

初学者を対象とした GitHub flow 支援ボット

西尾 泰介¹ 柳川 龍太郎¹ 玉田 春昭¹

概要：一般的に、初学者にはバージョン管理や開発フローの理解は難しい。初学者の実感として必要性が感じられないためである。そこで本稿では、バージョン管理を利用した開発フローの1つである GitHub flow に着目し、初学者に GitHub flow を定着させること目的とした支援ボットを提案する。

The supporting bot of GitHub flow for the novice developers

1. はじめに

GitHub などを利用した開発が一般化している中、これらの教育が重要である。一方で、バージョン管理や標準的な開発フローの理解は初学者には容易ではない。必要性が不明である上に、初学者の行う開発フローに比べ、圧倒的に複雑であるためである。つまり、初学者にとっては標準的な開発フローを実施するモチベーションがないと言える。

しかし前述の通り、世の中の開発のほとんどがバージョン管理を用いており、その多くは何らかの開発フローを採用している。そのため、学習の早い段階で開発フローの利用にモチベーションを持たせる必要がある。本稿では、バージョン管理、開発フローの利用のモチベーションを上げるため、開発フローの支援ボットについて提案する。

2. 準備

2.1 本研究の対象とする学習者の技術レベル

本稿で想定とする初学者は、基礎レベルのプログラミングは問題なく作成できるものとする。そして、ソフトウェア開発に関する知識、すなわち、工程管理やバージョン管理の存在を知っているものとする。

2.2 開発フロー

初学者の行う開発フローは編集、コンパイル、実行を繰り返すものとする。もちろん、中にはテストの自動化やビルドツールを用いての管理などもあるが、バージョン管理は実施していないものとする。

一方、実際の開発で利用されている開発フローは Git flow^{*1}, GitHub flow^{*2}, GitLab flow^{*3} がよく知られている。本稿ではこの中で一番シンプルである開発フロー、GitHub flow (GHF) を対象にする。

次に、GHF の流れを示す。なお、各行頭の  はローカルリポジトリに対する操作を表しており、 はリモートリポジトリに対する操作を表している。

- (1)  master ブランチからブランチ (some_feature) を作成し、some_feature ブランチに移動する。
- (2)  開発者が機能を実装する。
 - (2-1) エディタでソースファイルを編集する。
 - (2-2) 変更をコミットする。
- (3)   変更をリモートリポジトリに push する。
- (4)  プルリクエスト (PR; Pull Request) を発行する。
- (5)  レビューアがレビューする。
 - (5-1) デプロイする。
 - (5-2) 動作確認する。
- (6)  some_feature ブランチを master にマージする。
- (7)  master ブランチをローカルリポジトリに pull する。

3. 提案手法

3.1 モチベーション

第 2.2 節で述べたように、開発フローの中で一番平易な GHF であっても、初学者が実施することは難しい。リモートリポジトリやローカルリポジトリ、ブランチについての理解が求められ、開発者の役割も意識する必要があるため

¹ 京都産業大学
Motoyama, Kamigamo, Kita-ku, Kyoto 603-8555, Japan

^{*1} <https://nvie.com/posts/a-successful-git-branching-model/>

^{*2} <http://scottchacon.com/2011/08/31/github-flow.html>

^{*3} <https://about.gitlab.com/2014/09/29/gitlab-flow/>

である。そこで、GHF を支援するボットを考える。一般的に、ボットはあるプラットフォームにインストールされ、そのプラットフォーム上での操作に対して何らかの反応を見せるものである。例えば、チャットボットであれば、チャットシステムというプラットフォームに常駐し、チャット上の発言に反応して、何らかの動作を行うようになっている。

しかし、GHF は前述の通り、単一のプラットフォームで収まるものではない。操作する対象によって、作業のプラットフォームが変わるためである。特に中心となるプラットフォームは、Git の操作、IDE (エディタ) での編集、そして、GitHub をはじめとしたソーシャルコーディングプラットフォーム (SCP; Social Coding Platform) 上での PR に対する作業である。本稿では、これらの各プラットフォームで GHF を支援するボットを配し、それぞれが協調して GHF を支援することを考える。

