

ソフトウェア品質技術が品質特性に与える効果の 見える化と活用の一考察

Evaluation of Impact of Software Quality Assurance Technique on Quality Characteristics

富士通株式会社 ネットワークソリューション事業本部
Network Solutions Business Unit, FUJITSU LIMITED

○小島 嘉津江 森田 純恵¹⁾ 廣瀬 竹男²⁾ 若本 雅晶³⁾ 菊池 慎司⁴⁾ 鷺崎 弘宜⁵⁾
○Katsue Kojima Sumie Moirta¹⁾ Takeo Hirose²⁾ Masaaki Wakamoto³⁾ Shinji Kikuchi⁴⁾
Hironori Washizaki⁵⁾

Abstract Since the quality of software can make a huge impact on our society, it should be evaluated from various viewpoints such as usability, market competitiveness and requirement satisfaction. Regarding the requirements expected to a project as “quality characteristics” of software, we proposed an approach for software quality assurance based on it. Our approach consists of the selection criteria for quality assurance technique and the usage of metrics to evaluate the software quality. We investigated the proposed method based on SQuBOK and SQuaRE standards. We also conducted an evaluation of our approach referring WSQB2017 benchmark conducted by Waseda University in RISE project. Based on the result, we discuss how we can make our approach adopted widely.

1. はじめに

Internet of Things (IoT) や人工知能 (AI) など、社会にイノベーションを起こす技術の急速な進展に伴い、これらを具現化するソフトウェアの重要性はますます高まっている。特にその品質が社会基盤やビジネスに大きな影響を与え、ソフトウェア品質の向上はこれからも重要な課題であり続けると言える。その為、ソフトウェアの品質は、ソフトウェア製品そのものに対する要求、利用時の視点での要求、そして、市場競争力強化への要求など、多面的に捉えることが求められるようになった。これらを実現するための標準的な品質特性のモデル、品質技術の知識体系、

富士通株式会社 ネットワークソリューション事業本部 フィールドエバリュエーション統括部
統括部長

Vice President, Field Evaluation Division, Network Solutions Business Unit, FUJITSU LIMITED
〒211-8588 神奈川県川崎市中原区上小田中 4-1-1 Tel: 044-874-2124

e-mail: kojima.katsue@jp.fujitsu.com

1-1, Kamiodanaka 4-chome, Nakahara-ku, Kawasaki, Kanagawa 211-8588, Japan

1) 株式会社富士通研究所 ソフトウェア研究所 主席研究員

Research Principal, Software Laboratory, FUJITSU LABORATORIES LTD.

2) 富士通株式会社 ネットワークプロダクト事業本部

Engineer, Network Products Business Unit, FUJITSU LIMITED

3) 株式会社富士通研究所 ソフトウェア研究所 専任研究員 2017/6/20 現在定年退職

Expert, Software Laboratory, FUJITSU LABORATORIES LTD.

4) 株式会社富士通研究所 ソフトウェア研究所 主任研究員

Research Manager, Software Laboratories, FUJITSU LABORATORIES Ltd

5) 早稲田大学 グローバルソフトウェアエンジニアリング研究所 所長、理工学術院 教授

Director, Global Software Engineering Laboratory, Waseda University

Professor, Faculty of Science and Engineering, Waseda University

【キーワード】 SQuaRE、ソフトウェア品質技術、ソフトウェア品質特性、SQuBOK ガイド、RISE、QFD

標準的な品質測定方法の各要素は揃ってきているが、それらの関係が網羅的に整理されておらず、活用するプロセスも未整備である。従って、本稿では、「多面化するソフトウェア品質要求をどのように実現できるか」を課題と捉え、ソフトウェアの品質確保を、国際標準という尺度で考え、「多面化するソフトウェア品質確保プロセス」として関係性モデル[品質ボックス](図4)を提案し、それを検証する。すなわち、ソフトウェア品質要求を品質特性として捉え、その品質特性を実現する品質技術を適用し、品質特性の達成度をメトリクスにより評価する。さらに、メトリクス測定値を日本の21のソフトウェアパッケージ/SaaS製品を測定評価してベンチマークとして公開された結果である「RISE ベンチマーク」^[1]に照らし合わせることで、評価対象製品の品質特性単位の相対的なポジションを明らかにし、その製品の品質レベルの妥当性検証を試みた。

