Vol.2017-SE-195 No.30
2017/3/13

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 嘉津江[†1]　森田 純恵[†2]　若本 雅晶[†2]　宗像 一樹[†2]　鷲崎 弘宜[†3]

概要：ソフトウェア品質の向上に関しては，古くから様々な技術が提案・実用化されてきており，ソフトウェア開発方法の進歩に伴う新たな技術の開発も進んでいる．一方，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も研究が進んでおり，ソフトウェア品質モデルおよび品質特性が国際規格として規定されている（ISO/IEC 25000 シリーズ）．これら個々の品質技術と，ISO/IEC 25000 シリーズで規定された各品質特性の間には，たとえば，この技術はこの特性の向上に特に効果があるといった関係性が当然存在し，これはソフトウェア開発において重要な情報となり得る．しかし，これまで，このような関係性を網羅的に明確化する試みは行われておらず，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あった．

本稿では，個々の品質技術が品質特性に与える効果を網羅的に見える化することを目的として，品質技術と品質特性のマッピングを試みた．マッピングに当たっては，品質技術の網羅・体系化が必要となるが，世界で初めてこれを実現した日本発のガイドである「ソフトウェア品質知識体系ガイド（SQuBOK ガイド）」を参照することとした．SQuBOK ガイドと ISO/IEC 25000 シリーズをベースにしたマッピングの結果，品質向上に寄与する技術を品質特性ごとに効率的に選択できるようになり，ツールとしても有効であることがわかった．また，マッピング結果の分析を通して，品質特性による品質技術の偏りからわかる拡充すべき技術領域，データ品質に関する技術の体系化不足，OSS 活用の技法など最新技術の追加・体系化の必要性など，今後の課題も明確になった．

キーワード：ソフトウェア品質技術，ソフトウェア品質特性，ソフトウェア品質知識体系ガイド（SQuBOK ガイド）

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と考えられる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ることは，これまでの経験から周知の事実であり，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．

ソフトウェアの品質は，①ソフトウェア製品そのものに対する要求，②利用時の視点での要求，そして，③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ている．具体的には以下のとおりである．

① ソフトウェアで実現する機能の高度化や利用シーンの多様化に伴い，ソフトウェア製品は，単に機能不良がないというだけではなく，セイフティやセキュリティ要件を満たすことも品質の要素として求められるようになってきた．また，ソフトウェアの動作条件・環境を固定化するのではなく，変化し続ける多種多様な環境への展開や移行が速やかにできることもソフトウェア製品が具備する品質の一つとして求められるようになっている．

② 近年，様々なサービスをソフトウェア製品利用により実現することが多くなっており，ソフトウェア製品の品質を利用時の視点で見極め，高めていくようになってきた．

③ 市場競争力を高めてより多くの顧客満足を得るために，ソフトウェア製品を提供するスピードやコストの重要性が増しており，アジャイル開発や OSS・外購品の組み込みなどに代表されるように，ソフトウェアの開発プロセスが変化している．ウォータフォール開発で仕様を確定して上流工程で品質を作り込み，ホワイトボックス試験により堅実に品質を確保する方法からの思い切った脱却と，トレードオフの関係にある品質とスピード・コストを両立させることが期待されている．

ソフトウェア品質の向上に関しては，古くから様々な技術が提案・実用化されてきており，ソフトウェア開発方法の進歩に伴う新たな技術の開発も進んでおり，これらは「ソフトウェア品質知識体系ガイド（SQuBOK[a]ガイド）」[1]で整理・体系化されている．一方，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も研究が進んでおり，ソフトウェア品質モデルおよび品質特性が国際規格として規定されている（ISO/IEC 25000 シリーズ）．これら個々の品質技術と各品質特性の間には，たとえば，この技術はこの特性の向上に特に効果があるといった関係性が当然存在し，これはソフトウェア開発において重要な情報となり得る．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[2]．しかし，「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」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

†1 富士通株式会社
Fujitsu Limited
†2 株式会社富士通研究所
Fujitsu Laboratories Ltd.
†3 早稲田大学
Waseda University

a) Software Quality Body of Knowledge

ⓒ 2017 Information Processing Society of Japan

などの課題が残っている．

本稿では，個々の品質技術が品質特性に与える効果を網羅的に見える化することを目的として，品質技術と品質特性のマッピングを試みた．その結果，品質向上に寄与する技術を品質特性ごとに効率的に選択できるようになり，ツールとしても有効であることがわかった．また，マッピング結果の分析を通して，品質技術やその体系化に関する今後の課題も明確になった．

