

修士論文

ソフトウェア開発における検証者からのフィードバック
に基づく開発者の貢献期間の予測

則兼 卓人

2017年2月2日

奈良先端科学技術大学院大学
情報科学研究科

本論文は奈良先端科学技術大学院大学情報科学研究科に
修士(工学) 授与の要件として提出した修士論文である。

則兼 卓人

審査委員：

松本 健一 教授	(主指導教員)
飯田 元 教授	(副指導教員)
Graham Neubig 助教	(カーネギーメロン大学)
伊原 彰紀 助教	(副指導教員)

ソフトウェア開発における検証者からのフィードバック に基づく開発者の貢献期間の予測*

則兼 卓人

内容梗概

本論文では、OSS (Open Source Software) プロジェクトにおいて、検証者(パッチを評価する開発者)からのフィードバック(パッチを改善するためのコメント)が開発者の貢献期間に与える影響を明らかにする。OSSはコミュニティによる保守運用などの支援なしでは機能しなくなる。特に長期貢献者(1年以上貢献した開発者)は新規貢献者の支援などを行っており、一部はバージョン管理システムへの変更権限を持っている。従って、長期貢献者はプロジェクトを成功に導く役割を担っていると言える。開発者を長期貢献者に導くためには、開発者がパッチ投稿によって貢献する動機を理解することが重要である。開発者へのアンケート調査から、検証者からのフィードバックが開発者の貢献期間に影響を与えるという回答が得られた。ただし、影響を与える検証者からのフィードバックの内容は明らかになっていない。本論文では検証者からのフィードバックを、パッチに対する内的要因・外的要因の指摘、パッチへの指摘理由、指摘に含まれるキーワードの3つの観点から開発者の貢献期間に与える影響を分析した。その結果、パッチに対する内的要因の指摘や不完全な修正であるという指摘理由、キーワード“good”を含む指摘を受けた開発者は長期的にパッチを投稿することがわかった。これまで開発者の貢献期間の予測に検証者からのフィードバックは考慮されていなかったため、本論文での結果により従来研究での開発者の貢献期間の予測精度の向上も期待される。

*奈良先端科学技術大学院大学 情報科学研究科 修士論文, NAIST-IS-MT1551076, 2017年2月2日.

キーワード

オープンソースソフトウェア, コードレビュー, 長期貢献者, パッチ投稿, 開発者の動機

Which reviewers' feedback motivate developers in becoming long-term contributors in software development?*

Takuto Norikane

Abstract

The purpose of this thesis is to determine the influence of reviewers' feedback on developers' contribution period. Open Source Software (OSS) cannot survive without contributions from the community. In particular, long-term contributors (LTCs) (e.g., committers), defined as contributors who spend at least one year on OSS projects, play a crucial role in a project's success because they would have permission to add (commit) code changes to a project's version control system. Moreover, LTCs become mentors to beginners in OSS projects. We distributed a questionnaire, and the results show that reviewers' feedback influence developers' continuous contribution. However, we do not know which kind of reviewers' feedback influence developers' continuous contribution. In this thesis, we analyzed the influence of the reviewers' feedback on the developer's continuous contribution based on three points; the viewpoint of the internal and external factors for the patch, the reasons for pointing out to the patch, and the keywords contained in the reviewers' feedback. As a results, we found that reviewers' feedback that include internal factors for patches, "incomplete fix" reason, and the keyword "good" influence developer's contribution period. Reviewers' feedback are currently not considered in prediction of LTCs, but we expect that reviewers' feedback can improve the prediction accuracy of developers' contribution period.

*Master's Thesis, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT1551076, February 2, 2017.

Keywords:

open source software, code review, long term contributors, patch submission,
developers ' motivation

目次

図目次	vii
表目次	vii
1. はじめに	1
2. OSS プロジェクトにおけるコードレビュー	3
2.1 コードレビュープロセス	3
2.2 コードレビュー管理システムの機能と利用目的	3
2.3 開発者の共同活動における課題	4
3. 分析手法	8
3.1 開発者が受ける指摘理由の分布の計測方法	8
3.2 検証者からの指摘理由に基づいた開発者の貢献期間の予測方法	9
3.2.1 指摘理由を用いた手法	9
3.2.2 パッチに対する内的要因と外的要因の指摘を用いた手法	9
3.3 開発者の貢献期間の予測に役立つ要因の特定方法	10
3.4 検証者からのフィードバックに含まれるキーワードに基づいた開発者の貢献期間の予測方法	11
4. ケーススタディ	13
4.1 データセット	13
4.2 結果	15
4.2.1 長期貢献者と短期貢献者の活動量と受けた指摘の違い	15
4.2.2 検証者からの指摘理由に基づいた開発者の貢献期間の予測精度と分析結果	19
4.2.3 検証者からのフィードバックに含まれるキーワードに基づいた開発者の貢献期間の予測精度と分析結果	23
5. 考察	28

5.1 議論	28
5.2 制約	29
6. 関連研究	30
6.1 開発者が貢献する動機	30
6.2 検証者からのフィードバック	30
7. おわりに	32
謝辞	33
参考文献	34
発表リスト	38

図 目 次

1	コードレビュー管理システムの一画面	5
2	Qt プロジェクトにおける貢献期間と開発者数	14
3	Openstack プロジェクトにおける貢献期間と開発者数	15
4	内的要因の指摘数	22
5	パッチ投稿数	22
6	検証者数	22
7	不採択票数	22
8	プロジェクト数	22
9	“Incomplete fix”	22

表 目 次

1	開発者へのアンケート	7
2	内的要因と外的要因	10
3	Qt システムの概要	13
4	Openstack システムの概要	14
5	指摘理由とサンプルの指摘, 長期貢献者・短期貢献者が受けた指摘	16
6	長期貢献者と短期貢献者の活動量の中央値 [Qt]	17
7	長期貢献者と短期貢献者の活動量の中央値 [Openstack]	18
8	指摘理由に基づいた開発者の貢献期間の予測結果	19
9	内的要因と外的要因の指摘理由に基づいた開発者の貢献期間の予 測結果	19
10	初期の3ヶ月間の情報をもとに, 開発者の貢献期間の予測結果 (検 証者からのフィードバックの有無による比較)[Qt]	20
11	Scott-Knott を用いた重要な要素の特定 [指摘理由]	21
12	Scott-Knott を用いた重要な要素の特定 [内的要因と外的要因の指 摘理由]	22
13	キーワードに基づいた開発者の貢献期間の予測結果 [Qt]	23

14	キーワードに基づいた開発者の貢献期間の予測結果 [Openstack] .	23
15	初期の3ヶ月間の情報をもとに、開発者の貢献期間の予測結果 (検証者からのフィードバックの有無による比較)[Openstack]	24
16	開発者がよく受けたキーワード [Qt]	24
17	開発者がよく受けたキーワード [Openstack]	25
18	キーワードが使用された例文	26

1. はじめに

Open Source Software (OSS) はコミュニティからの支援なしでは機能せず、コミュニティ内の地理的に分散している貢献者による共同作業によって、高品質で高機能な OSS 製品が開発される [30]. 特に 1 年以上貢献した開発者 (長期貢献者) は新規貢献者の支援などを行っており、一部はバージョン管理システムへの変更権限を持っている. このことから、長期貢献者はプロジェクトを成功に導く役割を担っていると言える. また開発者が、ソフトウェア開発の生産性を高め、プロジェクトの中心的な役割をこなせるようになるには、時間を要するということが知られている [6] [29]. ただし、開発者はボランティアとして参加することが多いため、1 年以内にプロジェクトを離脱してしまうケースが多い [3] [15] [23].

開発者のプロジェクト離脱を妨げるには、彼らのプロジェクト参加に対する動機を理解することが必要である. 一部の開発者は実装技術の向上が見込めるため、プロジェクトに参加する [4]. OSS プロジェクトでは、開発者は才能あるプログラマーからレビューを受けることができるからである. 他に OSS プロジェクトに参加する動機としては、他の開発者と協力して開発できることが挙げられる [4]. Wu らの研究では、OSS プロジェクトに開発者が滞在する動機についてのアンケート調査がなされた [27]. 参加者のモチベーションという観点では、いくつかの既存研究で新規貢献者の維持に焦点が当てられてきた [12] [22] [28]. 特に、Zhou と Mockus らは、新規貢献者の活動量から、その新規貢献者が長期的に貢献するかを予測するモデルを構築した [28]. また、新規貢献者の 1 ヶ月間ごとのタスク量という観点から小・中規模のプロジェクトを中心に実験が行われ、大きなプロジェクトの場合は、3 ヶ月以上の貢献履歴を要するだろうと主張している. ただし、他の開発者との共同作業に着目した検証者からのパッチを改善するためのコメント (フィードバック) による影響を定量的に評価している研究は少ない. 検証者からのフィードバックが開発者の貢献期間に与える影響について、実際にプロジェクトに 1 年以上貢献している開発者 20 名に事前調査を行なった. その結果、2 名の開発者から回答を得ることができ、そのうち 1 名の開発者は、検証者からのフィードバックが開発者の貢献期間に影響を与えると回答した.

