

複数のオープンソースプロジェクトに参加する 開発者による貢献の分析

坂口 英司^{1,a)} 伊原 彰紀¹ 尾上 紗野¹ 畑 秀明¹ 松本 健一¹

概要: オープンソースソフトウェア (OSS) プロジェクトは、複数のプロジェクトに参加する開発者が OSS プロジェクトを成功に導いていると過去の研究では示唆している。しかし、具体的なエビデンスは示されていない。なぜなら、以前は多くのプロジェクトが独自で開発履歴を記録するサーバを構築していたため、複数のプロジェクトに参加する開発者の開発経験を正確に理解することが容易ではなかったからである。近年、多くの OSS プロジェクトがホスティングサイト GitHub を利用するようになり、複数のプロジェクトに参加する開発者の活動経験を調査することが容易になった。本論文では、複数のプロジェクトで活動する開発者が OSS プロジェクトに与える影響を明らかにするために、GitHub に記録される開発者の各プロジェクトへの貢献を調査する。

An Analysis of Developers' Activities Contributed to Multi-open Source Projects

SAKAGUCHI HIDESHI^{1,a)} IHARA AKINORI¹ ONOUE SAYA¹ HATA HIDEAKI¹ MATSUMOTO KEN-ICHI¹

1. はじめに

オープンソースソフトウェア (OSS) プロジェクトでは、世界中の不特定多数のボランティア開発者が専門知識を共有し、高機能、且つ、高品質なソフトウェアの開発を目指している。しかしながら、ボランティア開発者が自由に開発したソースコードをソフトウェアに反映することは、ソフトウェアの品質低下につながる恐れがある。多数の開発者を取りまとめ、ソースコードの品質を判断できるコミッター^{*1}をはじめとする経験豊富な開発者が OSS プロジェクトには必要である [2]。

従来研究では、複数のプロジェクトに参加した経験を持つ開発者がコミッターに昇格しているという結果を得ている [3]。しかし、成功したプロジェクトには複数のプロジェ

クトに参加する開発者が多かったという分析だけで、具体的にどのような活動に貢献していたか示されていない。また、著者らが OSS プロジェクト (Gimp, Digikam, Samba, X.org) のコミッターに開発者を推薦方法についてインタビュー^{*2}を行い、その後、個別に追加インタビューとして、コミッターに推薦する場合に他プロジェクトでの経験を確認するかを尋ねた。その結果、多くの開発者は他のプロジェクトでの経験はコミッターを選出する上で重要な要因になると考えられているが、そのような経験を認めていないことが多いということが分かった。数年前まで多くのプロジェクトが独自で開発履歴を記録するサーバを構築していたため、開発者が複数のプロジェクトに参加しているか否かを調査することも容易ではなかった。

近年、多くの OSS プロジェクトがホスティングサイト GitHub を利用するようになり、開発者の活動経験を調査することが容易になり、開発者がどのプロジェクトに参加しているか把握することが容易になった。従って、本論文では、複数のプロジェクトで活動する経験豊富な開発

¹ 奈良先端科学技術大学院大学

Nara Institute of Science and Technology

^{a)} sakaguchi.hideshi.rv4@is.naist.jp

^{*1} OSS プロジェクトからバージョン管理システムにソースコードをアップロードする権限を与えられた開発者。大規模なプロジェクトに参加するコミッターはプロジェクトに参加する開発者の 5%にも満たない。

^{*2} Interview Form: <http://goo.gl/Lv5gj2>

者が OSS プロジェクトに与える影響を明らかにするために、GitHub に記録された活動履歴から開発者が参加するプロジェクト数、及び、各プロジェクトでどのような活動を行っているのかを調査する。

Github には、ブランチの作成やコミットの反映を始めとする活動が記録されている。しかし、各開発者が行った過去 300 件の活動しか記録されていないため、2013 年 10 月からの記録を調査した。本論文では収集した開発履歴をもとに、複数の OSS プロジェクトに参加する開発者と単一の OSS プロジェクトに参加する開発者との活動内容の違いを分析した。

本論文の構成は以下のとおりである。続く 2 章では関連研究を紹介し、本論文の立場を明らかにする。3 章では分析手法を述べ、4 章では分析結果を示す。5 章で考察を行い、最後に 6 章で本論文のまとめを行う。

2. 関連研究

Daniela らは、2 つの商用ソフトウェア開発プロジェクトが関係するタスク依存や、開発チーム間のコミュニケーションを分析した [1]。開発者は自身と似た知識領域の開発者と頻繁にコミュニケーションをとり、異なる知識を持つ開発者とのコミュニケーションでは、コア開発者が仲介人となっていることが分かった。Daniela らの論文が対象とするコア開発者は、本論文が対象とする開発者の一端であるが、コア開発者が複数のプロジェクトでどのような経験をしているか明らかにされていない。本論文では、複数のプロジェクトに参加する開発者がどのような経験を持ち、参加するプロジェクトにどのような影響を与えているか調査する。

