

GitHub における開発フローの判別分析

若月 純*

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 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 別にやる事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。

英語のタイトル名

* Jun WAKATSUKI(s1242132hb@s.chibakoudai.jp)

† Chiba Institute of Technology

