

A First Look at Good First Issues on GitHub

Xin Tan, Minghui Zhou, Zeyu Sun
Peking University

論文情報

Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference
and Symposium on the Foundations of Software Engineering (ESEC/FSE 2020).
Association for Computing Machinery, New York, NY, USA, pp 398–409.

紹介者：堀口日向

- OSS開発プロジェクトは，
ボランティア開発者からの貢献に依存[1]
⇒ 常に新規開発者が必要[2]
- 新規開発者は，
自身のキャリアアップやスキルアップのため，
プロジェクトへ参加することに積極的[3]

しかし，新規開発者(特にOSS開発に不慣れな開発者)にとって
プロジェクトへの貢献や，定着には障壁が存在[4][5]
↳ オンボーディング

[1] Hoda Baytiyeh and Jay Pfaffman. 2010. Volunteers in Wikipedia: Why the community matters. *Journal of Educational Technology & Society* 13, 2 (2010), 128–140.
[2] Mohammad Y. Allaho and Wang-Chien Lee. 2013. Analyzing the Social Ties and Structure of Contributors in Open Source Software Community. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. Association for Computing Machinery, New York, NY, USA, 56–60.
[3] J. Feller, B. Fitzgerald, S. A. Hissam, and K. R. Huff. 2007. Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. MITP, USA, 3–21.
[4] Amanda Lee, Jeffrey C. Carver, and Amiangshu Bosu. 2017. Understanding the Impressions, Motivations, and Barriers of One Time Code Contributors to FLOSS Projects: A Survey. In *Proceedings of the 39th International Conference on Software Engineering*. IEEE Press, USA, 187–197.
[5] Minghui Zhou. 2019. Onboarding and Retaining of Contributors in FLOSS Ecosystem. In *Towards Engineering Free/Libre Open Source Software (FLOSS) Ecosystems for Impact and Sustainability*. Springer, USA, 107–117.

- オンボーディングの障壁の1つとして、新規開発者向けの課題を見つけることが困難[6]
- GitHubではこの障壁を緩和するため、新規開発者向けの課題に**Good First Issue**ラベルの付与を推奨

しかし、Good First Issue(GFI)が、現状

- どの程度活用されているのか
- どのような効果があるのか
- どのような問題があるか

といったことは明らかでない

[6] Igor Steinmacher, Ana Paula Chaves, Tayana Uchoa Conte, and Marco Aurelio Gerosa. 2014. Preliminary empirical identification of barriers faced by newcomers to Open Source Software projects. In 2014 Brazilian Symposium on Software Engineering. IEEE, USA, 51–60.

目的とリサーチクエスト

目的

GFIを活用した新規開発者のオンボーディングをより効果的に支援するための知見を得る

リサーチクエスト

- RQ1：GFIを活用するプロジェクトはどの程度一般的か？
- RQ2：GFIはどのように解決されるか？
- RQ3：GFIの有用性に関する要因と問題点は何か？
- RQ4：適切なGFIを特定する方法は？

データセット

分析対象プロジェクトの選択

GHTorrentをベースに，プロジェクトをフィルタ

使用言語：JavaScript, Python, Ruby, C (GitHub上で最もよく使われている)

以下の条件のプロジェクトを除外

- Issue数が50未満
- "CONTRIBUTING.md"ファイルが存在しない
- コミュニケーション用の言語が英語ではない

最終的に，**計816プロジェクト**が対象

GFIの特定方法

新規開発者向けの課題を表すラベルを目視で抽出 (計113件)



