

令和元年6月11日現在

機関番号：14701

研究種目：若手研究(B)

研究期間：2016～2018

課題番号：16K16037

研究課題名(和文)OSSとコミュニティの共進化に基づくソフトウェア信頼性の理解と操作

研究課題名(英文)Understanding and Controlling Software Reliability Model based on Co-evolution of OSS and OSS Community

研究代表者

伊原 彰紀 (Ihara, Akinori)

和歌山大学・システム工学部・講師

研究者番号：40638392

交付決定額(研究期間全体)：(直接経費) 3,000,000円

研究成果の概要(和文)：本研究では、オープンソースソフトウェア(OSS)の利活用促進に向けて、OSSとコミュニティの共進化に基づくOSSの信頼性を理解するためのOSS信頼性成長モデルの開発、及び、共進化の動向変化によるソフトウェア信頼性の変化を捉えるOSS信頼性成長シミュレーションを行う。商用ソフトウェア開発企業がOSSの利活用を敬遠する理由は、ソフトウェアの信頼性(品質、性能、等)、及び、OSSコミュニティの将来性が不透明であることが挙げられる。本研究では、OSSの信頼性成長過程を理解することで世界中で開発されているOSS市場から信頼性の高いOSSの選択を支援する。

研究成果の学術的意義や社会的意義

本研究課題は、従来研究のようにある任意の時点におけるソフトウェアの信頼性を評価する手法とは異なり、OSSの将来的な信頼性、保守サポートの予測に挑戦している。将来のソフトウェア開発状況を予測することは難しく、本研究では予測時点で計測可能な情報を用いて将来のソフトウェア品質を予測可能か否かを評価した。本手法は、技術が加速的に進化するソフトウェア開発とその品質をリアルタイムに評価する技術として学術的意義と社会的意義を持つ。

研究成果の概要(英文)：This study proposes an open source software (OSS) reliability growth model based on the co-evolution of software product and the developer community. Some industrial software projects avoid reusing OSS, since they concern the quality of software and the unclear schedule of software reliability growth. This proposed model would support to search and to use higher reliability OSS.

研究分野：ソフトウェア工学

キーワード：オープンソースソフトウェア ソフトウェア進化 ソフトウェアリポジトリマイニング メトリクス
ソフトウェア品質

1. 研究開始当初の背景

オープンソースソフトウェア (OSS) を利活用したシステム開発が世界的に加速し、情報処理推進機構 (IPA) の調査では、商用ソフトウェア開発企業が抱える 41.8%の事業で OSS が活用されている。旧来はコスト削減のために利用されていた OSS が、昨今では基幹システムにも Linux が活用されるなど、ミッションクリティカルなシステムに OSS を導入することが増え、ソフトウェア開発企業において OSS は積極的に導入すべきものという意識が高まっている。しかし、OSS の選択、導入時期の判断は OSS 開発モデルの特徴が課題の一つとなっている。伝統的なソフトウェア開発はソースコードを公開せず、限られたメンバで綿密な検証を行った後にリリースする。一方、OSS 開発は、短期間で新しいソフトウェアを公開することで、不特定多数の開発者・利用者から寄せられるフィードバックを基に OSS の機能性や品質を徐々に向上する。言い換えると、ソフトウェア開発企業が OSS を導入する段階 (特に新しいバージョンのリリース直後) で OSS が高品質・高性能であるとは限らない。実際にリリース直後のソフトウェアは多くの欠陥が発見され、OSS を導入する組織は、欠陥発見時の OSS プロジェクトの対応、OSS の将来的な保守サポートを課題として挙げており、OSS 導入前に、ソフトウェアの品質・性能のみならず、OSS 開発に貢献するコミュニティのサポート状況も考慮した OSS の信頼性評価が求められている。

