

コードレビューにおいて発見されるバグの特徴分析

西口 絢人[†] 門田 暁人[†] 上野 秀剛[‡] 松本 健一[†]

108名の学生を対象としたコードレビューの実験を行い、報告されるバグの傾向を分析した。その結果、仕様誤りに関連するバグは発見されにくい傾向があり、for文 while文、if文の条件式である分岐処理に関わるバグは多くの人が発見するといった傾向がみられた。

Characterizing Defects Found in Code Review

Kento Nishiguchi[†] Akito Monden[†] Hidetake Uwano[‡] Kenichi Matsumoto[†]

This paper presents an experiment of code review with 108 students, and characterizes defects found in the review. As a result, most of defects related to specification error were overlooked while defects related to control flow (e.g. for, while and if statements) were found by many students.

1. はじめに

ソフトウェア開発のコストパフォーマンス及び品質向上のためにバグは早期段階で取り除くことが重要である。また、早期段階でのバグ除去の手段の一つとしてソフトウェア成果物内の50~70%の誤りを発見できる[1]コードレビューは多くの開発現場で使用されている。

近年では、Checklist-Based Reading(CBR)を始めとした、様々なレビュー手法が提案されており、手法によるレビュー効率の影響を分析する実験も数多く行われている。

一方で、バグ発見率はレビュー手法の違いによる影響よりも、個人差による影響の方が大きく、手法間でのレビュー効率の差は1.2~1.5倍であるのに対し、個人差は3倍以上ある[2]。この個人差はプログラムの読解能力や理解戦略といった、個人の技術や経験の差によるものと考えられるが、明らかにはされていない。そのため、レビュー効率の個人差に着目した研究がされている。ただし、個々のバグの発見しやすさについては考慮されていない。教育の観点からは、発見されにくいバグの特徴や、その発見方法についての研究が求められているが、現状ではほとんど行われていない。

本研究では、直近では不具合の発見容易性と、その種類の関係を発見することを目的として、将来的には教育方針の作成支援を目的として、個人のコードレビューを集計、結果を分析する。

2. 本研究で取り組む RQ

本研究では以下のリサーチクエスト(RQ)について取り組む。現在はまだこれらのRQに答えられていないが、本稿では、RQ1への回答に向けた分析を行う。

[RQ1:発見されにくいバグの特徴は?]

発見されにくいバグの特徴が明らかとなれば、それらを発見するためのガイドラインの策定や開発者の教育に繋げることができると期待される。

[RQ2:発見バグの傾向に個人差はあるのか?]

同程度の能力の者でも、得意とするバグが異なる可能性がある。発見バグの傾向に基づいて個人の特徴を明らかにできれば、自分がどのタイプに属している、どのバグの発見が苦手かを明らかにでき、教育に繋げることができると期待される。

[RQ3:同時に発見されるバグの特徴は?]

同時に発見されやすいバグの特徴を明らかにできれば、一方を発見していて他方を発見していない場合に警告できると期待される。

3. 実験

3.1. レビュー対象のプログラム

レビュー対象は酒屋倉庫管理問題のプログラムである。C言語で記述されており、コード行数は604である。バグはデータ構造・データ操作に関するもの、戻り値や引数などの関数のインターフェースによるもの、条件・ループ制御に関わる判定によるもの、ロジックの誤

[†] 奈良先端科学技術大学院大学

[‡] 奈良工業高等専門学校

り(アルゴリズム, フラグの意味・条件, 境界値)の4種類をほぼ同数(合計31個)埋め込んだ。いずれも, コードを1行変更することで修正できるような小規模なバグである。

3.2. レビュー環境

前述のプログラムに対し, 奈良先端科学技術大学院大学情報科学研究科の博士前期課程の学生に対して, コードレビューを行う課題を実施した, これらは2008年と2012年の2回に分けられており, 2008年では38名, 2012年では70名の合計108名に対して実施された。実験にはソースコードのコピー, 仕様書, プログラムの概要説明文書をレビュー資料として配布した。概要説明書はプログラム中に定義されている各関数の概要が記述されている。

これらの文書を元に各被験者にコードレビューを実施してもらい, プログラムが正常に動作しない原因となる記述の誤りを探してもらった実験を行った。レビュー時間, 手法については指定していないが, プログラムの実行は認めなかった。

また, 課題として2008年では最低5個以上, 2012年では最低10個以上の不具合の報告を義務付けた。バグについては, 事前に埋め込んだバグ以外は存在しないものとし, それ以外の報告については対象としなかった。

4. 結果

図1に個人毎のバグの報告数の分布を示す。2008年は一人あたり平均8.8個の報告, 2012年は平均13.6個の報告, 全体では平均で11.9個のバグが報告された。

図2にバグ毎にそのバグを報告した人数の分布を示す。発見者が0人のバグが2個, 発見者が5人以下のバグは6個存在した。また, 被験者の半数以上が発見したバグは15個あり, 発見されるバグには偏りが存在することがわかった。

誰も発見しなかったバグは仕様自体に誤りであるケースであった。また, 報告者数が20人未満のバグの多くは, プログラム上は矛盾がないが仕様とは異なる動作をするというバグと, 例外発生に関係するバグであり, これらの要素については見逃されやすい傾向にあると考えられる。

一方, 発見者数が多いバグは, case文の条件, for文, while文, if文の初期値や終了条件の間違いを扱ったものであった。多くの被験者が分岐処理を重点的にレビューする傾向にあるのではないかと考えられる。

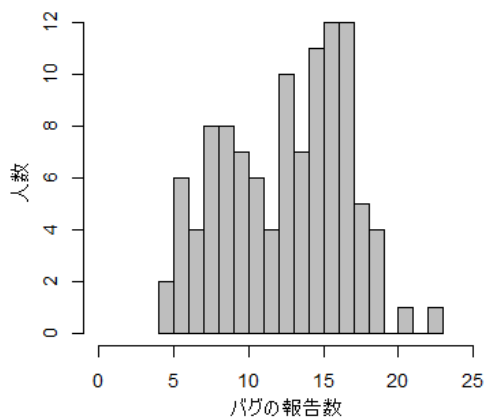


図1. 個人毎のバグ報告数の分布

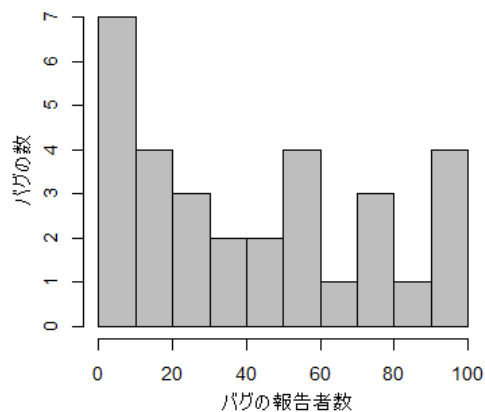


図2. バグ別の報告者数の分布

また, 発生条件が複雑なバグを発見している被験者は類似する簡単なバグを発見している事が多い傾向が見られた。

5. まとめ

本稿では, 108名の学生を対象としたコードレビュー実験の予備分析の結果を報告した。ワークショップでは, 今後の分析方法について議論したい。

参考文献

- [1] Karl E. Weigers (著), ピアレビュー, 日経BPソフトプレス, 2004.
- [2] Thelin, T., Andersson, C., Runeson, P., Dzamashvili-Fogelstrom, N, A replicated experiment of usage-based and checklist-based reading, Proc. METRICS2004, pp.246-256, 2004.