

## 科学研究費助成事業 研究成果報告書

平成 30 年 6 月 8 日現在

機関番号：15401

研究種目：基盤研究(C) (一般)

研究期間：2015～2017

課題番号：15K01192

研究課題名(和文) ソフトウェア信頼性評価のためのビッグデータアナリシス基盤整備

研究課題名(英文) Big data analysis platform for software reliability assessment

研究代表者

土肥 正 (Tadashi, Dohi)

広島大学・工学研究科・教授

研究者番号：00243600

交付決定額(研究期間全体)：(直接経費) 3,600,000円

研究成果の概要(和文)：近年のソフトウェア開発では、開発工程で得られた様々なメタデータ(設計情報、管理情報、プログラムソースコード等)をリポジトリで一元管理し、データ共有を通じて開発・保守に活用することが多い。ソフトウェア規模が大きいほど開発で蓄積されるリポジトリサイズも相対的に大きくなり、リポジトリから重要な情報を発掘するマイニング技術の開発が重要な課題となっている。本研究では、リポジトリに蓄積された情報からソフトウェア信頼性を定量的に評価するための基盤技術を世界に先駆けて開発した。オープンソースプロジェクトのリポジトリを横断的に分析し、ソフトウェアの定量的信頼性を算出する手法およびツールの開発を行った。

研究成果の概要(英文)：In the modern software development it is common to record the meta data observed in the development process, such as design information, process information, source code, etc., in the database called repository, and to utilize them for development and maintenance activities through the development project. It is known that as the development size increases the capacity of repository should be increasing. Then the data mining technologies for the repository data are becoming important. In this project, we developed a new fundamental technology to assess software reliability by utilizing the repository data. Especially, we analyzed several open source project data and developed a methodology and tool to quantify the software reliability measures.

研究分野：信頼性工学

キーワード：ソフトウェア信頼性 ソフトウェアメトリクス バグ予測 リポジトリ データマイニング 信頼性評価 ビッグデータ 確率モデリング

## 1. 研究開始当初の背景

ソフトウェアの高信頼化は近年の大きな課題である。ソフトウェアの信頼性を確保するためには、開発プロセス管理技術の向上と開発手法の向上を考慮した PDCA サイクル (Plan, Do, Check, Action) をうまく運用することが重要である。つまり、プロセスと製品の信頼性を「定量化」(Check) し、プロジェクト管理技術や開発手法に「フィードバック」(Action) する作業は高信頼化ソフトウェアを開発するための重要なプロセスである。

1970 年代初頭から研究が開始されたソフトウェア信頼性モデルでは、テストで発見されるフォールト数を統計的に推論することで、定量的なソフトウェア信頼度を算出することができる。このように、テストで発見されたバグの個数だけから信頼度を推定する手法は非常に扱いが簡単であり、現在でも多くの民間企業でソフトウェア品質を評価するために利用されている。しかしながら、従来のモデルではバグ数のみを利用しているため「何が信頼性へ寄与したのか」といった要因と信頼性への因果関係がブラックボックス化している。従って、「得られた数値の妥当性の検証が難しい」、「管理技術、開発手法への明示的なフィードバックが難しい」などの重大な欠陥をもつことが指摘されてきた。

ソフトウェア信頼性評価技術における最も重要なポイントは「信頼性評価の妥当性検証を可能にすること」と「管理技術、開発手法へフィードバック可能な情報を与えること」である。そのためには信頼性と要因の因果関係を明らかにすることが最も重要である。最近、申請者らによって提唱されている「実証的ソフトウェア信頼性工学」では、信頼性と要因の因果関係を明らかにすることを念頭に置いた分析を行っている。つまり、従来のようにバグの発見個数だけ

