

2015 年度ソフトウェア工学分野の先導的研究支援事業
「測定評価と分析を通じたソフトウェア製品品質の
実態定量化および総合的品質評価枠組みの確立」
成果報告書

平成 29 年 2 月
学校法人早稲田大学

本報告書は独立行政法人情報処理推進機構 技術本部 ソフトウェア高信頼化センターが実施した「2015年度ソフトウェア工学分野の先導的研究支援事業」の公募による採択を受け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）が実施した研究の成果をとりまとめたものである。

目次

研究成果概要	1
1 研究の目的・背景と期待される効果	12
1.1 研究目的とその背景	12
1.2 期待される効果	12
2 実施内容	14
2.1 研究アプローチ	14
2.1.1 研究の全体像	14
2.1.2 関連するこれまでの研究について	14
2.1.3 研究目標と研究課題	15
2.2 研究の活動実績・経緯	17
2.3 研究実施体制	21
3 研究成果	24
3.1 研究課題1「内部品質の測定評価方法の確立」	24
3.1.1 当初の想定	24
3.1.2 研究プロセスと成果	24
3.1.3 発生した問題および今後の展望	46
3.2 研究課題2「外部品質の測定評価方法の確立」	47
3.2.1 当初の想定	47
3.2.2 研究プロセスと成果	47
3.2.3 発生した問題および今後の展望	49
3.3 研究課題3「利用時の品質の測定評価方法の確立」	50
3.3.1 当初の想定	50
3.3.2 研究プロセスと成果	50
3.3.3 発生した問題および今後の展望	61
3.4 研究課題4「個々の品質実態把握」	62
3.4.1 当初の想定	62
3.4.2 研究プロセスと成果	62
3.4.3 発生した問題および今後の展望	83
3.5 研究課題5「品質間の関係分析」	84
3.5.1 当初の想定	84
3.5.2 研究プロセスと成果	85
3.5.3 発生した問題および今後の展望	100
3.6 研究課題6「全体パッケージ化」	100
3.6.1 当初の想定	100
3.6.2 研究プロセスと成果	101
3.6.3 発生した問題および今後の展望	102
4 考察	103
4.1 研究による効果や問題点等	103

4.2	産業界への展開と今後の研究の進め方.....	103
4.2.1	研究成果の産業界への展開.....	103
4.2.2	今後の研究の進め方.....	103
4.2.3	産業界および国際標準化団体への要望.....	103
	謝辞.....	104
	参考文献.....	106

研究成果概要

本調査研究のエグゼクティブサマリを以下の図1に示す。また詳細を以下に示す。



図1 エグゼクティブサマリ

(1) 研究目的とその背景

ソフトウェア製品の開発側や運用側において、開発・保守・運用中あるいは運用検討中のソフトウェア製品の品質を、客観的、定量的かつ総合的に評価可能とし、評価結果を開発・保守における品質改善や取捨選択の判断材料に役立てることを目的とする。さらに開発側では当該製品の次の改訂（バージョンアップ）においても、品質測定評価結果を品質要求定義に利用できる。

それらの達成のために、研究チームのメンバが国際的にリードするソフトウェア品質の国際規格群および品質測定法群を拡張・発展させ、日本の主要なソフトウェア製品群の品質の実態調査を通じて、ソフトウェアの内部品質、外部品質、および、顧客・利用者からの満足・評価を含む利用時の品質を定量的に測定評価し、それら間の影響関係を精密かつ総合的に明らかとする枠組みを確立した。得られる品質の実態は、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となる。

(2) 成果と期待される効果

以下をまとめ上げたソフトウェア製品の品質測定評価枠組みを成果とする。

- 製品品質（内部品質・外部品質）測定法と評価方法，評価ツール
 - 内部品質を具体的かつ網羅的に測定可能な内部品質測定法と評価方法，その一部の測定ツールと全体の評価ツール，および，その利用にあたっての制約や前提
 - 外部品質を具体的かつ網羅的に測定可能な外部品質測定法と評価方法，その一部の測定ツールと全体の評価ツール，および，その利用にあたっての制約や前提

- 利用時の品質測定法と評価方法，評価ツール
 - 利用時の品質を具体的かつ網羅的に測定可能な利用時の品質測定法と評価方法，および，その一部の測定ツールと全体の評価ツール，および，その利用にあたっての制約や前提.

以下をパッケージ化した個別および総合的な品質実態もあわせて成果とする.

- 21 の製品個々の測定評価結果，個々のコンテキスト情報，および，製品群の測定評価結果の分布や平均値等の統計量
- 21 の製品群における品質間の関係モデルおよび可視化結果

これらの成果により，日本のソフトウェア製品開発企業における製品の効率的かつ効果的な品質向上と確保に大きく貢献すると同時に，ソフトウェア製品を運用する組織や利用者における効率的かつ効果的な業務活動の推進に大きく貢献する.

(3) 研究の全体像

ソフトウェア，システム，およびそれらを利用したビジネス（業務）活動の全てについて，国際規格 ISO/IEC 25010 規定の品質モデルに基づき，国際規格 ISO/IEC 25022 および 25023 における測定法を具体化させて品質測定法をまとめる（研究課題 1～3）. その際，内部品質の測定については，研究チームがこれまでに考案し実用化しつつあるソースコードに対する測定法を応用して組み入れる（研究課題 1）.

測定結果について，個々に ISO/IEC 25051 に基づき品質を評価および傾向を分析すると同時に（研究課題 4），異なる品質間の関係を分析する（研究課題 5）. 最終的にこれらの測定法，評価分析の方法，さらには，それらの多数の実ソフトウェア製品群への適用結果をすべてまとめあげて，一般公開する（研究課題 6）.

(4) 研究実施体制

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において研究責任者および研究チームが，全体統括，ソフトウェア品質測定評価枠組みの確立，測定評価の一部，個々の品質の傾向と品質間の分析を実施する. また研究レビュー委員会を設け，品質測定評価対象のソフトウェア製品の選定，測定評価，結果の分析を円滑かつ効果的に進めるために必要な専門家からの助言を得る. さらに，コンピュータソフトウェア協会（CSAJ）が研究チームと協力して品質測定評価対象のソフトウェア製品の提供協力企業を募集および選定し，測定評価の円滑な進行を支援する. 品質測定評価の作業について部分的に，研究チームからの委託を受けて外部評価機関が実施し，結果を研究チームに提供する.

(5) 研究の方法と成果

各研究課題についてそれぞれ以下の成果を得た.

①研究課題 1: 内部品質の測定評価方法の確立

GQM 法の適用を通じて内部品質測定法の候補を識別した. 続いて，ISO/IEC 25023 や既存測定法の組み入れを考慮して内部品質測定法を定義し，測定に必要なデータを入力して測定値を得るための記入様式の策定を進めた. 最後に，測定結果の集約方法および評価基準が

らなる内部品質評価方法を定義し、ツール化した。

最終的に 66 メトリクスを具体的に定義することに成功した。それらのうちで、21 製品のデータ提供において 57 のメトリクスについて測定値が得られた。大多数のメトリクスについて測定値が得られたことから、多くのメトリクスを実効性のある形で具体的に定義できたことが分かる。

測定結果から評価を得るにあたっては、メトリクスの単位で測定値をパーセンタイルによりスコア化したうえで、スコアを品質特性・副特性単位で平均化する方式を実現した。これらの枠組みの概要を図 2 に示す。

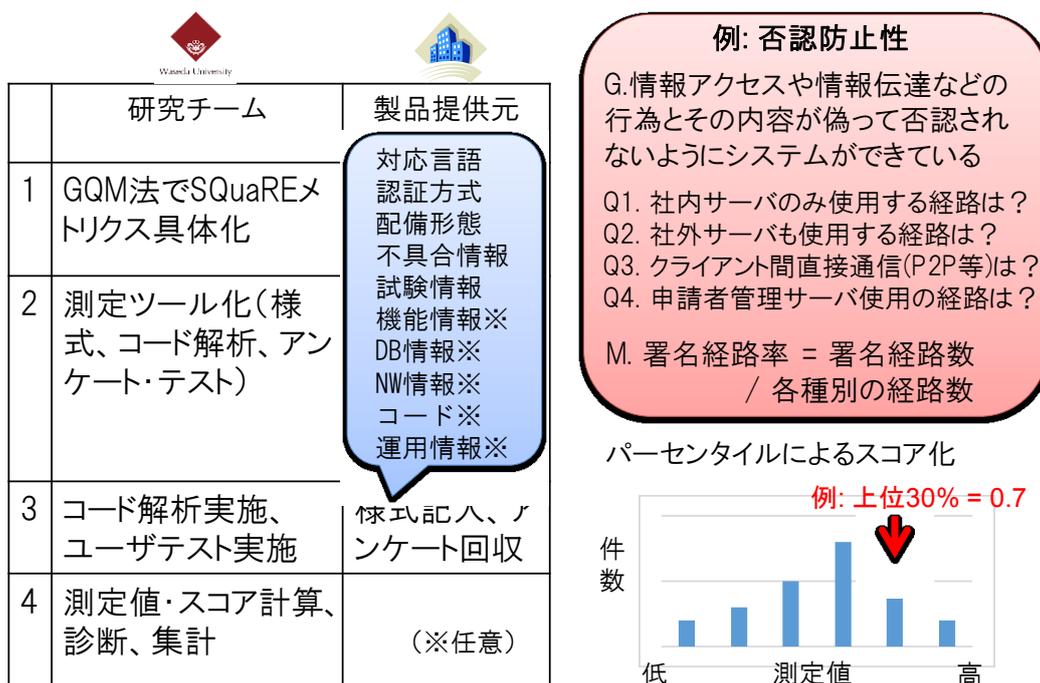


図 2 品質測定評価の枠組みの概要

②研究課題 2: 外部品質の測定評価方法の確立

GQM 法の適用を通じて外部品質測定法の候補を識別した。続いて、ISO/IEC25023 や既存測定法の組み入れを考慮して外部品質測定法を定義し、測定に必要なデータを入力して測定値を得るための記入様式の策定を進めた。最後に、測定結果の集約方法および評価基準からなる外部品質評価方法を定義し、ツール化した。

その中では特に、テストで発見される欠陥について、発見時間と数の関係を分析することで欠陥数を予測することに着目し、ソフトウェア信頼性モデルを用いてソフトウェア製品群の信頼性を評価した。分析の結果、リリースの前に予測欠陥数が安定な状態である(安定タイプ)、もしくは漸増的に増加する前兆がある(漸増タイプ)、または爆発的に増加する前兆がある(爆発タイプ)が存在することが分かった。それらの信頼性タイプを図 3 に示す。

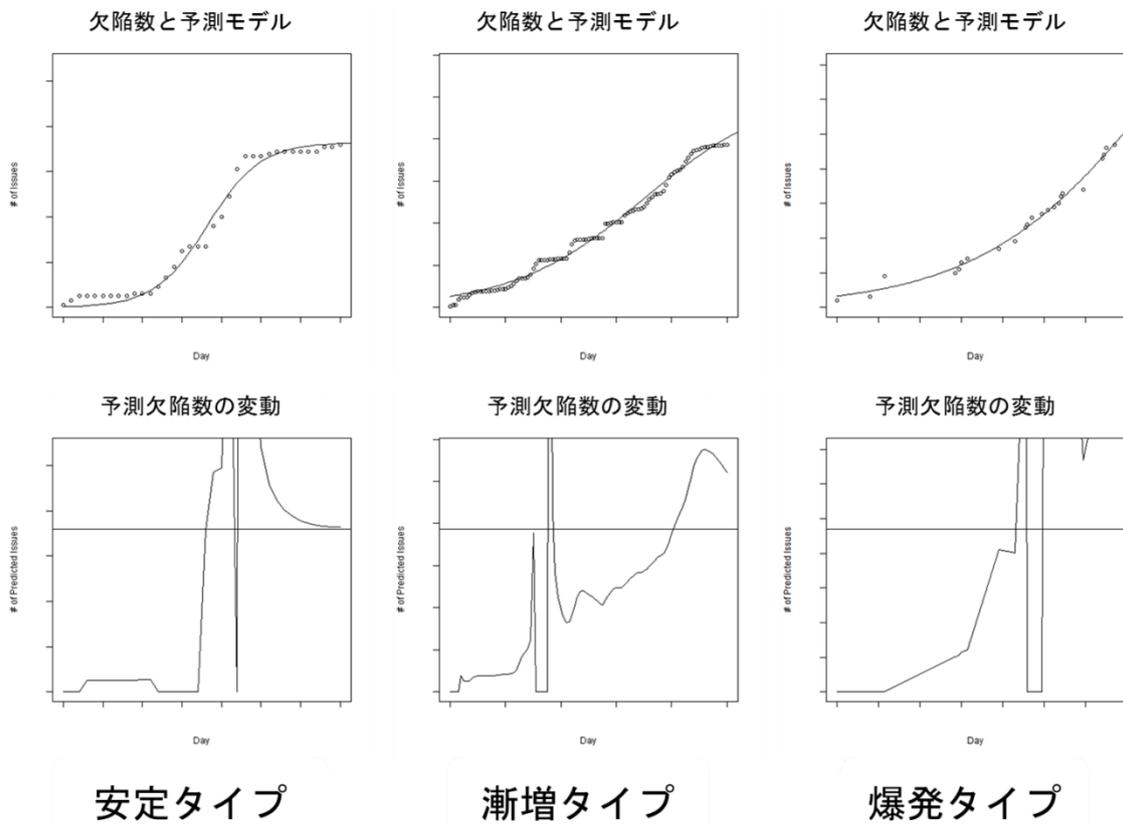


図 3 予測モデルと予測欠陥数の変動

③研究課題 3: 利用時の品質の測定評価方法の確立

利用者アンケートにおける評価項目を設計した。続いて、GQM 法の適用を通じた利用時の品質測定法の候補を識別すると共に、ISO/IEC25022 の組み入れを考慮して利用時の品質測定法を定義し、測定に必要なデータを入力して測定値を得るための記入様式およびユーザテストの設計方法の策定を進めた。最後に、測定結果の集約方法および評価基準からなる利用時の品質評価方法を定義し、ツール化した。

最終的に 17 メトリクスを具体的に定義することに成功した。それらのうちで、3.4 節において言及する 21 製品のデータ提供において全 17 メトリクスについて測定値が得られた。全てのメトリクスを実効性のある形で具体的に定義できたことが分かる。

④研究課題 4: 個々の品質実態把握

PSQ 認証制度を拡張した総合的品質測定評価制度を定義し、CSAJ の協力を経て、製品を提供する協力企業を募集し対象製品を決定した。続いて測定評価を実施し、統計処理と品質実態把握を実施した。

全 21 製品について、製品品質および利用時の利用品質の測定評価結果を得た。ただし各製品ごとに、研究チームに対して入力として提出されたデータによって測定評価できた品質特性が異なる。

本調査研究においては、品質測定評価結果を製品ごとにまとめて、品質特性単位の傾向、

実態における製品のポジション，ならびに，品質向上に向けた助言からなる品質診断レポートを作成し，製品開発元へ提供した．特に，信頼性予測，プログラムソースコード解析結果，ならびに，ユーザテストの実施結果から判明した事柄，傾向，助言について詳細に品質診断レポートに含めた．

品質特性ごとの傾向はそれぞれ大きく異なっており，特徴的な傾向を以下にまとめる．製品数のヒストグラムを図4に示す．

- 機能適合性： 分布としてなだらかに広がり，製品によって機能適合性の高さが異なることが分かる．
- 性能効率性： 分布としてなだらかに広がり，製品によって性能効率性の高さが異なることが分かる．
- 互換性： 測定評価できた製品数が限られているが，その中でスコアの高低において2極化している傾向にある．データ交換などの互換性に通じる仕組みを一部の製品において考慮していないことが原因として挙げられる．また，国際規格 SQuaRE シリーズにおける定義から具体化できたメトリクス数がわずかに二つにとどまったため，一つのメトリクスの測定値が品質特性のスコアへ多大な影響を及ぼすこととなった．この品質特性は，ISO/IEC 9126 から国際規格 SQuaRE シリーズへと改訂される際に主特性として格上げされたものであり，国際規格側において今後の実効性のあるメトリクスの拡充が望まれる．
- 使用性： 取りうる値の範囲内でスコアが低いほうに製品が集中しており，測定評価できた製品群において使用性を十分に考慮できていない，あるいは，エンドユーザ対象ではないといったことから意図的に考慮していない製品が多いことが分かる．
- 信頼性： 取りうる値の範囲が狭く，全体的に同程度のスコアを取っている．いわゆる狭義の「品質」として高い信頼性を作り込みの上で製品をリリースしていることがうかがえる．
- セキュリティ： スコアの高低について2極化している傾向にある．暗号化や破損防止などの高セキュリティ化に通じる仕組みを一部の製品において考慮していないことが原因として挙げられる．
- 保守性： 取りうる値の範囲内でスコアが低いほうに製品が集中しており，測定評価できた製品群において保守性を十分に考慮できていない，あるいは，エンドユーザ対象ではないといったことから意図的に考慮していない製品が多いことが分かる．
- 移植性： 分布としてなだらかに広がり，製品によって移植性の高さが異なることが分かる．
- 有効性： 測定評価できた製品数が限られているが，その中でスコアの高低において2極化している傾向にある．ユーザテスト実施時にタスク実行に難がありタスクを達成しにくい製品が一部見られた．
- 効率性： 測定評価できた製品数が限られているが，その中でスコアの高低において2極化している傾向にある．ユーザテスト実施時にタスク実行に難がありタスクの実施効率が低く時間を要する製品が一部見られた．
- 満足性，リスク回避性，利用状況網羅性： 測定評価できた製品数が3製品と極めて限られ，意味のある傾向を得ることができなかった．

さらに、時系列の欠陥票を得られた製品についてソフトウェア信頼性モデルに基づき信頼性タイプを特定した。

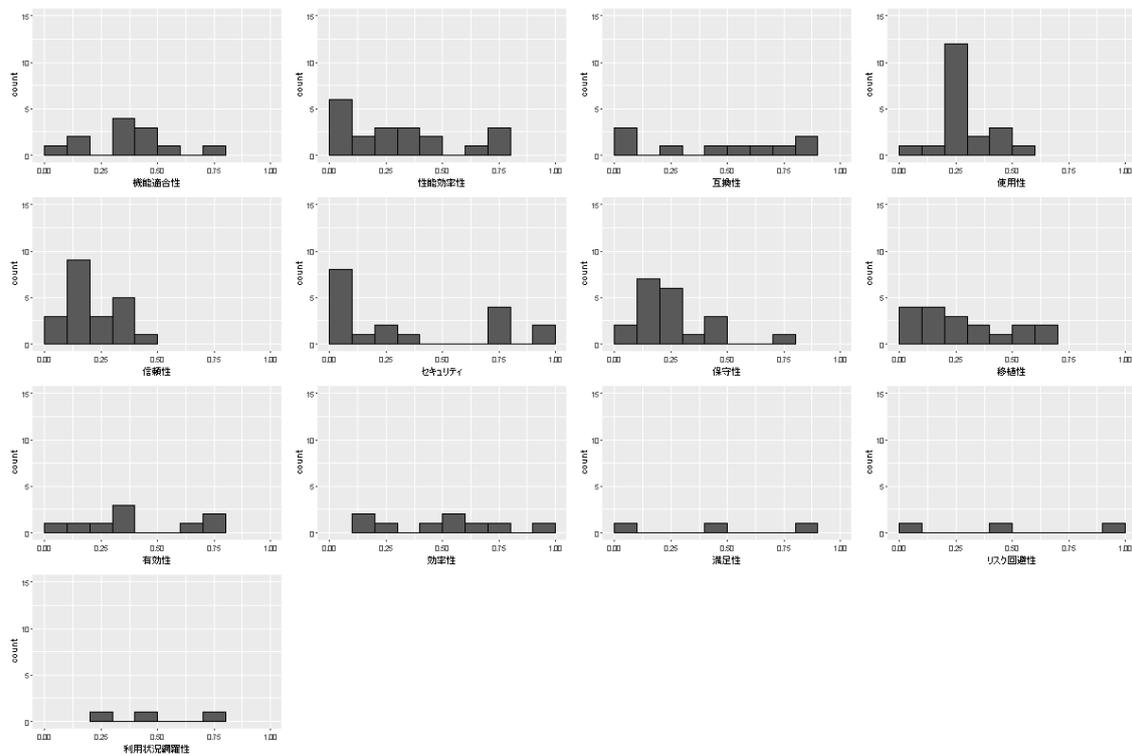


図4 品質特性ごとの傾向（製品数のヒストグラム）

⑤研究課題 5: 品質間の関係分析

21 製品の測定評価結果から、品質間の関係を分析した。さらに結果を関係モデルとしてまとめあげた。

製品品質（内部品質および外部品質）と利用時の品質の全ての品質特性間について、全 21 製品のうちで測定評価できたものについて品質評価スコアの相関を分析した。相関分析によって、得られているデータの範囲において観察しうる事柄を以下にまとめる。

- 信頼性が高いほど、性能も高い傾向にあった。信頼性と性能は、製品品質の中で主にテスト・実行によって測定評価される利用者向けの外部品質ということができ、近い関係にある品質特性と捉えることができる。ただし p 値は 0.10 以上であるため、得られた限られた結果からは統計的に有意ということとはできない。加えて、信頼性が高いほど保守性や移植性が高い傾向にあり、かつ統計的に有意と認められた。これは高信頼が求められる製品において、長期間における保守や様々な環境への移植や適合が求められるために正の相関が見られた可能性がある。
- 性能が高いほど、互換性も高い傾向にあった。この正の相関の原因としては、互換性が高いということは互換させる相手の製品群が存在するということであり、したがって

当該領域が成熟してきている中で競争優位となるために高性能化が図られているということが推測できる。ただし p 値は 0.10 以上であるため、得られた限られた結果からは統計的に有意ということはできない。これらの推測について今後の追加検証が必要である。

- 移植性が高いほど、使用性や信頼性および効率性が高いが、有効性は低い傾向にあった。移植性と信頼性の正の相関の原因としては、様々な環境に対する移植のしやすさを検討および作り込む過程において、様々な観点や側面から当該製品の品質を確認することとなり、結果として、あるいは、あわせて高い信頼性の作り込みにつながったということが推測できる。また、移植性と有効性の負の相関については、高い移植性を重視した結果として一つ一つの機能やタスク実行がしにくいものになってしまったという副作用の可能性と、逆に、有効性を重視してタスク実行をしやすいものを考慮した結果として優先度が低く設計・実装の複雑化を招きかねない移植性は強く考慮されなくなったという可能性の両方が推測できる。ただし移植性と有効性の間の相関について p 値は 0.10 以上であるため、得られた限られた結果からは統計的に有意ということはできない。これらの推測について今後の追加検証が必要である。
- 機能適合性が高いほど、使用性が低い傾向にあった。この機能適合性と使用性の負の相関について、(いわゆるアジャイル開発の考え方の対極として)当初計画通りの機能仕様の満足を最重視した結果として、顧客にとって本来重要な製品としての使いやすさを損なってしまったという副作用の可能性や、逆に使いやすさを重視した結果として一部の機能を満足できなかったという可能性、あるいは、そもそもエンドユーザ向けの製品ではないといった理由から使いやすさを軽視していたといった可能性が推測できる。ただし、これらの推測について今後の追加検証が必要である。
- セキュリティが高いほど、有効性が高い傾向にあった。これは製品によってはセキュリティ関連の機能(ログインなど)が一定割合を占め、それらをユーザテストにおいて正確に実行できたことが幾らか影響している可能性があるが、引き続きデータの拡充と追加検証が必要である。

また、信頼性タイプと品質特性の関係を分析し特定した。具体的には、信頼性モデルによる信頼性傾向と予測から得られた信頼性タイプ(安定、漸増、爆発)と、品質特性の主特性との関係について分析した。

分析の結果として、機能適合性、信頼性、有効性について安定タイプにおいて高品質であることがわかった。これは、機能適合性、信頼性、有効性が高いソフトウェア製品については、十分にテストされ欠陥を発見していると考えられる。性能効率性、互換性については爆発タイプにおいて低品質であることがわかった。これは、性能効率性、互換性が低いソフトウェア製品については、テストで十分に欠陥を発見できておらず、今後も欠陥が発見される可能性が高いと考えられる。他の特性については、顕著な違いは見られなかった。これは対象とした製品数の数が少ないことも原因と考えられる。

以上の分析結果を図 5 の関係としてまとめた。

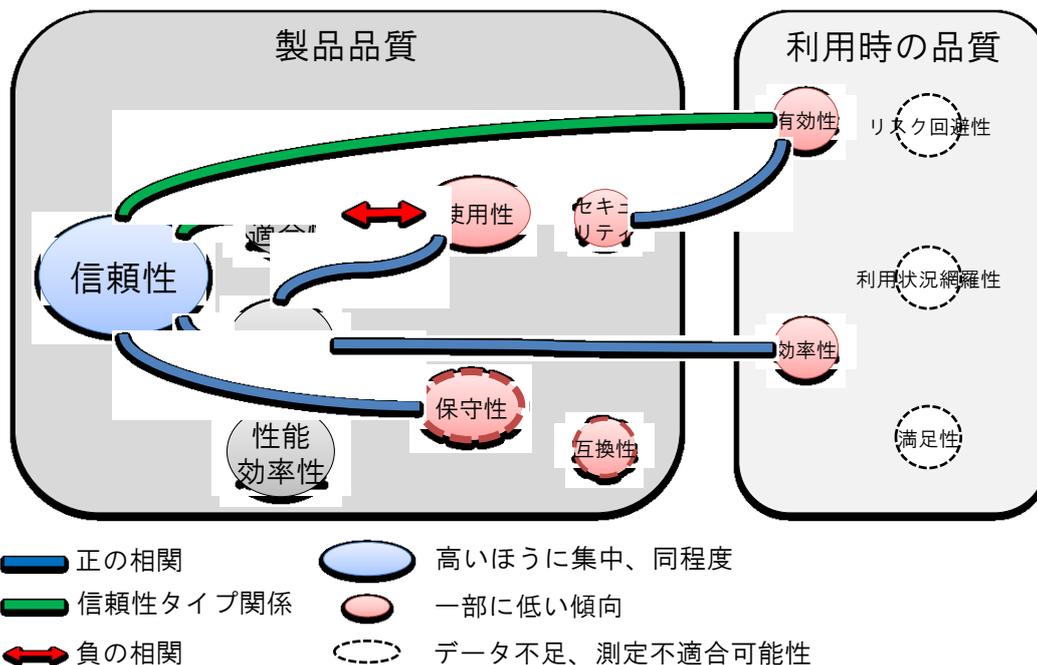


図5 品質特性間の関係のまとめ

さらに、各製品のコンテキストと品質間を分析するために、各製品のコンテキストとしてドメイン、対象開発期間、開発形態、提供種別、機能数、プログラムソースコード行数の各情報を製品提供元から得た。そのうえで、これらのコンテキストと品質評価スコアの間を分析し、得られたデータの範囲において観察しうる事柄を以下にまとめた。

- ドメイン別： 互換性およびセキュリティについてドメインによって平均スコアに顕著な差が見られた。例えばエンドユーザ向けサービス製品においてセキュリティは極めて高く、数値計算シミュレーション製品においてセキュリティは低い。これは主としてドメインごとに重視される品質特性の違いに起因するものと考えられる。しかし今後の本格的な IoT 時代において、互換性およびセキュリティはあらゆるソフトウェア製品にとって重要な品質特性となる可能性が高く、ソフトウェアの開発プロセスや方法、取り巻く環境が変革されつつある中で、ソフトウェア品質に対する意識についても、機能適合性や信頼性に代表されるいわゆる狭義の「品質」に加えて、このような IoT 時代の品質を考慮するように変革が求められてくると考えられる。もう一つの理由として、互換性およびセキュリティについて、国際規格 SQuaRE シリーズにおける定義から具体化できたメトリクス数が限られていたこともあげられる。特に互換性については、わずかに二つのメトリクスにとどまったため、一つのメトリクスの測定値が品質特性のスコアへ多大な影響を及ぼすこととなった。これらの品質特性は、ISO/IEC 9126 から国際規格 SQuaRE シリーズへと改訂される際に主特性として格上げされたものであり、国際規格側において今後の実効性のあるメトリクスの拡充が望まれる。
- 提供種別： パッケージ製品とクラウド製品において品質の全体的な傾向は概ね類似している。その中で、パッケージ製品についてはクラウド製品に比べるとセキュリティが著しく低く、ネットワーク接続や利用環境の変化を考慮する場合にセキュリティの強

化が課題ということが出来る。一方、クラウド製品についてはパッケージ製品に比べて移植性がやや低い結果となった。原因としては、クラウド製品におけるソフトウェアの移植においては DevOps に代表される運用体制と連携した機能拡張や、仮想化システムの標準への対応といったクラウド環境特有の考慮が必要となり、パッケージ製品とはやや異なる点が測定評価に用いた国際規格 SQuaRE シリーズにおいて考慮されていないという点が推測できる。また、信頼性についてもクラウド製品はパッケージ製品に比べて低いが、これは同様にメトリクスにおいてクラウド環境の考慮が不足している可能性がある。

- 規模：規模別でみた場合に品質に顕著な傾向の違いは見られなかった。
- 期間：期間別でみた場合に品質に顕著な傾向の違いは見られなかった。
- 開発形態：開発形態別でみた場合に品質に顕著な傾向の違いは見られなかった。

なお、メトリクスや品質特性によっては対象製品数や取得できたデータが少なく統計的に母数として十分な数量とは言い難いものもあることに留意が必要である。上述の分析結果はあくまでも今回取得したデータ規模における推計である。また、今回の研究はソフトウェアの全ての領域に渡っているのではなく、パッケージ等の供給側の立場からの品質評価が主になっている点にも留意が必要である。

⑥研究課題 6: 全体パッケージ化

本調査研究の成果と提言の概要を図 6 に示す。

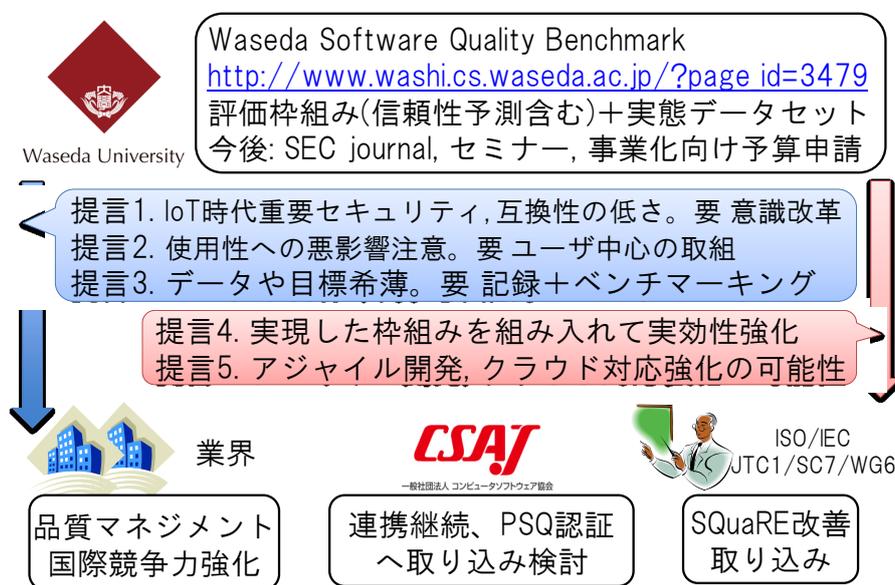


図 6 成果と提言の概要

本調査研究の成果として、国際規格 ISO/IEC 25000 SQuaRE シリーズに基づくソフトウェア製品の品質測定評価枠組み (Software Quality Measurement and Evaluation Suite) , および、同枠組みを用いて日本の 21 製品に対する測定評価結果をまとめたベンチマークと

しての公開データセット (Software Quality Dataset) を, WSQB17: Waseda Software Quality Benchmark として次の URL 上の Web ページにおいて, 図 7 に示すロゴマークと共に一般に公開した.

http://www.washi.cs.waseda.ac.jp/?page_id=3479

WSQB17

図 7 ロゴマーク

- WSQB17 Software Quality Measurement and Evaluation Suite ソフトウェア製品の品質測定評価枠組み, 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 (今後ツール等を随時改訂する予定)
 - WSQB17-MP (ZIP): ソフトウェア製品品質測定用 記入様式 Data Input Form for Product Quality
 - WSQB17-MQ (Excel): ソフトウェア利用時品質測定用 ユーザアンケート雛形 User Questionnaire Template for Quality in Use
 - WSQB17-MT (Excel): ソフトウェア利用時品質測定用 ユーザテスト雛形 User Testing Template for Quality in Use
 - WSQB17-MA (Excel, R): ソフトウェア品質評価用 集計ツール Quality Evaluation Aggregation Tool
 - WSQB17-MR (R): ソフトウェア信頼性予測ツール Software Reliability Prediction Tool (デモ動画, Demo movie)
- WSQB17 Software Quality Dataset 公開データセット 21 製品評価結果 2017 年 1 月版, 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
 - WSQB17-DC (Excel): データセット構成 21 製品コンテキストデータ Product Context Data
 - WSQB17-DQ (Excel): 製品品質・利用時品質測定評価データ Product Quality and Quality in Use Data

(6) 提言

本調査研究の成果として, 以下を産業界への提言とする.

- 提言 1. 全てがつながる IoT/IoE 時代に最重要となるはずの品質であるセキュリティおよび 互換性が一部低いパッケージ製品が見られた. ソフトウェアの開発プロセスや方法, 取り巻く環境が変革されつつある中で, ソフトウェア品質に対する意識の変革も必要である.
- 提言 2. 機能適合性の作り込みの結果として, 本来のユーザにとって重要な使用性が損なわれている可能性がある. 製品の価値を判断する立場は本来ユーザであり, 様々な品質特性を多面的に考慮するユーザ中心の取組み (例えばユーザ中心の設計など) が今後は求められる.