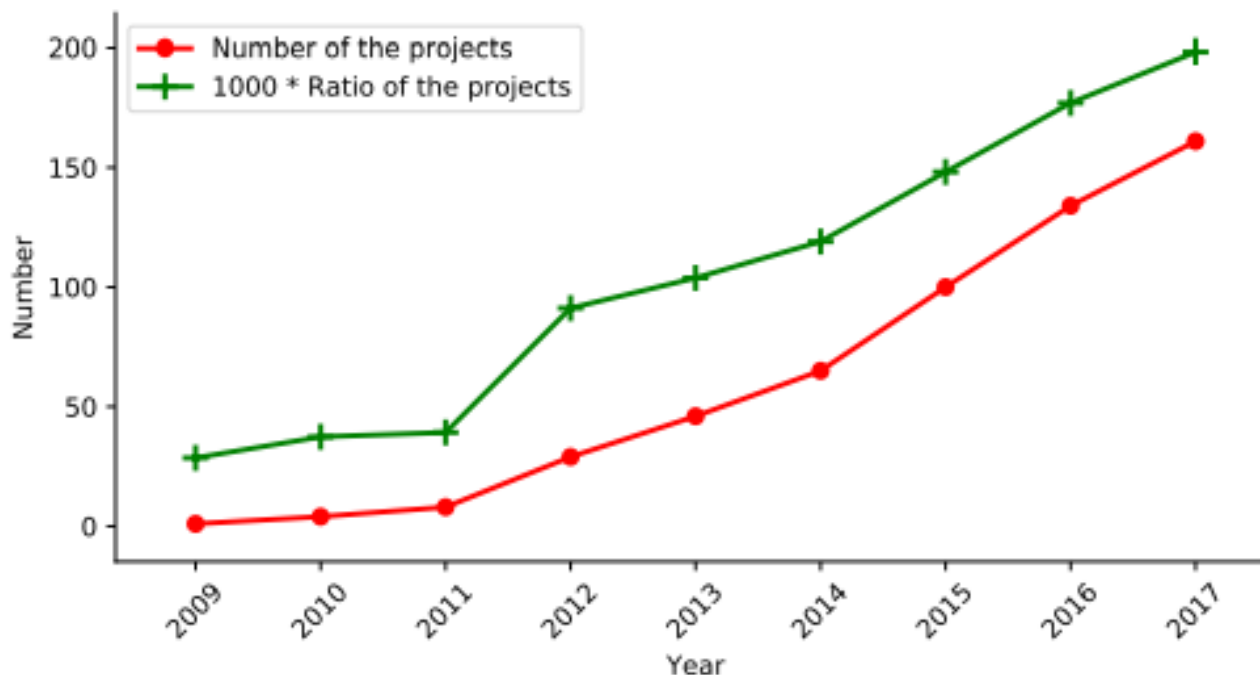
RQ1：GFIを活用するプロジェクトはどの程度一般的か？

分析の観点

- ・ GFIが存在するプロジェクト数の推移
- ・ プロジェクトごとのGFI数
- ・ GFIの活用とプロジェクトのスター数との相関

RQ1：GFIを活用するプロジェクトはどの程度一般的か？

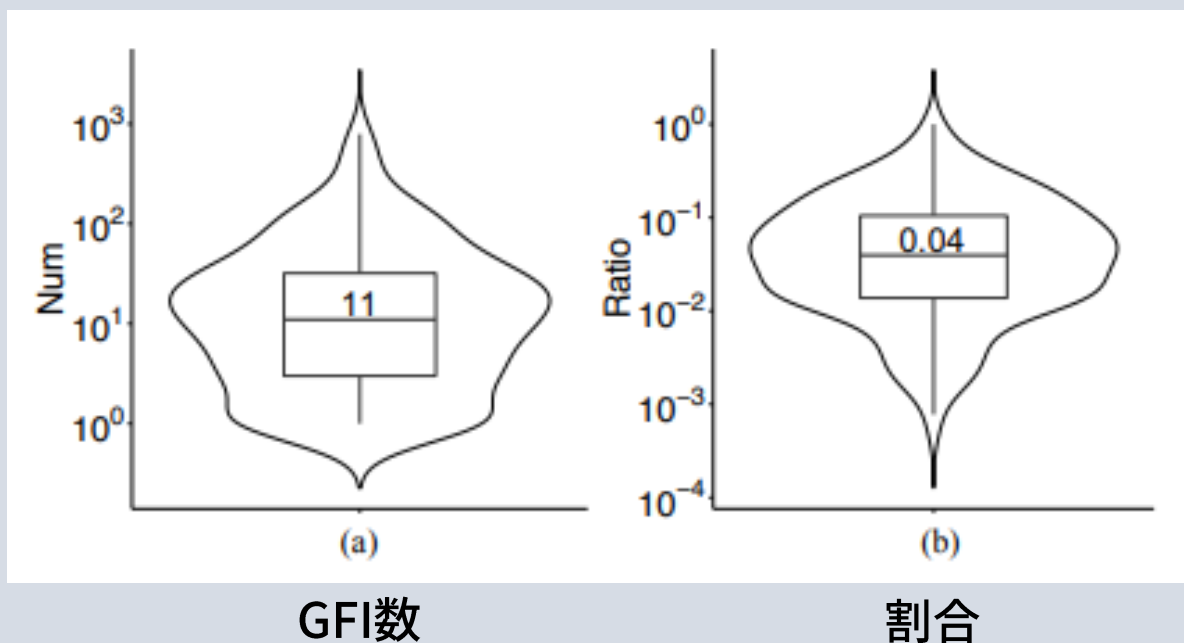
GFIが存在するプロジェクト数の推移



- GFIが存在するプロジェクト数
 - GFIが存在するプロジェクトの割合
- ⇒ 増加傾向

RQ1：GFIを活用するプロジェクトはどの程度一般的か？

プロジェクトごとのGFI数

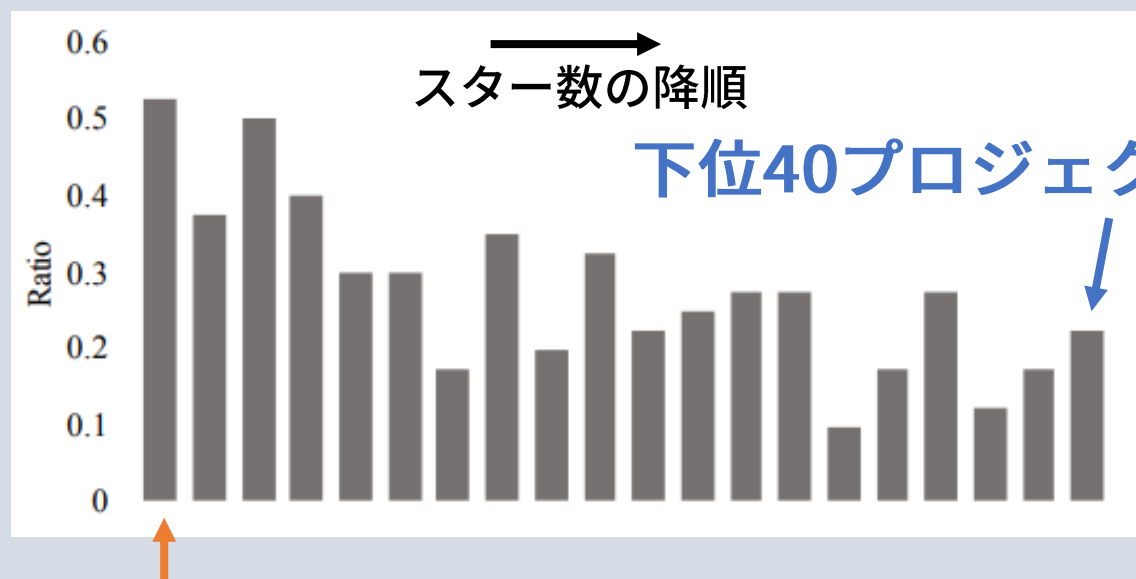


- プロジェクトによって大きく異なる
- 全体的な傾向として，中央値：11件 (4%)
⇒ GFIの数自体は通常のIssueに比べ少ない

RQ1：GFIを活用するプロジェクトはどの程度一般的か？

GFIの活用とプロジェクトのスター数との相関

40プロジェクトごとのGFIが存在するプロジェクトの割合



上位40プロジェクト：52.5%

- プロジェクトにGFIが存在するかと、スター数との相関
相関係数：0.19，非常に弱い正の相関

RQ2：GFIはどのように解決されるか？

分析の観点

- GFIの解決プロセス
- GFI解決の効果

RQ2：GFIはどのように解決されるか？

GFIの解決プロセス

GFI解決に至るまでの他の開発者とのやり取りなどを以下のメトリクスで定量化

- Issueが開かれてから解決されるまでの日数
- Issueの更新通知を有効に(サブスク)した開発者数
- 本文で言及された開発者数
- コミットで参照された回数
- コメントした開発者数
- コメント数

