B-007

ソフトウェア開発者の貢献タイプの分析
Analysis of Contribution Types of Software Developers

池本 和靖† 門田 暁人†
Kazuyasu Ikemoto Akito Monden

1. はじめに

今日のソフトウェア開発は，多数の開発者からの自発的な提案（pull request，パッチ投稿など）に基づいて進めるソーシャルコーディングと呼ばれる開発形態が広まっており，その代表的なプラットフォームとして GitHub が利用されている．GitHub は 2000 万件を超える Git リポジトリをホスティングしており，参加する開発は 100 万人を超える．このような開発形態においては，プロジェクトの成否・盛衰は，プロジェクトに 参加する開発者の人数，能力や開発における貢献タイプなどに依存する．本研究では，GitHub におけるソフトウェア開発者の貢献タイプの分類・分析，およびその貢献の度合いについて分析を行う．

従来からも GitHub 上の開発者を分析する試みは行われており，クラスタリングによる分析[1]，人工ピラミッドに基づく分析[2]，プロジェクト参入者と退出者に基づく分析[4]などがある．本稿では，多数の活動に基づいて貢献タイプを区別する 2 つのメトリクスを定義し，分類する点が異なる．

2. 分析データ

本研究では，GitHub 上で開発者ごとに記録されている活動履歴を分析することで分析を行う．活動履歴は GitHub API などから取得することができる．ただし，GitHub API から取得できるデータには制限があるため，本研究においては GHTorrent プロジェクト[3]において収集・蓄積されている GitHub データを使用する．

GHTorrent では，GitHub 上の全イベント(pull request，commit，issue など)を取得し， MySQL および MongoDB 形式のデータベースに蓄積している．本研究では，MSR 2014 Mining Challenge Dataset として公開されている，GHTorrent dataset のサブセット（90 プロジェクト）のデータセットを用いるた検討・分析を行う．

GHTorrent では，commit，issue，pull request などの活動ごとにテーブルが定義されており，活動者のユーザ id が記録されている．ユーザ id に基づいて情報を紐づけることで，開発者ごとの情報を集約することが可能である．

3. 分析手法

GitHub 上のユーザの活動には，権限を持つリポジトリへの commit，他ユーザのリポジトリに対する pull request，他ユーザからの pull request に対する merge，issue の assign，close や reopen といったアクション，commit や issue などに対する comment の付与などがあり，これらの活動から貢献タイプを区別することを考える．そのために，GHTorrent におけるユーザ id ごとに次の活動量を集約する．

- commit の数
- pull request の数
- issue の報告数，issue に対するアクション（assign や close など）ごとの数
- commit，pull request，issue のそれぞれに対するコメントを付与した数
- pull request をマージした数

これらの情報から，開発者の貢献タイプを分析するにあたり，2 つの評価指標を設けた．

1 つ目の評価指標は，コーディング／ディスカッション指向度である．ソフトウェア開発にあたってコーディングを行う人が必要であるが，例えばバグ報告やその解決に向けた議論など，ディスカッションを主として行う開発者もプロジェクトには必要である．本稿では，commit と pull request が多いほどコーディング指向度が高いとみなし，各種コメントや issue 報告数が多いほどディスカッション指向度が高いとみなす．

2 つ目の評価指標は，プロジェクトにおける開発者のコア（中心）度である．ソフトウェア開発においては，開発を主導するコア開発者のみならず，コア開発者をサポートする（非コア）開発者が必要となる．コア開発者は，issue の管理を行ったり，多数の開発者から寄せられた pull request を吟味し，プロジェクトのメインリポジトリに merge するといった作業を行うことが考えられる．報告された issue を解決することも，コアに近い開発者の作業であると考えられる．一方，非コア開発者は，pull request を行ったり，見つけたバグを issue として報告することで，プロジェクトの発展に貢献することが考えられる．本稿では，issue の close や reopen を行ったり，pull request をマージした数や issue を assigned された数が多いほどコア度が高いとみなし，pull request と issue の報告を行っているほどコア度が低い（非コア度が高い）とみなす．

コーディング／ディスカッション指向度は，(commit と pull request の総数)-(comment と issue 報告数の総数)，コア度は (close，reopen，merge 数，assigned された数の総数)-(pull request と issue 報告の総数)と定義する．ただし，外れ値の影響を緩和するために，次の正規化を行う．それぞれの評価指標の値を自然対数変換する．負の数については絶対値を取り，対数変換後に負の符号を付与する．

開発者を分析するにあたっては，GHTorrent に記録されている開発者の多くは活動履歴が 0 もしくはそれに近いため，対象をある程度の活動量を持った開発者に限定する．個々のソフトウェア開発プロジェクトについても，プロジェクトごとに参加者全員について同様の手法で分析を行い，プロジェクト間の比較を行うこととする．分析対象とするプロジェクトは，開発者が 10 人以上のプロジェクトとする．

† 岡山大学大学院自然科学研究科 Graduate School of Natural Science and Technology, Okayama University