- 提言 3. 製品数としては 21 製品の協力を得たが、残念ながら各品質特性単位でみると実際に測定評価できた製品数は半数に満たないものがほとんどであった。この原因としては、各製品提供元において、様々な品質特性の観点から評価可能とするための根拠となるデータを記録していないこと、および、SQuaRE シリーズにおいてもともと想定している品質測定の方法において必要な目標値がほとんど設定されていないことが挙げられる。多面的な品質測定評価を可能とするためのデータの記録、ならびに、本調査研究の成果とされられたベンチマークを参照したうえでの目標値設定が重要である。

さらに、本調査研究の成果として、以下を国際標準化団体、特に国際標準化機構 ISO/IEC JTC1/SC7/WG6 への提言とする。

- 提言 4. 本調査研究によって、SQuaRE シリーズにおける品質測定方法の多くは抽象度が高く定義のままでは適用困難であることが判明した。そこで、本調査研究において実効性のある形で具体化に成功した品質測定評価の枠組みを組み入れて実効性を強化すべきである。例えば、SQuaRE シリーズの一環として、ISO/IEC テクニカルレポートとして枠組みを公開することを提言する。
- 提言 5. SQuaRE シリーズにおける品質測定方法の多くは 1990 年代までのソフトウェア製品の形態および開発方法を念頭においており、アジャイル開発やプラットフォームとしてのクラウドに対する考慮を幾らか欠いており、今後の対応強化が求められる。例えばクラウド製品におけるソフトウェアの移植においては DevOps に代表される運用体制と連携した機能拡張や、仮想化システムの標準への対応といったクラウド環境特有の考慮が必要である。

1 研究の目的・背景と期待される効果

1.1 研究目的とその背景

ソフトウェア製品の開発側や運用側において、開発・保守・運用中あるいは運用検討中のソフトウェア製品の品質を、客観的、定量的かつ総合的に評価可能とし、評価結果を開発・保守における品質改善や取捨選択の判断材料に役立てることを目的とする。さらに開発側では当該製品の次の改訂（バージョンアップ）においても、品質測定評価結果を品質要求定義に利用できる。

それらの達成のために、研究チームのメンバが国際的にリードするソフトウェア品質の国際規格群および品質測定法群を拡張・発展させ、日本の主要なソフトウェア製品群の品質の実態調査を通じて、ソフトウェアの内部品質、外部品質、および、顧客・利用者からの満足・評価を含む利用時の品質を定量的に測定評価し、それら間の影響関係を精密かつ総合的に明らかとする枠組みを確立した。得られる品質の実態は、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となる。取り扱う範囲を図 1-1 に示す。

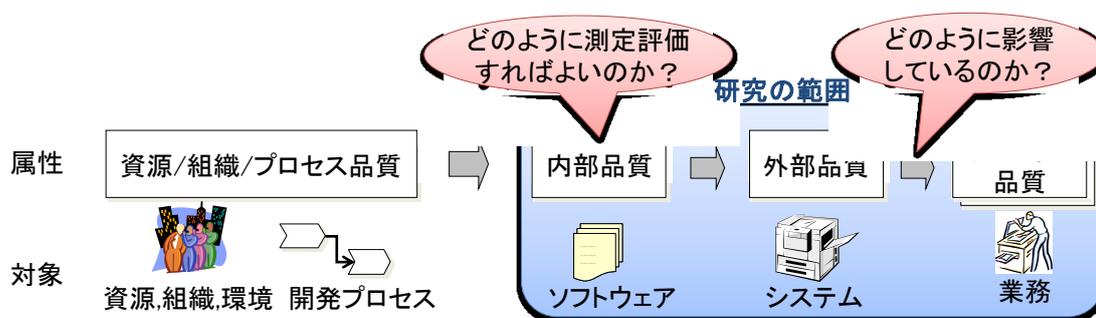


図 1-1 研究の範囲と扱う事柄

1.2 期待される効果

以下をパッケージ化した総合的な品質測定評価枠組みを成果とする。

- 内部品質を具体的かつ網羅的に測定可能な内部品質測定法と評価方法、その一部の測定ツールと全体の評価ツール、および、その利用にあたっての制約や前提
- 外部品質を具体的かつ網羅的に測定可能な外部品質測定法と評価方法、その一部の測定ツールと全体の評価ツール、および、その利用にあたっての制約や前提
- 利用時の品質を具体的かつ網羅的に測定可能な利用時の品質測定法と評価方法、および、その一部の測定ツールと全体の評価ツール、および、その利用にあたっての制約や前提

以下をパッケージ化した個別および総合的な品質実態もあわせて成果とする。

- 21 のソフトウェア製品個々の測定評価結果、個々のコンテキスト情報、および、製品群の測定評価結果の分布や平均値等の統計量
- 21 のソフトウェア製品群における品質間関係モデルおよび可視化結果

ソフトウェア製品の開発側や運用側において、公開する総合的な品質測定評価枠組みを適用あるいは適用結果を参照することで、開発・保守・運用中あるいは運用検討中のソフト

ウェア製品の品質を、客観的、定量的かつ総合的に評価可能とし、評価結果を開発・保守における品質改善や取捨選択の判断材料に役立てることができる。特に開発側においては、品質上問題のある個所や側面を特定し、重点的な改善を施すことを支援する。また当該製品の改訂（バージョンアップ）時に、品質測定評価結果を品質要求の定義に利用することができる。

さらに、公開する品質実態は、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となる。これをソフトウェア製品の開発側や運用側において、開発・保守・運用中あるいは運用検討中のソフトウェア製品の品質と照らし合わせることで、当該品質の日本の業界における位置づけを容易に把握可能とし、品質改善や取捨選択の判断材料に役立てることができる。

結果として、日本のソフトウェア製品開発企業における製品の効率的かつ効果的な品質向上と確保に大きく貢献すると同時に、ソフトウェア製品を運用する組織や利用者における効率的かつ効果的な業務活動の推進に大きく貢献する。

2 実施内容

2.1 研究アプローチ

2.1.1 研究の全体像

研究のアプローチ方法および研究課題の全体像を図 2-1 に示す（参考：品質ライフサイクルモデル [JIS25010]）。ソフトウェア、システム、およびそれらを利用したビジネス（業務）活動の全てについて、国際規格 ISO/IEC 25010 規定の品質モデルに基づき、国際規格 ISO/IEC 25022 および 25023 における測定法を具体化させて品質測定法をまとめる。その際、内部品質の測定については、研究チームがこれまでに考案し実用化しつつあるソースコードに対する測定法を応用して組み入れる。

測定結果について、個々に ISO/IEC 25051 に基づき品質を評価および傾向を分析すると同時に、異なる品質間を関係分析する。最終的にこれらの測定法、評価分析の方法、さらには、それらの多数の実ソフトウェア製品群への適用結果をすべてまとめあげて、一般公開する。

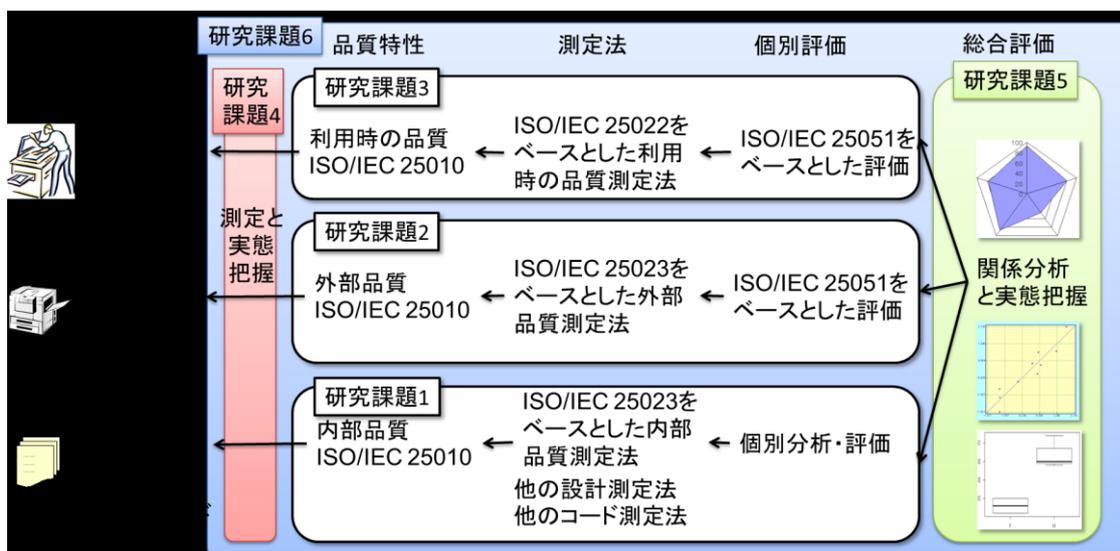


図 2-1 研究のアプローチ方法および研究課題の全体像（参考：品質ライフサイクルモデル [JIS25010]）

2.1.2 関連するこれまでの研究について

ソフトウェアシステムが重要な社会インフラとなった今日において、ソフトウェアおよびそれにより制御されるシステムの品質が、システムの運用を通じて実現されるビジネス（業務活動）の品質を左右し、それは結果として組織および社会の品質を決定づけると考えられる。実際にそのような関係の一部は、ソフトウェア品質に起因する事故のような形で現実に顕在化されつつある。従ってソフトウェア品質の向上は現代社会の最重要課題の一つである [東 15]。

しかしながらその精密な関係は依然として不明瞭であり、従って開発側や運用側のそれぞれの立場において、ソフトウェア製品の品質を客観的かつ定量的に把握および必要な方

策（例えば問題箇所の品質改善）を実施することが現状では困難である。

ソフトウェア開発の上流工程で検証されるソフトウェアの静的な内部品質が、結合以降の下流工程で検証されるソフトウェアシステムの動的な外部品質に影響し、外部品質がソフトウェアシステムの運用時に評価される利用時の品質に影響を与えることが謳われている[ISO25000]。その前提のもとで、開発の上流工程では設計や実装において内部品質を作り込み、テストにおいて高い外部品質を確認および作り込もうとする。しかし、それらの間の影響の関係は、ソフトウェアの対象および調べる側面が限定された範囲でのみ明らかであり、結果としてソフトウェアおよびシステムの総合的および様々な立場における品質評価に向けて品質測定法や測定結果は十分に活用されていない[鷺崎 07]。

例えば、開発の上流工程において REBOOT[Sindre95]や QMOOD[Bansiya02]、Ortega らのスイート [Ortega03]、さらには研究責任者らのメトリクススイート [鷺崎 07][鷺崎 10][Washizaki12][Sato13][Tsuda13][Nakai14]のように、設計モデルやソースコードを測定して内部品質を評価する枠組みは存在する。しかしながら、評価結果から、ソフトウェアシステムとしての外部品質、さらには最終的な顧客満足度を含む利用時の品質を網羅的に予測する取り組みは、研究責任者の知る限り実現されていない。

また国際規格としては、本調査研究における研究レビュー委員会のメンバがリードする形で、内部品質、外部品質、利用時の品質のそれぞれについて主要な測定量と測定方法を定義しているが [ISO25023] [ISO25022]、国際規格の性格上、それらの定義は汎用性を重視して抽象的なものにとどまり、具体的なソフトウェアやシステムのテスト実行結果、システムの運用における具体的な測定方法と解釈の方法はあきらかとなっていない。それらの測定結果の間関係も明らかとなっていない。

研究責任者および研究チームは、設計モデルやプログラムソースコードの測定を通じたソフトウェアの内部品質評価について長年にわたり国内外で評価される研究実績を有し、様々な企業における導入実績を有し[鷺崎 07][鷺崎 10][Washizaki12][Tsuda13][Nakai14]、書籍や講演（例えば[鷺崎 15]）を通じたソフトウェアの品質測定評価に関する指導的立場にある。さらに内部品質の評価結果を、外部品質の一部（システムテスト時の欠陥数など）と照らして関係を分析することに成功しつつある[Sato13]。

さらに本調査研究における研究レビュー委員会のメンバは、内部品質、外部品質、利用時の品質の品質モデルや品質測定法等の国際規格 ISO/IEC 25000 (SQuaRE) シリーズ策定の全体統括リーダを務めている[東 15][東 14][込山 14]。

2.1.3 研究目標と研究課題

(1) 研究目標

研究責任者、研究チームおよび研究レビュー委員会が国際的にリードするソフトウェア品質の国際規格群および品質測定法群を拡張・発展させ、日本のそれぞれ開発組織の異なる主要な 21 のソフトウェア製品群の開発実績および利用実績の実態調査を通じて、ソフトウェアの内部品質、外部品質、および、顧客・利用者の満足度・評価を含む利用時の品質を定量的に測定評価し、それらの間の関係を明らかとする枠組みを確立する。枠組みの確立の過程であわせて、日本のソフトウェア製品の品質の実態として、測定評価対象として用いる 21 の主要なソフトウェア製品群の測定評価結果の分布や平均値等の統計量をまとめる。

産業界における課題は、ソフトウェアの内部品質、外部品質および利用時の品質の関係が不明瞭であり、従って開発側や運用側のそれぞれの立場において、ソフトウェア製品の品質を客観的かつ定量的に把握および必要な方策を実施困難なことである。結果として、当該製品の改訂時において品質要求を適切に定義することも難しい。特定組織に限定せずに、様々な代表的な組織が開発した製品群の大規模な調査に基づくデータに裏付けされた事実は得られていない。

これに対して本研究目標は、特定の品質ではなく内部品質、外部品質、利用時の品質の全てを総合的に測定評価可能な枠組みおよび関係を分析し特定すること、および、それを産業界における様々な代表的開発組織における実際のソフトウェア製品群を対象として大規模に調査を実施することで特定企業に依らない産業界の実態を実データに裏付けされた形で明らかとすることにより課題を解決する。得られる枠組みを用いることでソフトウェア製品の開発側および利用側において総合的に品質を評価可能となり、評価結果を製品の品質改善や取捨選択、さらには改訂時の品質要求定義に役立てられる。

ソースコードを測定して内部品質を評価する枠組みは国内外で様々な存在するが（例えば[Sindre95][Bansiya02][Ortega03]）、その評価結果から、外部品質および利用時の品質を網羅的に予測する取り組みは実現されていない。また国際規格としては、研究チームのメンバがリードする形で、内部品質、外部品質、利用時の品質のそれぞれについて主要な測定量と測定方法を定義しているが [ISO25023][ISO25022]、具体的な測定方法と解釈の方法はあきらかではなく、それらの測定結果の間関係も明らかとなっていない。

一方で、第三者が利用可能な形で提供されるパッケージソフトウェアやクラウドアプリケーション等のソフトウェア製品について品質要求を定めて測定評価する枠組みとして国際規格 ISO/IEC 25051[ISO25051]が研究チームのメンバがリードする形で策定され、同国際規格の旧版および IPA のソフトウェア品質説明のための制度ガイドライン[IPA14]に基づき、一般社団法人コンピュータソフトウェア協会 (CSAJ) により ISO/IEC 25051 に準拠した PSQ 認証制度[PSQ]が 2014 年から開始されている。ただし PSQ 認証制度は、主としてソフトウェアの外部品質および利用時の品質のうちで特に利用者における運用に強く影響すると考えられる一部の側面を重点的に評価するものであり、内部品質の測定評価、ならびに、外部品質や利用時の品質の網羅的な評価を実施するものではない。また、それらの間関係の分析はなされていない。

本研究は、ソフトウェアの内部品質、外部品質、利用時の品質の全てを網羅的に測定評価する点、および、日本における開発元の異なる多数の実ソフトウェア製品群への適用を通じて品質間関係を総合的に明らかとする点で、独創的である。

根拠としては、内部品質、外部品質、利用時の品質のそれぞれに対する測定評価の枠組みや試みは多数存在するものの、それらの間関係については ISO/IEC 25000 において「影響の関係にある」[ISO25000]と謳われるにとどまり、詳細は明らかとされていない。開発組織を超えたソフトウェア開発の実態調査としては IPA データ白書[IPA15]や Cusumano らによる調査[Cusumano03]、Jones による調査[Jones09]などがあるが、いずれも外部品質と利用時の品質の一部およびそれらの間関係の一部の分析に留まり、内部品質の実態およびそれらの間関係の詳細に踏み込んでいない。

(2) 研究目標に向けた研究課題の設定

上述の研究目標の達成のためには、内部品質、外部品質、利用時の品質のそれぞれについて品質測定評価の方法を国際規格に基づき確立すること、ならびに、得られる測定評価結果に基づき個々の品質実態を把握するとともに関係を分析すること、さらには、それらの全ての成果をまとめあげることが必要である。

以上の理由に基づき、以下の6つの研究課題を設定した。

- 研究課題 1: 内部品質の測定評価方法の確立
内部品質の具体的かつ網羅的な測定法および評価方法を確立する。内部品質測定法と評価方法、および、一部の測定ツールと全体の評価ツールをアウトプットとする。
- 研究課題 2: 外部品質の測定評価方法の確立
外部品質の具体的かつ網羅的な測定法および評価方法を確立する。外部品質測定法と評価方法、および、一部の測定ツールと全体の評価ツールをアウトプットとする。
- 研究課題 3: 利用時の品質の測定評価方法の確立
利用時の品質の具体的かつ網羅的な測定法および評価方法を確立する。利用時の品質測定法と評価方法、および、一部の測定ツールと全体の評価ツールをアウトプットとする。
- 研究課題 4: 個々の品質実態把握
21 の製品群を測定評価し、個々の品質単位の実態としてまとめる。個々の測定評価結果および製品群の測定評価結果の分布や平均値等の統計量をアウトプットとする。
- 研究課題 5: 品質間の関係分析
21 の製品群における品質間の関係を分析し、日本の代表的製品群の品質の総合的な実態としてまとめる。品質間の関係モデルおよび可視化結果をアウトプットとする。
- 研究課題 6: 全体パッケージ化
品質測定評価枠組みと品質実態をパッケージ化しアウトプットとして公開する。

2.2 研究の活動実績・経緯

経緯として、当初に想定し1年目(2015年)に設計実現した品質測定評価の方法においてはソフトウェア製品提供元(ソフトウェアベンダ)の負担が大きく、2年間で当初計画していた40製品を品質測定評価することが困難な見通しとなった。そこで2年目(2016年)において、もともとの想定における製品提供元の作業を外部評価機関に委託することで、申請者側の負担を軽減し、多くの製品を品質測定評価対象とすることに成功した。なお、外部評価機関への委託内容が増大した関係で製品あたりの評価作業外注費が増大し、委託研究の全体予算のうちで外注費は50%未満に抑えるというという制約の都合上、最終的な品質測定評価対象数は21製品となった。品質の実態を明らかにする上で、一定の特徴を把握するにあたっては十分な製品数を確保することができた。

(1) 実施スケジュールの実績

実施スケジュールの実績を表2-1および表2-2に示す。各研究課題を具体的に遂行するにあたって必要な検討や計画を深めるために、研究チームにおける内部打合せを2015年6月から2017年2月まで週1回程度の頻度で実施した。また、品質測定評価対象のソフトウェア製品の選定、測定評価、結果の分析を円滑かつ効果的に進めるにあたり必要な専門家か

Framework for a Software Quality Evaluation based on ISO/IEC 25022 and ISO/IEC 25023,” Poster, The 2016 IEEE International Conference on Software Quality, Reliability & Security (QRS 2016)

3. K. Honda, N. Nakamura, H. Washizaki and Y. Fukazawa, “An Industrial Case Study of Project Management Using Cross Project Software Reliability Growth Model,” Poster, The 2016 IEEE International Conference on Software Quality, Reliability & Security (QRS 2016)
4. K. Honda, N. Nakamura, H. Washizaki and Y. Fukazawa, “Case Study: Project Management Using Cross Project Software Reliability Growth Model,” IEEE International Workshop on Trustworthy Computing, 2016
5. K. Honda, N. Nakamura, H. Washizaki and Y. Fukazawa, “Case Study: Project Management Using Cross Project Software Reliability Growth Model Considering System Scale,” 27th IEEE International Symposium on Software Reliability Engineering (ISSRE 2016), Industry Track,
6. H. Nakai, N. Tsuda, K. Honda, H. Washizaki, Y. Fukazawa, “Evaluating Software Product Quality based on SquaRE Series,” IEEE TENCON 2016

(4) イベント、講演

実施期間中に研究内容を説明したイベントや講演を以下に示す。

1. 2015年9月4日 第2回早稲田大学・Fraunhofer IESE 共催セミナー
2. 2016年2月29日早稲田大学グリーン・コンピューティング・システム研究機構シンポジウム
3. 2016年3月29日 ～JISX25051:2016 リリース記念セミナー～ 世界最高基準を目指す日本のソフトウェア品質 早稲田大学
4. 2016年6月22日 ISO/IEC 25022/25023 リリース記念セミナー CSAJ

(5) メディア・報道ほか

実施期間中に研究内容についてメディア等に取り上げられた内容を以下に示す。

1. マイナビニュース, 早大, ソフトウェアのベンチマークとなる品質実態調査と評価枠組みの研究, 2015年6月23日
2. Yahoo! Japan ニュース, 早大, ソフトウェアのベンチマークとなる品質実態調査と評価枠組みの研究, 2015年6月23日
3. Biglobe ニュース, 早大, ソフトウェアのベンチマークとなる品質実態調査と評価枠組みの研究, 2015年6月23日
4. ライブドアニュース, 早稲田大学がソフトウェア製品において世界初のベンチマークとなる品質実態調査と評価枠組みの研究に着手, 2015年6月23日
5. 財経新聞, 早稲田大学
6. がソフトウェア製品において世界初のベンチマークとなる品質実態調査と評価枠組みの研究に着手, 2015年6月22日
7. 産経ニュース, 早稲田大学がソフトウェア製品において世界初のベンチマークとなる

品質実態調査と評価枠組みの研究に着手，2015年6月22日

8. PR TIMES，早稲田大学がソフトウェア製品において世界初のベンチマークとなる品質実態調査と評価枠組みの研究に着手，2015年6月22日
9. 早稲田大学，ソフトウェア製品において世界初のベンチマークとなる品質実態調査と評価枠組みの研究に着手，2015年6月22日
10. PSQ 認証ポータル，早稲田大学がソフトウェア製品において世界初のベンチマークとなる品質実態調査と評価枠組みの研究に着手，2015年6月22日
11. コンピュータソフトウェア協会，早稲田大学がソフトウェア製品において世界初のベンチマークとなる品質実態調査と評価枠組みの研究に着手，2015年6月22日
12. 日本経済新聞 全国・朝刊，早大グローバルソフトウェアエンジニアリング研 未来の開発者育成探る，《ビジョン》驚崎弘宜所長 データ+経験則 社会を変えたい，2016年11月2日

2.3 研究実施体制

実施体制の全体を図 2-1 に示す。各利害関係者の役割を以下および表 2-3 に示す。

- グローバルソフトウェアエンジニアリング研究所：研究責任者および研究チームが，全体統括，ソフトウェア品質測定評価枠組みの確立，測定評価の一部，個々の品質の傾向と品質間の分析を実施する。
- 研究レビュー委員会：品質測定評価対象のソフトウェア製品の選定，測定評価，結果の分析を円滑かつ効果的に進めるために必要な専門家からの助言を得る。専門家として，ISO/IEC JTC1/SC7/WG6 のプロジェクトエディタならびにコンビーナとして当該領域を国際的にリードする東基衛氏，込山俊博氏の参画を得た。さらに下記 CSAJ より中野正氏，鈴木啓紹氏の参画を得た。
- コンピュータソフトウェア協会（CSAJ）：研究チームと協力して品質測定評価対象のソフトウェア製品の提供協力企業を募集および選定し測定評価の円滑な進行を支援する。
- 評価機関：研究チームからの委託を受けて品質測定評価の大部分を実施，結果を研究チームに提供する。（外注）
- 学生アルバイト：ソフトウェア製品の測定評価作業の一部を実施する。

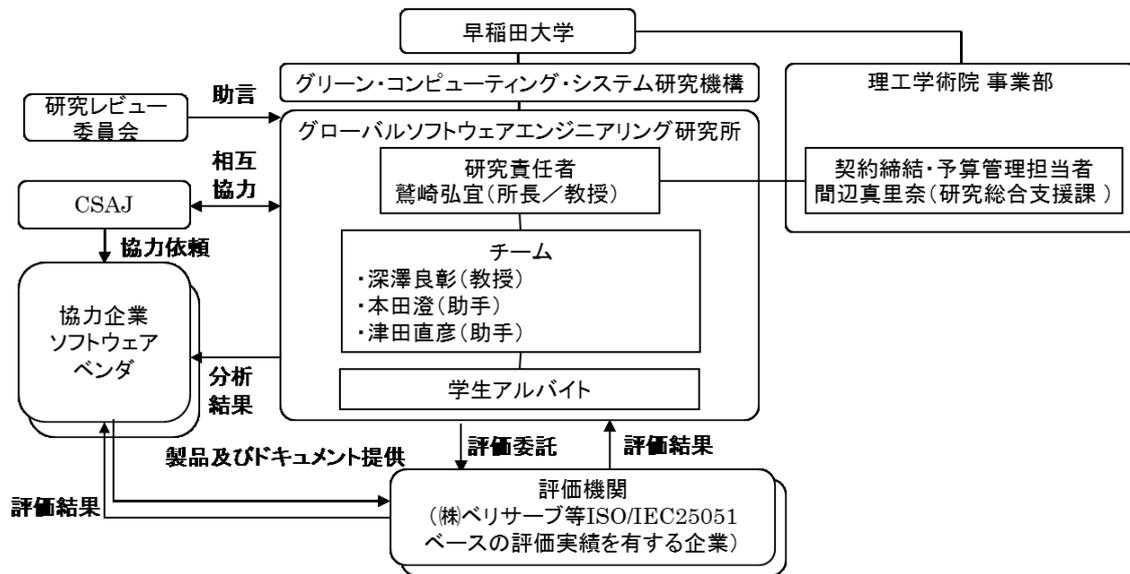


図 2-1 研究実施体制図

表 2-3 研究実施体制（役割分担表）

実施機関名	学校法人早稲田大学 グリーン・コンピューティング・システム研究機構 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	教授 鷲崎 弘宜		
連絡担当者	事務担当 篠原 直子		
主な実施場所 および研究者	早稲田大学西早稲田キャンパス 〒169-8555 住所 東京都新宿区大久保 3-4-1 早稲田大学 63 号館 0503 室 連絡先 TEL 03-5286-3272 FAX 03-5286-3272 e-mail rise2015@list.waseda.jp		
	氏名	職種	担当内容
	鷲崎 弘宜	所長・教授	研究全体統括，測定評価枠組み定義，測定評価結果の分析とまとめ
	深澤 良彰	教授	測定評価枠組み定義，評価結果分析
	本田 澄	助手	測定評価枠組み定義，ツール実装，評価結果分析
津田 直彦	助手	測定評価枠組み定義，ツール実装，評価結果分析	

また研究責任者のプロフィールを表 2-4 に示す。

表 2-4 研究責任者のプロフィール

(ふりがな) 氏名	washizaki ひろのり 鷺崎 弘宜	
生年月日	1976 年 11 月 19 日	
所属機関	学校法人早稲田大学	
所属 (部署名)	グリーン・コンピューティング・システム研究機構 グローバルソフトウェアエンジニアリング研究所 理工学術院 基幹理工学部 情報理工学科	
役職	所長・教授	
住所	〒169-8555 東京都新宿区大久保 3-4-1 早稲田大学 63 号館 0503 室	
TEL	03-5286-3272	
E-mail	washizaki@waseda.jp	
【学歴(大学卒業以降)】	1999 年 3 月 早稲田大学理工学部卒業 2001 年 3 月 早稲田大学大学院理工学研究科修士前期課程 修了 2003 年 3 月 早稲田大学大学院理工学研究科博士後期課程 修了、博士(情報科学)	【職歴】 2002 年-04 年 早稲田大学助手 2004 年-08 年 国立情報学研究所助手・助教 2008 年-16 年 早稲田大学准教授、国立情報学研究所客員准教授 2010 年-現在 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 2016 年-現在 早稲田大学教授、国立情報学研究所客員教授
【研究実績】	2002 年-04 年 効率的なコンポーネント試行、検索、テスト機構を実現。 2004 年-06 年 品質評価と再利用の研究に従事。資産からの部品抽出について JSSST 高橋奨励賞。ソースコード品質評価について SES 優秀論文賞、IPSJ 山下記念研究賞。 2008 年-現在 品質評価・品質保証と再利用の研究に従事。テスト支援環境について船井ベストペーパー賞、テスト技術者振興協会善吾賞。ソースコード品質評価について船井情報科学奨励賞。品質に関する知識体系 SQuBOK の整理と執筆により日経品質管理文献賞。バグ修正過程調査について IWESEP ベストポスター賞。	
【主な論文・著書】	R. Tsuchiya, H. Washizaki, et al., "Recovering Traceability Links between Requirements and Source Code using the Configuration Management Log," IEICE Trans., 98D, 2015. 鷺崎弘宜, "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, 2015. T. Kobashi, N. Yoshioka, H. Kaiya, H. Washizaki, et al., "Validating Security Design Pattern Applications by Testing Design Models," Int' l. J. Secure Software Engineering, 5(4), 2014. 鷺崎弘宜, "ソフトウェア品質の知識体系", 情報処理, 55(1), 2014.	

3 研究成果

3.1 研究課題 1「内部品質の測定評価方法の確立」

3.1.1 当初の想定

(1) 研究内容

ソフトウェアの機能説明書，UML で記述された設計モデル（得られる場合），および，代表的なプログラミング言語（C，C++，Java を想定）で記述されたプログラムソースコードを対象として，ISO/IEC 25010 で規定される内部品質の品質特性について，研究責任者および研究チームがこれまでに考案し実用化しつつある設計モデルおよびソースコードに対する測定法[鷺崎 07][鷺崎 10][Washizaki12][Sato13][Tsuda13][Nakai14]を組み入れた形で，ISO/IEC 25023 で規定される内部品質の測定法群を具体化する．さらに研究課題 4 の実現を通して測定結果の集約方法および評価基準を設定し，評価方法としてまとめあげる．測定法の一部については自動測定ツールを実装し，評価方法は全体としてツール化する．

(2) 想定問題と対応策

問題： 国際標準としての ISO/IEC 25023 における内部品質測定法は幾らか抽象的に定義されており，具体的な機能説明書や設計モデル，ソースコード上の特徴を捉えるにあたり具体化が必要である．ただし具体化にあたり，測定可能な事柄や測定法ありきで検討すると，品質を明らかとしようとする本来の目的を見失う可能性がある．

解決策： ISO/IEC 25010 で規定される内部品質の個々の品質特性を明らかとすることをゴールとし，その達成に有用なデータを得るための測定法をトップダウンに導出する．導出にあたりゴール指向の枠組みである Goal-Question-Metric (GQM) 法[Basili94][Nakai14]を応用適用し，必要な測定法の候補を識別する．識別した候補に対して，ISO/IEC 25023 の測定法，および，研究責任者・研究チームが実現済みのソースコードに対する測定法を対応付け可能であれば調整のうえ組み入れて，対応付けできない場合は新規の測定法として具体的に定義する．研究課題 4 の実現を通して必要に応じて測定法を修正し，一定規模の測定結果を収集，その分布および人手による意味解釈付けを通じて測定結果の集約方法および評価基準を設定し，評価方法としてまとめあげる．特定のソフトウェア製品やドメインに依らず汎用性が高いと想定される測定法の一部については自動測定ツールを実装する．

3.1.2 研究プロセスと成果

(1) 研究プロセス

- ① GQM 法の適用を通じて内部品質測定法の候補を識別する．
- ② ISO/IEC 25023 や既存測定法の組み入れを考慮して内部品質測定法を定義し，測定に必要なデータを入力して測定値を得るための記入様式 (RISE 様式と呼ぶ) を策定する．
- ③ 測定結果の集約方法および評価基準からなる内部品質評価方法を定義し，ツール化する．

これらのプロセスを図 3-1-1 に示す．当該プロセスは他の外部品質，利用時の品質においても共通である．



図 3-1-1 測定評価のプロセスと役割分担

(2) 具体的な研究成果の内容

① 内部品質測定法（研究プロセス①②③）

国際規格 ISO/IEC 25023 においては、内部品質と外部品質を測定するためのメトリクスが定義されており、両者は製品品質メトリクスとしてまとめられている。そこで、最初に ISO/IEC 25023 を参照し表 3-1-1 に示すすべてのメトリクスを初期候補として GQM 法を適用し、品質副特性を網羅するように合計 66 メトリクスを定義した（研究プロセス①）。適用にあたり、ISO/IEC 25023 の解釈や、入力として必要な測定データの現実的な利用可能性等について研究レビュー委員会からの助言を得た。続いて、測定に必要なデータを入力して測定値を得るための記入様式を策定した（研究プロセス②）。

GQM 法の適用の様子を例を図 3-1-2 に示す。図 3-1-2 においては、セキュリティの品質副特性の一つである否認防止性について、通信経路のデジタル署名に着目し、ゴール (Goal) を掲げたうえで、その目標達成可否を判断するための種々の側面を網羅するように質問 (Question) を多く設定している。最終的には、測定結果の分かりやすさを重視して、個々の質問について異なる測定法 (Metric) を定義するのではなく、種々の側面を全て総合する形で一つの測定法へと集約しまとめあげている。

例: 否認防止性

G.情報アクセスや情報伝達などの行為とその内容が偽って否認されないようにシステムができています

- Q1. 社内サーバのみ使用する経路は？
- Q2. 社外サーバも使用する経路は？
- Q3. クライアント間直接通信(P2P等)は？
- Q4. 申請者管理サーバ使用の経路は？

M. 署名経路率 = 署名経路数 / 各種別の経路数

図 3-1-2 GQM 法の適用の例 (測定法についてまとめ上げている)

具体的には 66 メトリクスを具体的に定義した。それらのうちで、3.4 節において言及する 21 製品のデータ提供において 57 のメトリクスについて測定値が得られた。品質特性単位のメトリクス定義数を図 3-1-3 に示す。大多数のメトリクスについて測定値が得られたことから、多くのメトリクスを実効性のある形で具体的に定義できたことが分かる。

また測定率は 66 メトリクス×21 製品のうちで 34%であった。未測定の原因の多くは根拠となるデータの未記録のためであり、これについては、より大規模な品質測定評価に向けて各製品開発におけるデータの記録が課題である。さらにデータ収集に必要な記入様式に説明が不十分な点があったため、製品提供側が記入した幾つかのデータの解釈にぶれが生じ、比較するデータとしては適さず未測定とせざるを得なかったことも影響している。記載内容についての解釈が容易となるよう、補足説明等を加えることを検討する。

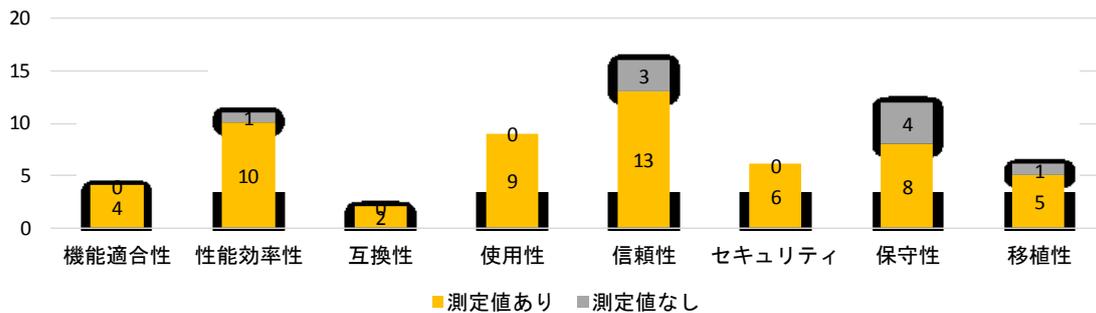


図 3-1-3 製品品質 (内部品質・外部品質) メトリクスの定義数

これらのメトリクスを半自動的に測定するための記入様式や解析ツールについて、WSQB17 に含めて http://www.washi.cs.waseda.ac.jp/?page_id=3479 にて公開している (研究プロセス③)。具体的には、RISE 様式への必要データの記入により、各メトリクスや品質副特性、品質特性単位における評価を得るための中間データを得る。中間データを集計用のスプレッドシートに転記することで、最終的な評価結果を得る。

今回の調査研究において具体化した製品品質メトリクスについては表 3-1-1 にまとめた。品質主特性(Goal 相当)、副特性(SubGoal 相当)、測定対象とする製品の側面(Question 相

当), ISO/IEC 25023 の元々のメトリクス ID (Metric), 本研究で具体化したメトリクス, 具体化したメトリクスの具体化経緯等を一つの表に整理している.

また, 赤色のメトリクス行は今回具体化せずに見送ったものと, 具体化したデータが集まらずに測定できなかったものを意味する. また, 黄色のメトリクス行は元々の ISO/IEC 25023 には無い追加したものを意味する. ソフトウェア製品の開発・提供元からのデータ提供が困難と予想されるメトリクスのみ定義されていた品質副特性について, より簡易なデータであっても測定可能なメトリクスを追加することで実効性を高めた.

表 3-1-1 製品品質 (内部品質・外部品質) メトリクスの GQM と具体化方針のまとめ

主特性(Goal) 副特性 (SubGoal) ISO メトリクス ID	測定種別 RISE メトリクス (Metrics) メトリクス見出し	具体化前後の GQM の Question (測定の側面) ※複数の Question がひとつ の Metric にリンクする	RISE メトリクスの測定式 (ISO 原典メトリクスではない)	測定の具体化方針 左記の測定式にした経緯
・機能適合性 ・機能完全性 ・FCp-1	・様式回答 ・FCp.1.1 ・要求実装率	(製品はユーザーに完全な機能を提供しているか) ・システムが備えるべき機能を実装できているか	$X=A/B$ A=見送らず, 着手・実現した要求の数 B=対象期間内で挙げた要求の数	開発現場においては往々に機能要求/非機能要求の区別と定義が異なってくるため, 両者を区別せずに数えるようにした.
・機能適合性 ・機能正確性 ・FCr-1	・様式回答 ・FCr.1.1 ・深刻不具合除去率	(製品はユーザーに正確な機能を提供しているか) ・システムや機能の深刻な不具合を修正できているか	$X=1-A/B$ (システム試験) A=そのうち, 残存している不具合の数 B=対象期間内で発見した深刻な不具合の数	個々の機能が正確さを測定したいが, それ判断する簡便で汎用的な手法が無いため, 不具合が除去されているかによって代替測定することとした. 本来は数値計算や入力チェック機能の正確さを測れるとよい.
・機能適合性 ・機能適切性 ・FAp-1	・様式回答 ・FAp.1.1 ・システム試験数 目標達成率	(製品はユーザーに適切な機能を提供しているか) ・システムや機能に不具合がないかどうか十分確認しているか	$X=A/B$ (システム試験) A=対象期間内でのテストケース実施数 B=対象期間内でのテストケースの実施目標数	原典の FAp メトリクスは測定が困難で, それは FCp と FCr の測定対象それぞれが同じ粒度で測れていなければならないからであった. そのため今回は, システム試験が十分なさされているかで機能の適切さを代替測定することとした. 今回の研究では, B の値に試験密度目標値を代入し, A の値にテストケース数を製品規模で割った値による代替測定をした.
・機能適合性 ・機能適切性 ・FAp-1	・ユーザーアンケート ・FAp.1.2 ・ユーザーの意図に則す度合い	・製品はユーザーの意図通りの機能を提供しているか	X' =各ユーザー回答群の X 平均 X =ユーザーアンケート Q5.1, Q5.2 の平均	FAp.1.1 と合わせて, 機能の適切さを代替測定する. ユーザーに対して直接, 機能が意図通りのものだったかを確認することとした.

