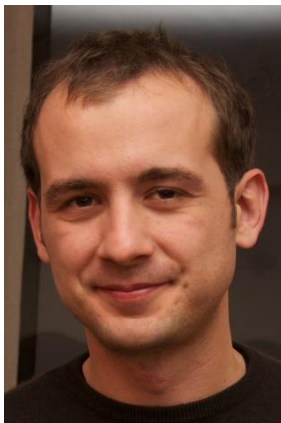


An Exploratory Study of the Pull-Based Software Development Model

ICSE'14 p345-355

Georgios Gousios Martin Pinzger Arie van Deursen



ABSTRACT

- 分散バージョン管理システムは新しい分散ソフトウェア開発手法をリードしている
 - GitHubなどが推奨しているPull-Basedでの開発
- この論文では, GHTorrentをつかって291のプロジェクトのPull-Based開発を調査して次のことを見つけた.
 - Pull Requestは改良が速い.
 - コミュニティがうまくかみ合うことが多くなる.
 - contributionが取り込まれるまでの時間が少なくなる.
- また, Pull Requestがリジェクトされる理由も調査した.

1.INTRODUCTION

- 多くの開発者がGitHubやBitbucketのようなPull-Based開発をサポートする所に移住してきている.
- Pull Requestの理解を深くし, Pull-Based開発モデルの与える影響の要因を分析するために以下のRQを行う
 - RQ1: Pull-Based開発モデルはどれくらい人気なのか？
 - RQ2: PRのライフサイクルの特徴は何か？
 - RQ3: どのような要因がPRをマージするかの決定やその決定までの時間に影響を与えているのか？
 - RQ4: なぜいくつかのPRがマージされないのか？
- データ: GitHub上の291のプロジェクト(合計で166,884のPR)

2.BACKGROUND

- GitHubの機能について書かれている.
- 2つのcontribution
 - リポジトリを共有する場合(branchを用いる)
 - Pull Requests(Forkを用いる)
 - GitHubでは両方とも対応している.
- GitHubのPRのマージ方法
 - 1.GitHubの機能を使う-自動的にマージが行われる
 - 2.Gitの機能を使ったマージ
 - 3.パッチをコミットする

3. RESEARCH DESIGN

- RQ1:

統計的にPRを分析する. 特に, どれくらいの数のプロジェクトがPRを使っているのか, どのようにPRがissuesに関連付けられているのか

- RQ2 and RQ3:

関連研究のデータからもっとも良い特徴を得るために相互相関分析を行う. そしてその特徴を使いPRの特徴の詳細な統計的分析を行う.

機械学習を使って要因についてのクラス分けをする. それが正しいのかの評価実験も行う.

- RQ4:

ランダムにとってきたマージされていないPRを詳しく分析する.

4.DATA

4.1 GitHub Data

- GHTorrentのデータを使う
 - 200,000超のプロジェクト(1,900,000のPRを含む)
 - 期間:2012年2月から2013年8月
- GHTorrentはGitHubApiを使ってデータを取得している.

4.DATA

4.2 Pull Request Project Sample

- プロジェクトの選び方
 - テストを含むプロジェクトで, Ruby, Python, Java, Scalaに限る.
 - 最近のコミットがPRからのものであること.
 - フレームワークかアプリケーションのプロジェクトであること.
- マージの発見
 - GitHubの機能の外でマージされた場合を見つける
- 最終的なデータセット
 - 291プロジェクト(99 Python, 91 Java, 87 Ruby, 14 Scala)

4.DATA

4.3 Qualitative Data

- どのようにPRが使われているのか, またなぜPRがマージされないのかの詳しい調査のためにコーディングとテーマ分析に基づくPRをランダムにサンプルをとってくる
- クローズされたPRの理由を調査のために1つのコーダーに使われた100のPR. (bootstrapping sample)
- 定めたカテゴリを有効にするために3つのコーダーに使われた100のPR. (cross-validation sample)
- 150のランダムに選ばれたPR.
- 合計350のPRをデータセットとした.

5. POPULARITY OF PULL-BASED DEVELOPMENT

- RQ1:
 - GitHubにおいて14%のリポジトリがPRを使っている
 - PRを使うプロジェクトは共有リポジトリも同様に使われている
 - PRの数は多くなってきているが、PRを使うリポジトリの割合は少し減っている
- 65~79%は一人の開発者しかいないプロジェクト
- 年間を通してPRを使う活発なリポジトリは14%

6. PULL REQUEST LIFECYCLE

- RQ2:
 - 多くのPRは行数が20以下で, 取り込まれるかの判断が1日以内
 - PRにつくコメントは3つが平均
 - コードレビューはPRがマージされるのに影響がある
 - テストを含むコードはマージするかの決定や決定までの時間に影響はない
 - PRは特別な処理でなければcontributor, core team 関係なく受け取る

7. MERGING AND MERGE TIME

- RQ3:
 - PRのマージの決定は主にPRが修正されるかに影響される
 - マージまでの時間は開発者の最近の活動, プロジェクトやテストの範囲の大きさ, そしてプロジェクトの解放具合に影響を受ける

8. UMMERGED PULL REQUESTS

- RQ4:
 - 53%のPRはPull-Basedの分散開発環境の性質によってリジェクトされる
 - 技術的な理由によってリジェクトされるのは13%だけである.

Table 3: Reasons for closing pull requests without merging.

Reason	Description	%
obsolete	The PR is no longer relevant, as the project has progressed.	4
conflict	There feature is currently being implemented by other PR or in another branch.	5
superseded	A new PR solves the problem better.	18
duplicate	The functionality had been in the project prior to the submission of the PR	2
superfluous	PR doesn't solve an existing problem or add a feature needed by the project.	6
deferred	Proposed change delayed for further investigation in the future.	8
process	The PR does not follow the correct project conventions for sending and handling pull requests.	9
tests	Tests failed to run.	1
incorrect implementation	The implementation of the feature is incorrect, missing or not following project standards.	13
merged	The PR was identified as merged by the human examiner	19
unknown	The PR could not be classified due to lacking information	15

9.DISCUSSION

- Contributorからすれば, ホットなプロジェクトであれば80%のPRがマージされるので良い
- Threats to Validity
 - マージまでの時間を調べる際に使った, 統計手法が違うデータセットでは使うことができないだろう
 - GHTorrentのデータベースの形式によって, 結果が異なる場合がある
 - この論文ではプロジェクトの大きさ関係なく, すべてのPRなどのデータを同じに扱っている

11. CONCLUSION

- この論文の目的はGitHubのデータを使ったPull-Based開発の深い理解
- 百万以上のPRから選んできた数百のPRを比較した.
- 以下の知見が得られた
 - 活発なプロジェクトでは14%しかPRを使っていなかった.
 - PRの60%は1日以内に処理される
 - マージの決定はPRの修正や最近のコードの修正に影響される
 - マージまでの時間は開発者の履歴やプロジェクトのテスト範囲などに影響される.
 - リジェクトされたPRの53% Pull-Based開発の性質で、技術的によるものは13%しかない
- データセットはGitHubにある
 - <https://github.com/gousiosg/pullreqs>

感想

- マージするまでの時間を求める時や、マージを決定する要因を決めるためにつかっていた機械学習などの意味が取れなかった.
- うまく日本語になおせない.
- 前段階としてGHTorrentをつくって、それからこの論文を書いているのがすごいと思った
- プロジェクトを絞ってPRを見るとまた別の結果があるのかなと思った.