本稿の構成は以下のとおりである。第2章で、ソフトウェアの品質技術を SQuBOK ガイド^[2]、ソフトウェアの品質特性を ISO/IEC 25000 シリーズに基づいて体系的に捉える。第3章は、品質技術と品質特性の関係性を発掘し、要求される品質特性を実現させる関係性モデルを提案する。第4章にてそのモデルの活用方法と「RISE ベンチマーク」より提案モデルの妥当性検証の結果を示し、「多面化するソフトウェア品質確保プロセス」(図12)の手順を示す。第5章で関連研究を述べ、最後に、第6章でまとめと今後の課題、さらなる検討の方向性を提案する。

2. ソフトウェア品質技術と品質特性

2.1 SQuBOK ガイドによる品質技術の体系化

品質特性を実現させるために、品質技術と品質特性の関係づけが必要になる。ソフトウェア品質技術は、様々な技術が提案・実用化されてきている。品質技術と品質特性のマッピングにあたっては、品質技術の網羅・体系化が必要なため、世界で初めてこれを実現した日本発の SQuBOK を参照することとした。SQuBOK は、図1に示すように、「ソフトウェア品質の基本概念」、「ソフトウェア品質マネジメント」、「ソフトウェア品質技術」の3つのカテゴリから成る。本稿で注目する「ソフトウェア品質技術」は、「工程に共通な品質技術」と「工程に個別な品質技術」、および、使用性やセキュリティなどの「専門的品質特性のソフトウェア品質技術」の3つの副カテゴリに分類される。さらに、15個の知識領域がこれら3つの副カテゴリに分類され、個々の知識領域は、複数の副知識領域に細分化される。具体的な品質技術はトピックスと称され、全部で142個あり、副知識領域ごとに整理されている。なお、紙面の都合上、図1では、副知識領域とトピックスは一例のみ記載した。

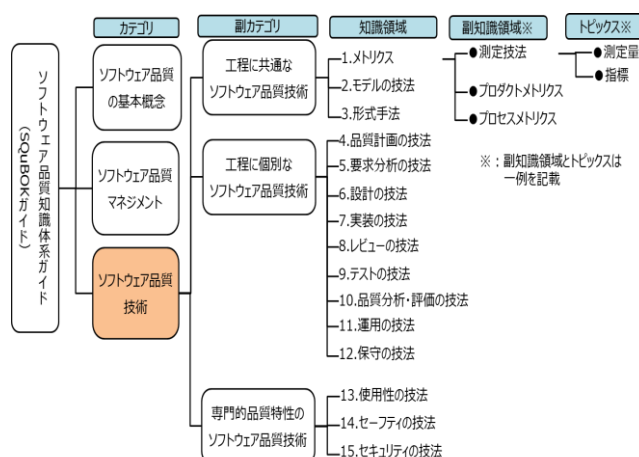


図1 SQuBOK ガイドの樹形図概観

2.2 ISO/IEC 25000 シリーズ (SQuaRE) における品質モデル

ソフトウェア品質に関する国際規格である ISO/IEC 25000 シリーズでは、ソフトウェア品質モデルおよび品質特性を規定している。品質モデルは、以下の3つで構成されている。

- 利用時の品質 (ISO/IEC 25010) ^[3]
- システム/ソフトウェア製品品質 (ISO/IEC 25010) ^[3]
- データ品質 (ISO/IEC 25012) ^[4]

利用時の品質モデルとシステム/ソフトウェア製品品質モデルをそれぞれ図2と図3に示す。両者とも、品質特性とこれを細分化した品質副特性から構成される。表1はデータ品質モデル特性を示しており、各特性を固有の視点とシステム依存の視点に分類していることが特徴である。

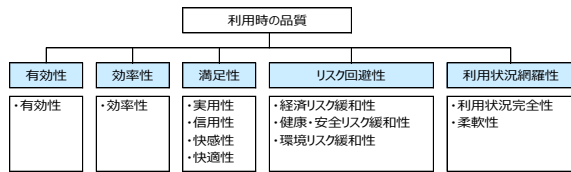


図2 利用時の品質モデル

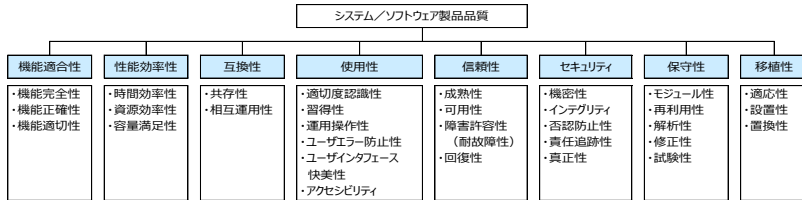


図3 製品品質モデル

表1 データ品質モデル特性

特性	データ品質	
	固有	システム依存
正確性 (Accuracy)	○	
完全性 (Completeness)	○	
一貫性 (Consistency)	○	
信ぴょう(憑)性 (Credibility)	○	
最新性 (Currentness)	○	
アクセシビリティ (Accessibility)	○	○
標準適合性 (Compliance)	○	○
機密性 (Confidentiality)	○	○
効率性 (Efficiency)	○	○
精度 (Precision)	○	○
追跡可能性 (Traceability)	○	○
理解性 (Understandability)	○	○
可用性 (Availability)		○
移植性 (Portability)		○
回復性 (Recoverability)		○