主特性(Goal) 副特性 (SubGoal) ISO メトリクスID	測定種別 RISE メトリクス (Metrics) メトリクス見出し	具体化前後の GQM の Question (測定の側面) ※複数の Question がひとつ の Metric にリンクする	RISE メトリクスの測定式 (ISO 原典メトリクスではない)	測定の具体化方針 左記の測定式にした経緯
<ul style="list-style-type: none"> 性能効率性 時間効率性 なし 	<ul style="list-style-type: none"> 様式回答 PTb.0.1 時間効率性の試験有無 	<ul style="list-style-type: none"> 時間効率性の試験をしているか 	$X=A/B$ $A=3$ 種の内, 実施しているもの数 $B=3$ ※応答時間, ターンアラウンドタイム, スループット	製品において性能試験は実施していたとしても, 研究チームで想定する形式の記録ではない懸念があった。そこで, 原典にあるメトリクスよりも簡易なメトリクスをゼロ番として用意した。
<ul style="list-style-type: none"> 性能効率性 時間効率性 PTb-1 	<ul style="list-style-type: none"> 様式回答 PTb.1.1 応答時間 平均 	<ul style="list-style-type: none"> 応答時間を十分に短くできたか 	X' =タスク群での X 平均値 X =ある性能試験タスクでの実測値 (応答時間)	製品ドメイン, 対象機能, ニーズによって タスク実施にかかる時間は違ってくる。一概に時間が短ければ短い程よいとも言いきれない。 「時間をかけてでも信頼性を上げる」等が別の品質特性を主としたもので, 性能効率性では時間がかからない度合いにフォーカスすると前提をおきつつ, 今回は小さければ小さいほど良いとした。
<ul style="list-style-type: none"> 性能効率性 時間効率性 PTb-2 	<ul style="list-style-type: none"> 様式回答 PTb.2.1 応答時間 実測対目標 	〃	X' =タスク群での X 平均値 $X=A/B$ (応答時間) A =ある性能試験タスクでの実測値 B =その性能試験タスクでの目標値	目標値が緩いか厳しいかも製品によって異なってくる。ただ PTb.1.1 と同じ前提に立ち, 今回は小さければ小さいほど良いとした。
<ul style="list-style-type: none"> 性能効率性 時間効率性 PTb-3 	<ul style="list-style-type: none"> 様式回答 PTb.3.1 ターンアラウンドタイム 平均 	<ul style="list-style-type: none"> ターンアラウンドタイムを十分に短くできたか 	X' =タスク群での X 平均値 X =ある性能試験タスクでの実測値 (ターンアラウンドタイム)	PTb.1.1 と同じ。
<ul style="list-style-type: none"> 性能効率性 時間効率性 PTb-4 	<ul style="list-style-type: none"> 様式回答 PTb.4.1 ターンアラウンドタイム 実測対目標 	〃	X' =タスク群での X 平均値 $X=A/B$ (ターンアラウンドタイム) A =ある性能試験タスクでの実測値 B =その性能試験タスクでの目標値	PTb.2.1 と同じ
<ul style="list-style-type: none"> 性能効率性 時間効率性 PTb-5 	<ul style="list-style-type: none"> 様式回答 PTb.5.1 スループット 目標達成率 	<ul style="list-style-type: none"> スループットを十分に多くできたか 	X' =タスク群での X 平均値 $X=A/B$ (スループット) A =ある性能試験タスクでの実測値 B =その性能試験タスクでの目標値	製品ドメイン, 対象機能, ニーズによって タスクとスループットも違ってくることを前提に参考までに測定する。各企業での目標値を分母において正規化している。

主特性(Goal) 副特性 (SubGoal) ISO メトリクスID	測定種別 RISE メトリクス (Metrics) メトリクス見出し	具体化前後の GQM の Question (測定の側面) ※複数の Question がひとつ の Metric にリンクする	RISE メトリクスの測定式 (ISO 原典メトリクスではない)	測定の具体化方針 左記の測定式にした経緯
<ul style="list-style-type: none"> 性能効率性 資源効率性 なし 	<ul style="list-style-type: none"> 様式回答 PRu.0.1 資源効率性の試験有無 	<ul style="list-style-type: none"> 資源効率性の試験をしているか 	$X=A/B$ $A=2$ 種の内, 実施しているものの数 $B=2$ (CPU 使用率系試験, メモリ使用率試験)	製品において性能試験は実施していたとしても, 研究チームで想定する形式の記録ではない懸念があった。そこで, 原典にあるメトリクスよりも簡易なメトリクスをゼロ番として用意した。
<ul style="list-style-type: none"> 性能効率性 資源効率性 PRu-1 	<ul style="list-style-type: none"> 様式回答 PRu.1.1 CPU 使用率 最大値 	<ul style="list-style-type: none"> CPU 使用率を押さえられているか 	X' =タスク群での X 平均値 X =ある性能試験タスク実施中の CPU 使用率最大値	<p>原典では CPU 時間という測定量を使用していたが, 開発元・製品提供元でその記録をしていない懸念があった。そのため, CPU 使用率によって代替測定することとした。</p> <p>また, CPU 使用率は低ければ必ず良いというものでもなく, CPU を大きく使用しても処理を早くするということもある。しかし, 原典のメトリクスでは元々, CPU 時間の節約を意図しているので CPU 使用率の最大値が低いかどうかを今回は扱うこととした。</p>
<ul style="list-style-type: none"> 性能効率性 資源効率性 PRu-2 	<ul style="list-style-type: none"> 様式回答 PRu.2.1 メモリ 使用率 最大値 	<ul style="list-style-type: none"> メモリ使用率を押さえられているか 	X' =タスク群での X 平均値 X =ある性能試験タスク実施中のメモリ使用率最大値	PRu-1 と同じく, メモリ使用量の代わりにメモリ使用率の最大値を扱うこととした。
<ul style="list-style-type: none"> 性能効率性 資源効率性 PRu-3 	<ul style="list-style-type: none"> なし なし I/O デバイス使用率関連 		~	未実施であることや, 記録の仕方が個々に異なることなどから統一的な測定が難しく省略した。RISE 様式自体が肥大化すると, 企業の回答協力が得にくくなる懸念があった。
<ul style="list-style-type: none"> 性能効率性 資源効率性 PRu-4 	<ul style="list-style-type: none"> なし なし ストレージ使用率関連 		~	〃
<ul style="list-style-type: none"> 性能効率性 資源効率性 PRu-5 	<ul style="list-style-type: none"> なし なし 帯域幅使用率関連 		~	〃

主特性(Goal) 副特性 (SubGoal) ISO メトリクスID	測定種別 RISE メトリクス (Metrics) メトリクス見出し	具体化前後の GQM の Question (測定の側面) ※複数の Question がひとつ の Metric にリンクする	RISE メトリクスの測定式 (ISO 原典メトリクスではない)	測定の具体化方針 左記の測定式にした経緯
<ul style="list-style-type: none"> 性能効率性 容量満足性 なし 	<ul style="list-style-type: none"> 様式回答 PCa.0.1 容量満足性の試験有無 	<ul style="list-style-type: none"> 容量満足性の試験をしているか 	$X=0$ または 1 ※ユーザ同時アクセス数系の試験をしているかどうか、YES なら 1, NO なら 0	製品において性能試験は実施していたとしても、研究チームで想定する形式の記録ではない懸念があった。そこで、原典にあるメトリクスよりも簡易なメトリクスをゼロ番として用意した。
<ul style="list-style-type: none"> 性能効率性 容量満足性 PCa-1 	<ul style="list-style-type: none"> なし なし トランザクション 		~	PRu-3 と同じ
<ul style="list-style-type: none"> 性能効率性 容量満足性 PCa-2 	<ul style="list-style-type: none"> 様式回答 PCa.2.1 ユーザ同時アクセス可能数 目標達成率 (許容量) 	<ul style="list-style-type: none"> 同時にアクセスできるユーザ数を十分に多くできたか 	X' =タスク群での X 平均値 $X=A/B$ (ユーザ同時アクセス数) A=ある性能試験タスクでの実測値 B=その性能試験タスクでの目標値	製品ドメイン、対象機能、ニーズによって 想定するユーザアクセス数が違ってくることを前提に参考までに測定する。各企業での目標値を分母において正規化している。
<ul style="list-style-type: none"> 性能効率性 容量満足性 PCa-3 	<ul style="list-style-type: none"> なし なし ユーザ同時アクセス可能数 目標達成率 (単位時間中累積) 		~	PRu-3 と同じ
<ul style="list-style-type: none"> 互換性 共存性 CCo-1 	<ul style="list-style-type: none"> 様式回答 CCo.1.1 他製品を共存させて試験する意図の有無 	<ul style="list-style-type: none"> (他製品を同じ環境に共存させても品質に影響がないか) 他製品を試験環境に共存させた上で試験をしているか 	X' =サーバマシン、クライアントマシンの X 平均 $X=A/B$ A=試験環境で意図的に共存させているソフトジャンル数 B=2 ※ソフトジャンル二種:セキュリティソフト、自社ソフト	原典では、対象製品と共存すると考えられる製品について、実際に共存させても品質に影響がないかどうかを扱う。しかし、個々の製品がどんな他製品と共存しうるのかを詳細に特定することは困難である。そのため、今回は共存相手を抽象的なジャンルとして用意して測定することとした。 セキュリティソフトはユーザからの利用要望が高いためソフトであるので、ジャンルのひとつとして用意した。 自社ソフトを試験環境に共存させていることは十分考えられ、また系列製品が多い程数が増えるだろう。そのため、自社ソフトというジャンルで数える。

主特性(Goal) 副特性 (SubGoal) ISO メトリクスID	測定種別 RISE メトリクス (Metrics) メトリクス見出し	具体化前後の GQM の Question (測定の側面) ※複数の Question がひとつ の Metric にリンクする	RISE メトリクスの測定式 (ISO 原典メトリクスではない)	測定の具体化方針 左記の測定式にした経緯
<ul style="list-style-type: none"> ・互換性 ・相互運用性 ・CIn-1 	<ul style="list-style-type: none"> ・様式回答 ・CIn.1.1 ・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率 	<ul style="list-style-type: none"> ・取り扱いファイルについて他製品とのデータ交換がでているか 	$X=A/B$ A=インポート・エクスポート両対応のもの数 B=取り扱っているファイル拡張子の数	サポートすべきファイル拡張子の種類を一概に決めることはできない。そこで仮定として、取り扱いファイルについてインポート/エクスポートの片方しかできない場合、ユーザの落胆があると踏まえて今回の測定式にした。開発側で機能要求として取り扱いに含めていなくても、ユーザ側での期待があると仮定している。
<ul style="list-style-type: none"> ・互換性 ・相互運用性 ・CIn-2 	<ul style="list-style-type: none"> ・なし ・なし ・ 		～	今回の研究においては、開発元・製品提供元からのデータ提供が難しく見送りとした。
<ul style="list-style-type: none"> ・互換性 ・相互運用性 ・CIn-3 	<ul style="list-style-type: none"> ・なし ・なし ・ 		～	〃
<ul style="list-style-type: none"> ・使用性 ・適切認識性 ・UAp-1 	<ul style="list-style-type: none"> ・なし ・なし ・ 		～	〃
<ul style="list-style-type: none"> ・使用性 ・適切認識性 ・UAp-2 	<ul style="list-style-type: none"> ・様式回答 ・UAp.2.1 ・機能の動画説明対応率 	<ul style="list-style-type: none"> ・ユーザが自分にとって適切な製品であるかどうか確認できているようになるか ・ユーザが確認できるデモが公開されているか 	$X=A/B$ A=説明動画が公開されている機能の数 B=機能の数	デモが用意されている機能、というものの数え方について議論した。ホームページでFAQがあったり、個別に説明会を開いているというのもデモと考えられるが、確認が難しい。そのため、今回は機能ごとの動画の有無によって数えることとした。
<ul style="list-style-type: none"> ・使用性 ・適切認識性 ・UAp-3 	<ul style="list-style-type: none"> ・なし ・なし ・ 			今回の研究においては、製品提供元からのデータ提供が難しいと判断し見送りとした。
<ul style="list-style-type: none"> ・使用性 ・習得性 ・ULe-1 	<ul style="list-style-type: none"> ・第三者評価機関(本調査) ・ULe.1.1 ・機能の説明記載率 カタログ 	<ul style="list-style-type: none"> ・製品の機能が説明書に記載されているか 	$X=A/B$ A=機能の内、カタログに載っている数 B=機能の数 ※カタログは製品購入者向けのチラシやPDFで、マニュアルより規模が小さいもの	研究チームでは機能一覧の抽出や個別確認が難しいため、機能一覧の提出は企業側に求め、カタログに記載されているかどうかの確認は第三者評価機関に依頼することとした。原典では、製品機能の詳細説

主特性(Goal) 副特性 (SubGoal) ISO メトリクスID	測定種別 RISE メトリクス (Metrics) メトリクス見出し	具体化前後の GQM の Question (測定の側面) ※複数の Question がひとつ の Metric にリンクする	RISE メトリクスの測定式 (ISO 原典メトリクスではない)	測定の具体化方針 左記の測定式にした経緯
				明の十分さまで扱うが、異なる製品の説明書の詳細度を評価する適切で簡便な手法がないため、今回は機能説明が記載されているかどうかを扱うこととした。
<ul style="list-style-type: none"> ・使用性 ・習得性 ・ULe-1 	<ul style="list-style-type: none"> ・第三者評価機関 (本調査) ・ULe.1.2 ・機能の説明 記載率 マニュアル 	〃	$X=A/B$ A=機能の内、マニュアルに載っている数 B=機能の数 ※マニュアルは製品の詳細説明書であり、製品により数百ページになる	〃
<ul style="list-style-type: none"> ・使用性 ・習得性 ・ULe-2 	<ul style="list-style-type: none"> ・なし ・なし ・ 			今回の研究においては、協力企業からのデータ提供が難しいと判断したため見送りとした。
<ul style="list-style-type: none"> ・使用性 ・習得性 ・ULe-3 	<ul style="list-style-type: none"> ・なし ・なし ・ 			〃
<ul style="list-style-type: none"> ・使用性 ・習得性 ・ULe-4 	<ul style="list-style-type: none"> ・なし ・なし ・ 			〃
<ul style="list-style-type: none"> ・使用性 ・運用操作性 ・UOp-1 	<ul style="list-style-type: none"> ・なし ・なし ・ 			〃
<ul style="list-style-type: none"> ・使用性 ・運用操作性 ・UOp-2 	<ul style="list-style-type: none"> ・なし ・なし ・ 			〃
<ul style="list-style-type: none"> ・使用性 ・運用操作性 ・UOp-3 	<ul style="list-style-type: none"> ・なし ・なし ・ 			〃
<ul style="list-style-type: none"> ・使用性 ・運用操作性 ・UOp-4 	<ul style="list-style-type: none"> ・なし ・なし ・ 			〃
<ul style="list-style-type: none"> ・使用性 ・運用操作性 ・UOp-5 	<ul style="list-style-type: none"> ・なし ・なし ・ 			〃

主特性(Goal) 副特性 (SubGoal) ISO メトリクスID	測定種別 RISE メトリクス (Metrics) メトリクス見出し	具体化前後の GQM の Question (測定の側面) ※複数の Question がひとつ の Metric にリンクする	RISE メトリクスの測定式 (ISO 原典メトリクスではない)	測定の具体化方針 左記の測定式にした経緯
<ul style="list-style-type: none"> ・使用性 ・運用操作性 ・UOp-6 	<ul style="list-style-type: none"> ・様式回答 ・UOp.6.1 ・機能の Undo 対応率 	<ul style="list-style-type: none"> ・実行結果の取り消しができ る機能の割合は高いか 	$X=A/B$ A=元に戻せる機能の数 B=Undo が必要な機能の数 <Undo タイプ 1>数回の画面遷移をしながらフォーム記入等する際に、前の画面に戻ってやり直すことができる。 <Undo タイプ 2>完了した単体の操作を取り消しできること。直前の 1~n 個の操作履歴を Ctrl+Z でなかったことのできる等。	機能によっては、そもそも Undo が不要・不適切なものもある。何をもって元に戻っているのかという判断は申請者に任せると基準がばらばらになりすぎてしまう。そのため、研究チームで Undo タイプを 2 つ想定して様式に含めた。 例えば、ログインやメール配信のような機能は通常やり直しがきかぬので Undo が不要な機能とみなす (Undo タイプ 1,2 に当てはまらない) また、Undo できるかどうかにおいて「30 度右回転した図を 30 度左回転させれば元通り」のように 逆の機能で打ち消すようなものは対象外とし、履歴を元に前の状態に戻るものを対象とする。
<ul style="list-style-type: none"> ・使用性 ・運用操作性 ・UOp-7 	<ul style="list-style-type: none"> ・なし ・なし ・ 			今回の研究においては、製品提供元からのデータ提供が難しく見送りとした。
<ul style="list-style-type: none"> ・使用性 ・運用操作性 ・UOp-8 	<ul style="list-style-type: none"> ・なし ・なし ・ 			//
<ul style="list-style-type: none"> ・使用性 ・ユーザエラー防止性 ・UEp-1 	<ul style="list-style-type: none"> ・様式回答 ・UEp.1.1 ・機能での入力内容チェック 対応率 	<ul style="list-style-type: none"> ・ユーザの入力が間違っているかどうかチェックしているか 	$X=A/B$ [機能数単位] A=B のうち、エラーメッセージや警告文が出る機能の数 B=ユーザインプットを求めている機能の数	原典では、ユーザアクションやユーザインプットによる故障を起こさないようにできているかを扱う。そのため機能において、まずユーザが文字や数字を入力するかどうかを確認する必要がある。あまり対象を多くすると回答協力が得られないため、数値・テキスト・アドレスなどを入力する機能に絞って許容性チェックをしているかどうかを回答してもらうことにした。 *)許容チェックでは入力内容の適切さのチェックをする。入

主特性(Goal) 副特性 (SubGoal) ISO メトリクスID	測定種別 RISE メトリクス (Metrics) メトリクス見出し	具体化前後の GQM の Question (測定の側面) ※複数の Question がひとつ の Metric にリンクする	RISE メトリクスの測定式 (ISO 原典メトリクスではない)	測定の具体化方針 左記の測定式にした経緯
				カミスがないようにレイアウト配慮をした, というようなケースは含まない.
・使用性 ・ユーザエラー 防止性 ・UEp-2	・なし ・なし ・			今回の研究においては, 製品提供元からのデータ提供が難しく見送りとした.
・使用性 ・ユーザエラー 防止性 ・UEp-3	・なし ・なし ・			〃
・使用性 ・ユーザインタ フェース快美 性 ・UIn-1	・ユーザアンケート ・UIn.1.1 ・UIの使いやすさ 度合い	・製品の見かけの要素がユ ーザにとって心地よい物か	X' =各ユーザ回答群の X 平均 X =ユーザアンケート Q-In-1 の 得点	パネルやレイアウト等, 画面を 構成する要素がソフトウェアに よって大きく異なる. そのた め, ユーザアンケートで直接 ユーザの実感を回答してもら うこととした.
・使用性 ・アクセシビ リティ ・UAc-1	・なし ・なし ・			今回の研究においては, 製品提供元からのデータ提供が難しく見送りとした.
・使用性 ・アクセシビ リティ ・UAc-2	・なし ・なし ・			〃
・使用性 ・アクセシビ リティ ・UAc-3	・様式回答 ・UAc.3.1 ・機能の聴覚ハン ディキャップ配慮 率	ユニバーサルデザインへの 対応(視覚・聴覚が主)	$X=A/B$ A=機能の内, 聴覚ハンディキ ャップへの配慮があるものの数 B=機能の数	ユニバーサルデザインについ ては, 意図せず聴覚障害のあ る人にとって使いやすかった という場合と, 意図してそのよ うに作った場合とが考えられる. 今回は意図的なもののみを数 えることとした.
・使用性 ・アクセシビ リティ ・UAc-4	・様式回答 ・UAc.4.1 ・機能の視覚ハン ディキャップ配慮 率	〃	$X=A/B$ A=機能の内, 視覚ハンディキ ャップへの配慮があるものの数 B=機能の数	〃

主特性(Goal) 副特性 (SubGoal) ISO マトリクスID	測定種別 RISE マトリクス (Metrics) マトリクス見出し	具体化前後の GQM の Question (測定の側面) ※複数の Question がひとつ の Metric にリンクする	RISE マトリクスの測定式 (ISO 原典マトリクスではない)	測定の具体化方針 左記の測定式にした経緯
・使用性 ・アクセシビリティ ・UAc-5	・様式回答 ・UAc.5.1 ・言語の対応度合い	複数観点別に、言語に対応できている度合い	X=A/B A=各言語の対応度重み総和 B=対応言語の数	ソフトウェアが日本語や英語に対応していると判断する際の基準は人によって違ってくる。基準を整理して、4つの観点に分割した。詳細は RISE 様式に記載。
・信頼性 ・成熟性 ・RMa-1	・様式回答 ・RMa.1.1 ・不具合除去率(単体試験)	・発見した不具合を修正できているか	X=A/B A=Bのうち修正されたものの数 B=対象期間内に発見した欠陥数(fault)	原典では「信頼性関係の不具合」を数えることとしているが、その区分が明確でないため今回は試験で発見した不具合は全て信頼性に影響を与えうるとみなした。
・信頼性 ・成熟性 ・RMa-1	・様式回答 ・RMa.1.2 ・不具合除去率(結合試験)	〃	X=A/B A=Bのうち修正されたものの数 B=対象期間内に発見した欠陥数(fault)	〃
・信頼性 ・成熟性 ・RMa-1	・様式回答 ・RMa.1.3 ・不具合除去率(システム試験)	〃	X=A/B A=Bのうち修正されたものの数 B=対象期間内に発見した欠陥数(fault)	〃
・信頼性 ・成熟性 ・RMa-2	・様式回答 ・RMa.2.1 ・MTBF 目標達成率	(故障頻度を目標より少なくできたか) ・MTBF が目標値より大きくなるように製品を運用できたか	X=C/D A=稼働時間 B=故障発生回数 C=MTBF 実測値=A/B D=MTBF 目標値	原典での測定対象が元々明確である。ただ、目標値を定めていない企業がある懸念がある。
・信頼性 ・成熟性 ・RMa-3	・様式回答 ・RMa.3.1 ・不具合発見率(単体試験)	・目標数をより少ない不具合しか起きずに済んだか	X=A/B A=対象期間内での不具合発見数(実測) B=対象期間内での不具合発見数の目標値 ※単体試験	今回は発見すべき不具合の目標値について、個別の企業側での目標値を使うこととした。運用試験の記録は残っていない懸念があるので、テスト期間中の単体・結合・システム試験による不具合を対象とする。
・信頼性 ・成熟性 ・RMa-3	・様式回答 ・RMa.3.2 ・不具合発見率(結合試験)	〃	X=A/B A=対象期間内での不具合発見数(実測) B=対象期間内での不具合発見数の目標値 ※結合試験	〃

主特性(Goal) 副特性 (SubGoal) ISO メトリクスID	測定種別 RISE メトリクス (Metrics) メトリクス見出し	具体化前後の GQM の Question (測定の側面) ※複数の Question がひとつ の Metric にリンクする	RISE メトリクスの測定式 (ISO 原典メトリクスではない)	測定の具体化方針 左記の測定式にした経緯
・信頼性 ・成熟性 ・RMa-3	・様式回答 ・RMa.3.3 ・不具合発見率(システム試験)	〃	$X=A/B$ A=対象期間内での不具合発見数 (実測) B=対象期間内での不具合発見数の目標値 ※システム試験	〃
・信頼性 ・成熟性 ・RMa-3	・不具合管理データ ・RMa.3.4 ・不具合発見率(チケットベース)	〃	$X=B/(B-abs(B-A))$ A=対象期間内での不具合発見数 (実測) B=対象期間内での不具合発見数の予測値 abs=絶対値関数 ※Bを信頼性曲線を元に算出	製品提供元側での目標値のみならず、不具合管理台帳やチケットを元に信頼性曲線を描き、対象期間での累積発見不具合数の予測値を出した。この予測値を発見数目標値とした。
・信頼性 ・成熟性 ・RMa-4	・様式回答 ・RMa.4.1 ・試験実施率 (結合試験)	・目標数を満たすようにテストケースを実施できたか	$X=A/B$ (結合試験) A=対象期間内でのテストケース実施数 B=対象期間内でのテストケースの実施目標数	今回の研究では、Bの値に試験密度目標値を代入し、Aの値にテストケース数を製品規模で割った値による代替測定をした。
・信頼性 ・成熟性 ・RMa-4	・様式回答 ・RMa.4.2 ・試験実施率 (システム試験)	〃	$X=A/B$ (システム試験) A=対象期間内でのテストケース実施数 B=対象期間内でのテストケースの実施目標数	〃
・信頼性 ・可用性 ・なし	・様式回答 ・RAv.0.1 ・運用試験有無	・運用試験をしているか	$X=0$ または 1 ※運用試験をしているかどうか。YESなら1, NOなら0	製品において運用試験は実施していたとしても、研究チームで想定する形式の記録ではない懸念があった。そこで、原典にあるメトリクスよりも簡易なメトリクスをゼロ番として用意した。
・信頼性 ・可用性 ・RAv-1	・様式回答 ・RAv.1.1 ・運用実時間 対 規定時間	・スケジュールした規定時間の通りに製品を運用できたか	$X=A/B$ A=継続運用時、実際に製品が正常稼働できた時間 B=継続運用時、製品が正常稼働し続けられる時間の予想値 (スケジュールにおける規定時間) ※長期稼働が前提	運用試験関連のメトリクスなので、記録のある製品が少ない懸念がある。メトリクスとしては原典のままでも具体的である。

主特性(Goal) 副特性 (SubGoal) ISO メトリクスID	測定種別 RISE メトリクス (Metrics) メトリクス見出し	具体化前後の GQM の Question (測定の側面) ※複数の Question がひとつ の Metric にリンクする	RISE メトリクスの測定式 (ISO 原典メトリクスではない)	測定の具体化方針 左記の測定式にした経緯
<ul style="list-style-type: none"> 信頼性 可用性 RAv-2 	<ul style="list-style-type: none"> 様式回答 RAv.2.1 システムダウン時間 実際対目標 	<ul style="list-style-type: none"> 運用中のダウン時間をで目標時間より短くできたか 	<ul style="list-style-type: none"> ※対象期間内の実運用・運用試験データにて $X=(A/B)/C$ A= システムダウンしていた時間合計 B=システムダウン回数 C=システムダウン時間平均の目標値 	<ul style="list-style-type: none"> 運用試験関連のメトリクスなので、記録のある製品が少ない懸念がある。メトリクスとしては原典のままでも具体的である。
<ul style="list-style-type: none"> 信頼性 可用性 RAv-3 	<ul style="list-style-type: none"> なし なし なし 			<ul style="list-style-type: none"> 今回の研究においては、製品提供元からのデータ提供が難しく見送りとした。
<ul style="list-style-type: none"> 信頼性 障害許容性 RFt-1 	<ul style="list-style-type: none"> 様式回答 RFt.1.1 fault-pattern テストケース(結合試験) 	<ul style="list-style-type: none"> システムで深刻な不具合が発生しないようになっているか 	<ul style="list-style-type: none"> $X=A/B$ A=Bのうち、成功した数 B=対象期間内での faultPattern テストケース数 	<ul style="list-style-type: none"> 原典では fault-pattern テストケースというものを扱うが、今回は意図的にシステムを望ましくない状態に持ち込むテストであるとみなした。 ※:faultPattern テストケースの例 負荷テスト, 誤入力テスト, 誤操作テスト, 途中シャットダウン, 途中うち止め, いきなり接続落とす, 境界値などのエラーが起きやすいとされる値を使ったテスト, 不正アクセス, 巨大ファイルアップロード, 大量アクセス, 超低性能マシンでの稼働, 低性能ネットワークでの運用, 未対応形式のファイル受信, 不正なメッセージ受信, パケットノイズ, マニュアル手順通りでない操作・運用, 低電力状態, 他の製品でcpu やメモリ食いつぶした状態, 高トラフィック状態, 未対応言語使用, 文字数オーバー, 設定ファイルに仕様外記述, 処理中の使用データの内容書き換え, など。

主特性(Goal) 副特性 (SubGoal) ISO メトリクスID	測定種別 RISE メトリクス (Metrics) メトリクス見出し	具体化前後の GQM の Question (測定の側面) ※複数の Question がひとつ の Metric にリンクする	RISE メトリクスの測定式 (ISO 原典メトリクスではない)	測定の具体化方針 左記の測定式にした経緯
<ul style="list-style-type: none"> 信頼性 障害許容性 RFt-1 	<ul style="list-style-type: none"> 様式回答 RFt.1.2 fault-pattern テストケース(システム試験) 	〃	〃	〃
<ul style="list-style-type: none"> 信頼性 障害許容性 RFt-2 	<ul style="list-style-type: none"> なし なし . 			今回の研究においては、製品提供元からのデータ提供が難しく見送りとした。
<ul style="list-style-type: none"> 信頼性 障害許容性 RFt-3 	<ul style="list-style-type: none"> なし なし . 			〃
<ul style="list-style-type: none"> 信頼性 回復性 RRe-1 	<ul style="list-style-type: none"> 様式回答 RRe.1.1 システムダウン回復時間 実際対目標 	<ul style="list-style-type: none"> ダウン回復をで目標時間より早く完了できたか 	<ul style="list-style-type: none"> ※対象期間内の実運用・運用試験データにて $X=(A/B)/C$ A= システムダウン回復にかかった時間合計 B=システムダウン回数 C=システム回復時間平均の目標値 	システムダウンを回復させるためにかけた時間記録としては、「ダウンしていた時間」と「回復のための作業時間」の二つが考えられるため、この質問を設問中に用意した。
<ul style="list-style-type: none"> 信頼性 回復性 RRe-2 	<ul style="list-style-type: none"> なし なし . 			今回の研究においては、製品提供元からのデータ提供が難しく見送りとした。
<ul style="list-style-type: none"> セキュリティ 機密性 SCo-1 	<ul style="list-style-type: none"> 様式回答 SCo.1.1 データのアクセス権限管理 対応率 	<ul style="list-style-type: none"> システム内のデータがアクセス権限管理で保護されているか 	<ul style="list-style-type: none"> $X=A/B$ A=Bのうち、アクセス権限管理が可能なもの数 B=データ種別の数 	原典では「データアイテム」を対象に測定をする。今回は様式記入の負担も踏まえ、データテーブル単位での回答をってもらうことにした。データベース中の個別のフィールドや列では粒度が細かいすぎて記入協力を得られない。また、ソフトウェアではデータベースを利用せずに隠しファイルやレジストリ等で保持しているデータもあるため、「データテーブル以外」の欄も用意した。
<ul style="list-style-type: none"> セキュリティ 機密性 SCo-2 	<ul style="list-style-type: none"> 様式回答 SCo.2.1 データの暗号化 対応率 	<ul style="list-style-type: none"> システム内のデータ自体が暗号化されているか（経路の暗号化でない） 	<ul style="list-style-type: none"> $X=A/B$ A=Bのうち、暗号化されているものの数 B=データ種別の数 	〃

主特性(Goal) 副特性 (SubGoal) ISO メトリクスID	測定種別 RISE メトリクス (Metrics) メトリクス見出し	具体化前後の GQM の Question (測定の側面) ※複数の Question がひとつ の Metric にリンクする	RISE メトリクスの測定式 (ISO 原典メトリクスではない)	測定の具体化方針 左記の測定式にした経緯
			(通信経路の暗号化含まない。 経路暗号化は別メトリクスで扱う)	
・セキュリティ ・機密性 ・SCo-3	・なし ・なし ・			今回の研究においては、製品提供元からのデータ提供が難しく見送りとした。
・セキュリティ ・インテグリティ ・SIn-1	・なし ・なし ・			〃
・セキュリティ ・インテグリティ ・SIn-2	・様式回答 ・SIn.2.1 ・データの破損防止策 対応率	・システム内のデータが破損防止技術で保護されているか	X=A/B A=Bのうち、破損防止機能のあるものの数 B=データ種別の数 ※ユーザが手動でバックアップの置き置きをしていなくても、データの復元が可能か	原典ではソフトウェアが持っているデータ破損防止策を数えることになっている。このままだと、「RAID あり」程度の回答しか得られないこともありうるので、観点を分割した。ソフトウェアで使っているデータアイテムについて、破損防止策が適用されているかどうかを回答してもらうこととした
・セキュリティ ・インテグリティ ・SIn-3	・なし ・なし ・			今回の研究においては、製品提供元からのデータ提供が難しく見送りとした。
・セキュリティ ・否認防止性 ・SNo-1	・様式回答 ・SNo.1.1 ・ネットワーク経路のデジタル署名対応率	・製品で利用する通信経路がデジタル署名によって改ざんから保護されているか ①社内サーバのみ使用する経路 ②社外サーバも使用する経路 ③クライアント間直接通信(P2P 等) ④申請者管理サーバを使用する経路(クラウド等)	X=A/B A=Bのうち、デジタル署名有効なもの数 B=利用している通信経路の種類数	原典ではデジタル署名を利用するイベント発生について扱うが、企業によっては記録を持っていない懸念があるために、管理の重要性の高い通信経路単位で扱うこととした。また、デジタル署名の名前を全面に出してなくても、内部的に利用している技術があるため、その名前を質問に含めることとした。(SSL, VPN) また、パッケージ製品などでは社内ネットワークの詳細を購入者に任せていることもある。そのため、通信経路を抽象化して分類し、各経路種別にお

主特性(Goal) 副特性 (SubGoal) ISO メトリクスID	測定種別 RISE メトリクス (Metrics) メトリクス見出し	具体化前後の GQM の Question (測定の側面) ※複数の Question がひとつ の Metric にリンクする	RISE メトリクスの測定式 (ISO 原典メトリクスではない)	測定の具体化方針 左記の測定式にした経緯
				いてデジタル署名の利用有無を回答してもらうことにした。
<ul style="list-style-type: none"> ・セキュリティ ・責任追跡性 ・SAc-1 	<ul style="list-style-type: none"> ・様式回答 ・SAc.1.1 ・データのアクセスログ 対応率 	<ul style="list-style-type: none"> ・システム内のデータはアクセスされた際のログが残るか 	$X=A/B$ $A=B$ のうち、アクセス履歴がログに残るものの数 $B=$ データ種別の数	原典では実際のアクセスのうち履歴が残るものの数を扱うが、この粒度のデータは企業側には無いとみなした。今回は他のセキュリティメトリクスと同じ枠組みを用い、データに対するアクセス(そのデータにアクセスする機能の利用)に履歴が残るかどうかを回答してもらうことにした。
<ul style="list-style-type: none"> ・セキュリティ ・責任追跡性 ・SAc-2 	<ul style="list-style-type: none"> ・なし ・なし ・ 			今回の研究においては、製品提供元からのデータ提供が難しく見送りとした。
<ul style="list-style-type: none"> ・セキュリティ ・真正性 ・SAu-1 	<ul style="list-style-type: none"> ・様式回答 ・SAu.1.1 ・ログイン機能での認証方式 対応率 	<p>(この製品では多様な認証方式をサポートしているか)</p> <ul style="list-style-type: none"> ・ログイン機能において、多様な認証方式をサポートしているか 	X' =ログイン機能についての X $X=A/B$ $A=$ 製品でサポートしている認証方式の種別数 $B=6$ ※6 種: 固定パスワード, ワンタイムパスワード, 期限付きパスワード, 物理的トークン, 生体認証, 解読型	原典では仕様上必要な認証方式をサポートできているかを扱うが、どの製品でも「完全サポート」となり違いがでない懸念があるので、今回は一般的に知られている認証方式6つの対応率を測ることにした。認証を利用する機能をいくつ持っているかは製品によって異なってくる。そのため今回は、多くの製品が備えているログイン機能に対象を絞って回答してもらうこととした。
<ul style="list-style-type: none"> ・セキュリティ ・真正性 ・SAu-2 	<ul style="list-style-type: none"> ・なし ・なし ・ 			今回の研究においては、製品提供元からのデータ提供が難しく見送りとした。

