

令和 6 年 5 月 14 日現在

機関番号：14603

研究種目：若手研究

研究期間：2021～2023

課題番号：21K17725

研究課題名（和文）リファクタリングにより破壊されるテストスイート予測技術の開発：自動修正への挑戦

研究課題名（英文）Developing techniques for estimating the size of test suites broken by refactoring

研究代表者

柏 祐太郎 (Yutaro, Kashiwa)

奈良先端科学技術大学院大学・先端科学技術研究科・助教

研究者番号：20886650

交付決定額（研究期間全体）：（直接経費） 3,500,000円

研究成果の概要（和文）：ソフトウェア品質を向上させるためにリファクタリングが広く実施されているが、リファクタリングは多大な時間と労力を要することが多く、開発期間が短く限られている場合は敬遠される傾向にある。特に、開発者にリファクタリングを躊躇させる心理的要因の一つが「リファクタリングによるテストスイートの破壊」である。本応募課題では、どのようなリファクタリングがテストスイートを破壊するかを分析し、リファクタリングによるテストスイートの影響範囲および必要修正箇所の特定技術を確立した。また、テストスイートの自動修正ツールを作成し、複数種のリファクタリングによって破壊されたテストスイートを修復できることを確認した。

研究成果の学術的意義や社会的意義

リファクタリングはソフトウェア品質を向上させるために欠かせない重要な活動である。その一方、リファクタリングを実施することによって、テストスイートが破壊されることが知られている。本研究における学術的意義、テストを破壊するリファクタリングの種類と、その破壊の程度を明らかにしている点である。また、本研究課題で開発した破壊箇所の予測技術および破壊したテストスイートを修復技術は、リファクタリングを実施しやすい環境を実開発者に提供しており、社会的意義が大きいと言える。

研究成果の概要（英文）：Refactoring operations are performed by many practitioners in order to improve software quality. However, developers sometimes hesitate to operate refactorings because refactorings often break the test suites. In this project, we analyzed what kinds of refactoring break test suites, and developed a technique for estimating the size of test suites broken by refactoring. We also implemented an automatic test-suite repairing tool and confirmed that it can repair tests broken by several types of refactoring.

研究分野：ソフトウェア工学

キーワード：テストスイート リファクタリング

1. 研究開始当初の背景

近年ではデジタルトランスフォーメーション (DX) が謳われ、ビジネスにおける競争優位性を確立するためにはソフトウェアが欠かせない存在である。特に、高品質のソフトウェアを、より短い期間でユーザーに提供するために、Agile や DevOps 等の開発手法を採用するプロジェクトが増加している。多くのプロジェクトに共通して行われるプラクティスとして、継続的インテグレーション (CI) が存在する。CI では、ソフトウェア変更の度にテストを自動実行し、常にリリース可能状態を維持することで、ソフトウェアをユーザーに提供するまでの間隔 (リリースサイクル) の短縮化を実現している。

リリースサイクルの短縮化が行われてもなお、ユーザーのソフトウェア品質への高い要求は変わることはない。従来、ソフトウェアの品質を向上させるため様々な方法が模索されてきた。最も代表的な方法が、リファクタリングである。大規模なリファクタリングはソフトウェア品質を劇的に向上させるが、リファクタリングを実施したメソッドを利用する他の箇所にも広範囲に影響が及ぶため、テストや不具合修正に長期間を要する。そのため、リリースサイクル短縮化の状況下での大規模リファクタリングは現実的でない。CI を採用するプロジェクトでは小さなリファクタリングを継続的に行うことが提唱されている。

リファクタリングの対象を小さくしてもなお、短いリリースサイクルでリファクタリングを実施することは容易ではない。我々の事前調査では、リリースサイクルの短縮後にリファクタリングの回数が減少することを観察している。また、既存研究[1]では、継続的なリファクタリングを阻害する要因として、リファクタリングにより (1) かえって不具合を埋め込んでしまう; (2) テストスイート (テストコードの集合) を破壊してしまう; 等が述べられている。要因 (1) に対する支援として、安全なリファクタリング方法などが従来盛んに研究されている。しかし、後者は、テストスイート修正作業により開発中断およびリファクタリング前後でテストの検証内容が変化する等の問題が発生するにも関わらず、これらのサポート (テストスイート破壊の検出や修正支援) はほとんど行われていない状況である。

