しやすさが判断できるようになった。
- (3) 要求の正当性確認と妥当性確認に層別した SRS 品質モデルの提案  
本研究で提案する SRS 品質モデルにより、要求の正当性確認プロセスで評価すべき PQC と、要求の妥当性確認のプロセスで評価すべき PQC が明確になった。これにより体系的な SRS の正当性確認と妥当性確認が可能となる。

### 9.3 SRS 品質モデルに基づく自動車 SRS の第三者インスペクション方法

- (1) 二段階インスペクションプロセスの提案  
正当性確認と妥当性確認を段階的に行うことで体系的な SRS の品質保証活動 (V&V) が行えるようにした。また、要求の正当性確認と妥当性確認を第三者インスペクタと開発プロジェクトチームがそれぞれ担務することで、人的リソースの効率的な割り当てと負荷の平準化、およびインスペクション効率向上によるコストの低減が可能となった。  
開発が完了した過去プロジェクトに対し、二段階インスペクションを導入した場合の効果を試算した。その結果、費用対効果にすると 82.8% の開発工数削減効果があることが確認された。
- (2) 第三者インスペクション方法の提案  
製品ドメインの知識を持たない第三者による多様な自動車 SRS のインスペクタを可能とした。提案方法を用いて複数の実製品の SRS をインスペクションした。その結果、構成の異なる多様な自動車 SRS を統一的にインスペクションできることが確認された。また、インスペクション結果から作成されるアセスメントレポートは SRS の品質向上に効果があることが明らかとなった。

## 10 考察

8 章での提案方法の適用結果に基づき関連研究との比較を行うことで、本研究の技術的貢献について考察する。

### 10.1 ステークホルダの関心事に基づく SRS 構成モデルの設計方法

#### (1) SRS 構成モデル

多様なステークホルダから要求を獲得し、それらを表現するためには、適切な構造を持った SRS 構成モデルが必要である。関連研究でも述べたように、SRS 構成モデルを提案する多くの文献があるが、その多くは経験的に重要と思われる要求項目を列挙した汎用的なもので、特定のドメインへそのまま適用するには不適切である。しかし、特定ドメインに適した SRS 構成モデルの設計に関する研究は少ない。複数の SRS 構成モデルからベースとなるモデルを選択するための基準を提案する研究[18]はあるが、選択した参照 SRS 構成モデルから派生 SRS 構成モデルを設計する方法については未確立である。

本研究は、ドメイン特性に関するステークホルダの関心事を多様な要求の源泉ととらえ、要求項目のメタモデルを定義した。既存研究では未確立であった、参照 SRS 構成モデルに対する過不足を明らかにし、ドメイン固有の SRS 構成モデルを設計する体系的な方法は新規性があると考えられる。

#### (2) 自動車ソフトウェアに適した SRS 構成モデル

本研究で調査した範囲では、自動車ソフトウェアのための SRS 構成モデルを提案した研究はない。本研究の提案方法に基づき構築した自動車向け SRS 構成モデルは、著者の所属する以外の自動車ソフトウェア開発組織にとっても有用であると考えられる。

### 10.2 パースペクティブに基づく SRS 品質モデルの設計方法

#### (1) ステークホルダのパースペクティブを考慮した SRS 品質特性

SRS に記述された要求は、読者が正しくその内容を理解できてはじめて意味がある。SRS 品質モデルを提案する多くの文献があるが、その多くは SRS 構成モデルと同様、経験的に重要と思われる品質を列挙したものであり、読者であるステークホルダにとっての理解のしやすさという観点は考慮されていない。

本研究は、SRS の読者であるステークホルダのパースペクティブに基づくプラグマティック品質特性の研究[58, 59]に基づき、自動車ソフトウェアという多様なパースペクティブを含む SRS への応用へ発展させたものであり、ステークホルダによる SRS の理解しやすさの観点から SRS 品質を評価できる。

#### (2) 要求の正当性確認と妥当性確認を層別した SRS 品質モデル

本研究で調査した範囲では、要求の品質特性を正当性確認と妥当性確認の観点から分類した研究はない。よって、本研究が提案する要求の正当性確認と妥当性確認を層別した SRS 品質モデルは新規性が高いと考える。