主特性(Goal) 副特性 (SubGoal) ISO メトリクスID	測定種別 RISE メトリクス (Metrics) メトリクス見出し	具体化前後の GQM の Question (測定の側面) ※複数の Question がひとつ の Metric にリンクする	RISE メトリクスの測定式 (ISO 原典メトリクスではない)	測定の具体化方針 左記の測定式にした経緯
<ul style="list-style-type: none"> 保守性 モジュール性 MMo-1 	<ul style="list-style-type: none"> コード解析 MMo.1.1 クラスの結合度 	(コンポーネント同士の結合度が抑えられているか) <ul style="list-style-type: none"> クラス同士の結合度が抑えられているか 	X' =クラス群での X 平均 X =クラス結合度(クラス単位)	「コンポーネントの結合度」を今回はクラス単位で測ることとした。原典では閾値を用いて望ましい値でないコンポーネントを数えることになっているが、複数の製品で一律に有用な閾値を決めるのは難しい。そのため、今回は閾値を使わず測定値そのままとした。(スコア化時に 0.0~1.0 の値に変換する)
<ul style="list-style-type: none"> 保守性 モジュール性 MMo-2 	<ul style="list-style-type: none"> コード解析 MMo.2.1 関数のサイクロマティック複雑度 	(モジュールのサイクロマティック複雑度が抑えられているか) <ul style="list-style-type: none"> 関数のサイクロマティック複雑度が抑えられているか 	X' =関数群での X 平均 X =サイクロマティック複雑度(関数単位)	「モジュールのサイクロマティック複雑度」を今回は関数単位で測ることとした。原典では閾値を用いて望ましい値でないモジュールを数えることになっているが、複数の製品で一律に有用な閾値を決めるのは難しい。そのため、今回は閾値を使わず測定値そのままとした。(スコア化時に 0.0~1.0 の値に変換する)
<ul style="list-style-type: none"> 保守性 再利用性 MRe-1 	<ul style="list-style-type: none"> コード解析 MRe.1.1 クラスの凝集性の欠如 	(アセットの再利用性が考慮されているか) <ul style="list-style-type: none"> クラスの凝集度が高くできているか 	X' =クラス群での X 平均 $X=100-LCOM2$ (クラス単位) $LCOM2$ =凝集性の欠如の定義 ver2 ※ここでは LCOM2 が 0~100 の値	原典ではシステムのアセットの内、再利用を考慮して作られたものを数える。今回はソースコードの再利用性指標である「凝集性の欠如」を測定することにした。 $LCOM$ は複数の測定バージョンがあるが、今回は測定ソールの都合で $LCOM2$ とした。
<ul style="list-style-type: none"> 保守性 再利用性 MRe-2 	<ul style="list-style-type: none"> コード解析 MRe.2.1 測定保留中 		特定のコーディング規約違反の検出率等	※ メトリクスとしての整理は保留中であるが、報告書では検出されたコーディング規約違反の傾向分析をした。
<ul style="list-style-type: none"> 保守性 解析性 MAn-1 	<ul style="list-style-type: none"> 様式回答 MAn.1.1 データのアクセスログ 対応率 	<ul style="list-style-type: none"> システム内のデータはアクセスされた際のログが残るか 	$X=A/B$ $A=B$ のうち、アクセス履歴がログに残るものの数 B =データ種別の数	原典では実際のアクセスのうち履歴が残るものの数を扱うが、この粒度のデータは企業側には無いとみなした。今回はセキュリティメトリクスと同じ枠組みを用い、データに対す

主特性(Goal) 副特性 (SubGoal) ISO マトリクスID	測定種別 RISE マトリクス (Metrics) マトリクス見出し	具体化前後の GQM の Question (測定の側面) ※複数の Question がひとつ の Metric にリンクする	RISE マトリクスの測定式 (ISO 原典マトリクスではない)	測定の具体化方針 左記の測定式にした経緯
				るアクセス(そのデータにアクセスする機能の利用)に履歴が残るかどうかを回答してもらうことにした。
・保守性 ・解析性 ・MA _n -2	・なし ・なし ・			今回の研究においては、製品提供元からのデータ提供が難しく見送りとした。
・保守性 ・解析性 ・MA _n -3	・なし ・なし ・			〃
・保守性 ・修正性 ・MMd-1	・様式回答 ・MMd.1.1 ・測定保留中		修正情報のうち、不具合修正情報について X=合計(Ai/Bi)/n Ai=修正 i にかかった時間 Bi=修正 i についての目標時間 n=修正の数 目標時間 B は研究チームが業界平均やその他に基づいて決める。	
・保守性 ・修正性 ・MMd-2	・様式回答 ・MMd.2.1 ・流用箇所の不具合除去率	(修正目的で手を加えて逆に不具合を発生させていないか) ・既存流用コードに対する回帰試験で不具合を発生させていないか	X2=A/B A=そのうち、残存不具合数 B=回帰試験で発見した不具合数	原典では、修正を行った回数と、手を加えたせいで逆に不具合を発生させてしまった修正の回数 を扱う。しかし、この詳細なデータは企業が持っていない懸念がある。 流用箇所に対し回帰試験で不具合が発見されるかどうかで、今回は代替測定することとした。
・保守性 ・修正性 ・MMd-2	・様式回答 ・MMd.2.2 ・測定保留中		修正情報のうち、不具合修正情報について X=A/B A=不具合を発生させてしまった修正案件の数 B=済んだ修正案件の数	

主特性(Goal) 副特性 (SubGoal) ISO メトリクスID	測定種別 RISE メトリクス (Metrics) メトリクス見出し	具体化前後の GQM の Question (測定の側面) ※複数の Question がひとつ の Metric にリンクする	RISE メトリクスの測定式 (ISO 原典メトリクスではない)	測定の具体化方針 左記の測定式にした経緯
<ul style="list-style-type: none"> 保守性 修正性 MMd-3 	<ul style="list-style-type: none"> 様式回答 MMd.3.1 不具合除去率(単体試験で発見分) 	(修正すべきアイテムが修正できているか) <ul style="list-style-type: none"> 発見した不具合を修正できているか 	$X=A/B$ A=修正済み不具合数 B=発見済み不具合数 (単体試験で発見した不具合分)	原典では、新たに起票された修正案件と、そのうちで解消されたものを取り扱うメトリクスであった。しかし詳細かつ規模の大きいデータになるため、今回は修正案件の内で見出した不具合の発見と修正に対象を絞った。
<ul style="list-style-type: none"> 保守性 修正性 MMd-3 	<ul style="list-style-type: none"> 様式回答 MMd.3.2 不具合除去率(結合試験で発見分) 	"	$X=A/B$ A=修正済み不具合数 B=発見済み不具合数 (結合試験で発見した不具合分)	"
<ul style="list-style-type: none"> 保守性 修正性 MMd-3 	<ul style="list-style-type: none"> 様式回答 MMd.3.3 不具合除去率(システム試験で発見分) 	"	$X=A/B$ A=修正済み不具合数 B=発見済み不具合数 (システム試験で発見した不具合分)	"
<ul style="list-style-type: none"> 保守性 試験性 MTe-1 	<ul style="list-style-type: none"> 様式回答 MTe.1.1 モジュールの単体試験 実施率 	(システム上のモジュールのテスト設備が用意されているか) <ul style="list-style-type: none"> 開発側でモジュールの単体テストを用意しているか 	$X1=C/D$ C=単体テストを一件でもしたモジュールの数 D=全モジュール数	原典では Number of test functions によって、開発元において試験が十分できる状況にあるかを取り扱う。しかし、何がテスト機能・テスト関数なのかの判断は製品ごとに異なってくる。そのため、今回はモジュールの単体テストを用意しているかどうかで代替測定することとした。単体テストがあれば、モジュールを微修正した後でも以前のモジュールと同じ結果を返すかどうかを確認することができる。(モジュール構造から大幅に変更しなければ)モジュールはソフトウェアの最小構成単位であるが、これについても企業によって捉え方が異なるので、様式ではモジュールの説明記述回答も求めた。

主特性(Goal) 副特性 (SubGoal) ISO メトリクスID	測定種別 RISE メトリクス (Metrics) メトリクス見出し	具体化前後の GQM の Question (測定の側面) ※複数の Question がひとつ の Metric にリンクする	RISE メトリクスの測定式 (ISO 原典メトリクスではない)	測定の具体化方針 左記の測定式にした経緯
<ul style="list-style-type: none"> 保守性 試験性 MTe-1 	<ul style="list-style-type: none"> 様式回答 MTe.1.2 コードカバレッジ 	各種コードカバレッジの測定結果	$X' = X$ の平均 $X =$ プログラミング言語別のコードカバレッジ ※昨今のソフトウェアは多言語開発のものも多い	製品提供元によってはコードカバレッジを自プロジェクトで測定している場合もあるので、今回は記録があれば求めるということにした。
<ul style="list-style-type: none"> 保守性 試験性 MTe-2 	<ul style="list-style-type: none"> なし なし . 			今回の研究においては、製品提供元からのデータ提供が難しく見送りとした。
<ul style="list-style-type: none"> 保守性 試験性 MTe-3 	<ul style="list-style-type: none"> なし なし . 			〃
<ul style="list-style-type: none"> 移植性 適応性 なし 	<ul style="list-style-type: none"> 様式回答 PAd.0.1 複数環境の試験有無 	<ul style="list-style-type: none"> 複数環境で試験をしているか 各機能において複数環境試験をしているか 	$X1 = A/B$ $A =$ 複数環境で試験してある主機能数 $B =$ 主機能数 $X2 = C/B$ $C =$ 複数環境で試験成功した主機能数	原典においては、ソフト・ハード・運用での三種の環境の違いを考慮して PAd-1, PAd-2, PAd-3 の三つのメトリクスを用意していた。しかし、ミドルウェアや OS 等が三種のどの区分に当てはまるかの判断は企業ごとに違う回答をしてくる懸念があった。そこで、原典にあるメトリクスよりも簡易なメトリクスをゼロ番として用意した。
<ul style="list-style-type: none"> 移植性 適応性 PAd-1 	<ul style="list-style-type: none"> なし なし . 			今回の研究においては、製品提供元からのデータ提供が難しく見送りとした。
<ul style="list-style-type: none"> 移植性 適応性 PAd-2 	<ul style="list-style-type: none"> なし なし . 			〃
<ul style="list-style-type: none"> 移植性 適応性 PAd-3 	<ul style="list-style-type: none"> なし なし . 			〃
<ul style="list-style-type: none"> 移植性 設置性 なし 	<ul style="list-style-type: none"> 様式回答 Pln.0.1 インストールの試験有無 	<ul style="list-style-type: none"> インストール試験をしているか 	$X = 0$ または 1 ※インストール時間系の試験をしているかどうか。YES なら 1, NO なら 0	製品においてインストール試験は実施していたとしても、研究チームで想定する形式の記録ではない懸念があった。そこで、原典にあるメトリクスよりも簡易なメトリクスをゼロ番として用意した。

主特性(Goal) 副特性 (SubGoal) ISO メトリクスID	測定種別 RISE メトリクス (Metrics) メトリクス見出し	具体化前後の GQM の Question (測定の側面) ※複数の Question がひとつ の Metric にリンクする	RISE メトリクスの測定式 (ISO 原典メトリクスではない)	測定の具体化方針 左記の測定式にした経緯
<ul style="list-style-type: none"> ・移植性 ・設置性 ・Pln-1 	<ul style="list-style-type: none"> ・様式回答 ・Pln.1.1 ・インストール時間平均 	<ul style="list-style-type: none"> ・インストール試験で目標時間より短く完了できたか 	X' =タスク群での X 平均値 $X=A/B$ (インストール時間) A=あるタスクでの実測値 B=あるタスクでの目標値	製品によって インストールにかかる時間は違ってくる。インストール作業も製品によって異なってくるので、インストールに関するタスクの実施時間 という枠組みで タスクの応答時間等と同様の枠組みで回答を求めた。
<ul style="list-style-type: none"> ・移植性 ・設置性 ・Pln-2 	<ul style="list-style-type: none"> ・様式回答 ・Pln.2.1 ・インストーラ提供形態 対応率 	<ul style="list-style-type: none"> ・ユーザの都合に合わせてられるように多種の媒体で製品を提供しているか 	$X=A/B$ A=3 種の内、対応している製品提供形態 B=3 ※種別: Web, CD, セットアップ代行	ユーザの都合に合わせてインストールの仕方を変えられるかどうかを判断するための大まかな基準として、インストーラ提供形態を回答してもらうことにした。
<ul style="list-style-type: none"> ・移植性 ・設置性 ・Pln-2 	<ul style="list-style-type: none"> ・様式回答 ・Pln.2.2 ・インストールオプション 対応率 	<ul style="list-style-type: none"> ・ユーザの都合に合わせてられる柔軟なインストールオプションがあるか 	X' =サーバソフトでの X $X=A/B$ A=8 種の内、対応しているインストールオプション B=8 ※詳細は様式に記載	また Pln.2.1 は大まかすぎるため、より詳細な回答を求める Pln.2.2 を用意した。
<ul style="list-style-type: none"> ・移植性 ・置換性 ・PRe-1 	<ul style="list-style-type: none"> ・ユーザアンケート ・PRe.1.1 ・追加学習必要度合い 	<ul style="list-style-type: none"> ・ユーザが追加学習の手間なく製品を利用できるか 	X' =各ユーザ回答群の X 平均 X =ユーザアンケート Q8.3 の得点	製品の各機能について、ユーザにとって追加学習が必要そうな度合いを推定する汎用の手法がないため、ユーザアンケートによる代替測定となった。
<ul style="list-style-type: none"> ・移植性 ・置換性 ・PRe-2 	<ul style="list-style-type: none"> ・なし ・なし ・ 			今回の研究においては、製品提供元からのデータ提供が難しく見送りとした。
<ul style="list-style-type: none"> ・移植性 ・置換性 ・PRe-3 	<ul style="list-style-type: none"> ・なし ・なし ・ 			〃
<ul style="list-style-type: none"> ・移植性 ・置換性 ・PRe-4 	<ul style="list-style-type: none"> ・なし ・なし ・ 			〃

② ソースコード解析（研究プロセス①②）

本調査研究においては、一部の内部測定法の測定に必要なデータを取得するために、ソースコードの静的解析ツールとして Understand と CheckStyle を用いた。コード解析は保守性に属する二つの副特性であるモジュール性と再利用性のメトリクスを測定するために必要であった。

Understand は C/C++, Java, Python といった複数のプログラミング言語に対応しており、コードの複雑度や結合度を測定することができる。一方、CheckStyle では Java のみであるが Google コーディング規約違反箇所の場所と個数を検出できる。

③ スコア化・集約方式（研究プロセス③）

本調査研究においては、品質主特性や副特性の比較をする場合にスコアを利用する。ある副特性 A に紐づいた各メトリクスのスコアの平均をその副特性 A のスコアとし、主特性については副特性の平均をとる。

各メトリクスのスコア化関数としてパーセンタイルを用いる。21 製品で測定できたメトリクスであれば、21 個の測定値の集合におけるパーセンタイルのランク値に変換される。パーセンタイルのランク値とは、図 3-1-4 のようにメトリクス測定値を値の大きさに並び替えた際にそれぞれの値がどの位置に来るかを表した数値である。例えば、上位 30% に位置する測定値のランク値は 0.7 (1.00-0.30) となる。変換後のランク値は通常 0.00~1.00 の値となるが、{0, 1, 5, 5} のように集合の中で最大値の要素が複数ある場合、ランク値の最大値が 1.00 にならない。

メトリクスによっては、値が大きい方が望ましいもの(望大)と、値が小さい方が望ましいもの(望小)とがある。そのため本調査研究においては、望大メトリクスは上述のように上位 30% のランク値が 0.7 となるよう計算し、望小メトリクスは上述のように下位 30% のランク値が 0.7 となるよう計算する。なお、このスコア化の方法は、「大きすぎず小さすぎず、適当な範囲の値をとることが望ましい」場合を考慮しておらず、その場合のスコア化の方法の検討は今後の課題である。

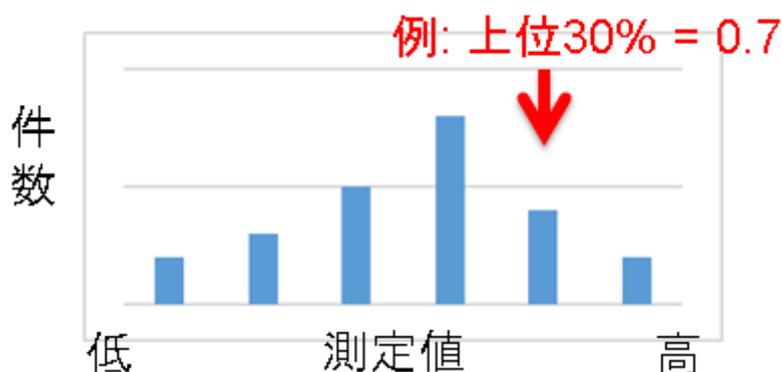


図 3-1-4 パーセンタイルによるスコア化

3.1.3 発生した問題および今後の展望

(1) 発生した問題

ISO/IEC 25023 において定義される多くの製品品質メトリクスについて定義の抽象度が高く、そのままでは具体的な測定が困難であることが分かった。そこで、初期案として一度具体化したうえで一つの製品について初期トライアルとして試験的な測定を実施し、その結果に基づき改訂を通じて段階的な具体化により解決した。

(2) 今後の展望

具体化できなかった（現実には測定困難な）製品品質メトリクスについて継続検討する予定である。これには、国際標準化機構 ISO に対する製品品質メトリクス自体の改訂の働きかけも含む。同時に、測定評価ツールの利便性を向上させる予定である。

例えば、表 3-1-1 における再利用性に関するメトリクス MRe-2 は再利用のしやすさを求めるものであるが、製品を超えて共通に得られるソフトウェアの構造や振る舞いの情報が限られているため、具体化は見送った。

3.2 研究課題 2 「外部品質の測定評価方法の確立」

3.2.1 当初の想定

(1) 研究内容

ソフトウェア製品の機能説明書、および、テストの計画・項目・結果を対象とし、ISO/IEC 25010 で規定される外部品質の品質特性について、ISO/IEC 25023 で規定される外部品質の測定法群を具体化する。さらに研究課題 4 の実現を通して測定結果の集約方法と評価基準を設定し、ISO/IEC 25051 に準拠する形で評価方法としてまとめる。測定法の一部については自動測定ツールを実装し、評価方法は全体としてツール化する。

(2) 想定問題と対応策

問題：国際標準としての ISO/IEC 25023 における外部品質測定法は幾らか抽象的に定義されており、具体的な機能説明書やテストの計画書、項目書、結果における特徴を捉えるにあたり具体化が必要である。ただし具体化にあたり、測定可能な事柄や測定法ありきで検討すると、品質を明らかとしようとする本来の目的を見失う可能性がある。

解決策：ISO/IEC 25010 で規定される外部品質の個々の品質特性を、ISO/IEC 25051 が規定する評価の観点から明らかとすることをゴールとして、拡張した GQM 法の適用により、ゴール達成に有用なデータを得るための測定法候補を識別し、その候補に対して ISO/IEC 25023 における外部品質測定法を具体化させる形で対応付けてトップダウンに導出する。ただし対応付けられない候補については新規に測定法を定義する。研究課題 4 の実現を通して必要に応じて測定法を修正し、評価方法をまとめあげ、特定のソフトウェア製品やドメインに依らず汎用性が高いと想定される測定法の一部については自動測定ツールを実装する。

3.2.2 研究プロセスと成果

(1) 研究プロセス

- ① GQM 法の適用を通じて外部品質測定法の候補を識別する。
- ② ISO/IEC25023 や既存測定法の組み入れを考慮して外部品質測定法を定義し、測定に必要なデータを入力して測定値を得るための記入様式を策定する。

- ③ 測定結果の集約方法および評価基準からなる外部品質評価方法を定義し、ツール化する。

(2) 具体的な研究成果の内容

① 外部品質測定法（研究プロセス①②③）

国際規格 ISO/IEC 25023 においては、内部品質と外部品質を測定するためのメトリクスが定義されており、両者は製品品質メトリクスとしてまとめられている。そこで、最初に ISO/IEC 25023 を参照し表 3-1-1 に示すすべてのメトリクスを初期候補として GQM 法を適用し、品質副特性を網羅するように合計 66 メトリクスを定義した（研究プロセス①）。適用にあたり、ISO/IEC 25023 の解釈や、入力として必要な測定データの現実的な利用可能性等について研究レビュー委員会からの助言を得た。続いて、測定に必要なデータを入力して測定値を得るための記入様式を策定した（研究プロセス②）。これらのメトリクスを半自動的に測定するための記入様式や解析ツールについて、WSQB17 に含めて http://www.washi.cs.waseda.ac.jp/?page_id=3479 にて公開している（研究プロセス③）。具体的には、RISE 様式への必要データの記入により、各メトリクスや品質副特性、品質特性単位における評価を得るための中間データを得た。その中間データを集計用のスプレッドシートに転記することで、最終的な評価結果を得た。

製品品質メトリクスの一覧や具体化については 3.1.2.(2) にて述べたため、本節ではテスト結果に関連した外部品質の中でも重要な信頼性を中心に述べる。

テストで発見される欠陥について、発見時間と数の関係を分析することで欠陥数を予測する、ソフトウェア信頼性モデルを用いてソフトウェア製品群の信頼性を評価する。本研究では 21 のソフトウェア製品群のうち 9 製品についてソフトウェアの欠陥を管理する不具合票を収集できた。

ソフトウェア製品の欠陥に関する情報（欠陥の発見時間、欠陥数など）が記載された欠陥票をもとに、今後発生する欠陥数を予測する。実際に発見された欠陥数と予測欠陥数の関係を分析することで対象のソフトウェア製品が十分に欠陥を発見された状態でリリースされているかどうか、もしくは不十分であるかを評価する。対象の 9 のソフトウェア製品については、予測欠陥数が実際に発見された欠陥数を下回るものが 2 製品、上回るものが 7 製品であった。

また、それぞれの欠陥が登録された時点で、逐次的に欠陥数を予測し、その変化を分析する。分析の結果、リリースの前に予測欠陥数が安定な状態である（安定タイプ）、もしくは漸増的に増加する前兆がある（漸増タイプ）、または爆発的に増加する前兆がある（爆発タイプ）が存在することが分かった。図 3-2-1 予測モデルと予測欠陥数の変動にそれぞれのタイプを示す。対象の 9 のソフトウェア製品については、安定タイプが 3 製品、漸増タイプが 3 製品、爆発タイプが 3 製品であった。

安定タイプについては、予測欠陥数の変動がリリースの前に収束し、リリース時の予測欠陥数と概ね同じ値をとっているタイプである。安定タイプについてはテストを実施し、十分な欠陥を発見できていると考えられる。漸増タイプについては、予測欠陥数の変動がリリースの前に収束せず、徐々に予測欠陥数が増加しているタイプである。漸増タイプについてはテストを実施するたびに欠陥が徐々に発見されており、リリース後も欠陥が発見される可

能性があると考えられる。爆発タイプについては、予測欠陥数の変動がリリースの前に収束せず、予測欠陥数が発見済みの欠陥数より大きな値をとっているタイプである。爆発タイプについては、今後も多くの欠陥が発見されると考えられる。

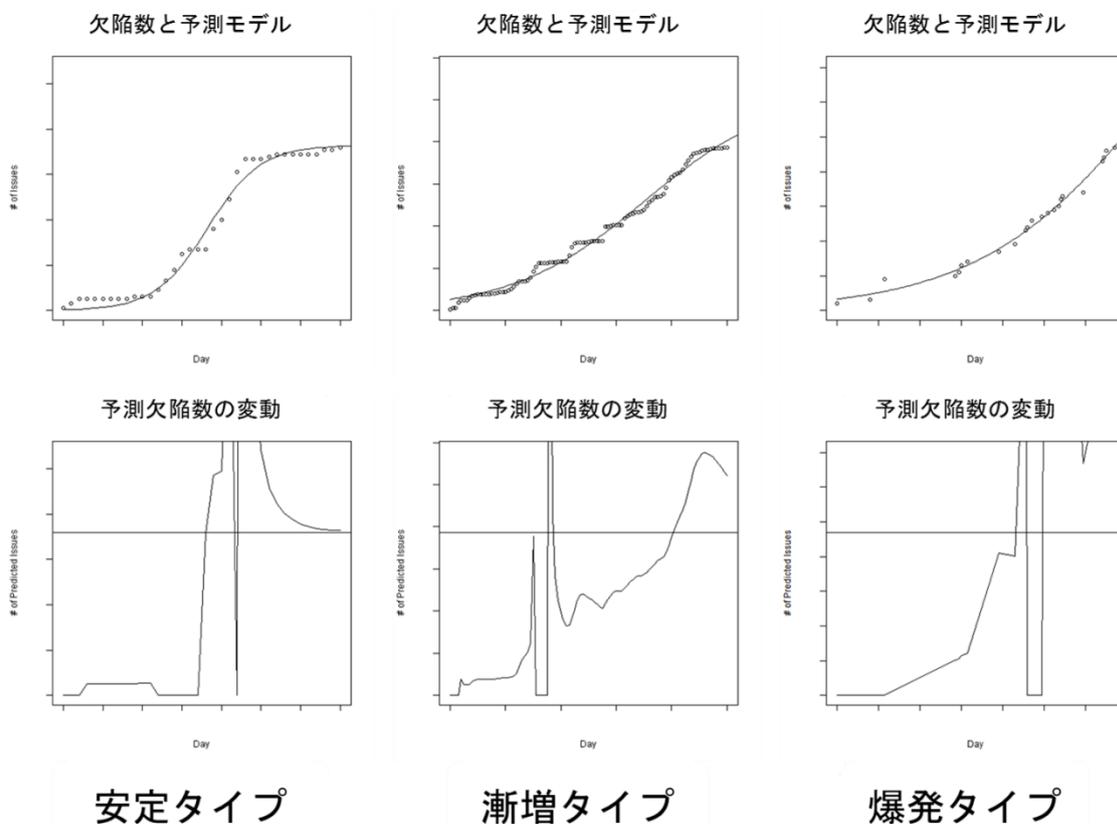


図 3-2-1 予測モデルと予測欠陥数の変動

② スコア化・集約方式（研究プロセス③）

内部品質と同様に、メトリクス単位で測定値をパーセンタイルによりスコア化したうえで、スコアを品質特性・副特性単位で平均化する方式を実現した。

3.2.3 発生した問題および今後の展望

(1) 発生した問題

ISO/IEC 25030 において定義される多くの製品品質メトリクスについて定義の抽象度が高く、そのままでは具体的な測定が困難であることが分かった。そこで、初期案として一度具体化したうえで一つの製品について初期トライアルとして試験的な測定を実施し、その結果に基づく改訂を通じて段階的な具体化により解決した。

(2) 今後の展望

具体化できなかった（現実には測定困難な）製品品質メトリクスについて継続検討する予定である。これには、国際標準化機構 ISO に対する製品品質メトリクス自体の改訂の働きかけも含む。同時に、測定評価ツールの利便性を向上させる予定である。

例えば、表 3-1-1 における修正性に関するメトリクス MMd-1 は不具合修正に関して目標時間に対する実際に修正に要した時間の比を求めるものであるが、修正の目標時間を設定していない場合が多いため、具体化は見送った。データ蓄積と伴って修正時間の業界におけるベンチマークを設定することができれば、そのベンチマークに対する比という形で当該メトリクスを具体化できる可能性がある。

3.3 研究課題 3「利用時の品質の測定評価方法の確立」

3.3.1 当初の想定

(1) 研究内容

ISO/IEC 25010・ISO/IEC 25051 に基づいた測定評価を念頭において、顧客満足度を含む形で利用者アンケートの評価項目を設計する。同アンケートの実施によりソフトウェア製品を実際に導入した利用者の意見と評価結果を対象として、ISO/IEC 25010 および ISO/IEC 25051 で規定される利用時の品質特性について、ISO/IEC 25022 で規定される利用時の品質の測定法群を具体化する。さらに研究課題 4 の実現を通して測定結果の集約方法および評価基準を設定し、ISO/IEC 25051 に準拠する形で評価方法としてまとめあげる。測定法の一部については自動測定ツールを実装し、評価方法は全体としてツール化する。

(2) 想定問題と対応策

問題：国際標準としての ISO/IEC 25022 における利用時の品質測定法は幾らか抽象的に定義されており、具体的な利用者意見や利用者評価における特徴を捉えるにあたり具体化が必要である。ただし具体化にあたり、測定可能な事柄や測定法ありきで検討すると、品質を明らかとしようとする本来の目的を見失う可能性がある。

解決策：ISO/IEC 25010 で規定される利用時の品質の個々の品質特性を、ISO/IEC 25051 が規定する評価の観点から明らかとすることをゴールとして、拡張した GQM 法の適用により、ゴール達成に有用なデータを得るための測定法候補を識別し、その候補に対して ISO/IEC 25022 における利用時の品質測定法を具体化させる形で対応付けてトップダウンに導出する。ただし対応付けられない候補については新規に測定法を定義する。研究課題 4 の実現を通して必要に応じて測定法を修正し、評価方法をまとめあげ、特定のソフトウェア製品やドメインに依らず汎用性が高いと想定される測定法の一部については自動測定ツールを実装する。

3.3.2 研究プロセスと成果

(1) 研究プロセス

- ① 利用者アンケートにおける評価項目を設計する。
- ② GQM 法の適用を通じた利用時の品質測定法の候補を識別する。
- ③ ISO/IEC25022 の組み入れを考慮して利用時の品質測定法を定義し、測定に必要なデータを入力して測定値を得るためのツールとしての記入様式およびユーザテストの設計方法の策定を進める。
- ④ 測定結果の集約方法および評価基準からなる利用時の品質評価方法を定義し、ツール化する。

(2) 具体的な研究成果の内容

① 利用時の品質測定法（研究プロセス①②③④）

国際規格 ISO/IEC 25022 においては、利用時の品質を測定するためのメトリクスが定義されている。そこで、ユーザアンケートの項目を ISO/IEC 25022 を参照して設計し雛形としたうえで（研究プロセス①）、ISO/IEC 25022 を参照し表 3-1-1 に示すすべてのメトリクスを初期候補として GQM 法を適用し、品質副特性を網羅するようにメトリクスを定義した（研究プロセス②）。適用にあたり、ISO/IEC 25022 の解釈や、入力として必要な測定データの現実的な利用可能性等について研究レビュー委員会からの助言を得た。続いて、測定に必要なデータを入力して測定値を得るための記入様式およびユーザテストの設計方法を策定した（研究プロセス③）。これらのメトリクスを半自動的に測定するための記入様式やユーザテストの設計方法について、WSQB17 に含めて

http://www.washi.cs.waseda.ac.jp/?page_id=3479

にて公開している（研究プロセス④）。具体的には、ユーザアンケートやユーザテストの実施を経て RISE 様式への必要データの記入により、各メトリクスや品質副特性、品質特性単位における評価を得るための中間データを得た。その中間データを集計用のスプレッドシートに転記することで、最終的な評価結果を得た。

最終的に 17 メトリクスを具体的に定義することに成功した。それらのうちで、3.4 節において言及する 21 製品のデータ提供において全 17 メトリクスについて測定値が得られた。品質特性単位のメトリクス定義数を図 3-3-1 に示す。全てのメトリクスを実効性のある形で具体的に定義できたことが分かる。

また測定率は 17 メトリクス×21 製品のうちで 24.4%に留まった。未測定の原因の多くは、製品提供元における都合等の理由により以下に示すユーザテストおよびユーザアンケートを必ずしも実施できなかったことにある。より大規模な品質測定評価に向けて、各製品開発における標準的な方法に従ったユーザテストおよびユーザアンケートの実施が課題である。

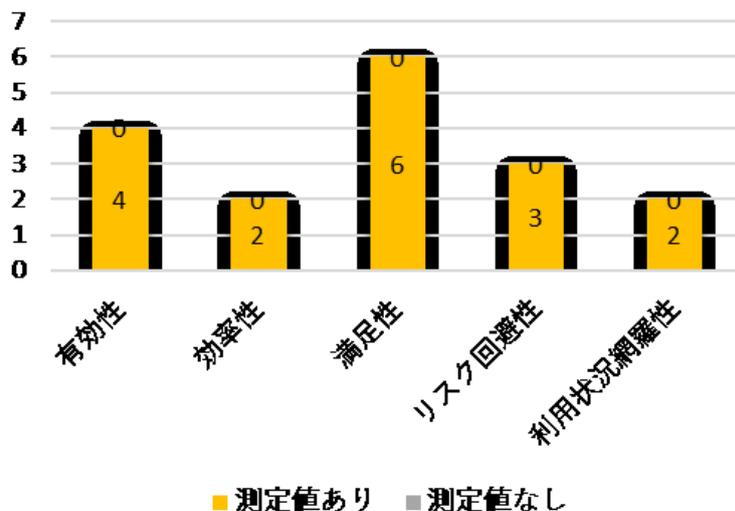


図 3-3-1 利用時の品質メトリクスの定義数

今回の調査研究において具体化した利用時の品質メトリクスについては表 3-3-1 にまとめた。品質主特性(Goal 相当), 副特性(SubGoal 相当), 測定対象とする製品の側面(Question 相当), ISO 25022 の元々のメトリクス ID (Metric), 本研究で具体化したメトリクス, 具体化したメトリクスの具体化経緯等を一つの表に整理されている。

また, 赤色のメトリクス行は今回具体化せずに見送ったものと, 具体化したデータが集まらずに測定できなかったものを意味する。また, 黄色のメトリクス行は元々の ISO 25022 には無い追加したものを意味する。製品提供元からのデータ提供が困難と予想されるメトリクスのみがもともと定義されている品質副特性について, より簡易なデータであっても測定可能なメトリクスを追加し, 実効性を高めた。

表 3-3-1 利用時の品質メトリクスの GQM と具体化方針のまとめ

主特性 (Goal) 副特性 (SubGoal) ISO メトリクス ID	測定種別 RISE メトリクス (Metrics) メトリクス見出し	GQM の Question (測定の側面) ※複数の Q がひとつ の M にリンクする	RISE メトリクスの測定式 (ISO 原典メトリクスでは ない)	測定の具体化方針や問題点 今の測定式にした経緯
・有効性 ・なし ・Ef-1	・ユーザテスト ・Ef.1.1 ・タスク完了率	・製品機能を利用する タスクを正しく完了で きるか	$X=A/B$ A=完了したタスク数 B=全タスク数	ユーザテストの形式は製品提供元ごとに 異なり, 研究チームでテスターを用意して 実際にユーザテストをすることにした. タ スクは企業からの情報を元に研究チーム で整理したものを利用した.
・有効性 ・なし ・Ef-2	・なし ・なし ・			今回の研究においては, 製品提供元から のデータ提供が難しく見送りとした.
・有効性 ・なし ・Ef-3	・ユーザテスト ・Ef.3.1 ・タスク当たりエラー 数	・製品機能を利用する タスクでエラー発生数 が少ないか	$X=A/B$ A=全エラー数 B=全タスク数	エラーについては, エンドユーザ目線での テストを前提として, ユーザがエラーと 感じたものを数えることとした. そのため, システムの不具合が原因となっているエ ラーだけでなく, ユーザの誤操作や勘違 いによって発生したエラーや操作ミスもカ ウントされる.
・有効性 ・なし ・Ef-4	・ユーザテスト ・Ef.4.1 ・エラーが発生し たタスクの率	・エラーが発生したタ スクが少ないか	$X=A/B$ A=エラーがあったタスク 数 B=全タスク数	Ef.3.1 と同じ.
・有効性 ・なし ・Ef-5	・ユーザテスト ・Ef.5.1 ・エラーを起こし た被験者の率	・エラーを発生させた テスターが少ないか	$X=A/B$ A=エラーを起こした人数 B=被験者人数	Ef.3.1 と同じ.
・効率性 ・なし ・Ey-1	・ユーザテスト ・Ey.1.1 ・タスクにかかった 時間の平均	・タスク完了までの時 間が短く抑えられて いるか	X' =タスク群での X 平均 $X=B-A$ A=タスクの開始時刻 B=タスクの終了時刻	タスクの開始時刻と終了時刻の差を タス クにかかった時間とした.
・効率性 ・なし ・Ey-2	・なし ・なし ・			今回の研究においては, 製品提供元から のデータ提供が難しく見送りとした.
・効率性 ・なし ・Ey-3	・なし ・なし ・			〃
・効率性 ・なし ・Ey-4	・なし ・なし ・			〃