マンホイットニーのU検定（有意水準：0.001）で
GFIと通常のIssueを比較

RQ2：GFIはどのように解決されるか？

GFIの解決プロセス

	Median		Mean		r
	GFI's	Others	GFI's	Others	
解決されるまでの日数	33	3	143.97	75.55	0.32*
サブスクした開発者数	1	0	1.40	1.13	0.06*
本文で言及された開発者数	1	0	1.85	1.14	0.16*
コミットで参照された回数	0	0	1.49	0.48	0.24*
コメントした開発者数	2	2	2.88	2.43	0.12*
コメント数	4	3	5.60	4.63	0.11*

検定の結果，すべて有意差あり．ただし，

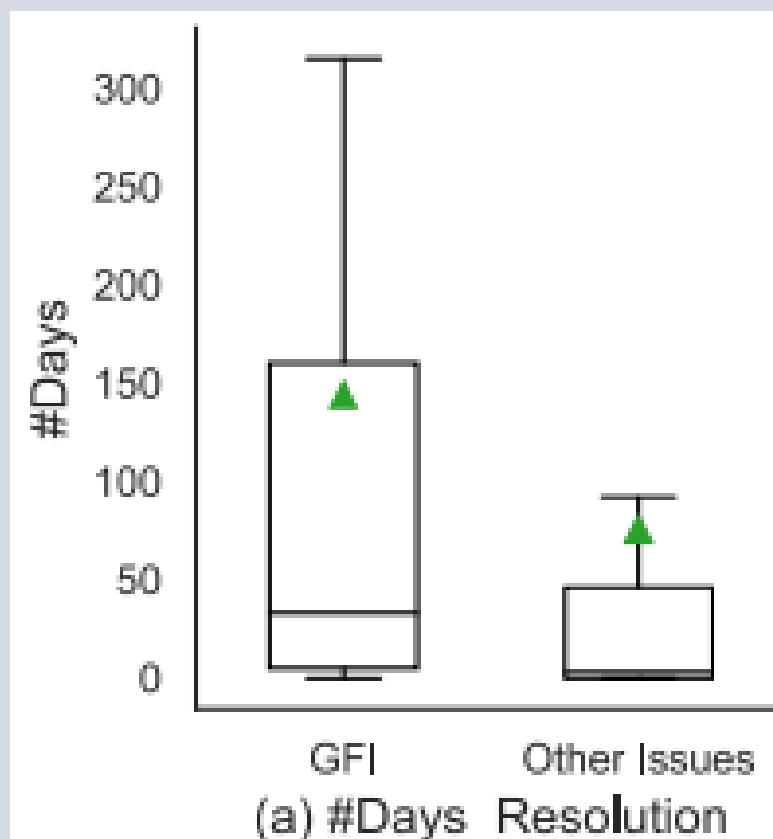
解決されるまでの日数以外は，中央値ほぼ同じ，効果量も小程度 (<0.3)