を用いるのではなく、製品やプロセスを特徴づけるメタ情報を考慮した信頼性評価を目指している。これを実現するためには信頼性評価モデルに統計的回帰構造を導入することが有効である。実際、これまでに申請者らの研究グループによっていくつかのモデル (Cox 回帰ベース NHPP モデル, ロジック回帰ベース NHPP モデル, ポアソン回帰ベース NHPP モデル, 一般化線形回帰ベース NHPP モデル, マルコフ従属ポアソン回帰モデル, 他) が提案されてきた。これらのモデルを用いることで、ソフトウェアあるいは開発プロセスを特徴付ける数値 (ソフトウェアメトリクスと呼ぶ) と信頼性の関係を対応付けることに成功している。しかし、一般的には、開発情報に関するメトリクスの計測や選択にはコストがかかるため、容易に計測可能なメトリクスだけを用いた小規模なケーススタディーしか扱うことが出来なかった。

## 2. 研究の目的

ソフトウェアメトリクスには様々な種類がある。コード行数、関数の数などソースコードを走査することで簡単に計測できるものから、プログラムの依存グラフのようにソースコードを詳細に分析しなければならないもの、テストカバレッジのように実際にテストを実行しないと計測できないものなど多種多様にわたっている。これまで申請者らが開発してきた回帰に基づいたソフトウェア信頼性モデルは、理論的にはどのようなソフトウェアメトリクスでも扱える構造になっている。端的に言えば、文字列やグラフといった構造的なデータに対しても適切な前処理さえ実行すれば適用できる構造となっている。しかしながら、実際のソフトウェアプロジェクトで管理されているような大規模なリポジトリでは、前処理のための計算コストが高く、これまでは小規模なリポジトリ且つ単一プロジェクトのリポジトリを扱うのみに留まっていた。ビッグデータのデータマイニングに代表されるように、データのスケールが増えることで初めて発掘される情報も存在するが、従来の回帰に基づいたソフトウェア信頼性評価では計算時間の観点からこのような情報を取り扱うことができなかった。

本研究では、ブースティングやランダム

フォレスト等のデータマイニング技術を応用することでリポジトリに格納される情報をランダムに分割し、分割されたデータを並列的に統計処理することで、複数のリポジトリに埋蔵されている情報を発掘し、それらを有効に活用したソフトウェア信頼性評価の体系を構築した。同時に、そのようなソフトウェア信頼性評価の体系を実現するためのアルゴリズム開発とそれを実装したツールを開発することを目的とした。

### 3. 研究の方法

より具体的には、ランダムなデータ分割後に、それぞれのデータに対して NHPP (非定常ポアソン過程) を仮定した一般化線形回帰モデルを適用し、その結果を集約することで、信頼性評価モデルの同定を行った。このような、ランダムなサンプリングに基づくデータ分割は、大規模な組合せ最適化問題に帰着される要因選択を排除できることが知られている。また、個々の回帰係数を理論的に推定する方法としては、最尤推定法だけでなく、ベイズ推定法、ウェーブレット縮小推定法の適用を考え、各々の推定作業の計算コストを削減するために EM (Expectation-Maximization) アルゴリズム、変分ベイズアルゴリズム、MCMC (Markov Chain Monte Carlo) 法の導入も行った。さらに、並列計算における各スロットからのアウトプットを平均化することで、未知変数の予測分布やその推定値を精度良く求めるための理論的な枠組みを新たに構築した。最終的に、オープンソースプロジェクトのリポジトリを対象にして有効性を検証するための実証分析を行った。

また、3年間の研究期間を通じて、ソフトウェアの定量的信頼性評価を行うためのツール開発を行った。これにより、ユーザは難解な確率過程や推定・検定作業を意識することなく、高精度の定量的信頼性評価を容易に実施することが可能になるものと期

待される。推定された回帰係数に対する検定結果を通じて、当該ソフトウェア開発において過不足であると予想される工程やフォールトブローンコンポーネントを可視化することが出来るため、高信頼性の観点からソフトウェアの開発工程を見直すための管理ツールとしての利用も考えられる。最終的に、開発したツールを無償で公開し、ソフトウェア開発現場への普及・啓蒙活動を行いたいと考えている。

