

科学研究費助成事業（科学研究費補助金）研究成果報告書

平成 25 年 6 月 2 日現在

機関番号：16301
 研究種目：若手研究(B)
 研究期間：2010 ～ 2012
 課題番号：22700035
 研究課題名（和文） オープンソースソフトウェアに対する品質指標とその予測法の開発

研究課題名（英文） A Study on Quality Criteria and Quality Prediction Methods for Open Source Software

研究代表者

阿萬 裕久 (AMAN HIROHISA)

愛媛大学・大学院理工学研究科・講師

研究者番号：50333513

研究成果の概要（和文）：本研究では、近年広く普及しているオープンソースソフトウェア開発に着目し、ソフトウェアの品質を定量的に評価するための指標について実データに基づく評価・分析を行った。その結果、プログラム中のコメント文の量が有益な品質指標の一つとなることを定量的に明らかにした。また、ソフトウェア開発では長期にわたって改変も多く行われるため、将来の改変に対する予測法として、確率過程モデルを応用した手法の提案と評価も行った。

研究成果の概要（英文）：This study focused on the open source development, which has been popular in the software industry, and did research on some quantitative criteria for software quality with many kinds of practical data. As the results, this study quantitatively revealed that the amounts of comments written in source programs could be one of useful criteria. Moreover, the study developed a predictive method for software upgrades, by using some stochastic process models, and conducted its empirical evaluations.

交付決定額

（金額単位：円）

	直接経費	間接経費	合計
2010年度	1,100,000	330,000	1,430,000
2011年度	500,000	150,000	650,000
2012年度	600,000	180,000	780,000
年度			
年度			
総計	2,200,000	660,000	2,860,000

研究分野：総合領域

科研費の分科・細目：情報学・ソフトウェア

キーワード：ソフトウェア工学，オープンソースソフトウェア，品質管理

1. 研究開始当初の背景

近年、オープンソースソフトウェアは、単なる個人向けアプリケーションに留まらず、OS やサーバ、組込み機器制御等で広く活用されるようになってきている。産業界においても製品の一部にオープンソースソフトウェアを活用しようとする場面は珍しくない。

しかしながら、その際には当該ソフトウェアの品質保証という点で大きな課題がある。すなわち、

“そのオープンソースソフトウェアの品質は、製品へ組み込めるようなレベルにあるのか？”

“仮に不具合が見つかった場合、今後も改善されるという見込みはあるのか？”

といった疑問に対し、適切な回答が得られるという保証はない。このことは、オープンソース開発の持つ“自由”という基本概念に起因するものであり、オープンソース開発に参加している人たちからして見れば、上述の課題は“課題ではない”と解釈されるであろう。その一方で、現場の技術者からは、上述の（品質保証に関する）課題解決を望む声をしばしば耳にする。理想を言えば、オープンソースソフトウェアに対する機能の追加・修正やテストは誰でも自由に行ってよい。しかし現実的には、そのための技術者や時間が（場合によってはそのためのスキルも）足りないという問題があり、可能な限り少ないコストでオープンソースソフトウェアを活用したいというニーズが現場にはある。

2. 研究の目的

研究の目的は主として次の2つである。

(1) オープンソースソフトウェアに対する品質評価と予測に関する研究を行う。

具体的には、ソースコードの（品質に関係すると思われる）特徴を定量的尺度、即ちメトリクスによって数値化し、多数のデータ収集と統計解析を行うことで欠陥の多いソースコードや変更の起こりやすい（不安定な）ソースコードとそうでないソースコードとの特徴の違いを見出し、評価や予測に役立てる。

(2) 個々のソースコードだけでなく、“よりマクロな視点”からの解析も行う。

ソフトウェアの安定性（変更の有無や大きさ）を予測しようとした場合、理想的には各ソースコードの安定性予測を積み上げることで予測を実現すべきである。しかし、現実には開発者という未知のファクタ（オープンソース開発の場合は未知と考えざるをえない）が影響し、個体差も大きいことから、集合を対象としたマクロな視点での解析も行う。

3. 研究の方法

本研究は大きく次の3ステップを進めていく。

(1) データの収集

さまざまなオープンソース開発サイトから、可能な限り多くのソフトウェアについて、その各バージョンのソースコード、変更履歴並びにバグデータを収集する。データはインターネットを介して収集可能ではあるが、手作業で収集するには限界があり、そのためのソフトウェア開発も行う。

(2) データの整理とデータベース化

収集したデータを円滑に利用できるよう、研究室内にデータベースサーバを用意し、そこに収集したデータを蓄積していく。またあわせて、対象としたソフトウェアにおける各ソースファイルの品質特性をメトリクスによって数値化し、データベースに記録していく。その際、必要に応じてメトリクス測定ソフトウェアも開発する。

