

## OSS を用いたチーム単位での生産性に影響を与える要因の検出

戸田 航史<sup>†1</sup> 亀井 靖高<sup>†2</sup>

ソフトウェア開発において、生産性は重要な指標の1つである。このため従来、開発環境や属人性要因に注目した研究が行われてきた。しかしながら現状のソフトウェア開発はチームを編成して行われることが多く、チーム単位での関係性要因を考慮すべきと考えられる。本研究では OSS を対象として分析を行い、その要因の検出を試みた。

### Team productivity analysis using OSS in software development

KOJI TODA<sup>†1</sup> and YASUTAKA KAMEI<sup>†2</sup>

Productivity is one of the most important criteria for software development. and development environment and human factor was focused in previous study. However actual software is developed by team, therefore team factor is considered for productivity. In this study, we analyze OSS to found the factor effecting to productivity.

#### 1. はじめに

ソフトウェア開発において生産性は重要な指標の1つである。これは生産性がプロジェクトチームの開発効率を表す指標であり、効率の向上は開発コストの低減に直結するためである。このことから、生産性に影響を与える要因について従来数多くの研究がなされてきた。これらの研究は生産性に影響を与える要因として、開発言語や開発技法（開発プロセス）などの開発環境要因や、各開発者の経験年数、プロジェクトマネージャの経験年数などの属人性要因を生産性に影響するものとして挙げている。

しかしながら、近年のソフトウェア開発はプロジェクトごとにチームを編成し、チーム単位で開発を行う。よって従来研究で注目している開発環境や開発者個人以外の要因、すなわち開発チームに属する開発者間の関係（チームワーク）が生産性に与える影響に注目すべきと考えられる。ここで言うチームワークとは、チームに属する開発者間のコミュニケーションの健全性や過去の共同開発経験等の要素を指す。

そこで本研究の最終目的はチームワークが生産性に与える影響を明らかにすることである。しかしながら現時点ではチームワークに影響を与える要因自体が不

明であり、当然ながら影響を評価することもできない。よって本研究ではそのための第一歩として、OSSに関わる開発者の行動の分析を行い、生産性に影響を与える要因の検出を行う。

#### 2. 検証実験

本研究では OSS プロジェクトのデータを対象として、生産性に影響を与える要因の検出を試みた。その実験概要を以下に述べる。

##### 2.1 OSS におけるパッチレビュープロセス

ここでは OSS プロジェクトにおいて開発者が機能拡張や不具合修正を行う場合のパッチの投稿と検証のプロセスについて説明する。

まず開発者はプロジェクトで提示されている機能拡張の方針や、不具合管理システム（BTS; Bug Tracking System）に投稿された不具合票を元にパッチを作成し、プロジェクトに投稿する。パッチ作成は全ての開発者が投稿する事ができる。次に投稿されたパッチに対し、その品質検証を目的としたレビューが行われる。パッチのレビューにも全ての開発者が参加できるが、パッチの内容が目的（不具合の修正や機能の改善・追加）を満たしているかどうかの最終的な判断は、パッチのプロダクトへの反映権を持つコミッターによって下される。パッチが不十分と判断されれば作成者へパッチ修正依頼が出され、十分と判断されればコミッターはパッチをプロダクトに適用（コミット）する。

本稿では、あるパッチについて、その開発者とコミッ

<sup>†1</sup> 福岡工業大学

Fukuoka Institute of Technology

<sup>†2</sup> 九州大学

Kyushu University

ターはチーム、過去のパッチレビュー回数はその2者の関係の強さ（チームワーク）、レビュー期間は生産性に対応づけられると考える。そして生産性の向上、すなわちパッチ検証期間（パッチが投稿されてからパッチがコミットされるまでの期間）に影響する要因の検出を行う。本稿では検出のためにパッチ作成者とコミッターの関係に注目する。

## 2.2 開発者、コミッターとレビュー期間の関係

パッチ検証期間の削減のために、検証期間に影響を与える要因について分析を行う。そのために、「特定の開発者が製作したパッチを特定のコミッターがレビューする場合、過去にその開発者のパッチをレビューした回数が多ければレビュー期間は短縮される」という仮説を立て、その検証を行う。

