

Pull Request 駆動型の開発を支援するツールの検討

齋藤 雄輔[†] 藤原 賢二[†] 井垣 宏^{††} 吉田 則裕^{†††} 飯田 元[†]

[†] 奈良先端科学技術大学院大学 情報科学研究科 〒630-0192 奈良県生駒市高山町 8916-5

^{††} 大阪大学 大学院情報科学研究科 〒568-0871 大阪府吹田市山田丘 1-5

^{†††} 名古屋大学 大学院情報科学研究科 〒464-8601 名古屋市千種区不老町

E-mail: [†]{saito.yusuke.sl9,kenji-f}@is.naist.jp, ^{††}igaki@ist.osaka-u.ac.jp, ^{†††}yoshida@ertl.jp,
^{††††}iida@itc.naist.jp

あらまし 分散型版管理システムとそのホスティングサービスを用いたソフトウェア開発手法として、プルリクエストを基点とした制御を行う開発プロセス (Pull Request 駆動型の開発) が提案されている。Pull Request 駆動型の開発では、開発タスクごとに版管理システム上でブランチを作成し、実装を開始する際に対応するブランチの統合をプロジェクト管理者へ要求する。機能の実装に際して開発者は、各状況下において適切に版管理システムを操作する必要がある。しかし、これらの作業は煩雑であり、さらに版管理システムに対する深い知識を要求する。そこで本研究では、Pull Request 駆動型の開発に適した開発者支援ツールを提案する。

キーワード プルリクエスト, 版管理システム, 可視化, 開発プロセス

Discussion of a Tool for Supporting Pull Request Driven Software Development

Yusuke SAITO[†], Kenji FUJIWARA[†], Hiroshi IGAKI^{††}, Norihiro YOSHIDA^{†††}, and Hajimu IIDA[†]

[†] Graduate School of Information Science, Nara Institute of Science and Technology
Takayama-cho 8916-5, Ikoma-shi, Nara, 630-0192 Japan

^{††} Graduate School of Information Science and Technology, Osaka University
Yamadaoka 1-5, Suita-shi, Osaka, 568-0871 Japan

^{†††} Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464-8601, Japan

E-mail: [†]{saito.yusuke.sl9,kenji-f}@is.naist.jp, ^{††}igaki@ist.osaka-u.ac.jp, ^{†††}yoshida@ertl.jp,
^{††††}iida@itc.naist.jp

Abstract The Pull Request Driven Development (PRDD) is one of the modern methodology for software development which strongly collaborates with distributed version control systems (DVCSs) and its hosting services. During PRDD, a developer creates a branch for each feature and sends a pull request to the project manager when she/he starts implementing a feature. While implementing, the developer needs to operate DVCS correctly for every situation. However, these operations are complicated and require the further knowledge of the DVCS. In this paper, we discuss the requirements of a tool for supporting PRDD and introduce a tool under development which satisfies the requirements.

Key words Pull Request, Version Control System, Visualization, Development Process

1. はじめに

近年, GitHub に代表される版管理システムのホスティングサービスを用いたソフトウェア開発が広く行われている。GitHub は企業やオープンソースコミュニティを問わず幅広く

利用されており, 2014 年 12 月までに 1800 万を超えるプロジェクトが公開されている^(注1)。

版管理システムとそのホスティングサービスを用いたソフト

(注1) : <https://github.com/features>

ウェア開発として、プルリクエスト駆動型の開発 (PRDD) がある。プルリクエストとは変更内容の含まれるブランチを統合する旨の要求を作成する、ホスティングサービスの機能である。PRDD ではプルリクエストを基点にしたプロセス管理を行うが、そのためには開発者が版管理システムを適切に操作する必要があり、版管理システムの知識と技術が要求される。しかし、代表的な版管理システムである Git には複雑かつ操作が煩雑という問題があると指摘されている [1]。この問題に対し、版管理システムの支援ツールが開発されている [1] [2] が、それらの支援ツールでは PRDD における煩わしさを解消することはできない。