主特性(Goal) 副特性 (SubGoal) ISO メトリクスID	測定種別 RISE メトリクス (Metrics) メトリクス見出し	GQM の Question (測定の側面) ※複数の Q がひとつ の M にリンクする	RISE メトリクスの測定式 (ISO 原典メトリクスでは ない)	測定の具体化方針や問題点 今の測定式にした経緯
<ul style="list-style-type: none"> ・効率性 ・なし ・Ey-5 	<ul style="list-style-type: none"> ・ユーザテスト ・Ey.5.1 ・タスク中の総アクションの 無駄でないアクションの率 	<ul style="list-style-type: none"> ・テスターが無駄なアクションをせずに済んでいるか 	$X' = \text{タスク群での } X \text{ 平均}$ $X = A/B$ $A = \text{必要アクション数}$ $B = \text{全アクション数}$	<p>操作の手間取りや間違った操作にかかるアクションをタスク完了に不要なアクションとみなした。</p> <p>また、製品毎に機能・タスク・実施環境・被験者が異なるため、画一的に扱いやすいマウスのクリック数をアクション数とした。</p> <p>クリック数を対象とするため、ショートカットキーによる操作は避けるように統一した。</p>
<ul style="list-style-type: none"> ・効率性 ・なし ・Ey-6 	<ul style="list-style-type: none"> ・なし ・なし ・ 			今回の研究においては、製品提供元からのデータ提供が難しく見送りとした。
<ul style="list-style-type: none"> ・満足性 ・実用性 ・SUs-1 	<ul style="list-style-type: none"> ・ユーザアンケート ・SUs.1.1 ・製品に対する満足度 	<ul style="list-style-type: none"> ・製品全体で総評する場合、ユーザの満足度は高いか 	$X' = \text{各ユーザ回答群の } X \text{ 平均}$ $X = \text{ユーザアンケート Q1 の得点}$	製品全体に対する満足度をまずアンケート冒頭で回答してもらうことにした。
<ul style="list-style-type: none"> ・満足性 ・実用性 ・SUs-1 	<ul style="list-style-type: none"> ・ユーザアンケート ・SUs.1.2 ・ NetPromoterScore 	//	$X' = \text{各ユーザ回答群の } X \text{ 平均}$ $X = \text{ユーザアンケート Q1.1 の得点}$	ユーザビリティ測定の業界で用いられているネットプロモータースコアについてもアンケートに含めることにした。
<ul style="list-style-type: none"> ・満足性 ・実用性 ・SUs-2 	<ul style="list-style-type: none"> ・ユーザアンケート ・SUs.2.1 ・機能に対する満足度 	<ul style="list-style-type: none"> ・機能単位で評価する場合、ユーザの満足度は高いか 	$X' = \text{各ユーザ回答群の } X \text{ 平均}$ $X = \text{ユーザアンケート Q4.N.1 系列の得点平均}$ ※N=1 から始まる整数	製品ごとにユーザアンケートに機能名を挙げて、各機能の満足度を回答してもらうこととした。
<ul style="list-style-type: none"> ・満足性 ・実用性 ・SUs-3 	<ul style="list-style-type: none"> ・なし ・なし ・ 			今回の研究においては、製品提供元からのデータ提供が難しく見送りとした。
<ul style="list-style-type: none"> ・満足性 ・実用性 ・SUs-4 	<ul style="list-style-type: none"> ・なし ・なし ・ 			//
<ul style="list-style-type: none"> ・満足性 ・実用性 ・SUs-5 	<ul style="list-style-type: none"> ・なし ・なし ・ 			//
<ul style="list-style-type: none"> ・満足性 ・実用性 ・SUs-6 	<ul style="list-style-type: none"> ・なし ・なし ・ 			//

主特性(Goal) 副特性 (SubGoal) ISO メトリクスID	測定種別 RISE メトリクス (Metrics) メトリクス見出し	GQM の Question (測定の側面) ※複数の Q がひとつ の M にリンクする	RISE メトリクスの測定式 (ISO 原典メトリクスでは ない)	測定の具体化方針や問題点 今の測定式にした経緯
・満足性 ・信用性 ・STr-1	・ユーザアンケート ・STr.1.1 ・信用度合い	・ユーザが製品を信用しているか	X'=各ユーザ回答群の X 平均 X=ユーザアンケート Q5.1, Q5.2, Q5.3 の得点	ユーザが製品を利用する際に、製品のセキュリティや機能の正確さを信用できているかを回答してもらうこととした。国際規格 ISO/IEC 25022 で参考文献として挙げられている最古メトリクス文献を参考にした。
・満足性 ・快感性 ・SPI-1	・ユーザアンケート ・SPI.1.1 ・快感度合い	・ユーザが機能を気持ちよく利用できているか	X'=各ユーザ回答群の X 平均 X=ユーザアンケート Q6.1～Q6.6 の得点総和 ※Q6.2, Q6.4, Q6.6 は ネガティブ項目なので負 で加算。	ユーザがストレスなくモチベーションの高まるように製品を利用できているかを回答してもらうこととした。国際規格 ISO/IEC 25022 で参考文献として挙げられている最古メトリクス文献を参考にした。
・満足性 ・快適性 ・SCo-1	・ユーザアンケート ・SCo.1.1 ・快適度合い	・ユーザが製品を快適に利用できているか	X'=各ユーザ回答群の X 平均 X=ユーザアンケート Q7.1, Q7.2 の得点平均	ユーザが製品を利用する際に、身体的・精神的に疲れずに済んでいるかを回答してもらうこととした。国際規格 ISO/IEC 25022 で参考文献として挙げられている最古メトリクス文献を参考にした。
・リスク回避性 ・経済リスク緩和性 ・なし	・ユーザアンケート ・REc.0.1 ・経済的損失を感じる頻度	・ユーザが経済的な影響を感じずに済んでいるか	X'=各ユーザ回答群の X 平均 X=ユーザアンケート Q9.1 の得点	原典の経済リスク緩和性メトリクスは、ユーザ企業が製品を利用してどれくらい経済的に影響を受けているかを扱うため、ベンダ企業でも把握しにくい。 そのため、ユーザアンケートにて簡易回答を求めることとした。
・リスク回避性 ・経済リスク緩和性 ・REc-1	・なし ・なし ・			今回の研究においては、製品提供元からのデータ提供が難しく見送りとした。
・リスク回避性 ・経済リスク緩和性 ・REc-2	・なし ・なし ・			〃
・リスク回避性 ・経済リスク緩和性 ・REc-3	・なし ・なし ・			〃
・リスク回避性 ・経済リスク緩和性 ・REc-4	・なし ・なし ・			〃

主特性(Goal) 副特性 (SubGoal) ISO メトリクスID	測定種別 RISE メトリクス (Metrics) メトリクス見出し	GQM の Question (測定の側面) ※複数の Q がひとつ の M にリンクする	RISE メトリクスの測定式 (ISO 原典メトリクスでは ない)	測定の具体化方針や問題点 今の測定式にした経緯
・リスク回避性 ・経済リスク緩和 性 ・REc-5	・なし ・なし ・			〃
・リスク回避性 ・経済リスク緩和 性 ・REc-6	・なし ・なし ・			〃
・リスク回避性 ・経済リスク緩和 性 ・REc-7	・なし ・なし ・			〃
・リスク回避性 ・経済リスク緩和 性 ・REc-8	・なし ・なし ・			〃
・リスク回避性 ・経済リスク緩和 性 ・REc-9	・なし ・なし ・			〃
・リスク回避性 ・健康・安全リス ク緩和性 ・なし	・ユーザアンケー ト ・RHe.0.1 ・健康や人命へ の影響を感じる頻 度	・ユーザが健康・安全 面への影響を感じず に済んでいるか	X' =各ユーザ回答群の X 平均 X =ユーザアンケート Q9.2, Q9.3 の得点平均	原典の健康・安全リスク緩和性メトリクス は、ユーザ企業が製品を利用してどれく らい健康・安全に影響を受けているかを 扱うため、開発元・製品提供元において 把握しにくい。そのため、ユーザアンケ ートにて簡易回答を求めることとした。
・リスク回避性 ・健康・安全リス ク緩和性 ・RHe-1	・なし ・なし ・			今回の研究においては、製品提供元から のデータ提供が難しく見送りとした。
・リスク回避性 ・健康・安全リス ク緩和性 ・RHe-2	・なし ・なし ・			〃
・リスク回避性 ・健康・安全リス ク緩和性 ・RHe-3	・なし ・なし ・			〃
・リスク回避性 ・環境リスク緩和 性	・ユーザアンケー ト ・REn.0.1	・ユーザが環境(自然 環境, 社会環境)への	X' =各ユーザ回答群の X 平均	原典の環境リスク緩和性メトリクスは、ユー ザ企業が製品を利用してどれくらい環境 に影響を受けているかを扱うため、開発

主特性(Goal) 副特性 (SubGoal) ISO メトリクスID	測定種別 RISE メトリクス (Metrics) メトリクス見出し	GQM の Question (測定の側面) ※複数の Q がひとつ の M にリンクする	RISE メトリクスの測定式 (ISO 原典メトリクスでは ない)	測定の具体化方針や問題点 今の測定式にした経緯
性 ・REn-0	・環境への影響を 感じる頻度	影響を感じずに済ん でいるか	X=ユーザアンケート Q9.4 の得点	製品提供元において把握しにくい。その ため、ユーザアンケートにて簡易回答を 求めることとした。
・リスク回避性 ・環境リスク緩和 性 ・REn-1	・なし ・なし ・			今回の研究においては、製品提供元から のデータ提供が難しく見送りとした。
・利用状況網羅 性 ・利用状況完全 性 ・なし	・主要な目的以外 での製品利用の 有無	・主要な目的以外でこ の製品は利用される か	X'=各ユーザ回答群の X 平均 X=ユーザアンケート Q8.2 で Yes と答えたな ら 1, No なら 0	原典の利用状況網羅性メトリクスは、ユー ザの製品利用形態や利用シナリオを網羅 的に把握する必要があり、企業側でも記 録を持っていない懸念があり測定が困難 である。そのため、ユーザアンケートにて 簡易回答を求めることとした。
・利用状況網羅 性 ・利用状況完全 性 ・CCm-1	・なし ・なし ・			今回の研究においては、製品提供元から のデータ提供が難しく見送りとした。
・利用状況網羅 性 ・柔軟性 ・なし	・ユーザアンケー ト ・CFL0.1 ・非主要目的での 製品利用時 タス ク達成度合い(ア ンケート)	(この製品は柔軟な利 用が可能か) ・主要でない目的もこ の製品で達成できた か	X'=各ユーザ回答群の X 平均 X=ユーザアンケート Q8.2.1 の得点	原典の利用状況網羅性メトリクスは、ユー ザの製品利用形態や利用シナリオを網羅 的に把握する必要があり、開発元・製品 提供元において記録を持っていない懸 念があり測定が困難である。そのため、 ユーザアンケートにて簡易回答を求め ることとした。
・利用状況網羅 性 ・柔軟性 ・CFI-1	・なし ・なし ・			今回の研究においては、製品提供元から のデータ提供が難しく見送りとした。
・利用状況網羅 性 ・柔軟性 ・CFI-2	・なし ・なし ・			〃
・利用状況網羅 性 ・柔軟性 ・CFI-3	・なし ・なし ・			〃

② ユーザテスト（研究プロセス③）

表 3-3-1 に示すように利用時の品質メトリクスのうちで有効性、効率性に関するものはユーザテストの実施により測定可能な形で定義した。限られた時間内で効率よく主要な機能の正常系および異常系テストを実施するために、図 3-3-2 に示すユーザテスト実施のプロセスを定義した。具体的には、機能一覧および正常系テスト項目の定義については、当該製品に対する詳しい理解が不可欠なため製品提供元が実施し、それらを参照することで研究チーム（および外注先の評価機関）において当たり前機能抽出および異常系テスト項目の定義、ならびにテストの実施を担当した。

	研究チーム	製品提供元
1		機能定義
2	当たり前機能抽出(評価機関)	
3		当たり前機能網羅の正常系テスト項目定義
4	異常系テスト項目定義	
5	テスト実施	
6	テスト結果から品質測定	

図 3-3-2 ユーザテスト実施のプロセス

③ アンケート（研究プロセス①）

表 3-3-1 に示すように利用時の品質メトリクスのうちで有効性、効率性に関するものはユーザテストの実施により測定可能な形で定義したが、満足性、リスク回避性および利用状況網羅性に関するものについては日常の業務や活動において当該ソフトウェア製品を用いる利用者の実感に基づく測定が望ましいため、その測定のためのユーザアンケートの雛形を作成した。その内容を以下に示す。

回答方法：設問に対して、あなたがもっともふさわしいと感じる選択肢に「○」をつけてください。（ ）についてはお感じになられていることをできるだけ具体的にご記入ください。		
Q1	この製品に対する 満足度 はどのくらいですか	
	1: 満足していない	2. あまり満足していない
	4. 少し満足している	5. そこそこ満足している
		3. 少し満足していない
		6. 満足している

Q1.1	この製品をほかの人に薦めたいと思いますか
	1: 全く思わない ~ 5: どちらともいえない ~ 10: 非常にそう思う
Q2	この製品において実用上問題を感じることはありますか
	1: 頻繁に感じる 2. しばしば感じる 3. あまり感じない 4: 感じない
Q3	この製品の利用目的は達成できましたか
	1: できなかった 2. あまりできなかった 3. 少しできた 4: できた
この製品の各機能に関して以下の設問にお答えください	
Q4.1	当製品の機能「 <input type="text"/> 」を利用したことはありますか
	1. 毎日利用 2. 週2~3回利用 3. 月2~3回利用 4. 年数回利用 5. 利用したことはない
Q4.1.1	機能「 <input type="text"/> 」を利用したことがある人のみお答えください。機能Aの満足度はどのくらいですか?
	1: 満足していない 2. あまり満足していない 3. 少し満足していない 4. 少し満足している 5. そこそこ満足している 6. 満足している
Q4.2	当製品の機能「 <input type="text"/> 」を利用したことはありますか
	1. 毎日利用 2. 週2~3回利用 3. 月2~3回利用 4. 年数回利用 5. 利用したことはない
Q4.2.1	機能「 <input type="text"/> 」を利用したことがある人のみお答えください。機能Bの満足度はどのくらいですか?
	1: 満足していない 2. あまり満足していない 3. 少し満足していない 4. 少し満足している 5. そこそこ満足している 6. 満足している
この製品の 信用性 についてお聞かせください。 ここでいう製品の信用性とは、ユーザーが意図した通りの動作・結果を得られることの度合のことを意味します。	
Q5.1	この製品はあなたが意図した動作を行いますか
	1: そう思わない 2. あまりそう思わない 3. 少しそう思う 4: そう思う
Q5.2	この製品の動作の結果は、あなたの意図に則していますか
	1: そう思わない 2. あまりそう思わない 3. 少しそう思う 4: そう思う
Q5.3	この製品を利用するにあたり、セキュリティに関する問題(例えば情報漏えいや攻撃に関する危険、不正なアクセス権設定)を感じたことがありますか
	1: 頻繁に感じる 2. しばしば感じる 3. あまり感じない 4: 感じない
以下について、この製品を利用する際に感じる度合いをお聞かせください	
Q6.1	この製品を使うことに夢中になる
	1. 感じる 2. 少し感じる 3. あまり感じない 4. 感じない
Q6.2	この製品を使っていてイライラする
	1. 感じる 2. 少し感じる 3. あまり感じない 4. 感じない
Q6.3	この製品を使うことにより、創造性がかきたてられる

	1. 感じる	2. 少し感じる	3. あまり感じない	4. 感じない
Q6.4	この製品を使うのは難しい			
	1. 感じる	2. 少し感じる	3. あまり感じない	4. 感じない
Q6.5	この製品を使うことで業務に対するモチベーションが上がった			
	1. 感じる	2. 少し感じる	3. あまり感じない	4. 感じない
Q6.6	この製品を使っていて、使い方や機能に不安を感じる			
	1. 感じる	2. 少し感じる	3. あまり感じない	4. 感じない
この製品を使う際の疲れ具合について教えてください				
Q7.1	目に疲れは感じますか			
	1. 使用するたび感じる	2. たまに感じる	3. あまり感じない	4. 感じない
		5. 覚えていない		
Q7.1.1	どのような時に感じますか			
	()			
Q7.2	手や腕の疲れは感じますか			
	1. 使用するたび感じる	2. たまに感じる	3. あまり感じない	4. 感じない
		5. 覚えていない		
Q7.1.2	どのような時に感じますか			
	()			
利用状況・環境についてお答えください				
Q8.1	この製品の主な利用目的はなんですか			
	()			
Q8.2	上記以外の目的でこの製品を利用したことはありますか?ある場合は具体的にどんな目的で利用したか教えてください			
	()			
Q8.2.1	Q8.2でYesと答えた方のみお答えください。 Q8.2でお答えいただいた利用目的はこの製品でどのくらい達成できましたか			
	1. 達成できなかった	2. あまり達成できなかった	3. 少し達成できた	4. 達成できた
Q8.3	この製品を利用するときにマニュアルを利用のたびにみる必要があったなど追加学習が必要でしたか?			
	1. 必要だった	2. 少し必要だった	3. どちらかといえば必要だった	4. どちらかといえば必要ではなかった
		5. あまり必要ではなかった		6. 必要ではなかった
Q9	この製品を利用するにあたって、以下について感じたことがありますか? 感じたことがある場合、その頻度はどのくらいですか?			
	経済的損失(不正な財務状況の提示, 情報漏えい など)	1. 利用の度感じる	2. 週2~3回感じる	3. 月2~3回感じる
		4. 年数回感じる	5. 感じたことはない	

	健康への影響(頭痛や肩こり など)	1. 利用の度感じる 3. 月 2～3 回感じる 5. 感じたことはない	2. 週 2～3 回感じる 4. 年数回感じる
	人命や身体の危険 (カルテ登録ミスや誤ったシミュレーション結果を出力した など)	1. 利用の度感じる 3. 月 2～3 回感じる 5. 感じたことはない	2. 週 2～3 回感じる 4. 年数回感じる
	環境への影響 (異常量の二酸化炭素の排出, 消費電力の増加 など)	1. 利用の度感じる 3. 月 2～3 回感じる 5. 感じたことはない	2. 週 2～3 回感じる 4. 年数回感じる
Q-In-1	この製品のメニューやボタンなどのユーザーインターフェースについて使いやすいと思いますか		
	1. 感じる 2. 少し感じる 3. どちらともいえない 4. あまり感じない 5. 感じない		
Q-In-1.1	(Q-In-1 で 4 or 5 と答えた方のみお答えください) 具体的にどのような部分が使いにくいですか		
	A. レイアウト要素 (フォント, 画面 (ウインドウやパネル), ボタン, メニュー 等) ()		
	B. カスタマイズ要素 (色, サイズ, 配置&並び 等) ()		

④ スコア化・集約方式 (研究プロセス④)

製品品質 (内部品質・外部品質) と同様に, メトリクスの単位で測定値をパーセンタイルによりスコア化したうえで, スコアを品質特性・副特性単位で平均化する方式を実現し, 測定データから最終スコアを算出するツールをスプレッドシートとして実現した

3.3.3 発生した問題および今後の展望

(1) 発生した問題

ISO/IEC 25020 において定義される多くの利用時の品質メトリクスについて定義の抽象度が高く, そのままでは具体的な測定が困難であることが分かった. そこで, 初期案として一度具体化したうえで一つの製品について初期トライアルとして試験的な測定を実施し, その結果に基づく改訂を通じて段階的な具体化により解決した.

(2) 今後の展望

具体化できなかった (現実には測定困難な) 利用時の品質メトリクスについて継続検討する予定である. これには, 国際標準化機構 ISO に対する利用時の品質メトリクス自体の改訂の働きかけも含む. 同時に, ユーザアンケートおよびユーザテストの実施方法を含めて, 測定評価ツール・仕組みの利便性を向上させる予定である.

例えば, 表 3-3-1 における環境リスク緩和性に関するメトリクス REn-1 は環境上の様々想定しうるリスクを緩和できている程度を測定するものであるが, 現実には製品ごとに想

定しうるリスクは大きく異なり、かつ、製品提供元において現状として把握困難であるため、直接的な具体化は断念した。代わりに、環境リスク全般という形で簡易にユーザアンケートにより利用者に直接問う形式を採用した。

3.4 研究課題4「個々の品質実態把握」

3.4.1 当初の想定

(1) 研究内容

2年間で21のソフトウェア製品群を測定評価し、個々の品質単位の実態として測定評価結果の分布や平均値等の統計量をまとめる。測定評価の窓口をCSAJが務め、過去にPSQ認証制度を受けた組織を中心として協力企業を募る。実際の測定評価の大部分はPSQ認証制度における評価実績のある評価機関に外注し、測定評価の一部を研究チームおよび学生スタッフが担う。外注により得られた測定評価結果および研究チーム・学生スタッフの一部作業により得られた結果を統合し統計処理し、個々の品質の実態を可視化する。進め方としては、トライアルに1-2製品について研究チーム・学生スタッフが測定評価し、結果を検討する。続いて2016年度において20製品の測定評価を本運用として外注および研究チーム・学生スタッフが一部測定評価する。

PSQ認証制度は、2006年策定の品質モデルISO/IEC 9126-1に基づくISO/IEC 25051:2006に準拠した評価基準によって運用が開始された。その後、インターネットの普及に伴うセキュリティの重要性等を組み入れた品質モデルISO/IEC 25010に基づき改訂されたISO/IEC 25051:2014に準拠した評価基準を策定し運用しているため、過去にPSQ認証を取得した製品を再度、本研究において新たに品質測定評価する場合は、上記のようなソフトウェア品質のモデル・特性の時代変遷に伴う品質の実態の捉え方の変化を幾らか捉えられる可能性がある（ただし本研究の測定評価対象を、過去にPSQ認証制度を受けた製品に限定しない）。

(2) 想定問題と対応策

問題：単一または少数の開発組織の製品群を対象とすれば容易に測定評価を実施できるが、日本の代表的なソフトウェアの品質の実態を反映しない可能性がある。一方、多数の開発組織の製品群を、これまでの測定評価実績なしに新たに大規模に測定評価実施することは難しい。

解決策：CSAJを通じて2014年より20以上の製品群についてPSQ認証制度を実施済みである。そこでPSQ認証制度を拡張した仕組みとして、研究課題1～3に品質の総合的測定評価を21の製品群に対し実施する。具体的な測定評価の大部分は、PSQ認証制度において過去にISO/IEC 25051に基づき品質評価を担当した評価機関に委託することで、円滑な実施を可能とする。

3.4.2 研究プロセスと成果

(1) 研究プロセス

- ① PSQ認証制度を拡張した総合的品質測定評価制度を定義する。
- ② 製品を提供する協力企業を募集し対象製品を決定する。
- ③ 測定評価を実施する。

- ④ 統計処理と品質実態把握を実施する。

(2) 具体的な研究成果の内容

① 製品ごとの測定評価データ（研究プロセス①②③）

CSAJ の協力を得て、研究課題 1～3 において得られた各メトリクスを適用するにあたって PSQ 認証制度を過去に利用済みの組織および PSQ 認証制度をこれから利用する組織に対する製品募集が効率的であることを把握し、PSQ 認証制度を過去に利用済みの組織に対して募集をかけるとともに、これから PSQ 認証制度を利用する組織に向けて合わせて品実態調査を実施する制度を定義した（研究プロセス①）。そのうえで、CSAJ の協力を得て募集した結果、ソフトウェア製品を開発提供している 12 社の協力を得ることに成功し（研究プロセス②）、全 21 製品について、製品品質および利用時の利用品質の測定評価結果を得た（研究プロセス③）。測定評価にあたり、内部品質および外部品質の測定評価に必要な試験関係の情報収集の作業、ならびに、利用時の品質の測定評価に必要な利用者からのクレーム・問い合わせ状況の情報収集の作業やユーザテスト実施に必要な当たり前機能等の抽出作業について外部評価機関に外注した。他の作業は研究チームが、調査対象の製品提供元の協力を得ながら進めた。

各製品ごとに、研究チームに対して入力として提出されたデータによって測定評価できた品質特性が異なる。製品 2017P1～2017P21 について品質特性単位の評価スコアを表 3-4-1 に示す。表 3-4-1 にあるように、製品によって評価スコアが大きく異なることが分かる。

傾向を見るために、全 21 製品を評価スコアの平均の高い順に並べた場合の上位 1 位、7 位、14 位、21 位の各製品の評価スコアを図 3-4-1、図 3-4-2、図 3-4-3、図 3-4-4 にそれぞれ示す。スコアの高い品質特性は製品によって大きく異なることが分かる。

表 3-4-1 全 21 製品の品質特性単位の評価スコア一覧

製品ID	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	有効性	効率性	満足性	リスク回避性	利用状況網羅性	主特性平均
2017P1	NA	NA	NA	NA	0.25	NA	0.75	NA	NA	NA	NA	NA	NA	0.50
2017P2	0.50	0.42	0.63	0.39	0.39	0.00	0.30	0.52	0.31	0.75	0.50	0.50	0.50	0.44
2017P3	0.71	0.61	0.71	0.18	0.44	0.94	NA	0.40	NA	NA	0.83	1.00	0.75	0.66
2017P4	0.00	0.70	0.87	0.45	0.30	0.73	0.25	0.16	0.78	0.19	0.00	0.00	0.25	0.36
2017P5	NA	0.26	NA	0.23	0.19	0.75	0.12	NA	NA	NA	NA	NA	NA	0.31
2017P6	NA	0.39	NA	0.45	0.19	0.00	0.12	0.63	NA	NA	NA	NA	NA	0.30
2017P7	0.17	0.29	0.60	0.53	0.35	0.25	0.26	0.64	0.28	0.56	NA	NA	NA	0.39
2017P8	0.33	0.07	NA	0.27	0.13	0.00	0.25	0.25	0.09	0.56	NA	NA	NA	0.22
2017P9	0.50	0.38	0.87	0.21	0.28	0.25	0.47	0.17	0.19	0.44	NA	NA	NA	0.37
2017P10	0.34	0.44	0.50	0.27	0.18	0.31	0.38	0.25	0.63	0.25	NA	NA	NA	0.35
2017P11	NA	0.77	NA	0.32	0.06	0.00	0.12	NA	NA	NA	NA	NA	NA	0.26
2017P12	0.48	0.70	0.00	0.24	0.36	0.18	0.46	0.52	0.34	0.63	NA	NA	NA	0.39
2017P13	0.15	0.00	0.21	0.41	0.16	NA	0.26	0.34	NA	NA	NA	NA	NA	0.22
2017P14	0.33	0.00	0.00	0.24	0.20	0.75	0.16	0.17	0.31	0.13	NA	NA	NA	0.23
2017P15	0.31	0.00	0.00	0.27	0.35	0.75	0.07	0.16	NA	NA	NA	NA	NA	0.24
2017P16	0.43	0.00	NA	0.10	0.07	1.00	0.14	0.25	0.78	1.00	NA	NA	NA	0.42
2017P17	NA	0.26	NA	0.21	0.18	0.00	0.10	0.09	NA	NA	NA	NA	NA	0.14
2017P18	NA	0.19	NA	0.21	0.19	NA	0.12	0.32	NA	NA	NA	NA	NA	0.21
2017P19	NA	0.30	NA	0.21	0.30	0.00	0.46	0.09	NA	NA	NA	NA	NA	0.23
2017P20	NA	0.17	NA	0.21	0.11	0.00	0.21	0.02	NA	NA	NA	NA	NA	0.12
2017P21	NA	0.07	NA	0.21	0.06	0.00	0.12	0.02	NA	NA	NA	NA	NA	0.08

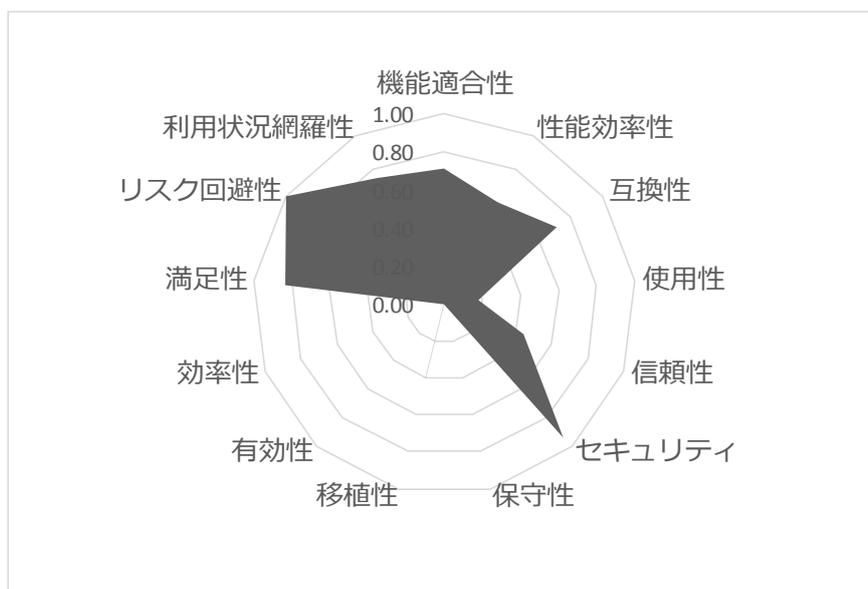


図 3-4-1 1 位の製品の品質評価スコア



図 3-4-2 7位の製品の品質評価スコア

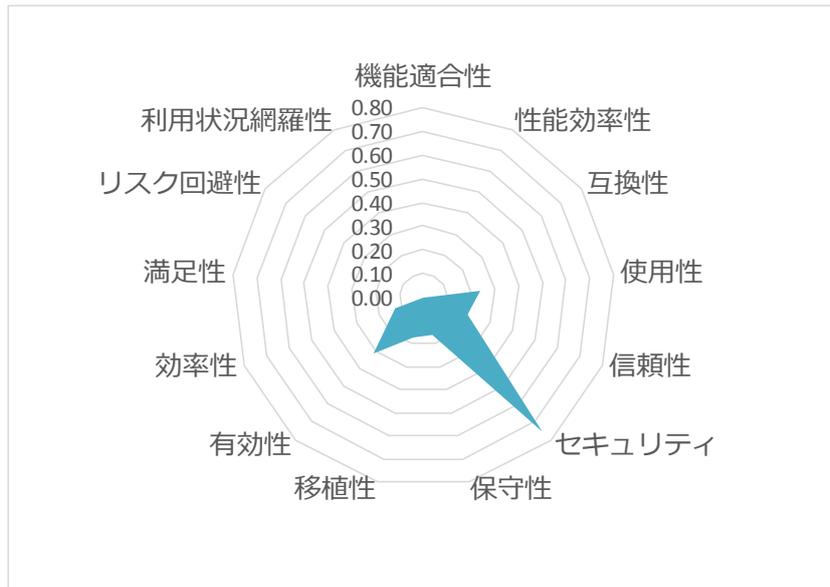


図 3-4-3 14位の製品の品質評価スコア



図 3-4-4 21 位の製品の品質評価スコア

② 製品ごとの品質診断レポート (研究プロセス③)

本調査研究においては、品質測定評価結果を製品ごとにまとめて、品質特性単位の傾向、実態における製品のポジション、ならびに、品質向上に向けた助言からなる品質診断レポートを作成し、製品開発元へ提供した (研究プロセス③)。特に、信頼性予測、プログラムソースコード解析結果、ならびに、ユーザテストの実施結果から判明した事柄、傾向、助言について詳細に品質診断レポートに含めた。

品質診断レポートの記述サンプルとして、信頼性・欠陥予測結果、各メトリクス測定状況、ソースコード解析結果、および、ユーザテスト実施結果の分析部分を中心に以下に示す。品質診断レポートの提供により、製品開発元における当該製品の以降の改訂における品質向上、ならびに、将来における他製品の品質要求定義および開発へと貢献することに成功した。

(1) 信頼性・欠陥予測結果

バグ票を基にソフトウェア信頼性曲線を描き、現状のバグ状況の評価を行った。以下にその結果を示す。すべてのデータを併せて、欠陥予測を行った。結果を図 (予測結果) に示す。

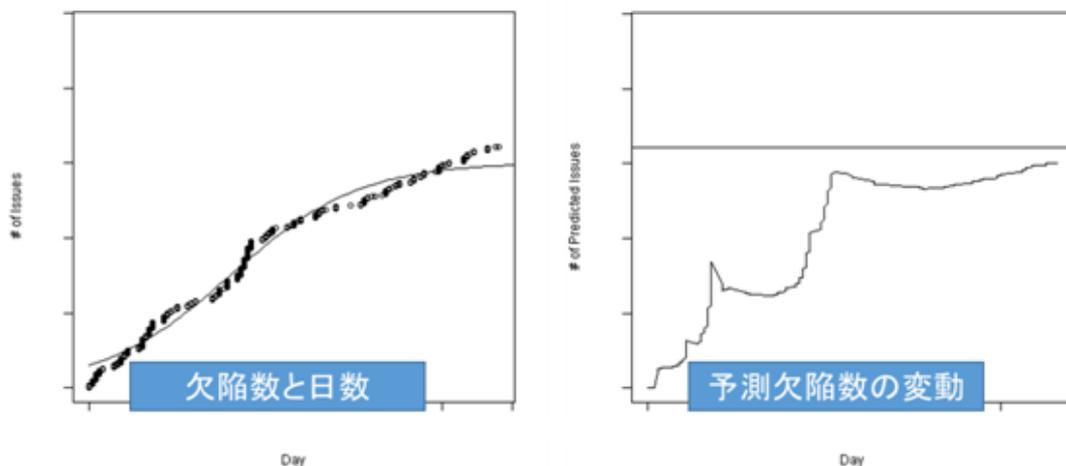


図 予測結果

発見された欠陥数は XXX 件であり，一方予測では YYY 件であった．予測欠陥数より多くの欠陥を発見しており，また予測欠陥数の変動もほぼ一定になっている．したがって，欠陥は収束していると判断する．以上の結果を表 1 予測結果に示す．

表 1 予測結果

対象	期間	発見欠陥数	予測欠陥数
全データ	20XX/XX/XX～20XX/XX/XX	XXX	YYY

表 1 予測結果からわかるように・・・については概ね十分な欠陥を発見できているが，バグや全データを見るとまだ欠陥は発見されることが読み取れる．よって，テストプロセスの見直しやテストケースの追加・コードレビュー等を行い，バグが完全に改修できているかを再考することを提案する．

(2) メトリクス測定状況

ISO/IEC-25022, 25023 の製品品質メトリクスについて，当該製品の機能内容や試験結果，試験内容などのヒアリングを通して各メトリクスを測定した．測定したメトリクスから得られる示唆について報告する．

(2-1) 良い傾向に関して

- ・性能試験での応答時間実測値が目標時間より短い

主機能である・・・について，応答時間の性能試験をしている．目標とした閾値の妥当性については研究チームでは判断しかねるものの，達成不能な目標を掲げていないことを確認できた．

- ・色覚，聴覚の障害を持つユーザに配慮した機能： XXX%

色や音のみで表現された情報を利用した機能項目が・・・と確認した。

・操作履歴が残る機能：XXX%

機能項目を操作した場合いずれも操作履歴が残ることを口頭で確認した。当該製品は・・・する製品であるため、不正なアクセスやアクセス傾向の分析を製品購入者が可能と考えられる。また、利用時の記録が残せるため、製品のユーザビリティテスト等を行った場合には被験者の振舞傾向を分析可能と考えられる。