3. 品質技術と品質特性、RISE メトリクスの関係性の発掘「品質ボックス」

ソフトウェア品質要求を実現させるために、横軸に品質特性(SQuaRE) (XX 性/YY 性等)、縦軸に品質技術(SQuBOK) (aaa/bbb 等)で構成したマトリクスに対し、以下のマイニング手順を施し、RISE メトリクス^[1]とも関連づけ、関係性モデル[品質ボックス]として活用することを提案する。

Mining1 (M1) : 品質特性の実現に寄与する品質技術に対し、マークする^[5]。

Mining2 (M2) : 品質保証の経験・実績を基に、関係性を深掘し、マークを追加する。尚、品質作り込み技術をターゲットに実施。

Mining3 (M3) : 縦軸の品質特性ごとに、RISE のメトリクスを割り当てる。

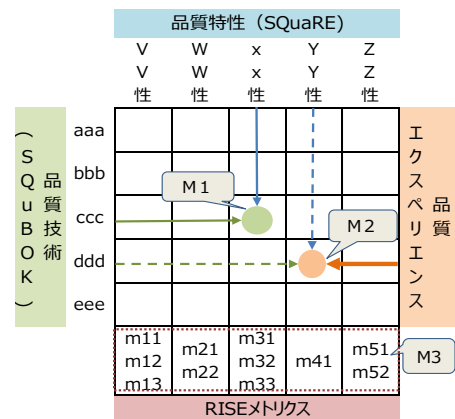


図4 関係性モデル「品質ボックス」

関係性モデル[品質ボックス]のイメージを図4に示し、上記のM1~M3を、それぞれ3.1~3.3項で詳説する。尚、本章で作成する関係性モデル「品質ボックス」を以下のURLに公開する。

http://www.washi.cs.waseda.ac.jp/wp-content/uploads/2017/07/SQIP_Quality_Box.pdf

3.1 SQuBOK ガイドの記述に基づく品質技術と品質特性のマッピング【M1】

品質技術(SQuBOK)と品質特性(SQuaRE)の相互マッピングは、それらの関係性を網羅的に見える化したものが『ソフトウェア品質技術が品質特性に与える効果の見える化』^[5]として示されている。文献[5]においては、品質技術と品質特性のマトリクスにつき、品質技術(トピックス)142個を縦軸に、品質副特性57個を横軸に配置し、筆者の1名が「SQuBOKに品質特性との関係が明記されていること」を基準にマーク(①印)し、他の筆者のレビューによる追加・修正した。図5は、このマトリクスにおいて、縦軸を知識領域(15個)に、横軸を品質特性(28個)に集約して、知識領域と品質特性のマッピングを示したものである。①印は、該当知識領域がカバーしている品質特性を示す。なお、データ品質特性については、「完全性」、「機密性」と「正確性他」で分類した。これは、SQuBOKでは、データ品質の技術について記載が完全性と機密性に関連するものみのためである。本稿においては、これを1つ目の手法(①)とする。

SQuBOKガイド第2版をベースに作成			ソフトウェア品質特性																		
カテゴリ	副カテゴリ	知識領域	利用時の品質				システム/ソフトウェア製品品質				データ品質										
			有効性	効率性	満足性	リスク回避性	利用状況網羅性	機能適合性	性能効率性	互換性	使用性	信頼性	セキュリティ	保守性	移植性	完全性	機密性	正確性他			
3・ソフトウェア品質技術	一般的なソフトウェア品質技術	1	メトリクス	①	①	①	①	①	①	①	①	①	①	①	①	①	①	①	①	①	
		2	モデル化の技法						①	①					①	①					
		3	形式手法						①						①						
	工程に個別なソフトウェア品質技術	4	品質計画の技法	①	①	①	①	①	①	①	①	①	①	①	①	①	①	①	①	①	①
		5	要求分析の技法	①	①	①	①	①	①	①	①	①	①	①	①	①	①	①	①	①	①
		6	設計の技法						①	①	①	①	①	①	①	①	①	①	①	①	①
		7	実装の技法						①	①				①	①	①	①	①	①	①	①
		8	レビューの技法				①	①	①	①			①	①	①	①	①	①	①	①	①
		9	テストの技法					①	①	①			①	①	①			①	①		
		10	品質分析・評価の技法						①	①			①	①			①	①			
		11	運用の技法							①			①	①	①	①	①	①	①	①	①
		12	保守の技法								①	①				①					
	専門的なソフトウェア品質技術	13	使用性の技法						①					①							
		14	セーフティの技法				①	①				①	①	①							
		15	セキュリティの技法						①			①	①	①							

