

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その検証

小島 嘉津江^{※1}森田 純恵^{※2}廣瀬 竹男^{※1}若本 雅晶^{※3}菊池 慎司^{※3}椋 晃歓^{※4}鷲崎 弘宜^{※5}

ソフトウェア品質に対する要求が多面化する一方で、要求される品質特性を実現するための方法は整備されていない。この問題を解決するために、本論文ではソフトウェアに要求される品質特性とそれを実現するための品質技術の関係を明らかにする「品質ボックス」モデルを提案する。提案モデルを複数のソフトウェア製品に適用し、その妥当性についての評価結果を示す。

Evaluation of Impact of Software Quality Assurance Technique on Quality Characteristics

Katsue Kojima^{※1}, Sumie Morita^{※2}, Takeo Hirose^{※1}, Masaaki Wakamoto^{※3}, Shinji Kikuchi^{※3}, Akiyoshi Hando^{※4}, Hironori Washizaki^{※5}

Since software qualities are required to be evaluated from various aspects, it is difficult to determine how to satisfy the required quality in systematic way. To solve the problem, we defined a “quality box” model representing the relation between required qualities and techniques to achieve them. Through the application of the model to some actual software products, we conducted the evaluation of the validity of the model.

※1 富士通株式会社 ※2 株式会社富士通ゼネラル(2017年末まで株式会社富士通研究所在籍)
 ※3 株式会社富士通研究所 ※4 株式会社富士通ビー・エス・シー ※5 早稲田大学

3.2 SQaRE 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 (SQaRE) では、ソフトウェア品質モデル及び品質特性を規定している。品質モデルは、以下の4つで構成されている。

- ・ 利用時の品質 (ISO/IEC 25010) [ISO25010]
- ・ システム/ソフトウェア製品品質 (ISO/IEC 25010) [ISO25010]
- ・ データ品質 (ISO/IEC 25012) [ISO25012]
- ・ サービス品質 (ISO/IEC TS 25011) [ISO25011]

本稿で対象とした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。データ品質モデル特性については、各特性を固有の視点(正確性, 完全性, 一貫性, 信憑性, 最新性, アクセシビリティ, 標準適合性, 機密性, 効率性, 精度, 追跡可能性, 理解性)とシステム依存の視点(アクセシビリティ, 標準適合性, 機密性, 効率性, 精度, 追跡可能性, 理解性, 可用性, 移植性, 回復性)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 システム/ソフトウェア製品品質モデル

3.3 メトリクス測定値による評価

品質技術と品質特性の関係の体系化及び、実製品への適用評価を実現するために、我々は、その多様性に対応できる点に着眼し、「異なる品質間関係を総合的に実証した世界初のベンチマーク (WSQB2017)」[早稲田大学 2017]を活用することとした。上記ベンチマーク中で提案されている RISE メトリクスは、SQaRE シリーズ中の ISO/IEC 25022, 25023 に規定されたメトリクスの中から、GQM 法の適用により実効性のある項目を抽出したものである [WSQB]。当該文献では、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 個のメトリクスを品質副特性ごとに定義している。RISE ベンチマークは、複数製品の RISE メトリクス値の分布を評価することで、各製品のポジショニングを可視化できる。本論文では、この RISE メトリクス及びベンチマークの枠組みを活用して、評価を行うこととした。

4 関係性モデル:品質ボックス

本節では、前節で示したソフトウェア品質技術と品質モデルを用いて、それらの関係を示すモデル:品質ボックスを提案する [品質ボックス 2017]。その品質ボックスの概念図を図4に示す。このモデルでは、横軸に SQaRE で示されるソフトウェアの品質特性(正確性, 可用性など)、縦軸に SQuBOK の品質技術(形式手法, テストの技法など)でマトリクスを構成し、それらの関係を明らかにしている。また、各品質特性に対応する RISE メトリクスも明示することで、各品質特性における重要メトリクスを示す。我々は、以下の手順で品質ボックスにマーク付けし、品質技術と品質特性の関係を明らかにした。