(3) データ解析とモデルの試作

①メトリクス値に対する統計解析を通じて、どういった特徴のソフトウェアにはバグの潜在が疑わしいかを解析する。バグの潜在は後にソースコードの修正を引き起こす主要な要因であるため、バグの潜在性と関係の強い特徴（メトリクス）を見出すことは本研究の柱となる。

②マクロな視点での解析として、確率過程モデルの応用を行う。ソースコード改変という事象はバグ修正や機能拡張といった理由から引き起こされるものであるが、第三者にとってその詳細は必ずしも入手可能な情報とは限らないため、それまでの改変履歴を入力として、そこに確率過程の概念を組み合わせることで今後の改変事象を予測していく。

4. 研究成果

(1) データの収集

オープンソースソフトウェアの開発サイト（リポジトリ、バグ管理システムを含む）Eclipse.org, SourceForge.net, Apache.orgを中心として、ソースコードそのものとソースコードの改変、バグ情報について約20万件のデータ収集を行った。また、バグ管理システムについては、データを自動収集するソフトウェアを開発し、代表者のWebサイトにて公開している（URL：<http://www.hpc.cs.ehime-u.ac.jp/~aman/group/index.php?BugDataCollector>）。

(2) データの整理とデータベース化

収集した20万件以上のデータをデータベース化し、容易に解析できるよう準備を行った。また、PROMISE データリポジトリからもメトリクスデータを入手可能であったため、それらもデータベースに格納した。

ソースコードを対象としたメトリクス測定に関しては、

①コード行数の測定

②メソッド情報の抽出

③リポジトリ上のコミットログでバグ修正と思われる案件の抽出

④コメント行数の測定

をそれぞれ自動的かつ円滑に行うため、専用

のソフトウェアを開発し、代表者の Web サイトで公開している (URL : <http://www.hpc.cs.ehime-u.ac.jp/~aman/project/tool/>).

(3) データ解析とモデルの試作

①いくつかのメトリクスについて、測定と分析を行った結果、メソッドや関数の中に局所的に書かれるようなコメント文の存在がバグの潜在性と関係しているという傾向が得られた。従来より、コメント文はソースコードの可読性を高める肯定的な存在であり、積極的に記述するように指導されることが多い。しかしながら、ソースコードが全て公開されるオープンソースソフトウェアでは、洗練されたソースコードが多いこともあり、かえってコメントの多いプログラムの方が内容的に分かりにくいという傾向があるものと考えられる。実際、リファクタリングの分野では、コメント文はコードの不吉な匂いと関係している (消臭剤としての役割) とも言われており、本研究ではその傾向を定量的に示すことができた。結果の一部 (国際会議 IWESep2012 にて発表) を図 1 に示す。

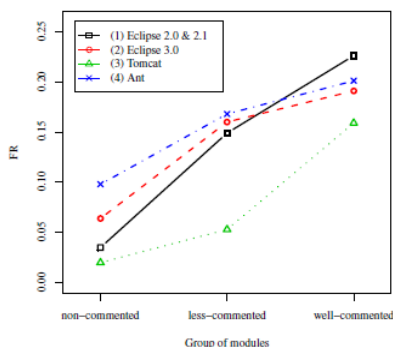


図 1. コメント文量の分類とバグ潜在率

図 1 では縦軸がバグ潜在率を表し、横軸はコメント文量のカテゴリ (コメント無し, 少なめ, 多め) を表している。各折れ線はソフトウェアごとの結果であり、いずれもコメント文が多いほどバグ潜在性が高いという結果が得られている。「コメント無し」に比べて「コメントの多い」プログラムではバグの潜在性が 2 倍から 8 倍高いという傾向を確認できた。すなわち、オープンソースソフトウェアにおいて、局所的なコメント文の存在はその品質を評価する上で重要な指標の一つになるといえる。

本研究に関しては、学界での評価も高く、国際会議 IWESep2012 で Best Presentation Award, 情報処理学会より山下記念研究賞, 電子情報通信学会ソフトウェアサイエンス研究会より研究奨励賞をそれぞれ受賞した。

②従来よりバグ検出の過程を確率過程モデルの一つである非同次ポアソン過程モデル (NHPP) にてモデル化し、ソフトウェアの信頼性評価や残存バグ数の予測を行うといった研究が盛んに行われていた。本研究ではこの概念をソースコード改変に適用し、実際のオープンソースソフトウェアにおけるコード変更量の予測を行った。具体的には図 2 に示すように過去のコード変更に対して NHPP モデルの当てはめを行い、曲線のパラメータ推定を行う。

