研究進捗状況

2018年11月13日（火）

発表者　吉富　楓雅

テストコードにおける欠陥

- ソフトウェアテストは，プロダクトコードが期待通りに機能するかを検証するために実施
  - プロダクトコードと，テストコードの記述される比率が１：１であるとも報告されている[1]
- プロダクトコードに欠陥が混入するのと同様に，テストコードにも欠陥が混入する恐れがある

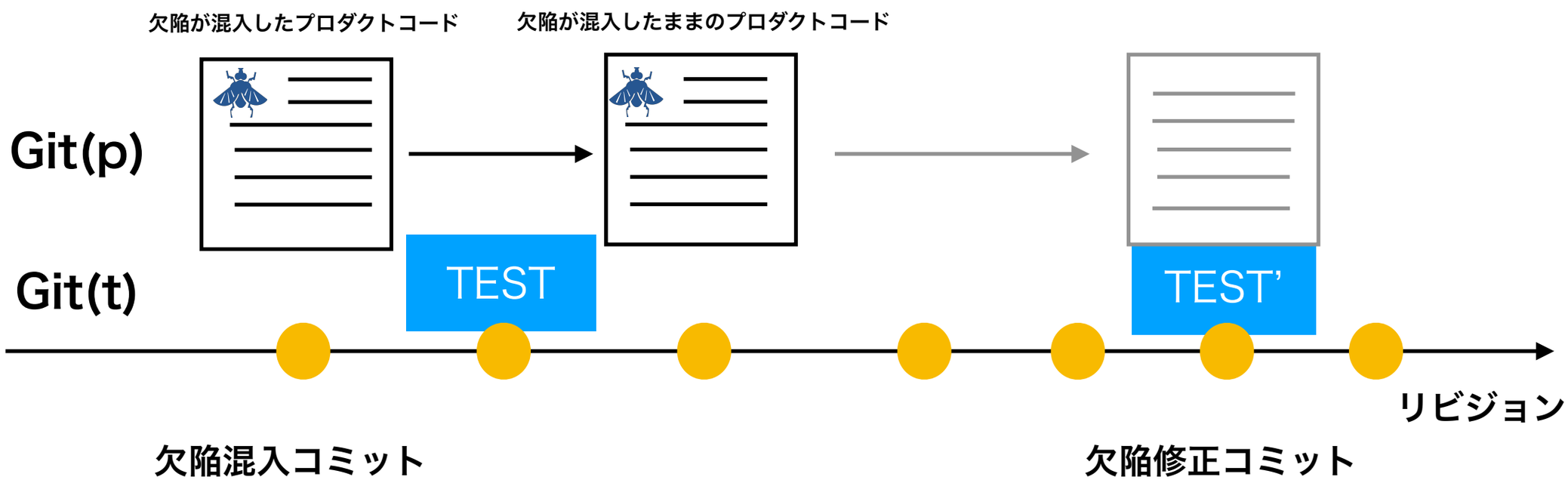
[1]A. van Deursen, L. Moonen, A. van den Bergh, and G. Kok, “Refactoring test code,” 2nd Int. Conf. on Extreme Programming and Flexible Processes in Software Engineering (XP 2001), 2001.

テストコードの欠陥による問題

- Cunninghamは２つに分類[2]
  - **偽陰性(False Negative)：**
    - **プロダクトコードに欠陥があるが，テストがPassとなる**
  - 偽陽性(False Positive)：
    - プロダクトコードに欠陥がないが，テストがFailとなる

