

GitHub 上のソフトウェア開発のためのフロー推薦手法

若月 純[†] 矢吹 太朗

千葉工業大学 社会システム科学部 プロジェクトマネジメント学科[†]

1 序論

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。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 [1]。

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、バージョン管理システムに加え、branch、Pull Request といった開発を補助する機能を提供するサービスである。branch とは、履歴を分岐して記録していくためのものである。branch を用いることにより、同一リポジトリ内で、別々の作業を並行して行うことが出来るようになる。Pull Request とは、自分のリポジトリから相手のリポジトリへ、変更を取り込んでもらうための要求を出す機能である。Pull Request を用いることにより、変更が追加される前に確認することが出来る。

GitHub を使用する手順を開発フローと呼ぶ。現在わかっている開発フローの数は 13 個ある [2]。開発フローの例を 2 つあげる。GitHub Flow は、作業をする branch を作成し、完成したら統合する。といった開発フローである。この開発フローはとてもシンプルなため、開発フローを実施するまでの学習コストは抑えられるが、開発規模が大きい場合、Pull Request がたまりやすく、コードレビューに時間がかかってしまうことがある。Git Flow は、develop branch から作業用 branch を作成する。完成したら Pull Request を行い、作業用 branch を develop branch に統合する。リリースができるレベルになったら、リリース用 branch を作成し、作業をする。リリース作業が終了すると master ブランチに統合され、バージョンタグを打ってリリースする。といった開発フローである。branch 別にやる事が決まっているため管理は容易であるが、branch が複数あるため、Pull Request を異なった branch に送ってしまう等の人的ミスが発生す

る場合がある [3]。

このように開発フローは、メリットとデメリットがある。しかし、選択する基準は定められていないため、状況にあった開発フローを選択するのは難しい。そのため、適切でない開発フローを選択し、開発に悪影響を与える危険がある。このような事態を防ぐため、適切な開発フローを選択できるようにするための基準が求められる。

そこで、本研究は、適切な開発フローを選択できるようにするための基準を求めるため、GitHub 上のプロジェクトを対象に、採用されている開発フローと、開発フローの採用に関わると思われる項目を調査し、分析した。

2 目的

GitHub を用いたソフトウェア開発プロジェクトの性質において、適切な開発フローを選択できるようにするための基準を求める。この基準により、適切な開発フローを選択することができれば、プロジェクトを成功させるためのスケジュール、コスト、品質コントロールを補助することができる。

3 手法

本研究は 3 段階に分かれる。

1. GitHub 上のプロジェクトから、採用されている開発フローと、開発フローの採用に関わると思われる項目を調査する。
2. 調査結果を分析する。
3. 分析結果の精度と再現率を求める。

初めに、GitHub 上のプロジェクトから、開発フローと開発フローの採用に関わると思われる項目を調査する。開発フローは、GitHub 上の branch と Pull Request の特性から求められる。branch に stable が用いられている場合は、Stable Flow である。master branch から記述的な名前の branch がある場合は、GitHub Flow である。develop branch と release branch がある場合は、Git Flow である。バージョンごとに branch が作られている場合は、LINE Flow である。Pull Request に WIP がある場合は、WIP Flow である。開発フローの採用に関わると思

Workflow recommendation method for software development on GitHub

[†] Jun WAKATSUKI · Department of Project Management, Social System Sciences, Chiba Institute of Technology

最後に、分析結果の精度と再現率を求める。調査したデータをランダムに2種類に分ける。片方のデータで決定木分析を行う。決定木分析結果ともう片方のデータを照らし合わせて、決定木の精度と再現率を調べる。これを10回行い、信頼度95%区間の精度と再現率を用いる。

決定木の精度と再現率について記述する．32 件のデータをランダムに 22 件と 10 件に分け，精度と再現率を求めた結果，精度の平均は 41%，95% 信頼区間は 2656% だった．再現率の平均は 51%，95% 信頼区間は 2973% だった．

また、一日あたりのコミット数で分かれていることから、プロジェクトの生産性により適切なフローが代わることが言える。

本研究により、プロジェクトの性質において、開発フローを推薦する手法を確立した。しかし、精度と再現率が低いことから、開発フローの採用に関わる項目を調査できていない可能性がある。プロジェクトの生産性といった時系列データにより、適切なフローが代わることから、コミットが増える時間帯や人数の増加傾向等、他の時系列データを調査することが今後の課題である。

- [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] 小野寺航己. バージョン管理システムを活用するソフトウェア開発の開発フロー. 卒業論文, 千葉工業大学, 2015.
- [3] 大塚弘記. GitHub 実践入門 Pull Request による開発の变革. 技術評論社, 2014.

プロジェクトの性質は、プロジェクト経過日数、行数、ファイル数、バイト数、Watch 数、Star 数、Fork 数、Commit 数、branch 数、Release 数、人数、Open Issue 数、Closed Issue 数、Issue 数、Open PullRequest 数、Closed PullRequest 数、PullRequest 数、Label 数、Open Milestone