⇒ 解決されるまでの日数以外は，通常のIssue解決プロセスとほぼ同じ

RQ2 : GFIはどのように解決されるか？

GFIの解決プロセス

解決されるまでの日数の分布



GFIは通常のIssueに比べて
解決されるまでの時間が長い

考えられる原因

- 新規開発者が手を付けるまで
放置される
- 新規開発者が問題に着手して
から解決するまでに時間か
かかる

RQ2：GFIはどのように解決されるか？

GFI解決の効果

9,110個のGFIからランダムに200件選択

新規開発者：プロジェクトへのコミット数3回未満

GFIのコメントを定性的に分析

- GFIは実際に新規開発者によって取り組まれ、解決されるのか
- GFIに最初にコメントした新規開発者によって解決されるのか

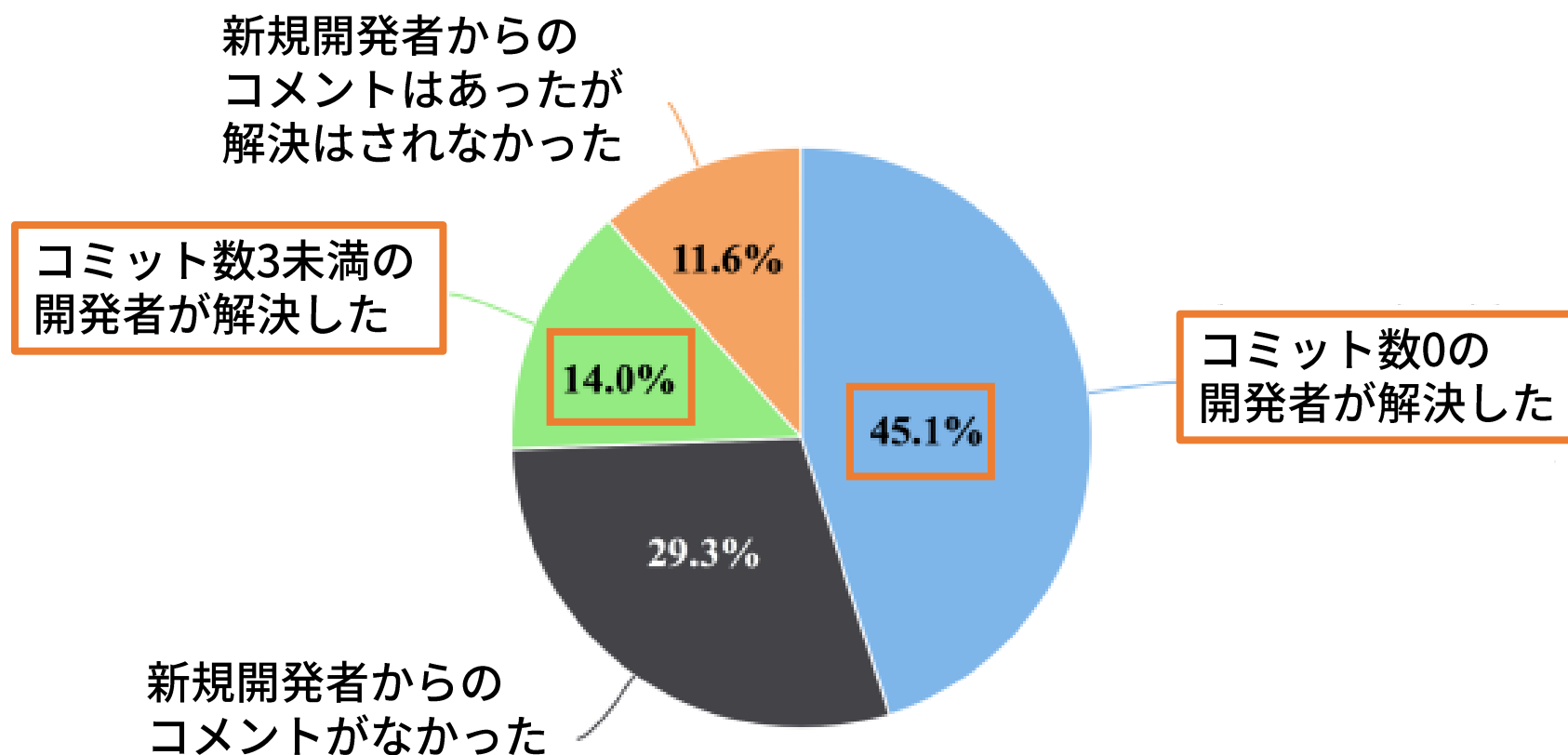
定量的な分析

- GFI解決後の新規開発者のプロジェクトへのコミット数

RQ2：GFIはどのように解決されるか？

GFI解決の効果

200件のGFIのうち、



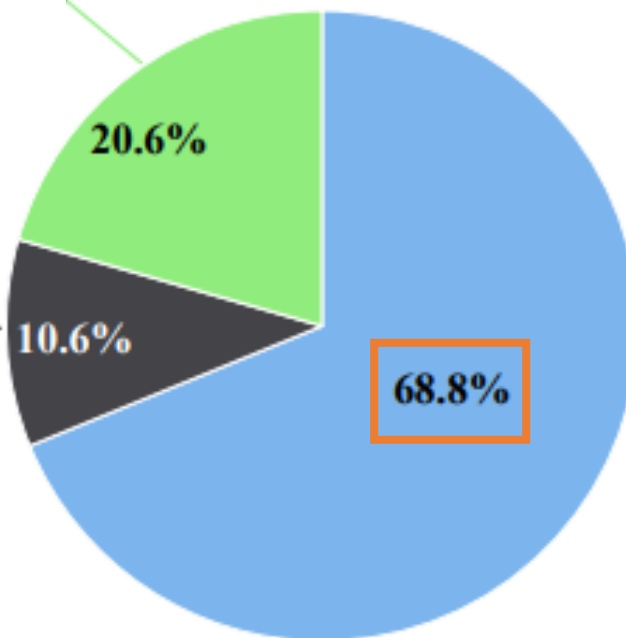
RQ2：GFIはどのように解決されるか？

GFI解決の効果

解決されたGFIのうち、

先に既存の開発者が
解決した

先に他の
新規開発者が
解決した

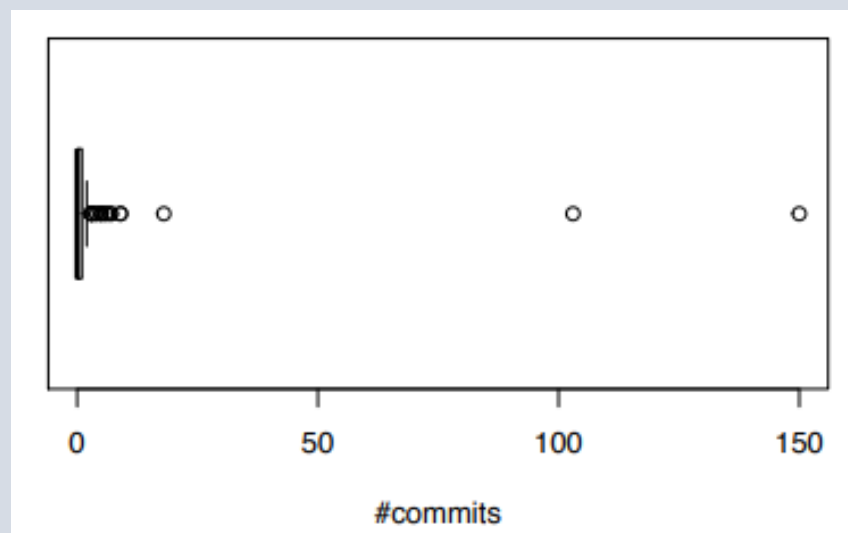


最初にコメントした
新規開発者がGFIを
解決できた割合

RQ2：GFIはどのように解決されるか？

GFI解決の効果

GFI解決後の新規開発者のプロジェクトへのコミット数



GFIを解決した新規開発者97人中,

- 1コミット：58.4%
- 10コミット以上：3人

⇒ 長期的な貢献を行う開発者は
ほぼいない

RQ3：GFIの有用性に関する要因と問題点は何か？

分析の観点

- ・新規開発者にとって解決しやすいGFIの要素
- ・GFIが解決されない原因

