

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

ソフトウェア開発コース 矢吹研究室 1142032 小野寺航己

1. 研究の背景

当研究では、バージョン管理システム **GitHub** を用いたソフトウェア開発プロジェクトにおける、開発フローを対象とする。

バージョン管理システムは、何度も変更を加えたファイルであっても、作成日時や、編集日時、変更点を随時保管しておくことで、複数の人間が過去のファイルや、変更点の確認、ファイルの状態を復元することなどの、管理を可能とするシステムである。

代表的なバージョン管理システムとして **Git** がある。**Git** では、このシステムを使っている各利用者が自由にできる領域（ローカルリポジトリ）に、**Git** に保管されたデータや情報の全履歴を含んだ、完全な複製が作られる。このような仕組みがあるため、ネットワークにアクセスできなくても、履歴の閲覧や変更の記録といった、作業ができる。

この **Git** を用いたウェブシステムとして **GitHub** がある。**GitHub** は **Git** の機能を提供するウェブサービスであり、世界中の人々が自分の作品を保存、公開することもでき、ソフトウェア開発プロジェクトのための共有サービスでもある。

ソフトウェア開発では、一つのソフトウェアに対して複数のメンバが同時に編集を行うことで、複数のリリースバージョンがある中で、機能の追加やバージョン管理といった、作業を並行して行わなければならない状況がある。

Git にはそのような状況を支援する機能としてブランチがある。ブランチは、履歴の流れを分岐して記録できる機能で、同じソフトウェアに対して複数の変更を、履歴を保持しつつ同時におこなうことができる。この機能があるため、ソフトウェア開発のツールとして **GitHub** が用いられることが多い。

2. 研究目的

GitHub を用いたソフトウェア開発プロジェクト毎に、適切な開発フローを選択する基準を明確にする。

3. 研究方法

GitHub を用いる開発フローを網羅的に調査して、調査した開発フローを導入コストや導入リスクを明らかにしつつ、PMBOK の知識エリアの観点で分類・整理する。その後、プロジェクトの性

質に応じて適切な開発フローを選択できるようなガイドを作成する。

4. 進捗状況

現在 **GitHub** を用いた開発フローについて調査している。調査内容として **GitHub Flow**[1], **Git Flow**[1], はてなブログチームでの開発フロー[2], の3種類の開発フローを記述する。

GitHub Flow は、開発中ソフトウェアのブランチ（マスターブランチ）から、追加機能や修正するためのブランチを作成し、その作成されたブランチに修正を加え、メンバ全員からレビューして許可が下りたときに、マスターブランチに統合する、という手順で行われる。利点は、シンプルな流れのため、高速で開発を進められることである。

Git Flow は、開発ソフトウェアの編集、作成したソフトウェアのリリース作業といったように目的ごとにブランチが作成され、関係のあるブランチ同士での工程が終了し次第、次の目的を持つブランチに移る、もしくは統合するという流れを繰り返す、という手順で行われる。利点は、ソフトウェア開発者の世界に馴染みがあるアジャイル型開発と流れが似ているため、ソフトウェア開発者からすれば、流れが理解しやすいということである。

はてなブログチームの開発フローは、マスターブランチから開発用ブランチを作りそこから機能ごとに複数の作業用ブランチを作っていく、作業ブランチでの作業が終わるたびに開発用ブランチに統合し、ある程度作業用ブランチを統合したら、開発用ブランチをマスターブランチに統合するという、**Git Flow** を簡略化したような手順で行われる。

当研究は、このような複数の開発フローがある中、ソフトウェア開発プロジェクトの性質に応じた適切な開発フローを選択できるようにするための、基準を調査する研究である。

参考文献

[1] 大塚弘記. **GitHub 実践入門 Pull Request** による開発の変革. p. 199-253. 技術評論社, 2014.

[2] 新野淳一. はてなブログチームの開発フローと **GitHub** (前編). **Publickey**.

http://www.publickey1.jp/blog/14/githubgithub_kaigi_2014.html (参照 2014-09-19)