一般にリファクタリングは、「外部の振る舞いを変更せずに、内部ロジックを変更する」と定義され、テストは変更されないと考えられている。しかし実際には、外見上 (システムレベルの動作) では変化が確認できなくても、メソッドレベルの構造が変更することも多い。現状では、どのようなリファクタリングがテストスイートを破壊し、どのような修正が必要かも明らかになっていない。そこで、本研究課題では「どのようなリファクタリングがテストスイートを破壊するか」を明らかにし、得られた知見を基にテストスイートの破壊箇所の特定および自動修正技術の確立を目指す。

2. 研究の目的

本研究の目的はリファクタリングが及ぼすテストスイートへの影響調査およびテストスイートの修正箇所特定技術・自動修正技術の確立である。

3. 研究の方法

本研究では、以下の3つの研究を実施した。

[研究 1] 解析方法の確立

テストスイート破壊の原因となったリファクタリングを特定する技術を開発する。具体的にはテストメソッドの各行が、プロダクションコードに施されたりリファクタリングのうち、どのリファクタリングによる影響を受けたかを特定する。

[研究 2] 影響範囲の分析および破壊テストケース予測

オープンソースソフトウェア (OSS) 開発プロジェクトから開発履歴データを取得し、(1) どのようなリファクタリングがテストスイートを破壊するか? (2) テストスイートの修正にどのような変更が必要か? を分析する。

(1)では、リファクタリング前であるリビジョン X-1 のテストコードで、リファクタリング後であるリビジョン X のプロダクションコードをテストし、テストメソッドが動作しなくなるか確認する。

(2)では、破壊されたテストメソッドの修正に必要となった変更をリファクタリング種別ごとに定性的・定量的に分析する。また、リファクタリングに伴って、破壊されるテストケースや修正必要箇所（行）を予測するモデルを構築する。

[研究 3] テストスイート自動修正の可能性検証

破壊されたテストスイートを修正するために施された、開発者による変更を目視分類する。また、分類された修正パターンのうち、実現性が高くかつ重要な修正パターンを対象を絞り、テストスイートの自動修正を試みる。例えば、返り値の型を変更するリファクタリングを行われたとき、`assertEquals(1, value)`と `assertEquals(1.0, value)`への変更が想定されるが、検証値としてはほとんど同意であるため、影響箇所が特定できれば修正可能と考えられる

4. 研究成果

[研究 1] 解析方法の確立

テストスイート破壊の原因となったリファクタリングを特定する技術を開発した。具体的にはテストメソッドにおけるそれぞれの変更行が、プロダクションコードに施されたリファクタリングのうち、どのリファクタリングによる影響のものかを特定した。

また、8件のオープンソースソフトウェア(OSS)開発プロジェクトから製品開発履歴データを取得し、合計 615,196 件のテストメソッドを実行することで、リファクタリングによって影響を受けるテストメソッドがどの程度であるかを分析した。具体的には、(1)テストケースを破壊するリファクタリングは何か? (2)テストスイートの修正にどの程度の変更行数が必要か?を明らかにした。

(1)では、リビジョン(X-1)のテストコードで、リビジョン X のプロダクトコード(リファクタリング後)をテストし、どの程度のテストメソッド(テストケース)が動作しなくなるか確認した。調査の結果、リファクタリングでは多くのテストメソッドが影響を受けるものの、実際に破壊されるテストメソッドは、全体の 2.5%程度であった。

(2)では、リファクタリングにより破壊されたテストコードの修正に必要となった行数を計測した。調査の結果、一部リファクタリング(メソッド引数の追加や返り値の型の変更)では、大きな修正が必要となることがわかった。

