

令和 3 年 6 月 15 日現在

機関番号：14701

研究種目：基盤研究(C) (一般)

研究期間：2018～2020

課題番号：18K11243

研究課題名(和文)モダンコードレビューにおける欠陥見逃しの削減手法の構築

研究課題名(英文)Building a Method for Reducing Missed Defects in Modern Code Review

研究代表者

大平 雅雄(Ohira, Masao)

和歌山大学・システム工学部・准教授

研究者番号：70379600

交付決定額(研究期間全体)：(直接経費) 3,300,000円

研究成果の概要(和文)：本研究の目的は、High Impact Defects (HID) の混入をモダンコードレビューの段階で防ぐ手段を提供することである。本研究課題では以下の研究目標を設定し研究を実施した。
(研究目標1) HIDの見逃しレビューの実態調査：HID混入時とHID発見後のコードレビューを比較分析し、HIDの見逃し原因と除去方法を明らかにした。(研究目標2) HID混入判別モデルの構築と評価：コードレビュー開始時にHIDに関係するコードかどうかを判別するためのモデルを構築した。(研究目標3) HID除去方法の知識ベース化：HIDを素早く、かつ、安全に除去するのを支援する知識ベースを構築した。

研究成果の学術的意義や社会的意義

近年普及しつつあるモダンコードレビューにおいて重大な欠陥(HID)を見逃す根本原因については不明なままであった。本研究では、HIDの見逃し原因と除去方法を明らかにするとともに、HIDの混入を判別するモデルを構築した。さらに、HID除去方法に関する知識ベースを構築した。信頼性の高いモダンコードレビューの実施を支援するという観点で、研究成果の社会的意義は大きいと考えている。

研究成果の概要(英文)：The goal of this research is to provide a means to prevent the inclusion of High Impact Defects (HIDs) in the modern code review stage. We set the following research goals and conducted the research in this project.

(Research Goal 1) Investigation of missed HID reviews: We compared and analyzed code reviews when HIDs were introduced in and after HIDs were found, and clarified the causes of missed HIDs and how to remove them. (Research Goal 2) Construction and evaluation of a prediction model for HID contamination: We constructed a model to determine whether a code is related to HID or not at the beginning of code review. (Research Goal 3) Development of a knowledge base for HID removal methods: We developed a knowledge base to support quick and safe removal of HIDs.

研究分野：ソフトウェア工学

キーワード：モダンコードレビュー High Impact Defects ソフトウェア品質 ソフトウェア保守 欠陥除去

様式 C - 19、F - 19 - 1、Z - 19 (共通)

1. 研究開始当初の背景

ソースコードレビュー(以降、コードレビューと呼ぶ)は、プログラムを動作させて検証するソフトウェアテストよりも前の段階から実施できるため、修正コスト(手戻りコスト)の低減に効果的であるとされている。コードレビューでは、「第三者にとって理解・利用しやすくコードが記述されているか?」から「信頼性・安全性に大きな影響を与える欠陥は存在しないか?」まで、様々な観点からソースコードの品質が検証され問題点が指摘される。ベテラン技術者によるレビュー指摘には、ソースコードの品質の向上に関して示唆に富む情報が多く含まれるため、コードレビューは若手技術者への教育手段としても活用されている。

これまでの伝統的なコードレビューは数名の担当者によって対面会議で実施するのが一般的であったが、オープンソースプロジェクトあるいは Microsoft や Google のような先進的な IT 企業では、効率化を目的としてコードレビュー支援システム(例えば、Review Board, Gerrit など)を用いてオンラインでのコードレビューを実施しており、モダンコードレビューとして広く認知されつつある。

コードレビュー支援システムには、(a)過去に行った全てのレビュー指摘(不備の理由)と(b)レビュー指摘前後のソースコード(コード変更差分)が記録されている。また、不具合修正のためにコードに変更を加える際には、(c)レビューと該当不具合の紐付け情報が記録される。オープンソースプロジェクトではこれら全ての情報が公開されているため、レビュー指摘・コード変更差分・紐付けられた不具合情報を抽出し分析することができる。そのため、有名なオープンソースプロジェクトのベテラン開発者がソフトウェアシステムの信頼性・安全性を確保するために行っているコード記述の方針や独自の工夫を明らかにできる可能性が高い。