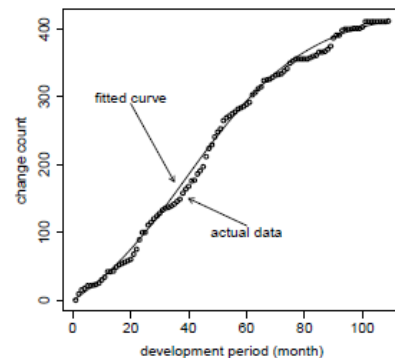


図 2. 開発期間 (横軸) と累積コード修正量 (縦軸) 及びそれに当てはめた習熟 S 字形成長曲線モデル

そして、そうして構築したモデルから数ヶ月後の修正量をどれだけ正確に予測できるか検討した。結果の一部を図 3 (国際会議 IWSM/Mensura2011 にて発表) に示す。

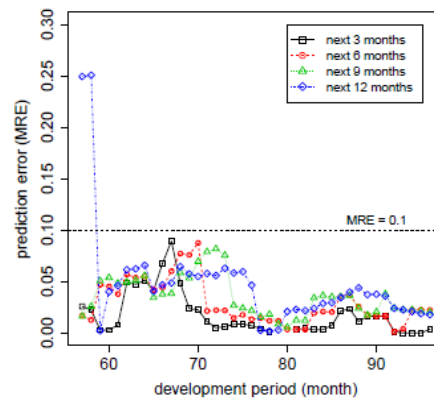


図 3. 開発期間 (横軸) と予測精度 (縦軸)

多少のばらつきはあるが、数ヶ月後の予測を 10 パーセント未満の誤差率で予測できることを示すことができた。ただし、すべてのオープンソースソフトウェアに対して成立する訳ではなく、コード発展の形状によってモデルを使い分ける必要があるというのが現時点での結論である。しかしながら、オープ

ンソースソフトウェア開発における今後のバグ修正や機能発展がどれくらいの規模起こりそうかを確率過程モデルで予測するというアプローチに一定の有用性があることは一つの大きな成果であると考えられる。

③その他

オープンソースソフトウェア開発の品質評価に関連して、メトリクスによる評価値をどのように活用するかという切り口でも研究を並行して行った。特にメトリクスによる評価値から各ソースファイルにおけるバグ潜在の可能性をそのファイルをレビュー・テストする価値と定義し、ファイル規模をレビュー・テストに要するコストと定義して、レビュー・テストの計画を数理計画問題として定式化する手法を提案した。提案手法は効率的にレビューやテストを行うための管理手法の一種であり、現時点ではオープンソースソフトウェア開発に対して直接的に貢献できるか否かは明らかにできていないが、学界・産業界からは一定の評価を得ている（日本ソフトウェア科学会・ソフトウェア工学の基礎ワークショップ2011では貢献賞を獲得）（東芝ソフトウェア技術センターとの共同研究に発展）。現在、他大学の研究者とともにオープンソースソフトウェア開発での修正担当者割り当て問題に応用できないかという切り口で研究を行っている。

5. 主な発表論文等

（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計4件）

- ① Myrizki Sandhi Yudha, Ryohei Asano and Hirohisa Aman, A Feature Analysis of Co-Changed Code Clone by using Clone Metrics, IEICE Trans. Fundamentals, Vol. E95-A, No. 9, pp. 1498-1500, 2012（査読有り）。
- ② 阿萬裕久, 論理的制約条件付き 0-1 計画モデルを用いた重点レビュー計画法, コンピュータソフトウェア, vol. 29, no. 3, pp. 115-120, 2012（査読有り）。
- ③ 阿萬裕久, オープンソースソフトウェアにおけるコメント記述及びコメントアウトとフォールト潜在との関係に関する定量分析, 情報処理学会論文誌, 第53巻, 2号, pp. 612-621, 2012（査読有り）。
- ④ 阿萬裕久, 山下裕也, 整数計画法を用いた重点レビュー対象モジュールの選択, コンピュータソフトウェア, vol. 27, no. 4, pp. 240-245, 2010（査読有り）。

〔学会発表〕（計16件）

- ① 井上慎也, 阿萬裕久, 0-1 計画モデルを

利用したメソッド単位でのコードレビュー計画について, ウインターワークショップ 2013・イン・那須論文集, pp. 27-28, 2013年1月25日, 那須町（査読有り）。