本論文では、他の開発者との共同作業に着目し、検証者からのフィードバックに

基づいた開発者の貢献期間の予測を行う。長期貢献者を予測する理由は、その開発者にプロジェクト内での役割を1年以内に与えることによって長期的に貢献してもらえるからである [3]。さらに、検証者からのどのような内容のフィードバックが開発者の貢献期間に影響を与えるのかについて調査する。ケーススタディとして Qt プロジェクトと Openstack プロジェクトを対象に、開発者の貢献期間の予測モデルを構築し、実験を行なった。

続く2章では、OSS プロジェクトにおけるコードレビューについて説明し、3章では分析手法を述べ、4章ではケーススタディの結果を述べる。5章ではケーススタディについての考察を行い、6章で関連研究について述べる。最後に、7章で本論文のまとめを述べる。

2. OSS プロジェクトにおけるコードレビュー

2.1 コードレビュープロセス

OSS の品質を高く保つために、ある開発者が投稿したパッチ (バグ修正や機能追加のためのソースコード) の検証作業がある。近年ではその検証作業のために、Gerrit^{*1} や Review Board^{*2} などのコードレビュー管理システムが OSS プロジェクトや企業などで導入されている。このようなツールによるコードレビューの管理は Modern Code Review と呼ばれている [1]。Modern Code Review のプロセスは以下の通りである。

1. 開発者は、Github のような web ベースのバージョン管理システムからソースコードを自身の PC に収集する。続いて、開発者はパッチ (バグ修正や機能追加のためのソースコード) を作成し、web ベースのコードレビューツールにパッチを提出する。
2. 検証者 (パッチを評価する開発者) やコミッター (OSS にパッチを統合する権限を与えられた開発者) が、提出されたパッチについて検証する。
3. 検証の結果、パッチを修正する必要があるば、パッチ作成者はパッチを改善し再投稿する。
4. 検証者やコミッターによって投稿したパッチが採択されれば、コミッターはパッチをプロダクトに統合する。一方、もし検証者やコミッターによって不採択にされれば、コミッターはそのパッチを破棄する。

2.2 コードレビュー管理システムの機能と利用目的

図 1 にコードレビュー管理システムの一つである Gerrit の一画面を示す。開発者がパッチを作成し、Gerrit にパッチを投稿した際に (1) パッチの基本情報とし

^{*1}Gerrit: <https://code.google.com/p/gerrit/>

^{*2}ReviewBoard: <https://www.reviewboard.org/>

て開発者名やプロジェクト名、投稿日時などが登録される。また (2) パッチの詳細情報としてそのパッチの URL も載せられており、検証者はそのパッチを取得できる。続いて、(3) 議論情報では、投稿されたパッチに対する検証者による指摘や、ソースコード内への直接的な指摘が管理されている。パッチ投稿からそのパッチの評価までコードレビュー管理システムで一元管理できるため、OSS プロジェクトや企業などで導入されている。

2.3 開発者の共同活動における課題

OSS プロジェクトにおけるコードレビューは、開発者にとっては、他の開発者と一緒にプログラムを開発する機会である。パッチ作成者は、検証者のフィードバックを通して、高い可読性のソースコードの書き方や開発技術について学ぶことができる。ただし、検証者からのフィードバックは、そのパッチ作成者にとって必ずしもパッチ修正に役立つとは限らない。たとえば、レビューアは不採択の投票をしているにも関わらず、コメントをしていない場合がある^{*3}。従って、レビューアからのコメントが開発者の貢献期間に影響を与えている可能性がある。Zhou と Mockus らの研究によると、検証者の数など開発者の周りの環境がパッチ作成者の貢献期間に影響を与えるということがわかっている [28]。また、Tao らの研究では、検証者からの指摘に関して、レビューレポートの分析、開発者へのアンケート、論文調査をしており、検証者のフィードバックには主な指摘理由 (Incomplete fix, Bad naming, Patch size too large など) があることを見つけた [26]。ただし、開発者が検証者から受ける指摘が開発者の貢献期間に与える影響については調査されていない。

本論文では、OSS プロジェクトにおいて、検証者からのフィードバックが開発者の貢献期間に影響を与えるのかどうかを確認するために、前調査として、開発者 20 名に対してアンケートを行う。具体的なアンケート内容を表 1 に示す。まず、開発者の OSS プロジェクトの活動履歴や初めて貢献したプロジェクトかなどの開発者の背景を確認する。その後、現在も Qt プロジェクトで開発による貢献をしているのかどうかを確認する。現在も貢献しているのであれば、本アン

^{*3}<https://codereview.qt-project.org/#/c/1644/>



Qt Open Governance
Code Review

(1) パッチの基本情報

Change-Id:	Teab68bb67e6421afbc4e07417c0196d419697f70	Commit Message	Permalink
Owner	hjk	QmlJSCodeStyleSettingsPage: compile fix	
Project	qt-creator/qt-creator	Was not compilable with namespaced Qt	
Branch	master	Change-Id: Teab68bb67e6421afbc4e07417c0196d419697f70	
Topic			
Uploaded	May 25, 2011 4:29 PM		
Updated	May 25, 2011 5:07 PM		
Status	Merged		
Reviewer		Code-Review	
Qt Sanity Bot		+1	
Christian Kamm			

Included in
Dependencies

(2) パッチの詳細情報

Reference Version:

Patch Set 1 227b45515b5b6b044081653a56867a852f56d4 (gitweb)

Patch Set 2 597d46cfc6cf96a749881a9c1a222445f5b6d03 (gitweb)

Author	hjk <qtco-commmitter@nokia.com> May 25, 2011 4:31 PM		
Committer	Christian Kamm <christian.d.kamm@nokia.com> May 25, 2011 5:06 PM		
Parent(s)	934a8238d4937f5bcf1dd22692e07d455af34506 Implement qml tab settings		
Download	checkout	pull	cherry-pick
	patch	Anonymous HTTP	
git fetch https://codereview.qt-project.org/qt-creator/qt-creator refs/changes/00/100/2 && git checkout FETCH_HEAD			

File Path	Comments	Size	Diff
Commit Message			Side-by-Side Unified
M src/plugins/qmljstools/qmljscodestylesettingspage.cpp		+1, -1	Side-by-Side Unified
M src/plugins/qmljstools/qmljscodestylesettingspage.h		+4, -3	Side-by-Side Unified
M src/plugins/qmljstools/qmljscodestylesettingspage.ui		+1, -1	Side-by-Side Unified
		+6, -5	All Side-by-Side All Unified

Comments	Expand Recent Expand All Collapse All
Qt Sanity Bot	Patch Set 1: Looks good to me, but someone else must approve May 25, 2011
Christian Kamm	Patch Set 1: Looks good to me, approved May 25, 2011
Gerrit Code Review	Your change could not be merged due to a path conflict. Please merge (or ... May 25, 2011
Christian Kamm	Uploaded patch set 2. May 25, 2011
Qt Sanity Bot	Patch Set 2: Looks good to me, but someone else must approve May 25, 2011
Christian Kamm	Patch Set 2: Looks good to me, approved May 25, 2011
Christian Kamm	

Change has been successfully cherry-picked as 0faf7ab9221b6df5a9314bc1d8022d44e8d84bc7.

[Add Comment](#)

(3) 議論情報

図 1 コードレビュー管理システムの一画面

ケートの目的である，検証者からのフィードバックが継続的な開発に影響を与えているのかについて調査し，継続的な開発に影響を与えるフィードバックや他の要因について確認する．現在，貢献をしていないのであれば，開発をやめるきっかけになった要因を尋ねる．最後に，開発者としての立場ではなく，検証者の立場として，開発者の貢献期間に影響を与えるフィードバック内容について尋ねる．

このアンケートを実施した結果，開発者2名(回答率10%)から回答を得ることができた．開発者2名とも，Qtプロジェクトが初めて貢献したプロジェクトではなく，貢献期間はそれぞれ3年以上4年未満，4年以上であったため両開発者とも長期貢献者であり，現在もパッチ投稿による貢献をしている．貢献する理由としては，「開発スキルの向上のため」，「ソフトウェア開発分野での名声を得たいから」であり，また両者とも「新たな機能を実装したいから」との回答をした．続いて，1人の開発者は検証者からのフィードバックが今後の貢献に影響を与えると回答したのに対して，もう一人の開発者は影響を与えないと回答した．まず，検証者からのフィードバックが影響を与えないと回答した開発者について述べたあと，検証者からのフィードバックが影響を与えると回答した開発者について述べる．検証者からのフィードバックが今後の貢献に影響を与えないと回答した開発者は，“I think qt is the best c++ frameworks I have ever seen, I like it, I want it be become more perfect”と述べており，より完璧なC++のフレームワークにすることが目的であり，検証者からのフィードバックは，今後の開発には影響を与えないという意見であった．ただし，検証者としての立場から，どのような指摘内容のフィードバックが開発者の貢献期間に影響を与えるのかを尋ねた結果，“Suboptimal solution”の指摘理由が影響を与えると回答した．次に，検証者からのフィードバックが今後の貢献に影響を与えると回答した開発者は，“constructive feedback aimed at improving my work”と述べており，建設的なフィードバックが今後の貢献に影響を与えるという意見であった．また，検証者としての立場から，どのような指摘内容のフィードバックが開発者の貢献期間に影響を与えるのかを尋ねた結果，“Incomplete fix”や“A feedback is not polite.”の指摘が影響を与えると回答した．アンケート結果として，開発者は検証者からのフィードバックが開発者の貢献期間に影響を与えていることがわかった．

表 1 開発者へのアンケート

質問番号	質問
Q1	Was Qt project the first project that you were engaged in OSS development?
Q2	How long have you worked in Qt project?
Q3	What is the motivation that you continuously contribute to submit a patch in Qt project?
Q4	Are you contributing to submit a patch on Qt project now?
Q5	Do you think that a feedback to my created patch from reviewers affect your motivation to continuously contribute to Qt project?
Q6	What feedback for your created patches will trigger to join continuously the project?
Q7	Please describe the other reason, if you have.
Q8	Please describe the other motivation to continuously contribute to Qt project.
Q9	Do you think that a feedback to my created patch from reviewers affect your motivation to continuously contribute to Qt project?
Q10	What feedback for your created patches will trigger to leave the project?
Q11	Why did you stop your working to submit a patch in the Qt project?
Q12	Why did you stop contributing on Qt project?
Q13	What kind of community do you hope to submit a patch continuously?
Q14	Have you worked a code review in Qt project?
Q15	As a reviewer, do you think developers who received reasons for pointed out tend to stop contributing?
Q16	If you were a reviewer ⁷ , do you think developers who received reasons for pointed out tend to stop contributing?

