



wakayama  
univ.

## テストコードにおける欠陥

**偽陰性**

欠陥が混入したプロダクトコード

欠陥が混入したままのプロダクトコード

Git(p)

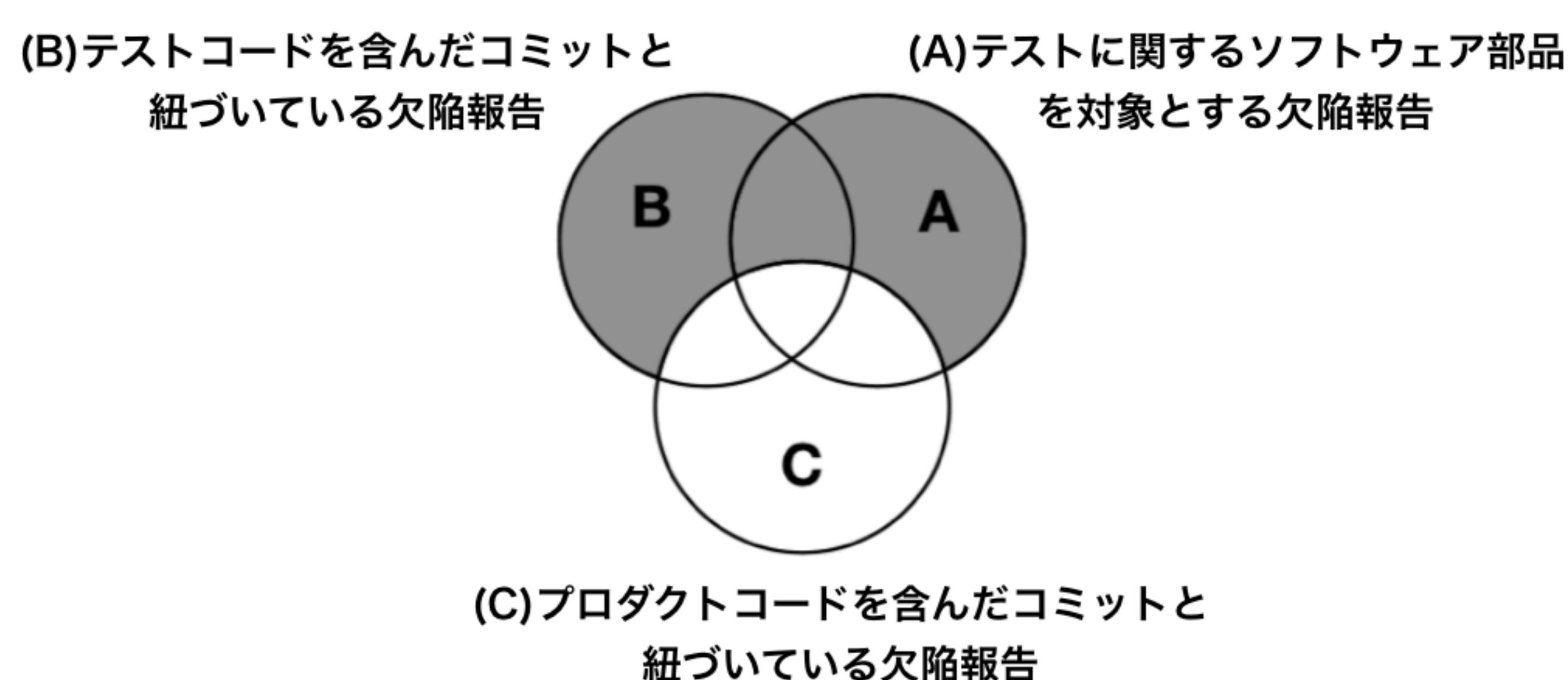
Git(t)