そこで本稿では、PRDD に適した開発者支援ツールを提案する。提案するツールでは、PRDD において不必要な版管理システムの操作を制限し、版管理システムとホスティングサービスを統合することによる開発効率の向上を期待する。

本稿では以降、2 章において版管理システムと版管理システムを用いた開発について述べ、3 章で版管理システムとそのホスティングサービスを用いた開発形態の 1 つである PRDD について述べる。4 章では PRDD の支援ツールに求められる要件を検討し、5 章で要件を満たす機能を備えた開発中のツールを紹介する。6 章では関連研究および技術について述べ、7 章で本稿のまとめと今後の課題を述べる。

2. 背景

本章では、本稿で扱う分散型版管理システムとそのホスティングサービスの機能の 1 つであるプルリクエストについて説明する。

2.1 版管理システム

版管理システムとはソースコードの変更履歴を管理するためのシステムである。版管理システムではソースコードに対する変更をコミットと呼ばれる単位で表現し、リポジトリと呼ばれるデータベースに保持する。版管理システムは Subversion や CVS に代表される集中型版管理システムと、Git や Mercurial に代表される分散型版管理システムに分類される。集中型版管理システムは 1 つのプロジェクトに対して 1 つのリポジトリがある一方で、分散型版管理システムは 1 つのプロジェクトに対して複数のリポジトリがある。近年においては、集中型版管理システムから分散型版管理システムへの移行が広く行われており、分散型版管理システムが主流となっている [3]。

分散型版管理システムではサーバー上にリモートリポジトリを持ち、開発者はそれぞれの端末上にローカルリポジトリを持つ。分散型版管理システムの利点として、リモートリポジトリに影響を与えずにローカルリポジトリへのコミットが可能のため、ローカルリポジトリでコミットの改変が容易となる。分散型版管理システムを用いた開発手順を以下に示す。

- (1) ローカルリポジトリを最新のリモートリポジトリに更新する。(プル)
- (2) ローカルリポジトリで開発を行い、ソースコードの変更をローカルリポジトリに書き込む。(コミット)

- (3) ローカルリポジトリの変更内容をリモートリポジトリに反映する。(プッシュ)

版管理システムにはコミットの履歴を分岐させる、ブランチと呼ばれる機能がある。ブランチによって枝分かれしたコミットは互いに影響を受けないため、機能の実装やバグ修正の開発を並行して進めることが容易となる。作成されたブランチはマージという操作によって、別のブランチに統合され、変更内容がまとめられる。

2.2 版管理システムのホスティングサービス

集中型および分散型版管理システムを用いたソフトウェア開発においては、代表となるリポジトリをプロジェクトが 1 つないし複数所持する。そのリポジトリの管理および運用を支援することを目的としたホスティングサービスがある。ホスティングサービスの例として、Git のホスティングサービスの GitHub^(注2) や Bitbucket^(注3) が挙げられる。ホスティングサービスにはコミット履歴やブランチ一覧の可視化だけでなく、版管理システムを用いた開発を支援する機能がある。またホスティングサービスの提供する代表的な機能としてプルリクエストがある。

プルリクエストとは、あるブランチを指定したブランチにマージする旨をプロジェクト管理者に要求するための機能である。プルリクエストの例を図 1 に示す。個々のプルリクエストには、対応するブランチのコミット一覧とコメント欄が表示される。開発者とプロジェクト管理者は、必要に応じてコメント欄を用いて実装内容について議論を行う。プロジェクト管理者が作成されたプルリクエストをマージしても良いと判断した後



図 1 プルリクエストの例 (GitHub)

Fig. 1 An example of GitHub Pull Request

(注2) : <https://github.com>

(注3) : <https://bitbucket.org>

にマージが実施される。このように、プルリクエストを作成することでブランチのマージを明確化し、開発者とプロジェクト管理者による議論の機会を設けることができる。図1は、プロジェクト管理者によりコードレビューが行われ、マージされた例である。