- (1) Step1: SQuBOK に基づき、汎用的かつ客観的な観点から、品質特性の実現に寄与する品質技術を特定する [小島 a2017]。
- (2) Step2: 各開発現場の品質保証の経験・実績 ((品質エクスペリエンス)) を基に、個別かつ主観的な観点から、品質特性の実現に寄与する品質技術を特定する [小島 b2017]。

上記の2ステップについて、4.1 ~ 4.2 節でそれぞれ詳説する。

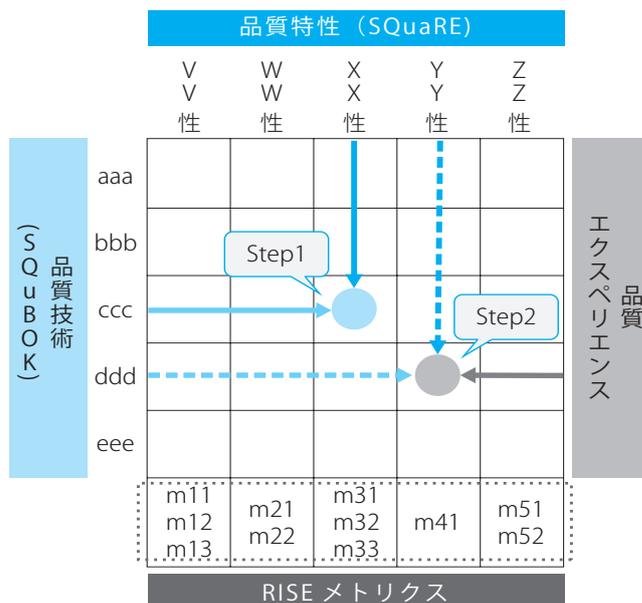


図4 関係性モデル:品質ボックス

4.1 汎用的かつ客観的な観点からの品質技術と品質特性の対応付け (Step1)

品質技術 (SQuBOK) と品質特性 (SQuaRE) の対応付けについては、『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』[小島 a 2017] に網羅的に示している。品質技術と品質特性のマトリクスは、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、筆者らが汎用的かつ客観的に「SQuBOK に品質特性との関係が明記されていること」を基準にマークした。図 5 は、縦軸を品質技術の知識領域 (15 個) に、横軸を利用時の品質とシステム/ソフトウェア製品品質の品質特性 (13 個) に集約して、知識領域と品質特性の関係を示したものである。①印が前述のマークした品質特性である。

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	副カテゴリー	知識領域	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	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
3. ソフトウェア品質技術	トピック領域	1. メトリクス	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	①
		3. 形式手法						①				①			
	工程に個別なソフトウェア品質	4. 品質計画の技法	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
		6. 設計の技法						①	①	①	①	①	①	①	①
		7. 実装の技法						①	①			①	①	①	①
		8. レビューの技法				①	①	①	①	①	①	①	①	①	①
		9. テストの技法					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	
		12. 保守の技法							①	①				①	
	トピック領域	13. 使用性の技法									①				
		14. セーフティの技法				①	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

4.2 個別かつ主観的な観点からの品質技術と品質特性の対応付け (Step2)

ここでは、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 種類) 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、下記の 3 手法を用い、文献 [SQuBOK2014] 編さんの観点に基づき実施した。

- (1) 現場で品質保証に携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した。例えば UML (品質技術) 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ほかのコンポーネントとの互換性 (品質特性) があると考えられる。
- (2) 文献 [SQuBOK2014] を品質技術ごとに更に詳細に

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できるものを抽出した。例えば、離散系のモデル化技法 (品質技術) には、「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」という記載があり、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、これらは信頼性と効率性 (品質特性) に寄与すると考えられる。

- (3) 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確認した。

図 6 は、図 5 の①印に加え、上記で述べた 3 つの手法による新たな関係性を発掘した個所に②印, ③印, ④印をマークして作成したマトリクスの一部抜粋である。