この仮説は以下の状況を想定している。OSS プロジェクトへの参加経験の無い開発者 A が不具合修正パッチを作成・提出し、そのパッチをコミッター B が検証する場合には、B はパッチを注意深く検証すると考えられる。次に、その後に A がまたパッチを作成し、同じく B がそれを検証する、というプロセスが複数回行われた場合を考える。検証回数が少ないうちはやはり B はパッチを注意深く検証するが、検証回数が増えるにつれて B は A の技術力やコードの書き方のクセを把握する事で、検証自体にかかる時間を減少させると考えられる。

## 2.3 実験環境

検証のため、Android Platform の変更履歴データ<sup>1)</sup>を用いて分析を行った。変更履歴データは 2005 年 4 月から 2012 年 10 月までに Android Platform に与えられた変更（パッチ適用）の履歴であり、各パッチについてその開発者名、投稿日、コミッター名、コミット日、パッチによって変更される行数（追加行数と削除行数）、パッチの対象ファイル数などが含まれている。

実験手順を以下に示す。

- (1) 変更履歴データからパッチ投稿者名、パッチ投稿日、コミッター名、コミット日、パッチにより変更される行数を抽出する
- (2) パッチ投稿日とコミット日からレビュー期間を算出する
- (3) レビュー期間が 720 時間以上、レビュー期間が 0 時間、パッチ投稿者とコミッターが同一人物のいずれかに該当するパッチデータを除外する
- (4) レビュー期間を変更行数（追加行数と削除行数の和）で割る（正規化レビュー期間と呼ぶ）
- (5) パッチ投稿者とコミッターの組み合わせ回数と正規化レビュー期間の関係について分析する

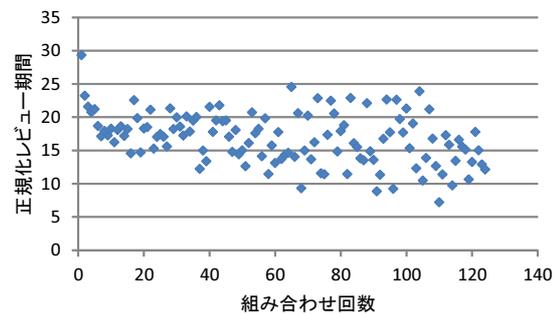


図 1 パッチ開発者とコミッターの組み合わせ回数と正規化レビュー期間の関係

正規化レビュー期間を算出し、分析に用いているのは、パッチによる変更行数の大小がレビュー期間に影響を与える（変更行数が大きければレビュー期間が長くなり、小さければ短くなる）と考えたためである。また、720 時間すなわち 1ヶ月経ってもレビューが完了していないパッチは放置されていると見なし、分析対象から除外した。

## 2.4 実験結果

図 1 に結果を示す。横軸はパッチ開発者とコミッターの組み合わせ回数、縦軸は正規化レビュー期間の平均値である。例えば横軸が 100 の場合、パッチ投稿者・コミッターの組み合わせが 100 回目の時の正規化レビュー期間を、100 回目の全ての組み合わせについて収集し、その平均値を表したのが縦軸の値になる。組み合わせ回数と正規化レビュー期間の相関係数を、組み合わせ回数 200 回まで（200 回目の時点で 26 組のパッチ製作者とコミッターの組み合わせが存在していた）で求めたところ -0.41 という結果が得られた。よって特定の開発者とコミッターの組み合わせ回数の増加に伴いレビュー期間が短期化する、すなわち仮説は支持される事が分かった。

## 3. おわりに

現時点ではパッチ製作者とコミッターの全組み合わせを分析対象としているが、今後は一定回数以上、例えば 100 回以上の組み合わせ回数を持つ組み合わせに限定して分析を行う事を考えている。またレビュー期間以外にもパッチによる変更行数に注目し、組み合わせ回数の増加がパッチによる変更行数に与える影響等についても分析する予定である。

## 参考文献

- 1) <https://android.googlesource.com/kernel/common/+android-3.4>