### 10.3 SRS 品質モデルに基づく自動車 SRS の第三者インスペクション方法

#### (1) 二段階インスペクションプロセス

N 重インスペクション[40]や段階的インスペクション[32]など多くのインスペクション実施プロセスが提案されている。しかし、SRS の正当性確認と妥当性確認は区別されておらず、2 つの異なる品質保証活動が不可分となっている。SRS の正当性確認と妥当性確認を明確に分離し、体系的に実施するプロセスは新規性が高いと考える。

#### (2) 第三者インスペクション方法

企業情報システムの SRS に対する第三者インスペクション方法である RISDM[59]を自動車ソフトウェアの多様な SRS 構成モデルがインスペクションできるよう拡張した。これにより、本研究の提案方法は、自動車ソフトウェアに限らず他ドメインの組込みソフトウェアの SRS インスペクションにも適用できる汎用性を有している。

## 11 今後の課題

本研究を発展させていくために、次の3点を今後の課題としてあげる。

### 11.1 ステークホルダの変化への対応

本研究では、ステークホルダの関心事に基づきある製品ドメインに適した SRS 構成モデルを新規に設計する方法と、ステークホルダのパースペクティブを考慮した SRS 品質モデルの設計方法を提案した。

自動車のようにモデルチェンジのたびに機能の更新や追加が起こる進歩の早いドメインでは、開発対象ソフトウェアの発展とともに関係するステークホルダが変化する。ステークホルダが変化すると SRS が表現すべき要求も変化する。そのため、SRS 構成モデルの陳腐化を防ぐには、対象ソフトウェアの発展にあわせて再設計する必要がある。

ステークホルダの変化は、SRS が表現すべき要求に加え、SRS が満たすべきプラグマティック品質特性にも影響を及ぼす。そのため、SRS 構成モデルと同じく SRS 品質モデルも対象ソフトウェアの発展にあわせて再設計する必要がある。

体系的かつ効率的に SRS 構成モデルと SRS 品質モデルを再設計するための方法と実施プロセスを確立できれば、継続的に発展するソフトウェアに対して有効なモデルを維持することが期待できる。

### 11.2 インспекションの対象範囲拡大と独立性の向上

本研究では、SRS 記述表現の正当性確認による表現品質の評価と、SRS 記述内容の妥当性確認による内容品質の評価を段階的に行う二段階のインспекションプロセスを提案した。そのうち、1段階目のプロセスである SRS 記述表現の正当性確認の具体的な実施方法として第三者インспекションを設計した。SRS 品質全体を評価するためには、表現品質の正当性確認に加え、内容品質の妥当性確認を体系的に行う必要がある。2段階目の妥当性確認プロセスについても具体的なインспекション方法を設計することで、SRS の V&V フレームワーク全体が体系化され、SRS 品質全体が評価されることを担保することが期待できる。

本研究では妥当性確認のプロセスは開発プロジェクトチームが実施するとしている。しかし、航空宇宙や自動車のような高い安全性を求められる分野では、品質保証活動の独立性が強く求められるようになってきている。正当性確認プロセスに加え妥当性確認プロセスも第三者インスペクタによって実施可能な方法として体系化することで、V&V をさらに進めた品質保証活動である IV&V (Independent Verification and Validation) の実現が期待できる。

### 11.3 設計方法論の他分野への適用

本研究では、提案した SRS の品質モデル設計方法論を自動車ソフトウェアシステムの SRS に適用しその有効性を評価した。設計方法論をさらに発展させるには、他分野への適用と評価が必要である。

自動車ソフトウェアを含む組込みソフトウェア開発と、Web アプリケーションソフトウェア開発、企業情報システム開発では適用される開発プロセスが大きく異なる。自動車ソフトウェアシステム製品の標準的な開発は V 字型の開発プロセスによって行われる。一方、Web アプリケーションソフトウェア開発ではアジャイル開発が広く普及している。アジャイル開発では、要求の分析定義とソフトウェアの実装開発が並行して行われる。そのため、V 字型プロセスに基づく開発のように、設計フェーズへの移行前に SRS の品質を確認する明確な品質ゲートを設置することはできない。このようなインクリメンタル型の開発プロセスを採用する他分野への適用により、継続的な SRS 品質の評価と改善が可能な設計方法論へと発展させることで、より多くの分野への貢献が期待できる。

