

System&Software に対する品質確保におけるシステム検証技術の提案

Proposal of system verification technology for quality assurance for System & Software

富士通株式会社
FUJITSU LIMITED
○原田 知広
○Tomohiro Harada

株式会社富士通ゼネラル¹⁾
FUJITSU GENERAL¹⁾
森田 純恵 高内 龍治
Sumie Morita Ryuji Takauchi

早稲田大学²⁾
Waseda University²⁾
鷺崎 弘宜 津田 直彦
Hironori Washizaki Naohiko Tsuda

Abstract With the recent remarkable advancement in the ICT, system and software products (S&S: System&Software) applying ICT have become increasingly diversified. And here, a question arises that who will, in what way shall take the responsibilities of the quality assurance for such System&Software. To answer this, based on our experiences, we propose a "System validation technique" focusing on the "Quality in Use" and "Product Quality". A validation was performed using a certain benchmark method (WSQB), to the result to which this "System validation technique" was applied. In this paper, the study of the result of the validation is presented.

1. はじめに

近年 ICT の急激な進歩によって、今まで独立していた機器はインターネット上からサービスを享受できるようになった。例えばルームエアコンは AI/IoT の活用により、エアコン自身が学習、成長をして、利用者が操作しなくても効率の良い運転を自動で行えるようになった^[1]。

また、自動車、医療、産業用途に対してもコネクテッドカーやデジタルヘルスケアの市場拡大、スマート工場の拡大が予想^[2]されている。そして、ICT を応用したシステム及びソフトウェア製品 (S&S: System&Software)^[3]は、ますます多様化していくことが見込まれ、ソフトウェア開発の重要性が高まる反面、開発の難易度も年々上昇している。

本論文では、多様化していく S&S に対する品質確保のためのシステム検証技術を提案する。提案にあたっては、国際規格 SQuaRE に基づくソフトウェア品質の測定評価手法である Waseda

富士通株式会社 ソフトウェア事業本部 ソフトウェア検証統括部
FUJITSU LIMITED SOFTWARE BUSINESS UNIT SOFTWARE QUALITY ASSURANCE DIV.
神奈川県川崎市中原区上小田中 4 丁目 1-1

4-1-1 Kamikodanaka, Nakahara-ku, Kawasaki-shi, Kanagawa

1) 神奈川県川崎市高津区末長 3 丁目 3 番 17 号

3-3-17 Suenaga, Takatsu-ku, Kawasaki-shi, Kanagawa

2) 東京都新宿区大久保 3-4-1

3-4-1 Okubo, Shinjuku-ku, Tokyo

【キーワード：】 System&Software, SQuaRE, システム分析, システム検証, テスト項目抽出, ブラックボックステスト, WSQB

Software Quality Benchmark (WSQB)^[4]で定義された WSQB メトリクス、および WSQB ベンチマークを活用して、弊社製品の評価を行った。本稿では、2章で S&S に対する品質確保における我々の課題、3章でその課題解決に向けた仮説、4章でシステム検証技術の提案、5章で評価結果と考察を述べる。最後に6章でまとめを行う。

2. S&S に対する品質確保における課題

まずは、今後ますます多様化していく我々のソフトウェア製品及び、それらを組合せて利用されるシステム(S&S: System&Software)に対応できるように以下の「仕組み」を構築した。

1. ソフトウェア・エンジニアリング・プロセスのV字モデル^[5]において検証と妥当性確認(V&V: Verification and Validation)を重視することで、内部品質、外部品質の双方を保証する多様性のある開発プロセスを構築。
2. システム設計時に、国際規格 SQuaRE で定義されている品質特性と、SQuBOK に掲載されている品質技術と、各開発現場の品質保証の経験・実績(品質エクスペリエンス)を掛け合わせたマトリクス(品質ボックス)^[6]を活用。品質ボックスを利用して注力する品質技術と品質特性を確認する。そして、WSQB メトリクスから品質特性に対する達成度を評価するとともに、WSQB ベンチマークを用いてポジショニングを確認する。
3. システム構成とシステム全体のインタフェースを記載するソフトウェア軸のシステム仕様書のテンプレートを作成、これを利用するプロセスを策定した。さらに、システムに要求される非機能要件として、ISO/IEC 25010 に定義されている「利用時品質」と「製品品質」を観点に目指すべき品質を明確にして、それをシステムテスト項目として抽出するプロセスを策定した。