本稿の構成は以下のとおりである．第 2 章ではソフトウェア品質技術を SQuBOK ガイドに則り整理する．第 3 章ではソフトウェアに求められる品質特性を ISO/IEC 25000 シリーズに基づいて体系的に捉える．第 4 章ではこれらの品質技術と品質特性の関係性を網羅的にマッピングし，ソフトウェア品質技術が品質特性に与える効果を見える化する．第 5 章ではマッピングの結果を考察し，第 6 章でマッピングの活用と効果について述べる．最後に，SQuBOK ガイドに関する今後の課題をまとめる．

2. SQuBOK ガイドによる品質技術の体系化

ソフトウェア品質技術は，ソフトウェア開発手法の進歩に伴い，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当たっては，品質技術の網羅・体系化が必要となるが，世界で初めてこれを実現した日本発のガイドである SQuBOK ガイドを参照することとした．

2.1 SQuBOK ガイド

SQuBOK ガイドは，図 1 に示すように，知識体系を「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 つのカテゴリに大別している．

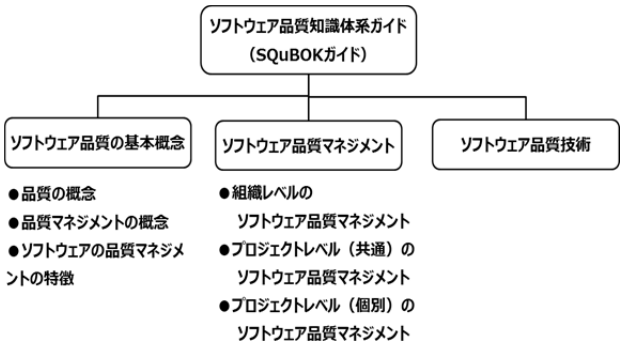

図 1　SQuBOK ガイドの樹形図概観

- ソフトウェア品質の基本概念：「品質の概念」,「品質マネジメントの概念」,「ソフトウェアの品質マネジメントの特徴」といった基本的な概念や考え方を整理している．
- ソフトウェア品質マネジメント：品質をマネジメントするアクティビティを「組織レベル」,「プロジェクトレベル（工程共通）」,「プロジェクトレベル（工程個別）」の 3 つの副カテゴリで構成し，それぞれの知識領域を整理している．
- ソフトウェア品質技術：ライフサイクルの中で用いることができるソフトウェア品質に関する具体的な技術を整理している．詳細を 2.2 項で述べる．

2.2 ソフトウェア品質技術

SQuBOK ガイドでは，品質の作り込みや確認のための多くの技術をソフトウェア品質技術として体系的に整理している．図 2 に「ソフトウェア品質技術」カテゴリの樹形図概観を示す．なお，紙面の都合上，副知識領域とトピックスは一部の例を記載した．

ソフトウェア品質技術は，開発工程に着目した「工程に共通な品質技術」と「工程に個別な品質技術」，ならびに，最近の品質トレンドである使用性やセキュリティなどに着目した「専門的品質特性の品質技術」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，それぞれ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 個の品質技術（トピックス）が副知識領域ごとに整理されている．

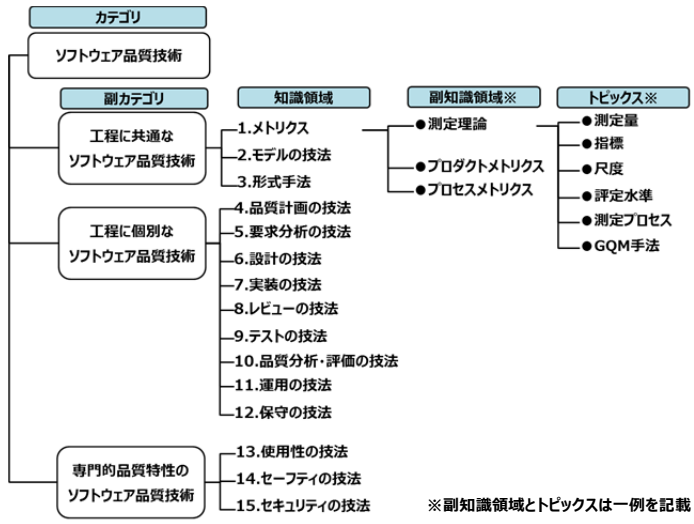

図 2　「ソフトウェア品質技術」カテゴリの樹形図概観

2.3 ソフトウェア製品開発における品質技術の適用

ソフトウェア製品開発の要件としては，表 1 に示すような，対象製品への機能的な要求事項・ニーズのほか，制約条件（方針制約，市場制約，資源制約など）やプロジェクトの特徴などがある．