品質向上に最もかかわる「品質の作り込み」フェーズを対象としたマッピング (Step2) の充実により、その関係を見出す手法の違い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。例えば、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することが可能になる。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。

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							
			有効性	効率性	満足性	リスク回避性	利用状況	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	離散系のモデル化技法	1. UML						①	①	②		①		①	①
		2. SysML						①	①	②		①		①	①
		3. 構造化チャート (P4D)						①	①	②		①		①	①
3. 形式手法	連続系のモデル化技法	1. ドメイン特化言語						④	③	②		③		③	③
		2. ドメイン特化言語						④	③	②		③		③	③
		3. ドメイン特化言語						④	③	②		③		③	③
3. 形式手法	形式仕様記述の技法	1. 形式仕様記述の技法						①				①			
		2. 形式検証の技法						①	①	①		①			

図 6 発掘された新たな関係性 (一部抜粋)

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る。そこでは、品質特性達成度の評価指標の定義と定量化が必要である。

また、RISE メ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共に、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。例として、品質特性ごとに整理した RISE メトリクスの代表 (最大 4 つ) を、図 7 に示す (紙面の都合により一部抜粋版)。

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	機能の動画面説明対応率	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	Net Promoter Score	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)	インストーラ提供形態対応率

図7 品質特性ごとのRISEマトリクス(一部抜粋)



図8 多面化するソフトウェア品質確保プロセス

5 品質確保プロセスの提案

本節では、図4で示した関係性モデル:品質ボックスとRISEベンチマークを用いて、多面化するソフトウェアの品質確保をするプロセスを提案する。図8にそのプロセスを示す。本プロセスにおいては、図の右側に書かれている以下の手順a~eにより、ソフトウェアの品質特性の評価を行う。

- (1) 手順a: まず、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。例えば、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。
- (2) 手順b: 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のトレードオフの考慮や、RISEマトリクスやベンチマークの活用などにより、戦略的に方針を決定する。

- (3) 手順c: 該当品質特性を実現可能な品質技術を抽出する。
- (4) 手順d: 品質特性の達成度合いをRISEマトリクスにより測定・評価する。
- (5) 手順e: RISEベンチマークによるポジショニングを評価し、製品の強み・弱みを判別する。必要に応じて、品質技術やマ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及びプロセスの改善につなげる。

6 品質確保プロセスの検証

本節では、前節で提案した品質確保プロセスの妥当性評価を行う。具体的には、RISE調査対象でもある製品A(パッケージ製品)、製品B(クラウドサービス)の開発において、図8の手順に基づき品質ボックスを適用し、検証を行った。

表1 要件を満たすための品質特性 (製品A)
(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			△	△	△			×					×

表2 要件を満たすための品質特性 (製品B)
(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
クラウドサービスとしての運用強化							○			◎	○	○	
W向け制御機能での競争力強化	◎	△	○	△	△	○		×	△				×

6.1 品質ボックスの適用

(1) 手順 a, b: 品質要件から品質特性を決定

製品A, 製品Bのそれぞれに求められる要件(品質要求)を品質特性で表現し, その注力する品質特性を決定したものを表1, 表2に示す。

製品Aは, 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, 信頼性を重視した。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ているため, 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。

製品Bは, クラウドサービスとして, 既存利用中の全ユーザに対し自動的に適用される追加機能であるため, 信頼性を優先事項とした。また, 対象開発での優先機能としてWindows向けの制御機能にて競合他社との差異化要素を追加し, 市場競争力の強化を行うことが求められたことから, 有効性を高優先とした。一方, いち早く差別化機能を市場投入するため, 互換性や移植性は優先度を下げることとした。

(2) 手順 c: 注力する品質特性から品質技術を決定

品質ボックスの品質特性と品質技術のマトリクスの抜粋を図9に示す。これを用いて, 製品Aについて最優先とする性能効率性を高めるために適用する品質技術を決定した。

品質ボックスで定義する性能効率性に関する知識領域ごとの品質技術数と, その内で本製品開発に適用した数を図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。