図5 知識領域と品質特性のマッピング

3.2 品質技術と品質特性の更なる関係性の発掘【M2】

品質の作り込みフェーズに対応する「モデル化の技法」、「形式手法」、「要求分析の技法」、「設計の技法」、「実装の技法」の5つの知識領域にターゲットを絞り、ここに分類される品質技術（全29個）と品質特性との関係をより詳細に調査・分析した。具体的には、3.1節に加え、下記の3手法(②③④)を用い、産学ならびに文献[2]編纂の観点から複数名でレビューした。

- ② 現場で品質保証にたずさわった経験をもとに、品質技術ごとに調査・分析を行い、経験から得られた知見に基づいて関係性を抽出
- ③ 文献[2]を品質技術ごとにさらに詳細に調査し、関係性に直接言及していなくとも、前後の文脈などから関係性が推測（演繹的推論など）できるものを抽出
- ④ 顧客要件を実現するという観点から「機能適合性」と品質技術との関係性を再度確認

上記で述べた手法で新たな関係性を発掘した箇所、手法の番号(②、③、④)を記載して作成したマトリクスの一部抜粋を図6に示す。マトリクスには現れていないが、手法②と③はオーバーラップする箇所が多かった。この場合、手法②

SQuBOKガイド第2版をベースに作成			ソフトウェア品質特性																	
知識領域	副知識領域	トピックス	利用時の品質				システム/ソフトウェア製品品質				データ品質									
			有効性	効率性	満足性	リスク回避性	利用状況網羅性	機能適合性	性能効率性	互換性	使用性	信頼性	セキュリティ	保守性	移植性					
2	モデル化の技法	1	UML							①	①	②		①		①	①			
		2	分散系のモデル化技法							①	①	②		①		①	①			
		3	構造化チャート(PAD)							①	①	②		①		①	①			
3	形式手法	1	連続系のモデル化技法						④	③	②		③		③	③				
		2	ドメイン特化言語						④	③	②		③		③	③				
		3	形式仕様記述の技法						①				①							
5	要求分析の技法	1	形式検証の技法				①	①	①			①			①					
		2	形式検証の技法				①	①	①			①			①					
		1	要求抽出			②	②	②	②	①	④									
		2	要求開発 (Openthology)			①	①	①	①	①	④									
	3	要求分析の技法	1	機能要求分析			②			①						①				
			2	非機能要求分析			②	②	②			①	②	①	①	①	①	①	②	
			3	品質機能展開			②	②				①	①	①	①	①	①	①	①	
			4	要求可変性分析							④							①	②	
4	要求の妥当性確認と評価	1	ソフトウェア要求仕様			②		②	②	①	②	②	②	②	②	②	②	②		
		2	USDM (要求仕様記述法)			②		②	②	①	①	②	②	②	②	②	②	②		
		3	要求の妥当性確認と評価			②		②	②	②	④	②	②	②	②	②	②	②		
		4	要求の妥当性確認と評価			②		②	②	②	④	②	②	②	②	②	②	②		

図6 発掘された新たな関係性（一部抜粋）

をマトリクス上に記載し、③は②の根拠を補強する位置づけとして、マトリクスには記載していない。このように、関係性を判断する根拠の強さは、手法②、③、④の順になると考える。

手法②および③により関係性があると判断した箇所の根拠の一例を以下に示す。(図6の枠)

手法②の根拠例：『UML』は、コンポーネント単位の仕様を明確に表現でき、モデルレベルで、コンポーネントの再利用性や他のコンポーネントとの『互換性』がある。

手法③の根拠例：『分散系のモデル化技法』に、「シミュレーション解析により信頼性や効率性

を確認できる」とあり、『連続系のモデル化技法』でもシミュレーション解析を行うことから、信頼性と効率性（性能効率性）に寄与すると考えられる。

品質向上に最も関わる「品質の作り込み」フェーズを対象としたマッピングの充実により、その関係を見出す手法の違い（①、②、③、④）も考慮した品質技術の選択が可能となり、ソフトウェア開発エンジニアのためのガイドとしてもより実用性の高いものとなったと考えられる。たとえば、プロジェクトの特性を考慮して重要視する品質特性を選び、それを実現するために必要な品質技術を、今回得たマトリクスから優先度を考慮して抽出できる。また、①印が少ない品質特性に対しては、②/③/④印により、見落としがちであった品質技術も選択の対象となり、品質向上に寄与できる。