3.2 支援方法

初学者を対象に GHF を支援すると言っても、目的によって様々な支援の方法がある。人の判断を支援する方法 [1] などをボットが行うことでも GHF を支援できる。

本稿では支援の目的を、初学者の GHF の理解と定める。すなわち、第 2.2 節に示した流れを実施できるようにすることが目的である。そのため、適切なタイミングで、(1) 次の作業の通知や、(2) その作業を行わせることの動機付けを与えることで支援を行う。

(1) 次の作業の通知では、次にどのプラットフォームで何をしなければならないのかをユーザに通知する。通知は、ユーザが作業中のプラットフォームや、OS の通知機能を使うことが考えられる。

(2) 作業の動機付けは、各作業を初学者に定着させるための一つの手段として導入する。そもそも、初学者は GHF の各作業の必要性に関する理解が不十分である。その理解は、学習や教育だけで得られるものではなく、実践の反復が必要であろう。そのため、その作業を行う上で、開発本来とは別の利益のユーザへの提供で動機を与えるものとする。言い換えれば、GHF の実施にゲーミフィケーションを採用し、ポイント制やトロフィー制度を導入する [2]。

3.3 連携方法

提案手法はプラットフォームを横断して、ユーザに通知を行う必要がある。あるときは、Git の操作に反応し、別のときには PR の発行に応答する必要がある。そのために、各プラットフォームにインターフェースとしてのボットを設置する必要がある。例えば、リモートリポジトリに対する操作であれば、GitHub やブラウザなど、Git の操作であれば、ターミナルや OS などにボットを設置する。

提案手法では、インターフェースとしてのボットが各プ

ラットフォームに設置されることになる。各タイミングでの定型的な反応であれば、連携の必要性は低い。しかし、初学者に対する支援を考えたとき、将来的には緩やかにフェードアウトさせることが望ましい。そのため、ユーザの GHF に対する理解度を把握した上で、支援内容を決定することが望ましい。

そのためには、ユーザの理解度をどこかで計測する必要があり、その結果を各ボットが活用することが必要である。また、連携するためには、各ボットを通じて、提案手法の提供するサービスにログイン、もしくはユーザ情報の通知を許可する必要がある、ユーザ獲得の障害になりうる。

4. まとめと議論したい項目

4.1 GHF 支援ボットについての議論

動機付けに有効なものは何か ポイント制、トロフィー制以外には何かがあるか。ポイント制は将来的には仮想通貨と連携することも考えられる。

支援内容は適切か 初学者に適切な支援とは何か。

協調動作は必要か 仮に協調動作しない場合、どのような支援ができるか。

4.2 協調ボットについての議論

どこまで行うのか 協調ボットの汎用性を高めていくと、サービス連携の話となり、IFTTT*4 に行き着く。それで良いのか、他に議論すべき項目があるのか。

協調させるためのユーザ情報をどのように持つか システムを超えて協調させるためには、各々のシステムでユーザを判別する必要がある。一般的にはログインや、ユーザを一意的トークン (ID) で識別する。ユーザに負担がかからず、心理的障壁も高くない方法は何か。

4.3 まとめ

本稿では、GitHub flow を支援するボットについて述べた。様々なプラットフォームにボットを配置し、それぞれが連携して、一つの目的である、初学者の GitHub flow の理解促進を実施する。今後は、実際にこれらのボットを開発し、提案手法の効果を確認する。

謝辞 本研究の一部は JSPS 科研費 17K00196, 17K00500, 17H00731 の助成を受けた。

参考文献

- [1] Yu, Y., Wang, H., Yin, G. and Ling, C. X.: Reviewer Recommender of Pull-Requests in GitHub, *IEEE Inter. Conf. on Software Maintenance and Evolution* (2014).
- [2] Hamari, J., Koivisto, J. and Sarsa, H.: Does Gamification Work? —A Literature Review of Empirical Studies on Gamification, *Proc. 47th Hawaii International Conference on System Sciences*, pp. 3025–3034 (2014).

*4 <https://ifttt.com>