OSS 開発は、ソフトウェア (品質、性能、等)、開発者個人 (コア開発者、ボランティア開発者の貢献、等)、コミュニティ (開放性、協調性、等) が日々進化し、頻繁に新しいバージョンをリリースするため、OSS の信頼性を予測し、世界中で開発されている OSS 市場から高い品質、機能性を有するソフトウェアを選択することは容易ではない。代表者は、本研究開始当初において James Surowiecki の “The Wisdom of Crowds (みんなの意見は案外正しい)” という群衆の英知の考え方にに基づき、多くの者が利活用する OSS は信頼性が高い (機能性が高い、品質が高い、仕様書が多い、等) と考え、OSS を利用する組織数を自動計測する技術 (OSS トレンディング解析技術) を開発し、信頼性の高い OSS を推薦するシステムを開発していた。しかし、本技術は、ある任意の時点における OSS の信頼性を計測し、他の OSS や他のバージョンの利用数を比較する技術であって、OSS 利活用者の課題として挙げている OSS の将来的な信頼性、保守サポートの予測を目的としていない。ソフトウェアは図 1 にも示すように、開発者個人やコミュニティと相互作用しながら進化 (共進化) する。例えば、コミュニティの大規模化 (参加開発者の増加) は、開発速度の加速化、ソフトウェアの機能拡張提案数や欠陥発見/修正数の増加など、OSS とコミュニティの共進化が信頼性に与える影響を理解することなしに OSS の将来を理解することはできないと考え、本着想に至った。

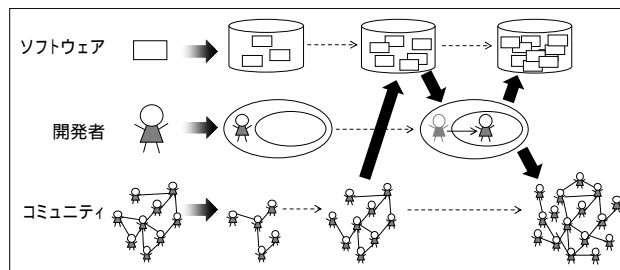


図 1 OSS 開発の共進化

2. 研究の目的

ソフトウェア工学分野における多くの研究では、ソフトウェアの開発・利用過程を表す膨大な過去のデータ (ソースコード、不具合票、コミュニケーションログ等) から機械学習を用いて、ある任意の時点におけるソフトウェア開発の意思決定を支援することを目的とした予測を試みてきた。しかし、OSS 開発では、ソフトウェアの成長過程などは過去のデータから単純に予測することは容易ではない。本研究課題では、OSS とコミュニティの共進化に基づいて OSS の成長過程を解明した。開発者や利用者の貢献、コミュニティの活性化などがソフトウェアに影響するため、OSS 開発に貢献する者の行動次第で OSS の信頼性が変化する [研究業績 5]。本研究課題では、OSS の信頼性の一つの計測方法である不具合の修正時間に着目し、OSS とコミュニティの変化により、ソフトウェアに発見された不具合が修正されるまでにかかる時間の違いを明らかにした。さらに、予測精度を検証することで、開発者や利用者による OSS 成長の操作が実現できるか否かを確認した。

3. 研究の方法

本研究課題は OSS とコミュニティの変化により、ソフトウェアに発見された不具合が修正されるまでにかかる時間の違いを明らかにするために OSS 信頼性モデルを開発した。具体的には、不具合報告から修正が完了するまでの時間 (不具合修正時間) を予測するモデルを構築した。提案する予測モデルは、次期リリースまでに修正する不具合の特定を可能にする。昨今、ラピッドリリースを行う Firefox をはじめとして、多くの OSS プロジェクトではリリースサイクルが短期化しているため、本論文では短期間で修正される不具合を対象とし、不具合の報告日から 1 日以内、3 日以内、1 週間以内 (7 日)、2 週間以内 (14 日)、1 ヶ月以内 (30 日) に修正されるか否かを予測するモデルを構築した。

3.1. 予測モデル構築のためのメトリクスの計測

本研究課題では、不具合修正時間の予測モデルを構築するために、従来研究で使用されている不具合メトリクスに加え、開発状況メトリクスを提案した。

不具合メトリクス: 不具合票は、不具合を発見した機能、利用している開発環境（コンピュータの種類、OS 等）、優先度、機能の振る舞い、再現方法などがあげられる。本研究課題では、報告時点で不具合票から取得可能な情報を不具合メトリクスとして用いた。

開発状況メトリクス: 本研究で提案する開発状況メトリクスは、不具合が報告された直前の時期におけるプロジェクトの開発状況（開発者の作業量、プロダクトの変化量等）を理解するためのメトリクスである。具体的な開発状況メトリクスとして、バージョン管理システム（VCS: Version Control System）、開発者メーリングリスト（ML: Mailing List）、不具合管理システム（BTS）、レビュー管理システム（RMS: Review Management System）の 4 つのリポジトリからメトリクスを計測する。これらのリポジトリは、OSS プロジェクトにおいて開発の進捗情報を随時記録するためのシステムとして使用され、不具合修正活動を理解するための研究で頻繁に利用されている。使用するメトリクスの詳細は 5 章の[研究業績 1]に示す。