3. 分析手法

本章では、検証者からのどのような内容のフィードバックが開発者の貢献期間に影響を与えるのかを明らかにするための分析手法について述べる。はじめに、検証者から特定の指摘理由を受ける割合を知るために、その割合の計測方法について述べる。その後、検証者からのフィードバックが開発者の貢献期間に与える影響を調査するために、検証者からの指摘理由に基づいた開発者の貢献期間の予測方法について述べたあと、予測に重要な要因の特定方法について述べる。最後に、検証者からの指摘を自動で計測する言語モデルについて述べる。

3.1 開発者が受ける指摘理由の分布の計測方法

本論文では、検証者から特定の指摘理由を受ける割合を計測した。一般的に開発者が特定の指摘理由を受ける割合について述べた後、長期的に活動した開発者と短期的に活動した開発者が初期の頃に受けた指摘理由の割合について述べる。一般的に検証者から、どのような指摘理由があるのかを理解するために、検証者からのフィードバックを信頼係数95%でランダムにサンプルを抽出する。そして、抽出してきた検証者からのフィードバックを、Tao らが定義した指摘理由 [26] に基づき、手作業で分類する。検証者からのフィードバックが開発者の貢献期間に与える影響を調査するために、長期的貢献者と短期貢献者が、初めてパッチ投稿した日から3ヶ月間で受けた指摘理由を調査する。開発者の貢献期間は、はじめにパッチを投稿した日から最後にパッチを投稿した日までとする。なお、短期貢献者は、3ヶ月以上かつ1年未満、パッチ投稿によって貢献した開発者のことを指す。長期貢献者は、1年以上貢献した開発者を対象にパッチ投稿数の80パーセンタイルを満たす開発者を指す。3ヶ月以上とした理由は、Pinto らの研究 [18] で、開発者の貢献期間の予測には、3ヶ月以上の貢献履歴が必要という知見が得られているからである。1年以上とした理由は、Bird らの研究 [3] で、1年以内にプロジェクト内で役割が与えられないとプロジェクトを離脱することが多いという知見が得られているからである。長期貢献者を総パッチ投稿数の8割で絞った理由は、OSS の分野ではパレートの法則に従う例 [10] [19] [20] が多いからであり、こ

の処理により、パッチ投稿数の少ない長期貢献者を除外した。これらの基準に基づき、長期貢献者と短期貢献者が、初めてパッチ投稿をした日から3ヶ月間に受けた検証者からのフィードバックを Tao らが定義した指摘理由 [26] に手作業で分類する。

3.2 検証者からの指摘理由に基づいた開発者の貢献期間の予測方法

3.2.1 指摘理由を用いた手法

7種類の活動量と21種類のフィードバックの特徴量を用いて、開発者の貢献期間を予測する分類器を構築する。7種類の活動量には、パッチ投稿数、検証者数、プロジェクト数、受けたコメント数、不採択票数、採択票数、ソースコードへの直接的なコメント数が含まれる。これらの活動量は、先行研究でも一般的に用いられる、開発者の経験を理解するためのものである [13] [14] [24]。フィードバックの種類は、Tao らによって定義された指摘理由 [26] を使用した。本論文では、3ヶ月以上活動した開発者の中から、開発者の貢献期間を予測するモデルを構築する。3ヶ月以上とした理由は、Zhou らの研究 [28] で、大規模プロジェクトの場合は、3ヶ月以上の貢献履歴を要するだろうと主張されているからである。

3ヶ月以上貢献した開発者だけにフィルタリングした後、ランダムフォレスト [5] を用いる。モデルの精度を評価するために、適合率、再現率、F 値を用いる。すべてのメトリクスは [0~1] の範囲をとり、1 がもっとも高い精度であり、0 がもっとも低い精度であることを表す。モデル構築の前に、R パッケージの Redun 関数を用いて重回帰分析によって、相関の高いメトリクスを除去する。この手順は、予測に重要なメトリクスの解釈をやすくするために行う。本論文では、分類精度の分布を正確にするために100回、このプロセスを繰り返す。

3.2.2 パッチに対する内的要因と外的要因の指摘を用いた手法

検証者からのパッチに対する内的要因と外的要因の指摘が、開発者の貢献期間に与える影響を調査するために、指摘理由を分類した。その分類方法を表2に示す。Tao らの指摘理由に該当しないフィードバックに関しては、そのコメントごと

に内的要因と外的要因に分類し，判断できないものは“The Others”とした．表 2

表 2 内的要因と外的要因

	指摘理由
内的要因	Bad naming
	Compilation errors
	Including unnecessary changes
	Incomplete fix
	Inconsistent or misleading documentation
	Introducing new bugs
	Low cost-benefit
	Missing documentation
	Misuse of (deprecated) API or library
	Patch size too large
	Performance
	Security
	Solution too aggressive or hostile for end users
	Suboptimal solution
	Test failures
	Violating coding style guidelines
	Wrong direction
外的要因	Duplication
	Integration conflicts with existing code
	Not well isolated
	Freeze the development of other features

に基づき，検証者からのフィードバックを分類した．モデルの評価手法は，指摘理由を用いた手法と同様である．

3.3 開発者の貢献期間の予測に役立つ要因の特定方法

開発者の貢献期間の予測のために，重要であった要因を明らかにする [5]．重要度をスコア計算し，それらのスコアを Scott-Knott [21] によってクラスタリングする．Scott-Knott は，要因ごとに順位をつけたあと，中央値を用いて，2つの異なるランクに分類する．もし2つのランク間で統計的な差があれば，Scott-Knott

は、新たなランクを見つけて、再帰的に処理を繰り返す。もし2つのランク間で統計的な差がなければ、それらの要因を同じランクとする。ランクの高い要因は予測にとって重要な要因であることを示し、ランクの低いスコアを示す要因は予測にとって重要ではない要因であることを示す。Scott-Knott の目的は、予測に用いた要因を分類にとって重要な度合いにランク付けすることである。

3.4 検証者からのフィードバックに含まれるキーワードに基づいた開発者の貢献期間の予測方法

Tao らの指摘理由に基づきマニュアルで分類する方法は、実際のプロジェクトでの運用のためには実用的でないため、検証者によるフィードバックから自動的に特徴量を抽出する言語モデルを構築する。また、言語モデルを通して、指摘理由ではなく、検証者からのフィードバックに含まれるキーワードが開発者の貢献期間に与える影響について調査する。言語モデルの構築プロセスを以下に示す。

1. 開発者が検証者から受けたフィードバックを取得する。
2. フィードバックを形態素解析し、名詞・動詞・形容詞のキーワードを抽出し、ステミング処理を行う。
3. 抽出されたキーワードからストップワードを除去する。
4. 手順 1-3 により、開発者が受けた各キーワード数を取得する。
5. 開発者が受けた各キーワード数から、開発者の貢献期間を SVM(Support Vector Machine) [25] を用いて学習させる。

形態素解析は TreeTagger ツール^{*4}を用いて、名詞・形容詞・動詞に該当する [FW JJ JJR JJS NN N NNS NP NPS SYM VB VBD VBG VBN VBP VBZ] と識別されたキーワードを抽出しステミング処理をした。ストップワード辞書は NTLK

^{*4}<http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/>

(Natural Language ToolKit) ツール^{*5}を用いて、学習に使用した SVM は scikit-learn ツール^{*6}を用いた。モデルの評価手法は、指摘理由を用いた手法と同様である。

^{*5}<http://www.nltk.org>

^{*6}<http://scikit-learn.org/stable/>

4. ケーススタディ

4.1 データセット

本論文では、コードレビュープロセスにおける検証者からのフィードバックを理解するために、大規模な Qt プロジェクトと Openstack プロジェクトでケーススタディを実施した。両システムとも、コードレビューツールを通してレビュー履歴を膨大に記録しており、McIntosh らによって公開されている Gerrit のレビューデータベースを使用した [16]。

Qt とは、Cross-Platform アプリケーションの 1 つで、開発は digia corporation によってサポートされているが、Qt はコミュニティ全体からの寄付を歓迎している。そのデータベースには、70,705 件のレビューレポートや 624,318 件の検証者からのコメントが含まれている。表 3 に Qt データセットの統計的な値を示す。検証者による指摘は、パッチに対する不採択コメントである。また、プロジェクトへのパッチ開発による貢献期間と開発者数を図 2 に示す。少なくとも 1 度でもパッチ投稿した開発者は 930 名であり、長期貢献者は 77 名 (8.3%)、短期貢献者は 227 名 (24.4%) であった。

