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 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 [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．

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 GitLab が用いられている場合は，GitLab フローである．master branch から記述的な名前の branch がある場合は，GitHub フローである．develop branch と release branch がある場合は，Git フローである．バージョンごとに branch が作られている場合は，LINE フローである．Pull Request に日本 CAW がある場合は，日本 CAW フローである．開発フローの採用に関わると思われる項目は，GitHub 上のデータを用いる．GitHub 上のデータは，GitHub ブラウザに載っているデータと載っていないデータがある．載っている

Workflow recommendation method for software development on GitHub

† Jun WAKATSUKI・Department of Project Management, Social System Sciences, Chiba Institute of Tchnology

データは，そのまま用いる．載っていないデータは，リポジトリをクローンして調査する．

次に，調査結果を分析する．調査したデータの分析は，決定木分析を行う．決定木分析により，プロジェクトがどのような性質を持つときに，どの開発フローが使われているかを明らかにする．決定木分析の目的変数は開発フロー，説明変数は，開発フローの採用に関わると思われる項目を用いる．

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．

4　結果

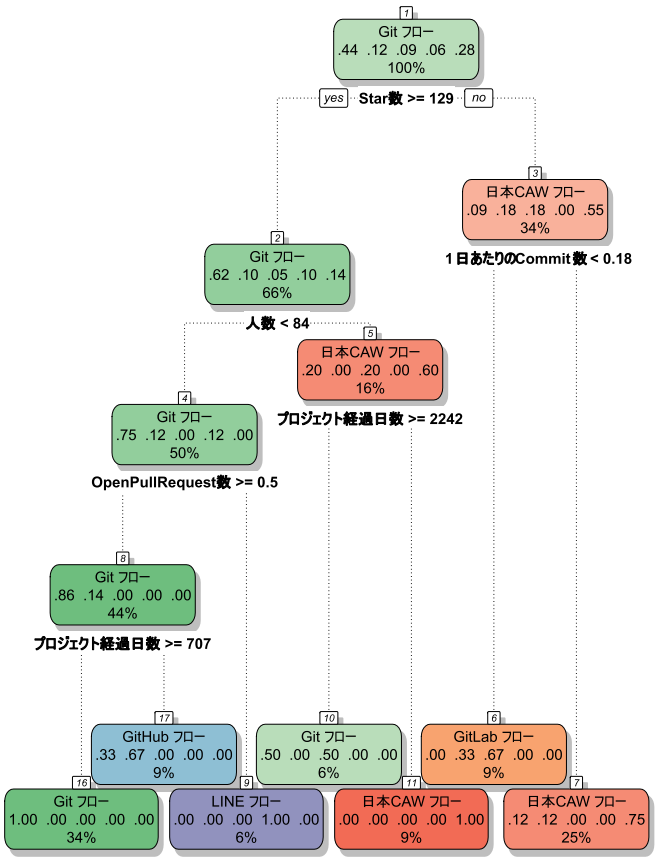

図 1　プロジェクトの性質により選択される開発フローの違い

GitHub 上の 32 個のプロジェクトから，プロジェクトの開発フローと性質を調査し，決定木分析を行った結果を，図 1 に示す．

開発フローは，Git フロー，GitHub フロー，LINE フロー，GitLab フロー，日本 CAW フローの 5 種類だった．

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 Pull Request 数，Closed Pull Request 数，Pull Request 数，Label 数，Open Milestone 数，Closed Milestone 数，Milestone 数，Wiki 数，言語，一日当たりの行数と Commit 数，1 人日当たりの行数と Commit 数を調査した．言語は 26 種類の項目を作成し，プロジェクトで使用している場合 1，使用していない場合 0 で判別した．26 種類に当てはまらない場合は，その他にまとめた．

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．精度の平均は 41%，95% 信頼区間は 26̃56% だった．再現率の平均は 51%，95% 信頼区間は 29̃73% 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．

また，一日あたりの 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，より適切な開発フローを選択できるようになると考えられる．

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確立した．精度と再現率が低いものの，本手法により，適切な開発フローを選択することが可能である．

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] 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 による開発の変革. 技術評論社, 2014.