

# 開発形態を考慮した企業内OSS事前品質評価手法

中野 大扉<sup>1,a)</sup> 松本 卓大<sup>1,b)</sup> 山下 一寛<sup>1,c)</sup> 亀井 靖高<sup>1,d)</sup> 鷗林 尚靖<sup>1,e)</sup> 高山 修一<sup>2</sup>  
岩永 裕史<sup>2</sup> 岩崎 孝司<sup>2</sup>

**概要：**企業がオープンソースソフトウェア（OSS）を製品開発に利用することが増えている。OSSの品質が原因で製品自体の品質を低下させてしまう事態を回避するために、OSSの品質を事前に把握することが望ましい。本研究では、OSSの事前品質評価手法における重み付けを系統的に行うための第一歩として、品質に対する開発者のアンケート結果に基づいて重み付けを行い、その効果を実証的に調べる。本研究では、1) 開発者がOSSに求める品質はどのようなものかについて、有識者の協議と開発者へのヒアリングによる調査を行う（アンケート項目の作成）。開発形態ごとに重要視する評価項目が異なるなら、開発形態を考慮した評価が可能であると考えられるため、2) 開発形態ごとに重要視する品質はどのようなものかについて、開発者にアンケート調査を行う（重み付けデータの取得）。アンケート調査から、開発形態ごとに重要視する評価項目に差異があれば、アンケートをもとに重みを算出し、3) 開発形態を考慮した評価方法の調整を行う（重み付けによる事前品質評価）。OSSの事前品質評価を企業に導入する際の知見、及び、課題を明らかにするために、富士通九州ネットワークテクノロジーズ株式会社（以降、富士通QNET）の開発プロジェクトで利用したOSSを題材とした実証実験を行った。結果として、1) 23項目の品質要求を抽出し、2) 開発目的の違いによって「重要視する」、「やや重要視する」と答えられた割合に30%以上の差がある項目は3つ、OSSの導入方法の場合は5つ存在し、3) 評価方法の調整の結果、問題ありOSSの平均スコアは2.38、問題なしOSSの平均スコアは3.10であり、問題ありOSSのスコアの方が低かった。

## 1. はじめに

ソフトウェア開発企業がOSSを製品開発に利用する事例が増えてきている。全国のソフトウェア会社や情報処理サービス会社のうち、66.8%がOSSを利用し、51.6%が顧客向けのシステムにOSSを利用している[10]。OSSを利用する利点として「低価格で顧客に提供できること」や「多くの種類のOSSから自社にあったものを利用できること」などがあげられる[10]。

OSSの利用が必ずしも低価格化や品質向上に繋がるわけではなく、OSSの品質が原因となり製品自体の品質を低下させてしまうこともある。例えば、OSSにバグが混入していることが開発途中で判明した場合、OSSには「緊急時の技術的サポートが得にくい」や「バグの改修や顧客からの要請対応に手間がかかる」という欠点が存在する[10]ため、バグ修正に想定外のコストがかかり、製品自体の品質

を低下させてしまうことがある。

OSSの品質に起因するOSS導入のコスト増加を防ぐため、OSS自体の品質を事前に明らかにすることが望ましい。そのためOSSの事前品質評価に関する研究が、これまでに取り組まれてきた[1,3-5,8,9]。一般的なOSSの事前品質評価は、評価項目に対応する評価指標を取得し、スコアリングすることで評価を行っている。スコアリングの際には、開発形態やドメインなどの特徴を反映した重み付けを評価指標に対して行う。

本研究では、重み付けを系統的に行うための第一歩として、品質に対する開発者のアンケート結果に基づいて重み付けを行い、その効果を実証的に調べる。本研究では、1) 開発者がOSSに求める品質はどのようなものかについて、有識者の協議と開発者へのヒアリングによる調査を行う（アンケート項目の作成）。開発形態ごとに重要視する評価項目が異なるなら、開発形態を考慮した評価が可能であると考えられるため、2) 開発形態ごとに重要視する品質はどのようなものかについて、開発者にアンケート調査を行う（重み付けデータの取得）。アンケート調査から、開発形態ごとに重要視する評価項目に差異があれば、アンケートをもとに重みを算出し、3) 開発形態を考慮した評価方法