3.2. 予測モデルの構築方法

3.2.1. メトリクスの計測方法

不具合メトリクスは、不具合情報が BTS に登録された時点で取得できる情報を取得した。開発状況メトリクスは、不具合報告の直前の期間における活動量や開発者数を計測した。図 2 は、本研究で提案する開発状況メトリクスの計測方法を示す。個々の不具合に対してメトリクスを計測する場合、不具合数とメトリクス数に比例してモデル構築の準備に時間を要するため、分析対象期間を任意の期間（本論文では 10 日間）で分割し、対象とする不具合が報告日を含む期間の一つ前の期間で開発状況メトリクスを計測した。例えば、不具合が任意の月の 13 日に報告された場合、同月の 1 日～10 日の活動量を開発状況メトリクスとして計測した。

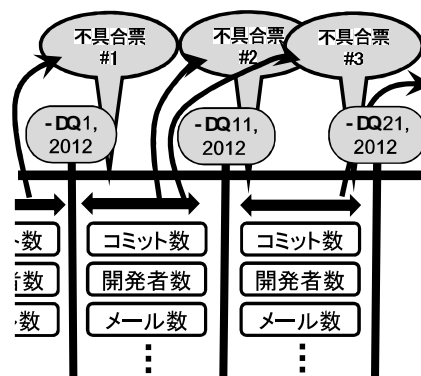


図 2 開発状況メトリクスの計測方法

本論文が開発状況メトリクスを計測するために 10 日間の期間で分割した理由は、不具合ごとが開発状況メトリクスを計測した場合、実験対象の不具合の量から、メトリクスの計測だけで膨大な時間がかかるためである。

3.2.2. 予測アルゴリズム

本論文で提案する予測モデルではランダムフォレスト法を用いて、不具合の報告日から、ある期間（1 日、3 日、7 日、14 日、30 日）以内に修正完了するか否かを予測した。ランダムフォレスト法は、回帰木を用いて集団学習を行う手法であり、2001 年に Breiman によって提案されたアルゴリズムである。ランダムフォレスト法を用いた理由は、集団学習を行うことに適していることや、提案手法のように多くの説明変数を用いる場合にも利用できること、また各メトリクスの重要度を抽出できることが挙げられる。モデル構築用のデータセットに対して繰り返しランダムサンプリングを行い、得られたサンプル群から多数の回帰木を構築した。各回帰木の出力の平均により最終的な予測結果を抽出した。開発状況メトリクスを用いることで予測可能な時期を明らかにすることで、OSS 信頼性成長シミュレーションとして開発者や利用者による OSS 成長の操作が実現できるか否かを確認した。

4. 研究成果

表 1 は、ケーススタディとして対象とした OpenStack プロジェクト、Qt プロジェクトで報告された不具合を説明変数の異なる 2 種類のモデル（提案手法: 不具合メトリクスと開発状況メトリクスを用いて構築したモデル、従来手法: 不具合メトリクスのみ用いて構築したモデル）、また、それぞれのモデルに対して目的変数の 5 種類の指定日数（1 日、3 日、7 日、14 日、30 日）の予測精度（適合率、再現率、F1 値）を示す。太字は、予測期間別に提案手法と従来手法を比較し、予測精度の高い値を示す。

OpenStack プロジェクトでは、報告日から 1 日以内、3 日以内、7 日以内に修正される不具合の予測精度（F1 値）が従来手法よりも高く、Qt プロジェクトにおいては、30 日以内に修正される不具合の予測精度も従来研究よりも精度が高いことを確認した。特に、再現率が向上していることがわかった。従って、従来手法に比べて予測期間以内に修正された不具合を正確に予測できたことを示す。従って、OpenStack プロジェクトにおいて 14 日以内、Qt プロジェクトにおいて 30 日以内に修正される不具合については、開発者や利用者による OSS 成長の操作により、ソフトウェアの品質に影響すると考えられる。