しかし、この作業をとおして、「仕組み」を適用する対象がソフトウェア開発をすべて自社で行う「自社開発」と、外部で開発されたソフトウェアを購入してシステムに組み込む「調達開発」では、異なる開発プロセスを持っており、同じ手法が適応できないことが判明した。特に「調達開発」においては、開発プロセスがブラックボックスなソフトウェアを利用してシステムを開発している。このため、誰がどんな品質をどのように責任をもって確保していくかという課題に直面した。ここで、「調達開発」とは、一般にソースコードが入手できないので、昨今、話題のオープンソースソフトウェア(OSS)活用よりも厄介な課題を抱える。

3. 課題解決に向けた仮説

これまで個々の開発プロセスによる定量評価を行ってきた。しかし課題解決に向けて、システム構築単位での「検証と妥当性確認(V&V: Verification and Validation)」を重視して、最終システム開発における S&S の品質の確保が必要と考えた。そのために、システムテストでの品質確認が最重要と捉えた。このため、「利用時品質」と「製品品質」の両観点でのシステム全体の品質の確保に向けて、本論文にて提案する「システム分析表」を用いて要件をレビューすることを前提に以下の仮説を立案した。

- ① 利用時品質は、利用者の要件を抽出する要件定義の時点で要件に対する利用時品質の品質特性の「妥当性確認」を行えば、「検証」におけるリスクを軽減させておくことで品質向上につながる。
- ② 製品品質は、製品品質の品質特性の観点でシステム構成要素ごとにその事象を捉えれば、異なる開発背景を持つシステムでも「妥当性確認」、及び「検証」による品質確保ができる

本仮説の検証に向け、開発背景の異なるソフトウェア製品として自社開発の製品と調達開発の製品の2つを選択。開発背景の異なる製品を組み合わせただけでも同レベルのシステムテスト項目を挙げるために、対象とするシステム構成を明確にした上で、システム全体を分析してテスト項目を抽出する施策を実施した。

4. システム検証技術の提案

4.1 テストの項目の抽出

通常は、システム仕様書からシステムテスト項目に落とし込む。しかし、テスト項目抜け漏れを無くするためにはテスト設計のスキルと経験が求められる。解決法として、予め定義したキーワードがシステム仕様書に含まれていないかチェックをして、含まれていたら自動でテスト項目に落とし込むという手法がある^[7]。しかし「調達開発」の場合、「自社開発」と異なり、開発プロセスがブラックボックス化されている場合が多い。このとき、システム仕様書のような要件を明確にかいた内部仕様書が提供されないことが多く、システム仕様書を用いたキーワード抽出ができない。これまではテスト設計者の経験やノウハウといった暗黙知に依存したテスト計画を行ってきたため、経験が浅いとテストの内容が不十分となってしまう、リリース後に障害が発覚する状態となっていた。

本施策では、S&S という視点で、テスト計画の経験が浅くても十分に計画できるように、予め汎用化した非機能要件とシステム構成を組み合わせた「システム分析表」という体系的なマトリクスを定義した。この「システム分析表」を活用して、具体的な開発プロセスがブラックボックス化されている場合でもテスト項目を抽出し、要件のレビューができるような手法を構築する。

4.2 システム分析表の作成

4.1章で述べた「調達開発」について、開発プロセスはブラックボックスだが調達先から提供される機能は保証されている。このため、非機能要件の保証が必要となってくる。非機能要件は機能要件と異なり、ISO/IEC 25010 で定義されている品質特性を観点に要件を明確化すれば製品に依存せず共通化できると考えた。