RQ3：GFIの有用性に関する要因と問題点は何か？

新規開発者にとって解決しやすいGFIの要素

新規開発者が解決できたかどうかを目的変数とした
ロジスティック回帰分析

説明変数に用いるメトリクスの観点

- GFIのタイトルや本文
- GFIを解決するための技術的な難易度
- プロジェクトの人気度
- GFIラベルを付けた開発者の経験

細かいメトリクスの説明は省略

結果は検定で有意差があったものののみ説明

RQ3 : GFIの有用性に関する要因と問題点は何か？

新規開発者にとって解決しやすいGFIの要素

Metrics	Est.	Std. Err.	z	Pr(> z)
(Intercept)	-6.61	0.85	-7.75	9.1e-15
length of the description	0.17	0.06	2.85	0.00436
#lines deleted	-0.16	0.06	-2.82	0.00487
#commits previously contributed	-0.11	0.03	-3.63	0.00031
#stars of the project	0.58	0.07	8.02	1.1e-15

- Issue本文の説明の長さ
長い方が解決しやすい

RQ3 : GFIの有用性に関する要因と問題点は何か？

新規開発者にとって解決しやすいGFIの要素

Metrics	Est.	Std. Err.	z	Pr(> z)
(Intercept)	-6.61	0.85	-7.75	9.1e-15
length of the description	0.17	0.06	2.85	0.00436
#lines deleted	-0.16	0.06	-2.82	0.00487
#commits previously contributed	-0.11	0.03	-3.63	0.00031
#stars of the project	0.58	0.07	8.02	1.1e-15