Vibha らは、OSS プロジェクトにおいてコミッターに昇格する開発者を過去の活動履歴から分析している [3]。Vibha らの研究では、コミッターに昇格するためには信頼と信用が重要であるという前提のもと 3 つの仮説を立て、Eclipse プロジェクトを対象に確認している。Eclipse プロジェクトでは、バグ管理システムでの活動がコミッターの活動を理解するために有効であり、また、コミッターに昇格する開発者は他のプロジェクトにも参加していることを明らかにしている。しかしながら、複数のプロジェクトに参加する開発者の他プロジェクトでの経験、及び、Eclipse プロジェクトへの貢献を明らかにしていない。

3. 分析手法

本章では、Github に参加する開発者が各プロジェクトに貢献した活動を計測する手法、及び、分析手法を説明する。

3.1 GitHub から開発者の活動履歴を収集

Github では、各開発者がプロジェクトに問わず、い

表 1 収集したイベントとその概要

活動名	概要
CommitComment	コミットに対するコメント投稿回数
Create	ブランチやタグの作成回数
Delete	ブランチやタグの消去回数
Gollum	wiki ページの作成・編集回数
IssueComment	不具合報告に対するコメント投稿回数
Issues	不具合報告回数
PullRequest	コードリポジトリにコードの反映依頼回数
PullRequestReviewComment	投稿されたコードへのコメント投稿回数
Push	clone したコードへ新しいコードを反映した回数

れかのプロジェクトに貢献した過去 300 件の活動内容 *3 を記録している。Github に記録されている活動は 23 種類ある。開発者の活動は GitHub API を用いることで取得することができる。しかしながら、開発者の活動履歴は任意の時点において過去 300 件の活動しか記録されていないため、我々は 2013 年 10 月時点で記録された内容から 2014 年 2 月までの記録を取得した。記録されている活動には従来からコミッターを選出するために参考にされている技術的活動（パッチ投稿、レビュー）、社会的活動（コメント投稿）などが含まれており、開発者の活動を評価するための指標としては妥当である。ただし、23 種類の活動には、取得期間中に新たに計測が開始された活動、及び、計測することが中止された活動があり、これらは分析対象外とする。プロジェクトに貢献する活動とは呼べないものは排除し、表 1 に示す 9 種類のイベントを分析対象とする。

3.2 収集した活動履歴をもとにした分析

分析 1. 単一のプロジェクトに参加する開発者と複数のプロジェクトに参加する開発者の活動内容の違い

複数のプロジェクトに参加している開発者の経験を分析するために、単一のプロジェクトにしか参加しない開発者との活動量の違いを調査する。また、複数のプロジェクトに参加している開発者は、単一のプロジェクトに参加する開発者よりも貢献しているのかを明らかにする。本分析により、複数のプロジェクトに参加する開発者がプロジェクトにとって必要な人物であるかを知ることができる。

分析 2. 複数のプロジェクトに参加する開発者のプロジェクトによる活動内容の違い

複数のプロジェクトに参加している多くの開発者がコミッターに昇格しているが、各プロジェクトでどのような経験をしているのかは明らかにされていない。本分析では、開発者の各プロジェクトへの貢献する活動内容を分析する。

4. 分析

4.1 データセット

Github には膨大なプロジェクトが登録しているため、本論文では頻繁に更新されるプロジェクトを対象に分析を行った。2013 年 10 月時点、Github に登録されているプロジェクト *4 の中で、コミット数が 100 以上の開発者

*3 Github ではイベントと呼ばれる。

*4 <https://github.com/trending>

表 2 単一/複数のプロジェクトに貢献する開発者数

	単一のプロジェクト で活動している	複数のプロジェクト で活動している
人数 (割合)	59 人 (9.6%)	554 人 (90.4%)

表 3 単一/複数のプロジェクトに参加する開発者の平均活動量
 (括弧内は中央値)

イベント名	単一のプロジェクト で活動する開発者	複数のプロジェクト で活動する開発者
CommitComment	2.10(0)	0.80(0)
Create	9.19(1)	2.27(0)
Delete	3.02(0)	0.82(0)
Gollum	0.03(0)	0.17(0)
IssueComment	6.13(0)	7.89(0)
Issues	0.13(0)	1.67(0)
PullRequest	2.84(0)	2.99(0)
PullRequest ReviewComment	1.11(0)	1.70(0)
Push	40.84(1)	11.53(1)