## 12 まとめ

### 12.1 研究の背景と課題

自動車ソフトウェアの発展とともに、その目的がアナログ制御のデジタル制御への置き換えから、ソフトウェアでのみ実現できる付加価値の創造へと変化しつつある。インターネットへの接続サービスや高度運転支援など新たな機能が加わることで、自動車ソフトウェアに対する要求がこれまで以上に多様化している。しかし、多様化する自動車ソフトウェアへの要求を高い品質を備えた要求仕様書へと変換する技術は十分に確立されているとは言えない。

以上の背景から、本研究は下記の課題を解決することで自動車ソフトウェア要求仕様書の品質向上に貢献することを目的とした。

- (1) 多様な要求の源泉であるステークホルダの関心事に基づく SRS 構成モデル設計方法の確立と、提案方法による自動車ソフトウェアに適した SRS 構成モデルの設計方法
- (2) SRS 読者のパースペクティブに基づいた SRS 品質特性の設計方法
- (3) SRS に対する V&V 実施方法の確立

### 12.2 課題解決

課題解決のために下記の技術を提案した。(2)と(3)は実際のプロジェクトに適用することでその有用性を評価した。

- (1) ステークホルダの関心事に基づく SRS 構成モデルの設計方法  
多様な要求の構造を明らかにする要求項目メタモデルを定義し、メタモデルの構成要素を体系的に具体化していくことで SRS 構成モデルを設計する方法を提案した。
- (2) パースペクティブに基づく SRS 品質モデルの設計方法  
SRS 読者であるステークホルダのパースペクティブを考慮したプラグマティック品質特性の定義と、それらの品質特性を正当性確認対象の表現品質と妥当性確認対象の内容品質に層別した SRS 品質モデルを提案した。
- (3) SRS 品質モデルに基づく自動車 SRS の第三者インスペクション方法  
提案する SRS 品質モデルに基づき SRS の V&V を体系的に行うための二段階インスペクションプロセスを提案し、V&V フレームワークのうち正当性確認プロセスを効率的に実施するための具体的な方法として第三者インスペクションを設計した。

### 12.3 本研究の貢献

本研究で提案した SRS 構成モデルの設計方法は、今後ますます自動車ソフトウェアへの要求が多様化していく中で、ステークホルダのドメイン特性に関する関心事に着目し、多様な要求の構造を明らかにしたことに意義がある。自動車ソフトウェアと同様に多様なステークホルダが関わる他の分野へ適用され、今後の議論が進展することが期待できる。

正当性確認のための品質特性と妥当性確認のための品質特性を明確に層別した SRS 品質モデルを提案することで、これまで曖昧だった SRS の正当性確認と妥当性確認を独立したプロセスとして実施することを可能とした。

自動車 SRS の第三者インスペクションは、SRS 品質の改善に有効なアドバイスを提供し、インスペクション実施工数に対し 82.8%の費用対効果 (ROI) で開発工数の削減に寄与することが確認された。

## 謝辞

本研究は，南山大学大学院理工学研究科ソフトウェア工学専攻青山研究室において，青山幹雄教授のご指導の下に実施されたものです．本研究の遂行にあたり，終始手厚いご指導，ご教授を賜りました青山幹雄教授に心より厚く御礼申し上げます．

また，本論文の審査を頂きました南山大学大学院理工学部ソフトウェア工学専攻，野呂昌満教授，阿草清滋教授におかれましては，本論文について詳細なアドバイスを頂きました．ここに深甚な謝意を表します．

