4K-03

GitHub のデータを利用した コードレビュー時間に関連した属性の分析

大山義人† 大場みち子‡

公立はこだて未来大学大学院 システム情報科研究科† 公立はこだて未来大学‡

1. 研究背景

システム開発プロセスにおいて，コードレビューが多くの開発現場で取り入れられている．コードレビューとは，IPA によると「開発者担当者が書いたソースコードを閲読してセキュリティ脆弱性あるいはそのきざしを読み取る作業」[1]とされている．その中でも GitHub などのバージョン管理サービスを使用したコードレビューが一般的である．GitHub はコードの変更内容を本番環境に反映する前にレビューできる PullRequest 機能を持っており，コードレビューに適したサービスである．コードレビューは『レビュワーのスキル』やレビュー対象の『ソースコードの属性』など多くの要因が関係しているため，レビューにかかる時間などの『レビュー結果』を推定しにくい．『レビュワーのスキル』や『ソースコードの属性』，『レビュー結果』の関係を分析することで，コードレビューの質や所要時間の見積もりができる可能性が高い．コードレビューに関する指標化と，それを用いた推定は多くの研究でされている．そのため，GitHub のソースコードを分析することで，どのような情報がコードレビュー時間に影響があるのかを明らかにする．

2. 研究目的

本研究の目的は，ソースコードのレビュー時間とソースコードの属性を分析することである．

3. 関連研究

コードレビュー時間に関係する研究を述べる．

ソースコードの属性として，循環的複雑度[2]と呼ばれるソースコードの複雑さを測定する手法がある．1＋分岐（if，for，while など）の数で表すことができ，その数字の大小によってコードの構造がどれだけ複雑か推定できる．循環的複雑度はソースコードの可読性を数量的に表すことを表している．

GitHub と StackOverflow というエンジニアコミュニティサイトの活動実績から開発者のスキルをスコア化した研究[3]がある．開発者の GitHub での活動や StackOverflow での回答から，開発者がどの言語や技術ジャンルに詳しいか分析をしている．

コードレビュー分析のためにレビュー内容のカテゴリ分けした研究[4]がある．コードレビューのコメントをどのような種類のコメントか手動でタグ付けし，カテゴリ分けをしている．具体的な分類の種類には，バグや欠陥の発見，コードの改善，代わりのアプローチの提案などが考えられる．この手法を利用することにより，コードレビュー内容の分析をすることができる．

前年度の研究[5]として，コードレビュー時間とソースコード属性の関係性を重回帰分析を用いて分析した．その結果，レビュー時間は属人的で，特にレビュイーに依存していた．具体的には，コードレビュー時間にはコードレビューを受ける人の過去の平均レビュー時間が最も関係が深いことがわかった．しかし，コードレビュー時間を十分な精度で推定することはできなかった．

その理由として，リポジトリの情報とコードレビュー時間の関係性を分析するのにあたって，説明変数が適切ではなかったことが考えられる．

4. 研究課題

レビュー時間は，多くの要素が関わっている．コードレビュー時間を正確に予測するためには，関係の深い属性を発見するのが難しいという課題がある．

5. 解決アプローチ

GitHub を対象にどのような要素がコードレビュー時間に関係しているのかを分析する．特に，前年度の研究で関係性があるとされた要素や説明変数として含められなかった要素について分析する．具体的には，レビューされる人，リポジトリが所属している組織・プロジェクトなどが候補として挙げられる．

分析では，レビュー時間とそれに関係する要素との関係性を調査する．レビュー時間は，多変量解析の 1 つである重回帰分析を利用して推定する．目的変数であるレビューにかかる時間，要因を分析する説明変数ともに数量データであ