<sup>1</sup> 九州大学

<sup>2</sup> 富士通九州ネットワークテクノロジーズ

a) 1TE13161E@s.kyushu-u.ac.jp

b) 2IE15096T@s.kyushu-u.ac.jp

c) yamashita@posl.ait.kyushu-u.ac.jp

d) kamei@ait.kyushu-u.ac.jp

e) ubayashi@ait.kyushu-u.ac.jp

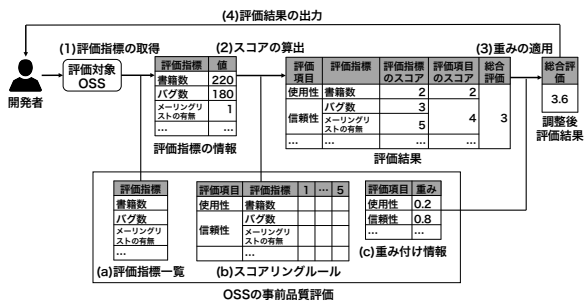


図 1 既存の取り組みにおける OSS 事前品質評価の方法

の調整を行う（重み付けによる事前品質評価）。

本研究では、OSS の事前品質評価を企業に導入する際の知見、及び、課題を明らかにするために、富士通九州ネットワークテクノロジーズ株式会社（以降、富士通 QNET）の開発プロジェクトで利用した OSS を題材とする。また、富士通 QNET 内のプロジェクトにおいてヒアリングとアンケートを行った。さらに、重み付けを行うことによって、富士通 QNET 内で過去に問題が発生した OSS をうまく評価できているかを調べるために、富士通 QNET の OSS 利活用データを利用した。OSS 利活用データには、開発プロジェクトが利用した OSS や開発中に OSS が起因となって発生した問題の情報が蓄えられている。

以降、2 章では、本研究における背景として関連研究を述べる。3 章から 5 章では、本研究において行った調査について述べる。具体的には、3 章では、開発者が OSS に求める品質はどのようなものかの調査について述べる。4 章では、開発形態ごとに重要視する品質の調査について述べる。5 章では、開発形態を考慮した評価方法の調整について述べる。6 章では、調査結果に対して行った開発者によるフィードバックについて述べる。7 章では、本研究のまとめと今後の課題について述べる。

## 2. 研究背景

OSS の事前品質評価手法として、Open Source Maturity Model (OSMM) [3], Open Business Readiness Rating (OpenBRR) [9], Qualification and Selection of Open Source software (QSOS) [1], Open BQR [8], OpenSource Maturity Model (OMM) [5], RepOSS [4] などが存在する。OSS の事前品質評価手法は、導入の容易さや使用しやすさといった評価項目ごとの指標群が定義されており、評価指標に基づき評価項目ごとに OSS のスコアを算出して OSS の品質を評価する。

OSS の事前品質評価手法を評価する取り組みも存在する [2, 6, 7]。例えば、Depez と Alexandre の研究 [2] では QSOS と OpenBRR のスコアリング手順と評価基準を比較し、各手法の利点と欠点をまとめている。Petrijnja らの研究 [6] では、OpenBRR, QSOS, OMM の 3 つの評価モデルを実証的に評価する実験を行っている。この実験では、3

つの評価モデルを Firefox と Chrome という 2 つの OSS に適用して評価している。結果として、3 つの評価モデルの評価結果は非常に類似していることがわかった。Petrijnja と Succi による研究 [7] では、OMM を用いて 6 つの OSS を評価し、OMM は効果的であると示した。従来研究に対して本研究では、OSS の事前品質評価手法を企業内にて適用する点、特にアンケートによる重み付けの効果を実証的に適用する点に新規性がある。

既存の OSS の事前評価手法が行っている評価方法を図 1 をもとに説明する。開発者は、事前品質評価を行うための指標が示されている「(a) 評価指標一覧」に基づき評価対象 OSS から (1) 評価指標の取得を行う。開発者は、評価指標の値からスコアを算出するためのルールが書かれた。「(b) スコアリングルール」に基づき (2) スコアの算出を行い評価結果を作成する。開発者は、開発形態を考慮した評価方法の調整を行うための、「(c) 重み付け情報」を用いて (3) 評価結果に重みの適用を行うことで調整後評価結果を算出する。開発者は、調整後評価結果を (4) 評価結果の出力として得ることで、評価対象 OSS の品質について判断を行う。

本研究では、図 1 の「(c) 重み付け情報」を富士通 QNET での開発形態を考慮するために、開発者にアンケートを実施する。プロジェクトの開発形態ごとに重要視する品質に特徴が見られれば、図 1 の (3) のように評価方法の調整を行う必要があると考えられる。開発形態としては、開発目的や OSS の導入方法、開発する製品がどのようなものか、利用する OSS の種類などが考えられるが、本研究では、初期調査として開発目的、OSS の導入方法の 2 つに限定する。算出した重みを実際の開発現場で利用された際に問題が発生した OSS と問題が発生しなかった OSS に適用することで開発形態を考慮した評価方法の調整を行う。

## 3. 開発者が OSS に求める品質はどのようなものか（アンケート項目の作成）

### 3.1 概要

開発形態を考慮した事前品質評価手法の実施のために、本研究ではアンケート調査によって重み付けを行う。アンケートを実施する上で、実施する組織の開発形態をより反映したアンケート項目を作成するため、開発者らにヒアリングを行ってアンケート項目の作成を行う。