3.3 品質特性達成度の測定方法【M3】

品質技術の適用により、期待した品質特性を得られたか否かを判断できることが、品質向上のために求められ、品質特性達成度の評価指標の定義と定量化が期待される。

「異なる品質間の関係を総合的に実証した世界初のベンチマーク（WSQB2017）」（RISE ベンチマーク）にて、SQuaRE シリーズ中の ISO/IEC 25022, 25023 に規定された測定を GQM 法の適用により実効性のある形で具体化したマトリクスを定義している。該文献では、利用時の品質として 17 個、システム/ソフトウェア製品品質として 66 個の合計 83 個のマトリクスを品質副特性ごとに定義している。ここでは、紙面の都合上、品質特性ごとに集約し、代表的なマトリクス（最大 4 つ）を、図 7 に示す。これらのマトリクスを対象製品の品質特性達成度の判断に利用するとともに、RISE ベンチマーク結果と照合することにより、強化すべき品質特性の抽出が可能になる。

ソフトウェア品質特性												
利用時の品質					システム/ソフトウェア製品品質							
有効性	効率性	満足性	リスク回避性	利用状況網羅性	機能適合性	性能効率性	互換性	使用性	信頼性	セキュリティ	保守性	移植性
タスク完了率	タスクにかかった時間の平均	製品に対する満足度	経済的損失を感じる頻度	主要な目的以外の製品利用の有無	要求実装率	時間効率性の試験有無	他製品を共存させて試験する意図の有無	機能の動画画説明対応率	不具合除去率（単体試験）	データのアクセス権限管理対応率	クラスの結合度	複数環境の試験有無
タスク当たりエラー数	タスク中の総アクションの無駄でないアクションの率	NetPromoterScore	健康や人命への影響を感じる頻度	非主要目的での製品利用時タスク達成度合い	深刻不具合除去率	応答時間平均	取り扱うファイル形式のうち、インポート/エクスポート両対応の比率	機能の説明記載率カタログ	不具合除去率（結合試験）	データの暗号化対応率	関数のサイクロマティック複雑度	インストールの試験有無
エラーが発生したタスクの率		機能に対する満足度	環境への影響を感じる頻度		システム試験数目標達成率	応答時間実測対目標		機能の説明記載率マニュアル	不具合除去率（システム試験）	データの破損防止策対応率	クラスの凝集性の欠如	インストール時間平均
エラーを起こした被験者の率		信用度合い			ユーザの意図に即す度合い	ターンアラウンドタイム平均		機能のUndo対応率	MTBF目標達成率	ネットワーク経路のデジタル署名対応率	コーディング規約違反（測定保留中）	インストール提供形態対応率

図 7 品質特性ごとのマトリクス（一部抜粋）

4. 「品質ボックス」を用いたソフトウェア品質確保プロセスの提案と実証

本章では、まず、図 4 で示した関係性モデル[品質ボックス]（品質技術と品質特性のマトリクス、および品質特性マトリクス）と「RISE ベンチマーク」を用いたソフトウェア品質確保プロセスを提案する。そして、あるソフトウェア製品（RISE 調査対象）開発に、[品質ボックス]を適用し、本プロセスの妥当性につき、RISE ベンチマーク報告を用いて検証した。

4.1 「品質ボックス」と「RISE ベンチマーク」を用いた品質確保プロセスの提案

「品質ボックス」と「RISE ベンチマーク」の活用手順を「多面化するソフトウェア品質確保プロセス」として図 8 に示す。

まず、手順 a でプロジェクトに求められる要件（品質要求）を、品質特性で表現する。たとえば、長期間にわたり利用されるシステムであれば、システム更改を考慮し、互換性や移植性を選択する。手順 b では、手順 a で抽出した品質特性の中から、注力するものを決定する。ここでは、QCD のトレードオフなどを考慮し、また、RISE メトリクスやベンチマークを活用して、戦略的の方針を決定する。手順 c で、該当品質特性を実現する品質技術を吟味・決定する。手順 d で品質

特性の達成度合いをRISEメトリクスにより測定・評価する。手順eでは、RISEベンチマークによるポジショニングを評価し、強み・弱みを判別し、必要に応じて、品質技術やメトリクス目標値にフィードバックし、ソフトウェアの品質はもとより、プロセスの成長・醸成を可能にする。