表 1 不具合修正時間予測の結果

| プロジェクト | 予測期間 | 提案手法 (不具合 + 開発状況) | | | 従来手法 (不具合のみ) | | |
|-----------|--------|-------------------|-------|-------|--------------|-------|-------|
| | | 適合率 | 再現率 | F1 値 | 適合率 | 再現率 | F1 値 |
| OpenStack | 1 日以内 | 46.73 | 7.45 | 12.85 | 63.33 | 0.78 | 1.55 |
| | 3 日以内 | 54.47 | 22.13 | 31.47 | 58.07 | 6.22 | 11.24 |
| | 7 日以内 | 58.40 | 48.20 | 52.81 | 57.84 | 48.16 | 52.56 |
| | 14 日以内 | 64.34 | 74.34 | 68.98 | 63.93 | 77.74 | 70.16 |
| | 30 日以内 | 72.08 | 91.52 | 80.64 | 70.04 | 99.83 | 82.32 |
| Qt | 1 日以内 | 60.82 | 3.97 | 7.45 | 50.00 | 1.13 | 2.21 |
| | 3 日以内 | 40.49 | 5.06 | 8.99 | 40.00 | 1.21 | 2.35 |
| | 7 日以内 | 46.06 | 16.36 | 24.15 | 48.06 | 7.09 | 12.36 |
| | 14 日以内 | 52.92 | 35.03 | 42.15 | 61.40 | 18.82 | 28.82 |
| | 30 日以内 | 60.18 | 59.25 | 59.71 | 60.65 | 56.37 | 58.43 |

5. 主な発表論文等

〔雑誌論文〕(計 2 件)

- [1] 伊原彰紀, 若元亮樹, 松本健一, 「開発状況メトリクスを用いた OSS 不具合修正時間予測モデル」, 情報処理学会論文誌, Volume.59, pp.834-844, 2018.
- [2] Shohei Ikeda, Akinori Ihara, Raula Gaikovina Kula, Kenichi Matsumoto, "An Empirical Study of README contents for JavaScript Packages," IEICE TRANSACTIONS on Information and Systems, Volume.E102.D Issue.2, pp.280-288, 2018.

〔学会発表〕(計 8 件)

- [1] 桂川大輝, 伊原彰紀, ラウラ ガイコビナ クラ, 松本健一, 「ソフトウェア開発における同時バージョン変更される併用ライブラリの推薦」, マルチメディア, 分散協調とモバイルシンポジウム 2018, pp.864-871, 2018 年. 査読有り.
- [2] Daiki Katsuragawa, Akinori Ihara, Raula Gaikovina Kula, Kenichi Matsumoto, "Maintaining Third-Party Libraries through Domain-Specific Category Recommendations," The 1st International Workshop on Software Health, pp.2-9, 2018 年. 査読有り.
- [3] 池田祥平, 伊原彰紀, ラウラガイコビナクラ, 松本健一, 「GitHub における README 記述項目の分析」, 第 24 回ソフトウェア工学の基礎ワークショップ, pp.135-140, 2017. 査読有り.
- [4] 桂川大輝, 伊原彰紀, Raula Gaikovina Kula, 松本健一, 「ソフトウェア開発に併用されるライブラリ機能の推薦」, 電子情報通信学会ソフトウェア工学研究会, pp.1-7 2017. 査読無し.
- [5] 桂川大輝, 伊原彰紀, Raula Gaikovina Kula, 松本健一, 「ソフトウェア開発に利用されるライブラリ機能の分析」, 第 24 回ソフトウェア工学の基礎ワークショップ, pp.141-146, 2017. 査読有り.
- [6] Akinori Ihara, Daiki Fujibayashi, Hirohiko Suwa, Raula Gaikovina kula, Kenichi Matsumoto, "Understanding When to Adapt a Library: A Case Study of an ASF Projects," Proc. of the International Conference on Open Source Software Systems, pp.63-70, 2017. 査読有り.
- [7] Hironori Suwa, Akinori Ihara, Raula Gaikovina Kula, Daiki Fujibayashi, Kenichi Matsumoto, "An Analysis of Library Rollbacks: A Case Study of Java Libraries," Proc. of the International Workshop on Software-driven Big Data Analytics, pp.1-10, 2017. 査読有り.
- [8] 池田祥平, 坂口英司, 伊原彰紀, 松本健一, 「操作履歴を利用した不具合票自動生成に向けて」, ウィンターワークショップ 2017・イン・飛騨高山. 査読無し.

科研費による研究は、研究者の自覚と責任において実施するものです。そのため、研究の実施や研究成果の公表等については、国の要請等に基づくものではなく、その研究成果に関する見解や責任は、研究者個人に帰属されます。