### 3.2 アプローチ

本研究では、開発者が OSS に求める品質の調査として、富士通 QNET 社内の有識者による協議と開発者へのヒアリングを行った。

有識者による協議。開発者が OSS に求める品質を富士通 QNET 内の開発経験が 10 年以上の有識者 3, 4 名の協議によって抽出した。

表 1 開発者へのヒアリング結果の一例

プロジェクト番号	開発目的	導入方法	OSS に求める品質
1	研究試作	ブラックボックス	メーリングリストやコミュニティなど、不具合情報などの監視が容易。
2			OSS のユーザが多く、Web 上に使い方が豊富にあるもの。
3			ドキュメントやサンプルコードが豊富にある。性能測定が容易にできる。
4			インターフェースに関する情報がある。正常系が問題なく動作する。
5			インストールが容易。API が充実している。
6			サンプルコードが充実している。性能測定が容易にできる。
7	商用開発	ホワイトボックス	OSS がリリースされて相当な時間が経っている。ドキュメントや書籍が揃っている。
8		ブラックボックス	OSS 採用時に高い品質の OSS を求めている。高い品質の判断基準として OSS の導入事例が多いことを重視している。

表 2 本研究で作成した品質要求（評価項目）

分類	番号	評価項目	有識者	プロジェクト番号									
				1	2	3	4	5	6	7	8		
機能適合性	1	OSS 自体が実現したい機能を満たしている	✓										
	2	OSS のリリースが迅速で早期に機能を利用できる		✓									✓
	3	正常系が正しく動作しており、早期に機能を実現できる				✓	✓			✓			
性能効率性	4	OSS 自体が想定されるリソース（CPU、メモリ、ディスク等）で動作する	✓										
	5	OSS が指定された性能を満たすことができる											✓
	6	性能測定のための環境作りが容易にできる				✓	✓			✓			
互換性	7	OSS が他の OSS やシステム内の機能と共存して利用可能である	✓										
使用性	8	OSS 自体の操作性が利用者の要求を満たしている	✓										
	9	OSS のインストールが容易にできる							✓				
	10	OSS を組み込む際に容易に実施できる	✓										
	11	API が豊富で容易に組み込める							✓				
	12	OSS 自体にドキュメントや書籍が揃っていて、十分な情報が得られる				✓	✓			✓	✓	✓	
	13	OSS を利用したサンプルコードが公開されている					✓						
信頼性	14	OSS 利用例が多い			✓								✓
	15	OSS 自体がリリースされ、相当の期間が経っている										✓	
	16	サポートするコミュニティがしっかりしていて、保守されている	✓										
	17	OSS のバグ情報や対応状況が把握できる		✓									
セキュリティ	18	OSS 自体の品質が良い											✓
	19	脆弱性が少ない	✓										
	20	信頼できるサイトで公開されている	✓										
保守性	21	OSS のプログラム構造が容易でホワイトボックス化しやすい	✓										
	22	ソースコード内のコメントが豊富で読解性が高い	✓										
移植性	23	他の代替可能な OSS がある	✓										

開発者へのヒアリング。富士通 QNET 内の 8 プロジェクトに対してヒアリングを行った。ヒアリングは 1 プロジェクトに対して約 30 分行った。ヒアリング内容は、1) プロジェクトの開発形態、2) OSS に求める品質、の 2 点に関して行った。開発形態のうち、開発目的は「商用開発」、「研究試作」のいずれかである。OSS の導入方法は、OSS のソースコードを理解した上で利用する「ホワイトボックスとして利用」とソースコードを理解せずに利用する「ブラックボックスとして利用」のいずれかである。

### 3.3 結果と考察

有識者の協議によって 11 項目の品質要求を得られた。その一例として、「脆弱性に関する報告が少ない」、「ソースコード内のコメントが豊富で読解性が高い」などの品質要求があげられた。

開発者へのヒアリング結果の一例を表 1 に示す。ヒアリング対象プロジェクトの内、開発目的が研究試作のものは全てブラックボックスとして OSS を利用していた。これ

らのプロジェクトが OSS に求める品質は、研究試作の場合は、「正常系が正しく動作しており、早期に機能を実現できる」、「サンプルコードが充実している」といった OSS を使用する際の容易さに関するものが多く見られた（プロジェクト 3, 4, 5, 6）。一方、商用開発の場合は「OSS がリリースされて相当な時間が経っている」、「OSS 利用例が多い」などの OSS の信頼性に関するものが多かった（プロジェクト 7, 8）。

有識者による協議の結果とヒアリングの結果をもとに 23 個の品質要求を抽出し、分類を行った。抽出した品質要求を表 2 に示す。表 2 の右部分は抽出された品質要求がどの調査から抽出されたものかを示している。これらの品質要求を本研究における OSS 事前品質評価の評価項目とした。品質要求の内容は、表 2 の 8 番から 14 番のような OSS の導入や利用の容易さに関する要求が多い傾向にある。また、21 番や 22 番のような要求は、OSS のソースコードを理解して利用する場合にのみ必要となる要求であり、これらの品質要求はソースコードを理解せずに OSS を利用す

表 3 開発形態の内訳

開発目的	導入方法	アンケート件数	割合
商用開発	ホワイトボックス	9	30%
	ブラックボックス	12	40%
研究試作	ホワイトボックス	2	7%
	ブラックボックス	7	23%