ⓒ 2017 Information Processing Society of Japan

表 1　ソフトウェア製品開発における要件（例）

要件分類	要件例
機能的な要求事項・ニーズ	実現する機能，オンラインシステム，オフラインシステム，データ管理システム，Webアプリケーション，社会基盤システム　など
制約条件（方針制約，市場制約，資源制約など）	戦略製品，大規模ユーザ，官公庁向け，短納期，高品質，高信頼，低コスト，競合製品・競合企業，協業企業，人材，職場環境，開発環境　など
プロジェクトの特徴	大規模，新技術適用（IoT，AIなど），オフショア開発，新開発手法適用，システム更改　など

これらの要件を満たすために，開発計画立案時点で的確なソフトウェア品質技術を選択することが必要であり，これを実現できるか否かが，ソフトウェアの品質確保とプロジェクト成功の鍵を握ると考えられる．たとえば，表 1 における「プロジェクトの特徴」の「システム更改」に関しては，「設計の技法」（図 2 の知識領域 No.6）にある複数のトピックス（品質技術）から，どの技術をどのような考え方で選ぶのが最適かを検討し決定することが，プロジェクト開始時に要求される．

3. ソフトウェア品質のモデル化と品質特性

3.1 ISO/IEC 25000 シリーズ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，ソフトウェア品質を客観的に評価するための尺度として，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]

図 3 に各品質モデルがカバーする対象領域を示す[3]. システムを構成するソフトウェアとデータ，ならびに利用に関わる部分に対して品質モデルが適用される．

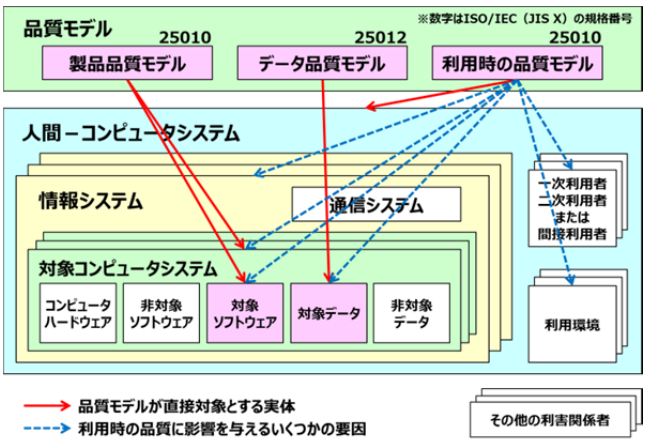

図 3　品質モデルの対象 [3]

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 4 と図 5 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 2 はデータ品質モデル特性を示しており，データ品質特性を固有の視点からのものとシステム依存の視点からのものに分類していることが特徴となっている．

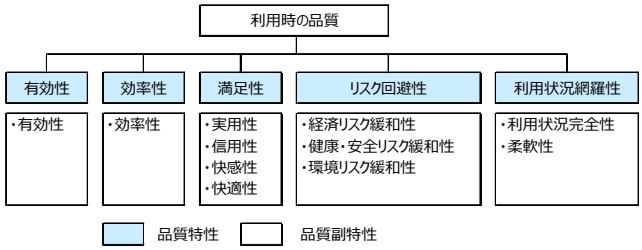

図 4　利用時の品質モデル [3]

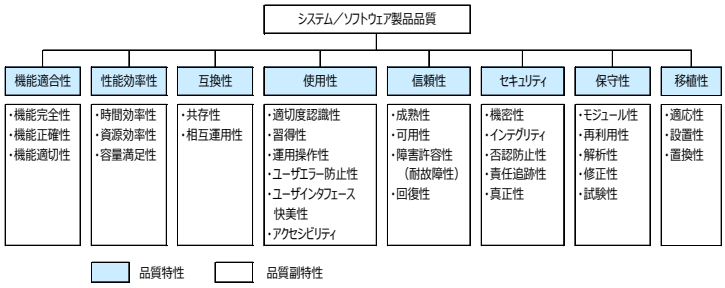

図 5　製品品質モデル [3]

表 2　データ品質モデル特性 [4]

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)		○

3.2 ソフトウェア製品開発における品質特性の活用

昨今のソフトウェア製品開発においては，従来からの製品そのものの品質に加え，利用時の視点での品質が強く求められ，単に製品品質が良いのみでなく，利用時の評価が重要になってきている．また，ISO/IEC 25000 シリーズの製品品質モデルでは，先行規格である ISO/IEC 9126 シリーズの品質モデルから「互換性」と「セキュリティ」が追加になっている．これは，昨今のソフトウェア製品に求めら

