オープンソースソフトウェア開発における開発者あたりのコミット数のパレート分析

PM コース　矢吹研究室　1342100　春川 直幸

1　研究の背景

ソフトウェア開発では，複数のメンバが同時に開発を行うため，ファイルの最新バージョンが分からなくなる，同一ファイルに対する変更が競合する等の問題が発生する．このような問題を解決するため，バージョン管理システムを用いる [1]．バージョン管理システムとは，変更履歴を管理するシステムのことである．具体的にはソフトウェアのソースコードを書き足したり，変更したりする過程を記録していき，特定の段階まで戻ったり，誤って消してしまったファイルを復活させたりなど，変更履歴を管理するシステムはソフトウェア開発の現場において無くてはならない機能である [2].

バージョン管理システム (VCS:Version Control System) のソフトウェアの 1 つに Git がある．[3].

バージョン管理システムを提供するサービスに，GitHub がある．GitHub は Git のリモートリポジトリとさまざまな Web ツールを提供するサービスである．GitHub が人気を集めている理由は，Web ベースで提供させるコラボレーションツールにある．オンラインでのコミュニケーションを円滑に進めるために，GitHub では相談して課題を解決する Issue(イシュー) や改善案を手軽に提案できる Pull Request(プルリクエスト) などの機能が提供されている [3].

Git には修正をひと固まりにして保存することができる Commit(コミット) という機能がある．修正内容のほかに，修正日時，修正した人，修正した内容についてのコメントが含まれている．コミットを意味のある単位でまとめることで，コミット単位の差分を見たり，コミット単位で修正を取り消すことができる [3].

Git にはリポジトリのコミットされたログを確認できる git log コマンドというものがある．git log では誰がいつコミットやマージをしてどのような差分を発生したのか確認するができる．[2]．この git log コマンドを解析することによって，開発者の貢献度を求める．なお今回はコミット数を貢献度の基準とする．GitHub でのソフトウェア開発でもパレートの法則が成り立つのか調査する．

2　研究の目的

GitHub を用いたソフトウェア開発プロジェクトにおいて，コミット数を基準とするプロジェクトへの貢献度を調査，可視化し，結果を分析する．

3　プロジェクトマネジメントとの関連

コミットを基準としたプロジェクトへの貢献度を分析し，パレート図を作成することにより，GitHub を用いたソフトウェア開発プロジェクトでのプロジェクトの工程改善を検討し，改善できる問題を抽出する．ソフトウェア開発プロジェクトでの開発設計工程の品質向上を目的とするため，プロジェクトマネジメントの知識エリアにおいて，品質マネジメントに該当する．

4　研究の方法

本研究は 2 段階に分かれる．

1. GitHub 上のプロジェクトから，開発者数とコミット回数を調査する．
2. 調査したデータの分析をする．

調査したデータの分析は，デシル分析で行う．デシル分析とは，複数の事物や現象について、データを 10 等分に分け分析し，管理効率を高めようとする分析手法である．今回はデシル分析でコミット数の多い順に開発者を 10 等分して，各ランクのコミット構成比率を算出し，コミット貢献度を明らかにする．

5　現在の進捗状況

GitHub 上の 10 個のプロジェクトから，プロジェクトの開発者数、コミット数を調査し，デシル分析を行った．

開発者人数は最少 1 人，最大 3503 人だった．コミット数は最小 1，最大 54421 だった．

10 件のプロジェクトをデシル分析し平均を算出した．結果は，図 1 である．分析結果から，デシル 1 の構成比が 88 %になっており，約 9 割の成果は 1 割の開発者によって生み出されていることがわかった．

このことから，GitHub におけるソフトウェア開発では，パレートの法則とは違う結果になっていると言えるであろう．

こうなった要因として調査したすべてのプロジェクトでオーナーのコミット数が一位だったことが関係していると考える．コミット数が一位だった開発者のコミット構成比の平均を求めた結果，42.4 %だった．今回はマージコミットもコミット数のカウント対象に含んだのがオーナーのコミット数が多かった要因として考えられる．

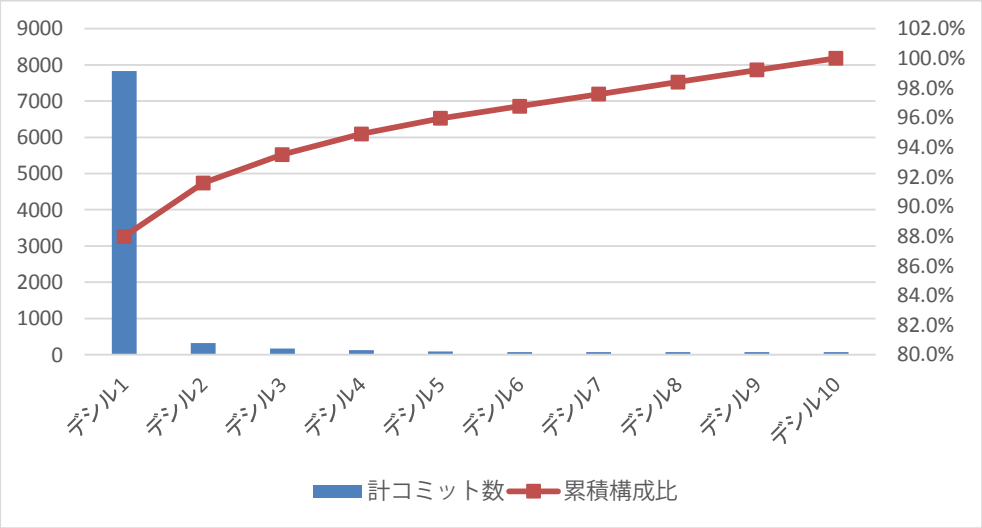

図 1　デシル分析 グラフ

6　今後の計画

以下のように研究を進める計画である．

1. 一定期間内のアクティブな開発者の数や，プロジェクトで使用しているワークフローなど貢献に関する基準について考える．
2. 調査対象となるプロジェクトの数を増やしたり，他の分析手法も用いて分析を行う．
3. 論文の執筆を行う．

参考文献

[1] 池田尚史, 藤倉和明, 井上史彰. チーム開発実践入門 共同作業を円滑に行うツール・メソッド. 技術評論社, 2014.

[2] 大塚弘記. GitHub 実践入門　Pull Request による開発の変革. 技術評論社, 2014.

[3] 塩谷啓, 紫竹佑騎, 原一成, 平木聡. Web 制作者のための GitHub の教科書チームの効率を最大化する共同開発ツール. インプレス, 2014.