

# OSSのレビューデータ分析におけるバグ見逃しコミットの取り扱いについて

戸 田 航 史<sup>†1</sup>

本稿では OSS においてレビューでバグを見逃したコミットの分析上での取り扱いに注目した。すなわち、バグを見逃したコミットについて、バグ有り/無しの 2 値として取り扱うべきか、それともコミットに含まれるバグをそのコミットでの変更行数で割ったいわばバグ密度として取り扱うべきかについて調査した。調査の結果、分析における差異は小さいことが分かった。

## An investigation about analytical treatment of patch with overlooked bugs in review process of OSS

KOJI TODA<sup>†1</sup>

This paper focuses on patch without or with overlooked bugs in review process of OSS project. The patch can be treated as 0/1 in analysis, but bug density is frequently used in bug prediction research area domain. I analysed these criteria using OSS project data and a little difference was shown.

### 1. はじめに

ソフトウェア開発におけるレビューとは、設計文書やソフトウェアのソースコードを人が読み、設計の誤りやコードの記述ミス、コーディングルールの違反などの問題がないかを検査するプロセスのことである。レビューの実施により、欠陥の早期発見、修正が可能となり、欠陥のおよそ 60% を発見可能であることが報告されている<sup>1)</sup>。

実際のレビューのプロセスでは必ずレビュイーとレビュアーが存在する。レビュイーとはレビュー対象となる設計文書やソースコードを作成した開発者を指し、レビュアーとはレビューを行う開発者を指す。そして、この 2 者の関係はレビュー対象物に含まれる欠陥 (バグ) の見逃しに影響を与える可能性がある。

筆者は以前の研究で OSS でのパッチレビューにおけるレビュイーとレビュアーの関係の強さがバグの見逃しに与える影響について分析を行っている<sup>2)</sup>。ただし、当該研究ではパッチをバグ有り/無しの 2 値データとして取り扱っており、よってバグ見逃し割合を表す値としてバグを含むパッチ数/全パッチ数を用いている。しかしながら後述する通り、バグ予測分野においては、バグに対して有り/無しという 2 値データではなく「含まれるバグ数/コード行数」で算出される

バグ密度を採用する研究が増えつつある。そこで本研究では、パッチをバグの有無という 2 値で取り扱った場合とバグ密度で取り扱った場合の分析結果を比較し、その差異を調査する。分析は文献<sup>2)</sup>と同様に、パッチレビューにおける特定のパッチ開発者と特定のレビュアーの組み合わせでのレビュー回数 (関係の強さ) と、バグ見逃し割合およびバグ密度の関係を調査する。以降、OSS におけるコードレビューを対象とするため「レビュイー」は「パッチ開発者」と表記し、「レビュアー」は必ず「レビュアー」と表記する。

### 2. 分 析

本研究では Chromium Project<sup>\*1</sup> のデータを対象として、バグを含むパッチの取り扱いについて調査するため、開発者とレビュアーの関係がバグ見逃しに与える影響の分析を行った。

#### 2.1 パッチ中のバグの有無とバグ密度

コードに含まれるバグに関わる研究としてはバグモジュール予測が挙げられる。バグモジュール予測は、ソフトウェアを構成するモジュールのうち、どれにバグが含まれているかを予測することを目的とする。ただし、モジュールの規模 (コード行数) が大きくなればバグを含みやすくなるため、モジュール中のバグの有無のみを予測対象とするバグモジュール予測は、規

<sup>†1</sup> 福岡工業大学  
Fukuoka Institute of Technology

\*1 The Chromium Project <http://www.chromium.org/>

分析対象期間	2008/9/2~2012/10/31
コミット数	113,209
レビュー数	209,618
バグ票数	157,205

模の大きなモジュールをバグ有りモジュールとして検出する傾向にある。テストに要する時間は規模に比例する傾向にあり<sup>3)</sup>、この手法では実際のテストにけるバグの検出効率への貢献が小さいという問題があり、これを踏まえ、バグ密度の予測が用いられる事が多くなってきている。バグ密度予測はモジュールに含まれるバグ数をモジュールの行数で割った値として与えられ、前述の通り、テストにかかる時間は規模に比例するため、「バグ密度が高い」とは時間あたりの検出バグ数が多い、テスト効率が高い事を意味する。パッチレビューにおいてもテストと同様にレビューにかかる時間はパッチ行数に比例すると考えられる。よって本研究ではパッチに含まれるバグ数をパッチで変更される行数で割った値をバグ密度とみなし、分析対象として採用する。