ⓒ 2017 Information Processing Society of Japan

れる品質特性が多様化していること，また，IoT を介した大量データの処理と，これを活用した AI によるソリューションの品質を高めるためにも重要な品質特性であることを意味する．

これらのことから，ソフトウェア製品開発における要件（表 1）が，品質特性と密接に結びついていることがわかる．たとえば，従来システムを新環境に載せ替えるプロジェクトの特徴である「システム更改」は，「互換性」や「機能適合性」などの品質特性が重要視されると考えられる．また，IoT や AI のように各所に散在するデータをもとにシステム化するケースでは，「データ品質」や「セキュリティ」が重要になると考えられる．

このように，ソフトウェア製品開発における要件を満たすための品質特性を明確にし，その品質特性を評価項目に加えることが，プロジェクトを成功に導く一つのポイントになると考えられる．

4. 品質技術と品質特性のマッピング

第 2 章で述べた品質技術と第 3 章で述べた品質特性のマッピングを試みる．マッピングによって，たとえば，個人や組織において保有する知識・スキルが十分であるかの評価や，ある知識に対して複数の品質特性がマッピングされた場合，それらを矛盾なく扱えるかのトレードオフ分析[5][6]が可能になると期待できる．

4.1 トピックスのマッピング

SQuBOK ガイドの品質技術は，15 個の知識領域と 38 個の副知識領域に分類され，全部で 142 個の技術がトピックスとして定義されている．一方，品質特性は，利用時の品質モデル，システム/ソフトウェア製品品質モデル，データ品質モデル合わせて，28 個の品質特性，57 個の品質副特性に整理されている．

これらを相互にマッピングし，品質技術と品質特性の関係性を網羅的に見える化した．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 ガイドには，データ品質の技術についてほとんど言及されておらず，記載されている技術が完全性と機密性に関連するもののみのためである．

トピックス 142 個を縦軸に，品質特性 28 個を横軸にマッピングした全体図を付録 A.1 に添付する[b]．●印は，当該品質技術において，カバーできる品質特性であることを示している．●印は，著者の 1 名が「SQuBOK ガイドに品質特性との関係が明記されていること」を基準に初期案を作成し，他の著者のレビューによる追加・修正の後，最終案とした．なお，本マッピングは，著者らの経験・主観的な判断によるところが大きく，今後，事例や判断を積み重ね，さらには一般からのレビューを経て改訂していきたい．

4.2 知識領域のマッピング

図 6 は，付録 A.1 の縦軸を知識領域（15 個）のレベルに集約し，知識領域と品質特性のマッピングを示したものである．●印は，知識領域がカバーしている品質特性を示す．

SQuBOKガイド第2版をベースに作成				利用時の品質					ソフトウェア品質特性 システム／ソフトウェア製品品質								データ品質		
カテゴリ	副カテゴリ		知識領域	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	工程に共通なソフトウェア品質技術	1	メトリクス	●	●	●	●	●	●	●	●	●	●	●	●	●			
		2	モデル化の技法						●	●			●		●	●			
		3	形式手法						●				●						
	工程に個別なソフトウェア品質技術	4	品質計画の技法	●	●	●	●		●	●	●	●	●	●	●	●			
		5	要求分析の技法	●	●	●	●	●	●	●	●	●	●	●	●	●			
		6	設計の技法						●	●	●	●	●	●	●	●			
		7	実装の技法						●	●			●	●	●	●			
		8	レビューの技法				●	●	●	●		●	●	●	●	●			
		9	テストの技法					●	●	●		●	●	●			●	●	
		10	品質分析・評価の技法						●	●		●	●		●	●			
		11	運用の技法							●		●	●	●	●				
		12	保守の技法							●	●				●				
	特定の専門的品質特性のソフトウェア品質技術	13	使用性の技法					●				●							
		14	セーフティの技法				●	●		●		●	●	●					
		15	セキュリティの技法					●		●		●	●	●					

