

進捗状況

2018年10月18日 (木)

ソフトウェアテスト

- 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

* できるだけ多くの欠陥を発見することが求められる

▶ しかし**全ての欠陥を発見することは不可能**

- **作成したテストコードには欠陥が混入する可能性がある**

* 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]

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

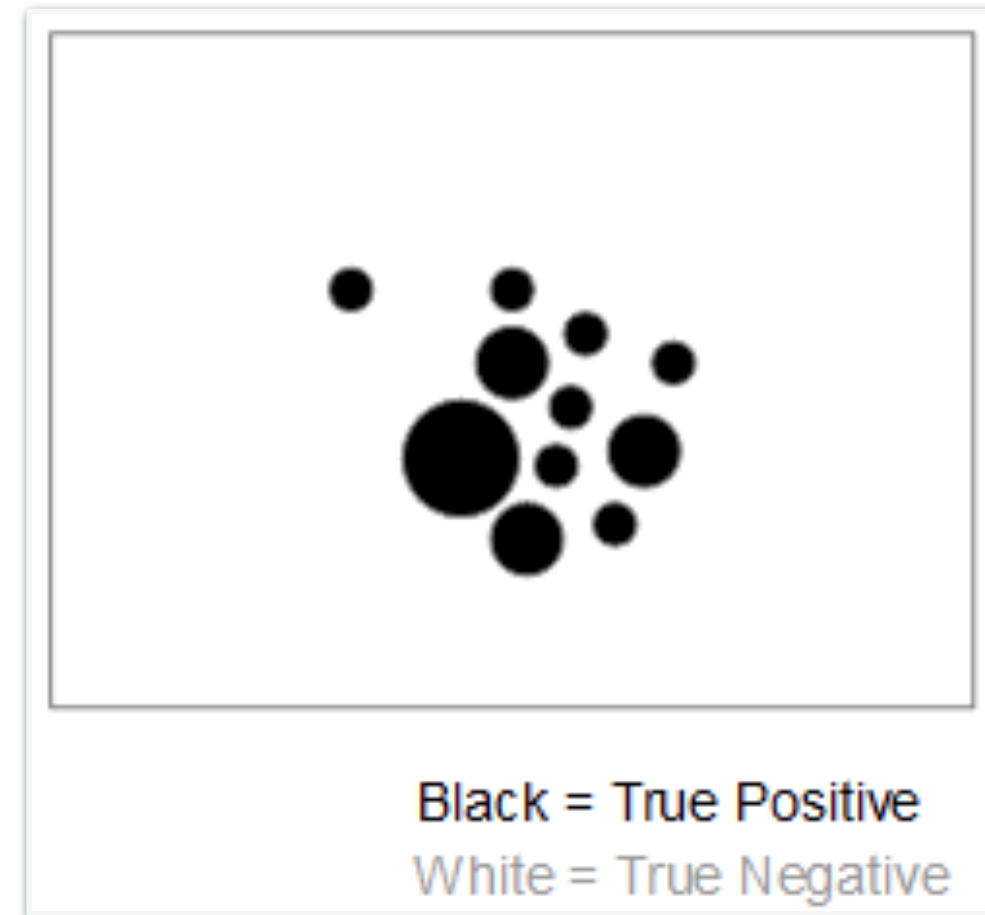
[3]S. McConnell. Code Complete. Microsoft Press., Redmond, WA, 1993.

テストコードに欠陥が混入

- テストコードに欠陥が混入すると、誤検知が発生する
- 誤検知には 2 種類存在する
 - * **偽陰性(False Negative)**
 - * **偽陽性(False Positive)**