知識領域	副知識領域	トピックス	システム/ソフトウェア製品品質										
			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			
9 テストの技法	5 基つ利用に した技法	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					
		5 オブジェクト指向テスト	①										
		6 Webシステムのテスト	①	①			①	①	①				
	6 ソフトウェアの 形態に 基ついた技法	7 GUIテスト	①										
		8 サーバーサイドのテスト	①	①				①					
		9 データベーステスト	①										
		10 並行プログラムのテスト	①					①					
		11 プロトコル適格性テスト	①										
		12 実時間のテスト	①						①				
		13 モバイルアプリケーションのテスト	①	①								①	

図9 品質ボックスから抜粋

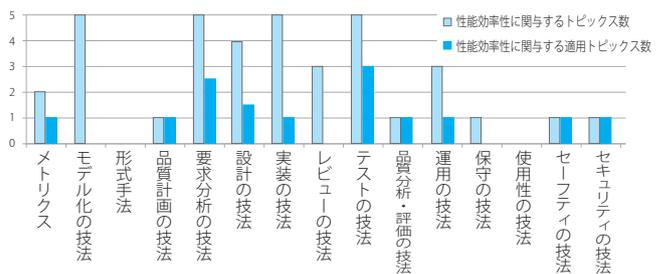


図10 適用した品質技術数(性能効率性)

製品Bも同様に最優先とする信頼性を高めるために適用する品質技術を決定した。

(3) 手順 d: 決定メトリクスの測定と達成度評価

品質ボックスのRISEメトリクスの内, 検証対象の製品A, 製品Bが最優先とした品質特性に関する測定値をそれぞれ図11に示す。どちらも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。

製品Aは, 性能効率性についてRISEメトリクスで推奨されている大小比較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分かる。一方, 製品Bは信頼性について可用性/障害許容性は適切なメトリクスが得られているが, 成熟性の結合試験とシステム試験密度/結合不具合発見率について目標とする水準に達していないことが分かる。この点について開発チームの調査を行ったところ, チーム内で用いている指標値において, プラットフォーム特性に応じた準備ができていなかったことが判明した。このことから, 品質確保プロセスの適用によって, 開発における課題が抽出できたと言える。

6.2 品質ボックス適用評価

図8の手順eについて, 製品Aと製品BのRISEベンチマークとしてそれぞれ特徴のある一部の品質特性に関し中央値と該当製品のポジションを図12, 図13に示し, その分析

RISE メトリクス (性能効率性)											
製品 A	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 利用率最大値	メモリ利用率最大値	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	-	ユーザ数
	-	-	-	データ検索 (500 件)	-	-	-	データ検索	データ検索	-	-

RISE メトリクス (信頼性)													
製品 B	成熟性						可用性		障害許容性				
	単体不具合除去率	結合不具合除去率	シス不具合除去率	単体不具合発見率	結合不具合発見率	シス不具合発見率	予測数 / 不具合数	結合試験：実際 / 目標	シス試験：実際 / 目標	運用試験有無	運用実時間	異常系結合試験成功率	異常系シス試験成功率
大 or 小が望ましい	大	大	大	小	小	小	-	大	大	大	大	大	大
メトリクス測定値	-	1.00	1.00	NA	0.56	1	1.01	0.42	0.45	1.00	1.00	0.97	0.96
測定条件 (サンプル)	試験項目数 / 障害数 / 除去数									クラウドサービス環境	MTBF	アプリ異常等	サーバーインフラ故障等