図 6　知識領域と品質特性のマッピング

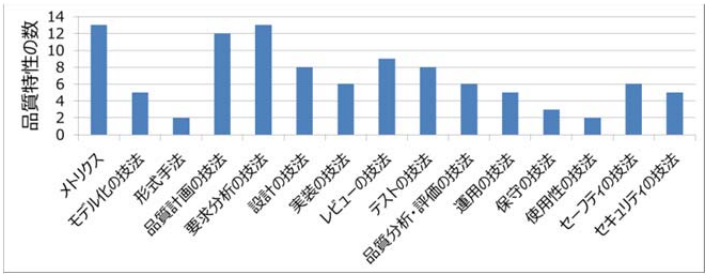

図 7 各知識領域がカバーしている品質特性の数

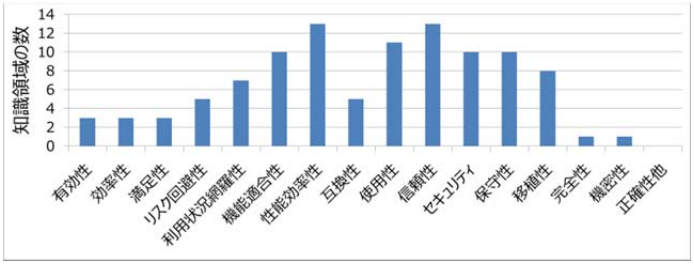

図 8　各品質特性に寄与している知識領域の数

図 6 に示したマッピング結果から，知識領域と品質特性が相互にどの程度カバーしているかを示したものが，図 7 と図 8 である．図 7 は，各知識領域がカバーしている品質特性の数を示し，図 8 は，各品質特性に寄与している知識領域の数を示す．たとえば，図 7 では，知識領域「メトリクス」が 13 個の品質特性をカバーできることを示す．また，

b) 実際のマッピングでは，横軸を品質副特性（57 個）としたが，付録 A.1 では紙面の都合上，横軸を品質特性（28 個）に集約した．集約に当たっては，ある品質特性の下位の品質副特性の中で一つでも●印が付けば，その品質特性に●を付けるというルールとした．

ⓒ 2017 Information Processing Society of Japan

図 8 では，利用時の品質「有効性」に寄与する知識領域が 3 つであることを示す．

5. マッピング結果の考察

4.2 項で述べたマッピングの結果から明らかになった，知識領域と品質特性の間の網羅性の特徴を表 3 に示す．

表 3　知識領域と品質特性の間の網羅性の特徴

網羅性の特徴	該当する知識領域および品質特性
①カバーする品質特性が多い（多面的）知識領域	メトリクス 品質計画の技法 要求分析の技法
②カバーする品質特性が少ない（特定の特性にフォーカスしている）知識領域	形式手法 保守の技法 使用性の技法
③寄与している知識領域が多い品質特性	機能適合性 性能効率性 使用性 信頼性 セキュリティ 保守性
④寄与している知識領域が少ない品質特性	互換性 利用時の品質特性が少ない データ品質特性が全体的に少ない

特徴②に該当する知識領域は，特定の品質特性にフォーカスしている．しかしながら，これらの領域が影響を及ぼす品質特性は他にもあると考えられる．たとえば，「保守の技法」には，「プログラム理解」や「リエンジニアリング」などのトピックスがある．これらは単に保守性をカバーするだけでなく，機能適合性や信頼性にも影響を及ぼすと考えられる．このような領域を深堀し，品質特性との関連性を深めると，さらなる品質の底上げに寄与できると考える．

特徴④に関しては，以下の課題があると考える．

- 互換性に寄与する技術が少ない： 互換性は ISO/IEC 25000 シリーズで強化された品質特性であるが，これをカバーする品質技術が少ない．昨今は，スピードやコストの面から，新しいシステム開発よりも更改を指向する傾向にあり，互換性をカバーできる品質技術の整理，発掘が急務である．
- 利用時の品質に寄与する技術が少ない： サービスをソフトウェアで実現することが多くなっているため，サービス利用の視点で見た品質は重要であり，これに寄与する技術の整理，発掘が急務である．
- データ品質に寄与する技術が全体的に少ない： 図 3 に示すとおり，データ品質モデルの対象はデータに特化しているため，寄与する品質技術もデータベース関連のものが主体になると考えられる．今後の IoT や AI によるデータ活用型のシステム/ソフトウェア開発において，データ品質とこれに寄与する技術の重要性はますます高まると考えられる．

6. マッピングの活用と効果

図 6 に示したマッピング結果を開発計画時の手法検討の際に利用することで，品質向上に効果のある品質技術を品質特性ごとに効率的に選択できるようになる．また，個々の品質技術が寄与する品質特性をマッピングにより整理したことで，各技術の効果が見える化でき，技術強化のポイントも明確になった．これらのことから，このマッピング結果はツールとしても有効であることがわかった．

たとえば，図 9（付録 A.1 の抜粋）において，「方式設計の技法」は 5 つのトピックスを持ち，それぞれがカバーする品質特性が異なっている．2.3 項および 3.2 項で示したように，ソフトウェア製品開発における要件（表 1）から必要な品質特性を導き出し，最適な「方式設計の技法」を選ぶことが可能になる．