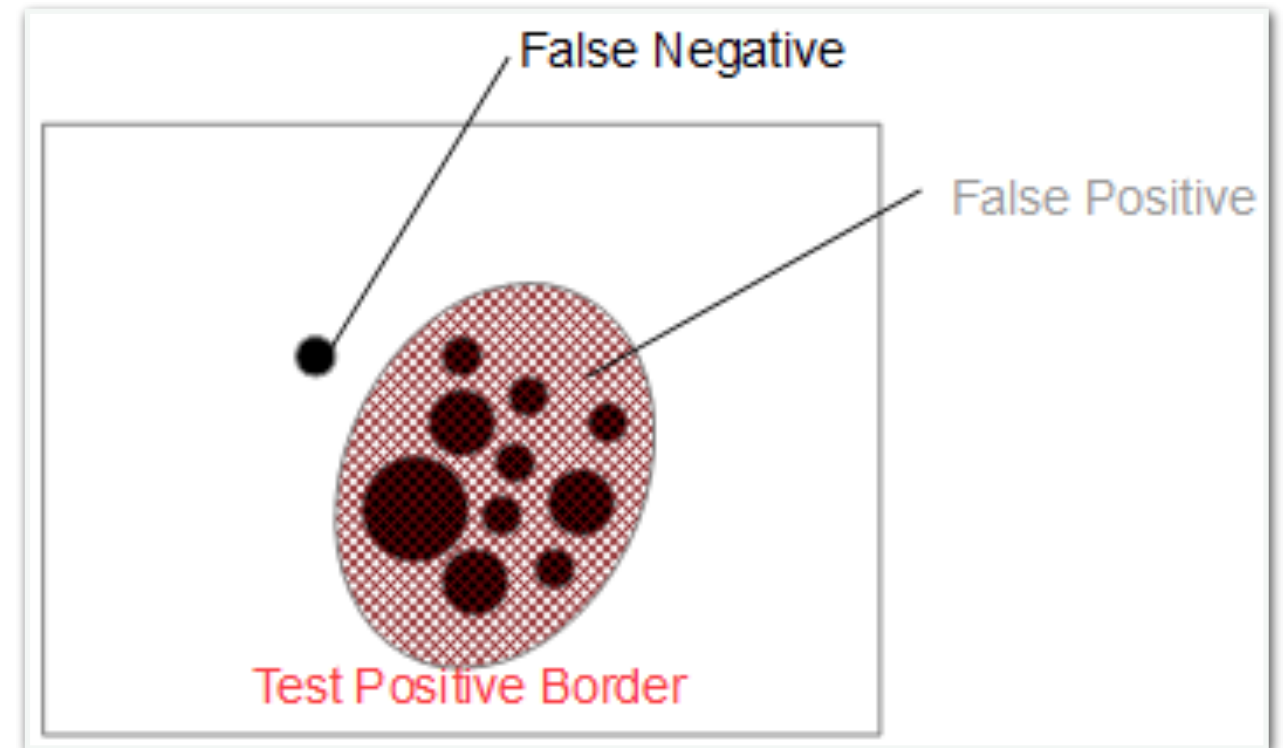
真陽性と真陰性

- 真陽性(True Positive)
 - * 実際にプロダクトコードに欠陥がある
- 真陰性(True Negative)
 - * 実際にプロダクトコードに欠陥がない



誤検知：偽陽性と偽陰性

- 偽陽性(False Positive)
 - * テストでプロダクトコードに欠陥ありと判定されたが、実際には欠陥はなかった
- 偽陰性(False Negative)
 - * テストでプロダクトコードに欠陥なしと判断されたが、実際には欠陥があった