実際、モダンコードレビューにおけるレビュー品質に関して上記の情報に基づいて定量的に調査した研究¹²³もすでによくつか報告されている。しかしながら、コードレビューにおいて見逃した欠陥の数とレビューアの数やレビューに費やした時間との関係について定量的に相関を示した程度の表層的な分析にとどまっております。かつ、見逃した欠陥の種類や影響の大きさを区別していない(タイプミス等によって生じる軽微な欠陥かどうかも区別していない)ため、「モダンコードレビューではどのような種類の欠陥が見逃されやすいのか?」「重大な欠陥も同様に見逃されるのか?」「モダンコードレビューで欠陥を見逃す根本原因は何か?」といった本質的な問いに未だ答えがない状態にあった。

そこで本研究課題では、モダンコードレビューにおいて見逃されやすい欠陥の中でもソフトウェアシステムの信頼性および安全性に大きな影響を与える欠陥(High Impact Defects)に焦点を当て、High Impact Defectsが見逃される根本原因の解明を試みるとともに、モダンコードレビューの段階で High Impact Defects の混入を防ぐ(適切に除去する)手段を提供することを目指した。

2. 研究の目的

本研究の目的は、ソフトウェアシステムの信頼性および安全性に大きな影響を与える High Impact Defects の混入をモダンコードレビューの段階で防ぐ手段を提供することである。そのために本研究課題では、以下の研究目標を設定し研究を実施した。

(研究目標1) High Impact Defects の見逃しレビューの実態調査

High Impact Defects の混入を見逃した際のコードレビューと High Impact Defects 発見後のコード修正に伴うコードレビューとの比較分析を通じて、High Impact Defects の見逃し原因(混入原因)と除去方法を抽出する。

(研究目標2) High Impact Defects 混入判別モデルの構築と評価

抽出した High Impact Defects の混入原因を機械学習することで、High Impact Defects の混入を予測する(すなわちコードレビューの対象が High Impact Defects を含むコードであることを検知する)モデルを構築する。

(研究目標3) High Impact Defects 除去方法の知識ベース化

検出した High Impact Defects をコードベースに混入させないように、(研究目標1)で抽出した High Impact Defects の除去方法を知識ベース化し素早く参照できるようにする。

3. 研究の方法

前述の研究目標をそれぞれ達成するために、本研究では次の方法で研究を実施した。

(1) 研究目標1: High Impact Defects の見逃しレビューの実態調査

High Impact Defects の混入を見逃した際のコードレビューと High Impact Defects 発見後のコード修正に伴うコードレビューとの比較分析(図1)を通じて、High Impact Defects の見逃し原因(混入原因)と除去方法を明らかにする。大規模なオープンソースプロジェクトを数件選択

¹ S. McIntosh and et al., "The Impact of Code Review Coverage and Code Review Participation on Software Quality: A Case Study of the Qt, VTK, and ITK Projects," Proc. of MSR2014, pp.192-201, 2014

² P. Thongtanunam and et al., "Investigating code review practices in defective files: an empirical study of the Qt system," Proc. of MSR2015, pp.168-179, 2015

³ O. Kononenko and et al., "Investigating Code Review Quality: Do People and Participation Matter?," Proc. of ICSME2015, pp.111-120, 2015

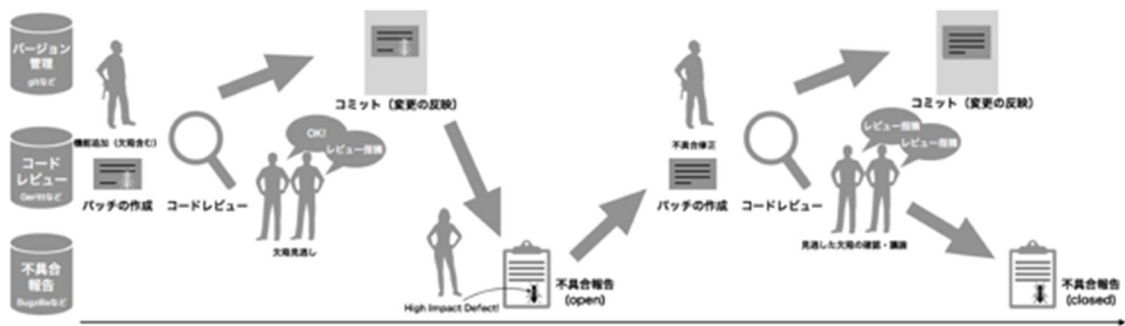


図 1 欠陥見逃しレビューと不具合修正レビューの全体像

し、上記の手順に従って、各プロジェクトにおいて数百件程度の欠陥見逃しレビューを実態調査する。各プロジェクト数百件程度と数が少ないのは本研究が対象とする High Impact Defects が報告された全欠陥に比べ少ないためである。