- ② Hirohisa AMAN, An Empirical Analysis of the Impact of Comment Statements on Fault-Proneness of Small-Size Module, Proc. 19th Asia-Pacific Software Engineering Conference (APSEC 2012), pp. 362-367, 2012年12月6日, 中国・香港市（査読有り）。
- ③ Hirohisa AMAN, An Empirical Analysis on Fault-proneness of Well-Commented Modules, Proc. 4th International Workshop on Empirical Software Engineering in Practice (IWESPE 2012), pp. 3-9, 2012年10月3日, 大阪市（査読有り）。
- ④ 阿萬裕久, コメントの多いプログラムとフォールトとの関係について ---コード行数との相関を利用したコメント量評価と実証的分析---, 2012年8月27日, 東京都（査読有り）。
- ⑤ 浅野遼平, 阿萬裕久, コメント記述量の増加傾向とフォールト潜在との関係に関する定量分析, 信学技報, vol. 112, no. 164, pp. 109-114, 2012年7月28日, 函館市（査読無し）。
- ⑥ 浅野遼平, 阿萬裕久, ソースファイルにおけるコメントの増加傾向に着目したフォールト潜在予測, ソフトウェア・シンポジウム 2012 論文集, 2012年6月14日, 福井市（査読有り）。
- ⑦ 阿萬裕久, Fault-prone モジュール予測に対するコメント記述量の効果に関する考察, ウインターワークショップ 2012・イン・琵琶湖論文集, pp. 45-46, 2012年1月19日, 彦根市（査読有り）。
- ⑧ Myrizki Sandhi YUDHA, Ryohei ASANO and Hirohisa AMAN, An Analysis of Co-changeable Code Clone by using Clone Metrics, Supplemental Proc. IEEE 22nd International Symposium on Software Reliability Engineering (ISSRE 2011), CD-ROM, 2011年12月2日, 広島市（査読有り）。
- ⑨ Hirohisa AMAN, A Proposal of NHPP-Based Method for Predicting Code Change in Open Source Development, Proc. The Joint Conference of the 21st International Workshop on Software Measurement and the 6th International Conference on Software Process and Product Measurement (IWSM-MENSURA 2011), pp. 38-47, 2011年11月3日, 奈良市（査読有り）。

- ⑩ 阿萬裕久, Fault-Prone モジュール予測を利用した効率的なレビュー計画及びテスト計画について, オンライン, 2011年9月12日, 東京都 (査読有り).
- ⑪ Myrizki Sandhi Yudha, 阿萬裕久, コードクローンの長さと同様変更の起こりやすさとの関係に関する解析, ソフトウェア・シンポジウム 2011 論文集, 2011年6月9日, 長崎市 (査読有り).
- ⑫ 浅野遼平, 阿萬裕久, オープンソース開発におけるコード変更の連鎖とその発生率について, 信学技報, vol. 111, no. 71, pp. 31-36, 2011年5月26日, 高松市 (査読無し).
- ⑬ 阿萬裕久, ソースコード変更事象の数理モデル化と予測について, ウィンターワークショップ 2011・イン・修善寺論文集, pp. 109-110, 修善寺市, 2011年1月21日 (査読有り).
- ⑭ 阿萬裕久, フォールト潜在予測に向けたコメント文記述及びコメントアウトの定量分析, 信学技報, vol. 110, no. 305, pp. 13-18, 2010年11月24日, 浜松市 (査読無し).
- ⑮ Hirohisa AMAN and Takahiro OHKOCHI, An Application of Growth Curve Model for Predicting Code Churn in Open Source Development, Proc. Ninth Joint Conference on Knowledge-Based Software Engineering (JCKBSE' 10), pp. 46-54, 2010年8月25日, リトアニア・カウナス市 (査読有り).
- ⑯ 大河内嵩博, 阿萬裕久, オープンソース開発におけるコード変更量の数理モデル化と予測について---成長曲線モデルを用いた実験---, 信学技報, vol. 110, no. 158, pp. 25-30, 2010年7月30日, 釧路市 (査読無し).

[図書] (計3件)

- ① 浅野遼平, 阿萬裕久, ソースコード改変におけるコメント文の相対的な変化量に着目したフォールト潜在性の分析, ソフトウェア工学の基礎 XIX (鶴林尚靖, 亀井靖高 編), pp. 45-50, 近代科学社, 2012.
- ② 阿萬裕久, 論理的制約条件付 0-1 計画問題として定式化した重点レビュー対象モジュールの選択, ソフトウェア工学の基礎 XVIII (門田暁人, 上野秀剛 編), pp. 197-206, 近代科学社, 2011.
- ③ 阿萬裕久, オープンソースソフトウェアにおけるコメント文記述とフォールト潜在率との関係に関する実証的考察, ソフトウェアエンジニアリング最前線 2010 (松下誠, 紫合治 編), pp. 97-100, 近代科学社, 2010.

6. 研究組織

(1) 研究代表者

阿萬 裕久 (AMAN HIROHISA)

愛媛大学・大学院理工学研究科・講師

研究者番号：50333513