なお、これらを纏めた結果をソフトウェアメンテナンスに関するトップカンファレンスである ICSME にフルペーパーとして投稿し、採択された。

[研究 2] 影響範囲の分析、破壊テストケース予測および修正必要箇所予測モデルの構築および検証

リファクタリングに伴って破壊されるテストケースや補修必要箇所を予測するモデルを構築した。予測モデルを構築するにあたって、プロダクションコードから様々なメトリクス(リファクタリングの種別、対象クラスの規模数、変更行数等)を取得した。その後、様々な予測アルゴリズム(ランダムフォレスト等)を試行しながら、最も精度の良いアルゴリズムとメトリクスを検討した。

また、破壊されたテストを修復するために自動修正モデルを試作した。試作モデルの初期評価を纏め、ソフトウェア工学におけるトップカンファレンス(CORE A)のひとつである SANER に投稿した結果、論文が採録された。

[研究 3] テストスイート自動修正の可能性検証

提案アプローチの実用性を評価するために、オープンソースソフトウェアプロジェクトからソフトウェア開発データを取得し、リファクタリングによって破壊されたテストスイートの補修工数と予測された工数の違いを評価した。また、テストスイートの自動修正ツールを作成し、テストスイートを破壊しやすい ADD PARAMETERS リファクタリングにより破壊されたテストを修復できることを確認した。

<引用文献>

[1] C. Vassallo, F. Palomba and H. C. Gall, "Continuous Refactoring in CI: A Preliminary Study on the Perceived Advantages and Barriers," In Proc. of ICSME 2018, pp. 564-568, 2018.

5. 主な発表論文等

〔雑誌論文〕 計3件（うち査読付論文 3件/うち国際共著 2件/うちオープンアクセス 0件）

| | |
|---|-----------------|
| 1. 著者名 Hiroki Kuramoto, Dong Wang, Masanari Kondo, Yutaro Kashiwa, Yasutaka Kame, Naoyasu Ubayashi | 4. 巻 - |
| 2. 論文標題 Understanding the Characteristics and the Role of Visual Issue Reports | 5. 発行年 2024年 |
| 3. 雑誌名 Empirical Software Engineering | 6. 最初と最後の頁 - |
| 掲載論文のDOI（デジタルオブジェクト識別子） なし | 査読の有無 有 |
| オープンアクセス オープンアクセスではない、又はオープンアクセスが困難 | 国際共著 - |

| | |
|---|--------------------|
| 1. 著者名 Nourry Olivier, Kashiwa Yutaro, Lin Bin, Bavota Gabriele, Lanza Michele, Kamei Yasutaka | 4. 巻 33 |
| 2. 論文標題 The Human Side of Fuzzing: Challenges Faced by Developers during Fuzzing Activities | 5. 発行年 2023年 |
| 3. 雑誌名 ACM Transactions on Software Engineering and Methodology | 6. 最初と最後の頁 1~26 |
| 掲載論文のDOI（デジタルオブジェクト識別子） 10.1145/3611668 | 査読の有無 有 |
| オープンアクセス オープンアクセスではない、又はオープンアクセスが困難 | 国際共著 該当する |

| | |
|--|-----------------------------|
| 1. 著者名 Kashiwa Yutaro, Nishikawa Ryoma, Kamei Yasutaka, Kondo Masanari, Shihab Emad, Sato Ryosuke, Ubayashi Naoyasu | 4. 巻 146 |
| 2. 論文標題 An empirical study on self-admitted technical debt in modern code review | 5. 発行年 2022年 |
| 3. 雑誌名 Information and Software Technology | 6. 最初と最後の頁 106855~106855 |
| 掲載論文のDOI（デジタルオブジェクト識別子） 10.1016/j.infsof.2022.106855 | 査読の有無 有 |
| オープンアクセス オープンアクセスではない、又はオープンアクセスが困難 | 国際共著 該当する |