SQuBOKガイド第2版をベースに作成							ソフトウェア品質特性							
							システム／ソフトウェア製品品質							
カテゴリ	副カテゴリ	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
3．ソフトウェア品質技術	工程に個別なソフトウェア品質技術	6	設計の技法	1	方式設計の技法	1 部品化の技法	●				●		●	
						2 アーキテクチャパターン	●	●	●	●	●	●	●	●
						3 品質に基づくアーキテクチャ設計・評価技法	●	●	●	●	●	●	●	●
						4 DSM（依存関係マトリクス）							●	
						5 フレームワーク	●	●			●		●	●
				2	詳細設計の技法	1 テスト駆動開発							●	
						2 デザインパターン		●				●	●	
						3 設計原則	●						●	
		7	実装の技法	1	-	1 コーディング規約	●	●			●	●	●	●
						2 リファクタリング	●	●					●	●
						3 契約による設計	●	●					●	●
						4 IDE（統合開発環境）	●	●					●	●
						5 CI（継続的統合）	●	●					●	●

図 9　方式設計の技法の効果

SQuBOKガイド第2版をベースに作成				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									データ品質		
カテゴリ	副カテゴリ		知識領域	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	工程に共通なソフトウェア品質技術	1	メトリクス	●	●	●	●	●	●	●	●	●	●	●	●	●			
		2	モデル化の技法						●	●			●		●	●			
		3	形式手法						●				●						
	工程に個別なソフトウェア品質技術	4	品質計画の技法	●	●	●	●		●	●	●	●	●	●	●	●			
		5	要求分析の技法	●	●	●	●	●	●	●	●	●	●	●	●	●			
		6	設計の技法						●	●	●	●	●	●	●	●			
		7	実装の技法						●	●			●	●	●	●			
		8	レビューの技法				●	●	●	●		●	●	●	●	●			
		9	テストの技法					●	●	●		●	●	●			●	●	
		10	品質分析・評価の技法						●	●		●	●		●	●			
		11	運用の技法							●		●	●	●	●				
		12	保守の技法							●	●				●				
	専門的特性のソフトウェア品質技術	13	使用性の技法					●				●							
		14	セーフティの技法				●	●		●		●	●	●					
		15	セキュリティの技法					●		●		●	●	●					

図 10　性能要件に寄与する知識領域

ⓒ 2017 Information Processing Society of Japan

また，図 10 に示されるように，性能要件が強く求められるシステム/ソフトウェアを開発する場合は，開発工程を網羅的に見た上で，適用する知識領域を検討することが可能になる．さらに，各知識領域においては，付録 A.1 を用いて，図 9 のようにトピックスまでブレークダウンして確認することで，「性能効率性」以外にカバーできる品質特性を把握することもできる．これにより，さらに効果的な品質技術の選択が可能になると考える．

このように，品質向上に寄与する技術を網羅的に把握することが可能になる一方，すべての技術を適用することは，コスト高，納期延伸などのトレードオフをもたらす可能性がある．これに関しては，知識領域には複数の技術が定義されているので，各技術がカバーする品質特性の範囲を考慮した上で，SQuBOK ガイドに記載の技術内容を確認し，どれを適用するのが最適かを判断することで，かなりの部分は解決できると考える．

7. おわりに

ソフトウェア品質技術と品質特性のマッピングにより，品質技術が品質特性に与える効果を見える化した．また，この結果が，システム/ソフトウェア製品の開発において，どの品質技術を適用すべきかの判断に活用できることを示した．

一方，今回のマッピングを通して，現状の SQuBOK ガイド（第 2 版）に以下の課題があることもわかった．

- ●品質特性によって，寄与する品質技術の数に偏りが見られる．技術そのものが足りないのか，技術の抽出・整理が不十分なのかを明確にした上で，技術の拡充を図ることが期待される．
- ●データ品質に関する技術とその体系化の拡充が必要である．
- ●OSS 活用の技法（目利きに関する技法，設計に関する技法，品質保証に関する技法）[7]など，最新技術の追加・体系化が必要である．
- ●各品質技術（トピックス）の説明において，品質特性との関係に関する記術方法や表現のばらつきを是正する必要がある．

これらの課題に取り組むことにより，SQuBOK ガイドの拡充が図れるとともに，システム/ソフトウェア製品開発において，開発要件を満たしつつ，より良い品質の製品を提供するための指針が得られると考える．