[2]http://wiki.c2.com/?BugsInTheTests

偽陰性のイメージ



欠陥が混入したプロダクトコードに対して
テストを実行した際に欠陥が見逃されている

偽陰性解決までの流れ

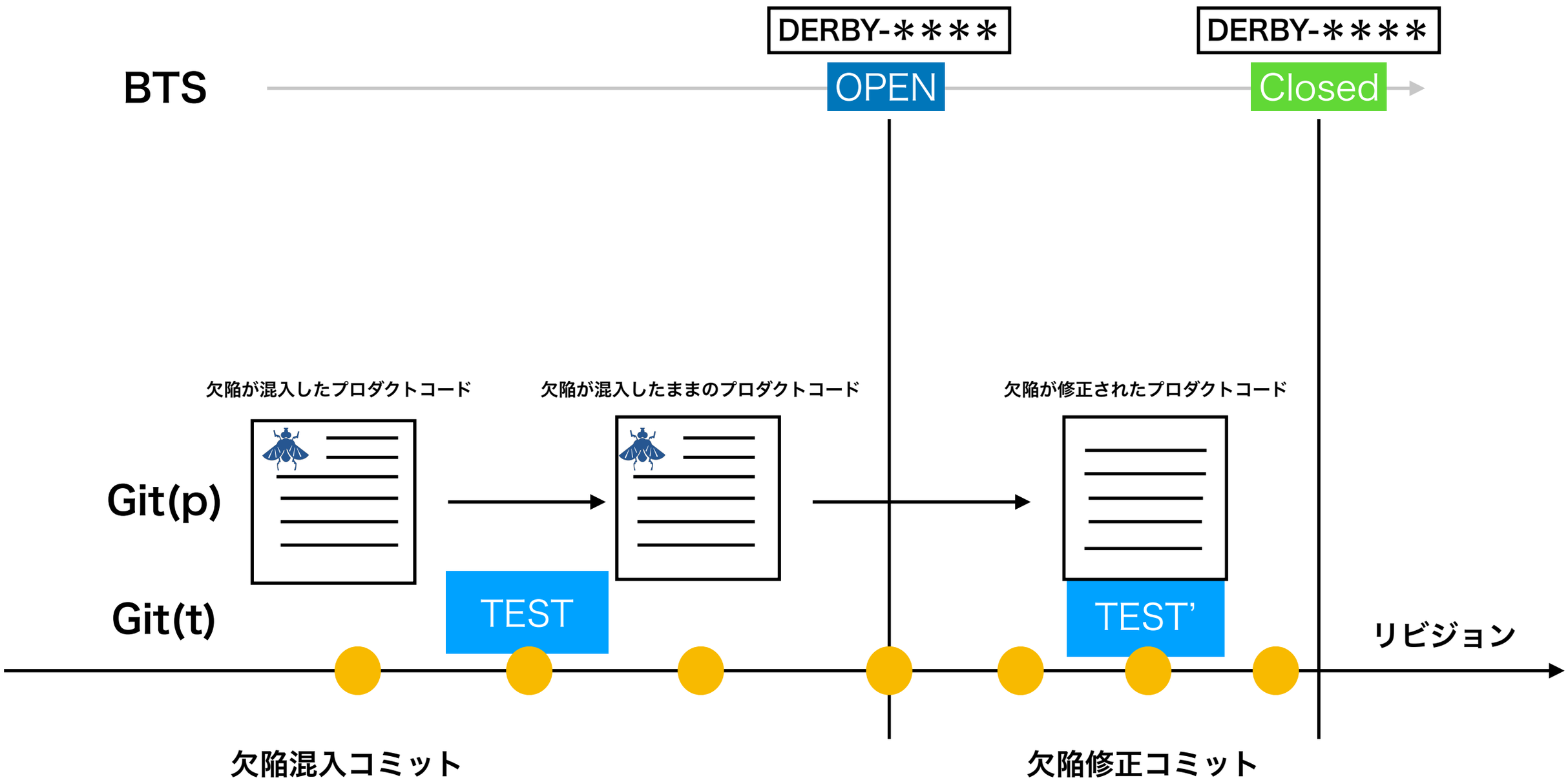
1. 予期しないエラーや期待していない動作が発生することでBTSに欠陥報告(DERBY-****)が登録される

2. 精査した結果プロダクトコードに欠陥が混入していたことが確認される
(ステータス:**OPEN**)

3. プロダクトコードは修正され，見逃し欠陥を発生させる原因となったテストコードも修正される

4. 欠陥が取り除かれると，issueは閉じられる
(ステータス:**CLOSED**)

テストコードとプロダクトコードの修正過程



既存研究

- Vahabzadehらは，5,556件のテストコードに関係する欠陥報告を抽出[3]
  - ランダムに443件の欠陥報告を手作業で抽出
    - 偽陰性は15件と全体の3%であった