Analysis of Attributes Related to Code Review Time Using GitHub Data

†Yoshito Oyama †Michiko Oba

†School of System Information Science, Future University Hakodate

‡ Future University Hakodate

るため，重回帰分析を利用する．

6. 実験

6.1 実験目的

本実験の目的はソースコードのどのような属性がコードレビュー時間と関係しているのかを明らかにすることである．

6.2 実験対象

GitHub での対象とするリポジトリの選定基準はスター数が多いこと，コードレビューが行われている OSS リポジトリであることである．スター数とは，GitHub 上で個人がリポジトリを覚えるため，コントリビューターに感謝を示すために使用されており，リポジトリの信頼度の目安として使用することができる．以上の条件を満たすリポジトリを 6 つ選択した．また，分析に利用する PullRequest の年代が大幅に異なると，使用されてるツールやオンライン上でのコードレビューの普及度合いが異なるため，2015～2019 年の５年間のデータを使用する．データ取得方法として，GitHub REST API v3 を使用する．API クライアント PyGitHub を使用して，GitHub REST API v3 を利用しデータを対象リポジトリから取得する．

6.3 実験方法

実験は GitHub からデータ取得，データの前処理，重回帰分析を使用してコードレビュー時間との関係性の分析順番で行う．

取得したデータは統計的解析手法で分析しやすいように，ノイズデータを取り除くなどの前処理をする．その後，目的変数をコードレビュー時間として，重回帰分析し，コードレビュー時間に関する要因分析・推定する．分析に使用する目的変数はコードレビューにかかった時間，説明変数はレビューされる人，リポジトリが所属する組織，リポジトリが所属するプロジェクト，PullRequest のコミット数の 4 つである．これらの説明変数は，前年度の研究で関係性があったとされるレビューされる人の情報，リポジトリ固有のルールによる影響を調査するために，リポジトリが所属する組織とプロジェクトの情報，PullRequest あたりのコミット数が増えるとレビュー時間が増加しそうなことからコミット数を選択した．を推定モデルの精度を測るためにデータの 25%を切り分けておき，テストデータとして利用する．

6.4 実験結果

実験の結果，説明変数の中では特にレビューを受ける人の相関係数が 0.52 と高く，相関関係がみられた．しかし，決定係数が 0.32 と低く十分な精度を得ることは出来なかった．

7. まとめ

本研究では GitHub の情報を利用して，コードレビュー時間の関係性を重回帰分析によって分析した．実験では GitHub のリポジトリのデータを使用し，コードレビュー時間と関係している要素の分析を行った．その結果，レビューされる人が最もレビュー時間と相関関係がみられた．しかし，コードレビュー時間を十分な精度で推定することはできなかった．

今後の展望として，相関係数が高いレビュイーのリストからレビュー時間に関係がありそうな共通点を見つけ，分析して行きたいと考えている．また精度が低い理由の一つとして，データ数が少ないことが考えられる．リポジトリの数を増やせば，レビュー時間との精度の高い関係性分析が可能になると考えている．コードレビュー時間と関係の深いソースコードの属性が明らかになった場合，コードレビュー時間を推定するモデルを作成し，それを用いてコードレビュー時間推定によるタスク管理支援システムを作成する予定である．

参考文献

[1] IPA:IPA ISEC セキュア・プログラミング講座：C/C++言語編 第 2 章 脆弱性回避策とソフトウェア開発工程：ソースコードレビュー，IPA(オンライン),入手先 <https://www.ipa.go.jp/security/awareness/vendor/programmingv2/contents/c103.html>(参照 2021-1-5)

[2] Yang, Cheng, Xun-hui Zhang, Ling-bin Zeng, et al.: RevRec: A Two-Layer Reviewer Recommendation Algorithm in Pull-Based Development Mode, Journal of Central South University, Vol.25, No.5, pp.1129-43 , (2018).

[3] Bacchelli, Alberto, and Christian Bird.: Expectations, Outcomes, and Challenges of Modern Code Review, Proc. Proceedings of the 2013 International Conference on Software Engineering(ICSE '13), USA: IEEE Press, pp.712-721, (2013).

[4] McCabe, T. J.: A Complexity Measure, IEEE Transactions on Software Engineering, Vol.2, No.4, pp.308-20, (1976).

[5] 大山義人，大場みち子：GitHub のデータを利用したコードレビュー時間の推定，第 82 回全国大会講演論文集，Vol.2020，No.1，pp.199-200（2020）．