謝辞 本稿の作成にご協力頂いた皆様に，謹んで感謝の意を表する．

参考文献

[1] 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.
[2] 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.
[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] 鷲崎弘宜，ソフトウェア品質プロフェッショナルに求められる専門性と倫理性, SEC journal/情報処理推進機構, Vol.10 No.5, pp.32-37，2015.
[6] CMU Software Engineering Institute, Architecture Tradeoff Analysis Method, http://www.sei.cmu.edu/architecture/tools/evaluate/atam.cfm
[7] 森田純恵，菊池慎司，水谷拓人，伊藤雅子，土屋雅生，野中誠，OSS を活用したエンタープライズソフトウェア開発向けメトリクスの抽出，ソフトウェア品質シンポジウム 2016, Software Quality Profession (SQiP)/日本科学技術連盟，2016.

ⓒ 2017 Information Processing Society of Japan

付録

付録 A.1 品質技術（トピックス）と品質特性のマッピング　（1/2）

SQuBOKガイド第2版をベースに作成								ソフトウェア品質特性														
								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		
カテゴリ	副カテゴリ	知識領域		副知識領域		トピックス		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	工程に共通なソフトウェア品質技術	1	メトリクス	1	測定理論	1	測定量																
						2	指標																
						3	尺度																
						4	評定水準																
						5	測定プロセス																
						6	GQM手法																
				2	プロダクトメトリクス	1	内部メトリクス	●	●	●	●	●	●	●	●	●	●	●	●	●			
						2	外部メトリクス	●	●	●	●	●	●	●	●	●	●	●	●	●			
						3	利用時の品質メトリクス	●	●	●	●	●											
						4	複雑度のメトリクス												●	●			
						5	LOC（ソースコード行数）																
						6	ファンクションポイント						●										
				3	プロセスメトリクス	-	-																
		2	モデル化の技法	1	離散系のモデル化技法	1	UML						●	●			●		●	●			
						2	SysML						●	●			●		●	●			
						3	構造化チャート（PAD）						●	●			●		●	●			
				2	連続系のモデル化技法	-	-																
				3	ドメイン特化言語	-	-																
		3	形式手法	1	形式仕様記述の技法	-	-						●										
				2	形式検証の技法	-	-						●				●						
	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	費用便益分析																
						2	品質計画書	●	●	●	●		●	●	●	●	●	●	●	●			
		5	要求分析の技法	1	要求抽出	1	ステークホルダー識別					●											
						2	要求開発（Openthology）	●	●	●	●	●											
				2	要求分析	1	機能要求分析						●					●					
						2	非機能要求分析							●		●	●	●	●				
						3	品質機能展開						●	●	●	●	●	●	●	●			
						4	要求可変性分析												●				
				3	要求仕様化	1	ソフトウェア要求仕様						●						●				
						2	USDM（要求仕様記述法）						●	●									
				4	要求の妥当性確認と評価	-	-																
		6	設計の技法	1	方式設計の技法	1	部品化の技法						●				●		●				
						2	アーキテクチャパターン						●	●	●	●	●	●	●	●			
						3	品質に基づくアーキテクチャ設計・評価技法						●	●	●	●	●	●	●	●			
						4	DSM（依存関係マトリクス）												●				
						5	フレームワーク						●	●			●		●	●			
				2	詳細設計の技法	1	テスト駆動開発												●				
						2	デザインパターン							●				●	●				
						3	設計原則						●						●				
		7	実装の技法	1	-	1	コーディング規約						●	●			●	●	●	●			
						2	リファクタリング						●	●					●	●			
						3	契約による設計						●	●					●	●			
						4	IDE（統合開発環境）						●	●					●	●			
						5	CI（継続的統合）						●	●					●	●			
		8	レビューの技法	1	レビュー方法	1	ピアレビュー						●										
						2	インスペクション						●										
						3	チームレビュー						●										
						4	ペアプログラミング						●										
						5	ピアデスクチェック						●										
						6	パスアラウンド						●										
						7	ラウンドロビンレビュー						●										
						8	ウォークスルー						●										
						9	アドホックレビュー						●										
				2	仕様・コードに基づいた技法	1	形式手法に基づくレビュー						●										
						2	インターフェース分析						●										
						3	複雑度分析						●						●				
						4	パストレース						●										
						5	ラン・スルー						●										
						6	制御フロー分析						●	●					●				
						7	アルゴリズム分析						●	●									
						8	モジュール展開						●						●				
						9	七つの設計原理【高士通】						●										
						10	静的解析						●				●	●		●			
				3	フォールトに基づいた技法	1	ソフトウェアFMEA, FMECA						●				●						
						2	FTA（フォールトの木解析）				●	●	●				●						
						3	EMEA（エラーモード故障解析）						●			●	●						
						4	CFIA（構成要素障害影響分析）【IBM】						●				●						
						5	PQ（パタン・キュー）デザインレビュー【日立】						●	●			●						