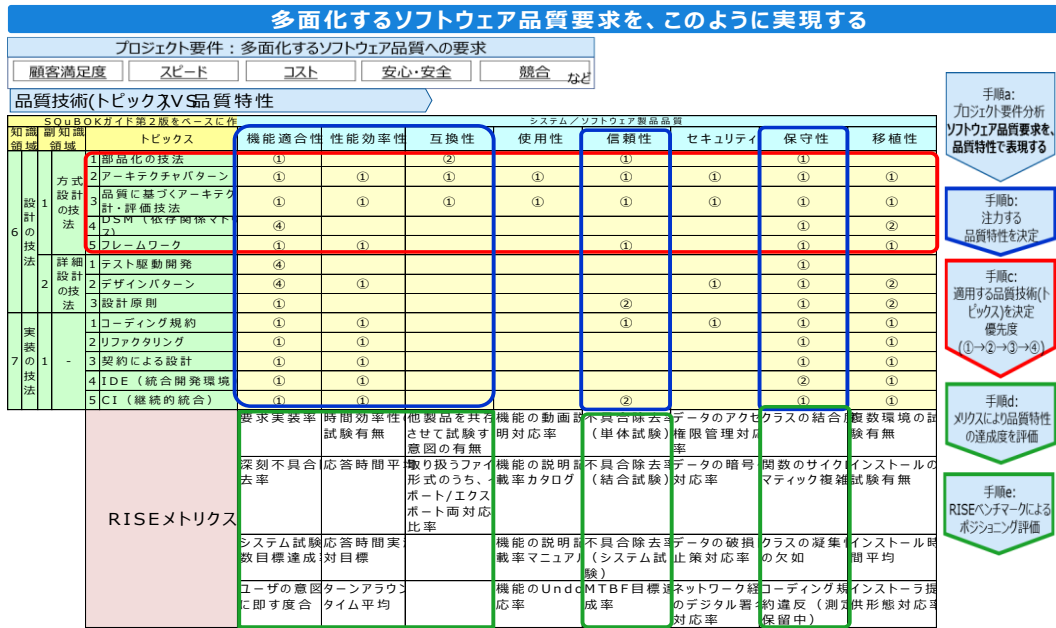


図 8 多面化するソフトウェア品質確保プロセス

4.2 「品質ボックス」の適用

(1) プロジェクト品質要件から品質特性を決定

検証対象のソフトウェア製品に求められる要件を満たすために考慮すべき品質特性を、表2のように決定した。データ処理能力を強く要求されているため、製品品質の性能効率性を最優先とし、使用性、信頼性も重視した。また、利用時品質として、有効性や効率性を重視した。一方、市場優位性を勝ち取ることがビジネス上の戦略でもあり、早期デリバリーを強く要求されたため、システム動作条件(OSなど)を固定化することとし、互換性・移植性は低優先とした。

表2 要件を満たすための品質特性 (最優先：◎→○→△→×：低優先)

プロジェクト要件	利用時の品質					システム/ソフトウェア製品品質							
	有効性	効率性	満足性	リスク回避性	利用状況網羅性	機能適合性	性能効率性	互換性	使用性	信頼性	セキュリティ	保守性	移植性
データ処理能力重視	○	○				○	◎		○	○	△	○	
早期デリバリー			△	△	△			×					×

(2) 注力する品質特性から適用する品質技術を決定【M1、M2】

「品質ボックス」の品質特性と品質技術のマトリクス(M1, M2)の抜粋を図9に示す。これを用いて、性能効率性を高めるために適用する品質技術(トピックス)を選択した。

知識領域	副知識領域	トピックス	機能適合性	性能効率性	互換性	使用性	信頼性	セキュリティ	保守性	移植性
テストの技法	5 利用に基づいた技法	1 運用プロファイルによるテスト	①					①		
		2 ローカライゼーションテスト	①							
		3 ユーザー環境シミュレーションテスト	①	①		①				
		4 整合性確認テスト	①	①			①			
	6 ソフトウェアの形態に基づいた技法	1 オブジェクト指向テスト	①							
		2 Webシステムのテスト	①	①		①		①		
		3 GUIテスト	①							
		4 サーバーサイドのテスト	①	①			①			
		5 データベーステスト	①							
		6 並行プログラムのテスト	①				①			
		7 プロトコル適合性テスト	①							
		8 実時間のテスト	①				①			
		9 モバイルアプリケーションのテスト	①	①				①		

図 9 品質特性と品質技術のマトリクス(M1, M2) (抜粋)