(2) 研究目標 2: High Impact Defects 混入予測モデルの構築と評価

コードレビュー開始時に High Impact Defects に関係するコードかどうかを判別するためには不具合修正時と機能追加時の状況に対応するために 2 つのモデルが必要となる。後者は判別精度が現時点では不明なため評価実験を行う。また、必要に応じてモデルの改良も行う。

(3) 研究目標 3: High Impact Defects の除去方法を知識ベース化

(研究目標 1) で得られた情報を整理することで、今後 High Impact Defects が検知された場合に素早く、かつ、安全に除去するための知識ベースとする。

4. 研究成果

(1) 2018 年度は(研究目標 1)に取り組んだ。具体的には以下の手順で調査を実施した。

- (1-1) 研究代表者の過去の研究課題の成果に基づいて、OSS プロジェクトの不具合管理システムから High Impact Defects が報告されている不具合票の ID を抽出。
- (1-2) 不具合 ID と紐付いているレビュー ID を特定し、コード修正に伴うレビュー指摘およびコード差分を抽出。
- (1-3) 不具合 ID と紐付いているコミット ID を特定し、SZZ アルゴリズムに基づいて該当する High Impact Defects を混入したコミット ID を特定。
- (1-4) High Impact Defects を混入したコミット ID に基づいて、High Impact Defects の混入を見逃す原因となったレビュー ID を特定し、レビュー指摘およびコード差分を抽出。
- (1-5) (1-2)および(1-4)のレビュー指摘及びコード差分を比較し、High Impact Defects の見逃し原因を特定。

本手順を複数の OSS プロジェクトに対して実施しそれぞれのプロジェクトにおける対処方法(除去方法)を目視により調査した。

(2) 2019 年度は(研究目標 2)に取り組んだ。具体的には以下の研究を実施した。

- (2-1) コードレビューが不具合修正を対象として行われる場合：不具合票をベースに High Impact Defects が否かを判別すれば良いので研究代表者の過去の研究課題⁵で構築した判別モデルを利用した。
- (2-2) コードレビューが機能追加等を対象として行われる場合：不具合票とコードレビューは紐付いていないので、当該コード変更が High Impact Defects をもたらすかどうかを予測する判別モデル(コード差分情報をベースにした判別モデル)を(研究目標 1)の成果に基づいて構築した。

(3) 2020 年度は(研究目標 3)に取り組んだ。具体的には以下の研究を実施した。

(研究目標 1) の成果からオープンソース開発に携わるベテランの開発者が「High Impact Defects をコードレビューにおいて何故見逃したのか?」と「High Impact Defects をどのような方針に基づいて除去したのか?」についての情報が得られた。これらの情報を High Impact Defects の種類別、見逃し原因別、対処方法別に整理することで、今後 High Impact Defects が検知された場合に素早く、かつ、安全に除去するための知識ベースを構築した。

⁴ M. Ohira, et al. "A Dataset of High Impact Bugs: Manually-Classified Issue Reports", Proc. of MSR2015, pp.518-521 (2015)

⁵ 科研費 基盤(C), 派生開発における異常検知のためのプロアクティブマイニング手法の構築(研究代表者), 課題番号 15K00101, H27-H29

5. 主な発表論文等

〔雑誌論文〕 計2件（うち査読付論文 2件／うち国際共著 0件／うちオープンアクセス 1件）

1. 著者名 KASHIWA Yutaro, OHIRA Masao	4. 巻 E103.D
2. 論文標題 A Release-Aware Bug Triaging Method Considering Developers' Bug-Fixing Loads	5. 発行年 2020年
3. 雑誌名 IEICE Transactions on Information and Systems	6. 最初と最後の頁 348 ~ 362
掲載論文のDOI (デジタルオブジェクト識別子) 10.1587/transinf.2019EDP7152	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 Higashi Yunosuke, Ohira Masao, Kashiwa Yutaro, Manabe Yuki	4. 巻 27
2. 論文標題 Hierarchical Clustering of OSS License Statements toward Automatic Generation of License Rules	5. 発行年 2019年
3. 雑誌名 Journal of Information Processing	6. 最初と最後の頁 42 ~ 50
掲載論文のDOI (デジタルオブジェクト識別子) 10.2197/ipsjjip.27.42	査読の有無 有
オープンアクセス オープンアクセスとしている（また、その予定である）	国際共著 -