[3] 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

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

研究の目的

- 偽陰性の分析を行うために必要となるデータを抽出するツールを提案し，偽陰性となるデータがどの程度存在するのかを分析する

偽陰性の候補となるコミットの抽出

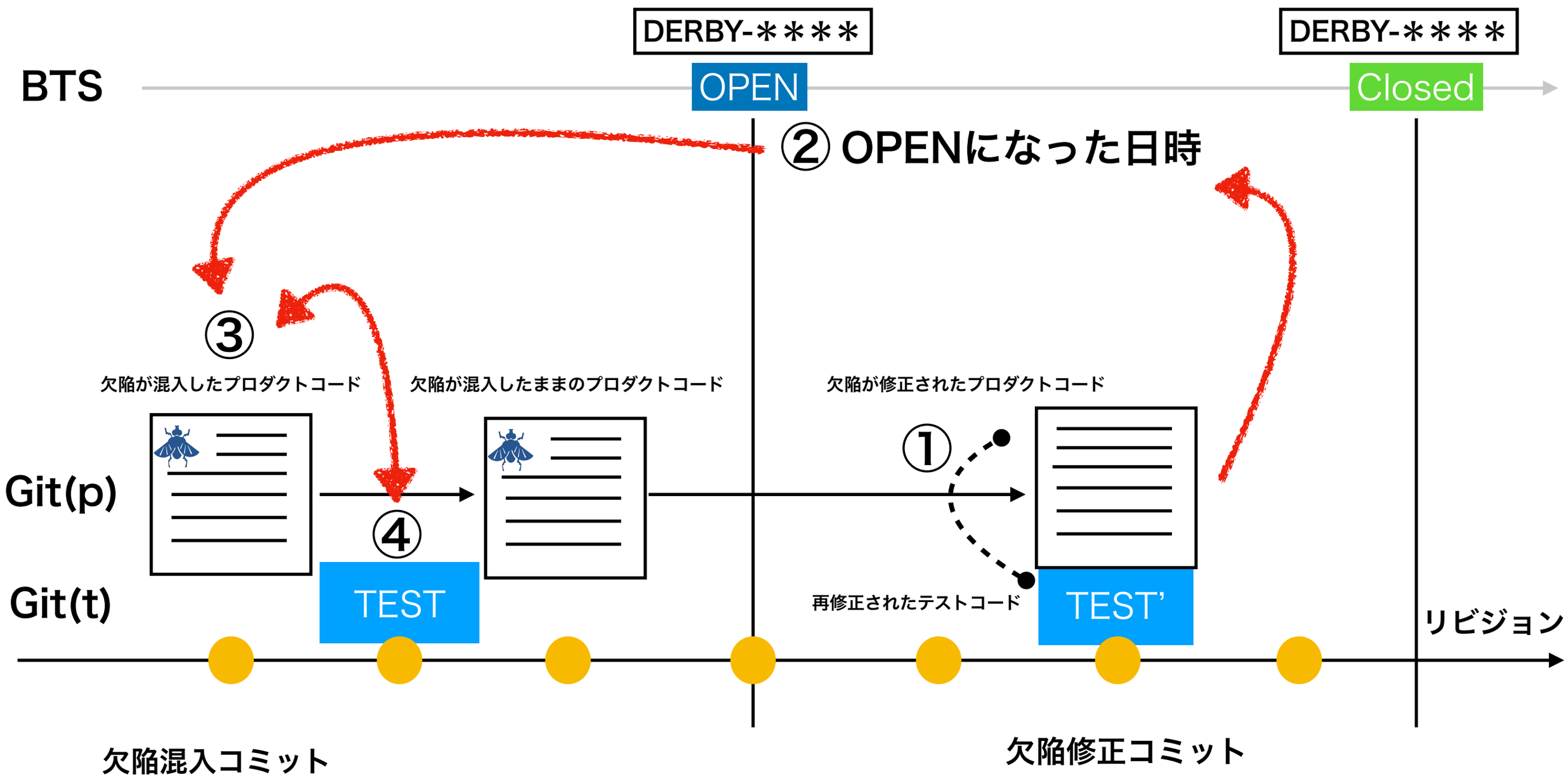
1. プロダクトコードとテストコードが同時に修正されたコミットを発見する

2. 修正コミットの不具合票のステータスがOPENになった日時を特定する

3. 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

4. もしプロダクトコードにテストコードがあれば偽陰性の候補とする

欠陥修正コミットの抽出と欠陥混入コミットの特定



プロダクトコードとテストコードが
同時にコミットされているのものだけを抽出

今後の予定

- TODO
  - データ抽出ツールの実装
- DOING
  - 卒論の執筆(ソフトウェアテスト，既存研究など)