ソフトウェアテストにおける Silent Horrors の分析へ向けて

吉富 楓雅[†1]　宮崎 智己[†2]　柏 祐太郎[†2]　大平 雅雄[†1]

概要：テストコードに欠陥が混入している場合，プロダクトコードの欠陥を見逃してしまう Silent Horrors が発生する可能性がある．Silent Horrors の発生原因を明らかにすることを最終的な目標とし，本稿では，まずテストコードの欠陥の発生原因を明らかにするための分析を行う．

キーワード：Silent Horrors，テストコード，欠陥報告，REOPENED

1. はじめに

ソフトウェアテストは，ソースコードが期待通りに機能するか検証するために実施される．しかしながら，作成したテストコードには欠陥が混入する可能性があり，多くの開発者がテストコードはプロダクトコードよりも欠陥を含む可能性が高いと述べている[2][3]. テストコードに欠陥が混入すると，欠陥が見逃されたり，プロダクトコードに問題がないにもかかわらずテストが失敗することがある．

Vahabzadeh らは，テストコードの欠陥に関する分析を行っており，テストコードの欠陥が原因で生じる問題を以下の 2 つに分類している[1].

1. Silent Horrors : テストコードに欠陥があるためにテストが成功する．すなわちプロダクトコードの欠陥を見逃している．
2. False Alarms : テストコードに欠陥があるためにテストが失敗する．すなわちプロダクトコードに欠陥は混入していない．

Silent Horrors はプロダクトコードの欠陥を見逃してしまうため，システムの品質に重大な悪影響を及ぼす可能性がある．

既存研究では Silent Horrors の定量的な分析や発生した欠陥の種類については明らかにしているが，Silent Horrors の発生原因については明らかにしていない．

そこで，本研究では Silent Horrors の発生原因を明らかにすることを最終的な目標とする．しかしながら，Silent Horrors を特定するのは困難であるため，本稿では，まずテストコードの欠陥の発生原因を明らかにする．

2. 既存研究

Vahabzadeh らは，どの程度の頻度でテストコードに欠陥が発生するのかを調査するために，Apache プロジェクトの不具合管理システムとバージョン管理システムを対象に分析を行った．分析の結果， Apache の計 448 のサブプロジェクト中，211 プロジェクト(47%)で，テストに関する欠陥が発見された．

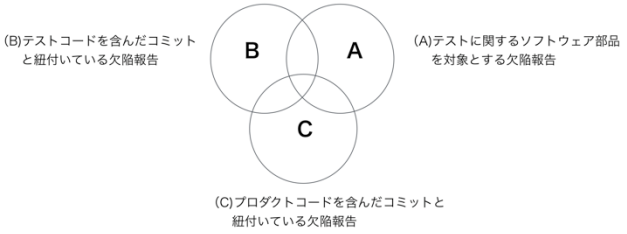

図 1：収集したデータ（[1]の図を改変）

既存研究では，欠陥報告を以下の 3 種類に分類している．各データのイメージを図 1 に示す．

A: テストに関するソフトウェア部品を対象とする欠陥報告
B: テストコードを含んだコミットと紐付いている欠陥報告
C: プロダクトコードを含んだコミットと紐付いている欠陥報告

プロダクトコードを含まない欠陥報告（（A ∪ B）- C）のうち，5,556 件のテストコードに関係する欠陥報告が抽出された．抽出した欠陥報告から手作業で抽出した 443 件の欠陥報告のうち，Silent Horrors は 15 件と全体の 3%であった．その内訳は，Silent Horrors は Assertion Fault と Missing Assertion であることを明らかにした．

既存研究では，Silent Horrors の定量的な分析や，Silent Horrors によって発生した欠陥の種類については調査されているが，Silent Horrors が発生した原因については明らかにされていない．そこで本研究では，Silent Horrors の発生原因を明らかにすることを目指す． しかしながら，全ての Silent Horrors を対象とするのは困難である．Silent Horrors はテストコードの欠陥によって引き起こされるため，本稿ではまず，テストコードの欠陥の発生原因を調査する．