際には必要とされない。

調査結果から、開発者が OSS に求める品質を 23 項目抽出した。23 項目のうち 7 項目が OSS の導入や利用の容易さに関する要求だった。

## 4. 開発形態ごとに重要視する品質はどのようなものか（重み付けデータの取得）

### 4.1 概要

従来の OSS 事前品質評価に関する研究 [1, 3-5, 8, 9] において、取得した評価指標を画一的に用いるのではなく、開発形態に応じて重み付けすることが推奨されている。本研究では、その重み付けのために、開発者に対して実施するアンケート結果を活用する。本研究では、開発者が各評価項目をどれだけ重要視しているかについて富士通 QNET 内の開発者にアンケートを行うことで、開発形態ごとに重要視する品質について調査する。

### 4.2 アプローチ

本研究では、開発形態ごとに重要視する品質の調査として富士通 QNET 内のプロジェクトマネージャー 30 人を対象にアンケートを行った。アンケート内容は、1) プロジェクトの開発形態、2) 評価項目の重要度の 2 点である。このアンケートから、各開発形態において重要視する品質に違いがあるかを調査する。

アンケートの回答方法は、1) プロジェクトの開発形態は、開発目的が「商用開発」か「研究試作」の 2 択と、OSS の導入方法が「ホワイトボックスとして利用」と「ブラックボックスとして利用」の 2 択である。2) 評価項目の重要度は、a) 評価項目ごとの重要度、b) 重要度が最上位の評価項目の 2 つの質問を行った。a) 各評価項目に対してどれだけ重要視しているかは、「重要視する」、「やや重要視する」、「やや重要視しない」、「重要視しない」の 4 択で回答する質問である。b) 重要度が最上位の評価項目は、表 2 に示した 23 の評価項目から重要度が高い項目を上位 5 つまでを選ぶ質問である。

### 4.3 結果と考察

アンケートの開発形態の内訳を表 3 に示す。開発形態で 1 番多い組み合わせは商用開発-ブラックボックスで 12 件 (40%) であり、1 番少ない組み合わせは研究試作-ホワイトボックスで 2 件 (7%) であった。以下、評価項目の重要

OSSが他のOSSやシステム内の機能と共存して利用可能である

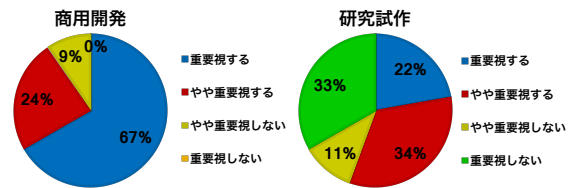


図 2 評価項目 7 番のアンケート結果

性能測定のための環境作りが容易にできる

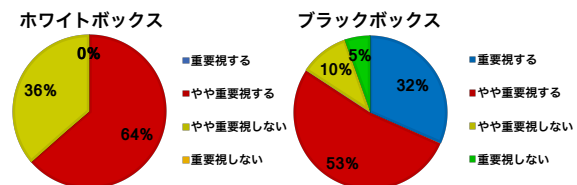


図 3 評価項目 6 番のアンケート結果

度のアンケート結果を示す。

各評価項目の重要度。開発者が「重要視する」または「やや重要視する」と答えた割合を比較する。評価項目 1 番 (OSS 自体が実現したい機能を満たしている) は、全ての開発形態で「重要視する」または「やや重要視する」と答えられた割合が 100%であった。プロジェクトの開発目的で比較すると、図 2 に示す評価項目 7 番 (OSS が他の OSS やシステム内の機能と共存して利用可能である) は、商用開発では 91%、研究試作では 56%、4 番 (OSS が想定されるリソース (CPU、メモリ、ディスク等) で動作する) は、商用開発では 100%、研究試作では 66%であり、商用開発で重要視されている傾向が見られた。一方で、評価項目 11 番 (API が豊富で容易に組み込める) は、商用開発では 52%、研究試作では 78%であり、研究試作で重要視されている傾向があることがわかった。

表 4 は開発形態によって「重要視する」または「やや重要視する」と答えられた割合の差が 20%以上の評価項目を示している。表 4 に示すように、開発目的によって「重要視する」または「やや重要視する」と答えられた割合に 30%以上の差がある項目は 3 項目、20%以上かつ 30%未満の差がある項目は 3 項目存在した。

OSS の導入方法で比較すると、評価項目 21 番、22 番のようなソースコードに関する項目に加え、評価項目 23 番 (他に代替可能な OSS がある) は、ホワイトボックスでは 45%、ブラックボックスでは 10%であり、ホワイトボックスとして利用しているプロジェクトで重要視されている傾向がある。一方で、図 3 に示す評価項目 6 番 (性能測定のための環境作りが容易にできる) は、ホワイトボックスでは 64%、ブラックボックスでは 85%、評価項目 7 番 (OSS が他の OSS やシステム内の機能と共存して利用可能である) は、ホワイトボックスでは 64%、ブラックボックスで