### 4. 研究成果

#### (1) データ整理・分析：

代表的なオープンソースソフトウェアのリポジトリを種別・開発規模の観点から整理・分類し、ソースコードリポジトリ、バグトラッキング情報、構成情報(開発言語、モジュール構成、プラグイン構成、GUI ツール)、品質情報(発見バグ数、バグ修正率、平均バグ修正時間)等の情報を集約するためのデータベースを開発した。開発・運用期間が長いシステムやモジュール数が極端に大きいシステムのデータを整理するために、オープンソースデータベース MySQL を利用し、定量的信頼性評価に特化したデータ構造を定義した。また、設計・コーディング・テスト段階で計測可能な各種メトリクス情報(設計メトリクス、ソースコードメトリクス、プロセスメトリクス、複雑性メトリクス、テストメトリクス)を測定するためのツールを組込むことで、データベースと連動した分析基盤を開発した。特に、テストケース情報を利用する場合に対して、予め準備されたテストベッドに対して各テストケースの入力とテスト実行パスを計測するツールを開発した。その後、テストケース間の入力と実行パスから計算される各種「距離」(C0 カバレッジによる距離、ステートメントの実行回数による距離、C1 カバレッジによる距離)を計算した。ソースコード情報を利用する場合には、

ソースコードから PDG (Program Dependence Graph) を生成し、グラフ構造データを構成する必要がある。これには、大阪大学で開発されたコードクローン検出ツール CCFinder を部分的に利用し、解析されたトークン列と文字列を情報として抽出した。

(2) 一般化線形モデルの改良：

定量的信頼性評価の基本的な枠組みは、バグ発見情報とその他の開発・テスト情報の因果関係を確率モデルとして記述することにある。そのため、ポアソン回帰や負の二項回帰を含む通常の線形回帰の方法を直接適用出来ず、統計ソフト R のようなフリーツールをそのままの形で利用することも難しい。NHPP 上で回帰構造を持つ確率点過程としてのソフトウェア信頼性モデルを用いることが有効である。但し、回帰モデルに非線形構造を導入し、かつトークン列や文字列情報を扱う必然性があるため、従来モデルの大幅な改良が必要となる。ここでは、これまでの研究成果を活用し、ポアソン回帰とロジット回帰（もしくはカーネル回帰）の両方を同一モデル内に含む「一般化線形モデル」に着目し、その統計パラメータを効率的に推定するための近似推定アルゴリズムを開発した。EM 原理に基づいた最尤推定アルゴリズム、変分ベイズとラプラス近似に基づいたアルゴリズム、縮小推定に基づいたアルゴリズムをそれぞれ開発した。最終的に、パラメータの漸近収束性や精度の観点からアルゴリズムの評価を行い、データ数に関する計算限界水準を同定し、並列化統計分析のための準備を実施した。

(3) ランダムサンプリングによる並列計算スキームの開発：

数年間にもものぼる長期の開発データや数百から数千オーダーのモジュールを有する大規模ソフトウェアの分析では、要因選択の前処理をすることなく全てのデータを一括して扱うことは絶望的である。そこで直交表を用いて因子の絞り込みを行い計算量を削減することも考えられるが、これさえも膨大な組合せ最適化問題を避けることが出来ず、有用な開発・テスト情報を信頼性評価に十分活用することが困難である。ここでは最も単純かつ効果的なランダム分割の方法を改良し、ブースティングやラン

ダムフォレストを用いた計算量削減の方法を提案した。これにより、問題を並列化された統計的推定問題に帰着させることが可能となり、EM アルゴリズムや変分ベイズなどのアルゴリズムを適用出来る規模まで問題を縮約することが出来た。最終的に複数スロットにおいて算出される推定結果をランダム分割の統計的性質に応じて平均化し、大規模ソフトウェアに対する定量的信頼性評価の基本統計量である各種パラメータの推定値を求めた。