表 3 Qt システムの概要

期間	2011 年 3 月 - 2013 年 11 月
レビュー票数	142,798
開発者数	930
検証者数	1,076
検証者による指摘数	18,659
検証者によるコメント数	624,318

Openstack とは、オープンソースで開発されているクラウド環境構築用のソフトウェア群であり、非営利団体と理事会、技術委員会、ユーザー委員会によってサポートされている。そのデータベースには、92,984 件のレビューレポートや 1,662,892 件の検証者からのコメントが含まれている。表 4 に Openstack データセットの統計的な値を示す。検証者による指摘は、パッチに対する不採択コメン

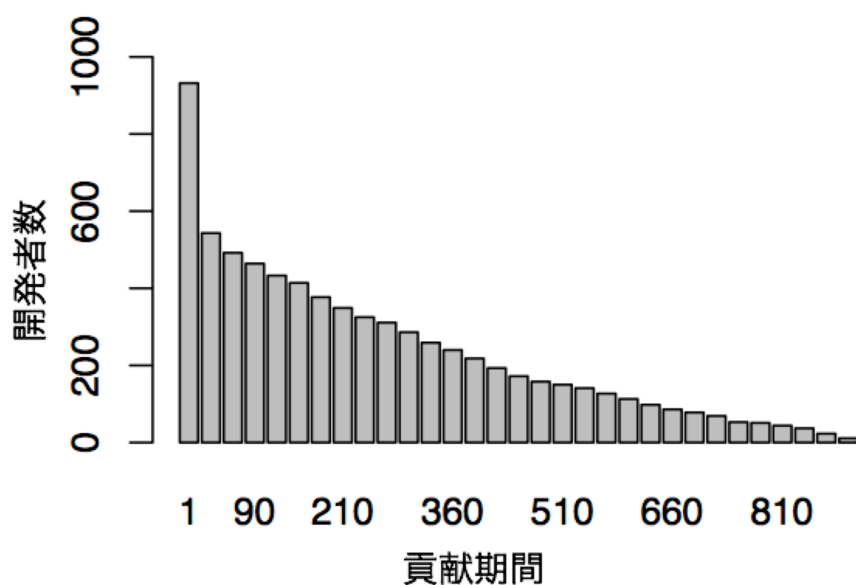


図 2 Qt プロジェクトにおける貢献期間と開発者数

トである。また、プロジェクトへのパッチ開発による貢献期間と開発者数を図 3 に示す。少なくとも 1 度でもパッチ投稿した開発者は 2,637 名であり、長期貢献者は 139 名 (5.2%)、短期貢献者は 776 名 (29.4%) であった。

表 4 Openstack システムの概要

期間	2011 年 8 月 - 2014 年 5 月
レビュー票数	92,984
開発者数	2,636
検証者数	2,925
検証者による指摘数	165,621
検証者によるコメント数	1,662,892

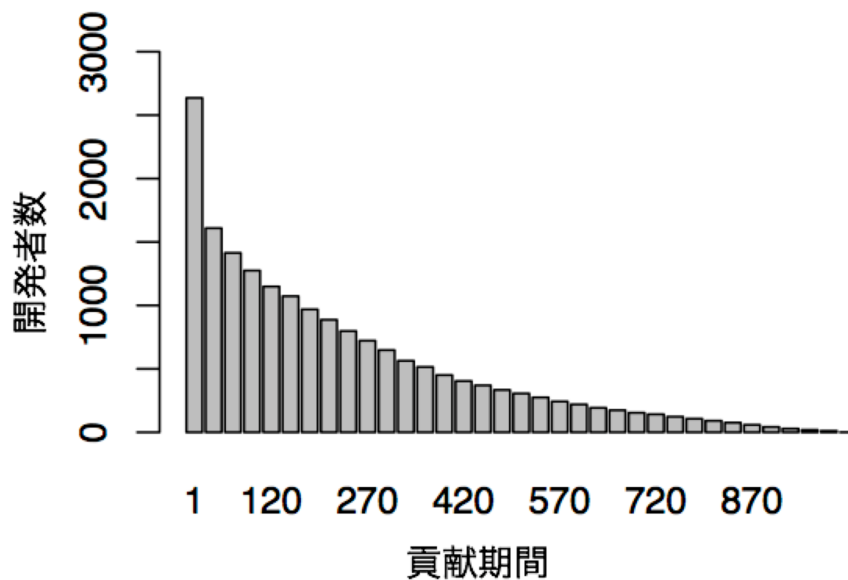


図 3 Openstack プロジェクトにおける貢献期間と開発者数

4.2 結果

4.2.1 長期貢献者と短期貢献者の活動量と受けた指摘の違い

一般的に検証者からどのような指摘があるのかを理解するため、Qt プロジェクトにおける検証者からの指摘理由の分布を調査した。全体の比率を推定するために、信頼係数 95% で、サンプリングして、376 件を抽出し、それらをマニュアルで Tao らの指摘理由 [26] に基づき分類した。その結果を表 5 に示す。38% の指摘が特定の理由ではなく 62% が特定の理由であり、“Incomplete fix”, “Including unnecessary changes”, “Suboptimal solution” という指摘理由が多い結果であった。それぞれの実際の指摘内容を例として、以下に示す。

- **Incomplete fix** “Does not work ;-(”
- **Including unnecessary changes** “Unnecessary change.”
- **Suboptimal solution** “Still some polishing needed (see my two comments)...”

表 5 指摘理由とサンプルの指摘，長期貢献者・短期貢献者が受けた指摘

指摘理由	サンプルの指摘		長期貢献者		短期貢献者	
	数	割合	数	割合	数	割合
Bad naming	9	2.4 %	12	1.2 %	17	1.3 %
Compilation errors	18	4.8 %	18	1.8 %	20	1.5 %
Duplication	17	4.5 %	17	1.7 %	28	2.6 %
Freeze the development of other features	0	0.0 %	28	1.4 %	16	1.2 %
Including unnecessary changes	21	5.6 %	31	3.1 %	26	2.0 %
Incomplete fix	61	16.2 %	130	13.2 %	187	14.4 %
Inconsistent misleading documentation	9	2.4 %	19	1.9 %	32	2.5 %
Integration conflicts with existing code	8	2.1 %	2	0.2 %	6	0.5 %
Introducing new bugs	3	0.8 %	14	1.4 %	8	0.6 %
Low cost-benefit	3	0.8 %	1	0.1 %	0	0.0 %
Missing Documentation	12	3.2 %	19	1.9 %	33	2.5 %
Misuse of (deprecated) API or library	1	0.3 %	0	0.0 %	0	0.0 %
Not_well_isolated	8	2.1 %	2	0.2 %	4	0.3 %
Patch size too large	3	0.8 %	6	0.6 %	7	0.5 %
Performance	1	0.3 %	2	0.2 %	0	0.0 %
Security	1	0.3 %	1	0.1 %	1	0.1 %
Solution too aggressive or hostile for end users	6	1.6 %	3	0.3 %	2	0.2 %
Suboptimal solution	19	5.1 %	64	6.5 %	52	4.0 %
Test Failures	8	2.1 %	8	0.8 %	17	1.3 %
Violating coding style guidelines	18	4.8 %	9	0.9 %	25	1.9 %
Wrong direction	7	1.9 %	49	5.0 %	36	2.8 %
The Others	143	38.0 %	578	58.7 %	784	60.3 %
All	376	100.0 %	985	100.0 %	1,301	100.0 %

これらの一般的な指摘理由の割合を元に、長期貢献者と短期貢献者が受けた指摘の分布の調査を行なった。Qt データベース内では、長期貢献者は 77 名 (8.3%), 短期貢献者は 227 名 (24.4%) が該当した。表 6 は、長期貢献者と短期貢献者の初期の 1ヶ月、2ヶ月、3ヶ月間の活動量の中央値を示している。

表 6 長期貢献者と短期貢献者の活動量の中央値 [Qt]

長期貢献者				
	パッチ投稿数	プロジェクト数	検証者数	指摘数
初期の 1ヶ月間	13	1	5	2
初期の 2ヶ月間	29	2	9	4
初期の 3ヶ月間	49	2	11	7

短期貢献者				
	パッチ投稿数	プロジェクト数	検証者数	指摘数
初期の 1ヶ月間	4	1	3	1
初期の 2ヶ月間	5	1	4	2
初期の 3ヶ月間	6	1	5	2

表 6 より、初期の 3ヶ月間において検証者から受け取った指摘数では、長期貢献者は 7 件、短期貢献者は 2 件であった。そのため、長期貢献者の方が、短期貢献者よりも多くの検証者からの指摘を受け取っている。また、長期貢献者の方が、短期貢献者よりも多くのパッチ投稿を開発しており、多くのプロジェクトに携わり、多くの検証者にフィードバックを得ていることがわかった。