表 4 開発形態による重要度の差が 20%以上の評価項目

評価項目	商用開発	研究試作	差
4	100%	67%	-33%
7	91%	56%	-35%
8	76%	44%	-32%
9	67%	89%	22%
11	53%	78%	25%
19	100%	78%	-22%
評価項目	ホワイトボックス	ブラックボックス	差
6	64%	85%	21%
7	64%	89%	25%
9	91%	63%	-28%
10	100%	68%	-32%
13	73%	95%	22%
15	36%	68%	32%
21	55%	16%	-39%
22	55%	21%	-34%
23	45%	10%	-35%

は 89%であり、ブラックボックスとして利用しているプロジェクトで重要視されている傾向が見られた。

表 4 に示すように、OSS の導入方法によって「重要視する」または「やや重要視する」と答えられた割合に 30%以上の差がある項目は 5 項目、20%以上かつ 30%未満の差がある項目は 4 項目存在した。

重要度が最上位の評価項目。上位 5 つに選ばれた回数をカウントし、回数が多い順に上位 5 位までの評価項目を表 5 に示す。プロジェクトの開発目的で比較すると、開発者は評価項目 1 番（OSS 自体が実現したい機能を満たしている）と評価項目 3 番（正常系が正しく動作しており、早期に機能を確認できる）をどちらも重要視していることがわかる。OSS の導入方法で比較すると、上位 5 つの評価項目のうち 4 つが同じであることから導入方法による重要度が最上位の評価項目はほとんど見られなかった。このことから、どの開発形態でも重要度が最上位の評価項目は共通していることがわかった。

開発目的の違いによって「重要視する」、「やや重要視する」と答えられた割合に 30%以上の差がある項目は 3 つ、OSS の導入方法の場合は 5 つ存在した。

## 5. 開発形態を考慮した評価方法の調整（重み付けによる事前品質評価）

### 5.1 概要

4 章において開発形態ごとに開発者が各評価項目をどれだけ重要視しているかを調査した。この調査結果から、開発形態によって重要視する評価項目には、重要度が最上位の評価項目は共通しつつも、各々の評価項目の重要度については開発形態によって差異が見られた。本研究では、4 章の結果をもとに開発形態ごとに評価項目に対する重みを

表 5 上位 5 つに選ばれた回数が多い評価項目

商用開発の順位		
1 位	1 番	OSS 自体が実現したい機能を満たしている
2 位	18 番	OSS 自体の品質が良い
3 位	3 番	正常系が正しく動作しており、早期に機能を確認できる
4 位	5 番	OSS が指定された性能を満たすことができる
5 位	4 番	OSS 自体が想定されるリソースで動作する
研究試作の順位		
1 位	1 番	OSS 自体が実現したい機能を満たしている
2 位	3 番	正常系が正しく動作しており、早期に機能を確認できる
同率 3 位	10 番	OSS を組み込む際に容易に実施できる
同率 3 位	13 番	OSS を利用したサンプルコードが公開されている
5 位	12 番	OSS 自体にドキュメントや書籍が揃っていて、十分な情報が得られる
ホワイトボックスの順位		
1 位	1 番	OSS 自体が実現したい機能を満たしている
同率 2 位	18 番	OSS 自体の品質が良い
同率 2 位	3 番	正常系が正しく動作しており、早期に機能を確認できる
4 位	4 番	OSS 自体が想定されるリソースで動作する
5 位	5 番	OSS が指定された性能を満たすことができる
ブラックボックスの順位		
1 位	1 番	OSS 自体が実現したい機能を満たしている
2 位	3 番	正常系が正しく動作しており、早期に機能を確認できる
3 位	18 番	OSS 自体の品質が良い
同率 4 位	5 番	OSS が指定された性能を満たすことができる
同率 4 位	12 番	OSS 自体にドキュメントや書籍が揃っていて、十分な情報が得られる

算出し、適用することで開発形態を考慮した評価方法の調整を行った。

本研究では、重み付けによる事前評価の効果を調べるために、富士通 QNET 内の OSS 利活用データを用いる。OSS 利活用データには、富士通 QNET で OSS を利用した際に OSS が起因となって発生した問題が記載されている。

対象 OSS プロジェクトは、4 章にてアンケート対象となった OSS のうちトラブルシューティング事例が存在する OSS（以降、問題あり OSS）3 件を対象とした。トラブルシューティング事例が存在しない OSS（以降、問題なし OSS）はアンケート対象の OSS から各開発形態ごとに無作為に 2 件選び、評価対象とした。ただし、研究試作－ホワイトボックスに関しては、アンケート対象となった問題なし OSS が 1 件しかなかったため 1 件としている。

### 5.2 アプローチ

本研究では、アンケート結果に基づく重み付けによる評価方法の調整を行う。上記の調査を行うためには、評価項目のスコアと開発形態ごとの重みが必要となる。以下に評価項目のスコア算出方法と重みの算出方法を示す。

評価項目のスコア算出方法。評価項目のスコアを算出するためには、図 1 に示すスコアリングルールが必要となる。スコアリングルールは、マッピング情報（表 6 に示すような評価項目と評価指標を対応付けるための情報）とスコア情報（表 7 に示すような評価指標の値をスコアに変換するための情報）に分けられる。