(4) 信頼性評価尺度の算出とアラーム・判定機能の設計：

一旦ソフトウェア信頼性モデルを特徴づけるパラメータが推定されると、その結果を定量的信頼性評価尺度の評価（ソフトウェア信頼度、ソフトウェア MTTF、期待残存フォールト数、フォールトフリー確率、他）に適用し、リリース判定と前工程のフィードバック評価を行う必要がある。特に大規模ソフトウェアのリポジトリ活用では多くの回帰パラメータの推定値が求まるので、各々のパラメータと工程の評価項目（開発労力、テストケース数、プログラム複雑性などに関する複数項目）を合致させる必要がある。これを分析結果に応じて逐次検討することは大変コストがかかるため、予め選択されたメトリクス情報の集合と評価項目のマッチングを自動的に行い、前工程のフィードバック評価を実現するツールの開発を行った。

(5) 未踏プロジェクトへの対応：

類似開発プロジェクトや同質的な種別や規模のシステムに対しては、過去の開発情報を利用したプラグイン推定を行うことが可能であるが、これまでに開発経験のない未踏プロジェクトにおける定量的信頼性評価を行うためには、過去の開発情報から推定された回帰パラメータをそのまま利用することは出来ない。このような統計的予測

推論では統計量の予測分布を導出しなければならないが、一般的には技術的に困難な問題が数多く存在する。しかしながら、本研究で提案するランダム分割の手法を用いることで、各並列計算タスクから出力された推定量を独立で同一な確率変数として見なせるため、統計的ブートストラップの方法を適用することが可能となった。これにより、各種信頼性評価尺度やパラメータの予測分布を数値的に算出することが出来た。

(6) 実証分析:

本研究で提案する大規模ソフトウェアリポジトリに対する定量的信頼性評価の方法を、いくつかの実在するオープンソースプロジェクトに適用し、その有効性を検証した。リポジトリに格納されているデータの種類によっては、アラーム・判定機能をそのまま適用することが困難である場合も散見された。また、ランダム分割や並列計算のスロットを調整することで扱うことの出るデータサイズを同定し、NHPPベースの一般化線形モデルの粒度を決定しなければならなかった。

(7) プロトタイプ開発:

3年間の各研究フェーズで開発したツールを結合し、ソフトウェア信頼性評価のためのビッグデータアナリシス基盤整備のための基本ツールの開発を行った。バグトラッキングシステムとの互換性やリポジトリのデータ構造の統一化作業を念頭に、(実用ベースではなく)実験ベースで利用可能なプロトタイプ開発を実施した。リポジトリのサイズが大きくなればなる程、ツールのテストケースの数は膨大となるため、2か月間のテスト期間を準備した。ツールの結合テストとシステムテストは広島大学で行い、従前までに開発した各種信頼性評価ツール(SRATS, RSRATS)を特別な場合として含むため、これらの評価結果との整合性を確認しながら、開発ツールの問題点を洗い出す作業を行った。

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20件)

(1) B. Momotaz and T. Dohi, Optimal stopping time of software system test via artificial neural network with fault count data, 査読有, Journal of Quality in Maintenance Engineering, vol. 24, no. 1, pp. 22--36, 2018.

(2) J. Zheng, H. Okamura and T. Dohi, A comprehensive evaluation of software rejuvenation policies for transaction systems with Markovian arrivals, 査読有, IEEE Transactions on Reliability, vol. 66, no. 4, pp. 1157--1177, 2017.

(3) B. Momotaz and T. Dohi, Optimal release time estimation of software system using Box-Cox transformation and neural network, 査読有, International Journal of Mathematical, Engineering and Management Sciences, vol. 3, no. 2, pp. 177--194, 2017.