が 5 人以上存在する 26 プロジェクト (angular.js, ansible, bootstrap, cocos2d-x, discourse, django, elasticsearch, git, gitlabhq, homebrew, intellij-community, ipython, jenkins, jquery, libgdx, linux, node, pandas, rails, reddit, ruby, scikit-learn, Specs, spring-framework, togetherjs, xbmc) を対象に、2013 年 10 月から 2014 年 2 月に当該プロジェクトに参加した開発者を分析した。

表 2 は、単一/複数のプロジェクトに参加する開発者数を示す。613 人の開発者は対象プロジェクトのいずれかに参加しており、その中で、554 人 (90.4%) は複数の OSS プロジェクトに参加し、59 人 (9.6%) は単一のプロジェクトに参加している。

4.2 分析結果

分析 1. 単一のプロジェクトに参加する開発者と複数のプロジェクトに参加する開発者の活動内容の違い

表 3 は、単一のプロジェクトに参加している開発者と複数のプロジェクトに参加している開発者の平均活動量を示す。太字で示されている活動は、単一のプロジェクトに参加している開発者と複数のプロジェクトに参加している開発者の活動量の差をマンホイットニーの U 検定により統計的有意差 (5%水準) を確認した活動である。分析の結果、技術的貢献 (Create)、社会的貢献 (IssueComment) は統計的有意差を確認した。従って、複数のプロジェクトに参加する開発者は、単一のプロジェクトに参加する開発者より技術的貢献、及び、社会的貢献を行っている。

分析 2. 複数のプロジェクトに参加する開発者のプロジェクトによる活動内容の違い

図 1 は、複数の OSS プロジェクトに参加する開発者の中で、活動の種類にかかわらず活動量の最も高い 5 人を対象に、開発者が積極的に参加する 5 プロジェクトの活動内容の違いを示す。各プロジェクトに貢献した活動量の合計を 100%とし、各活動量の割合を分析する。分析の結果から、開発者によって活動内容が異なることがわかる。例えば、dev.E はいずれのプロジェクトに対しても PushEvent, PullRequestEvent, IssuesEvent, IssueCommentEvent, DeleteEvent を同じ割合で貢献しており、技術的貢献、社会的貢献ともに経験を持つ。一方で、dev.C 氏は homebrew, homebrew, gith.. プロジェクトに対しては様々な貢献をしているが、他のプロジェクトに対しては pushEvent のみ貢献している。dev.B は社会的貢献よりも技術的貢献の経験を持つ。従って、開発者によって、プロジェクトへの貢献は異なっており、単純に複数のプロジェクトに貢献しているからといって同等の貢献が期待できるとは限らない。

5. 考察

5.1 複数プロジェクトに参加する開発者による貢献

複数のプロジェクトに参加する開発者がコミッターに昇格していることが多い [3]。しかしながら、分析 2 の結果から、開発者によってプロジェクトへの貢献の種類、貢献数は異なる。今後の研究でプロジェクトの特徴と開発者の関係が見つかれば、開発者が各プロジェクトでどのような活動に貢献する予測することができると考えられる。また、プロジェクトにとって有益な貢献を期待される開発者には早い時期にコミッターの権限を与えてプロジェクトに定着させることができると考えられる。具体的な例では、図 1 において dev.A は不具合報告、不具合報告に対するコメント投稿を頻繁に行っているため新たにプロジェクトに参加したときも不具合報告などを積極的に活動することが期待される。

5.2 制約

本分析では、頻繁に更新されるプロジェクトにおける積極的に活動する 5 名の開発者のみに対象者を絞って分析を行った。たった 5 人の分析にもかかわらず、dev.E のようにどのプロジェクトでも同様な活動頻度で貢献している開発者もいれば不具合に関連する活動を中心に行っている開発者を発見することができた。今後は、開発者の活動によって自動分類すること、そして、時系列に開発者の活動の変化とプロジェクト参加の変化を分析することを検討している。