2.3 分散型版管理システムを用いたソフトウェア開発

これまでに、開発プロセスの管理にブランチを活用するブランチモデルが提唱されている。ブランチモデルではブランチの作成および運用のルールを設けることで、ブランチの管理の容易さや不必要なブランチのマージを防ぐといった利点がある。例として git-flow [4] では、リリース用ブランチ (master)、リリース準備用ブランチ (release)、開発用ブランチ (develop)、機能の実装やバグ修正用のブランチ (feature) にそれぞれのブランチが分類される。git-flow の運用手順を以下に示す。

- (1) master ブランチから develop ブランチを作成する。
- (2) develop ブランチから実装する機能やバグ修正ごとに feature ブランチを作成し、開発が完了した際に develop ブランチにマージする。
- (3) develop ブランチの内容をリリースする際は、develop ブランチから release ブランチを作成し、リリースの準備に必要なコミットを行う。
- (4) release ブランチを master ブランチにマージする。

上記のようにブランチごとに役割を分担させ、各ブランチの状況を参照することで、開発状況の把握が容易となる。

3. プルリクエスト駆動型開発

本章ではプルリクエスト駆動型開発 (PRDD) とその問題点について説明する。

3.1 概要

PRDD は git-flow のブランチモデルを基に提案された、分散型版管理システムとそのホスティングサービス上のプルリクエストを用いてソフトウェア開発を管理する手法である [5]。本稿では Git(分散型版管理システム) と GitHub(ホスティングサー

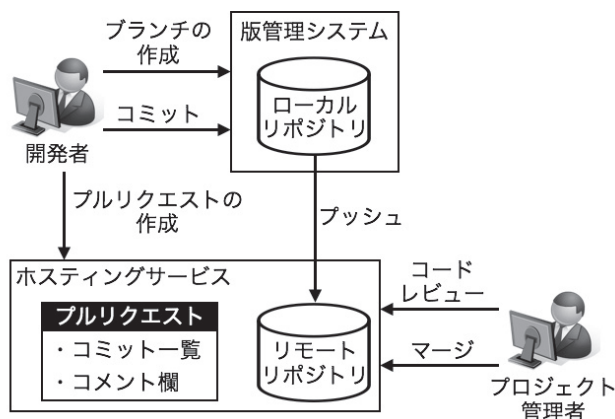


図2 PRDDの開発手順

Fig. 2 An overview of the PRDD process

ビス)を想定する。

PRDDの開発手順を図2に示す。初めに、開発者は機能の実装やバグ修正などの開発タスクに対してブランチを作成する。開発タスクの作業を行うブランチをトピックブランチと呼ぶ。このトピックブランチを作成した際に、ホスティングサービスでトピックブランチに対応するプルリクエストを作成する。開発者はトピックブランチで開発を行い、随時コミットをプッシュする。プロジェクト管理者は、プルリクエストに表示される差分の内容を基にコードレビューを実施し、不具合が存在した場合にコメントで開発者に修正を要求する。そして、開発者はソースコードを修正し、再度プッシュする。管理者が変更内容をリリース用のブランチにマージしても良いと判断した際に、トピックブランチをマージする。このようにPRDDではgit-flowのmasterブランチとfeatureブランチをリリース用のブランチとトピックブランチとして管理する。

3.2 PRDDにおける問題点

開発者はPRDDにおいて、開発中の様々な状況に適した操作を版管理システムに指示する必要がある。しかし分散型版管理システムは複雑かつ操作が煩雑であり、知識と技術が要求されると指摘されている [1] [3]。具体的に開発者は、以下に示す点に注意を払う必要がある。

- プロジェクト管理者や他の開発者が理解し易いコミット履歴を作成する。そのためには、追加の修正や変更漏れのコミットが存在する際は複数のコミットを1つに圧縮し、同一コミットに複数の意味を持つ変更が存在した際はコミットを複数に分割する必要がある。
- リリース用ブランチの開発状況を考慮して開発する。リリース用ブランチに更新があった際、開発者は更新内容がトピックブランチの実装内容に影響を及ぼすかを確認する。影響のあるコミットが存在した場合は、トピックブランチにおける変更の整合性を保つようにトピックブランチの分岐元をリリース用ブランチの最新コミットに変更し、テストする。