「品質ボックス」で定義する性能効率性に関与する知識領域ごとの品質技術(トピックス)数と、その内、本製品開発に適用したトピック数を図10に示す。性能効率性という特徴から、実動作での確認を重視し、テストの技法を多く取り入れた。また、昨今、強く求められる安心・安全確保を考慮し、該当する品質技術も採用した。

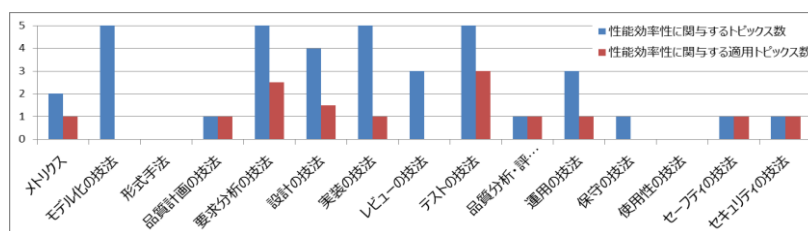


図10 適用した品質技術数(性能効率性)

(3) 品質特性に対応するメトリクスの決定と測定結果【M3】

「品質ボックス」のRISEメトリクスの内、性能効率性に関する測定値を図11に示す。測定条件ごとに目標値と実績値を測定し、RISEメトリクスとして各種平均値や実績対目標値を導き出した。RISEメトリクスで推奨されている大小比較(対1.00)も参考にすると全般的に適切なメトリクス測定値を得られたことがわかる。RISEベンチマークの詳細は、文献[1]を参照のこと。

		RISEメトリクス(性能効率性)										
		時間効率性の試験有無	応答時間平均	応答時間実測対目標	ターンアラウンドタイム平均	ターンアラウンドタイム実測対目標	スループット目標達成率	資源効率性の試験有無	CPU使用率最大値	メモリ使用率最大値	容量満足性の試験有無	ユーザ同時アクセス可能数目標達成率
大or小が望ましい		大	小	小	小	小	大	大	小	小	大	大
メトリクス測定値		1	-	-	-	-	2.08	1	0.03	NA	1	NA
測定条件(サンプル)		-	ログイン画面表示	-	プロセス起動	-	データ蓄積	-	データ保存	データ保存	-	1-ダ数
		-	-	-	データ検索(500件)	-	-	-	データ検索	データ検索	-	-

図11 メトリクス測定結果(性能効率性)

4.3 「RISEベンチマーク」によるポジショニング評価

RISEベンチマーク結果の一部を図12に示す。RISEベンチマークでは、メトリクス測定値をスコア化し21製品内の当該製品のポジショニングが可能であり、以下個々に分析する。

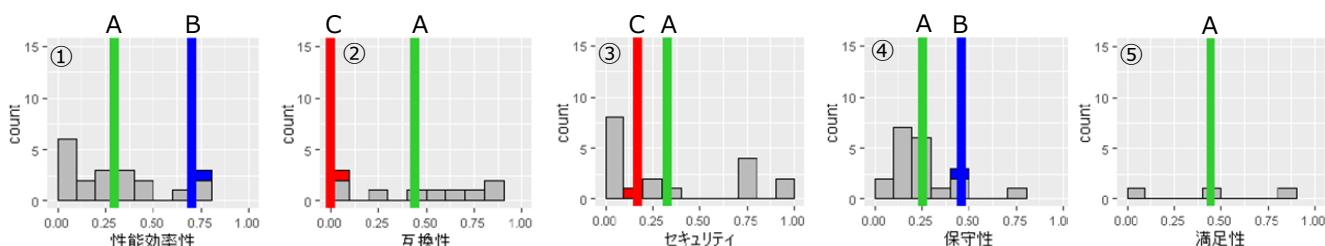


図12 品質特性のベンチマーク結果(一部抜粋)

(Aは21製品の平均値、B・Cは該当製品の位置(B:平均以上、C:平均以下)、棒グラフは21製品のスコア分布)

- ① 性能効率性：平均を上回り、他製品より優れていることがわかる。本開発で高優先としたメトリクスであり、製品としての強みを実現できたことが確認できた。
- ② 互換性：優先度を下げたため、平均を下回り、優位性が低い結果となった。また、本メトリクスのスコアは、広い範囲でなだらかに分布している特徴がみられる。今後のエンハンスなどを考えると優位性決定時の意識改革が必要である。
- ③ セキュリティ：今回の開発で優先度を高くしておらず、妥当な結果であるとする。また、本メトリクスのスコア分布には幅広く二極化している特徴がみられる。セキュリティは、今後の社会で重要な品質特性であることからこうした意識改革が求められる。
- ④ 保守性：今回の開発で優先度をやや高めに設定したものであり、想定通りの結果が得られたと考える。本メトリクスは、スコア分布が低い方に集中しており、本製品のポジショニングは優位性を強調できるものである。継続して品質向上に取り組むことを考える。