次に、長期貢献者と短期貢献者が受けた指摘理由について理解するために、検証者から長期貢献者と短期貢献者への初期 3ヶ月間でのフィードバック 2,286 件をマニュアルで読んだ。表 5 にその結果を示す。短期貢献者の方が長期貢献者よりも “The Others” の指摘をより頻繁に受けるということがわかった。言い換えると、長期貢献者の方が、短期貢献者よりも多くの特定の指摘理由を受けている。特に、長期貢献者は、“Wrong Direction” と “Suboptimal solution” の指摘理由を短期貢献者よりも受けていることがわかった。

表 7 に、Openstack プロジェクトにおける長期貢献者と短期貢献者の初期の 1ヶ月

月、2ヶ月、3ヶ月間の活動量の中央値を示す。

表 7 長期貢献者と短期貢献者の活動量の中央値 [Openstack]

長期貢献者

	パッチ投稿数	プロジェクト数	検証者数	指摘数
初期の1ヶ月間	5	1	8	5
初期の2ヶ月間	10	2	12	10
初期の3ヶ月間	16	3	17	18

短期貢献者

	パッチ投稿数	プロジェクト数	検証者数	指摘数
初期の1ヶ月間	3	1	8	6
初期の2ヶ月間	5	2	10	10
初期の3ヶ月間	7	2	13	14

表 7 より、初期の3ヶ月間において検証者から受け取った指摘数では、長期貢献者は18件、短期貢献者は14件であった。そのため、Openstack プロジェクトにおいても、長期貢献者の方が、短期貢献者よりも多くの検証者からの指摘を受けている。また、初期の1ヶ月間では活動量に大きな差は見られないが、初期の3ヶ月間では、長期貢献者の方が、短期貢献者よりも多くのパッチ投稿を開発しており、多くの検証者からフィードバックを得ていることがわかる。

結論

検証者からの62%の指摘は、“Incomplete fix” や “Including unnecessary changes”, “Suboptimal solution” などの特定の指摘理由であった。Qt プロジェクト、Openstack プロジェクトともに初期の3ヶ月間では、長期貢献者の方が短期貢献者よりも多くのパッチを投稿し、多くのプロジェクトに参加し、多くの検証者からフィードバックを得ており、多くの指摘を受けていることがわかった。Qt プロジェクトにおいては、長期貢献者の方が短期貢献者よりも “Suboptimal solution” や “Wrong direction” の指摘理由を多く受けていることがわかった。

4.2.2 検証者からの指摘理由に基づいた開発者の貢献期間の予測精度と分析結果

Qt プロジェクトを対象に，長期貢献者と短期貢献者の初期の1ヶ月間，2ヶ月間，3ヶ月間での開発者が受けた指摘理由をもとに開発者の貢献期間を予測した結果を表8に示し，内的要因と外的要因の指摘理由をもとに開発者の貢献期間を予測した結果を表9に示す．なお，多重相関性により，パッチ投稿数とコメント数，採択票数の相関が高く，不採択票とソースコードへの直接的なコメント数，“The Others”の相関が高かったため，モデル構築のときに，コメント数，採択票数，ソースコードへの直接的なコメント数，“The Others”は除去した．

表8 指摘理由に基づいた開発者の貢献期間の予測結果

	適合率	再現率	F 値
初期の1ヶ月間	55 %	30 %	38 %
初期の2ヶ月間	55 %	35 %	44 %
初期の3ヶ月間	62 %	41 %	49 %

表9 内的要因と外的要因の指摘理由に基づいた開発者の貢献期間の予測結果

	適合率	再現率	F 値
初期の1ヶ月間	55 %	34 %	42 %
初期の2ヶ月間	57 %	40 %	46 %
初期の3ヶ月間	61 %	45 %	51 %

表8，表9から，適合率，再現率，F 値ともに初期の1ヶ月の情報量から初期の3ヶ月の情報量に増やすとともに，高い予測精度を示すことがわかった．また，指摘理由ではなく，パッチに対する内的要因と外的要因に分類したあとに，開発者の貢献期間を予測した方が精度が高くなることがわかった．

検証者からのフィードバックが開発者の貢献期間の予測に与える影響を調査するために，検証者からのフィードバックを使用した場合と使用しない場合の分類精度を比較した結果を表10に示す．表10から検証者からのフィードバックが，開発者の貢献期間の予測精度の中でも適合率に大きな影響を与えるということがわかった．

表 10 初期の3ヶ月間の情報をもとに、開発者の貢献期間の予測結果 (検証者からのフィードバックの有無による比較)[Qt]

	適合率	再現率	F 値
フィードバック無し	56 %	45 %	49 %
内的要因と外的要因	61 %	45 %	51 %
指摘理由	62 %	41 %	49 %

次に、表 11 に、開発者が3ヶ月間に、検証者から受けた指摘理由に基づいた開発者の貢献期間の予測に重要な要因を示し、表 12 に、内的要因と外的要因の指摘理由に基づいた開発者の貢献期間の予測に重要な要因を示す。

表 11, 表 12 より、検証者からのソースコードに対する内的要因の指摘が、開発者の貢献期間に影響を与えることがわかり、その中でも “Incomplete fix” が指摘理由の中では大きな影響を与えることがわかった。また内的要因の指摘、パッチ投稿数, 検証者数, 不採択票数, プロジェクト数, “Incomplete fix” が予測にとって重要な要因であることがわかった。長期貢献者と短期貢献者でのそれぞれ6つの要因についての分布を図 4, 図 5, 図 6, 図 7, 図 8, 図 9 に示す。黒が長期貢献者であり、灰色が短期貢献者である。結果として、中央値で見ると、すべての要因において、長期貢献者の方がより多くの検証者から指摘を受けており、より多くのパッチ開発をしていることがわかった。また、6つの要因の分布すべてにおいて統計的有意差が1%以下で確認された。そのため、短期貢献者と比較すると長期貢献者の方が、初期のパッチ投稿から3ヶ月間で、より多くの “Incomplete fix” の指摘理由を受けており、より多くのパッチを投稿しており、多くの検証者から評価をもらっていることがわかった。

表 11 Scott-Knott を用いた重要な要素の特定 [指摘理由]

Group	初期の 3ヶ月間	
	要因	割合
4	パッチ投稿数	5.252 %
3	検証者数	2.073 %
2	不採択票数	0.404 %
	プロジェクト数	0.374 %
	Incomplete fix	0.292 %
1	Suboptimalsolution	0.154 %
	Wrong direction	0.068 %
	Including unnecessary changes	0.060 %
	Test failures	0.052 %
	Missing documentation	0.040 %
	Inconsistent or misleading documentation	0.040 %
	Integration conflicts with existin gcode	0.034 %
	Duplication	0.026 %
	Introducing new bugs	0.014 %
	Not well isolated	0.009 %
	Patch size too large	0.009 %
	Performance	0.000 %
	No accompanied test cases	0.000 %
	Misuse of deprecated API or library	0.000 %
	Changes to internal APIs	0.000 %
	Low cost-benefit	0.000 %
	Security	-0.003 %
	Solution too aggressive or hostile for end users	-0.004 %
	Bad naming	-0.007 %
	Security	-0.007 %
	SViolating coding style guidelines	-0.018 %
	Freeze the development of other features	-0.021 %

表 12 Scott-Knott を用いた重要な要素の特定 [内的要因と外的要因の指摘理由]

Group	初期の3ヶ月間	
	要因	割合
4	パッチ投稿数	6.236 %
3	検証者数	2.268 %
2	内的要因の指摘数	0.573 %
	“The Others”	0.488 %
1	プロジェクト数	0.308 %
	外的要因の指摘数	0.107 %

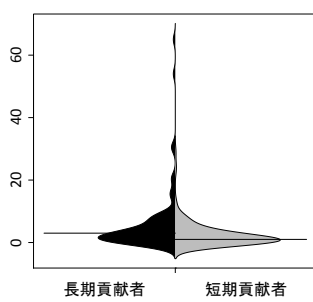


図 4 内的要因の指摘数

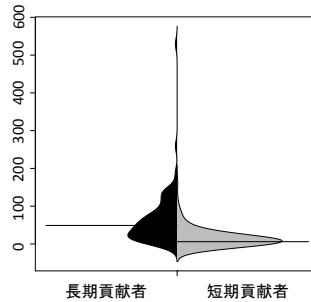


図 5 パッチ投稿数

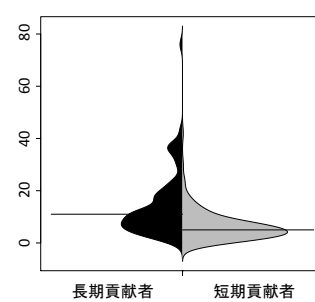


図 6 検証者数

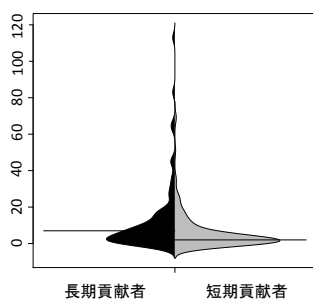


図 7 不採択票数

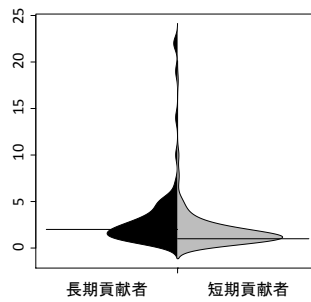


図 8 プロジェクト数

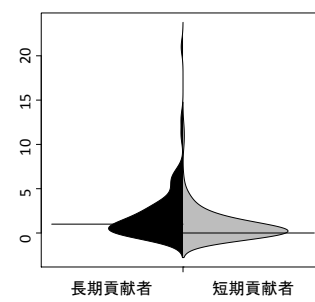


図 9 “Incomplete fix”

