5J-04

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．開発フローの種類を調査した結果，13 種類あることがわかった．開発フローの例として，GitHub フローと Git フローを紹介する．

GitHub フローは，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 フローは，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 別にやること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 [2].

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．

そこで本研究では，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．そのために，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指標を調査し，分析する．

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．

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，開発フローの採用に関わると思われる指標と，採用されている開発フローを調査する．
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，開発フローの採用に関わると思われる指標を調査する．本研究で用いた指標は，プロジェクト経過日数，行数，ファイル数，バ

Workflow recommendation method for software development on GitHub.

† Jun WAKATSUKI(1242132hb@s.chibakoudai.jp)
‡ Department of Project Management, Social System Sciences, Chiba Institute of Tchnology.

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 Pull Request 数，Closed Pull Request 数，Pull Request 数，Label 数，Open Milestone 数，Closed Milestone 数，Milestone 数，Wiki 数，言語，1 日あたりの行数，1 日あたりの Commit 数，1 人日あたりの行数，1 人日あたりの Commit 数である．

次に，採用されている開発フローを調査する．開発フローの正解データは，人手で作成する．開発フローは，GitHub 上の branch と Pull Request の特性から求められる．ここでは，32 件のプロジェクトの，5 個の開発フロー（Git フローと GitHub フロー，LINE フロー，GitLab フロー，日本 CAW フロー）を特定した．たとえば，develop branch と release branch がある場合は，Git フローである．

プロジェクトに関する上述の指標から，開発フローを決めるための決定木の作成を試みる．具体的には，32 件のプロジェクトをランダムに 22 件の訓練データと 10 件のテストデータに分け，訓練データを用いて決定木を作成し，テストデータを用いてその性能を測定する．そのような実験を 10 回繰り返す．

4　結果

GitHub 上の 32 件のプロジェクトから，開発フローの採用に関わると思われる指標と，採用されている開発フローを調査し，決定木分析を行った．その結果が，図 1 である．

開発フローのわかっているプロジェクトを使って作成された開発フローの決定木が，開発フローが未知のプロジェクトの開発フローを予測できるかどうかを試したところ，精度は平均 41%（信頼区間は 26 ~ 56%），再現率は平均 51%（信頼区間は 29 ~ 73%）だった．

5　考察

図 1 は，全データを Star 数で分類している．Star とは，注目度を表す指標である．Star 数が 129 以上の場合，プロジェクトを主に branch で管理する Git フローが選択されている．Star 数が 129 未満の場合，プロジェクトを主に Pull Request で管理する日本 CAW フローが選択されている．ここから，開発人数だけでなく，チェックしているユーザ数により，最適な開発フローが異なることがわかる．

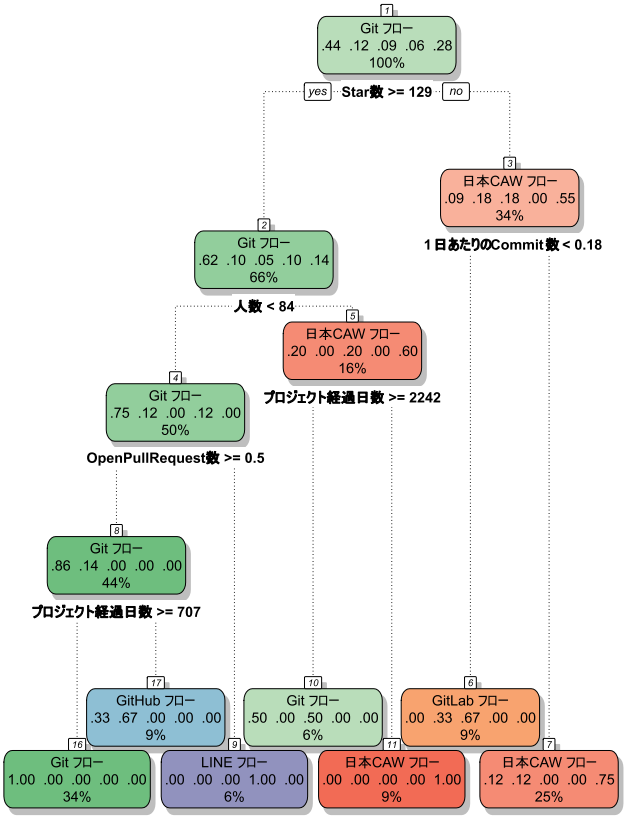

図 1　プロジェクトの性質により選択される開発フローの違い

また，1 日あたりの Commit 数といった，時系列データにより分類されていることが分かった．ここから，Commit 増加傾向や，人数の増減傾向等，他の時系列データを調査することで，より精度と再現率をあげられると考えられる．

6　結論

本研究では，決定木を用いた，開発フローを推薦する手法を提案した．現状では，精度と再現率が高いとは言えないが，このような手法を発展させることによって，GitHub の経験が少ないチームでの開発でも，最適なフローを決定できるようになることが期待される．

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.