本研究にて用いるマッピング情報は次のように作成し

表 6 マッピング情報の一例

評価項目	評価指標
OSSのバグ情報や対応情報が把握できる	バグトラッキングシステムの有無
	メーリングリストの流量
ソースコード内のコメントが豊富で読解性が高い	コメント行数
	コード行数に占めるコメント行割合

表 7 スコア情報の一例

評価指標	1	2	3	4	5
月ごとの平均コミット数	43 未満	43 以上 74 未満	74 以上 120 未満	120 以上 158 未満	158 以上
パッケージインストーラの有無	無し	-	-	-	有り

た。評価項目は3章にて作成したものを利用する。対応付ける評価指標の候補は既存の評価モデルで利用されているものを対象とした。各評価項目に対応する評価指標は著者たちの協議によって決定した。

スコア情報は評価指標を数値データと2値データに分けて作成した。数値データは指標の値が数値のものであり、例として月ごとの平均コミット数や関連書籍数があげられる。数値データの指標は富士通 QNET 内で利用された120件のOSSの指標値の分布に基づきスコアを決定する。月ごとの平均コミット数の場合、表7内に示すように、評価指標の数値に応じてスコアが決定される。2値データの例としてメーリングリストの有無やパッケージインストーラの有無があげられる。2値データの指標は、「有り」の場合は5、「無し」の場合は1とする。

例外として、評価項目に対応する評価指標が全て取得不可能であった場合、その評価項目のスコアを評価結果に考慮しないこととしている。評価指標が取得不可能な場合の例として、バグトラッキングシステムを導入していないOSSの場合はバグ数を取得不可能であるといった場合が存在する。

**重みの算出方法。**4章において行ったアンケート結果をもとに評価項目ごとの重みを算出する。アンケート結果を得点化し、その得点をもとに重みを算出する。各評価項目の重要度のアンケート結果を「重要視する」、「やや重要視する」、「やや重要視しない」、「重要視しない」をそれぞれ4点、3点、2点、1点とする。これらの得点を合計し、各評価項目の得点とする。全評価項目数を  $n$  とする時、 $i$  番目の評価項目の得点を  $S_i$ 、重み  $W_i$  は次の式によって算出する。

$$W_i = S_i / \sum_{j=0}^n S_j \quad (1)$$

重要度が最上位の評価項目に関するアンケート結果は、4章において開発形態によらず共通していることがわかったことから、重みには反映させないこととする。

重みは、開発形態を開発目的と導入方法の両方を考慮するもの、開発目的のみを考慮するもの、導入方法のみ考慮するものの3つから算出する。

表 8 各開発形態の評価結果

商用開発-ホワイトボックス	商用開発-ホワイトボックス	商用開発	ホワイトボックス	
問題なし OSS	OSS-A	3.02	2.99	3.03
	OSS-B		2.91	2.92
商用開発-ブラックボックス	商用開発-ブラックボックス	商用開発	ブラックボックス	
問題あり OSS	OSS-C	2.04	2.04	2.03
	OSS-D	2.69	2.72	2.72
問題なし OSS	OSS-E	3.67	3.63	3.67
	OSS-F	2.77	2.78	2.77
研究試作-ホワイトボックス	研究試作-ホワイトボックス	研究試作	ホワイトボックス	
問題あり OSS	OSS-G	2.41	2.43	2.42
問題なし OSS	OSS-H	3.06	3.12	3.05
研究試作-ブラックボックス	研究試作-ブラックボックス	研究試作	ブラックボックス	
問題なし OSS	OSS-I	3.14	3.12	3.14
	OSS-J	3.14	3.11	3.18

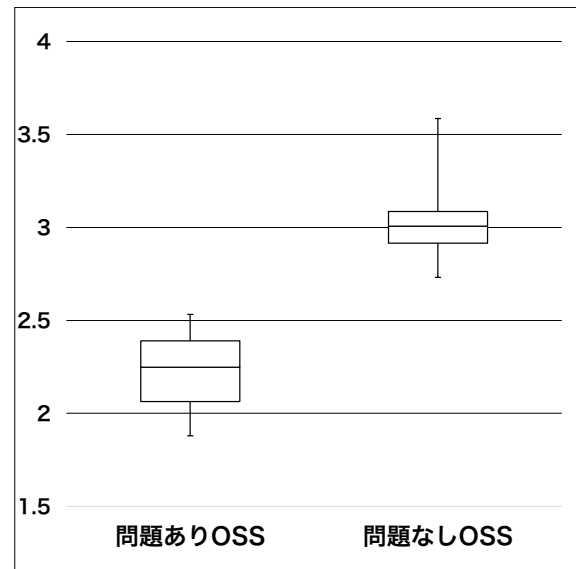


図 4 問題の有無によるスコア分布の差

### 5.3 結果と考察

重み付けを適用した評価結果については表8に示す。また、問題ありOSSと問題なしOSSの評価スコアの分布を図4に示す。図4の評価結果は、3つの重みのうち開発形態を開発目的と導入方法の両方を考慮したものを適用した場合の評価結果を示している。図4から、問題の有無によってスコア分布に差があることがわかる。また、問題ありOSSの平均スコアは2.38、問題なしOSSの平均スコアは3.10であり、問題ありOSSのスコアの方が低いことがわかる。