・発見された不具合の修正率：XXX%

開発において、発見された不具合を全て修正してからリリースしていることを確認した。

(2-2) 改善余地に関して

・ユーザ用の説明動画がある機能：XXX%

機能項目の説明動画が・・・ことを確認した。・・・を増やすことが改善点として挙げられる。また当日の口頭確認通り・・・とすると、当該製品は・・・ソフトなので、・・・ことがユーザの満足度に繋がりがやすいと考えられる。

・自動化されたテストケース：0%

製品開発時のテストを・・・確認した。・・・ため、・・・できるとよい。・・・だけでなく、・・・テストを・・・ツールも近年では有償/無償ともに出てきている。

(3) ソースコードメトリクス複数間の分析

メトリクス ID として UM017 等を本文中では用いるが、これはソースコード解析ツール Understand で測定したメトリクスに研究室側で便宜的に割り当てた記号であり正規のものではない。メトリクス測定値は CSV ファイルとして付録に記載した。

(3-1) 分析結果について

分析 1：クラス結合度(UM017)はネスト最大値(UM096：クラス内関数のネスト)と実行コード行数(UM064：空白や宣言部分など除いた処理部分)との関係が強く、相関係数が NN を超えている。

推測されることは、他クラス複数と関わりを持つコアクラスの中にはネストの多い複雑なメソッドがあり、また担っている処理の多さが実行コード行数に表れていることである。そのため、開発チーム人材の移り変わりの際に、コアクラスのメソッド全体を把握できる人がいなくなる可能性があり、コアクラスの細分化検討の余地がある。その場合、「責務の分割」と呼ばれるリファクタリングをすることになる。

分析 2：クラス内のサイクロマティック複雑度総和(UM100)は WMC(Weighted Methods per Class)とも呼ばれ、クラスの担う責務の大きさを表す。UM100 はクラス内の public メソッド数(UM043)や実行コード行数(UM064)との関係が強く、相関係数が NN を超えている。

また、対象製品では public メソッドしか持っていないクラスが全体の約 NN%となっており、コード修正の際にはほぼ自由にクラスを結合させる。急な機能修正などに柔軟に対応できるものの、修正者によってはせっかく別々のクラスに任せている処理を互いに依存しあうような形にしてしまう可能性がある。また、既存 public メソッドを private 等に変える場合、根本的構造変更の検討が必要となり大変になる可能性がある。公開範囲設定に悩んだ開発者が public のまましておいたメソッドは、その意図を知らない他の開発者によっていつの間にか他クラスと結合されている場合があるからである。

総評：コアクラスの構造を簡略化して内容のわかりやすさを向上させる場合、複雑なメソッドの細分化をするとよい。しかし、他クラスに呼び出されているメソッドは下手に処理手続きを変えると不具合に繋がる可能性がある。そのクラス内で完結する処理を対象に絞ってメソッドの細分化をするとよい。そして、今後の新機能追加等の際には public 公開範囲でなくてもよいメソッドかどうかを検討するとよい。

(3-2) 分析の詳細について(スピアマン順位相関を使用)

Java クラス単位メトリクスについて、二種のメトリクス測定値分布の相関関係を分析した。相関係数の高さは二種のメトリクスの関係性の強さを表す。本調査では相関係数が 0.85 以上で強い関係を、0.70 以上でやや強い関係を表すとみなした。ただし、関係の強さだけが数値化されるので、原因は調査対象の背景情報を元に推測する必要がある。

ソフトウェアメトリクスは測定値が正規分布ではなくロングテール分布をとるものが多い。そのため、本調査では正規分布を前提としていないスピアマンの順位相関係数を計算した。

以下の図は相関行列グラフの一部抜粋である。詳細は付録に記載した。相関行列の右上側には統計情報、中央にはヒストグラム、左下側には散布図がある。星の数は p 値による統計の有意さを表す。星の左下にある数値は相関係数の値 (0.0~1.0) である。

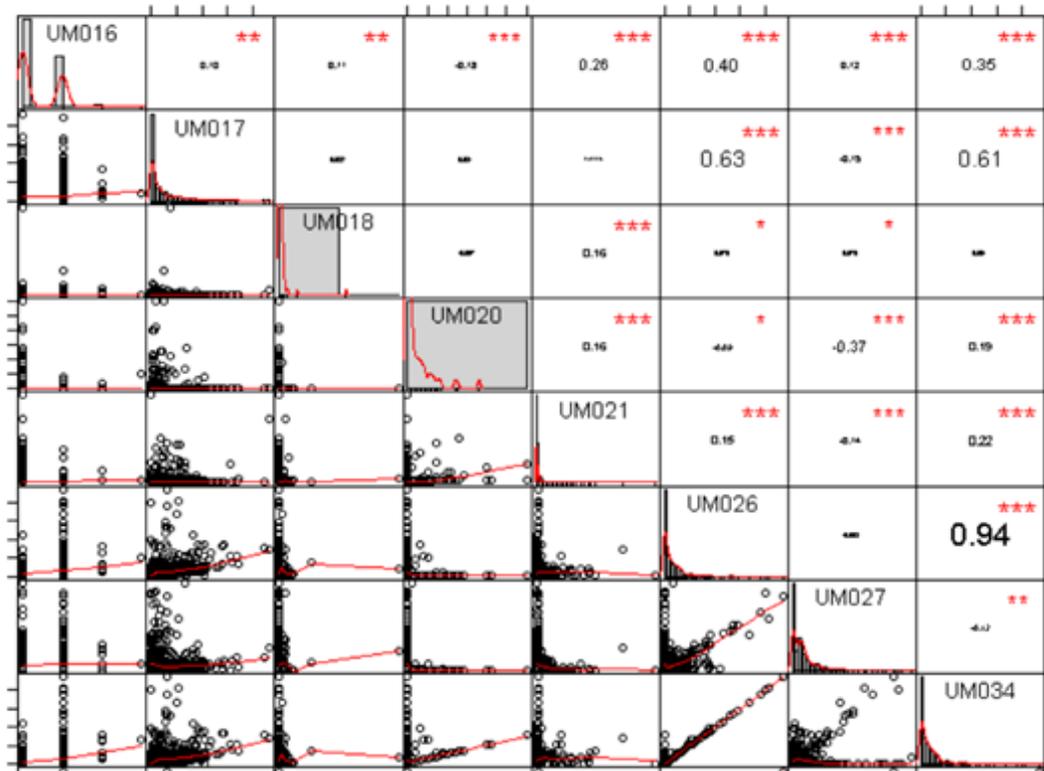


図 各メトリクス同士の相関行列グラフ（一部抜粋）

(3-3) コーディング規約(Google Java Style)による分析

コードの書き方には開発者ごとの癖があるため、ソフトウェア開発者が多かったり担当者の入れ替わりがあると既存のコードを読みづらく感じる人が出てきやすい。一定のコーディング規約に従ってコードを書くことで、開発チーム全体で見たコードの保守性低下に対処できる。（開発者個々人で見れば自分の慣れた書きの方が保守しやすい）

本調査では、Java ソースコードを対象にコーディング規約違反箇所の検出をした。違反検出に使用したのは Java 開発ツール Eclipse のプラグインである CheckStyle と、Google Java Style の規約セットである。CheckStyle で検出できる規約違反の種類は、変数等の名前付け方や括弧({ })や空白文字の使い方といった構文に関するものに限られる。プログラムの内容や適切さは取り扱い対象外となっている。

本調査では CheckStyle が出力するエラーメッセージの内容と Google Java Style の各構文ルールとの紐づけをした。対象製品のプログラムにおいて集計したのは、Google Java Style のどの構文ルールではエラーメッセージが出力されなかったか(OK だったか)である。以下の図は集計情報の一部抜粋である。詳細は付録に記載した。

Google's Java Style (GoogleのJavaコーディング規約)							
大項目	中項目	小項目	小々項目	番号	
						規約状況	規約違反 件数
2 Source f	2.1 File name					2.1	OK 0
2 Source f	2.3 Special characters	2.3.1	Whitespace characters			2.3.1	OK 0
2 Source f	2.3 Special characters	2.3.2	Special escape sequences			2.3.2	NG 1
2 Source f	2.3 Special characters	2.3.3	Non-ASCII characters			2.3.3	
3 Source file structure						3	
3 Source f	3.2 Package statement					3.2	
3 Source f	3.3 Imports	3.3.1	No wildcard imports			3.3.1	
3 Source f	3.3 Imports	3.3.2	No line-wrapping			3.3.2	
3 Source f	3.3 Imports	3.3.3	Ordering and spacing			3.3.3	
3 Source f	3.4 Class declarations	3.4.1	Exactly one top-level class declaration			3.4.1	
3 Source f	3.4 Class declarations	3.4.2	Class declarations	3.4.2.1	Overloads: never split	3.4.2.1	
4 Formatting	4.1 Braces	4.1.1	Braces are used where optional			4.1.1	
4 Formatting	4.1 Braces	4.1.2	Nonempty blocks: K & R style			4.1.2	
4 Formatting	4.1 Braces	4.1.3	Empty blocks: may be concise			4.1.3	
4 Formatting	4.2 Block indentation: +2 spaces					4.2	
4 Formatting	4.3 One statement per line					4.3	
4 Formatting	4.4 Column limit: 100					4.4	
4 Formatting	4.5 Line wrapping	4.5.1	Where to break			4.5.1	
4 Formatting	4.5 Line wrapping	4.5.2	Indent continuation lines at least +4 spaces			4.5.2	
4 Formatting	4.6 Whitespace	4.6.1	Vertical Whitespace			4.6.1	
4 Formatting	4.6 Whitespace	4.6.2	Horizontal whitespace			4.6.2	
4 Formatting	4.8 Specificity	4.8.2	Variable declarations	4.8.2.1	One variable per declaration	4.8.2.1	
4 Formatting	4.8 Specificity	4.8.2	Variable declarations	4.8.2.2	Declared when needed	4.8.2.2	

図 コーディング規約 (Google's Style) チェックの要約抜粋

対象製品は実際にはXXXXXXXX社独自のコーディング規約に従っていたので、一部のルールではGoogleコーディング規約上の違反数が多くなっている。下記のもの違反数が多く出ているものの、ルール内容自体を御社で重要と思わない場合は無視してもよい。

- ・クラスごとのインポート宣言文：他パッケージをインポートする際の順番。
- ・if文等の中括弧{ }を使う場合にどこで改行するか。
- ・Javadoc関連のルール：段落の間の改行の仕方。

また、関数内でローカル宣言された変数についての規約違反がNN件検出されている。変数宣言箇所と変数使用箇所が3ステートメントより離れていると違反になるが、その理由は関数内で使う変数を一挙に冒頭で宣言すると変数使用箇所を見つけにくくなる弊害を重視しているからである。既存の変数宣言箇所の移動はエラーにも繋がるため、無理に違反解消をせず、今後の新規コード開発時の注意にとどめておくのがよい。

次に、Google 規約によるものとはいえ改善指摘の参考となる規約違反について述べる。対象製品では、1 行につき 1 ステートメントがほぼ徹底されている。しかし、一行が NN 文字を超えるケースが多く検出されている。「・・・」等の文字が含まれている・・・行は除外した上での検出数なので、以下のようなステートメントが該当していると想定される。

- ・長い関数名やパラメータ名の多用。

<例>・・・

- ・メソッドチェーンの多用。

<例>・・・

- ・関数のパラメータで関数を何度も使っている場合。

<例>・・・

複数の内容の処理をひとつのステートメントとして表現している場合、そのステートメントの意味を記述者本人以外が理解しにくくなるリスクがある。ただし、上述した三つの例については極端に長くしている場合が問題なのであって使用を禁止しているわけではない。

(3-4) コーディング規約違反情報の発展的な利用について

付録には CheckStyle プラグインで検出した規約違反情報をまとめた xlsx ファイルがある。違反箇所がどのクラスのどの行にあるかが記録されている。御社内のソースコード変更履歴情報から、どのクラスをどのタイミングで誰が編集していたかを追跡可能であれば、いつ誰がどこに規約違反のコードを書きやすいかの分析が可能である。

ただし、これらの傾向を特定することが必ずしも開発効率の向上に繋がるとは限らないため、本調査では可能性の示唆にとどめる。

(4) ユーザテスト結果の分析

本調査では学生を被験者として対象製品のユーザテストを実施した。被験者が実施したタスクセットの内訳は御社が用意した正常系タスク NN 件と、研究室側で用意した異常系タスク NN 件となっている。

タスク実施時の記録は被験者が自筆で行っており、{経過時間, 遭遇エラー数, 無駄でないアクション数, 無駄だと感じたアクション数}を記録した。アクション数についてはユーザの操作が画面に反映されるような行動を基準に数えた。(マウスマウスカーソルを動かす程度のものは含めないが、ボタンクリックなどは含めた)。

以下の図はユーザテスト実施時の記録シートの抜粋であるが、詳細は付録に記載した。

ID	タスク	完了可否	時間(分)	エラー数	エラー理由	アクション数	無駄なアクション
1		完了	0:06			14	4
2-1		完了	0:08			20	16
2-2		完了	0:02			7	1
3-1		完了	0:03			15	6
3-2		完了	0:02			4	0
4-1		完了	0:03			10	6
4-2		完了	0:02			7	3
4-3		完了	0:01			4	0
4-4		未実施	0:00				

図 ユーザテスト実施記録（一部抜粋）

(4-1) 正常系タスクについての考察

ユーザテスト当日には、立ち合い参加していただいた御社社員の方から操作説明を適宜受けていた。タスク実施中に操作方法がわからなくなった際に学生から質問をしていた。途中までは学生が自力で考える時間が長かったため、完了までにかかった時間とアクション数が多いタスクがいくつかある。これはエラーが発生しているタスクにおけるアクション数の多さに表れている。

一方で操作説明を受けたタスクにおいてはアクション数が少なく済んでいる。そのため、説明役がいても初心者が理解できないほどの難しさの機能は今回のタスクで取り上げたエンドユーザ向け機能には含まれていなかったと考えられる。

(4-2) 異常系タスクについての考察

異常系タスクについては、ほとんどがNN分以内で確認が完了した。しかし、・・・しようとした場合に不自然な挙動が確認された。

・・・に時間がかかったのだが、・・・しまった。勘違いの原因となりうるため、このような場合は・・・がエラーメッセージとして表示されるのが望ましい。

(4-3) タスク実施中に発生したエラーの考察

・・・のタスク実施中にエラーとしてカウントしたNN件は、・・・手順を被験者が勘違いしていたためにエラーと思ったものだった。マニュアルを熟読する前に手探りでブラウザ操作をしていたことに起因する。

・・・しようとしてできなかったエラーの原因は、ユーザテスト時に・・・が理由だった。別の設定に切り替えて当日は対処した。

・・・いなかったエラーの原因は、・・・していないことが理由だった。これについては当日の御社からの指摘で判明した。

③ 品質特性の傾向（研究プロセス④）

品質特性別の評価スコアの傾向をヒストグラムとして図 3-4-5～図 3-4-17 に示す。図に

見て取れるように品質特性ごとに傾向が大きく異なっており、特徴的な傾向を以下にまとめる。

- 機能適合性：分布としてなだらかに広がり、製品によって機能適合性の高さが異なることが分かる。
- 性能効率性：分布としてなだらかに広がり、製品によって性能効率性の高さが異なることが分かる。
- 互換性：測定評価できた製品数が限られているが、その中でスコアの高低において2極化している傾向にある。データ交換などの互換性に通じる仕組みを一部の製品において考慮していないことが原因として挙げられる。また、国際規格 SQuaRE シリーズにおける定義から具体化できたメトリクス数がわずかに二つにとどまったため、一つのメトリクスの測定値が品質特性のスコアへ多大な影響を及ぼすこととなった。この品質特性は、ISO/IEC 9126 から国際規格 SQuaRE シリーズへと改訂される際に主特性として格上げされたものであり、国際規格側において今後の実効性のあるメトリクスの拡充が望まれる。
- 使用性：取りうる値の範囲内でスコアが低いほうに製品が集中しており、測定評価できた製品群において使用性を十分に考慮できていない、あるいは、エンドユーザ対象ではないといったことから意図的に考慮していない製品が多いことが分かる。
- 信頼性：取りうる値の範囲が狭く、全体的に同程度のスコアを取っている。機能適合性とあわせて、いわゆる「品質」として高い信頼性を作り込みの上で製品をリリースしていることがうかがえる。
- セキュリティ：スコアの高低について2極化している傾向にある。暗号化や破損防止などの高セキュリティ化に通じる仕組みを一部の製品において考慮していないことが原因として挙げられる。
- 保守性：取りうる値の範囲内でスコアが低いほうに製品が集中しており、測定評価できた製品群において保守性を十分に考慮できていない、あるいは、エンドユーザ対象ではないといったことから意図的に考慮していない製品が多いことが分かる。
- 移植性：分布としてなだらかに広がり、製品によって移植性の高さが異なることが分かる。
- 有効性：測定評価できた製品数が限られているが、その中でスコアの高低において2極化している傾向にある。ユーザテスト実施時にタスク実行に難がありタスクを達成しにくい製品が一部見られた。
- 効率性：測定評価できた製品数が限られているが、その中でスコアの高低において2極化している傾向にある。ユーザテスト実施時にタスク実行に難がありタスクの実施効率が低く時間を要する製品が一部見られた。
- 満足性、リスク回避性、利用状況網羅性：測定評価できた製品数が3製品と極めて限られ、意味のある傾向を得ることができなかった。

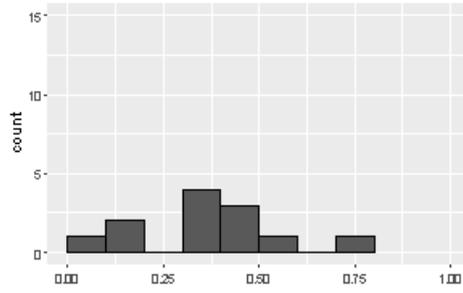


図 3-4-5 機能適合性の傾向

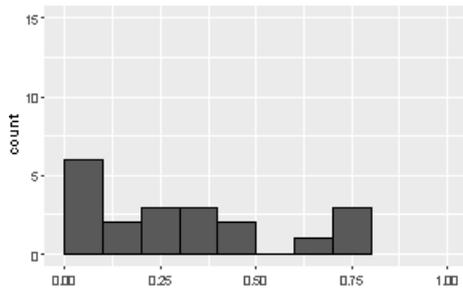


図 3-4-6 性能効率性の傾向

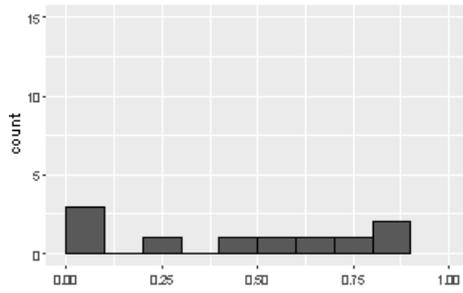


図 3-4-7 互換性の傾向

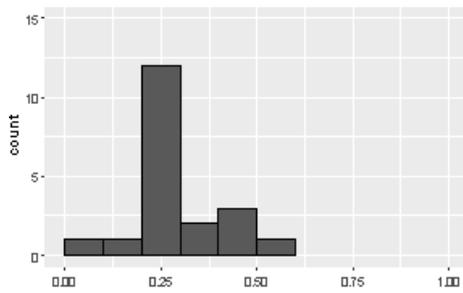


図 3-4-8 使用性の傾向

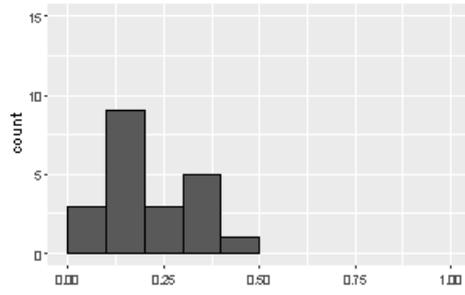


図 3-4-9 信頼性の傾向

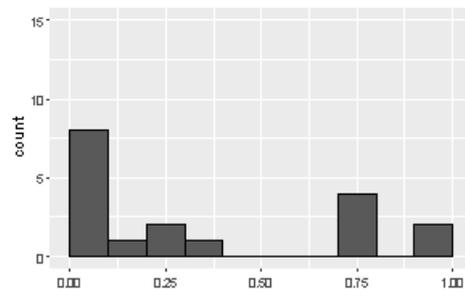


図 3-4-10 セキュリティの傾向

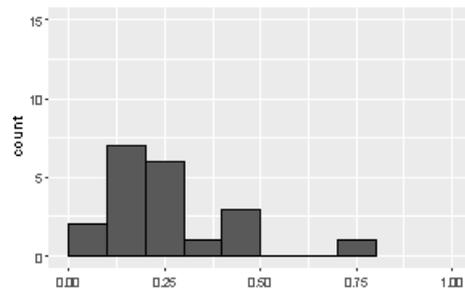


図 3-4-11 保守性の傾向

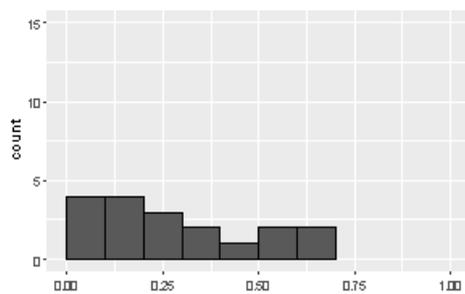


図 3-4-12 移植性の傾向

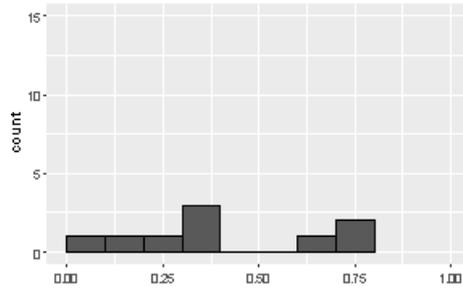


図 3-4-13 有効性の傾向

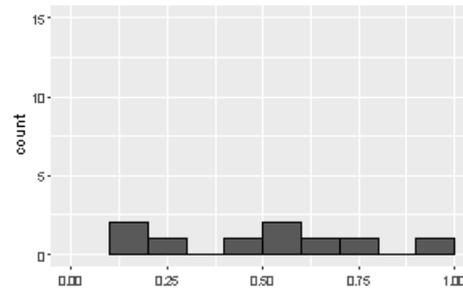


図 3-4-14 効率性の傾向

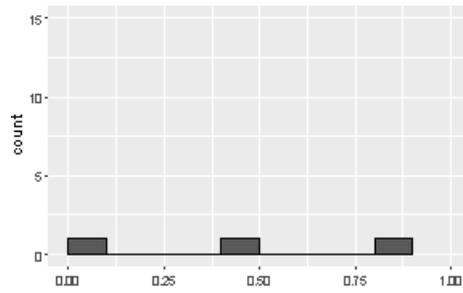


図 3-4-15 満足性の傾向

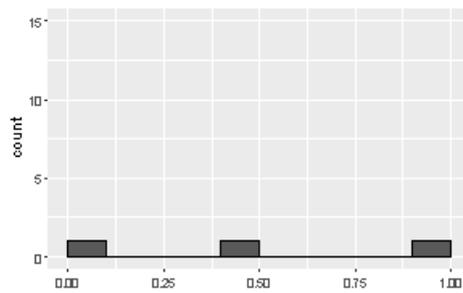


図 3-4-16 リスク回避性の傾向

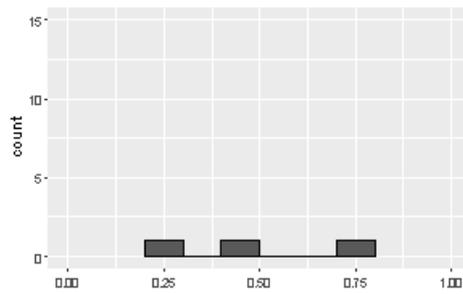


図 3-4-17 利用状況網羅性

④ 信頼性モデルによる信頼性傾向と予測（研究プロセス③④）

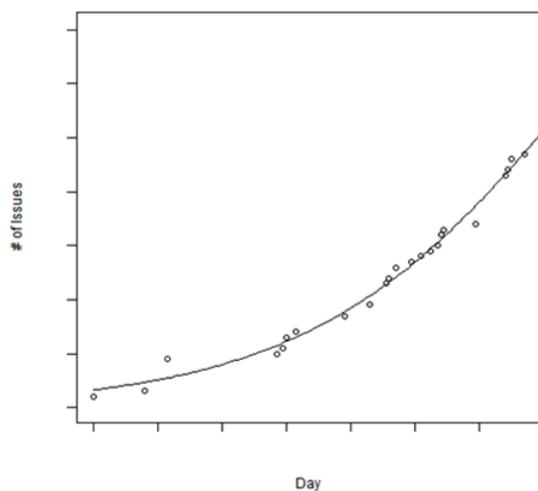
表 3-4-2 ソフトウェア製品の予測欠陥数と増加タイプの一覧に，製品 ID とソフトウェア信頼性モデルから得られた予測欠陥数から欠陥票から得られた欠陥数を引いた値，予測欠陥数の変動タイプを記載した．それぞれの製品の詳細については以下に記す．個々の信頼性傾向の把握は研究プロセス③の一部に該当し，製品群の変動タイプ一覧に基づく全体的な傾向の把握は研究プロセス④の一部に該当する．

表 3-4-2 ソフトウェア製品の予測欠陥数と信頼性タイプの一覧

製品 ID	予測欠陥数 - 欠陥数	タイプ
2017P1	78	爆発
2017P3	-21	安定
2017P4	270	漸増
2017P7	9	漸増
2017P9	4	漸増
2017P10	1	安定
2017P13	21840	爆発
2017P14	148	爆発
2017P15	-2	安定

2017P1： 図 3-4-18 2017P1 の欠陥数と信頼性モデルと予測欠陥数の変動，および表 3-4-2 ソフトウェア製品の予測欠陥数と信頼性タイプの一覧から，予測欠陥数が発見された欠陥数を上回り，予測欠陥数の変動については爆発タイプであることがわかる．これらのことから，今後も爆発的に欠陥数が増加することが考えられる．

欠陥数と予測モデル



予測欠陥数の変動

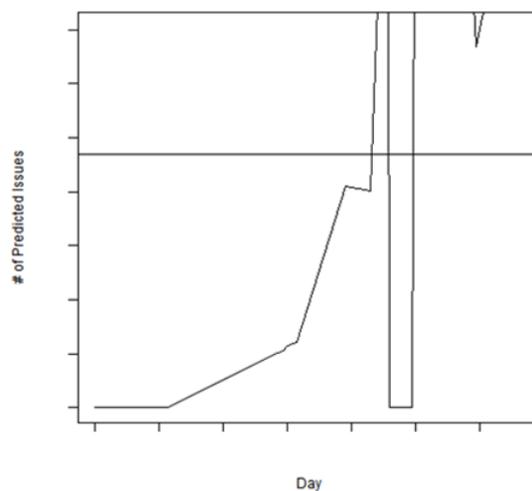
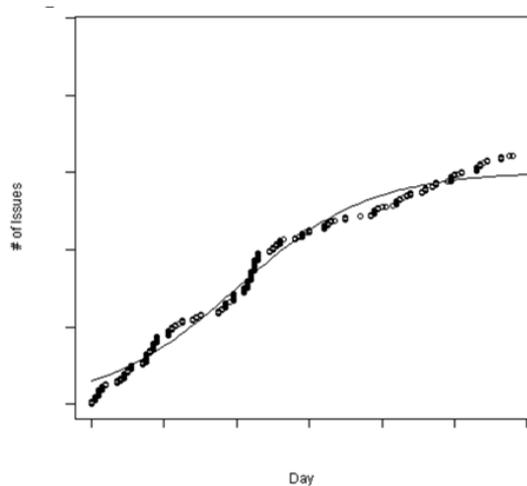


図 3-4-18 2017P1 の欠陥数と信頼性モデルと予測欠陥数の変動

2017P3： 図 3-4-19 2017P3 の欠陥数と信頼性モデルと予測欠陥数の変動， および表 3-4-2 ソフトウェア製品の予測欠陥数と信頼性タイプの一覧から， 予測欠陥数が発見された欠陥数を下回り， 予測欠陥数の変動については安定タイプであることがわかる． これらのことから， 欠陥は十分に発見されていることが考えられる．

欠陥数と予測モデル



予測欠陥数の変動

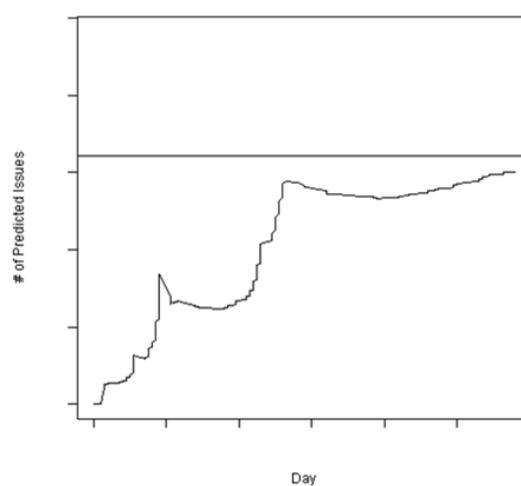


図 3-4-19 2017P3 の欠陥数と信頼性モデルと予測欠陥数の変動

2017P4： 図 3-4-20 2017P4 の欠陥数と信頼性モデルと予測欠陥数の変動， および表 3-4-2 ソフトウェア製品の予測欠陥数と信頼性タイプの一覧から， 予測欠陥数が発見された欠陥数を上回り， 予測欠陥数の変動については漸増タイプであることがわかる． これらのこと

から、今後も漸増的に欠陥数が増加することが考えられる。

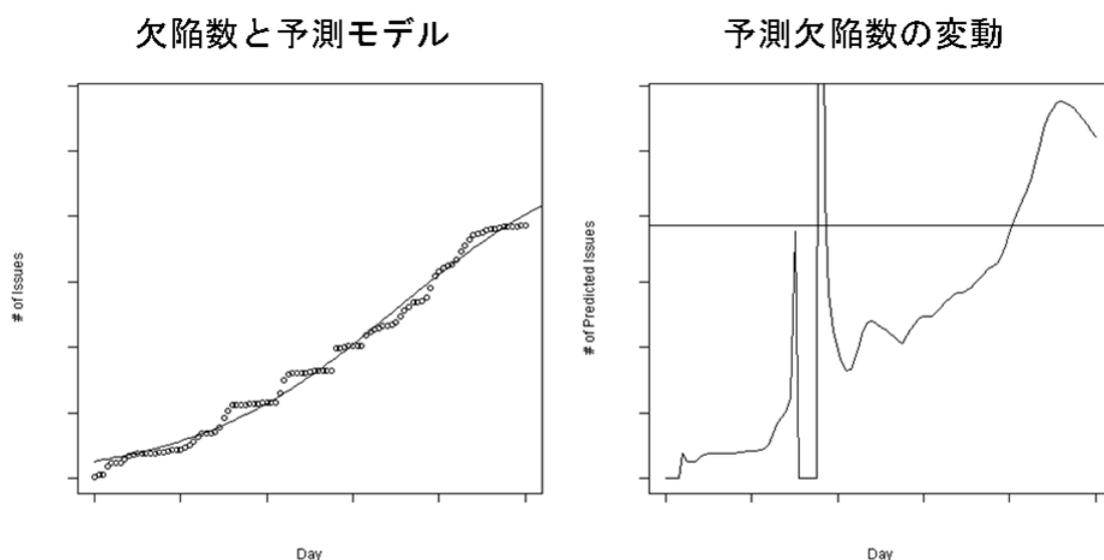


図 3-4-20 2017P4 の欠陥数と信頼性モデルと予測欠陥数の変動

2017P7： 図 3-4-21 2017P7 の欠陥数と信頼性モデルと予測欠陥数の変動，および表 3-4-2 ソフトウェア製品の予測欠陥数と信頼性タイプの一覧から，予測欠陥数が発見された欠陥数を上回り，予測欠陥数の変動については漸増タイプであることがわかる．これらのことから，今後も漸増的に欠陥数が増加することが考えられる。

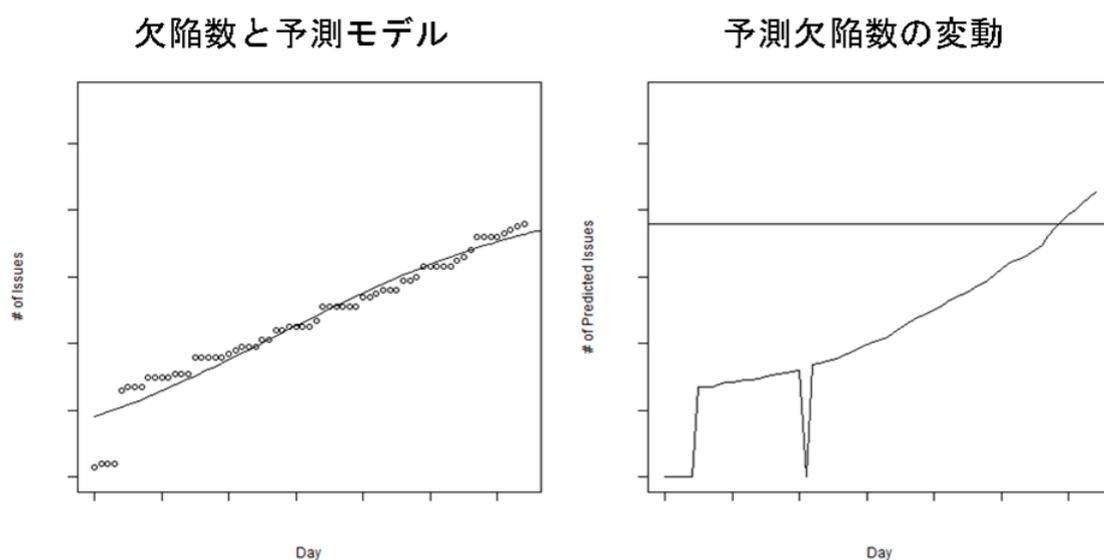


図 3-4-21 2017P7 の欠陥数と信頼性モデルと予測欠陥数の変動

2017P19： 図 3-4-22 2017P9 の欠陥数と信頼性モデルと予測欠陥数の変動，および表 3-4-

2 ソフトウェア製品の予測欠陥数と信頼性タイプの一覧から、予測欠陥数が発見された欠陥数を上回り、予測欠陥数の変動については漸増タイプであることがわかる。これらのことから、今後も漸増的に欠陥数が増加することが考えられる。

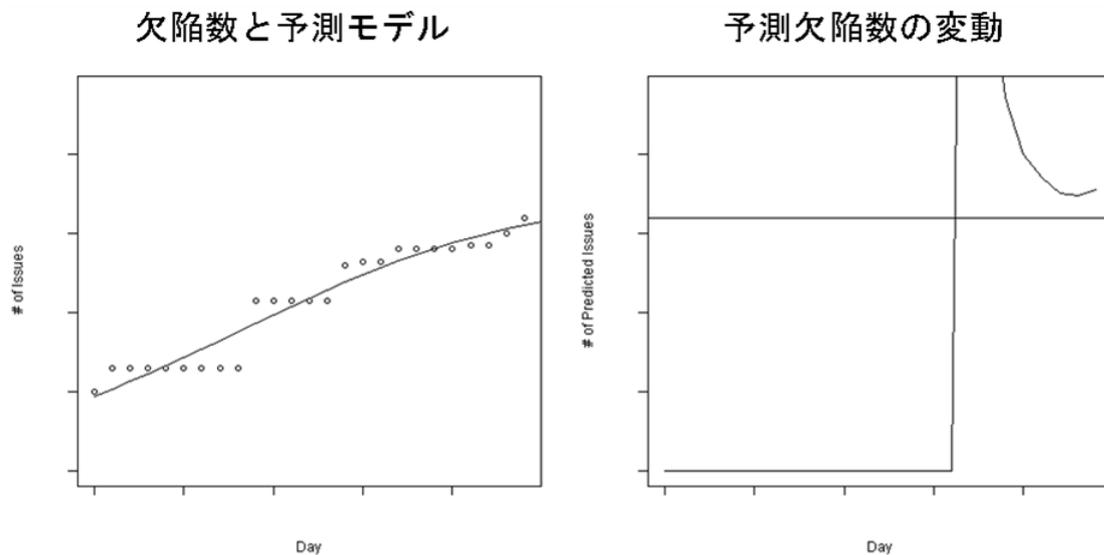


図 3-4-22 2017P9 の欠陥数と信頼性モデルと予測欠陥数の変動

2017P10： 図 3-4-23 2017P10 の欠陥数と信頼性モデルと予測欠陥数の変動，および表 3-4-2 ソフトウェア製品の予測欠陥数と信頼性タイプの一覧から、予測欠陥数が発見された欠陥数を上回っているが、予測欠陥数の変動については安定タイプであることがわかる。これらのことから、欠陥は十分に発見されていることが考えられる。