- Issue解決時に削除したコード行数
少ない方が解決しやすい

⇒ コードを削除する変更は，新規開発者にとってリスクを感じやすい

RQ3 : GFIの有用性に関する要因と問題点は何か？

新規開発者にとって解決しやすいGFIの要素

Metrics	Est.	Std. Err.	z	Pr(> z)
(Intercept)	-6.61	0.85	-7.75	9.1e-15
length of the description	0.17	0.06	2.85	0.00436
#lines deleted	-0.16	0.06	-2.82	0.00487
#commits previously contributed	-0.11	0.03	-3.63	0.00031
#stars of the project	0.58	0.07	8.02	1.1e-15

- GFIラベルを付与した開発者の過去のコミット数少ない方が解決しやすい
⇒ 熟練者は新規開発者との間にギャップがあり、適切なGFIのラベル付けが難しくなる

RQ3 : GFIの有用性に関する要因と問題点は何か？

新規開発者にとって解決しやすいGFIの要素

Metrics	Est.	Std. Err.	z	Pr(> z)
(Intercept)	-6.61	0.85	-7.75	9.1e-15
length of the description	0.17	0.06	2.85	0.00436
#lines deleted	-0.16	0.06	-2.82	0.00487
#commits previously contributed	-0.11	0.03	-3.63	0.00031
#stars of the project	0.58	0.07	8.02	1.1e-15

- プロジェクトのスター数
多い方が解決されやすい

⇒ プロジェクトの人气がないと，そもそも新規開発者の数が少ないため，GFIが解決されにくくなる

RQ3：GFIの有用性に関する要因と問題点は何か？

GFIが解決されない原因

インタビューによる調査

- ・ GFIが解決できなかった新規開発者に対して
「なぜGFIを解決できなかったのか？」
- ・ 解決されなかったIssueに関係したプロジェクトメンバー
に対して
「新規開発者がGFIを解決できなかった原因は
何だと思うか？」

新規開発者から9件，プロジェクトメンバーから11件の回答
回答を合わせて，テーマ分析による定性的分析

RQ3 : GFIの有用性に関する要因と問題点は何か？

GFIが解決されない原因

(■: Newcomer ■: Project member)

Problems	Categories	Responses
Insufficient GFIs in the project	Project	 11
Inappropriate GFIs	Mechanism	 10
Lacking of motivations to retain	Newcomer	 7
Hard to start on a new project	Newcomer	 6
Few newcomers in the project	Project	 5
Uneven level of newcomers' skill	Newcomer	 5
Getting snapped up soon	Mechanism	 3
Lacking of mentors	Mechanism	 2

- プロジェクト側の問題
 - GFIの数が少ない（RQ1での定量的な分析と一致）
 - 新規開発者が少ない

RQ3 : GFIの有用性に関する要因と問題点は何か？

GFIが解決されない原因

(■: Newcomer ■: Project member)

Problems	Categories	Responses
Insufficient GFIs in the project	Project	11
Inappropriate GFIs	Mechanism	10
Lacking of motivations to retain	Newcomer	7
Hard to start on a new project	Newcomer	6
Few newcomers in the project	Project	5
Uneven level of newcomers' skill	Newcomer	5
Getting snapped up soon	Mechanism	3
Lacking of mentors	Mechanism	2

- GFIの仕組みそのものの問題
 - 不適切なGFI (e.g. 問題点が明確でない)
 - すぐに他の人が解決してしまう
 - GFI解決を補助するメンターの不足

RQ3 : GFIの有用性に関する要因と問題点は何か？

GFIが解決されない原因

(■: Newcomer ■: Project member)

Problems	Categories	Responses
Insufficient GFIs in the project	Project	11
Inappropriate GFIs	Mechanism	10
Lacking of motivations to retain	Newcomer	7
Hard to start on a new project	Newcomer	6
Few newcomers in the project	Project	5
Uneven level of newcomers' skill	Newcomer	5
Getting snapped up soon	Mechanism	3
Lacking of mentors	Mechanism	2