4. ツールに求められる要件

本章では、3.2節の問題を踏まえて、PRDDにおける開発者を支援するツールに求められる要件を版管理システムの操作と開発者に提示する情報に分けて検討する。

4.1 版管理システムの操作に対する支援

PRDDにおいて用いる版管理システムの操作では、複数の操作を組み合わせる必要があり、版管理システムに対する知識と技術が要求される。またそのような複雑な操作時の誤操作は、手戻りが発生するため復元に時間が費やされてしまう。例として、圧縮や分割による操作は新たなコミットが作成されるため、操作前のコミットに戻ることが困難となる。そのためPRDDにおいて使用頻度の高い複合的な操作は、すべて単一の操作に置き換えることが望ましく、各操作における前後のコミット履歴が容易に復元できる必要がある。

要件 I-A 使用頻度の高い一連の版管理操作を単一のツール操作で実現する。

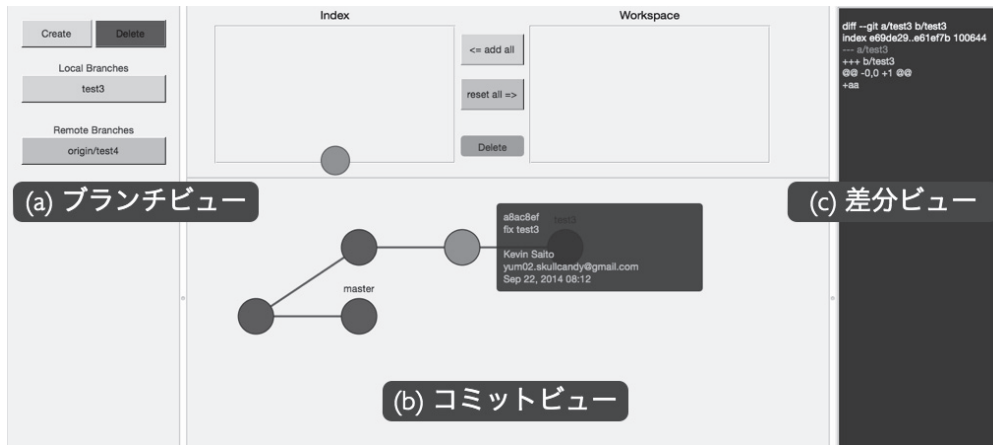


図 3 提案ツールの画面

Fig. 3 A view of implementing tool

要件 I-B 版管理システムの操作ごとにコミット履歴を記録し、操作前後の状態に復元する。

PRDD では開発者に版管理システムのすべての操作を提供する必要はないため、ツールにおいて一部の操作を制限することにより、版管理システムの煩雑さを緩和させることができると考えられる。

要件 I-C PRDD の版管理システムの操作において不要な版管理システムの操作を制限する。

PRDD では、個々のトピックブランチに対してプルリクエストを作成しなければいけない。そのため、ブランチの作成と同時にプルリクエストを作成することで、プルリクエストの作成漏れを防ぐことができると考えられる。

要件 I-D ブランチが作成された際に、自動的にプルリクエストを作成する。

4.2 開発者に提示する情報

PRDD において、開発者はトピックブランチだけでなくリリース用ブランチの開発状況を考慮しなければいけない。その際には、トピックブランチが作成されたコミット以降に更新されたりリリース用ブランチのコミットを閲覧する必要がある。そのため開発者にトピックブランチのコミットだけでなく、ブランチ作成後に更新されたりリリース用ブランチのコミットを同時に提示することが求められる。

要件 II-A トピックブランチのコミット履歴だけでなく、トピックブランチが分岐されたコミット以降のリリース用ブランチのコミット履歴を提示する。

開発者はプルリクエストへのコメントを基にソースコードの修正をする。そのため、開発者はブランチを切り替える際にブランチに対応するプルリクエストの検索およびコメントの閲覧をウェブブラウザ上で行う。ブランチの切り替え時にブランチに対応したプルリクエストの情報およびコメントを開発者に提示することで、ホスティングサービスでの作業を省くことができると考える。またコメントが追加されたことに気付かず開発を進めてしまい、無意味な作業を行ってしまう可能性がある。