結論

検証者からのフィードバックの内的要因・外的要因，指摘理由に基づいた開発者の貢献期間の予測は，初期の1ヶ月間～3ヶ月間に期間を延ばすにつれて予測精度が向上した．検証者からのフィードバックを用いない場合よりも，検証者からのフィードバックをパッチに対する内的要因・外的要因に分類した予測結果の方が高い予測精度を示した．検証者からのフィードバックに関する要因で予測にとって重要であったのは内的要因であり，内的要因の中でも“Incomplete fix”の指摘理由を受けた数が予測にとって重要な要因であった．

4.2.3 検証者からのフィードバックに含まれるキーワードに基づいた開発者の貢献期間の予測精度と分析結果

Qt プロジェクトと Openstack プロジェクトで，長期貢献者と短期貢献者の初期の1ヶ月間，2ヶ月間，3ヶ月間での情報をもとに開発者の貢献期間を予測した結果を表 13，表 14 に示す．Qt プロジェクトと Openstack とともに，初期の2ヶ月

表 13 キーワードに基づいた開発者の貢献期間の予測結果 [Qt]

	適合率	再現率	F 値
初期の1ヶ月間	45 %	26 %	32 %
初期の2ヶ月間	51 %	43 %	48 %
初期の3ヶ月間	52 %	44 %	48 %

表 14 キーワードに基づいた開発者の貢献期間の予測結果 [Openstack]

	適合率	再現率	F 値
初期の1ヶ月間	47 %	40 %	42 %
初期の2ヶ月間	48 %	46 %	46 %
初期の3ヶ月間	51 %	47 %	48 %

と初期の3ヶ月ではほとんど精度に違いは見られないが，初期の1ヶ月と比較すると高い結果となった．Qt プロジェクトの結果を表 10 のフィードバック無しと比

較すると、検証者からのフィードバックを使用しない場合とほとんど差は見られなかった。そのため、Qt プロジェクトにおいては、検証者からの指摘理由とソースコードに対する内的要因の指摘は開発者の貢献期間に影響を与えるが、検証者からのフィードバックに含まれるキーワードは開発者の貢献期間に大きな影響は与えないことがわかった。Openstack プロジェクトの結果を検証者からのフィードバック無しと比較した結果を表 15 に示す。Openstack プロジェクトにおいて

表 15 初期の3ヶ月間の情報をもとに、開発者の貢献期間の予測結果 (検証者からのフィードバックの有無による比較)[Openstack]

	適合率	再現率	F 値
フィードバック無し	36 %	15 %	22 %
フィードバック有り	51 %	47 %	48 %

は、検証者からのフィードバックを含むことにより、分類精度が向上した。フィードバック無しでの分類精度が低い理由は、表 7 に示したように、Openstack プロジェクトでは短期貢献者と長期貢献者の活動量の差が小さいからだと考えられる。

開発者の貢献期間の予測に重要であった要因（キーワード）を調査するために、Qt プロジェクトと Openstack プロジェクトのキーワードを分析する。長期貢献者 X 名のうちキーワード i を受けた開発者を x_i 名とし、短期貢献者 Y 名のうちキーワード i を受けた開発者を y_i 名としたとき、キーワード i を受けた長期貢献者の割合を $\frac{x_i}{X}$ 、キーワード i を受けた短期貢献者の割合を $\frac{y_i}{Y}$ とする。Qt プロジェクトと Openstack プロジェクトでこれらの割合が高かったキーワードを表 16、表 17 に示す。Qt プロジェクトでは“good”，“see”，“remove”を約 20～25%ほど長期貢献

表 16 開発者がよく受けたキーワード [Qt]

キーワード	長期貢献者	短期貢献者
good	42.9 %	23.3 %
see	40.3 %	20.7 %
remove	39.0 %	13.6 %

者の方が多く受けており、Openstack プロジェクトでは“good”，“get”，“thing”

表 17 開発者がよく受けたキーワード [Openstack]

キーワード	長期貢献者	短期貢献者
good	67.6 %	49.2 %
get	52.5 %	33.9 %
thing	43.2 %	27.2 %

を約 16～19%ほど長期貢献者の方が多く受けている結果となった。また、Qt プロジェクト Openstack プロジェクトともに、長期貢献者数に対する‘good’を含むフィードバックを受けた開発者の割合と、短期貢献者数に対する‘good’を含むフィードバックを受けた開発者の割合を用いて、比率の差検定を行なった結果、統計的有意差が 1%以下で確認された。このことから、初期の段階で‘good’を含むフィードバックを多く受けた開発者の方が長期間、貢献することがわかった。

実際にこれらの単語が検証者からのフィードバック内で使用されている例を表 18 に示す。

表 18 より、Qt プロジェクトでは、“remove”はある関数の除去を依頼する指摘でよく使われており、“see”は自身もしくは他の開発者が書いたフィードバックの閲覧を依頼する指摘でよく使われ、“good”は問題箇所以外の部分に関しては良いというフィードバックで使われている。Openstack プロジェクトでは、“get”は検証者が実際にソースコードをダウンロードし、ビルドした結果について述べるもしくは、インラインコメントを書いたことを伝えるためによく使われており、“good”は、Qt プロジェクトと同様に問題箇所以外の部分に関しては良いという

*⁷<https://codereview.qt-project.org/#/c/832/>

*⁸<https://codereview.qt-project.org/#/c/10457/>

*⁹<https://codereview.qt-project.org/#/c/103/>

*¹⁰<https://codereview.qt-project.org/#/c/11822/>

*¹¹<https://codereview.qt-project.org/#/c/421/>

*¹²<https://codereview.qt-project.org/#/c/935/>

*¹³<https://review.openstack.org/#/c/7608/>

*¹⁴<https://review.openstack.org/#/c/12874/>

*¹⁵<https://review.openstack.org/#/c/1436/>

*¹⁶<https://review.openstack.org/#/c/3144/>

*¹⁷<https://review.openstack.org/#/c/190/>

*¹⁸<https://review.openstack.org/#/c/23817/>

表 18 キーワードが使用された例文

プロジェクト	キーワード	例文
Qt	remove	Lines 479-482 in qpaintengine_raster.cpp, where is-Plain45DegreeRotation is set to true, need to be removed as well. ^{*7}
		Lets remove the configure_file() for the .qrc files. ^{*8}
	see	Good change, but see comments. ^{*9}
		See Friedemanns comments ^{*10}
	good	Minor mistakes, otherwise looks good. ^{*11}
		Please update the wayland sha1 file as well :) except for that its all good! ^{*12}
Openstack	get	Changes look good overall, but Im getting an XML syntax error when trying to build this. Also, some inline comments about removing some other euca- refs. ^{*13}
		What a simple fix, and thanks for figuring out the testing. Ive got a few inline comments for you. ^{*14}
	good	Looks good but has a nit. ^{*15}
		looks good overally; I have some comments I would like you to address. ^{*16}
	thing	This looks great, Jay. Found a few things: ^{*17}
		There is a lot of unneeded reformatting here. It should be easy enough to undo that. Otherwise things look great! ^{*18}

フィードバックでよく使われ, “thing” は, 少数の問題があったことを述べるもしくは, “good” のように問題箇所以外の部分に関しては良いというフィードバックで使われていた. Qt プロジェクトと Openstack プロジェクトにおいて “good” を含む検証者からのフィードバックは 527 件あり, 信頼係数 95% でランダムにサンプリングし, 222 件を調査した結果, 検証者からの 74% のフィードバックで問題箇所以外の部分に関しては良いという指摘で使われていた.

結論

検証者からのフィードバックに含まれるキーワードに基づいた開発者の貢献期間の予測は, 初期の 1ヶ月間~3ヶ月間に期間を延ばすにつれて予測精度が向上した. 検証者からのフィードバックに含まれるキーワードを用いることにより, Qt プロジェクトでは予測精度に変化はなかったが, Openstack プロジェクトでは予測精度が向上した. 両プロジェクトともに, 検証者からのフィードバックに含まれるキーワードの中で予測にとって重要であったのは “good” であり, 長期貢献者の方が短期貢献者よりも “good” を含むフィードバックを多く受けていることが統計的有意差 1% 以下で確認された.