6. まとめ

従来研究では、複数の OSS プロジェクトに参加している開発者がプロジェクトの成功に寄与していることを示し

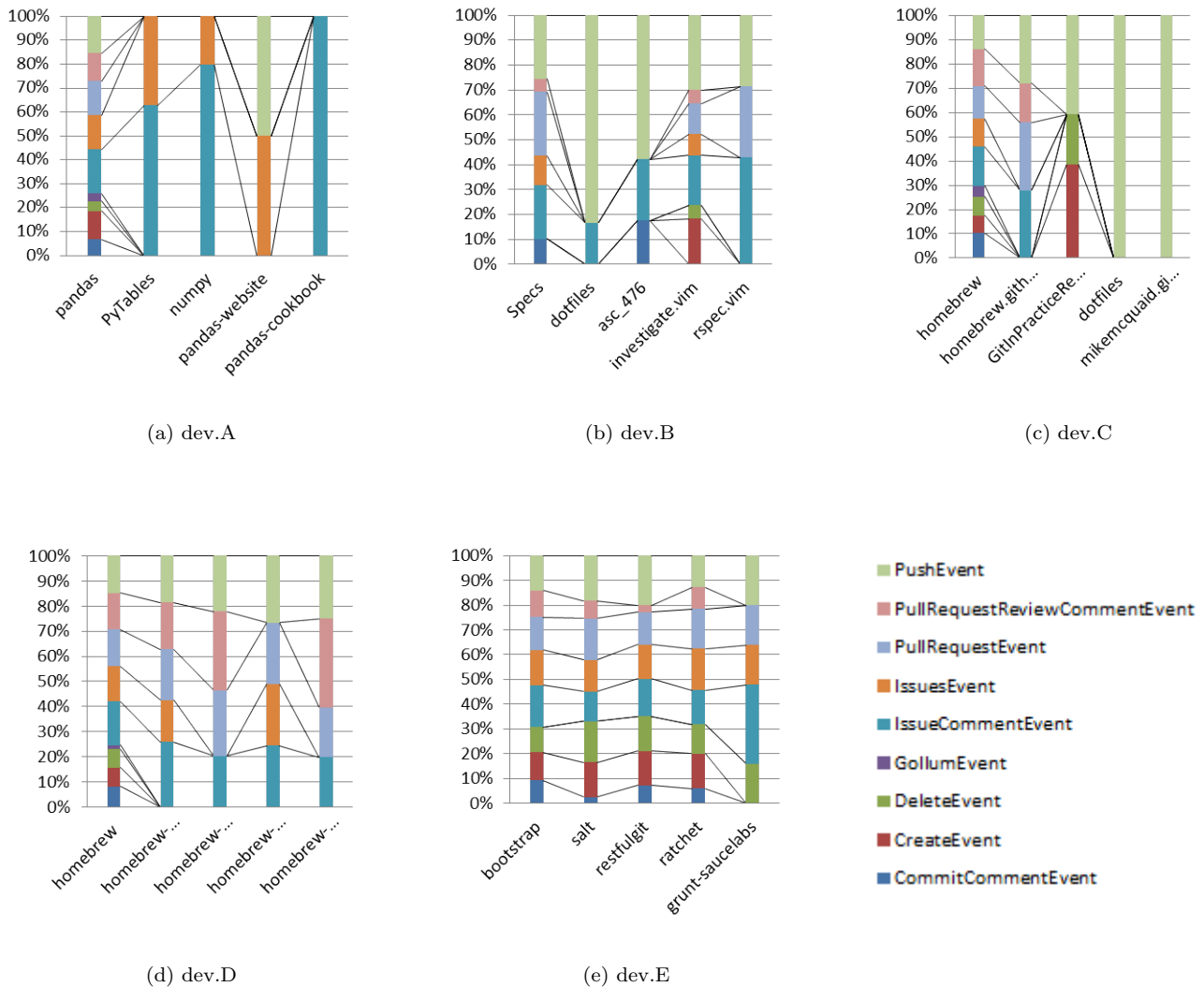


図 1 複数の OSS プロジェクトに参加する開発者の、各 OSS プロジェクトでの活動内容

ていたが、プロジェクト内での具体的な活動や貢献は明らかにされていなかった。近年、GitHub をはじめとしたホスティングサイトの普及によって、複数のプロジェクトに貢献する開発者の活動履歴を容易に分析することが可能になったことをきっかけに、本論文では、複数の OSS プロジェクトに貢献する開発者が OSS プロジェクトに与える影響を分析した。分析の結果、以下の知見を得た。

- 複数のプロジェクトに参加する開発者は技術的貢献 (Create)、社会的貢献 (IssueComment) を単一プロジェクトに参加する開発者より積極的に行っている。
- 複数の OSS プロジェクトに貢献する開発者の活動は一様ではなく、開発者によって異なる。

今後、複数の OS プロジェクトに参加する開発者の活動内容からプロジェクトへの貢献を明らかにする。

謝辞 本研究の一部は、文部科学省科学研究補助費 (若手 B: 課題番号 25730045, 挑戦的萌芽: 課題番号 26540029) による助成を受けた。

参考文献

- [1] Daniela Damian, Remko Helms. The role of domain knowledge and cross-functional communication in socio-technical coordination. In the 2013 International Conference on Software Engineering (ICSE'13), 2013.
- [2] K. Fogel. Producing open source software: how to run successful free software project. O'Reilly Media, Sebastopol, CA, 2005.
- [3] S. S. Vibha Singhal Sinha, Senthil Mani. Entering the circle of trust: developer initiation as committers in open-source projects. In the 8th Working Conference on Mining Software Repositories (MSR'11), 2011.