## 2.2 分析環境

本稿では Chromium Project を分析対象とした。Chromium Project ではバージョン管理システムとして git と Subversion (リポジトリは同期されている) を、バグ管理システムとして Google Code を、レビュー管理システムとして Rietveld を採用している。これらのシステムから変更履歴、バグ票、レビュー履歴の各データを取得し、それらを各コミットのハッシュ値と SZZ アルゴリズムでひも付けし、分析に用いた。分析対象期間と、期間中のコミット数、パッチレビュー数、バグ票数は表 1 の通りである。

## 2.3 分析結果

図 1, 図 2 に分析結果を示す。縦軸はバグ密度およびバグの見逃し割合、横軸は特定のパッチ開発者と特定のレビュアーによる累積レビュー回数である。ただし、特定の組み合わせでのレビュー回数が増えると該当する組み合わせの数は減少する。このため、30 回以上のレビュー経験のある組み合わせだけを対象として、1 回目から 30 回目までのパッチのバグ密度、レビュー時間を分析した。

図から、バグ密度をメトリクスとして採用した場合、バグの有無よりもデータのばらつきが大きい事が分かる。現時点ではこのようなばらつきが表れた原因は不明であり、今後の課題とする。ただし、分散、標準偏差といった統計量で見た場合にはほとんど差は表れなかった (分散では 0.00, 標準偏差では 0.02 の差) こ

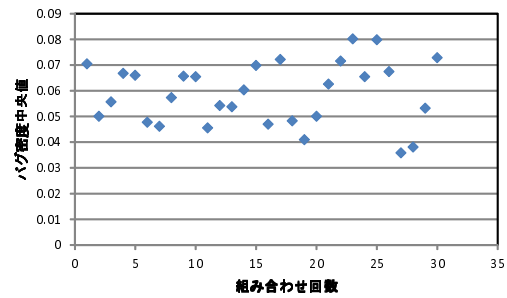


図 1 組み合わせ回数とバグ見逃し率の関係 (バグ密度)

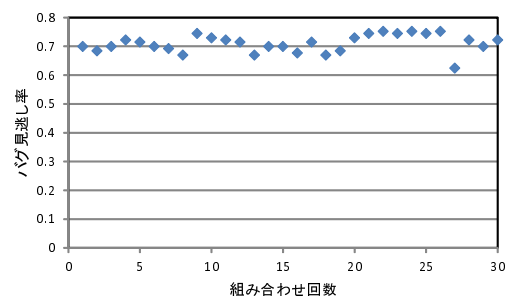


図 2 組み合わせ回数とバグ見逃し率の関係 (バグの有無)

と、相関係数で見た場合、バグ密度を採用した場合には 0.05, バグの有無を採用した場合には 0.21 と、どちらにも明確な相関は表れなかったことから、バグ密度とバグの有無という指標の違いによる差はそれほど大きくは表れないと考えられる。2.1 で述べた通り、バグ密度はレビューにかかるコストを反映できるため、パッチのバグに関わる分析を行う場合にも、バグ予測と同様にバグ密度を指標として採用すべきと考えられる。ただし、同時に今回のばらつきの違いの原因について今後

## 参考文献

- 1) Boehm, B. and Basili, V.: Software Defect Reduction Top 10 list, *Computer*, Vol.34, No.1, pp.135-137 (2001).
- 2) 戸田航史, 濱崎一樹, 亀井靖高, 吉田則裕: Chromium Project におけるレビュアーとパッチ開発者の関係がレビュー効率に与える影響の分析, ソフトウェア工学の基礎 XX FOSE 2013, 日本ソフトウェア科学会, 近代科学社 (2013).
- 3) Schneidewind, N. F. and Hoffmann, H.-M.: An experiment in software error data collection and analysis, *Software Engineering, IEEE Transactions on*, No.3, pp.276-286 (1979).