- 新規開発者側の問題
 - モチベーションの維持
 - 新しいプロジェクトへの参加障壁
 - 新規開発者ごとのスキルの違い

RQ4：適切なGFIを特定する方法は？

分析の観点

- GFIラベル付けの基準
- GFIの本文に含まれるキーワード

RQ4：適切なGFIを特定する方法は？

GFIラベル付けの基準

ラベル付けを行ったことがあるプロジェクトメンバーに以下の質問を行う

- ・なぜそのIssueが新規開発者に適していると考えたか？
- ・IssueにGFIラベルを付与する際の判断基準は何か？

24人のプロジェクトメンバーが回答

過半数が言及したトピックについて詳細説明
他のトピックは紹介程度（論文には記載）

RQ4：適切なGFIを特定する方法は？

GFIラベル付けの基準

- ・ 変更すべき範囲がごく一部（20人/24人）
- ・ 問題が明確（16人/24人）
- ・ 必要なスキルが限定的（15人/24人）
- ・ 問題解決に必要な時間が短い（14/24人）
- ・ メンテナがサポート可能な問題（8人/24人）
- ・ 問題解決時に目で見えるような影響がある（3人/24人）
- ・ 問題の緊急性が低い（3人/24人）

RQ4：適切なGFIを特定する方法は？

GFIの本文に含まれるキーワード

解決されたGFIと解決されていないGFIを区別できるキーワードの抽出手順

1. GFIの本文からテキスト以外のコード片，URL，画像などを削除
2. 各GFIで単語のTF-IDFを計算
 - TF：各GFIの本文に含まれる特定の単語の出現頻度
 - IDF：特定の単語を本文に含むGFIの割合
 - $TF-IDF = TF * IDF \Rightarrow$ 各GFIを特徴づける単語の指標
3. TF-IDFが上位20単語を各GFIのキーワードとする
4. キーワードの正規化（レマタイズ）
5. GFI全体で出現頻度が高い上位6つのキーワードについて，
解決された/解決されていないGFIでのそれぞれの出現頻度を比較

RQ4：適切なGFIを特定する方法は？

GFIの本文に含まれるキーワード

GFI全体で出現頻度が高い上位6つのキーワード

(■: Solved ■: Unsolved)

Keywords	Comparison	Keywords	Comparison
version	 107	map	 37
documentation / doc	 89	platform	 36
return	 49	subsystem	 34

- version, platform, subsystem

問題が発生した環境を説明するためのキーワード

⇒ これらの情報が含まれている方が解決しやすい

RQ4：適切なGFIを特定する方法は？

GFIの本文に含まれるキーワード

GFI全体で出現頻度が高い上位6つのキーワード

(■: Solved ■: Unsolved)

Keywords	Comparison	Keywords	Comparison
version	 107	map	 37
documentation / doc	 89	platform	 36
return	 49	subsystem	 34

- return

“エラーを返す”，”～の値を返すべき”といった使われ方

⇒ 具体的なプログラムのふるまいが明記されている方が
解決しやすい

RQ4：適切なGFIを特定する方法は？

GFIの本文に含まれるキーワード

GFI全体で出現頻度が高い上位6つのキーワード

(■: Solved ■: Unsolved)

Keywords	Comparison	Keywords	Comparison
version	 107	map	 37
documentation / doc	 89	platform	 36
return	 49	subsystem	 34

- documentation

ドキュメントの編集や追加の必要性の説明

⇒ ドキュメント関連のGFIは解決しやすい

- map

論文内での言及はなし．解釈困難？

- GFIを活用するプロジェクトは年々増加
- GFIの解決プロセス
開かれてから閉じられるまでに時間がかかること以外、ほかのIssueと変わらない
- GFIの効果
GFIを解決しても、長期的にプロジェクトに貢献する開発者はほぼいない
- GFIの基準
問題が明確で十分な説明がされており、変更範囲や必要なスキルが限定的

現状のGFIは、プロジェクトへの利点が少ない？

- GFI解決後貢献が続かない（RQ3）のでは、プロジェクトにとって、GFIを用意するメリットがない
- プロジェクト，新規開発者の双方にとってWin-Winになるような施策を考えないと，GFIの仕組み自体が崩壊するかも・・・
- こんなGFIの使い方すれば，双方にとって役に立つ！と結論付けられる研究が必要