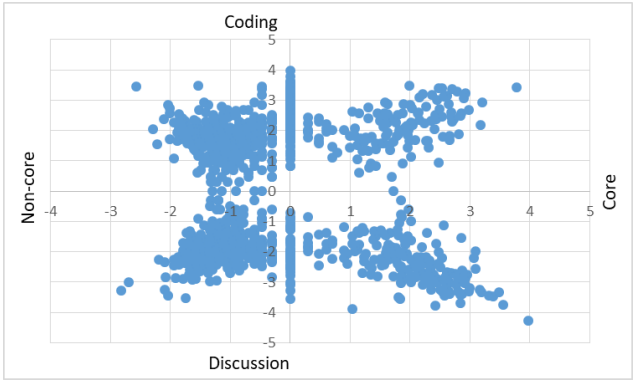

図 1. 活動量 100 以上の開発者の貢献タイプ

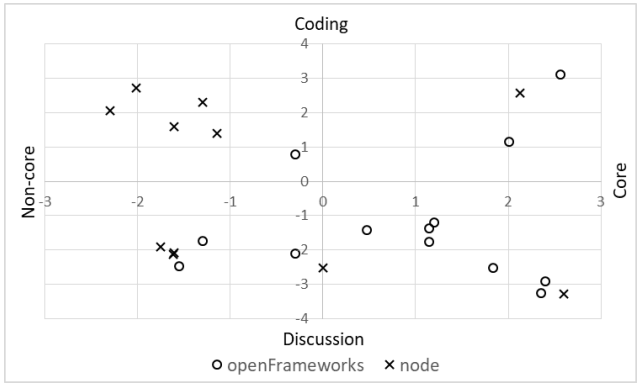

図 2. 2 つのプロジェクトでの貢献タイプの比較

4. 分析結果

4.1 開発者個人の分析結果

活動量の合計が 100 以上の開発者（1388 人）を対象とした分析結果を図 1 に示す．Y 軸のプラス方向がコーディング指向，マイナス方向がディスカッション指向の度合いを表し，X 軸のプラス方向がコア，マイナス方向が非コアの度合いを表す．

図 1 より，開発者の貢献タイプは，第 1～第 4 象限のそれぞれに示される 4 つのタイプに明確に分かれている．また，図の中央付近に線状に並ぶ（Y 軸上の）多数の開発者は，コア活動数と非コア活動数に差がほとんどなかった開発者であるが，これらの大部分は X 軸方向の活動がほぼ 0 の開発者であった．つまり，pull request，merge，issue の報告，close，assigned，reopen といった活動を全く行っていない開発者である．これらの開発者は，pull request を全く扱わないプロジェクトや GitHub の issue の機能を全く使わないプロジェクトに属している可能性があり，今後，そのようなプロジェクトを除外して分析するなどの検討が必要である．各タイプの開発者は次のように特徴付けできる．

- 第 1 象限（右上）の開発者は，コア開発者かつコーディング志向であり，コーディングを行いながらプロジェクトを率いる，現場リーダーのような開発者であるといえる．
- 第 2 象限（左上）の開発者は，非コア開発者かつコーディング志向であり，多数の pull request や issue 報告によりプロジェクトを発展させる，寡黙なエキスパートといえる開発者である．
- 第 3 象限（左下）の開発者は，pull request や issue 方向を多数行いつつも，議論に数多く参加しており，雄弁なエキスパートまたはエンドユーザといえる開発者である．
- 第 4 象限（右下）の開発者は，コア開発者かつディスカッション志向であり，多数の議論に参加しつつ管理的活動を行っており，プロジェクト管理者的な開発者といえる．

Y 軸上の開発者を除くと，第一象限に 126 人，第二象限に 335 人，第三象限に 440 人，第四象限に 179 人となった．この結果より，コア開発者は非コア開発者と比較して若干人数が少なく，また貢献度についてもより幅広く分布していることがわかる．各タイプを構成する開発者の詳細な分析については今後の課題である．

4.2 プロジェクトごとの分析結果

開発者が 10 人以上のプロジェクトは 32 件あり，それらにおいて活動を行っている開発者のうち活動量が 100 を超える者は 399 人であった．プロジェクト間の比較の実例として，2 つのプロジェクト（openFrameworks，node）の分析結果を図 2 に示す．図 2 より，openFrameworks ではコアメンバと呼べる開発者が多く，node では逆に非コアメンバが多いことが分かる．node のように非コアメンバの方が多いプロジェクトは他にも多数存在していた．openFrameworks についてはコアメンバが 2 人しかいないが，そのうち 1 人はコーディング志向であり，もう 1 人はディスカッション志向であるために役割分担ができているともいえる．openFrameworks のタイプについては非コア的な開発者を広く募集し，逆に node のタイプについては非コアメンバをコアメンバへと照会させることで，プロジェクトのさらなる発展を見込める可能性がある．

5. まとめ

本研究では，OSS 開発における代表的なプラットフォームである GitHub 上の開発者を対象として，開発貢献のタイプ分類・貢献度の分析を行う方法を提案した．90 プロジェクトを対象とした分析を行った．その結果，貢献者は 4 つのタイプに明確に分かれていることが分かった．また，プロジェクトごとの分析を行うことで，各タイプの貢献者の過不足を認識できることが分かった．今後は，各タイプの貢献者の内訳についてより詳細な分析を行う予定である．

謝辞

本研究の一部は，科学研究費補助金（17H00731）の助成を受けたものである．

参考文献

[1] 尾上 紗野，畑 秀明，松本 健一，GitHub 上の活動履歴分析による開発者分類，情報処理学会論文誌, Vol. 56, No. 2，pp. 715-719, Feb. 2015.
[2] S. Onoue, H. Hata, A. Monden, and K. Matsumoto, Investigating and Projecting Population Structures in Open Source Software Projects: A Case Study of Projects in Github, IEICE Transactions on Information and Systems, Vol. E99-D, No. 5, pp. 1304-1315, May 2016.
[3] The GHTorrent project，http://ghtorrent.org/
[4] K. Yamashita, Y. Kamei1, S. McIntosh, A. E. Hassan, N. Ubayashi, Magnet or Sticky? Measuring Project Characteristics from the Perspective of Developer Attraction and Retention, Journal of Information Processing, Vol. 24, No. 2 pp. 339-348, 2016.