〔学会発表〕 計8件（うち招待講演 0件/うち国際学会 8件）

| |
|--|
| 1. 発表者名 Suwanachote Nabhan, Pornmaneerattanatri Soratouch, Kashiwa Yutaro, Ichikawa Kohei, Leelaprute Pattara, Rungsawang Arnon, Manaskasemsak Budit, Iida Hajimu |
| 2. 発表標題 A Pilot Study of Testing Infrastructure as Code for Cloud Systems |
| 3. 学会等名 The 30th Asia-Pacific Software Engineering Conference (APSEC) (国際学会) |
| 4. 発表年 2023年 |

| |
|--|
| 1. 発表者名 Fujita Shun, Kashiwa Yutaro, Lin Bin, Iida Hajimu |
| 2. 発表標題 An Empirical Study on the Use of Snapshot Testing |
| 3. 学会等名 2023 IEEE International Conference on Software Maintenance and Evolution (ICSME) (国際学会) |
| 4. 発表年 2023年 |

| |
|---|
| 1. 発表者名 Issei Morita, Yutaro Kashiwa, Masanari Kondo, Jeongju Sohn, Shane McIntosh, Yasutaka Kamei, Naoyasu Ubayashi |
| 2. 発表標題 TraceJIT: Evaluating the Impact of Behavioral Code Change on Just-In-Time Defect Prediction |
| 3. 学会等名 The 31th IEEE International Conference on Software Analysis, Evolution and Reengineering (国際学会) |
| 4. 発表年 2023年 |

| |
|--|
| 1. 発表者名 Kiyosuke Yamate, Masanari Kondo, Yutaro Kashiwa, Yasutaka Kamei, Naoyasu Ubayashi |
| 2. 発表標題 Hey APR! Integrate Our Fault Localization Skill: Toward Better Automated Program Repair |
| 3. 学会等名 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC) (国際学会) |
| 4. 発表年 2022年 |

| |
|--|
| 1. 発表者名 Olivier Nourry, Yutaro Kashiwa, Bin Lin, Gabriele Bavota, Michele Lanza, Yasutaka Kamei |
| 2. 発表標題 AIP: Scalable and Reproducible Execution Traces in Energy Studies on Mobile Devices |
| 3. 学会等名 2022 IEEE International Conference on Software Maintenance and Evolution (ICSME) (国際学会) |
| 4. 発表年 2022年 |

| |
|---|
| 1. 発表者名 Hiroki Kuramoto, Masanari Kondo, Yutaro Kashiwa, Yuta Ishimoto, Kaze Shindo, Yasutaka Kamei, Naoyasu Ubayashi: |
| 2. 発表標題 Do visual issue reports help developers fix bugs?: a preliminary study of using videos and images to report issues on GitHub |
| 3. 学会等名 30th IEEE/ACM International Conference on Program Comprehension (ICPC) (国際学会) |
| 4. 発表年 2022年 |

| |
|--|
| 1. 発表者名 Daisuke Fukumoto, Yutaro Kashiwa, Toshiki Hirao, Kenji Fujiwara, Hajimu Iida |
| 2. 発表標題 Empirical Investigation on the Performance of Domain Adaptation for T5 Code Completion |
| 3. 学会等名 30th IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER) (国際学会) |
| 4. 発表年 2023年 |

| |
|--|
| 1. 発表者名 Yutaro Kashiwa, Kazuki Shimizu, Bin Lin, Gabriele Bavota, Michele Lanza, Yasutaka Kamei, Naoyasu Ubayashi |
| 2. 発表標題 Does Refactoring Break Tests and to What Extent? |
| 3. 学会等名 IEEE International Conference on Software Maintenance and Evolution (国際学会) |
| 4. 発表年 2021年 |

〔図書〕 計0件

〔産業財産権〕

〔その他〕

-

6. 研究組織

| 氏名 (ローマ字氏名) (研究者番号) | 所属研究機関・部局・職 (機関番号) | 備考 |
|---------------------------|-----------------------|----|
| | | |

7. 科研費を使用して開催した国際研究集会

〔国際研究集会〕 計0件

8 . 本研究に関連して実施した国際共同研究の実施状況

| 共同研究相手国 | 相手方研究機関 | | | |
|---------|------------------------------------|---------------------|--|--|
| カナダ | Concordia University | Waterloo University | | |
| スイス | Universita della Svizzera italiana | | | |
| オランダ | Radboud University | | | |
| スイス | Universita della Svizzera italiana | | | |
| タイ | Kasetsart University | | | |
| ルクセンブルク | Luxembourg University | | | |