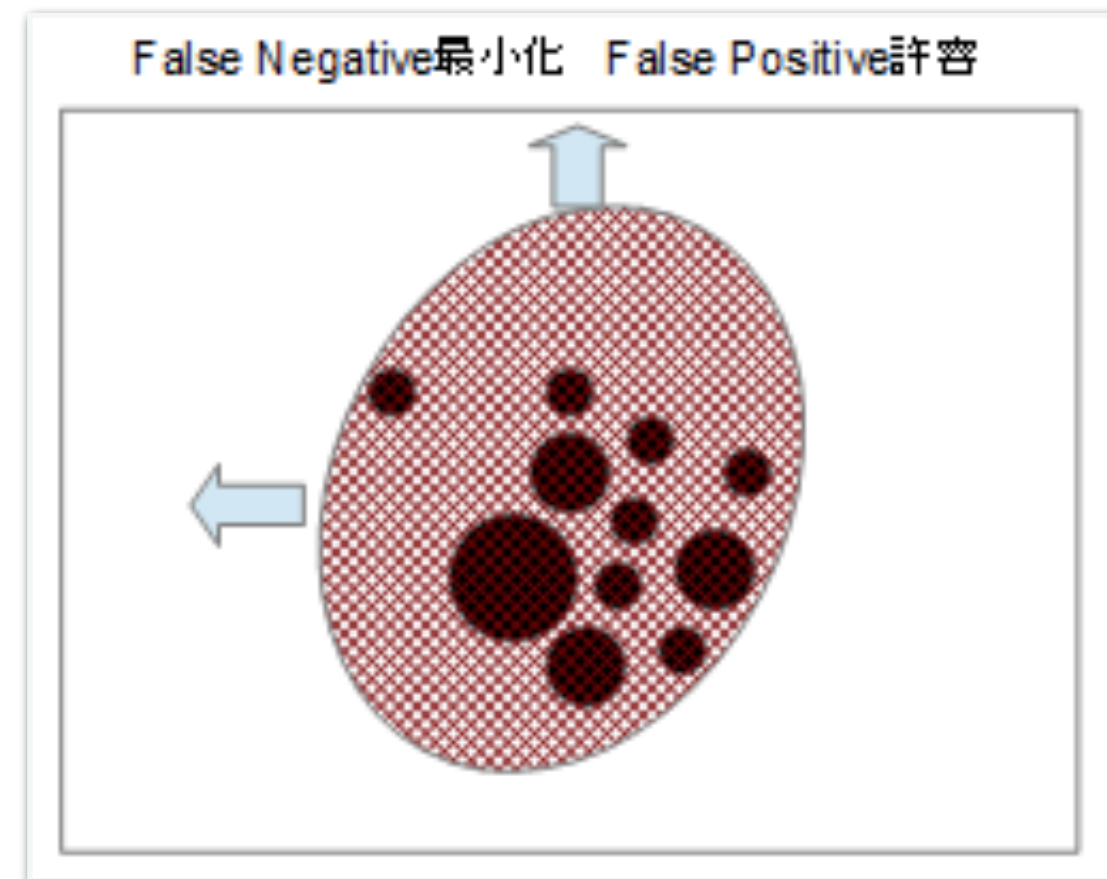
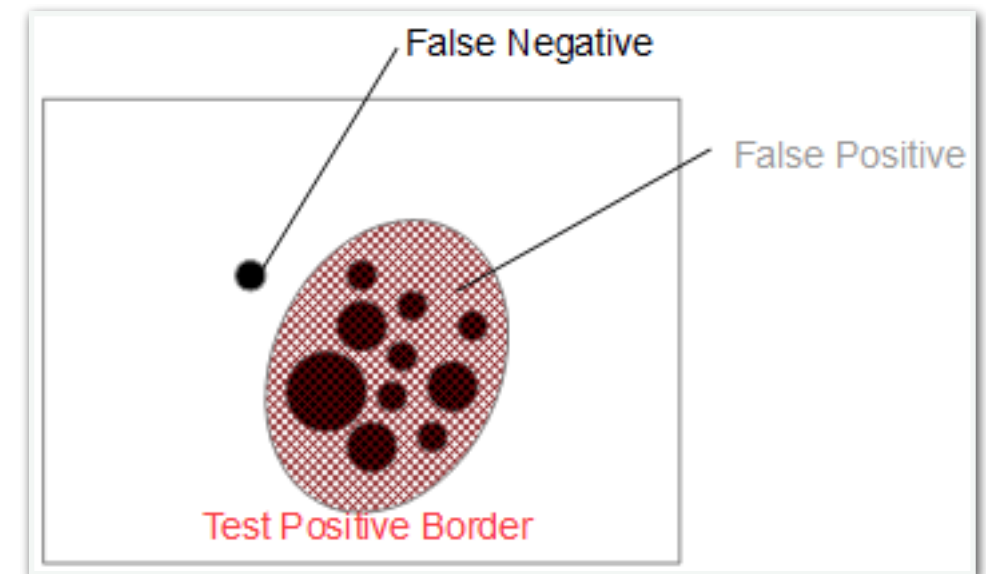