5. 考察

5.1 議論

前章の結果より，Qt プロジェクト・Openstack プロジェクトともに，長期貢献者の方が短期貢献者よりも初期の3ヶ月間で，多くの指摘を受けていることがわかった．特に Qt プロジェクトにおいては，長期貢献者はソースコードに対する内的要因の指摘を多く受けており，内的要因の中でも“Imcomplete fix”の指摘理由を多く受けている傾向があることがわかった．開発者の貢献期間の予測結果では，Qt プロジェクト・Openstack プロジェクトともに，初期の1ヶ月の情報量から初期の3ヶ月の情報量に増やすとともに予測精度が高くなった．初期の3ヶ月への期間を延ばすにつれて予測精度が向上する結果は，Zhou らの研究結果 [30] でも同様の知見が得られていたため，他の OSS プロジェクトでも同様の結果が得られると考えられる．検証者からのフィードバックを利用すると，予測精度は減少することではなく，プロジェクトによっては向上することがわかった．それは，Qt プロジェクトと Openstack プロジェクトを比較した結果，長期貢献者と短期貢献者の活動量の差が小さい場合，検証者からのフィードバックを利用することで，予測精度が向上すると考えられる．検証者からのフィードバックに含まれるキーワードを分析した結果，Qt プロジェクト・Openstack プロジェクトともに，短期貢献者と比べ，長期貢献者の方が“good”を含む検証者からのフィードバックを受けている割合が表 ??より約 18～20%ほど多いことがわかった．どちらのプロジェクトでも“good”は“Minor mistakes. otherwise looks good.”のように，問題箇所以外の部分に関しては良いという文脈で，検証者からの約 74%のフィードバックで使われており，このようなフィードバックを初期の段階で多く受けている開発者が長期的に貢献することがわかった．事前アンケートの結果から，検証者からのフィードバックが開発者の貢献期間に影響を与えることがわかったが，長期的に貢献する開発者の特徴として，内的要因の指摘や不完全な修正の指摘，“good”を含む指摘を受けやすいかもしれない．その因果関係について，今後，調査していく必要がある．

また，開発者の活動量の観点から見ると，パッチ投稿数や検証者数の多い開発

者の方が長期貢献者になりやすい傾向であることがわかった。パッチ投稿数に関しては Ihara らの研究結果 [11], 検証者数に関しては Zhou と Mockus の研究結果 [30] から同様の知見が得られている。したがって、活動量の観点に関しては、Qt プロジェクトや Openstack プロジェクト以外の OSS プロジェクトでも同様の結果が得られると考えられる。ただし、検証者からの指摘理由と開発者の貢献期間に関しては、Qt は膨大な開発履歴を持つプロジェクトであるが、他のプロジェクトでも同様の結果が得られるかどうかはわからないため、他の OSS プロジェクトについても調査する必要があると考えられる。

5.2 制約

内的妥当性: 本論文では、検証者からのフィードバックに着目し、それが開発者の貢献期間に与える影響を調査した。ただし、検証者は開発者が投稿したパッチに基づいてフィードバックを投稿するため、開発されたパッチについても考慮する必要があるが、本論文では、開発されたパッチについての分析はしていない。

また、アンケート結果として、検証者からのフィードバックが自身の貢献期間に影響を与えないと回答した開発者がいるように、今回対象とした開発者全員が、検証者からのフィードバックが彼らの貢献期間に影響を与えているとは限らない。プロジェクトと関係のないこと（私生活や仕事など）が影響して、プロジェクトから離脱した開発者も含まれていると考えられる。そのため、実際には検証者によるパッチに対する内的要因の指摘、“Incomplete fix”の指摘理由が開発者の貢献期間に影響を与えていないことが一つの可能性として考えられる。

外的妥当性: 本論文では、Qt システムだけを用いて検証者からのフィードバックの内的要因・外的要因、指摘理由が開発者の貢献期間に与える影響を調査した。その結果、膨大な開発履歴を有する Qt プロジェクトを通して、パッチに対する内的要因の指摘、“Incomplete fix”の指摘理由が開発者の貢献期間の予測に重要な要因であることを示したが、他のプロジェクトでも同様の結果が得られるかは明らかではない。

6. 関連研究

6.1 開発者が貢献する動機

昨今、開発者が OSS プロジェクトに参加する動機を理解するための論文が数多く発表されている。Bonaccorsi らの研究 [4] によると、OSS プロジェクトに開発者が貢献する動機には主に 3 つあるとされている。経済的な動機と技術的な動機、社会的な動機である。経済的な動機の観点から、Wu らの研究 [27] では、開発者が貢献する動機について調査がなされた。この調査では、開発者 148 名へのアンケート結果から、開発を通しての満足感や開発者間の助け合いが、開発者の長期的な貢献につながると結論づけられている。また、技術的な動機の観点から、Jiang らの研究 [13] では、開発者の活動量と貢献期間について調査されており、37 プロジェクトに対して、開発者の貢献期間の予測で F 値 0.818 と高い精度を示した。Hannebauer らの研究 [8] [9] では、OSS に携わる開発者がプロジェクトで貢献するにあたっての障壁についてのアンケート調査がなされた。この調査の結果として、障壁の大部分としてアプリケーションをビルドできないことが挙げられており、またビルドし開発できたとしても CI システムによるテストが障壁であると結論づけられている。続いて、社会的な動機の観点からは、開発者の貢献期間と、プロジェクトに携わってから初期の段階での検証者の数や検証者からのフィードバックの有無、開発者間のつながりに関する研究がなされている [7] [22] [28]。ただし、いずれも本研究の目的である検証者からのフィードバックが開発者の貢献期間に与える影響の分析とは観点が異なる。

6.2 検証者からのフィードバック

近年、開発者が投稿したパッチに対する、検証者からのフィードバックを対象とした研究が数多く発表されている。Tao らの研究 [26] によると、指摘理由がパッチの採択率に与える影響について調査しており、“Suboptimal solution”, “Duplication”, “Incomplete fix” と指摘されたパッチが不採択にされやすいと結論づけられている。Beller らの研究 [2] では、検証者からのフィードバックによっ

て、パッチがどう変化していくのかについて調査された。この調査の結果、検証者からの10-35%のコメントによってパッチは変更されず、残りの65-90%のコメントのうち75%が開発者に構造レベルの大きな変更を要し、25%が開発者に機能単位の変更を要すると示されている。また、Nurolahzadeらの研究[17]では、検証者が開発者を支援するピアレビューについて調査された。結果として、17.3%のレビューのみがピアレビューであり、モジュールオーナーによるレビューと比較すると、ピアレビューでは非機能的な欠陥(例:ソフトウェアの使いやすさなど)についての指摘が多いことが示された。そのため、モジュールオーナーがパッチをレビューする頻度が高く、非機能的な欠陥は実装後に実施されるテストでは発見は困難であるため、非機能的な欠陥はモジュールオーナーによるレビューを通して発見されるべきであると結論づけられている。ただし、いずれも本研究の目的である検証者からのフィードバックが開発者の貢献期間に与える影響の分析とは観点が異なる。

7. おわりに

従来研究から開発者間の繋がりが貢献期間に影響を与えることは知られているが、検証者からのフィードバックが開発者の貢献期間に与える影響は調査されてこなかった。本論文では、プロジェクト側がどの開発者が長期的に貢献するのかを調査する際の指針の一つとして、開発者の活動量と検証者からのフィードバックに注目し開発者の貢献期間を分析した。その結果、長期的に貢献する開発者の方が、初期の3ヶ月間で、多くのパッチを開発し、多くのプロジェクトに参加していることがわかり、検証者から多くの指摘を受けていることがわかった。指摘の中でも、長期貢献者の方がパッチに対する内的要因の指摘を多く受けており、内的要因の中でも“Incomplete fix”の指摘を多く受けていることがわかった。指摘に含まれるキーワードとしては、長期貢献者の方が“good”を含む指摘を受けたことのある開発者の割合が多いことがわかった。

内的要因・外的要因に分類した検証者からのフィードバックに基づいた開発者の貢献期間の予測結果としては、適合率 61%，再現率 45%，F 値 51%であり、指摘理由に分類した検証者からのフィードバックに基づいた開発者の貢献期間の予測結果としては、適合率 62%，再現率 41%，F 値 49%であった。キーワードに基づいた開発者の貢献期間の予測結果としては、適合率 51～52%，再現率 44～47%，F 値 47～48%であった。いずれの場合も開発者の活動履歴を初期の1ヶ月間～3ヶ月間に期間を延ばすにつれて予測精度が向上し、検証者からのフィードバックを利用することで予測精度を保つもしくは向上した。

本論文では、プロジェクトにより検証者からのフィードバックが開発者の貢献期間に与える影響の大きさは異なるが、検証者からのフィードバックは開発者の貢献期間に影響を与えると結論付ける。今後は、検証者からのフィードバックは開発者が投稿したパッチに基づいて行われるためパッチと検証者からのフィードバックの分析を深めることで、より長期的に貢献する開発者の特徴を理解できると考えられる。

謝辞

本研究を進めるにあたり、多くの方々にご指導、ご協力、ご支援を賜りました。ここに謝意を添えてお名前を記させていただきます。主指導教員であり、本論文の審査委員を務めていただいた奈良先端科学技術大学院大学 情報科学研究科 松本健一教授に対し、厚く御礼申し上げます。研究に対する直接的な指導に限らず、研究に取り組む上での心構えから研究者としての在り方まで、ありとあらゆる面で熱心なご指導を賜りました。

副指導教官であり、本論文の審査委員を務めていただいた奈良先端科学技術大学院大学 情報科学研究科 飯田元教授には、学内発表において研究の質を高める上で非常に貴重なご意見を頂きました。心より感謝申し上げます。

カーネギーメロン大学 言語技術研究所 Graham Neubig 助教には、研究を進める上での相談に乗っていただきました。Graham Neubig 助教の支えなしには乗り切ることができなかったと思っております。心より感謝申し上げます。

奈良先端科学技術大学院大学 情報科学研究科 伊原彰紀助教には、研究の進め方から、論文執筆、プレゼンテーション作法まで何度も明敏なご指導を頂きました。伊原彰紀助教から頂いた数々の指導は私の将来の財産になると確信しております。また、修士1年の途中で研究室を移籍したにも関わらず、熱心に研究を見てくださいったことには感謝しきれません。心より感謝申し上げます。