TEST

リビジョン

欠陥混入コミット

## 既存研究[2]

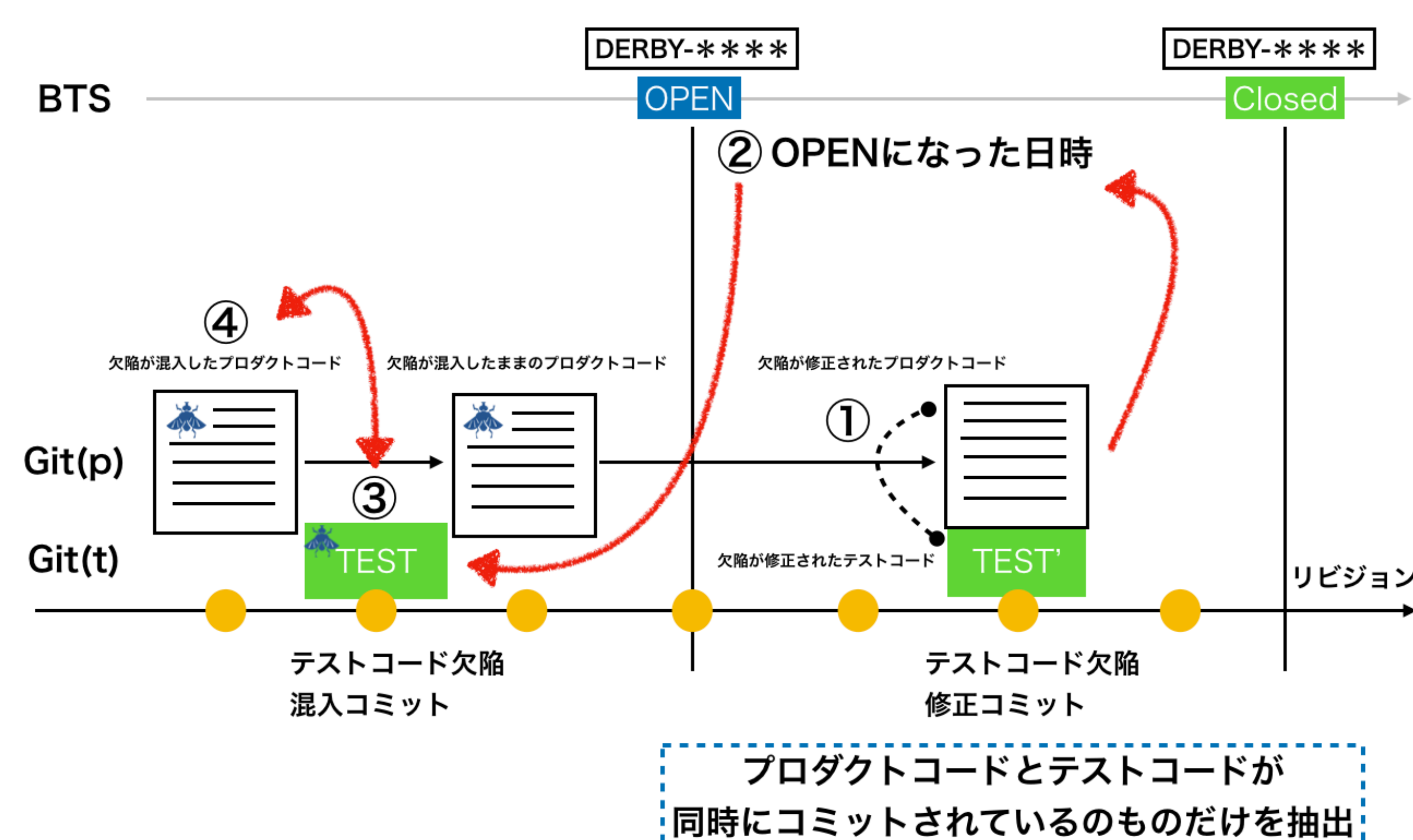


## ・既存研究の問題点

偽陰性の調査を目視によって行っているが、目視ではコミットログをたどることにコストがかかるため、発見が困難である

## 本研究の目的

## 偽陰性の候補となるコミットの抽出方法



1. プロダクトコード (A.java) とテストコード (ATest.java) が同時に修正されたコミットを特定する
2. 実際にテストコードの不具合が報告されているのかを確認する
3. 報告されていれば、直近に変更されたテストコードを特定する
4. その日時より過去で最近プロダクトコードに変更があったコミットを特定する

## 今後の予定

- ・ 偽陰性を自動抽出するツールを実装する
- ・ ツールが抽出したコミットを目視して、偽陰性が正しく検出されたのかを分析する

## 参考文献

- [1] A. van Deursen, L. Moonen, A. van den Bergh, and G. Kok, "Refactoring test code," 2nd Int. Conf. on Extreme Programming and Flexible Processes in Software Engineering (XP 2001), 2001.
- [2] 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