問題ありOSSであるOSS-Cに関して評価結果の分析を行った。OSS-Cの重みと評価結果の詳細を表9に示す。表9内で適用した重みは、商用開発-ブラックボックスのアンケート結果から算出した重みである。表9内の変化割合は、評価項目に重みを等分に割り当てた場合と比較した時の重みの変化を割合で示したものである。変化割合が10%以上の評価項目は、1番、3番、5番、14番、18番の5つであり、内4つが表5の重要度が最上位の評価項目として上位5つに選ばれた回数が多い項目に入っていることから、開発者が重要視する評価項目の重みが大きくなっていることがわかる。

表 9 OSS-C の重みと評価結果の詳細

分類	項目番号	評価項目	評価項目のスコア	重み	調整後スコア	変化割合	発生した問題との対応
機能適合性	1	OSS 自体が実現したい機能を満たしている	2.33	0.63	0.148	+27%	
	2	OSS のリリースが迅速で早期に機能を利用できる	2.00	0.034	0.068	-32%	
	3	正常系が正しく動作しており、早期に機能を実現できる	3.00	0.062	0.185	+23%	
性能効率性	4	OSS 自体が想定されるリソース (CPU、メモリ、ディスク等) で動作する	5.00	0.051	0.254	+2%	
	5	OSS が指定された性能を満たすことができる	5.00	0.055	0.275	+10%	
	6	性能測定のための環境作りが容易にできる	2.33	0.049	0.115	-2%	
互換性	7	OSS が他の OSS やシステム内の機能と共存して利用可能	3.00	0.054	0.162	+8%	
使用性	8	OSS 自体の操作性が利用者の要求を満たしている	2.33	0.047	0.111	-5%	
	9	OSS のインストールが容易にできる	1.00	0.044	0.044	-12%	
	10	OSS を組み込む際に容易に実施できる	1.00	0.050	0.050	0%	✓
	11	API が豊富で容易に組み込める	1.00	0.046	0.046	-8%	
	12	OSS 自体にドキュメントや書籍が揃っていて、十分な情報が得られる	1.00	0.052	0.052	+4%	✓
	13	OSS を利用したサンプルコードが公開されている	1.00	0.052	0.052	+4%	✓
信頼性	14	OSS 利用例が多い	2.33	0.057	0.132	+13%	
	15	OSS 自体がリリースされ、相当の期間が経っている	-	-	-	-	
	16	サポートするコミュニティがしっかりしていて、保守されている	2.20	0.041	0.090	-18%	
	17	OSS のバグ情報や対応状況が把握できる	1.00	0.054	0.054	+8%	
セキュリティ	18	OSS 自体の品質が良い	1.00	0.059	0.059	+18%	
	19	脆弱性が少ない	1.00	0.052	0.052	+5%	
	20	信頼できるサイトで公開されている	1.00	0.052	0.052	+5%	
保守性	21	OSS のプログラム構造が容易でホワイトボックス化しやすい	-	-	-	-	
	22	ソースコード内のコメントが豊富で読解性が高い	-	-	-	-	
移植性	23	他の代替可能な OSS がある	1.00	0.026	0.026	-48%	
合計			39.5	1.00	2.04	0%	

OSS-C の分類ごとのスコアを図 5 に示す。OSS-C において発生した問題の一例として、「ドキュメント等の情報不足によって把握できなかった内部の処理において問題が発生した」というものが存在する。富士通 QNET の開発者は OSS-C において発生した問題は、評価項目 10 番 (OSS を取り込む際に容易に実施できる)、12 番 (OSS 自体にドキュメントや書籍が揃っていて、十分な情報が得られる)、13 番 (OSS を利用したサンプルコードが公開されている) に起因する問題であるとした。表 9 から、発生した問題に対応する評価項目のスコアは、全て 1.00 と最低点である。さらに、発生した問題に対応する 3 つの評価項目は全て使用性に分類される評価項目であり、図 5 から OSS-C の使用性のスコアが低いことから、評価結果は発生した問題をうまく捉えられていると考えられる。

問題に対応する評価項目の変化割合は、平均 2.7% 大きくなっていることから、重みを等分に割り当てた場合に比べて重みが大きくなっていることがわかる。しかしながら、発生した問題には関係ない項目である評価項目 4 番、5 番のスコアは 5.00 と最高点であり、重みは平均 6% 大きくなっている。OSS-C は問題が発生しているにも関わらず、発生した問題に関係ないの評価項目の影響で評価スコアが高くなっていることがわかる。このことから、重み算出方法には改善の余地があると考えられる。

本調査結果から、評価結果のスコアだけでなく、各評価項目のスコアにおいて著しく低いものがないかを確認することが重要であると考えられる。図 5 の OSS-C の場合、性能効率性や互換性のスコアは高かったが、使用性や信頼