本研究の遂行にあたり，評価題材のソフトウェア要求仕様書の提供と，提案方法の効果検証に協力していただきました株式会社デンソーの関係各位に深く感謝の意を表します．

最後に，日々の研究活動を心身両面に渡って支えてくれた家族に心から感謝します．

## 参考文献

- [1] D. Aceituna, Survey of Concerns in Embedded Systems Requirements Engineering, SAE Int'l J. of Passenger Cars- Electronic and Electrical Systems, Vol. 7, No. 1, May 2014, pp. 1-7.
- [2] I. Alexander, A Taxonomy of Stakeholders, Human Roles in System Development. Issues and Trends in Technology and Human Interaction, 2006, pp. 25-71.
- [3] C. Allmann, L. Winkler, T. Kölzow, et al. The requirements engineering gap in the oem-supplier relationship. Journal of Universal Knowledge Management, 1(2), 2006, pp. 103–111.
- [4] International Workshop on Automotive Requirements Engineering (AuRE 2004), <http://app.nise.org/AuRE/AuRE2004top.htm> (accessed on Jan. 2018).
- [5] A. Aurum, H. Petersson, and C. Wohlin, State-of-the-Art: Software Inspections after 25 Years, J. of Software Testing Verification and Reliability, Vol. 12, No. 3, Sep. 2002, pp. 133-154.
- [6] V.R. Basili, S. Green, O. Laitenberger and et al., The Empirical Investigation of Perspective-Based Reading, Empirical Software Engineering, Vol. 1, No. 2, Jan. 1996, pp. 133-164.
- [7] P. Braun, et al., Guiding Requirements Engineering for Software-Intensive Embedded Systems in the Automotive Industry, Computer Science - Research and Development, Vol. 29, No. 1, Springer, Feb. 2014, pp. 21-43.
- [8] M. Ciolkowski, C. Differding, O. Laitenberger and et al., Empirical Investigation of Perspective-Based Reading : A Replicated Experiment, Technical Report No. 13/97, Int'l Software Engineering Research Network (ISERN), 1997, 28 pages.
- [9] M. Broy, Challenges in automotive software engineering, Proc. of ICSE 2006, IEEE Computer Society, May 2006, pp. 33–42.
- [10] B. H. C. Cheng and J. M. Atlee, Research Directions in Requirements Engineering, Proc. of FOSE 2007 (Future of Software Engineering), IEEE Computer Society, May 2007, pp. 285-303.
- [11] DoD, DI-IPSC-81433A, Data Item Description: Software Requirements Specification, Dec. 1999.
- [12] C. Ebert and C. Jones, Embedded Software: Facts, Figures, and Future, IEEE Computer, Vol. 42, No. 4, Apr. 2009, pp. 42-52.
- [13] ESA Software Engineering Standards, ESA PSS-05-0 Issue 2 Revision 1, European Space Agency, ESA Publications Division, 1994.
- [14] F. Fabbrini, M. Fusani, V. Gervasi and et al., Achieving Quality in Natural Language Requirements, Proc. of 11<sup>th</sup> Int'l Software Quality Week Conference (QW 1998), May 1998, 17 pages.
- [15] M. Fagan, Design and Code Inspections to Reduce Errors in Program Development, IBM Systems J., Vol. 15. No. 3, Sep. 1976, pp. 182-211.
- [16] G. Fanmuy, et al., Requirements Verification in the Industry, Proc. of CSDM 2011, Springer, Dec. 2011, pp. 145-160.
- [17] G. Fanmuy and G. Foughali, A Survey on Industrial Practices in Requirements Engineering, INCOSE International Symposium, Vol. 22, No. 1, Jul. 2012, pp. 1021-1040.
- [18] E. A. Giakoumakis and G. Xylomenos, Evaluation and Selection Criteria for Software Requirements Specification Standards, Softw. Eng. J., Vol. 11, No. 5, 1996, pp. 307-319.
- [19] F. Houdek and S. Schmerler, Automotive Future and its Impact on Requirements Engineering., Proc. of REFSQ Work., 2017.
- [20] F. Houdek and M. Weber, Future Trends in Automotive Requirements Engineering, INCOSE Int. Symp., Vol. 14, No. 1, 2004, pp. 1830-1843.
- [21] IEEE, Std. 830-1998: IEEE Recommended Practice for Software Requirements Specifications, IEEE Computer Society, 1998.
- [22] IEEE, Std. 1012-2004: IEEE Standard for Software Verification and Validation, IEEE Computer Society, 2004
- [23] IEEE, Std. 1028-2008: IEEE Standard for Software Reviews and Audits, IEEE Computer Society, 2008.
- [24] ISO/IEC, ISO/IEC 20246:2017, Software and Systems Engineering – Work product reviews,

2017.