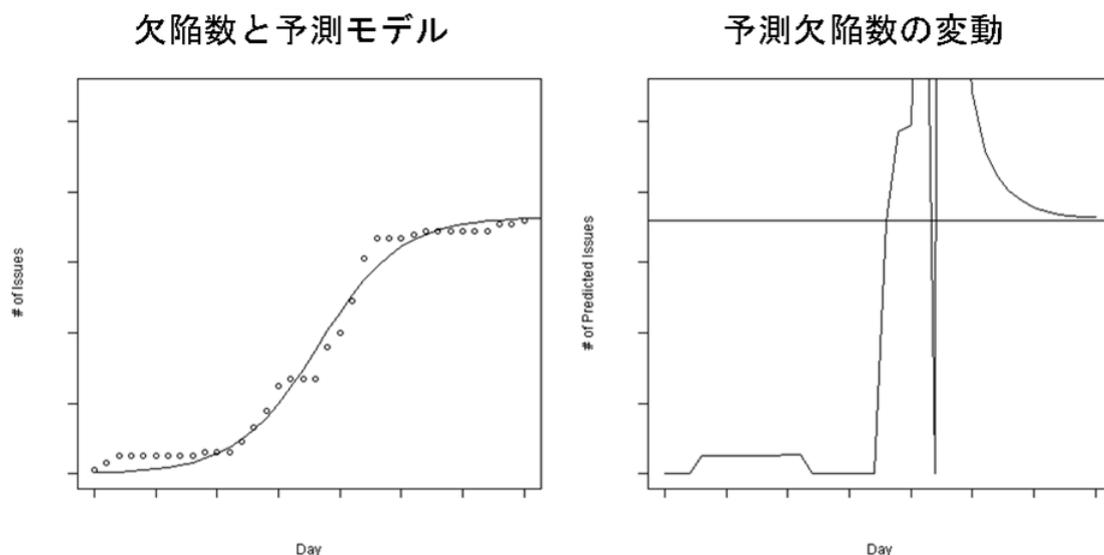


図 3-4-23 2017P10 の欠陥数と信頼性モデルと予測欠陥数の変動

2017P13： 図 3-4-24 2017P13 の欠陥数と信頼性モデルと予測欠陥数の変動，および 3-4-2 ソフトウェア製品の予測欠陥数と信頼性タイプの一覧から，予測欠陥数が発見された欠陥数を上回り，予測欠陥数の変動については爆発タイプであることがわかる．これらのことから，今後も爆発的に欠陥数が増加することが考えられる．

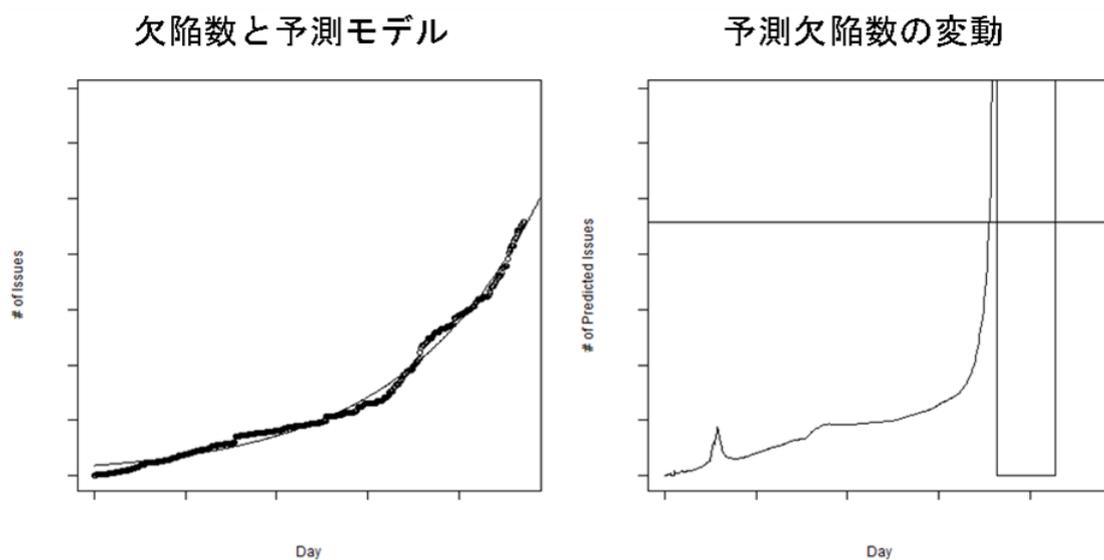
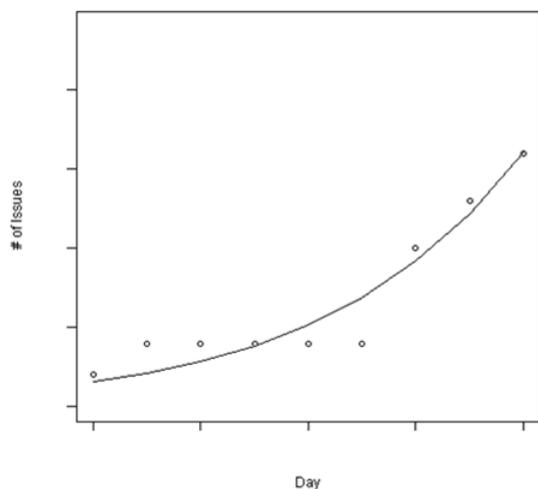


図 3-4-24 2017P13 の欠陥数と信頼性モデルと予測欠陥数の変動

2017P14： 図 3-4-25 2017P14 の欠陥数と信頼性モデルと予測欠陥数の変動，および 3-4-2 ソフトウェア製品の予測欠陥数と信頼性タイプの一覧から，予測欠陥数が発見された欠陥数を上回り，予測欠陥数の変動については爆発タイプであることがわかる．これらのことから，今後も爆発的に欠陥数が増加することが考えられる．

欠陥数と予測モデル



予測欠陥数の変動

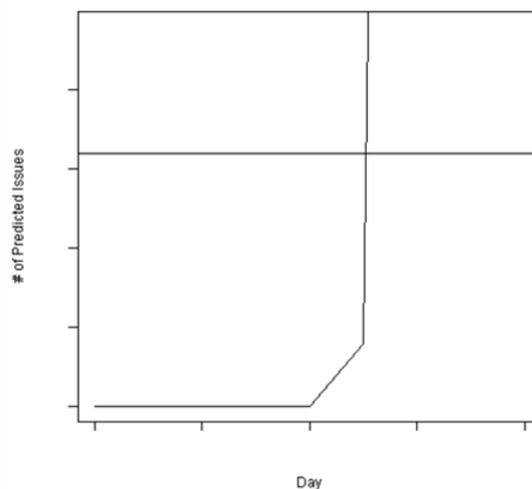
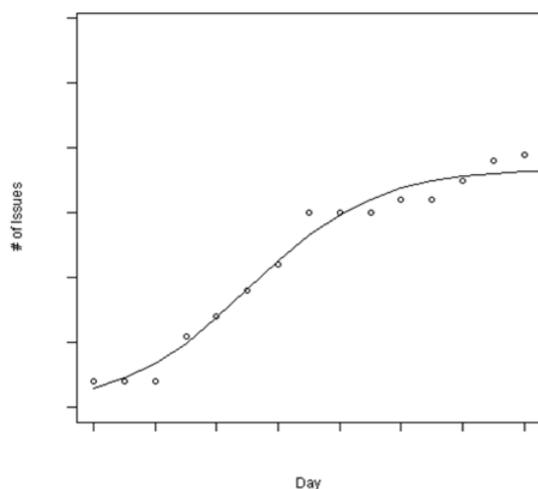


図 3-4-25 2017P14 の欠陥数と信頼性モデルと予測欠陥数の変動

2017P15： 図 3-4-26 2017P15 の欠陥数と信頼性モデルと予測欠陥数の変動，および表 3-4-2 ソフトウェア製品の予測欠陥数と信頼性タイプの一覧から，予測欠陥数が発見された欠陥数を下回り，予測欠陥数の変動については安定タイプであることがわかる．これらのことから，欠陥は十分に発見されていることが考えられる．

欠陥数と予測モデル



予測欠陥数の変動

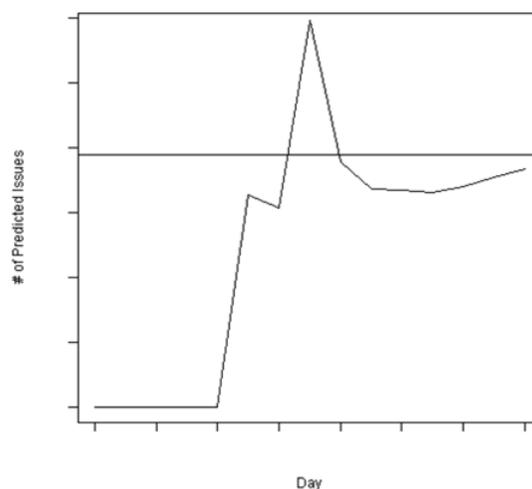


図 3-4-26 2017P15 の欠陥数と信頼性モデルと予測欠陥数の変動

3.4.3 発生した問題および今後の展望

(1) 発生した問題

当初に想定し 1 年目（2015 年）に設計実現した品質測定評価の方法においてはソフトウ

ウェア製品提供元（ソフトウェアベンダ）の負担が大きく、2年間で当初計画していた40製品を品質測定評価することが困難な見通しとなった。そこで2年目（2016年）において、もともとの想定における製品提供元の作業を外部評価機関に委託することで、申請者側の負担を軽減し、多くの製品を品質測定評価対象とすることに成功した。

なお、外部評価機関への委託内容が増大した関係で製品あたりの評価作業外注費が増大し、委託研究の全体予算のうちで外注費は50%未満に抑えるというという制約の都合上、最終的な品質測定評価対象数は21製品となった。品質の実態を明らかとする上で、一定の特徴を把握するにあたっては十分な製品数を確保することができた。

品質測定評価にあたり、前述のように内部品質・外部品質のメトリクスについて測定率は66メトリクス×21製品のうちで34%であった。未測定の原因の多くは根拠となるデータの未記録のためであり、これについては、より大規模な品質測定評価に向けて各製品開発におけるデータの記録が課題である。さらにデータ収集に必要な記入様式に説明が不十分な点があったため、製品提供側が記入した幾つかのデータの解釈にぶれが生じ、比較するデータとしては適さず未測定とせざるを得なかったことも影響している。記載内容についての解釈が容易となるよう、補足説明等を加えることを今後検討する。また同じく前述のように、利用時の品質のメトリクスについて測定率は17メトリクス×21製品のうちで24.4%に留まった。未測定の原因の多くは、製品提供元における都合等の理由により以下に示すユーザテストおよびユーザアンケートを必ずしも実施できなかったことにある。より大規模な品質測定評価に向けて、各製品開発における標準的な方法に従ったユーザテストおよびユーザアンケートの実施が課題である。

(2) 今後の展望

CSAJとの連携を継続し、品質実態調査を継続することでデータ数を増やす予定である。

3.5 研究課題5「品質間の関係分析」

3.5.1 当初の想定

(1) 研究内容

21のソフトウェア製品群における内部品質、外部品質、利用時の品質間の関係を統計分析し、品質間の関係として導出する。関係そのものおよび関連する各種の値（例えば二つの集約後の品質評価結果間の相関係数など）を合わせて、日本の代表的製品群の品質の総合的な実態としてまとめる。

(2) 想定問題と対応策

問題：規模やドメイン、品質要求の厳しさの異なる様々なソフトウェア製品を対象とする結果として、単純に全ての製品群の品質間の関係や品質特性を集約した結果を分析しようとしても明確な関係を得にくい可能性がある。

解決策：必要に応じて、規模やドメイン等の違いに応じた層別分析を試みる。また、二つの品質の測定評価結果間の相関分析や複数の品質の測定評価結果に基づく他の品質の回帰分析のような「細かく鋭敏」な分析に加えて、特定の特徴で大別した2群に対する比較検定のような「粗くロバスト」な分析も実施し、両面から品質間の関係を総合的に捉える。さら

に、個々の品質特性単位による関係の分析と、集約した結果による関係の分析の両方を実施する。

3.5.2 研究プロセスと成果

(1) 研究プロセス

- ① 品質間関係を、直接的な相関、信頼性タイプとの関係、コンテキスト別の傾向のそれぞれの観点で分析する。
- ② 得られた結果をまとめる。

(2) 具体的な研究成果の内容

① 品質間関係分析結果（研究プロセス①）

1) 品質評価・スコアの相関分析および分析結果

製品品質（内部品質および外部品質）と利用時の品質の全ての品質特性間について、全21製品のうちで測定評価できたものについて品質評価スコアの相関を分析した。スピアマンの順位相関係数を図3-5-1に、散布図を図3-5-2にそれぞれ示す。また図3-5-1において、相関分析におけるp値が0.10未満であり統計的に一定程度有意であると認められた箇所を点線の枠で印をつけている。

相関分析によって、得られているデータの範囲において観察しうる事柄を以下にまとめる。

- 信頼性が高いほど、性能も高い傾向にあった。信頼性と性能は、製品品質の中で主にテスト・実行によって測定評価される利用者向けの外部品質ということができ、近い関係にある品質特性と捉えることができる。ただしp値は0.10以上であるため、得られた限られた結果からは統計的に有意ということはできない。加えて、信頼性が高いほど保守性や移植性が高い傾向にあり、かつ統計的に有意と認められた。これは高信頼が求められる製品において、長期間における保守や様々な環境への移植や適合が求められるために正の相関が見られた可能性がある。
- 性能が高いほど、互換性も高い傾向にあった。この正の相関の原因としては、互換性が高いということは互換させる相手の製品群が存在するということであり、したがって当該領域が成熟してきている中で競争優位となるために高性能化が図られているということが推測できる。ただしp値は0.10以上であるため、得られた限られた結果からは統計的に有意ということはできない。これらの推測について今後の追加検証が必要である。
- 移植性が高いほど、使用性や信頼性および効率性が高いが、有効性は低い傾向にあった。移植性と信頼性の正の相関の原因としては、様々な環境に対する移植のしやすさを検討および作り込む過程において、様々な観点や側面から当該製品の品質を確認することとなり、結果として、あるいは、あわせて高い信頼性の作り込みにつながったということが推測できる。また、移植性と有効性の負の相関については、高い移植性を重視した結果として一つ一つの機能やタスク実行がしにくいものになってしまったという副作用の可能性と、逆に、有効性を重視してタスク実行をしやすいものを考慮した結果として優先度が低く設計・実装の複雑化を招きかねない移植性は強く考慮されなくなっ

たという可能性の両方が推測できる。ただし移植性と有効性との相関について p 値は 0.10 以上であるため、得られた限られた結果からは統計的に有意ということとはできない。これらの推測について今後の追加検証が必要である。

- 機能適合性が高いほど、使用性が低い傾向にあった。この機能適合性と使用性の負の相関について、(いわゆるアジャイル開発の考え方の対極として)当初計画通りの機能仕様の満足度を最重視した結果として、顧客にとって本来重要な製品としての使いやすさを損なってしまったという副作用の可能性や、逆に使いやすさを重視した結果として一部の機能を満足できなかったという可能性、あるいは、そもそもエンドユーザ向けの製品ではないといった理由から使いやすさを軽視していたといった可能性が推測できる。ただし、これらの推測について今後の追加検証が必要である。
- セキュリティが高いほど、有効性が高い傾向にあった。これは製品によってはセキュリティ関連の機能(ログインなど)が一定割合を占め、それらをユーザテストにおいて正確に実行できたことが幾らか影響している可能性があるが、引き続きデータの拡充と追加検証が必要である。

	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	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	0.31	0.19	-0.72	0.37	-0.05	0.50	0.31	-0.14	0.52	1.00	1.00	1.00
性能効率性		0.44	0.24	0.36	-0.17	0.37	0.32	0.32	-0.10	-0.50	-0.50	-0.50
互換性			0.04	0.17	-0.06	0.36	-0.04	-0.14	0.05	-0.50	-0.50	-0.50
使用性				0.17	-0.21	0.11	0.44	-0.09	-0.20	-1.00	-1.00	-1.00
信頼性					0.30	0.41	0.45	-0.08	0.11	1.00	1.00	1.00
セキュリティ						-0.06	0.19	0.64	-0.34	0.50	0.50	0.50
保守性							0.26	-0.29	0.01	1.00	1.00	1.00
移植性								-0.21	0.67	0.50	0.50	0.50
有効性									0.03	-1.00	-1.00	-1.00
効率性										1.00	1.00	1.00
満足性											1.00	1.00
リスク回避性												1.00

図 3-5-1 品質特性単位の品質評価スコア間の相関係数

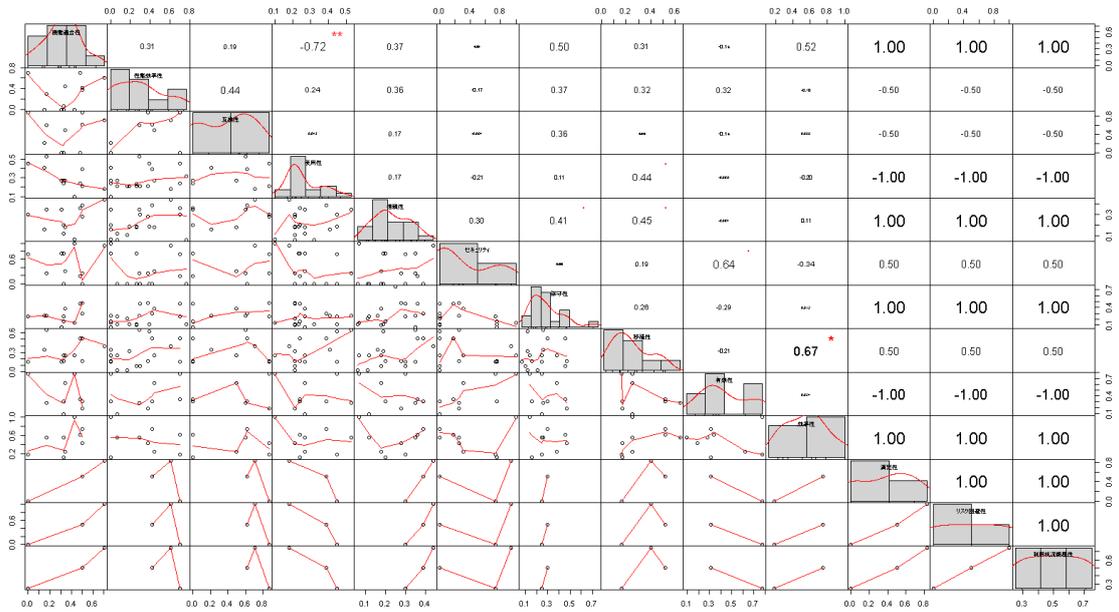


図 3-5-2 品質特性単位の品質評価スコアの散布図

2) 信頼性タイプと品質特性の関係

研究課題 2「外部品質の測定評価方法の確立」で取り上げた、信頼性モデルによる信頼性傾向と予測から得られた信頼性タイプ（安定、漸増、爆発）と、品質特性の主特性との関係について分析した結果を示す。

機能適合性、信頼性、有効性について安定タイプにおいて高品質であることがわかった。これは、機能適合性、信頼性、有効性が高いソフトウェア製品については、十分にテストされ欠陥を発見していると考えられる。性能効率性、互換性については爆発タイプにおいて低品質であることがわかった。これは、性能効率性、互換性が低いソフトウェア製品については、テストで十分に欠陥を発見できておらず、今後も欠陥が発見される可能性が高いと考えられる。他の特性については、顕著な違いは見られなかった。これは対象とした製品数の数が少ないことも原因と考えられる。

詳細については以下に示す。

機能適合性：図 3-5-3 信頼性タイプと機能適合性の関係に、3種類の信頼性タイプに分類したソフトウェア製品群に関して機能適合性スコアの値を箱ひげ図で示した。安定タイプにおいては他のタイプと比べ機能適合性スコアが高く、高品質であることがわかる。

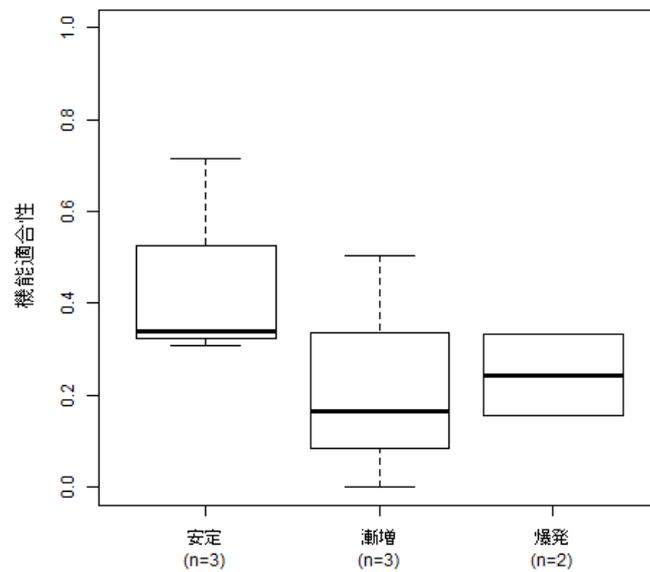


図 3-5-3 信頼性タイプと機能適合性の関係

性能効率性: 図 3-5-4 信頼性タイプと性能効率性の関係に, 3種類の信頼性タイプに分類したソフトウェア製品群に関して性能効率性スコアの値を箱ひげ図で示した. 爆発タイプにおいては他のタイプと比べ性能効率性スコアが低く, 低品質であることがわかる.

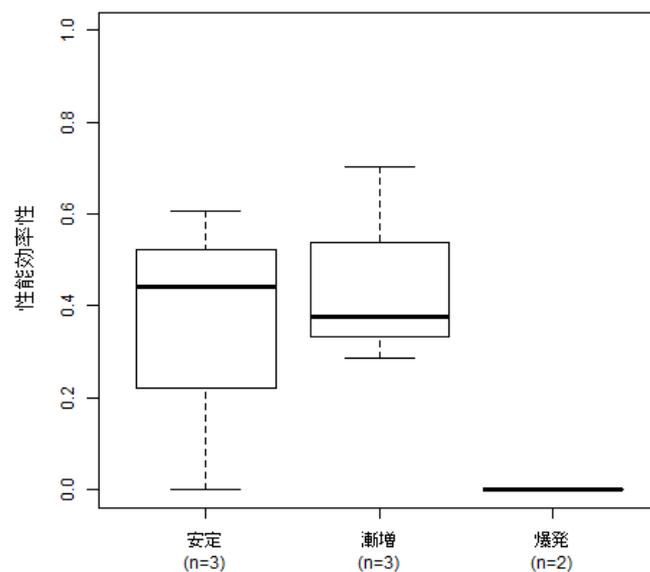


図 3-5-4 信頼性タイプと性能効率性の関係

互換性: 図 3-5-5 信頼性タイプと互換性の関係に, 3種類の信頼性タイプに分類したソフ

トウェア製品群に関して互換性スコアの値を箱ひげ図で示した。爆発タイプにおいては他のタイプと比べ互換性スコアが低く、低品質であることがわかる。

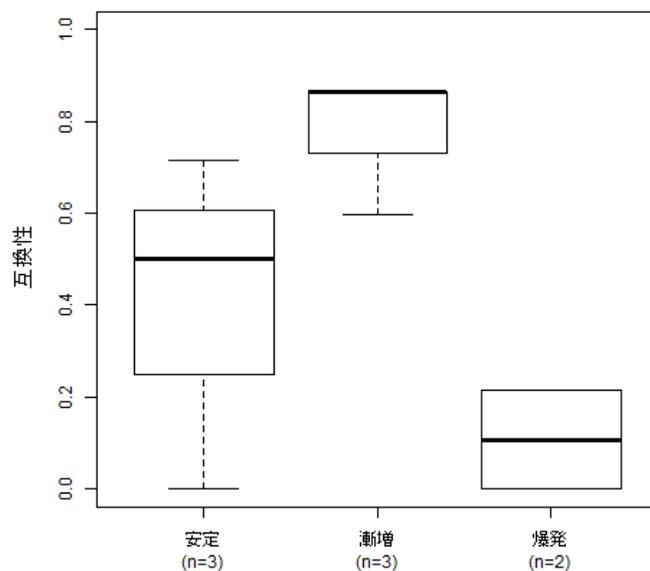


図 3-5-5 信頼性タイプと互換性の関係

使用性：図 3-5-6 信頼性タイプと使用性の関係に、3種類の信頼性タイプに分類したソフトウェア製品群に関して使用性スコアの値を箱ひげ図で示した。概ね3種類の信頼性タイプに関係なく同様の使用性スコアをとっており、顕著な違いは見られない。

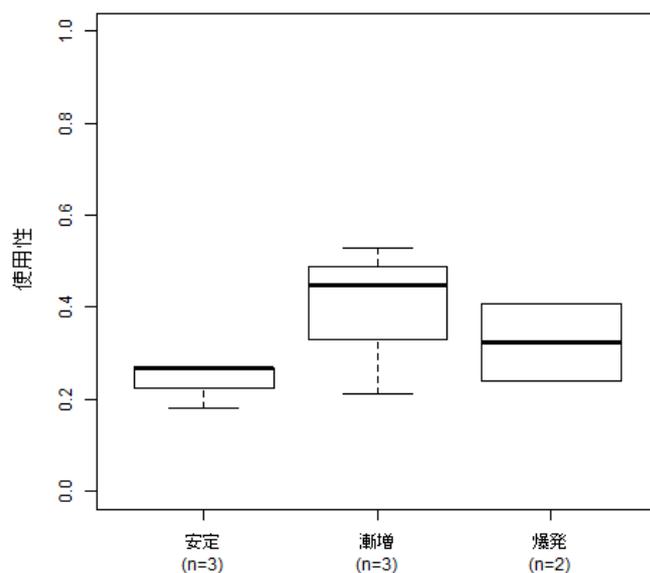


図 3-5-6 信頼性タイプと使用性の関係

信頼性: 図 3-5-7 信頼性タイプと信頼性の関係に, 3 種類の信頼性タイプに分類したソフトウェア製品群に関して信頼性スコアの値を箱ひげ図で示した. 安定タイプにおいては他のタイプと比べ信頼性スコアが高く, 高品質であることがわかる.

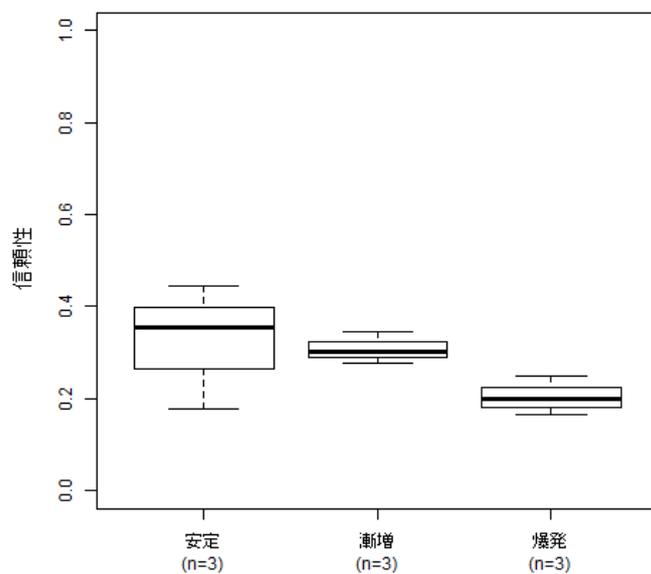


図 3-5-7 信頼性タイプと信頼性の関係

セキュリティ: 図 3-5-8 信頼性タイプとセキュリティの関係に, 3 種類の信頼性タイプに分類したソフトウェア製品群に関してセキュリティスコアの値を箱ひげ図で示した. 概ね 3 種類の信頼性タイプに関係なく同様のセキュリティスコアをとっており, 顕著な違いは見られない.

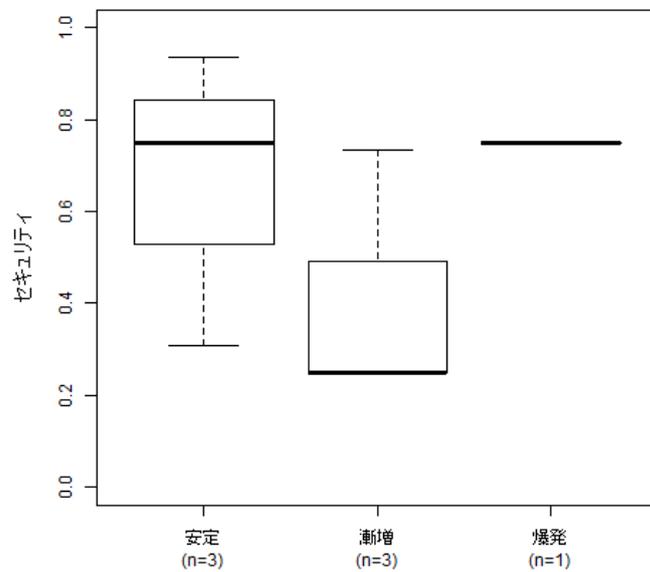


図 3-5-8 信頼性タイプとセキュリティの関係

保守性：図 3-5-9 信頼性タイプと保守性の関係に、3 種類の信頼性タイプに分類したソフトウェア製品群に関して保守性スコアの値を箱ひげ図で示した。概ね 3 種類の信頼性タイプに関係なく同様の保守性スコアをとっており、顕著な違いは見られない。

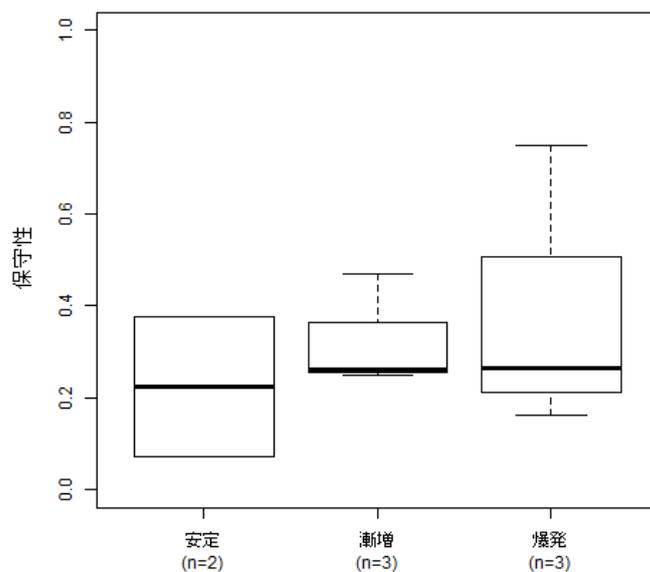


図 3-5-9 信頼性タイプと保守性の関係

移植性：図 3-5-10 信頼性タイプと保守性の関係に、3 種類の信頼性タイプに分類したソフ

トウェア製品群に関して移植性スコアの値を箱ひげ図で示した。概ね 3 種類の信頼性タイプに関係なく同様の移植性スコアをとっており、顕著な違いは見られない。

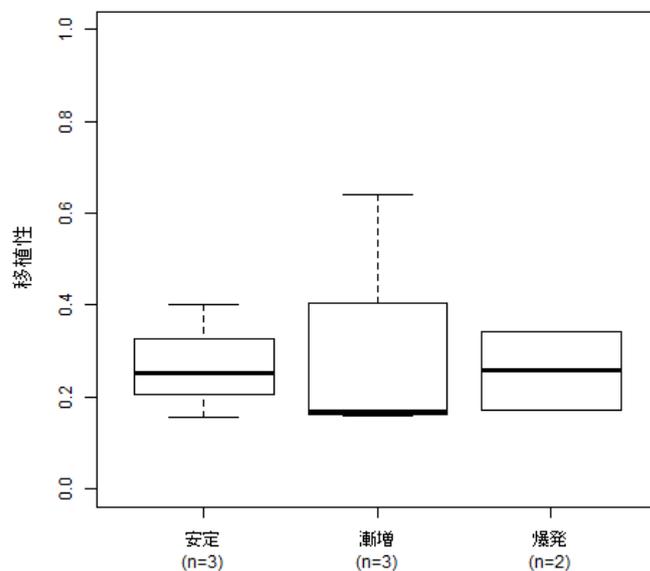


図 3-5-10 信頼性タイプと移植性の関係

有効性: 図 3-5-11 信頼性タイプと利用時品質 (有効性) の関係に、3 種類の信頼性タイプに分類したソフトウェア製品群に関して利用時品質 (有効性) スコアの値を箱ひげ図で示した。安定タイプにおいては他のタイプと比べ利用時品質 (有効性) スコアが高く、高品質であることがわかる。

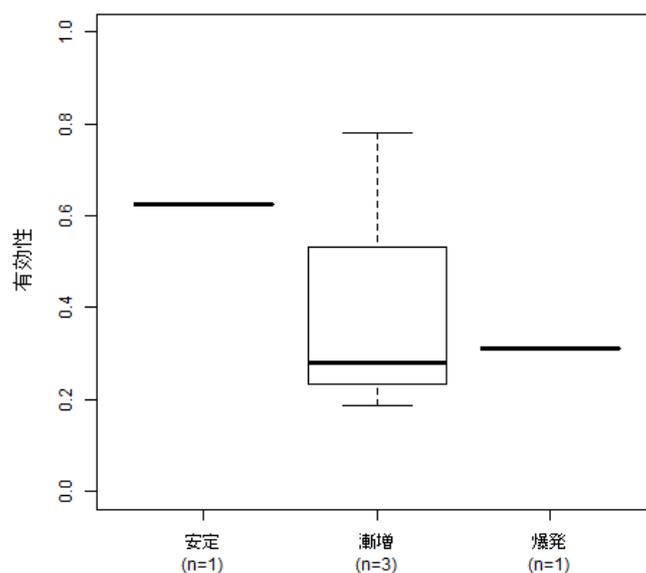


図 3-5-11 信頼性タイプと利用時品質 (有効性) の関係

効率性：図 3-5-12 信頼性タイプと利用時品質（効率性）の関係に、3種類の信頼性タイプに分類したソフトウェア製品群に関して利用時品質（効率性）スコアの値を箱ひげ図で示した。概ね3種類の信頼性タイプに関係なく同様の利用時品質（効率性）スコアをとっており、顕著な違いは見られない。

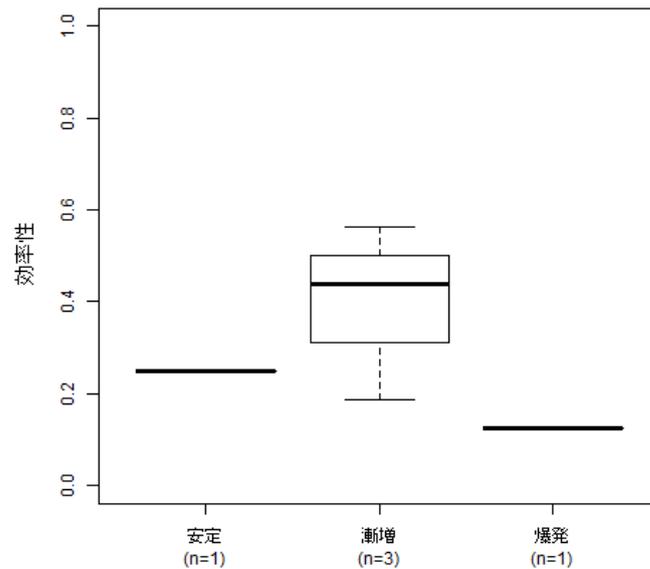


図 3-5-12 信頼性タイプと利用時品質（効率性）の関係

利用時品質（利用状況網羅性）：図 3-5-13 信頼性タイプと利用時品質（利用状況網羅性）の関係に、3種類の信頼性タイプに分類したソフトウェア製品群に関して利用時品質（利用状況網羅性）スコアの値を箱ひげ図で示した。3種類の信頼性タイプについて、安定タイプ1製品、漸増タイプ1製品、爆発タイプ0製品であったため、統計的な分析はできない。