付録 A.1 品質技術（トピックス）と品質特性のマッピング　（2/2）

SQuBOKガイド第2版をベースに作成								ソフトウェア品質特性														
								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		
カテゴリ	副カテゴリ	知識領域		副知識領域		トピックス		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	工程に個別なソフトウェア品質技術	9	テストの技法	1	経験及び直感に基づいた技法	1	アドホックテスト						●										
						2	探索的テスト						●										
				2	仕様に基づいた技法	1	ブラックボックステスト						●										
						2	同値分割						●										
						3	境界値分析／境界値テスト						●										
						4	デシジョンテーブルによるテスト						●										
						5	原因結果グラフによるテスト						●										
						6	状態遷移テスト						●										
						7	ランダムテスト						●				●						
						8	モデルベースドテスト						●										
						9	要求分析技法【富士通】						●										
						10	CFD技法						●										
				3	コードに基づいた技法	1	ホワイトボックステスト						●										
						2	制御フローテスト						●										
						3	データフローテスト						●										
						4	トランザクションフローテスト						●										
				4	フォールトに基づいた技法	1	エラー推測テスト						●										
						2	ミューテーションテスト						●										
				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					●	●				●						
						2	ローカライゼーションテスト						●										
						3	ユーザー環境シミュレーションテスト						●	●		●							
						4	整合性確認テスト						●	●			●						
				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						●										
						2	Webシステムのテスト						●	●		●	●	●					
						3	GUIテスト						●										
						4	サーバーサイドのテスト						●	●			●						
						5	データベーステスト						●									●	
						6	並行プログラムのテスト						●				●						
						7	プロトコル適格性テスト						●										
						8	実時間のテスト						●				●						
						9	モバイルアプリケーションのテスト						●	●				●					●
				7	組み合わせの技法	1	直交配列表（実験計画法）を用いたテスト						●										
						2	All-pair法を用いたテスト						●										
				8	リスクに基づいた技法	1	テストマネジメントにおけるリスクベースドテスト						●										
						2	テスト設計におけるリスクベースドテスト						●										
				9	テスト技法の選択と組み合わせ	1	機能的なテスト設計と構造的なテスト設計の組み合わせ						●										
						2	確定的なテスト設計と非確定的なテスト設計の組み合わせ						●										
				10	テスト自動化技法	-	-						●										
			品質分析・評価の技法	1	信頼性予測に関する技法	1	ソフトウェア信頼性モデル						●				●						
						2	ソフトウェア信頼度成長モデル						●				●						
						3	Fault-Prone分析						●										
						4	欠陥除去モデル						●										
						5	探針【日立】						●				●						
				2	品質進捗管理に関する技法	1	異常値管理																
						2	工数・成果モデル																
						3	コード統合パターン																
						4	PTRサブモデル																
						5	PTR発生及びバックログ予測モデル																
						6	Rayleighモデル						●				●						
				3	障害分析に関する技法	1	ODC（直交欠陥分類）						●			●	●						
						2	バグ分析						●				●						
				4	データ解析・表現に関する技法	1	QC七つ道具																
						2	新QC七つ道具																
						3	多変量解析																
						4	EDA（探索的データ解析）手法																
						5	p管理図、u管理図																
						6	品質座標																
						7	レーダーチャート						●	●		●	●		●	●			
		11	運用の技法	1	-	1	仮想化							●		●			●				
						2	クラウドによるサービス提供							●			●	●	●				
						3	ソフトウェア若化							●			●						
		12	保守の技法	1	-	1	プログラム理解												●				
						2	リエンジニアリング							●					●				
						3	リバースエンジニアリング								●				●				
						4	コードクローン分析												●				
	専門的品質特性のソフトウェア品質技術	13	使用性の技法	1	-	1	ユーザビリティテスト									●							
						2	インタラクティブシステムの人間中心設計プロセス（ISO 9241-210）					●				●							
		14	セーフティの技法	1	-	1	セーフティ実現のためのリスク低減技法				●	●		●		●	●	●					
						2	セーフティ・クリティカルシステムのテスト				●	●					●						
		15	セキュリティの技法	1	-	1	セキュリティ要求分析					●		●		●	●	●					
						2	セキュリティ設計										●	●					
						3	セキュリティパターン										●	●					
						4	セキュアコーディング											●					
						5	セキュリティテスト											●					

ⓒ 2017 Information Processing Society of Japan