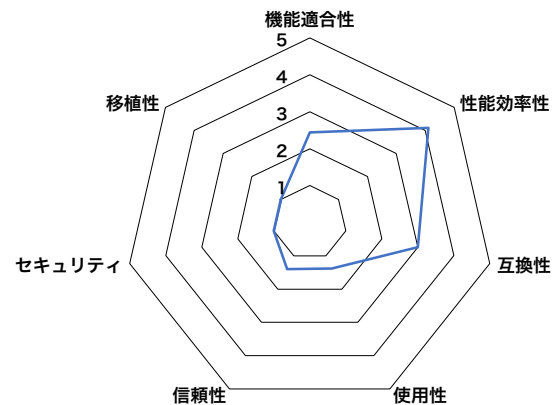


図 5 OSS-C の分類ごとのスコア

性、セキュリティのスコアが低く、導入にあたってリスクとして管理する必要があることを示している。OSS-C においては、このうちの使用性に関するリスクが問題として顕在化したと考えられる。

評価方法の調整の結果、問題あり OSS の平均スコアは 2.38、問題なし OSS の平均スコアは 3.10 であり、問題あり OSS のスコアの方が低かった。

## 6. 開発者との議論

本研究の調査結果をより理解するために富士通 QNET の開発者と議論を行った。

4 章のアンケート結果が、全評価項目で「重要視している」、「やや重要視している」と回答された割合の平均が

73%であったことを報告し、富士通 QNET の開発者のコメントを検討した結果、研究チームは評価項目の作成がうまくいっていると認識した。表5から、評価項目1番（OSS自体が実現したい機能を満たしている）や評価項目3番（正常系が正しく動作しており、早期に機能を確認できる）といった評価項目が開発形態によらず重要視されている事に関して富士通 QNET の開発者と議論した結果、研究チームはこれらの評価項目を重要視することは開発者として当然であると判断した。また、開発形態ごとに重要視する評価項目について富士通 QNET の開発者と議論した結果、研究チームは商用開発で重要視されている項目は、製品として運用する上で必要となる品質を重要視していると認識し、研究試作では OSS を製品に組み込む上での容易さ、使用する上での容易さを重要視していると認識した。

## 7. まとめと今後

本研究では、OSSの事前品質評価方法を企業で適用すること、特に、その重み付けを系統的に行うことを目標として、1) OSSに求める品質はどのようなものか（アンケート項目の作成）、2) 開発形態ごとに重要視する品質に差があるか（重み付けデータの取得）、3) 開発形態を考慮した評価方法の調整（重み付けによる事前品質評価）を行った。結果として、1) 有識者や開発者が OSS に求める品質を 23 項目抽出し、2) 開発目的の違いによって「重要視する」、「やや重要視する」と答えられた割合に 30%以上の差がある項目は 3 つ、OSS の導入方法の場合は 5 つ存在し、3) 評価方法の調整の結果、問題あり OSS の平均スコアは 2.38、問題なし OSS の平均スコアは 3.10 であり、問題あり OSS のスコアの方が低かった。

今後の課題として、4 章に行ったアンケートの件数が少

ないので、追加のアンケートを行うことがあげられる。さらに、重み算出の改善として、開発形態の分類を見直すことや重み付けの対象となる評価項目を見直すことも課題として考えられる。

## 参考文献

- [1] Atos: Qualification and selection of open source software (QSOS), version 2.0. (2013).
- [2] Deprez, J.-C. and Alexandre, S.: Comparing assessment methodologies for free/open source software: OpenBRR and QSOS, *International Conference on Product Focused Software Process Improvement*, Springer, pp. 189–203 (2008).
- [3] Duijnhouwer, F.-W. and Widdows, C.: Open Source Maturity Model, *Capgemini Expert Letter* (2003).
- [4] Northeast Asia, O.: RepOSS: A Flexible OSS Assessment Repository (2012).
- [5] Petrinja, E., Nambakam, R. and Sillitti, A.: Introducing the opensource maturity model, *Emerging Trends in Free/Libre/Open Source Software Research and Development, 2009. FLOSS'09. ICSE Workshop on, IEEE*, pp. 37–41 (2009).
- [6] Petrinja, E., Sillitti, A. and Succi, G.: Comparing OpenBRR, QSOS, and OMM assessment models, *IFIP International Conference on Open Source Systems*, Springer, pp. 224–238 (2010).
- [7] Petrinja, E. and Succi, G.: Assessing the open source development processes using OMM, *Advances in Software Engineering*, Vol. 2012, p. 8 (2012).
- [8] Taibi, D., Lavazza, L. and Morasca, S.: OpenBQR: a framework for the assessment of OSS, *IFIP International Conference on Open Source Systems*, Springer, pp. 173–186 (2007).
- [9] Wasserman, T. and Das, A.: Using FLOSSmole Data in Determining Business Readiness Ratings, *WoPDS* (2007).
- [10] 独立行政法人情報処理推進機構：第3回オープンソースソフトウェア活用ビジネス実態調査 (2010).