奈良先端科学技術大学院大学 情報科学研究科 畑秀明助教 には、研究の進め方やプレゼンテーションについてご指導を頂きました。また、研究に行き詰った時はいつも優しいお言葉をかけて頂きました。心より感謝申し上げます。

奈良先端科学技術大学院大学 情報科学研究科 ソフトウェア工学研究室の皆様(一ノ瀬 智浩, 小野 健一, 上村 康輔, 則行 祐作, 平尾 俊貴, 藤林 大毅, 南 智孝, 弥谷 圭朗, Jirayus Jiarapakdee, Stevche Radevski) および環境知能学研究室の皆様(魚谷 果那, 大石 陽波, 岡島 知也, 光崎 将人, 中田 彩, 西川 直希, 萩原 康平) とは、日頃より苦楽を共にさせて頂きました。お陰様で非常に楽しく充実した二年間を送ることができました。ありがとう。最後に、大学院進学、就職活動と私の人生における転機において、暖かく私の決断を見守り、支援してくれた父、母、兄、祖父と祖母に深く感謝致します。

参考文献

- [1] Bacchelli, A. and Bird, C.: Expectations, Outcomes, and Challenges of Modern Code Review, *Proceedings of the 35th International Conference on Software Engineering*, pp. 712–721 (2013).
- [2] Beller, M., Bacchelli, A., Zaidman, A. and Juergens, E.: Modern Code Reviews in Open-source Projects: Which Problems Do They Fix?, *Proceedings of the 11th Working Conference on Mining Software Repositories*, pp. 202–211 (2014).
- [3] Bird, C., Gourley, A., Devanbu, P., Swaminathan, A. and Hsu, G.: Open Borders? Immigration in Open Source Projects, *Proceedings of the 4th International Workshop on Mining Software Repositories (MSR'07)*, pp. 6–13 (2007).
- [4] Bonaccorsi, A. and Rossi, C.: Comparing motivations of individual programmers and firms to take part in the open source movement: From community to business, *Knowledge, Technology & Policy*, Vol. 18, No. 4, pp. 40–64 (2006).
- [5] Breiman, L.: Random forests, *Machine learning*, Vol. 45, No. 1 (2001).
- [6] Canfora, G., Di Penta, M., Oliveto, R. and Panichella, S.: Who is going to mentor newcomers in open source projects?, *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering*, ACM, p. 44 (2012).
- [7] Fang, Y. and Neufeld, D.: Understanding sustained participation in open source software projects, *Journal of Management Information Systems*, Vol. 25, No. 4, pp. 9–50 (2009).
- [8] Hannebauer, C., Book, M. and Gruhn, V.: An exploratory study of contribution barriers experienced by newcomers to open source software projects,

- Proceedings of the 1st International Workshop on CrowdSourcing in Software Engineering*, ACM, pp. 11–14 (2014).
- [9] Hannebauer, C. and Gruhn, V.: Motivation of Newcomers to FLOSS Projects, *Proceedings of the 12th International Symposium on Open Collaboration*, ACM, p. 1 (2016).
 - [10] Hatton, L.: Power-law distributions of component size in general software systems, *IEEE Transactions on Software Engineering*, Vol. 35, No. 4, pp. 566–572 (2009).
 - [11] Ihara, A., Kamei, Y., Ohira, M., Hassan, A. E., Ubayashi, N. and Matsumoto, K.: Early identification of future committers in open source software projects, *Quality Software (QSIC’14), 2014 14th International Conference on*, IEEE, pp. 47–56 (2014).
 - [12] Jensen, C. and Scacchi, W.: Role migration and advancement processes in OSSD projects: a comparative case study, *Proceedings of the 29th International Conference on Software Engineering (ICSE’07)*, pp. 364–374 (2007).
 - [13] Jiang, J., Feng, F., Lian, X. and Zhang, L.: Long-Term Active Integrator Prediction in the Evaluation of Code Contributions.
 - [14] Jiang, Y., Adams, B. and German, D. M.: Will My Patch Make It? And How Fast?: Case Study on the Linux Kernel, *Proceedings of the 10th Working Conference on Mining Software Repositories*, pp. 101–110 (2013).
 - [15] Labuschagne, A. and Holmes, R.: Do onboarding programs work?, *Mining Software Repositories (MSR’15), 2015 IEEE/ACM 12th Working Conference on*, IEEE, pp. 381–385 (2015).
 - [16] McIntosh, S., Kamei, Y., Adams, B. and Hassan, A. E.: The Impact of Code Review Coverage and Code Review Participation on Software Quality: A Case Study of the Qt, VTK, and ITK Projects, *Proceedings of the 11th*

- Working Conference on Mining Software Repositories(MSR'14)*, pp. 192–201 (2014).
- [17] Nurolahzade, M., Nasehi, S. M., Khandkar, S. H. and Rawal, S.: The role of patch review in software evolution: an analysis of the mozilla firefox, *Proceedings of the Joint International and Annual ERCIM Workshops on Principles of Software Evolution and Software Evolution Workshops (IWPSE-Evol'09)*, pp. 9–18 (2009).
 - [18] Pinto, G., Steinmacher, I. and Gerosa, M. A.: More Common Than You Think: An In-depth Study of Casual Contributors, *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER'16)*, Vol. 1, pp. 112–123 (2016).
 - [19] Robles, G. and Gonzalez-Barahona, J. M.: Developer identification methods for integrated data from various sources, *ACM SIGSOFT Software Engineering Notes*, Vol. 30, No. 4, pp. 1–5 (2005).
 - [20] Samoladas, I., Stamelos, I., Angelis, L. and Oikonomou, A.: Open source software development should strive for even greater code maintainability, *Communications of the ACM*, Vol. 47, No. 10, pp. 83–87 (2004).
 - [21] Scott, A. J. and Knott, M.: A Cluster Analysis Method for Grouping Means in the Analysis of Variance, *Biometrics*, Vol. 30, No. 3, pp. 507–512 (1974).
 - [22] Steinmacher, I., Chaves, A. P. and Gerosa, M. A.: Why do newcomers abandon open source software projects?, *Proceedings of the International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE'13)*, pp. 25–32 (2013).
 - [23] Steinmacher, I., Conte, T., Gerosa, M. A. and Redmiles, D.: Social Barriers Faced by Newcomers Placing Their First Contribution in Open Source Software Projects, *Proceedings of the 18th ACM Conference on Computer Sup-*

ported Cooperative Work & Social Computing, CSCW '15, New York, NY, USA, ACM, pp. 1379–1392 (2015).

- [24] Steinmacher, I., Wiese, I., Conte, T. U. and Gerosa, M. A.: Increasing the self-efficacy of newcomers to Open Source Software projects, *Software Engineering (SBES), 2015 29th Brazilian Symposium on*, IEEE, pp. 160–169 (2015).
- [25] Suykens, J. A. and Vandewalle, J.: Least squares support vector machine classifiers, *Neural processing letters*, Vol. 9, No. 3, pp. 293–300 (1999).
- [26] Tao, Y., Han, D. and Kim, S.: Writing Acceptable Patches: An Empirical Study of Open Source Project Patches, *Proceedings of the International Conference on Software Maintenance and Evolution(ICSME'14)*, pp. 271–280 (2014).
- [27] Wu, C.-G., Gerlach, J. h. and Young, C. E.: An Empirical Analysis of Open Source Software Developers' motivation and continuance intentions, *Information and Management*, Vol. 44, No. 3, pp. 153–262 (2007).
- [28] Zhou, M. and Mockus, A.: Who will stay in the FLOSS community? modeling participant 's initial behavior, *Transactions on Software Engineering*, Vol. 41, No. 01, pp. 82–99 (2015).
- [29] Zhou, M. and Mockus, A.: Developer fluency: achieving true mastery in software projects, *Proceedings of the 18th International Symposium on Foundations of Software Engineering (FSE'10)*, pp. 137–146 (2010).
- [30] Zhou, M. and Mockus, A.: What make long term contributors: Willingness and opportunity in oss community, *Software Engineering (ICSE'12), 2012 34th International Conference on*, IEEE, pp. 518–528 (2012).

発表リスト

関連発表論文

- [1] 則兼卓人, 伊原彰紀, 平尾俊貴, 松本健一, “コードレビューアの指摘記録に基づく開発者の実装技術の成長に関する調査”, ソフトウェアエンジニアリングシンポジウム (SES2016), 2016 年.
- [2] 則兼卓人, 伊原彰紀, 平尾俊貴, 松本健一, “ソフトウェア工学分野における公開データとその活用 -ソフトウェア検証に関する公開データについて-”, 電子情報通信学会技術研究報告 (PRMU, パターン認識・メディア理解) , pp.73-78, 2016 年.
- [3] Norikane, T., Ihara, A. and Matsumoto, K., “Which Review Feedback Did Long-Term Contributors Get on OSS Projects?”, Proc. of the 24th IEEE International Conference on Software Analysis, Evolution, and Reengineering (SANER), 2017, (to appear).

その他の研究

- [1] 安藤聡志, 伊原彰紀, 関浩之, 平尾俊貴, 則兼卓人, 松本健一, “OSS 開発におけるパッチの特徴量を用いた再投稿要求の予測”, 研究報告 ソフトウェア工学 (SE) Vol.2016, No.4, pp.1-5, 2016 年.