

GitHub における開発フローの判別分析

PM コース 矢吹研究室 1242132 若月 純

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。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。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。というような開発フローである。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、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。というような開発フローである。branch 別にやるが決まっているため管理は容易であるが、複雑すぎるため、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。適切でない開発フローを選択すると、デメリットが多く発生する場合がある。このような事態を防ぐために、開発フローをプロジェクトの性質から選択する基準が必要である。しかし、選択する基準は定められていないため、適切な開発フローを選択する基準を明確にすることが必要である。

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提供する。

3 プロジェクトマネジメントとの関連

ソフトウェア開発プロジェクトにおいて、プロジェクトマネージャーは、QCD を達成させるためにスケジュール、コスト、品質コントロールを行う。これらのコントロールを、GitHub を用いた開発フローを導入することで、補助することが出来る。

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、プロジェクトの性質と開発フローを調査する。
2. 調査したデータの分析をする。

プロジェクトの性質は、開発人数、言語、行数を調査する。開発人数と言語は、GitHub 上のデータを用いる。

開発フローは、GitHub 上の branch、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。

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。

5 現在の進捗状況

GitHub 上の 32 個のプロジェクトから、プロジェクトの性質と開発フローを調査し、決定木分析を行った。

プロジェクトの性質は、開発人数、言語、行数を調査した。開発人数は最少 3 人、最大 156 人だった。言語は、Java、JavaScript、Ruby 等 19 種類だった。行数は最小 695、最大 875841 だった。

開発フローは、Git Flow、GitHub Flow、LINE Flow、Stable Flow、WIP Flow の 5 種類だった。

決定木分析結果は、図 1 である。分析結果から、プロジェクトの性質により選択される開発フローが明らかにされた。22 人以上の場合、GitHub Flow が使われていないことがわかる。GitHub Flow は、開発人数が多くなると管理が難しくなるためであると考えられる。また、言語により使われている開発フローに違いがあることがわかる。これは、言語により開発スピードが異なり、管理方法も異なるためであると考えられる。

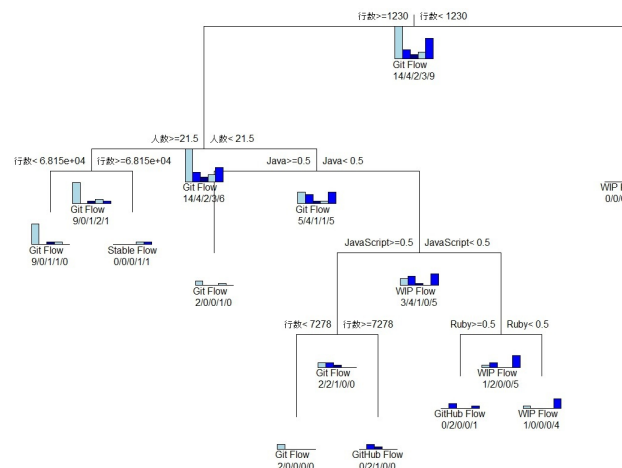


図 1 プロジェクトの性質により選択される開発フローの違い

6 今後の計画

現在の開発フローを選択する基準は、人数の場合、21 人以下と 22 人以上で別れている。そのためプロジェクトメンバが 5 人の場合と 15 人の場合、同一の開発フローを選んでしまう。しかし、プロジェクトメンバが 5 人の場合と 15 人の場合では、プロジェクトの性質は大きく異なり、同一の開発フローを選択するとデメリットが発生する可能性がある。そのような事態を防ぐため、開発フローを選択する基準を、より詳細なプロジェクトの性質で決められるようにする。

データ数が増えれば、より詳細なプロジェクトの性質で開発フローを選択する基準を明確にすることが出来ると仮定し、以下のように研究を進める計画である。

1. GitHub 上のプロジェクトから、プロジェクトの性質と開発フローを調査する。
2. 調査したデータを元に分析を行い、開発フローを選択する基準を明確にする。
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.