(4) K. Rinsaka and T. Dohi, An adaptive cost-based software rejuvenation scheme with nonparametric predictive inference approach, 査読有, Journal of the Operations Research Society of Japan, vol. 60, no. 4, pp. 461--478, 2017.

(5) 土肥正, 岡村寛之, 定性的信頼性/安全性解析支援ツールの開発, 査読有, SECジャーナル, vol. 13, no. 2, pp. 28--35, 2017.

(6) B. Momotaz and T. Dohi, A neuro-based software fault prediction with Box-Cox power transformation, 査読有, Journal of Software Engineering and Applications, vol. 10, pp. 288--309, 2017.

(7) T. Dohi and H. Okamura, Dynamic software availability model with rejuvenation, 査読有, Journal of the Operations Research Society of Japan, vol. 59, no. 4, pp. 270--290, 2016.

(8) H. Okamura and T. Dohi, Phase-type software reliability model: parameter estimation algorithms with grouped data, 査読有, Annals of Operations Research, vol. 244, no. 1, pp. 177--208, 2016.

(9) B. Momotaz and T. Dohi, Estimating prediction interval of cumulative number of software faults using back propagation algorithm, 査読有, International Journal of Computer and Information Science, vol. 17, no. 2, pp. 25--34, 2016.

(10) Y. Saito and T. Dohi, Predicting software reliability via completely monotone nonparametric estimator with grouped data, 査読有, Journal of Systems and Software, vol. 117, pp. 296--306, 2016.

(11) H. Okamura and T. Dohi, Analysis of optimal restart policies for software systems, 査読有, Journal of the Japan Industrial Management Association, vol. 66, pp. 416--425, 2016.

- (12) Y. Saito and T. Dohi, Optimal software release decision based on nonparametric inference approach, 査読有, Journal of the Japan Industrial Management Association, vol. 66, pp. 396--405, 2016.
- (13) K. Rinsaka and T. Dohi, Toward high assurance software systems with adaptive fault management, 査読有, Software Quality Journal, vol. 24, pp. 65--85, 2016.
- (14) H. Okamura and T. Dohi, PH fitting algorithm and its application to reliability engineering, 査読有, Journal of the Operations Research Society of Japan, vol. 59, no. 1, pp. 72--109, 2016.
- (15) C. Luo, H. Okamura and T. Dohi, Optimal planning for open source software updates, 査読有, Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability, vol. 230, no.1, pp. 44--53, 2016.
- (16) 土肥正, 岡村寛之, 次世代ソフトウェア信頼性評価技術の開発に向けて, 査読有, SEC ジャーナル, vol. 10, no. 10, pp. 28--35, 2015.
- (17) T. Imanaka and T. Dohi, Software reliability modeling based on Burr XII distributions, 査読有, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences (A) vol. E98-A, no. 10, pp. 2091--2095, 2015.
- (18) H. Okamura, J. Guan, C. Luo and T. Dohi, Quantifying resiliency of virtualized system with software rejuvenation, 査読有, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences (A), vol. E98-A, no. 10, pp. 2051--2059, 2015.
- (19) Y. Saito and T. Dohi, Software reliability assessment via non-parametric maximum likelihood estimation, 査読有, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences (A), vol. E98-A, no. 10, pp. 2042--2050, 2015.
- (20) K. Shibata, K. Rinsaka and T. Dohi, M-SRAT: Metrics-based software reliability assessment tool, 査読有, International Journal of Performability Engineering, vol. 11, no. 4, pp. 369-380, 2015.

〔学会発表〕(計 34 件)

- (1) H. Okamura and T. Dohi, A generalized bivariate modeling framework of fault detection and correction processes, The 26th International Symposium on Software Reliability

Engineering (ISSRE 2017), Toulouse, France, October 23--26, 2017.

## 6. 研究組織

研究代表者

土肥 正 (DOHI Tadashi)

広島大学・大学院工学研究科・教授

研究者番号：002436