効率の良い開発を行うために、追加されたコメントを開発者に通知する必要がある。

要件 II-B ブランチに対応したプルリクエストの情報およびコメントを提示する。

要件 II-C トピックブランチのプルリクエストにコメントが追加された際に開発者に通知する。

5. 開発中のツール

本章では、4. 章の要件を基に現在開発中の PRDD の支援ツールについて説明する。開発中のツールは Git と GitHub を対象とした PRDD の開発者支援ツールで、デスクトップアプリケーションとして実装されている。開発者は開発中のツールを用いて、Git のローカルリポジトリの操作と GitHub 上の作業を行う。現在、要件 I-A, I-C, II-A に対応しており、要件 I-B, I-D, II-B, II-C は未対応である。

開発中のツールのスクリーンショットを図 3 に示す。開発中のツールはブランチの操作を行うブランチビュー (a)、コミットの作成および履歴を表示するコミットビュー (b)、コミットや変更のあるソースコードの差分情報を表示する差分ビュー (c) に分かれる。

ブランチビューでは、ブランチの作成および切り替えを行う。Git はブランチ作成において分岐点を指定しなければ通常、現在のコミットが分岐点となるブランチを作成する。一方 PRDD においては、リリース用のブランチからのみトピックブランチを作成する。開発中のツールでは要件 I-C に従い、リリース用のブランチ以外からの分岐を制限し、リリース用ブランチのコミットが分岐点となるトピックブランチを作成する。複数の開発タスクを並行して行う際は、ブランチの切り替えが頻繁に行われるが、コミットされていない編集内容はブランチの切り替えに影響を及ぼすため、編集内容を一時的に記録しなければいけない。開発中のツールでは、ブランチ一覧のボタンから容易にブランチを切り替えることができ、Rosso らの Gitless [1] と同様にコミットされていない編集内容をブランチと対応させて

記録する。記録された編集内容は、以前のブランチに切り替えた際に復元されるため、ブランチの切り替えが容易となる。このように、開発中のツールでは要件 I-A に従い版管理システムの操作を簡略化している。

コミットビューのコミット履歴の表示はトピックブランチとリリース用のブランチの各コミットを円で表す。要件 II-A に従い、トピックブランチが分岐されたコミット以降におけるリリース用ブランチのコミットを表示する。コミットをドラッグし、別のコミットの上にマウスを被せる、コミットとコミットの間で置くといった操作を行うことで、コミットの並び替えや圧縮といったコミットの実行を行うことができる。これらは要件 I-A に従い、使用頻度が高く、複雑な操作を要するコミットの並び替えや圧縮を単一の操作に置き換えている。コミットビュー上部は、Git でインデックスとワークスペースと呼ばれる機能に当たる。変更されたファイルをドラッグすることで、右のワークスペースと左のインデックスを行き来することが可能である。インデックスに追加されたファイルは左の円をクリックすることにより、コミットする。また変更されたファイルや各コミットをクリックすることで、変更内容が差分ビューに表示される。

6. 関連研究および技術

6.1 ソフトウェア開発手法に関する技術

開発タスクをバグ管理システムのチケットに割り当てて開発を行うソフトウェア開発手法として、チケット駆動開発 (TiDD) がある [6]。TiDD では開発タスクごとにチケットを発行し、実装が完了した際にチケットをクローズする。開発タスクの実装とコードレビューのチケットはそれぞれ別に発行され、コードレビューを行う際はチケットに対応する編集履歴を探す必要がある。本稿で扱う PRDD では、開発タスクごとにプルリクエストを作成し、コードレビューおよび議論を基に改良されたコミットがマージされた際にプルリクエストをクローズする。プルリクエストを作成することによりコードレビューの機会が設けられ、編集履歴を容易に確認できることからコードレビューの効率化を図ることができる。また、リリース用のブランチにマージされていないプルリクエストは未完了のタスクであり、マージが完了したプルリクエストは完了したタスクとなるため、開発タスクを TiDD と同様に管理することができる。

