

# OSS 開発におけるパッチレビュープロセス追跡技術の提案

大坂 陽<sup>†1</sup> 伊原 彰 紀<sup>†2</sup>  
亀井 靖 高<sup>†1</sup> 鵜林 尚 靖<sup>†1</sup>

オープンソースソフトウェア (OSS) 開発ではバグ修正, 機能拡張を行ったコード (パッチ) を作成後のレビュー依頼, 及び, パッチ作成者へのフィードバックをメーリングリストで行っている. パッチが幾度と更新されプロダクトに反映されるプロセスを分析することによりパッチの進化過程を明らかにすることが可能となる. 本論文では, メーリングリストに投稿されたパッチと Git のコミットログのリンク付けを行った.

## Tracing Patch Review Process in Open Source Software Project

ATARU OSAKA,<sup>†1</sup> AKINORI IHARA,<sup>†2</sup> YASUTAKA KAMEI<sup>†1</sup>  
and NAOYASU UBAYASHI<sup>†1</sup>

In Open Source Software (OSS) projects, developers and OSS committees communicate to create patches about bug fixing and new functions on their mailing list. Analyzing the communication helps understanding the evolution of the patch review process (until the patch is commit to source code repository). In this paper, we investigated how many patches on the mailing list link to source codes committed to Git.

### 1. はじめに

オープンソースソフトウェア (OSS) コミュニティは, 開発がバグ修正, 機能拡張のために変更したコード (パッチ) をメーリングリストで受け付けるため, 多くの参加者がパッチの内容を確認することができる. しかしながら, すべてのパッチがプロダクトに取り込まれるとは限らない<sup>1)</sup>. レビューの途中で開発をあきらめるパッチや, レビューすら受けられないパッチが存在する<sup>2)3)</sup>. パッチが幾度と更新されプロダクトに反映されるプロセスをたどることで, パッチのレビュープロセスにおけるコミュニケーションが, 継続的にパッチ提出する開発者の変化, パッチの進化を理解することができる.

本研究では, パッチレビュープロセスを追跡する技術の確立を目的とし, どの程度のパッチがプロダクトに反映された変更 (コミットログ) とリンク付けできるのか明らかにする. また, パッチの進化を分析するためには, メールやパッチを 1 件ずつ解析する必要がある. そのための処理には多くの時間を要するので,

処理の高速化についても検討する.

### 2. パッチレビュープロセス

図 1 は, OSS 開発におけるパッチ投稿から, OSS プロダクトに取り込まれるまでの流れを説明する.

- (1) OSS 開発者は不具合の報告を受けると, 当該不具合を修正するためにパッチを作成する.
- (2) 作成したパッチをメーリングリストでコミュニティに投稿する.
- (3) 他の開発者のレビューを受け, フィードバックを得る.
- (4) 投稿されたパッチが適切でないと判断された場合, (1) から (3) を繰り返す.
- (5) 投稿されたパッチが適切であると判断された場合, そのパッチはコミット権限を持つ開発者 (コミッター) によってプロダクトに取り込まれる.

OSS 開発では, パッチ投稿にメーリングリストが使用されることが多い. それぞれの OSS コミュニティでメールの書き方のドキュメントが用意されており, そのドキュメントに従い, 且つ, パッチの内容が適切であると判断された場合に, その OSS 開発のプロダクトに取り込まれる.

<sup>†1</sup> 九州大学

Kyushu University

<sup>†2</sup> 奈良先端科学技術大学院大学

Nara Institute of Science and Technology

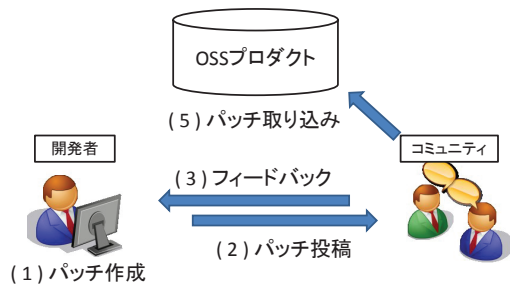


図 1 OSS 開発におけるパッチが取り込まれるまでの流れ

プロダクト	メール数	スレッド数
Git	29,303	3,465

表 2 パッチとコミットログのリンク付けできた割合 (%)