図 11 メトリクス測定結果

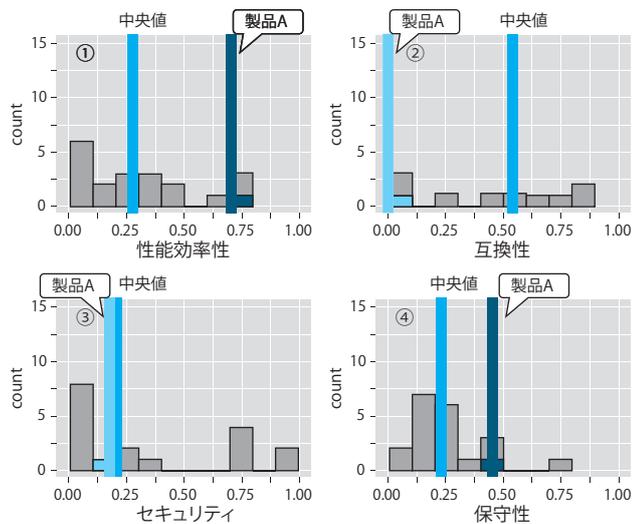


図 12 品質特性のベンチマーク結果 (製品 A)

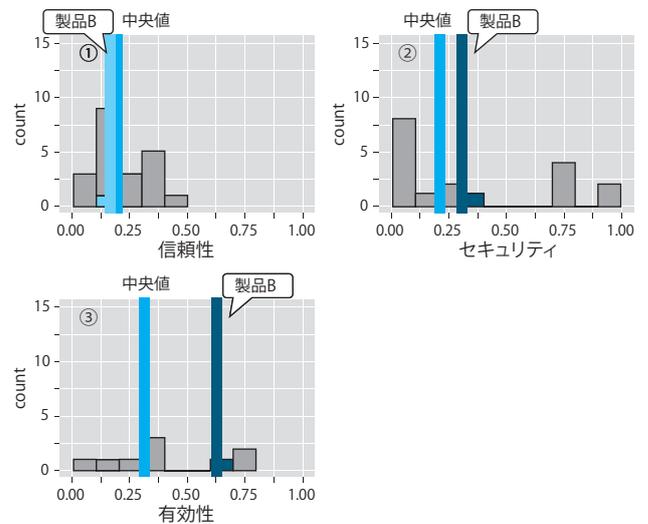


図 13 品質特性のベンチマーク結果 (製品 B)

結果と考察を示す。各グラフは、RISE ベンチマークによってメトリクス測定値をスコア化し、21 種類のお製品の品質指標の分布と、当該製品の位置を示している。

(1) 製品 A

製品 A の RISE メトリクスによるベンチマーク結果を図 12 に示し評価する。

① 性能効率性

結果：中央値を上回り、他製品より優れている。

考察：最優先とした品質特性 (表 1) であり、品質ボックスの性能効率性に寄与する品質技術からとくにテスト技法を多く取り入れ (図 10)、十分な検証を行った成果であり、製品としての強みが確保できた。

② 互換性

結果：中央値を下回り、優位性が低い結果となった。

考察：低優先とした品質特性 (表 1) であり、品質ボッ

クスによる品質技術の選択を行わなかった。他製品のスコアが広範囲で分布している特徴があり、エンハンス時には品質ボックスにより意図的に品質技術を利用することで、優位性を高められると考える。

③ セキュリティ

結果：中央値とほぼ同等の優位性をもつ。

考察：低めの優先度とした品質特性 (表 1) であり、品質ボックスによる品質技術の選択を行ったが適用範囲を一部としたため、本結果になったと考える。なお、本メトリクスにおいては、スコア分布が二極化している特徴が見られる。本製品の優位性を高めるためにも、次回エンハンス時は、品質ボックスから選択した技術の適用範囲拡大を検討する。

④ 保守性

結果：中央値を上回り、他製品より優れている。

考察：高優先とした品質特性 (表 1) であり、品質ボッ

クスから選択した品質技法としてモデル化の技法や設計の技法を多く取り入れた成果であり、製品としての優位性が確保できた。

(2) 製品 B

製品 B の RISE メトリクスによるベンチマーク結果を図 13 に示し評価する。

① 信頼性

結果: 中央値をやや下回った。

考察: 運用強化として最優先とした品質特性 (表 2) であるが、品質ボックスによる品質技術の選択を行なっておらず、メトリクスの数値から弱点であることが判明した。今後は品質技術の適用を考える。