- ⑤ 満足性：メトリクス測定が不足しベンチマークできなかったが、他製品のスコア分布からサンプル数が少ないことが明白である。利用者視点の品質を向上させサービス品質を早期に高めることにより、新たな製品の強みを出していくことが可能になると考える。

5. 関連研究

ソフトウェア品質を客観的に評価するための尺度についての研究は、国際規格（ISO/IEC 25000 シリーズ）、これら個々の品質技術と各品質特性の関係性を示したものがあり、各品質特性の向上に寄与する技術群をいくらか整理した成果は過去にも存在し、CMU/SEI のアーキテクチャ手法群においては、独自の品質モデルに基づいて、種々の技術（Tactics）を品質特性ごとに整理している^[6]。さらに、品質要求からのマトリクス作成という観点では、QFD（Quality Function Deployment：品質機能展開）^[7]が若干関係する。しかし、これらは「網羅的ではない」「最新のものではない・改訂されていない」「技術の詳細説明がしばしばない」「個々に作成する必要がある」といった問題があった。そのため、品質要求に基づく効率的な品質技術の選択の難しさ、品質技術の研究開発が必要な方向・領域の不透明さなどの課題が残っている。

6. まとめと今後の課題

本稿では、図4の関係性モデル[品質ボックス]（品質技術と品質特性のマトリクス、および品質特性メトリクス）と「RISE ベンチマーク」を用いた「多面化するソフトウェア品質確保プロセス」を提案・実証した。ソフトウェア品質技術と品質特性のマトリクスの活用（3.1 節、3.2 節）は、品質確保のための十分条件であるため、その必要十分条件となるよう RISE 研究結果を加えて（3.3 節）、4 章にて実製品における関係性モデル[品質ボックス]（図4）の妥当性検証を行ったものである。今後は、さらに今回の関係性の深堀りや実証を通して、SQuBOK や RISE メトリクスに対し、以下を考慮すると更なる品質向上の実現ができると考える。

SQuBOK では、製品品質の一部の品質特性で、寄与する品質技術の数に偏りが見られる。今回、一部機能に絞って深掘りしたが、SQuaRE で強化された互換性や、利用時の品質、さらに IoT や AI のデータ活用型のシステム/ソフトウェア開発に今後より重要となっていくデータ品質に関する技術の拡充を図ることが必要である。

今回提案した品質確保プロセスを開発現場で実践を進め、SQuBOK V3 研究チームや RISE 研究機関との議論を重ね、関係性モデル[品質ボックス]の精度と利用しやすさの向上を進めていく。そして、関係性モデル[品質ボックス]は、2020 年発行予定の SQuBOK V3 の掲載をここに提案する。

謝辞 本稿の執筆にあたり、東洋大学・野中誠教授、日本電気(株)・誉田直美主席品質保証主管、(株)日立ハイテクノロジーズ・飯泉紀子専門部長、公立はこだて未来大学・大場みち子教授、及び、SQuBOK V3 研究チームの方々に多くのご助言を頂いた。ここに謝意を表する。

参考文献

- [1] 早稲田大学グローバルソフトウェアエンジニアリング研究所（研究責任者鷺崎弘宜）、「異なる品質間を総合的に実証した世界初のベンチマーク（WSQB2017）」、2015年度ソフトウェア工学分野の先導的研究支援事業（RISE），2017
- [2] SQuBOK 策定部会編，ソフトウェア品質知識体系ガイド第2版，オーム社，2014.
- [3] ISO/IEC 25010:2011 Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models (JIS X 25010:2013 システム及びソフトウェア製品の品質要求及び評価 (SQuaRE) システム及びソフトウェア品質モデル)
- [4] ISO/IEC 25012:2008 Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Data quality model (JIS X 25012:2013 ソフトウェア製品の品質要求及び評価 (SQuaRE) - データ品質モデル) .
- [5] 小島 嘉津江，森田 純恵，若本 雅晶，宗像 一樹，鷺崎 弘宜，ソフトウェア品質技術が品質特性に与える効果の見える化，情報処理学会第195回ソフトウェア工学研究発表会，2017
- [6] Len Bass, Paul Clements, Rick Kazman, 前田卓雄・佐々木明博・加藤滋郎・新田修一・吉野圭一（訳），実践ソフトウェアアーキテクチャ，日刊工業新聞社，2005.
- [7] 山田 洋二，マツダ技法 No. 33 (2016) 品質機能展開を活用した技術開発プロセス，2016