	メール	スレッド
リンク付けできた割合	7.49	26.70

### 3. パッチの種類

本研究では、どの程度のパッチがプロダクトのコミットログとリンク付けできるのか明らかにする。パッチの種類については様々な研究がなされており、次項に述べる様なパッチはリンク付けができるパッチとできないパッチで特徴が現れると考える。

**変更箇所が少ないパッチ：** サイズが小さく、変更が少ない様なパッチは、レビューアに理解されやすいので、プロダクトに反映されやすい傾向にある<sup>1)</sup>。この様なパッチはコミットログとのリンク付けができやすいと考える。

**レビューを受けないパッチ：** メーリングリストに投稿したパッチが見過ごされて、開発者からレビューを受けないことがある<sup>2)</sup>。この様なパッチはプロダクトに反映されておらず、コミットログとのリンク付けができにくいと考える。

### 4. データセット

本研究では、GitHub で開発されている Git プロダクトを対象とした。本研究で用いたデータセット<sup>\*1</sup>の統計を表 1 に示す。約 1 年間のメーリングリストのデータを使用した。スレッドとは、あるパッチ投稿に対する一連のメールのやりとりのことを指す。

### 5. 結果

リンク付けできたパッチの割合を表 2 に示す。本研究では、パッチ投稿の日付と送信者の名前が、コミットログの Author の日付と名前と一致した場合がリンク付けできたとした。リンク付けできたメールの割合

\*1 <http://www.spinics.net/lists/git/threads.html>

変更	レビュー	リンク数	リンク率 (%)
多	有	1,577	71.88
少	有	520	23.70
多	無	63	2.87
少	無	34	1.55

がスレッドの割合に比べて低いことは、議論やフィードバックが行われたメールもメールの総数に入っているためである。

メールの内訳を表 3 に示す。レビューを受けないパッチでも約 4%がプロダクトに反映されていたことがわかった。これらのメールについては、今後の研究で明らかにしていきたい。

### 6. 今後の展望

パッチの進化を分析するためには、メールやパッチを 1 件ずつ解析する必要があり、今後複数プロダクトを対象とすると、これらの処理にはより多くの時間を要する。著者らはこれまで Hadoop を使用したりポジトリマイニングの高速化について研究してきた<sup>4)</sup>。この Hadoop の運用経験を活かして、パッチ解析処理の高速化を図り、研究を行っていききたい。

本研究により次のようなことが理解できると考える。

- レビューによる違いが、開発者のモチベーションにどのように影響するかわかる。
- パッチのレビュープロセス全体を見ることで、パッチの進化を追うことができる。

**謝辞：** 本研究の一部は、JST CREST「ポストベタスケール時代のスーパーコンピューティング向けソフトウェア開発環境」による助成を受けた。

### 参考文献

- 1) Weißgerber, P., Neu, D. and Diehl, S.: Small Patches Get in!, *Proceedings of the International Working Conference on Mining Software Repositories*, pp.67–76 (2008).
- 2) Nurolahzade, M., Nasehi, S. M., Khandkar, S. H. and Rawal, S.: The Role of Patch Review in Software Evolution: An Analysis of the Mozilla Firefox, *Proceedings of the Joint International and Annual ERCIM Workshops on Principles of Software Evolution and Software Evolution Workshops*, pp.9–18 (2009).
- 3) 藤田将司, 伊原彰紀, 大平雅雄, 松本健一: OSS 開発におけるパッチレビュープロセスの効率化に向けたコミッターの分類, 平成 22 年度情報処理学会関西支部支部大会 講演論文集, Vol.2010 (2010).
- 4) 大坂 陽, 山下一寛, 亀井靖高, 鶴林尚靖: リポジトリマイニングに対する Hadoop の導入に向けた性能評価, ソフトウェアエンジニアリングシンポジウム 2013(SES 2013) (2013).