研究の目的

- 偽陰性の発生原因を明らかにすることで見逃し欠陥を検出し、欠陥の発見率を向上する

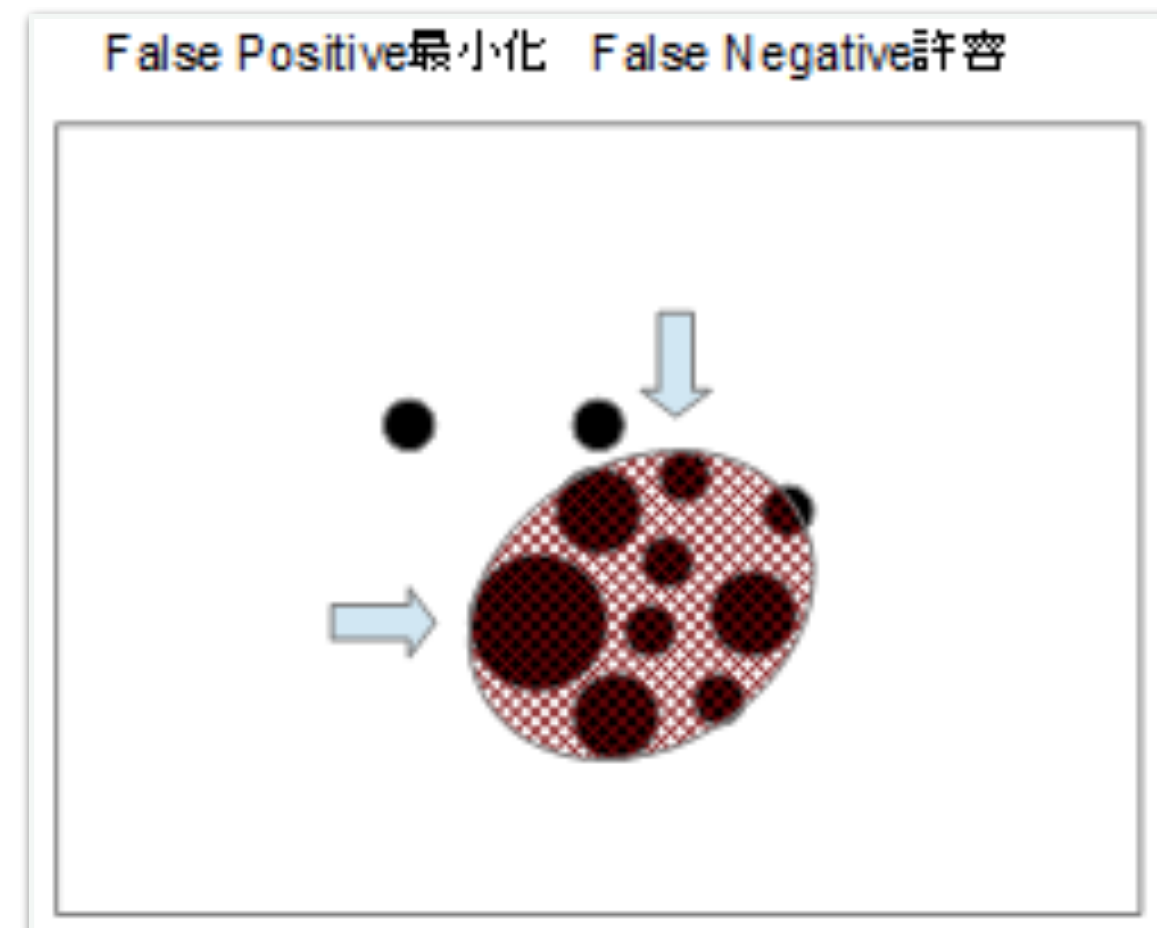
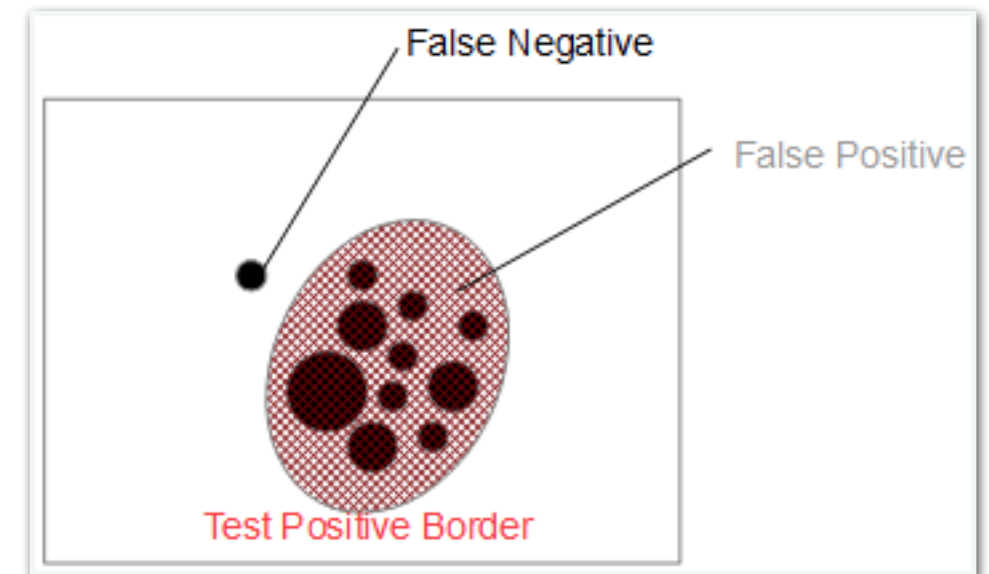
False Negativeの最小化

- テストで陽性反応が出なかったものを検知できるように，陽性判定の範囲を広げる
- より多くの欠陥を発見することを考えると False Negativeの最小化が適切
 - * プロダクトコードに欠陥が存在したというケースをスルーすると，適切な修正ができなくなる
- デメリット
 - * False Positiveが増加する恐れがある



False Positiveの最小化

- テストで陽性反応が出たものを真の結果が得られるように，陽性判定の範囲を狭める



卒論での目的

- REOPENEDというタグ付けがされたテストでは偽陰性による見逃し欠陥が発見しやすいのではないかという仮定のもとで、偽陰性の発生原因を明らかにする
- (という最終目的に向けたファーストステップを行う)

リサーチクエスト ヨン

リサーチクエスチョン

- RQ1：REOPENEDされたテストはどの程度の頻度で偽陰性を引き起こすのか？
 - * REOPENEDのタグがつけられたテストは、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。よって偽陰性を引き起こしたテストの発見が容易になる
- RQ2：REOPENEDされたテストのうち、偽陰性を引き起こしたテストに不足であったテストケースは何か？
 - * 偽陰性を引き起こしたテストに不足だったテストケースが何かを分析することにより、テストが必要な箇所を特定し、テストを行う範囲を広げることができる