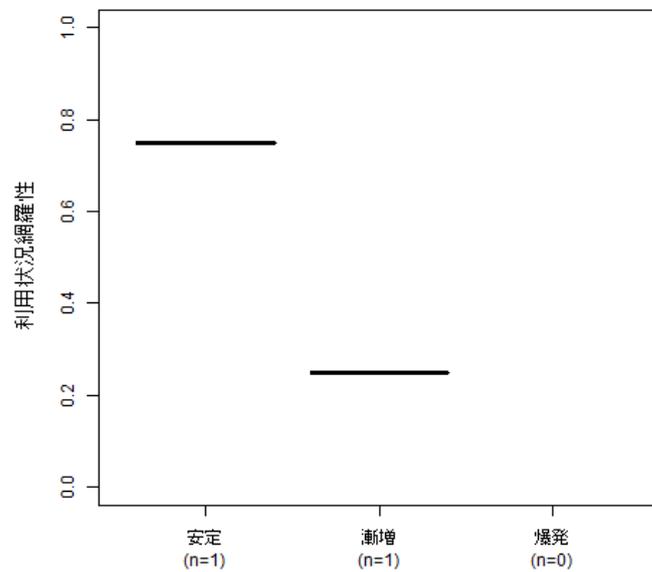


図 3-5-13 信頼性タイプと利用時品質（利用状況網羅性）の関係

利用時品質（リスク回避性）：図 3-5-14 信頼性タイプと利用時品質（リスク回避性）の関係に、3 種類の信頼性タイプに分類したソフトウェア製品群に関して利用時品質（リスク回避性）スコアの値を箱ひげ図で示した。3 種類の信頼性タイプについて、安定タイプ 1 製品、漸増タイプ 1 製品、爆発タイプ 0 製品であったため、統計的な分析はできない。

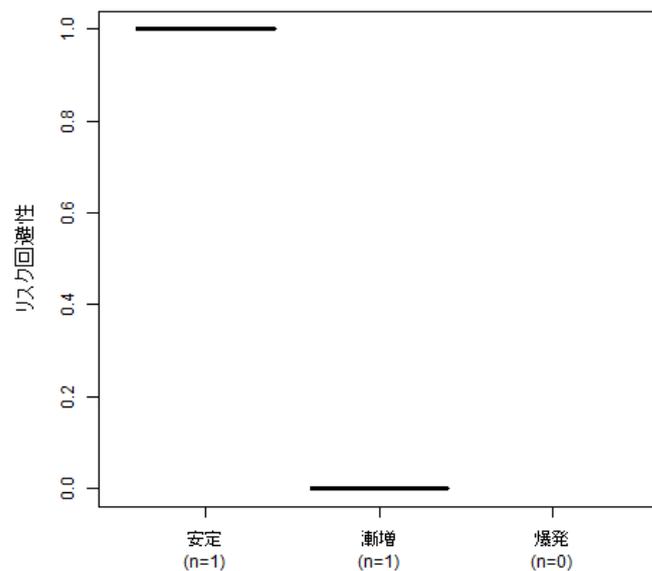


図 3-5-14 信頼性タイプと利用時品質（リスク回避性）の関係

利用時品質（満足性）：図 3-5-15 信頼性タイプと利用時品質（満足性）の関係に、3種類の信頼性タイプに分類したソフトウェア製品群に関して利用時品質（満足性）スコアの値を箱ひげ図で示した。3種類の信頼性タイプについて、安定タイプ1製品、漸増タイプ1製品、爆発タイプ0製品であったため、統計的な分析はできない。

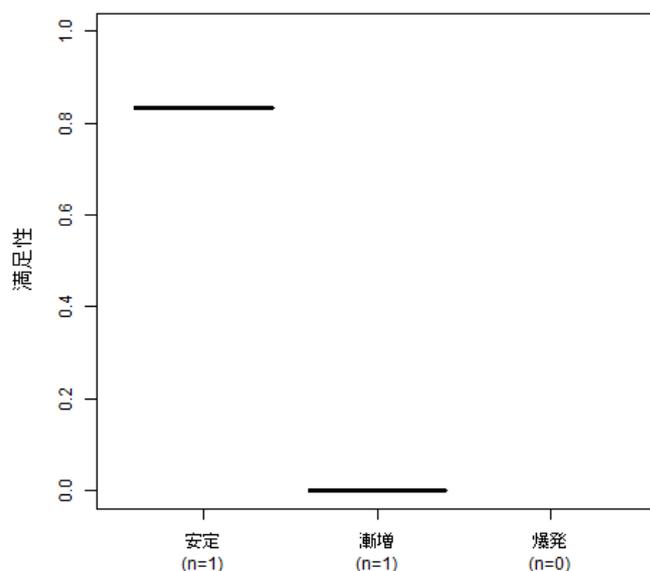


図 3-5-15 信頼性タイプと利用時品質（満足性）の関係

3) 品質とプロジェクトコンテキスト間の関係分析結果

各製品のコンテキストと品質間関係を分析するために、各製品のコンテキストとしてドメイン、対象開発期間、開発形態、提供種別、機能数、プログラムソースコード行数の各情報を製品提供元から得た。全 21 製品のコンテキストの内訳を図 3-5-16～図 3-5-20 に示す。なお製品によってはコンテキスト情報の一部が得られていない。

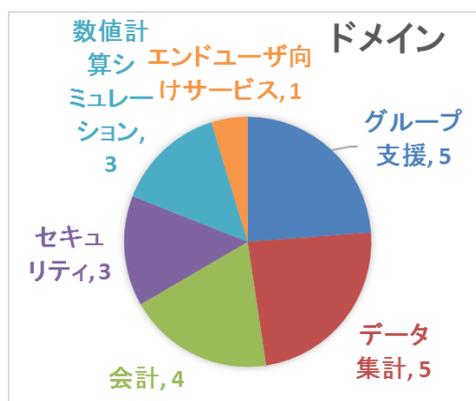


図 3-5-16 製品コンテキストの内訳：ドメイン

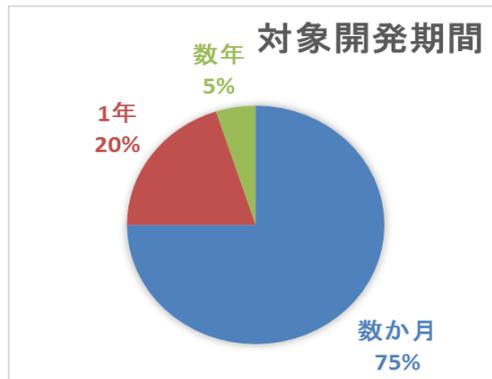


図 3-5-17 製品コンテキストの内訳：対象開発期間

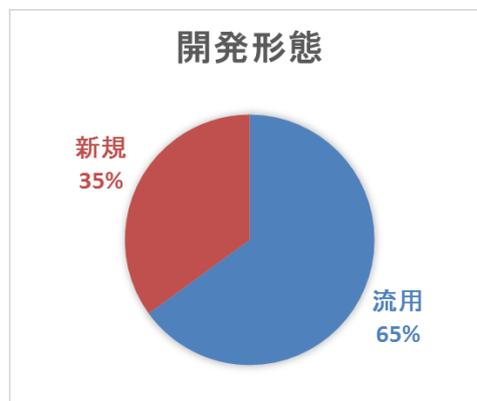


図 3-5-18 製品コンテキストの内訳：開発形態

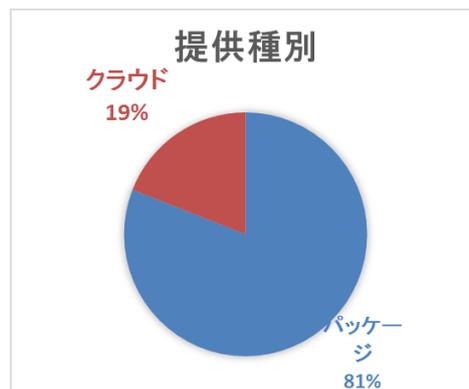


図 3-5-19 製品コンテキストの内訳：提供種別

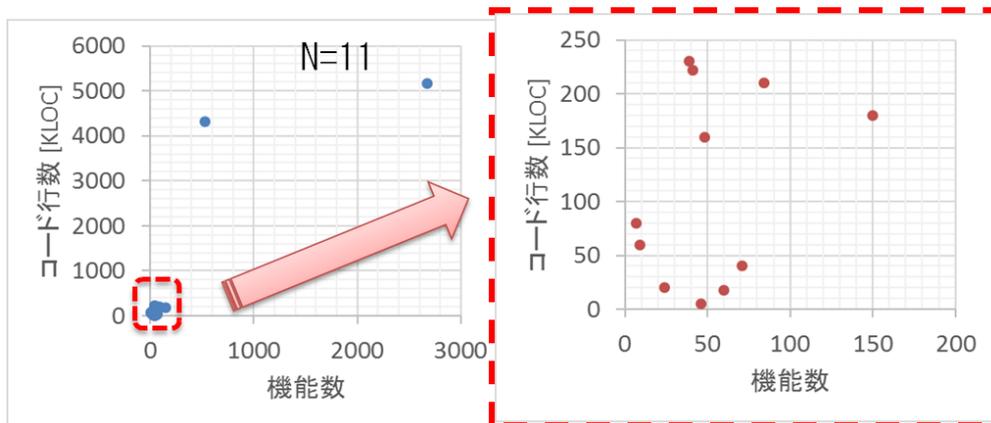


図 3-5-20 製品コンテキストの内訳：規模（機能数およびコード行数）

続いて、これらのコンテキストと品質評価スコアの間を関 3-5-21～関 3-5-23 に示す。関係の分析により、得られたデータの範囲において観察しうる事柄を以下にまとめる。

- **ドメイン別：** 互換性およびセキュリティについてドメインによって平均スコアに顕著な差が見られた。例えばエンドユーザ向けサービス製品においてセキュリティは極めて高く、数値計算シミュレーション製品においてセキュリティは低い。これは主としてドメインごとに重視される品質特性の違いに起因するものと考えられる。しかし今後の本格的な IoT 時代において、互換性およびセキュリティはあらゆるソフトウェア製品にとって重要な品質特性となる可能性が高く、ソフトウェアの開発プロセスや方法、取り巻く環境が変革されつつある中で、ソフトウェア品質に対する意識についても、機能適合性や信頼性に代表されるいわゆる狭義の「品質」に加えて、このような IoT 時代の品質を考慮するように変革が求められてくると考えられる。もう一つの理由として、互換性およびセキュリティについて、国際規格 SQuaRE シリーズにおける定義から具体化できたメトリクス数が限られていたこともあげられる。特に互換性については、わずかに二つのメトリクスにとどまったため、一つのメトリクスの測定値が品質特性のスコアへ多大な影響を及ぼすこととなった。これらの品質特性は、ISO/IEC 9126 から国際規格 SQuaRE シリーズへと改訂される際に主特性として格上げされたものであり、国際規格側において今後の実効性のあるメトリクスの拡充が望まれる。
- **提供種別：** パッケージ製品とクラウド製品において品質の全体的な傾向は概ね類似している。その中で、パッケージ製品についてはクラウド製品に比べるとセキュリティが著しく低く、ネットワーク接続や利用環境の変化を考慮する場合にセキュリティの強化が課題といえることができる。一方、クラウド製品についてはパッケージ製品に比べて保守性や移植性がやや低い結果となった。原因としては、クラウド製品におけるソフトウェアの保守および移植においては DevOps に代表される運用体制と連携した機能拡張や、仮想化システムの標準への対応といったクラウド環境特有の考慮が必要となり、パッケージ製品とはやや異なる点が測定評価に用いた国際規格 SQuaRE シリーズにおいて考慮されていないという点が推測できる。また、信頼性についてもクラウド製品はパッケージ製品に比べて低いが、これは同様にメトリクスにおいてクラウド環境の考慮が不足している可能性がある。

- 規模：規模別でみた場合に品質に顕著な傾向の違いは見られなかった。
- 期間：期間別でみた場合に品質に顕著な傾向の違いは見られなかった。
- 開発形態：開発形態別でみた場合に品質に顕著な傾向の違いは見られなかった。

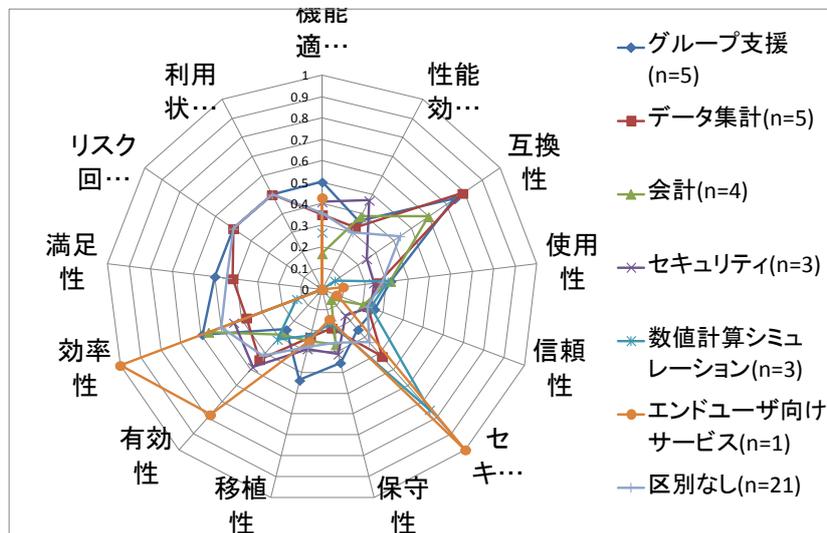


図 3-5-21 ドメイン別の品質評価スコア平均

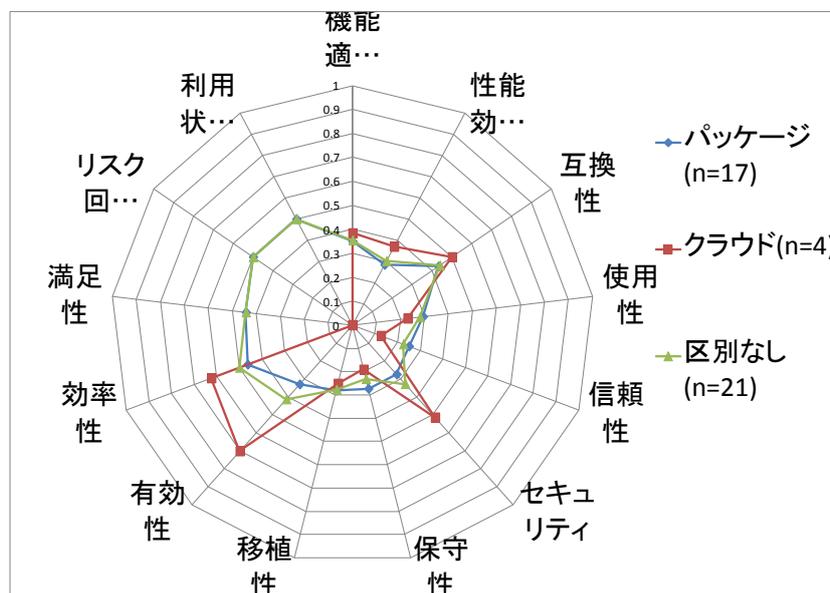


図 3-5-22 提供種別に基づく品質評価スコア平均

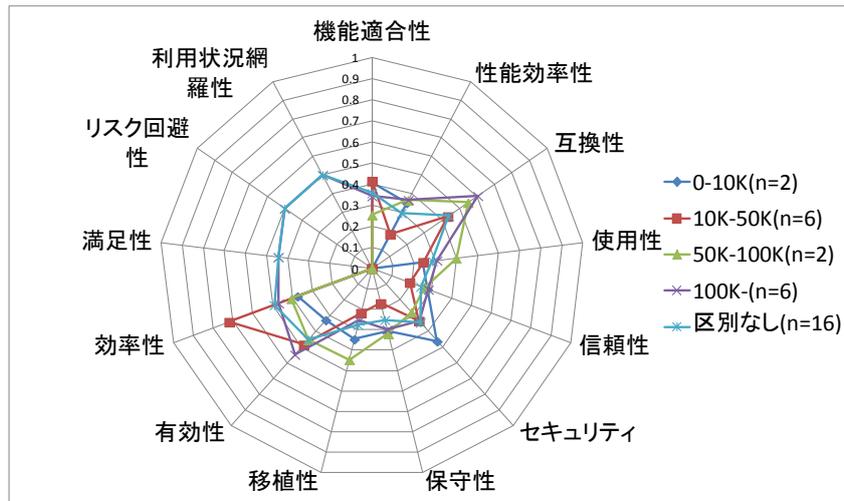


図 3-5-23 コード行数別の品質評価スコア平均

②品質実態のまとめ (研究プロセス②)

品質実態として、以上の分析により得られた品質特性間の関係および傾向を図 3-5-10 にまとめる。

製品品質の中では、特に機能適合性および信頼性は多くの製品においてスコアが高いところに集中しており、いわば狭義の「品質」として十分考慮され作り込まれていたと考えられる。他方、使用性や保守性、セキュリティ、互換性については製品によってばらつきがあり、IoT 時代に重要なセキュリティおよび互換性については 2 極化の傾向にあった。これらについて、今後の環境を見据えた必要性や考慮の余地があると考えられる。特にパッケージ製品におけるセキュリティ強化は大きな課題といえることができる。また、機能適合性の作り込みの中で使用性が損なわれた可能性があり（あるいは後者の軽視）、製品の価値を判断する立場は本来ユーザであるため様々な品質特性を多面的に考慮するユーザ中心の取組みが今後は求められる。

利用時の品質については残念ながら多くの製品について測定評価できなかったが、効率性と有効性については一定数を測定評価でき、その中で 2 極化の傾向が見られた。開発者視点の製品品質を作り込む中で、常にユーザ視点の利用時の品質もあわせて意識し、それらの測定評価および改善の取り組みが今後はより一層重要である。

製品品質と利用時の品質の関係としては、信頼性と有効性および移植性と効率性の間に正の相関が見られた。加えて、信頼性モデルにより予測評価される信頼性タイプが安定タイプである場合に有効性が高く、逆に信頼性タイプが漸増タイプおよび爆発タイプであると有効性が低い傾向にあった。有効性を高めるためには、開発中において確実に摘出欠陥数を収束させることが望ましいと考えられる。

なお、マトリクスや品質特性によっては対象製品数や取得できたデータが少なく統計的に母数として十分な数量とは言い難いものもあることに留意が必要である。上述の分析結果はあくまでも今回取得したデータ規模における推計である。また、今回の研究はソフトウェアの全ての領域に渡っているのではなく、パッケージ等の供給側の立場からの品質評価が主になっている点にも留意が必要である。

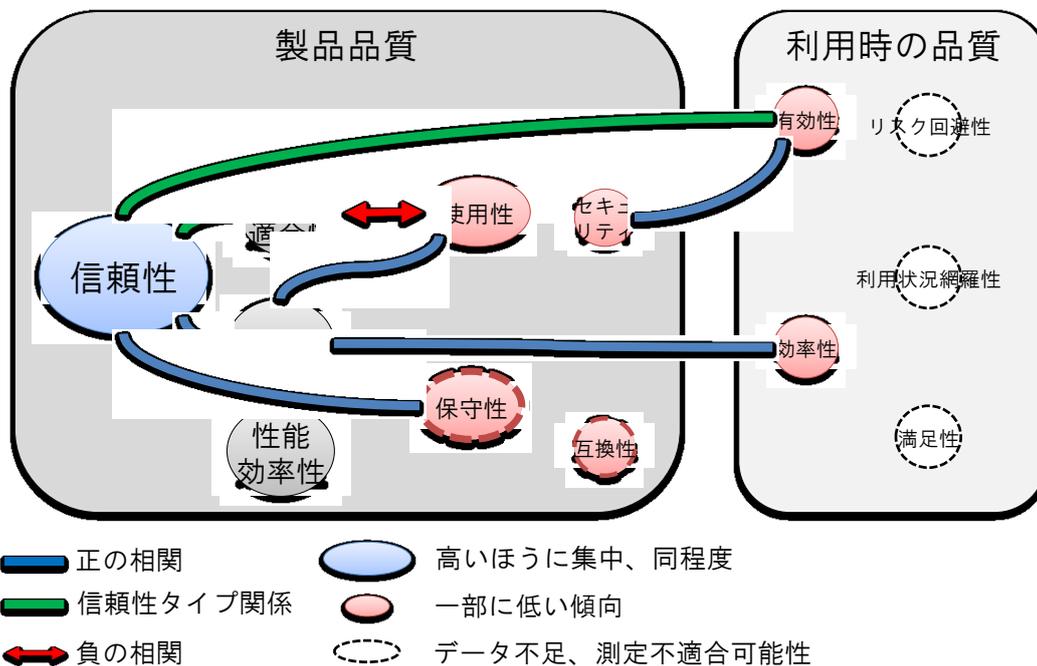


図 3-5-10 品質特性間関係のまとめ

3.5.3 発生した問題および今後の展望

(1) 発生した問題

メトリクスおよび品質特性によっては測定値のある製品に限られており、統計解析による精密な関係モデルの導出が難しい。そこで、データ群を可視化し人手による分析と意味づけにより関係の特定に成功した。

(2) 今後の展望

品質実態調査を継続することでデータ数を増やし、継続的に関係分析結果を更新する予定である。データ数が一定数に達した時点で、より精緻な関係モデルの導出を試みる。

3.6 研究課題 6「全体パッケージ化」

3.6.1 当初の想定

(1) 研究内容

研究課題 1~3 の実施により得られた品質測定評価方法とツール群を総合的な品質測定評価枠組みとしてまとめる。さらに研究課題 4・5 の実施により得られた個々の品質実態および関係としての総合的な品質実態をまとめ一般に公開する。

(2) 想定問題と対応策

問題： 公開する品質測定評価枠組みは、測定対象についてそれぞれ特定の制約に従うことを課すため、第三者による利用の幾らかの妨げとなる可能性がある。枠組みの一部において特定のプラットフォームを前提とする可能性がある。また、公開する品質実態における全

ての数値が、精密にあらゆるソフトウェア製品について該当すると捉えられてしまう可能性がある。

解決策：品質測定評価枠組みの公開にあたり，測定評価対象における制約を合わせて明示する。枠組みを構成する個々の要素について特定プラットフォームに対する依存度を明示することで，前提とは異なるプラットフォームにおいても再利用可能な要素と，対象プラットフォームに応じて修正・再実装が必要な要素を明確とし利用の便を図る。また品質実態について，測定評価対象の規模やドメイン等のコンテキスト情報も付与することで，当該実態を把握したコンテキストの設定を明確とし，それとは異なるコンテキストにおける利用における留意を促す。

3.6.2 研究プロセスと成果

(1) 研究プロセス

- ① 品質測定評価枠組みをパッケージ化する。
- ② 品質実態をパッケージ化する。
- ③ 公開サイトを準備し，パッケージを公開する。

(2) 具体的な研究成果の内容

① 品質測定評価枠組みのパッケージ化（研究プロセス①）

以下をパッケージ化した。

- WSQB17 Software Quality Measurement and Evaluation Suite ソフトウェア製品の品質測定評価枠組み，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（ツールについては今後随時改訂する予定）
 - WSQB17-MP (ZIP)：ソフトウェア製品品質測定用 記入様式 Data Input Form for Product Quality (研究課題 1①②③，研究課題 2①②③)
 - WSQB17-MQ (Excel)：ソフトウェア利用時品質測定用 ユーザーアンケート雛形 User Questionnaire Template for Quality in Use (研究課題 3①)
 - WSQB17-MT (Excel)：ソフトウェア利用時品質測定用 ユーザーテスト雛形 User Testing Template for Quality in Use (研究課題 3②③)
 - WSQB17-MA (Excel, R)：ソフトウェア品質評価用 集計ツール Quality Evaluation Aggregation Tool (研究課題 1③、研究課題 2③、研究課題 3④)
 - WSQB17-MR (R)：ソフトウェア信頼性予測ツール Software Reliability Prediction Tool (研究課題 2②③)

② 品質実態のパッケージ化（研究プロセス②）

以下をパッケージ化した。

- WSQB17 Software Quality Dataset 公開データセット 21 製品評価結果 2017 年 1 月版，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
 - WSQB17-DC (Excel)：データセット構成 21 製品コンテキストデータ Product Context Data (研究課題 4①②)
 - WSQB17-DQ (Excel)：製品品質・利用時品質測定評価データ Product Quality and

Quality in Use Data (研究課題 4③④、研究課題 5①②)

③ 公開サイトの準備とパッケージ公開 (研究プロセス③)

本調査研究の成果として、国際規格 ISO/IEC 25000 SQuaRE シリーズに基づくソフトウェア製品の品質測定評価枠組み (Software Quality Measurement and Evaluation Suite) , および、同枠組みを用いて日本の 21 製品に対する測定評価結果をまとめたベンチマークとしての公開データセット (Software Quality Dataset) を, WSQB17: Waseda Software Quality Benchmark として次の URL 上の Web ページにおいて、図 3-6-1 に示すロゴマークと共に一般に公開した。

http://www.washi.cs.waseda.ac.jp/?page_id=3479

The logo for WSQB17, featuring the text 'WSQB17' in a bold, red, sans-serif font. The letters have a slight shadow effect, giving it a 3D appearance. The '17' is slightly smaller than the 'WSQB'.

図 3-6-1 ロゴマーク

3.6.3 発生した問題および今後の展望

(1) 発生した問題

当初設計した測定評価用の様式について、そのままでは第三者による集計・分析を実施しにくい問題があった。集計・分析しやすいように中間様式を設けて部分的に集計作業を自動することで対応した。

(2) 今後の展望

品質実態調査を継続してデータ数の増加共に品質ベンチマークを 1 年程度の頻度で更新する。また、内部・外部・利用時品質の測定評価ツールの更新に伴って品質測定評価枠組みを更新して公開する。両パッケージの利用ガイドを作成して公開するとともに、利用事例を広く募集して蓄積掲載する。

4 考察

4.1 研究による効果や問題点等

世界的な基準から客観的に製品を評価し国際競争力を高めることが必要であり、本調査研究においては国際規格に基づいてソフトウェアの総合的な品質評価の枠組みを確立することに成功した。具体的には、国際規格 ISO/IEC 25000 SQuaRE シリーズを具体化し、製品品質と利用時品質を網羅的に測定評価するとともに関係を分析することに成功した。対象として 21 製品を個別評価するとともに、業界の実態を分析し得られたデータの範囲内で明らかとすることに成功した。

得られた成果として、評価枠組み(信頼性予測含む)および実態データセットをパッケージ化し、Waseda Software Quality Benchmark (WSQB17)として公開した。この成果は国際標準のベンチマークとして認知され幅広く活用および発展することが期待される。得られた成果を広く一般に向けて SEC journal や公開セミナー等を通じて広く周知する予定である。

本調査研究は品質実態調査として、品質特性によっては記録の不足や目標値未設定といった理由からデータが得られた製品数が限られており、品質実態調査を継続することでデータ数を増やし、継続的に関係分析結果を更新することが今後の課題である。そのための事業化に向け予算申請の準備を進めている。また、データ数が一定数に達した時点で、より精緻な関係モデルの導出を試みる予定である。

4.2 産業界への展開と今後の研究の進め方

4.2.1 研究成果の産業界への展開

成果として得られた品質測定評価の枠組みを、CSAJ における PSQ 認証制度における認証サービスに組み入れることで、契約期間終了後においても産業界へ継続して適用およびデータ収集を通じた分析を実施可能とする。この計画について、研究責任者・研究チームと CSAJ において合意済みである。

また、パッケージ化した研究成果を企業が参照し、品質向上に取り組むことを期待する。さらに研究成果を企業が利用することで、ソフトウェアの品質向上が期待できる。

4.2.2 今後の研究の進め方

研究責任者および研究チームは契約期間終了後も引き続き、測定評価対象を拡充させる形で総合的な品質測定評価枠組みの拡張・洗練に取り組む予定である。さらに、ソフトウェア提供企業の協力を得て、同一のソフトウェア製品系列に対する(隔年単位等の)継続的な品質測定評価を実施することで、品質の変化過程を調査分析し、結果をまとめる予定である。

また研究チームのメンバは、本研究において適用する国際規格 ISO/IEC 25000 (SQuaRE) シリーズの全体統括を務めており、本研究の成果として得られた品質測定評価の枠組みにおける測定法および評価方法を、当該 ISO/IEC 25000 シリーズの改訂に反映させる予定である。

4.2.3 産業界および国際標準化団体への要望

本調査研究の成果として、以下を産業界への提言とする。

- 提言 1. 全てがつながる IoT/IoE 時代に最重要となるはずの品質であるセキュリティおよび 互換性が一部低いパッケージ製品が見られた。ソフトウェアの開発プロセスや方法, 取り巻く環境が変革されつつある中で, ソフトウェア品質に対する意識の変革も必要である。
- 提言 2. 機能適合性の作り込みの結果として, 本来のユーザにとって重要な使用性が損なわれている可能性がある。製品の価値を判断する立場は本来ユーザであり, 様々な品質特性を多面的に考慮するユーザ中心の取組み (例えばユーザ中心の設計など) が今後は求められる。
- 提言 3. 製品数としては 21 製品の協力を得たが, 残念ながら各品質特性単位でみると実際に測定評価できた製品数は半数に満たないものがほとんどであった。この原因としては, 各製品提供元において, 様々な品質特性の観点から評価可能とするための根拠となるデータを記録していないこと, および, SQuaRE シリーズにおいてもともと想定している品質測定の方法において必要な目標値がほとんど設定されていないことが挙げられる。多面的な品質測定評価を可能とするためのデータの記録, ならびに, 本調査研究の成果としえられたベンチマークを参照したうえでの目標値設定が重要である。

さらに, 本調査研究の成果として, 以下を国際標準化団体, 特に国際標準化機構 ISO/IEC JTC1/SC7/WG6 への提言とする。

- 提言 4. 本調査研究によって, SQuaRE シリーズにおける品質測定方法の多くは抽象度が高く定義のままでは適用困難であることが判明した。そこで, 本調査研究において実効性のある形で具体化に成功した品質測定評価の枠組みを組み入れて実効性を強化すべきである。例えば, SQuaRE シリーズの一環として, ISO/IEC テクニカルレポートとして枠組みを公開することを提言する。
- 提言 5. SQuaRE シリーズにおける品質測定方法の多くは 1990 年代までのソフトウェア製品の形態および開発方法を念頭においており, アジャイル開発やプラットフォームとしてのクラウドに対する考慮を幾らか欠いており, 今後の対応強化が求められる。例えばクラウド製品におけるソフトウェアの移植においては DevOps に代表される運用体制と連携した機能拡張や, 仮想システムの標準への対応といったクラウド環境特有の考慮が必要である。

謝辞

本調査研究を実施するにあたり, 12 社の製品提供元各位から快く品質測定評価のための 21 製品のデータ提供をいただいたことに深く御礼申し上げます。また測定評価の一部の作業は複数の外部評価機関各位に実施いただいたことに深く御礼申し上げます。製品募集や進行にあたり多くのご支援をいただいた CSAJ 殿に深く御礼申し上げます。ユーザテストをはじめとする測定評価に関する作業について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, 鷺崎研究室および情報理工学科における次の学生ならびにスタッフが関係したことに謝意を示します: 橋本慧氏, 中井秀矩氏, 青木耀平氏, 角田武人氏, 礎良輔氏, 野寄祐樹, 渡邊泰宏氏, 先崎祐一郎氏, 細野将揮氏, 山田龍平氏, 志村千万輝氏。最後に,

本調査研究の機会をいただいた IPA 殿ならびに関係各位に深く御礼申し上げます。

参考文献

- [JIS 25010] JIS X25010 : 2013 (ISO/IEC 25010 : 2011) 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
- [東 15] 東基衛, ” システム・ソフトウェア品質標準 SQuaRE シリーズの歴史と概要”, SEC Journal, Vol.10, No.5, 2015.
- [ISO25000] ISO/IEC 25000:2014 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE
- [鷺崎 07] 鷺崎弘宜, 波木理恵子, 福岡呂之, 原田洋子, 渡辺博之, “プログラムソースコードのための実用的な品質評価枠組み”, 情報処理学会論文誌, Vol.48, No.8, 2007.
- [Sindre95] G. Sindre et al., “The REBOOT Approach to Software Reuse,” J. Systems and Software, 30(3), 1995.
- [Bansiya02] J. Bansiya and C.G. Davis, “A Hierarchical Model for Object-Oriented Design Quality Assessment,” IEEE Transactions on Software Engineering, Vol.28, No.1, 2002.
- [Ortega03] M. Ortega, M. Perez and T. Rojas, “Construction of A Systematic Quality Model for Evaluating A Software Product,” Software Quality Journal, Vol.11, No.3, 2003.
- [鷺崎 10] 鷺崎弘宜, 田邊浩之, 小池利和, ” ソースコード解析による品質評価の仕組み”, 日経エレクトロニクス, 2010年1月25日号
- [Washizaki12] H. Washizaki, et al., “Reusability Metrics for Program Source Code Written in C Language and Their Evaluation,” 13th International Conference on Product-Focused Software Development and Process Improvement (Profes 2012), 2012.
- [Sato13] S. Sato, H. Washizaki, et al., “Effects of Organizational Changes on Product Metrics and Defects,” 20th Asia-Pacific Software Engineering Conference (APSEC 2013), 2013.
- [Tsuda14] N. Tsuda, H. Washizaki, Y. Fukazawa, “Evaluating Structural Validity of UML Class Diagrams by Measuring the Number of Highly Responsible Classes,” 13th IASTED SE2014, 2014.
- [Nakai14] H. Nakai, K. Honda, H. Washizaki, et al. “Initial Industrial Experience of GQM-based Product-Focused Project Monitoring with Trend Patterns,” 21st Asia-Pacific Software Engineering Conference (APSEC 2014)
- [鷺崎 15] 鷺崎弘宜, ” ゴール指向の測定によるソフトウェア品質評価と改善の実践的取組み (Yahoo! JAPAN, 富士通, コマツ等とのそれぞれ個別の産学連携成果)”, 早稲田大学 & IESE&IPA/SEC 共催セミナー, 2015年2月19日
- [ISO25023] ISO/IEC DIS 25023 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Measurement of system and software product quality
- [ISO25022] ISO/IEC DIS 25022 Systems and software engineering -- Systems and software quality requirements and evaluation (SQuaRE) -- Measurement of quality in use

- [東 14] 東基衛, “ICT 応用システムおよびソフトウェア (S&S) の品質向上のための課題と取り組み”, 情報処理, Vol. 55, No.1, 2014.
- [込山 14] 込山俊博, “システムおよびソフトウェアの品質基準の体系化”, 情報処理, Vol. 55, No.1, 2014.
- [ISO25051] ISO/IEC 25051:2014 Software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Requirements for quality of Ready to Use Software Product (RUSP) and instructions for testing
- [IPA' 14] IPA, “製品・システムにおけるソフトウェアの信頼性・安全性等に関する品質説明力強化のための制度構築ガイドライン”, 2014.
- [PSQ] CSAJ, パッケージソフトウェア品質認証制度(PSQ 認証制度),
<http://www.csaj.jp/psq/>
- [IPA15] IPA/SEC, “ソフトウェア開発データ白書 2014-2015”, 2015.
- [Cusumano03] Michael Cusumano, Alan MacCormack, Chris F. Kemerer, William Crandall, “Software Development Worldwide: The State of the Practice,” IEEE Software, 20(6), 2003.
- [Jones09] Capers Jones, “Software Engineering Best Practices: Lessons from Successful Projects in the Top Companies,” McGraw-Hill, 2009.
- [Basili94] V. Basili, et al., “Goal, Question, Metric Paradigm,” Encyclopedia of Software Engineering, Vol. 1, 1994.