そこで、縦軸を ISO/IEC 25010 で定義されている品質特性別に汎用化した観点、横軸をシステム構成要素とした図1のようなマトリクス表を用意した。これを「システム分析表」と呼ぶ。

品質特性	観点	システム構成要素		
	着眼点	要素A	要素B	要素C
性能効率性	処理速度			
	最大台数			
	最大同時接続台数			
	CPU負荷			
	メモリ枯渇			
	ログ領域枯渇			
信頼性	DB枯渇			
	電源ダウン/復旧			
	プロセスダウン/復旧			
	ネットワーク接続ダウン/復旧			
	ハード異常/復旧			
	データベースダウン/復旧			
保守性	連続稼働時間			
	問題発生時の情報収集			
	修正パッチの適用方法			
互換性	旧版からの刷新			
	OS違い			
移植性	機種(プラットフォーム)違い			
	異なるシステムへの移行			

図1 システム分析表(抜粋)

以下の手順で「システム分析表」を使いシステムを分析する。

- 手順1. 構築するシステムの構成要素、インタフェースを全て洗い出し、システム構成図(図2)としてまとめる。
 その際、各構成要素の台数やインタフェース間で利用するプロトコルについても可能な限り明確にした上でシステム構成図の中に落とし込む。

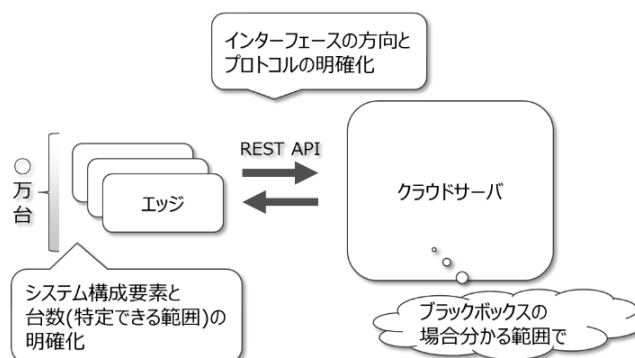


図2 システム構成図の一例

- 手順2. 手順1で洗い出した構成要素を「システム分析表」に反映。構成要素に対して品質特性別に定義されている観点に照らし合わせて、システム設計時に定義した要件に従い情報を入力していく。なお、品質特性別に定義している観点は、非機能要件の見える化と確認の手段を実現するために、独立行政法人情報処理推進機構で公開されている非機能要求グレード^[8]や過去のシステム障害の分析結果から抽出したものを汎用化させたものとする。

入力する情報は予め定型文化することで入力者のスキルに依存しないように工夫している。例えば信頼性の「電源ダウン/復旧」の場合、想定されるダウン内容、ダウン時の影響、復旧方法、復旧時間を穴埋めで入力する。また、構成要素がブラックボックスな場合、詳細な情報が書き出せないため、抽出できない旨を記載する。

手順3. 手順2で作成した「システム分析表」で入力した情報に対して、システムテスト項目に落とし込む。入力した情報から、そのままテスト項目に落とし込めるように定型文を作成している。例えば信頼性の「電源ダウン/復旧の場合」、ダウン内容が検証項目、ダウン時の影響、復旧方法、復旧時間が検証内容と期待値に当てはめることができる。

4.3 システム分析表の活用

4.2章で作成した「システム分析表」を2章で述べたとおり、構築した「しくみ」に適用する。「システム分析表」を適用した要件レビュー(妥当性確認)のプロセスの流れを図3に示す。