†1 和歌山大学システム工学部
†2 和歌山大学大学院システム工学研究科

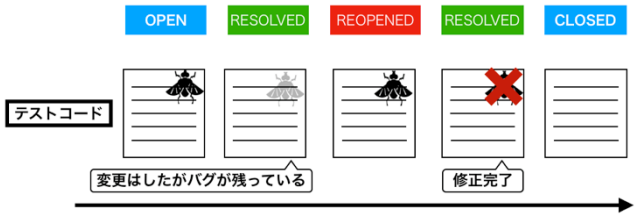

図 2：テストコードの修正例

3. 分析方法

3.1 分析対象

本稿では，図 1 の A に当たるテストに関するソフトウェア部品を対象とする欠陥報告のうち，CLOSED されており，かつ，REOPENED のタグ付けがされた欠陥報告を分析対象とする．REOPENED を用いる理由として，テストコードを修正する必要があった原因とテストコードの修正パッチについての議論があるため，テストコードの欠陥の発生原因を特定することが可能であると考えたためである．

REOPENED された欠陥に関するテストコードが Silent Horrors の発生原因であるかどうかを明らかにするために以下のリサーチクエスチョンに取り組む．

RQ1：REOPENED されたテストコードの欠陥の発生原因は何か？

RQ2：REOPENED されたテストコードの欠陥はどの程度の頻度で Silent Horrors を引き起こすのか？

3.2 対象プロジェクト

本稿では Apache Derby[4]を対象プロジェクトとする．Apache Derby は，Java で実装されているリレーショナルデータベース管理システムで，API として任意の Java データベースの結果に組み込み JDBC ドライバを提供する．

Apache Derby プロジェクトを対象プロジェクトとした理由は，既存研究[1]が分析した Apache Software Foundation のプロジェクトのうち，Apache Derby プロジェクトがテストコードに関する欠陥の件数が最も多かったこと，プロダクトコードに対して記述されたテストコードの比率が最も多かったことが挙げられる．

3.3 データの抽出

Apache Derby プロジェクトのテストコードの欠陥報告は，Apache JIRA[5]に保存されている．保存されている欠陥報告のうち，CLOSED されており，かつ，一度でも REOPENED されている欠陥報告を，タイプやステータスなどのタグを用いて絞り込みを行った（Type: Bug，Status: Closed，Component/s: Test）．その結果，全テストコードの欠陥報告は 3,496 件あり，このうち REOPENED とタグ付けされたテストコードの欠陥報告は 73 件であった．

3.4 目視による Silent Horrors の分析

3.3 節で抽出した 73 件の欠陥報告について，OPEN から CLOSED されるまでの全てのテストコードの差分と欠陥報告に寄せられたコメントを目視する．図 2 を例に，方法を説明する．まず，OPEN から CLOSED までのコメントを目視し，発生した欠陥について理解する．次に，報告された欠陥が発生した原因，修正するために行った変更，及び修正することができなかった変更について調査する．最後に，テストコードの差分を目視することで，実際にコメントで寄せられた修正が行われているか確認する．

現在 73 件の欠陥報告を調査中である．調査で発見した具体例を以下に挙げる．DERBY-6352 の報告によると，RecoveryAfterBackupTest を実行した際，断続的にアクセスが拒否される欠陥が生じていた．欠陥報告に寄せられたコメントによると，既定のセキュリティ管理においてアクセス許可は必要なかったが，ContextService.java にアクセス認証が実装されていた．コメントではアクセス認証を削除するよう要求されており，テストコードの目視によって，アクセス認証を削除していることを確認した．

4. おわりに

本研究では，Silent Horrors の発生原因を明らかにすることを最終的な目的としている．本稿では，最終的な目的を達成するために，まず，REOPENED されたテストコードを対象として，テストコードの欠陥の発生原因について分析する方法について述べた．

対象としている 73 件の欠陥報告について発生原因を特定した後，REOPENED されたテストコードの欠陥が実際に Silent Horrors の発生原因になっているのかを明らかにする．

本稿では，REOPENED されたテストコードの欠陥のみを対象としているが，今後は REOPENED されていないテストコードの欠陥の発生原因についても分析していく．

謝辞 本研究の一部は，JSPS 特別研究員奨励費（JP17J03330）および JSPS 科研費（基盤（A）：JP17H00731，基盤（C）：JP18K11243）による助成を受けた．

参考文献

[1] Arash Vahabzadeh, Amin Milani Fard, Ali Mesbah. An Empirical Study of Bugs in Test Code The 31st International Conference on Software Maintenance and Evolution(ICSME2015) University of British Columbia Vancouver, BC, Canada

[2] C. Jones. *Programming Productivity*. McGraw-Hill., New York, NY, 1986.

[3] S. McConnell. *Code Complete*. Microsoft Press., Redmond, WA, 1993.

[4] Apache Derby(https://db.apache.org/derby/)

[5] Apache JIRA(https://issues.apache.org/jira/)