- [25] ISO/IEC, ISO/IEC 25010:2011, Systems and Software Engineering- Systems and Software Quality Requirements and Evaluation (SQuaRE)- Systems and Software Quality Models, 2011.
- [26] ISO/IEC, ISO/IEC 25030:2007, Software Engineering - Software Product Quality Requirements and Evaluation (SQuaRE) - Quality Requirements, 2007.
- [27] ISO/IEC/IEEE, ISO/IEC/IEEE 29148:2011 - Systems and Software Engineering - Life Cycle Processes - Requirements Engineering, ISO, 2011.
- [28] INCOSE Requirements Working Group, Guide for Writing Requirements Version/Rivision: 2.1, INCOSE, Jun. 2017
- [29] JISA REBOK 企画 WG, 要求工学知識体系 第1版, 近代科学社, 2011
- [30] E. Kantorowitz, A. Guttman, and L. Arzi, The Performance of the N-Fold Requirement Inspection Method, *J. of Requirements Engineering*, Vol. 2, No. 3, Sep. 1997, pp. 152-164.
- [31] E. Knauss and D. Ott, (Semi-) automatic Categorization of Natural Language Requirements, *Proc. of REFSQ 2014*, LNCS Vol. 8396, Springer, Apr. 2014, pp. 39-54.
- [32] J. C. Knight, and E. A. Myers, An Improved Inspection Technique, *CACM*, Vol. 36, No. 11, Nov. 1993, pp. 51-61.
- [33] S. Konrad and B.H.C. Cheng, Real-Time Specification Patterns, *Proc. of ICSE 2005*, IEEE Computer Society, May 2005, pp. 372-381.
- [34] J. Krisch, and F. Houdek, The Myth of Bad Passive Voice and Weak Words: An Empirical Investigation in the Automotive Industry. *Proc. of RE 2015*, IEEE Computer Society, Aug. 2015, pp. 344-351.
- [35] J. Krogstie, O. I. Lindland and G. Sindre, Towards a Deeper Understanding of Quality in Requirements Engineering, *Proc. of CAiSE 1995*, LNCS Vol. 932, Springer, 1995, pp. 82-95.
- [36] V. Langenfeld, et al., Requirements Defects over a Project Lifetime: An Empirical Analysis of Defect Data from 5-Year Automotive Project at Bosch, *Proc. of REFSQ 2016*, LNCS Vol. 9619, Springer, Mar. 2016, pp. 145-160.
- [37] O. I. Lindland, G. Sindre, and A. Srølverg, Understanding Quality in Conceptual Modelling. *IEEE Software*, Vol. 11, No. 2, Apr. 1994, pp. 42-49.
- [38] S. McConnell, The Best Influences on Software Engineering, *IEEE Software*, Vol. 17, No. 1, Jan./Feb. 2000, pp. 10-17.
- [39] J. C. Maldonado, J. Carver, F. Shull and et al., Perspective-Based Reading: A Replicated Experiment Focused on Individual Reviewer Effectiveness, *J. of Empirical Software Engineering*, Vol. 11, No. 1, Springer, Mar. 2006, pp. 119-142.
- [40] J. Martin, and W. T. Tsai, N-Fold Inspection: A Requirements Analysis Technique, *CACM*, Vol. 33, No. 2, Feb. 1990, pp. 225-232.
- [41] NASA, NASA-STD-2100-91, NASA Software Documentation Standard, NASA Headquarters Software Engineering Program, Jul. 1991.
- [42] R. Oshana and M. Kraeling, *Software Engineering for Embedded Systems: Methods, Practical Techniques, and Applications*, Newnes, May 2013.
- [43] D. Ott, Defects in Natural Languages Requirements Specifications at Mercedes Benz: An Investigation Using a Combination of Legacy Data and Expert Opinion, *Proc. of RE 2012*, IEEE Computer Society, Sep. 2012, pp. 291-296.
- [44] D. Ott, Automatic Requirement Categorization of Large Natural Language Specifications at Mercedes-Benz for Review Improvements, *Proc. of REFSQ 2013*, LNCS Vol. 7830, Springer, Apr. 2013, pp. 50-64.
- [45] D. Ott, F. Houdek, Automatic Requirement Classification: Tackling Inconsistencies Between Requirements and Regulations, *Int. J. of Semantic Computing*, Vol. 8 No. 1, World Scientific, Mar. 2014, pp. 47-66.
- [46] D. Ott, A. Raschke, Evaluating Benefits of Requirement Categorization in Natural Language Specifications for Review Improvements, *Open Access Repository der Universität Ulm*, Oct. 2013, pp. 1-10.
- [47] K. Pohl. The Three Dimensions of Requirements Engineering: A Framework and Its Applications, *Information Systems*, Vol. 19 No. 3, Springer, Apr. 1994, pp. 243-258.