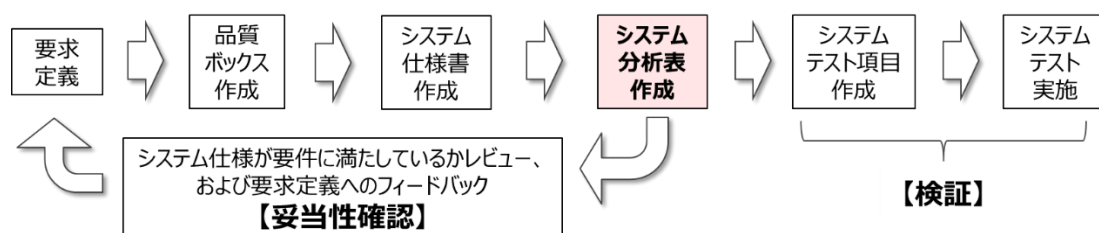


図3 「システム分析表」を活用したプロセス

また、利用時品質、製品品質に対する活用を以下に示す。

4.3.1 利用時品質に対する活用

「システム分析表」に入力する情報は、要件定義された内容を、定型文化して入力することで、システム評価経験が浅くても容易にシステム分析ができるように工夫した。また、ブラックボックスである旨も書き出されているので、どの範囲が不明確な部分となっているか、そして、その影響範囲がどこまであるのかを確認できるように工夫している。

「システム分析表」に入力された情報を基に、システム仕様が要件を満たしているか、利用時品質の品質特性の観点と照らし合わせて要件レビューを実施。そのレビュー結果から、要件の矛盾点、要件定義不足のような内容を要件定義にフィードバック後、再度システム設計を行い、「システム分析表」を更新してレビューを重ねることで要件に対する「妥当性確認」を行う。そして、この要件レビュー時に「システム分析表」の品質特別の観点(縦軸)の不足を検出した場合、随時追加していくことで分析精度の向上を計る。

4.3.2 製品品質に対する活用

「システム分析表」に入力する情報は、システムの構成要素に対して各品質項目別に定めた観点とその期待値が、マトリクス表として紐づけられるように工夫している。このため、システムテストの項目作成からテスト実施に対して、「システム分析表」以下のように活用する。

1. システムテスト項目では、各品質項目別に定めた観点はテスト項目、システム構成要素ごとに入力された情報はテスト項目に対する期待値として作成する
2. システムテストを実施する際は、テスト項目に対してシステムの構成要素ごとに定められている期待値通りになっているか検証する

上記のように、「システム分析表」を活用してテスト計画から実施までの一連の流れを実行することで、テスト経験が浅くても、異なる開発背景を持つシステムの検証が行えるようにする。

5. 評価結果と考察

本論文では、2章で示した自社開発と調達開発の2製品に対して3章・4章で提案する「システム検証技術」を適用して実際にシステム評価を行い、検出された不具合を削除するといった品質担保を行った。その後、早稲田大学と協力して、WSQB メトリクスの測定、および評価を実施。WSQB メトリクスを採用することで、今回提案したシステム検証技術に対して他社実績と比較した効果測定が期待できる。以下に、このベンチマーク結果を用いて、3章で示した仮説に対して、4章で提案した「利用時品質」と「製品品質」に着眼したシステム検証技術の妥当性を分析・検証する。

5.1 WSQB ベンチマークの測定結果

まず、「調達開発」と「自社開発」の2つの製品に対する「利用時品質」のWSQB ベンチマークの測定結果を図4、「製品品質」のWSQB ベンチマーク結果を図5に示す。