② セキュリティ

結果: 中央値を上回ったものの、ほぼ平均にある。

考察: 運用強化として優先とした品質特性 (表 2) であり、品質ボックスによる品質技術の選択もしたが、他分布をみるとログイン認証方式の多様性など、更なる機能追加の検討が必要と認識できた。

③ 有効性

結果: 中央値を上回り、かなりの優位性を持つ。

考察: 競争力強化として、最優先とした品質特性 (表 2) であり、品質ボックスによる品質技術としては、要求分析の技法を選択。既存ユーザからの要望の取り入れや、要件の妥当性評価 (第三者試験) の重視が結果につながったと考えられる。

6.3 考察

従来、ソフトウェア開発に採用する品質技術は類似製品開発の経験などから選ぶことが多かった。今回、ソフトウェアに期待される要件から品質特性を特定し、その実現に寄与できる品質技術から適切な品質技術を選ぶプロセスを提案し、一定の効果が得られたと考える。また、ベンチマーク結果から効果が得られなかった点は、その原因を深堀することにより品質ボックスやメトリクス目標値にフィードバック可能であると考えられる。

7 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・検証した。提案手法を 2 種類の異なる製品に適用評価し、重要な品質指標の抽出や、製品の品質評価において有効であることを示した。

今後は、今回提案した品質確保プロセスの開発現場で実践を進めると同時に、IoT や AI などの新しい技術に要求される品質技術と品質特性の関係性の体系的整理を進めていく予定である。それにより、様々な製品に対して、最適な品質技術の選択決定が可能になると考えられる。

謝辞: 本稿の執筆にあたり、日科技連 SQiP ソフトウェア品質委員会、SQuBOK V3 研究チームの方々、富士通、富士通研究所、富士通ビー・エス・シーの RISE 研究関係者に多くのご協力、ご助言を頂いた。ここに謝意を記す。

【参考文献】

- [早稲田大学 2017] 鷲崎弘宜, 「異なる品質間関係を総合的に実証した世界初のベンチマーク (WSQB2017)」, 先導的研究支援事業 (RISE), 2017 [WSQB] WSQB17:Waseda Software Quality Benchmark, http://www.washi.cs.waseda.ac.jp/?page_id=3479
- [品質ボックス 2017] 品質技術と品質特性とメトリクスの関係性モデル。筆者らが公開したもの。
http://www.washi.cs.waseda.ac.jp/wp-content/uploads/2017/07/SQIP_Quality_Box.pdf
- [Gordon2005] Dan Gordon, Ted Stehney, Neha Wattas, Eugene Yu, "System Quality Requirements Engineering (SQUARE): Case Study on Asset Management System, Phase II", CMU/SEI-2005-SR-005
- [Firesmith2006] Donald Firesmith, "QUASAR: A Method for the Quality Assessment of Software-Intensive System Architectures", CMU/SEI-2006-HB-001
- [Jagdish2002] Jagdish Bansiya and Carl G. Davis, "A hierarchical model for object-oriented design quality assessment", IEEE Transactions on Software Engineering, Vol.28, No.1, 2002
- [Bass2005] Len Bass, Paul Clements, Rick Kazman, 前田卓雄・佐々木明博 (訳), 実践ソフトウェアアーキテクチャ, 日刊工業新聞社, 2005.
- [山田 2016] 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. [Zhu2005] Hong Zhu, "Software Design Methodology: From Principles to Architectural Styles", Elsevier, 2005
- [SQuBOK2014] 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.
- [ISO25010] ISO/IEC 25010:2011 Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models
- [ISO25012] ISO/IEC 25012:2008 Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Data quality model.
- [ISO25011] ISO/IEC TS 25011:2017 Information technology -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Service quality models
- [小島 a 2017] 小島 嘉津江, 森田 純恵, 若本 雅晶, 宗像 一樹, 鷲崎 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
- [小島 b 2017] 小島 嘉津江, 森田 純恵, 菊池 慎司, 鷲崎 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と活用の一考察, 日科技連ソフトウェア品質シンポジウム 2017, A4-1