〔学会発表〕 計12件（うち招待講演 1件／うち国際学会 4件）

1. 発表者名 Katsunori Fukui, Tomoki Miyazaki, Masao Ohira
2. 発表標題 Suggesting Questions that Match Each User 's Expertise in Community Question and Answering Services
3. 学会等名 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2019) (国際学会)
4. 発表年 2019年

1. 発表者名 Yusuke Kukita, Kojiro Noguchi, Masao Ohira
2. 発表標題 Code Clone Tracer (CCT): a Tracking Tool for Analyzing Human and Social Factors in Creating and Reusing Code Clones
3. 学会等名 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2019) (国際学会)
4. 発表年 2019年

1. 発表者名 Katsunori Fukui, Tomoki Miyazaki, Masao Ohira
2. 発表標題 A Bot for Suggesting Questions that Match Each User ' s Expertise
3. 学会等名 2019 IEEE/ACM 1st International Workshop on Bots in Software Engineering (BotSE '19) (国際学会)
4. 発表年 2019年

1. 発表者名 Yutaro Kashiwa, Akinori Ihara, Masao Ohira
2. 発表標題 What Are the Perception Gaps between Floss Developers and Se Researchers?
3. 学会等名 The 15th International Conference on Open Source Systems (国際学会)
4. 発表年 2019年

1. 発表者名 野口 耕二郎, 大平 雅雄
2. 発表標題 コードクローンへの欠陥混入防止に向けた欠陥混入クローンの特徴分析
3. 学会等名 ソフトウェア・シンポジウム 2019 in 熊本
4. 発表年 2019年

1. 発表者名 大東 誠弥, 福井 克法, 宮崎 智己, 大平 雅雄
2. 発表標題 日本語非機能要件の自動分類における教師あり学習アルゴリズムの評価
3. 学会等名 ソフトウェア・シンポジウム 2019 in 熊本
4. 発表年 2019年

1. 発表者名 宮崎 智己, 伊原 彰紀, 大平 雅雄, 東 裕之輔, 山谷 陽亮
2. 発表標題 Politeness 分析に基づくOSS 開発者の活動継続性の理解
3. 学会等名 情報処理学会ソフトウェアエンジニアリングシンポジウム2018
4. 発表年 2018年

1. 発表者名 大平 雅雄
2. 発表標題 ソフトウェアリポジトリマイニングの研究動向とソフトウェア工学におけるAI技術の活用
3. 学会等名 日本SPIコンソーシアム (JASPIC) プロセス改善への機械学習応用分科会 (招待講演)
4. 発表年 2018年

1. 発表者名 宮崎 智己, 大平 雅雄
2. 発表標題 非機能要件の詳細な自動分類手法の構築に向けて: セキュリティ要件の分類
3. 学会等名 情報処理学会ソフトウェアエンジニアリングシンポジウム2018
4. 発表年 2018年

1. 発表者名 吉富 楓雅, 宮崎 智己, 柏 祐太郎, 大平 雅雄
2. 発表標題 ソフトウェアテストにおけるSilent Horrorsの分析へ向けて
3. 学会等名 情報処理学会ソフトウェアエンジニアリングシンポジウム2018
4. 発表年 2018年

1. 発表者名 福井 克法, 大平 雅雄, 川辺 義勝
2. 発表標題 トピックモデリングに基づく開発者検索手法の構築へ向けて
3. 学会等名 ソフトウェア技術者協会ソフトウェア・シンポジウム2018 in 札幌
4. 発表年 2018年

1. 発表者名 山崎 大輝, 大平 雅雄, 伊原 彰紀, 柏 祐太郎, 宮崎 智己
2. 発表標題 開発者の活動量の経時的変化がコミッター候補者予測に与える影響の分析
3. 学会等名 電子情報通信学会ソフトウェアサイエンス研究会
4. 発表年 2018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