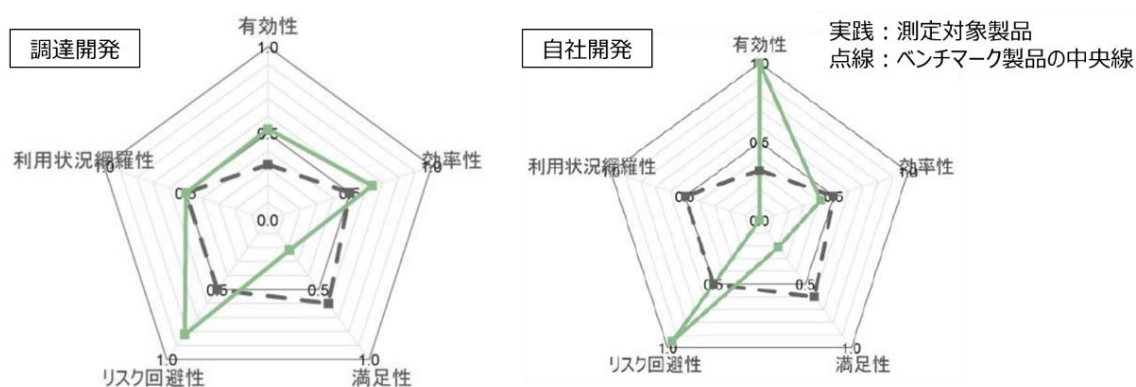


図4 利用時品質のWSQB ベンチマーク測定結果

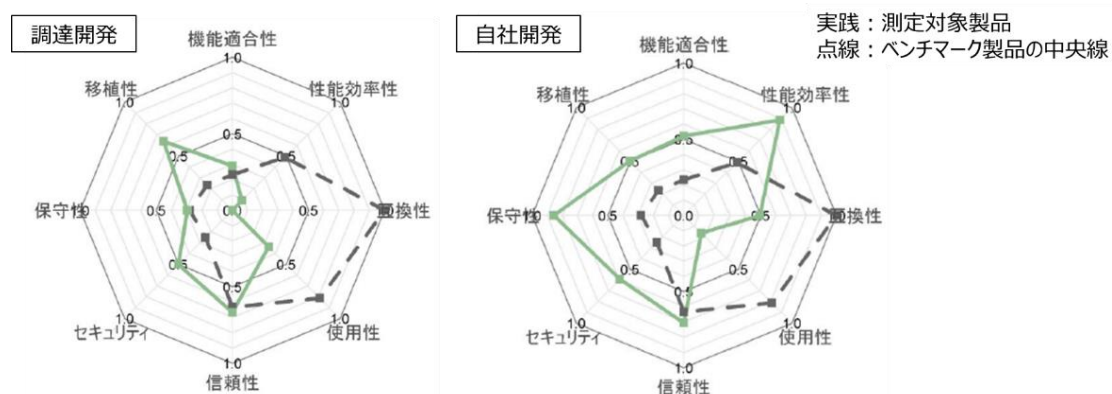


図5 製品品質のWSQB ベンチマーク測定結果

利用時品質に対するWSQB ベンチマーク測定結果の総評は、「調達開発」製品、「自社開発」製品

ともに、ベンチマーク測定者からどちらもおおむね良好と判定された。製品品質に対する WSQB ベンチマーク測定結果の総評は、「自社開発」製品はおおむね良好であるのに対して、「調達開発」製品は性能効率性、互換性のスコアが低いと判定された。

5.2 WSQB ベンチマークの測定結果からの考察

WSQB ベンチマークの測定結果より、能力良好と判定されたものについては世の中の製品と比較しても高い品質を保っていると判断できる。しかし、一部の品質特性は他社と比較してスコアが低くなった。低くなった理由についての考察は以下で述べる通り、「システム分析表」で分析はできていたが、テスト環境が不足して実際にテストできなかつたものや、製品の特性により優先度を低く設定したことに起因している。さらに、これまでシステムテストを作成したことがない担当者にテスト計画を実施してもらった結果、テスト計画経験者とはほぼ同等なテスト項目を計画することができた。また、今回「システム分析表」から作成したテスト項目を実施したところ、システムテスト全体のバグ件数に対して 18%のバグを検出することができた。

