

REreview: ソフトウェア開発 PBL における コードレビュー学習支援システム

浅野耀介¹ 古川貴一² 山田侑樹² 樋山淳雄¹

概要: ソフトウェア開発では、品質の高いコードを求められ、コードレビューという活動が行われている。Project Based Learning に参加する学習者は品質の高いコードに関する知識が少なく、コードレビューを行うことは難しい。本研究では、経験の浅いレビューアに対して、品質の低いコードのパターンを提示、ブランチごとにレビューの振り返りの場を提供するシステム REreview を開発して、コードをレビューする力を身につけていくことの支援を目指す。

REreview: A Learning Support System for Code Reviewers in Software Development PBL

YOUSUKE ASANO^{†1} KIICHI FURUKAWA^{†2}
YUKI YAMADA^{†2} ATSUO HAZEYAMA^{†1}

Abstract: In the software development process, high quality code is required, and code review is performed. Learners participating in Project Based Learning have little knowledge of quality code, making it difficult to perform code reviews. This study aims to support inexperienced reviewers to acquire the ability to review code by providing REreview, a system that presents patterns of low-quality code and provides a place for reviewing at each branch.

1. はじめに

ソフトウェア開発では、品質の高いコードが求められ、コードレビューが行われている。コードレビューは、欠陥を見つけるだけでなく、システムに関する知識の共有もできる[1][2]。このことから Project Based Learning (以下 PBL) でもコードレビューを行うべきである。しかし、実務経験のない学習者がコードレビューを行うのは難しい [3]。

本研究では、PBL における学習者間のコードレビューに焦点を当てる。レビューアとなる学習者に対して、品質解析ツールの指摘項目を品質の低いコードのパターンとして事前に提示し、学習者にコードレビューをさせ、コードの品質解析ツールの結果を用いた振り返りを行うシステム REreview を開発し、コードレビューの技術の育成を目指す。

2. 関連研究

本節では、コードレビューに関する先行研究を紹介し、本研究の位置づけを示す。

Krushe ら[4]は、PBL にてブランチベースのコードレビューを実践した。レビューアは経験豊富な学生が務めた。ブランチベースのコードレビューが学生に対するアンケートで高い評価を得たことを報告した。

Lu ら[5]は、学習者のプロジェクトに SonarQube によるコード解析を行う環境を構築した。SonarQube が学習者の悪い習慣を認識させて、学習者のコードの質を向上させることを示した。

木崎ら[3]は、アジャイル開発 PBL において、コード解析

ツール (PMD)による解析結果をレビューアに提示することで、経験の少ない学習者に対して、レビューの支援を行う方法を提案している。

本研究では、ブランチベースのコードレビューワークフローを実施し、レビューアの下承ではじめて変更をマージできるものとする。しかし多くの PBL では、経験豊富な学習者は少ない。そのために、レビューアとなる学習者には、何らかの支援が必要である。そこで Lu らの研究を参考に SonarQube と同等の機能を持った SonarCloud[6]を利用する。しかし、SonarCloud の解析結果をレビュー前にレビューアに提示してしまうと、レビューアはコードの欠陥を探す必要がないので、レビューアとなる学習者のコードレビューの学習機会を奪ってしまう。木崎らの研究とは解析結果をフィードバックするタイミングが異なり、SonarCloud の解析結果をレビュー終了後に提示し、レビューを振り返る機会を作ることによってレビュー遂行能力を向上させることを目指す。

3. 本研究のアプローチ

本研究では、レビューアの学習のために、振り返りを支援するシステム REreview を開発する。本節では、REreview の目的と満たすべき要件について説明し、導入前の準備、REreview での振り返りを行うまでの流れ、レビューアの振り返りについて説明する。

3.1 REreview の目的と満たすべき要件

REreview は振り返りを通じて、コードの欠陥を発見し、

¹ 東京学芸大学

² 東京学芸大学大学院

指摘する技術を育成することを目的としている。その目的を果たすべく、「レビューは自身のレビューの質を客観的に評価できる」という要件を満たすように開発する。

3.2 RReview 導入前の準備

RReview の導入前に、過去に本学の PBL で作成されたコードを SonarCloud で解析し、どのような欠陥が発生しやすいのかの情報を収集し、レビューアに対してコードをレビューする際の着眼点の一つとして提示する。この提示によって、コードの欠陥に関する事前知識を与える。

3.3 RReview で振り返りを行うまでの流れ

レビューアが振り返りを行うまでの流れを図 1 に基づき説明する。

3.3.1 コードの欠陥のデータの収集

ブランチベースの開発では、開発者はブランチを作成し、機能の追加や修正を行う。コーディングを終えると、リモートリポジトリに変更をプッシュ、プルリクエストを作成し、レビューアに対してコードレビューの申請を行う (図中①)。レビューアはレビューを行い、欠陥があればコードの修正を依頼する (図中③)。開発者はレビュー結果をもとに、コードの修正を行う (図中④)。SonarCloud は①、④が行われるたびにそのブランチで加えられた変更を解析する (図中②、⑤)。