6.2 版管理システムの支援に関する技術

版管理システムの支援ツールとして Atlassian 社の SourceTree が挙げられる。SourceTree は Git や Mercurial 用の可視化ツールである [2]。このツールは、コミット履歴をグラフ表示し、版管理システムの操作を各ボタンで操作することができる。また git-flow などのブランチモデルの開発支援も提供されている。それに対し、本提案ツールは PRDD に不必要な版管理システムの操作を制限することによる利用性の向上、ホスティングサービスの統合といった点で異なる。

6.3 コミット履歴の整理に関する研究

PRDD において、プロジェクト管理者やその他開発者が理解し易いようにコミット履歴を整理する必要がある、整理する

ことによりコミットの理解性や利便性の向上に役立つ。このコミット履歴の整理に関する研究がいくつか行われている。

例えば、松田らの研究ではコミットポリシーに従ったコミットの自動生成を行っている [7]。コミットポリシーとは、コミット 1 つの大きさや含まれる修正内容を記述したものである。開発者の編集操作履歴を基に階層の木構造を作成し、指定されたコミットポリシーの粒度に応じてコミットの作成を行う。

また切貫らの研究では、過去のコミットの情報をもつれた修正のコミットの検出および分割方法の提示をしている [8]。もつれた修正とは、複数のタスクを含む修正のことである。もつれた修正の検出は開発者がコミットを行う際に行われ、検出された際に分割方法を提示する。この研究の目的は開発者に不利益を生む、もつれた修正のコミットの低減である。

これらの研究は、整理されたコミットの作成および提示であるのに対し、本提案ツールはコミット履歴の整理の操作支援を行う。

7. まとめと今後の課題

本稿では版管理システムとそのホスティングサービスを用いた PRDD の開発者支援ツールに求められる要件の検討を行い、その要件に基づいて開発中の支援ツールを紹介した。開発中のツールはデスクトップアプリケーションとして実装されており、Git と GitHub を対象としている。

今後、以下の未実装の要件をツールに実装し、ツールの有効性を評価するためにケーススタディを行う予定である。

- ブランチの作成と同時にプルリクエストを作成する。
- 現在のブランチに対応したプルリクエストを表示する。
- プルリクエストにコメントが追加された際に通知を表示する。
- 版管理システムの操作前後の状態に復元する。
- 使用頻度の高い版管理システムの操作の内、未実装のものを単一のツール操作で実現する。

謝辞 本研究の一部は、JSPS 科研費 24700030, 26540182, 26730036 の助成を受けたものである。

文 献

- [1] S.P.D. Rosso and D. Jackson, “What’s Wrong with Git?: A Conceptual Design Analysis,” Onward! 2013 Proceedings of the 2013 ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming & Software, pp.37–52, 2013.
- [2] Atlassian, “Sourcetree,” <https://www.atlassian.com/software/sourcetree>.
- [3] K. Muşlu, C. Bird, N. Nagappan, and J. Czerwonka, “Transition from Centralized to Decentralized Version Control Systems: A Case Study on Reasons, Barriers, and Outcomes,” ICSE 2014 Proceedings of the 36th International Conference on Software Engineering, pp.334–344, 2014.
- [4] V. Driessen, “A successful Git branching model,” <http://nvie.com/posts/a-successful-git-branching-model>.
- [5] G. Gousios, M. Pinzger, and A. vanDeursen, “An Exploratory Study of the Pull-based Software Development Model,” ICSE 2014 Proceedings of the 36th International Conference on Software Engineering, pp.345–355, 2014.
- [6] 小川明彦, 阪井誠, Redmine によるタスクマネジメント実践技法, 翔泳社, 2010.

- [7] 松田淳平, 林晋平, 佐伯元司, “編集操作履歴の階層的なグループ化を用いたポリシー準拠のコミットの構成支援,” ソフトウェアエンジニアリングシンポジウム 2014, pp.76-84, 2014.
- [8] 切貫弘之, 堀田圭佑, 肥後芳樹, 楠本真二, “修正履歴情報を利用したコミット分割支援手法の提案,” ソフトウェアエンジニアリングシンポジウム 2014, pp.85-94, 2014.