このことから、「システム分析表」を用いて分析を行い、それに基づいてシステムテストを実施することで、世の中の製品と比較しても高い品質を確保に貢献できたと言える。

- 「調達開発」製品の性能効率性のスコアが低くなった理由は、WSQB ベンチマーク測定の中で定義されている性能評価試験のうち、CPU 使用率、メモリ使用量の試験が実施できていないことに起因している。実施できなかつた理由は、評価環境の準備が十分できなかつたことにある。なお、「システム分析表」では、性能効率性に定義している CPU 負荷とメモリ負荷の観点を明記しており、期待値も記入できていてテスト項目への落とし込みはできていた。
- 「調達開発」製品の互換性のスコアが低くなった理由は、他システムの接続は今後の踏まえても予定がなく、システムの組合せが限定されていることから、当初より優先度を低く設定していたため問題ない。
- 「調達開発」製品と「自社開発」製品の使用性のスコアが低くなった理由は、今回の施策で、使用性のような利用者の感覚に依存した製品特性については、十分に検討しきれていないためである。
- 「自社開発」製品の利用状況網羅性のスコアが低くなった理由は、対象とした製品が特定のユーザーのみに提供する製品だったため、優先度を低く設定したため問題ない。

5.3 課題の解決度合

WSQB メトリクスの測定結果、他社実績と比較しても品質面において良好であることが実証された。このため、2章で示した課題に対する、3章で示した仮説に基づく、4章で提案したシステム検証技術は、「異なる開発背景を持ち、具体的な開発プロセスがブラックボックスなソフトウェアを利用して構築されたシステムの品質確保」に対しても有効と判断できる。ただし、WSQB ベンチマークにより認識した課題は、今後も継続して手法の改善が必要である。

- ・ 利用時品質：要件定義の時点で利用時品質の品質特性の観点で「妥当性確認」を十分に行うことで、「検証」におけるリスクを軽減できる
- ・ 製品品質：システム全体に今回提案した「システム分析表」を活用して分析をすれば、異なる

開発背景を基システムでも「検証」および「妥当性確認」の観点が抽出できる。

6. まとめ

製品品質においては今回提案した「システム分析表」のように品質特性ごとにシステムを構成する要素に対して分析を行うしくみや施策を適用した結果、複雑かつブラックボックスなシステムにおける品質を自分たちで保証できることが実証できた。

そして、「システム分析表」を活用して、システム要件をシステムの期待値として明確化することで利用時品質も高めることができることが分かった。

今後の取り組みは以下の3点が挙げられる。

- 今回の施策で、「使用性」のような利用者の感覚に依存した製品特性については、十分に検討しきれていない。これらも「システム分析表」で分析できるような手法の検討を継続する
- 利用時品質もレビューの強化だけでなく、プロトタイピングを含め、利用者の要求に依存した指標の中から共通事項を抽出、製品品質と同様に「システム分析表」でマトリクス化できる手法を検討していく
- 今回は「調達開発」と「自社開発」と独立して分析していた。今後は「調達開発」と「自社開発」両方を組み合わせた開発についても対応できるように検討する

謝辞 本稿の執筆にあたり、(株)富士通ゼネラル 上田担当部長(2019.3まで富士通研究所時在籍)に多くの御助言を頂いた。ここに、謝意を表する。

参考資料・引用文献

- [1] 日刊電気通信社、Anthropithecus electrojin、63(1)、p33-38、2019
- [2] 総務省、情報通信白書平成30年版、第1部 第1節2、2018
- [3] 東 基衛、システムとソフトウェアの品質：1. 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 55 巻』、p4-9、2013
- [4] 鷺崎 弘宜、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、<https://www.ipa.go.jp/sec/reports/20170420.html>、2017
- [5] 独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター、組込みソフトウェア向け開発プロセスガイド、p62-130、2007
- [6] 小島 嘉津江・森田 純恵・廣瀬 竹男・若本 雅晶・菊池 慎司・椋 晃歓・鷺崎 弘宜、ソフトウェア品質技術が品質特性に与える効果の見える化とその検証、『SEC journal』、p50-57、2018
- [7] 大橋 恭子・麻岡 正洋・笹川 文義・若杉 賢治・難波 功、テスト対象と実行条件を考慮したテストタイプに基づくシステムテスト計画書導出手順の開発、研究報告ソフトウェア工学 (SE) 16、p1-8、2013
- [8] 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、非機能要求グレード 2018、<https://www.ipa.go.jp/sec/softwareengineering/std/ent03-b.html>、2019