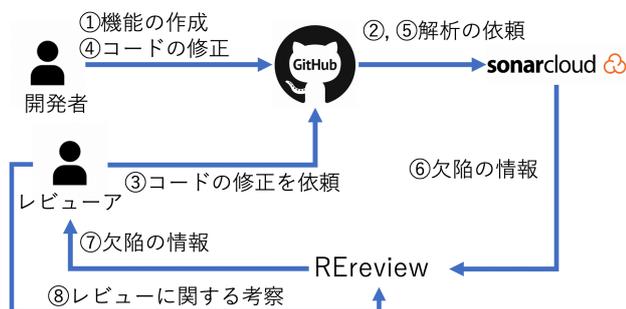


図 1 振り返りを行うまでの流れ

3.3.2 レビューアへのフィードバック

RReview は、SonarCloud から欠陥の情報を取得 (図中⑥) し、レビューアに対してレビューを実施したブランチに存在した欠陥の情報をフィードバックとして返す (図中⑦)。フィードバックは、レビューを行ったブランチがマージされたタイミングでなされる。

要件を満たすべく、レビューアは以下の情報を確認することができる。

- ・ブランチ名
- ・修正完了した欠陥, 完了していない欠陥
- ・欠陥の種類, 欠陥のあるコードの位置, 解消する方法

3.4 レビューアの振り返り

レビューアが行う 2 つの振り返りについて説明する。

3.4.1 ブランチごとの振り返り

レビューアは、3.3.2 で受け取ったフィードバックをもとに、レビューしたブランチに存在していた欠陥を把握する

ことで、次にレビューするブランチでは、欠陥が起きてても発見、指摘できるようにするために振り返りを行う。

レビューアはなぜ欠陥が修正できなかったのかを考察し、RReview に考察内容を記述する。

3.4.2 過去の実績確認

開発者は、機能の開発に比べてコードレビューに時間を使いたくなく、レビューを行うことにインセンティブが必要になる[4]。学習者にレビューを行うインセンティブを与えるために、レビュー遂行能力がついた実感を与えたい。

RReview では、レビューアが実績として修正できた欠陥の割合の推移をグラフで確認することができるようにする。これは要件の一助にもなる。

4. 妥当性への脅威

本節では、本研究の限界について記述する。

SonarCloud が検出する項目には限界がある。要求仕様と異なる実装を、検出することができない。SonarCloud が指摘するのは、保守性を低下させる code smell や潜在的なバグである。SonarCloud が検出する欠陥をすべて指摘するレビューを行えば、完全なレビューであるというわけではないが、コードレビューの質が向上することは期待される。

本学で行われる PBL に参加する学習者は、コード開発者にもレビューアにもなる。事前に提示される項目や、コードレビューの振り返りによって得られる知見により、自身で欠陥に気づき、修正できる可能性もある。その修正が、レビューによって修正されたものなのか、それとも自分で気づいたものなのかは、判断することができない。

5. まとめ

学習者がコードレビューを行うことが難しいという問題に対し、レビューアのレビューの振り返りを支援するシステム RReview を提案した。

本システムによる支援を通して、学習者がレビューの振り返りを通じて、コードの欠陥を発見し、指摘する能力を向上させることを期待する。

謝辞 本研究の一部は科研費 20K12089 の助成の下で行われた。記して謝意を表す。

参考文献

- [1] Bacchelli, A. and Bird, C. Expectations, Outcomes, and Challenges of Modern Code Review. Proc. ICSE. IEEE. 2013. pp. 712–721.
- [2] Rigby, P. and Bird, C. Convergent Contemporary Software Peer Review Practices. Proc. FSE. ACM. 2013. pp. 202–212.
- [3] 木崎悟, 田原康之, 大須賀昭彦. アジャイル型ソフトウェア開発 PBL における CCBP を拡張したコードレビュー支援環境の提案. IPJSJ SIG Technical Report. 2013. Vol.2013-CE-118. No.9 pp. 1-8.
- [4] Krusche, S. et al., Teaching Code Review Management using Branch Based Workflows. IEEE/ACM. 2016. pp. 384-393
- [5] Yao, L. et al., Improving students' programming quality with the continuous inspection process: a social coding perspective. Frontier of Computer Science. 2020. Vol.14. No.5. pp.1-18.
- [6] SonarCloud. <https://sonarcloud.io>. (参照 2020-7-8)