- [48] A. Post and J. Hoenicke, Formalization and analysis of real-time requirements: A feasibility study at BOSCH, Proc. of VSTTE 2012, Springer, Jan. 2012, pp. 225-240.
- [49] A. Post, J. Hoenicke and A. Podelski, rt-Inconsistency: A New Property for Real-Time Requirements, Proc. of FASE 2011, LNCS, Vol. 6603, Springer, Mar. 2011, pp. 34-49.
- [50] A. Post, J. Hoenicke and A. Podelski, Vacuous real-time requirements, Proc. of RE 2011, IEEE Computer Society, Sep. 2011, pp. 153-162.
- [51] A. Post, I. Menzel, J. Hoenicke and et al., Automotive Behavioral Requirements Expressed in a Specification Pattern System: A Case Study at BOSCH, J. of Requirements Engineering, Vol. 17, No. 1, Mar. 2012, Springer, pp. 19-33.
- [52] A. Post, I. Menzel and A. Podelski, Applying Restricted English Grammar on Automotive Requirements, Proc. of REFSQ 2011, Springer, Mar. 2011, pp. 166-180.
- [53] A. Pretschner, et al., Software Engineering for Automotive Systems: A Roadmap, Proc. of FOSE 2007 (Future of Software Engineering), IEEE Computer Society, May 2007, pp. 55-71.
- [54] J. Pruitt and T. Adlin, The Persona Lifecycle: Keeping People in Mind throughout Product Design, Morgan Kaufmann, 2006 [秋本 芳伸 (訳), ペルソナ戦略—マーケティング, 製品開発, デザインを顧客志向にする, ダイヤモンド社, 2007].
- [55] REMsES (Requirements Engineering and Management of software-intensive Embedded Systems), <http://remses.org/> (accessed on Jan. 2018).
- [56] J. Robertson, and S. Robertson, Volere Requirements Specification Template Edition 17, Atlantic Systems Guild, 2014.
- [57] S. Robertson, and J. Robertson, Mastering the Requirements Process, 2<sup>nd</sup> ed., Addison-Wesley, 2006.
- [58] S. Saito, M. Takeuchi, M. Hiraoka, and et al., Requirements Clinic: Third Party Inspection Methodology and Practice for Improving the Quality of Software Requirements Specifications, Proc. of 21st Int'l Requirements Engineering Conference (RE 2013), IEEE Computer Society, Jul. 2013, Rio de Janeiro, Brazil, pp. 290-295.
- [59] S. Saito, M. Takeuchi, S. Yamada, and M. Aoyama, RISDM: A Requirements Inspection Systems Design Methodology, Proc. of 22nd Int'l Requirements Engineering Conference (RE 2014), IEEE Computer Society, Aug. 2014, pp. 223-232.
- [60] F. Salger, S. Sauer and G. Engels, An Integrated Quality Assurance Framework for Specifying Business Information Systems, Proc. of CAiSE Forum 2009, Jun. 2009, pp. 25-30.
- [61] C. Sauer, D. R. Jeffery, L. Land and et al., The Effectiveness of Software Development Technical Reviews: A Behaviorally Motivated Program of Research, IEEE Trans. on Software Engineering, Vol. 26, No. 1, Jan. 2000, 1-14.
- [62] J. Schäuffele and T. Zurawka, Automotive Software Engineering 2<sup>nd</sup> ed., SAE International, Sep. 2016.
- [63] M. Staron, Automotive Software Architectures: An Introduction, Springer, Jul. 2017.
- [64] A. Takoshima, and M. Aoyama, Assessing the Quality of Software Requirements Specifications for Automotive Software Systems, Proc. of APSEC 2015, IEEE CPS, Dec. 2015, pp. 393-400.
- [65] A. Takoshima, and M. Aoyama, A Two-Stage Inspection Method for Automotive Software Systems and Its Practical Applications, Proc. of RE 2016, IEEE Computer Society, Sep. 2016, pp. 313-322.
- [66] 蛸島 昭之, 青山 幹雄, 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, 情報処理学会論文誌, Vol. 58, No. 4, Apr. 2017, pp. 780-794.
- [67] S. Thiel, M. Babar, G. Botterweck, and L. O'Brien, Software Product Lines in Automotive Systems Engineering, SAE Int. J. Passeng. Cars - Electron. Electr. Syst. Vol. 1 No. 1, Apr. 2008, pp. 531-543.
- [68] VDA QMC Working Group 13/ Automotive SIG, Automotive SPICE Process Assessment Model V. 3.1, Nov. 2017, <http://www.automotivespice.com/> accessed on Jan. 2018.
- [69] H. Wang, et al., Envisioning a Requirements Specification Template for Medical Device Software, Proc. of PROFES 2014, LNCS Vol. 8892, Springer, Dec. 2014, pp. 209-223.
- [70] M. Weber and J. Weisbrod, Requirements Engineering in Automotive Development:

- Experiences and Challenges, IEEE Software, Vol. 20, No. 1, Jan./Feb. 2003, pp. 16-24.
- [71] K. Wiegers and J. Beatty, Software Requirements, 3rd ed., Microsoft Press, 2013.
- [72] W. Wilson, Writing Effective Natural Language Requirements Specifications, J. of Defense Software Engineering, Feb. 1999, pp. 16-19.
- [73] J. A. Zachman, A Framework for Information Systems Architecture, IBM Syst. J., Vol. 26, No. 3, 1987, pp. 276–292.
- [74] Y. Zhu, Software Reading Techniques, Apress, 2016.

## 研究業績

- [1] Akiyuki Takoshima, and Mikio Aoyama, Assessing the Quality of Software Requirements Specifications for Automotive Software Systems, Proc. of 22nd Asia-Pacific Software Engineering Conference (APSEC 2015), IEEE Conference Publishing Services, Dec. 2015, pp. 393-400 (査読有).
- [2] 蛸島 昭之, 青山 幹雄, 自動車ソフトウェア要求仕様書の二段階インスペクション方法の提案と適用評価, SES (ソフトエンジニアリングシンポジウム)2016 論文集, 情報処理学会, Sep. 2016, pp. 161-171 (査読有).
- [3] Akiyuki Takoshima, and Mikio Aoyama, A Two-Stage Inspection Method for Automotive Software Systems and Its Practical Applications, Proceedings of IEEE 24th International Requirements Engineering Conference (RE 2016), IEEE Computer Society, Sep. 2016, pp. 313-322 (査読有).
- [4] 蛸島 昭之, 青山 幹雄, 第三者インスペクションによる車載システム要求仕様書品質の見える化, 自動車技術会 2016 年 秋季大会学術講演会講演予稿集, No. 026, 自動車技術会, Oct. 2016, pp. 134-139 (査読有).
- [5] 蛸島 昭之, 青山 幹雄, 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, 情報処理学会論文誌, Vol. 58, No. 4, Apr. 2017, pp. 780-194 (査読有).
- [6] 蛸島 昭之, 青山 幹雄, ステークホルダ関心事に基づくソフトウェア要求仕様書構成モデルの設計方法, 第 198 回ソフトウェア工学研究会, No. 36, 情報処理学会, Mar. 2018, pp. 1-8 (査読なし).