An Empirical Study of Bugs in Test Code

Arash Vahabzadeh　Amin Milani Fard　Ali Mesbah

The 31st International Conference on Software Maintenance and Evolution(ICSME2015)
University of British Columbia Vancouver, BC, Canada

発表者　吉富　楓雅

目次

- 背景とリサーチクエスチョン
- 分析方法
- 分析結果
- 結論と所感

背景と
リサーチクエスチョン

ソフトウェアテストとは

- 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検証
- テストケースは, 以前に動作していた機能がまだ動作するかどうか確認
  - しかしこれも開発者が作成したコードであり, バグが混入する可能性がある
- 実際多くの開発者が, テストケースはテストされるコードよりもエラーを含む可能性が高いと述べている[1][2][3]

[1]W. Cunningham. BugsInTheTests. Bugs in the tests. http://c2.com/cgi/wiki?

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

[3]R.J.Weiland.TheProgrammer'sCraft:ProgramConstructionComputer Architecture, and Data Management. Reston Publishing., Reston, VA, 1983.

テストコードのバグ

- 2つに分類することができる

  * Silent Horrors :

    - テストコードに誤りがあるとプロダクトコードの誤りが見つからないことがある

  * False Alarms :

    - プロダクトコードに誤りがないがテストが失敗することがある

リサーチクエスチョン

1. テストバグはどのくらいの頻度で発生するのか？

2. テストバグはどのような種類が一般的なのか？

   * Silent HorrorsとFalse Alarmsの比率

3. テストバグの一般的なカテゴリは何か？

4. バグ検出ツールはテストバグを発見できるのか？

   * 今回はFindBugsを使用

テストバグを修正するシナリオ

テストが成功するか失敗するかで分類

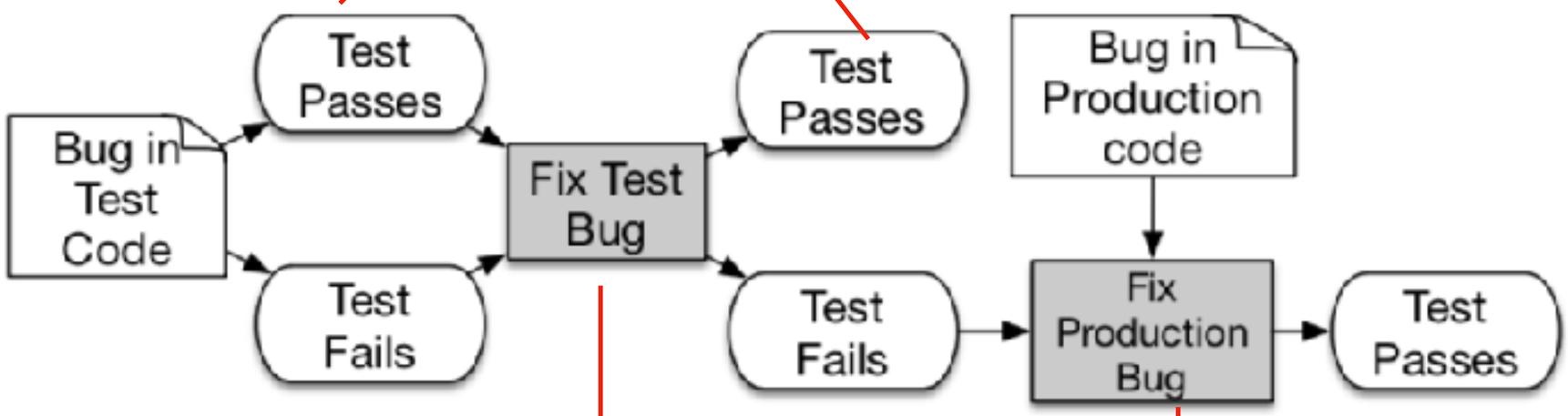


Fig. 1: Different scenarios for fixing test and production bugs.

まずテストバグを修正

次にプロダクトコードのバグを修正

分析方法

使用するプロジェクト

- ASF(Apache Software Foundation)のプロジェクト
  - 110以上のプロジェクト
  - 448個のサブプロジェクト

- Apacheプロジェクトのバグリポジトリとバージョン管理システムを検索

データ収集

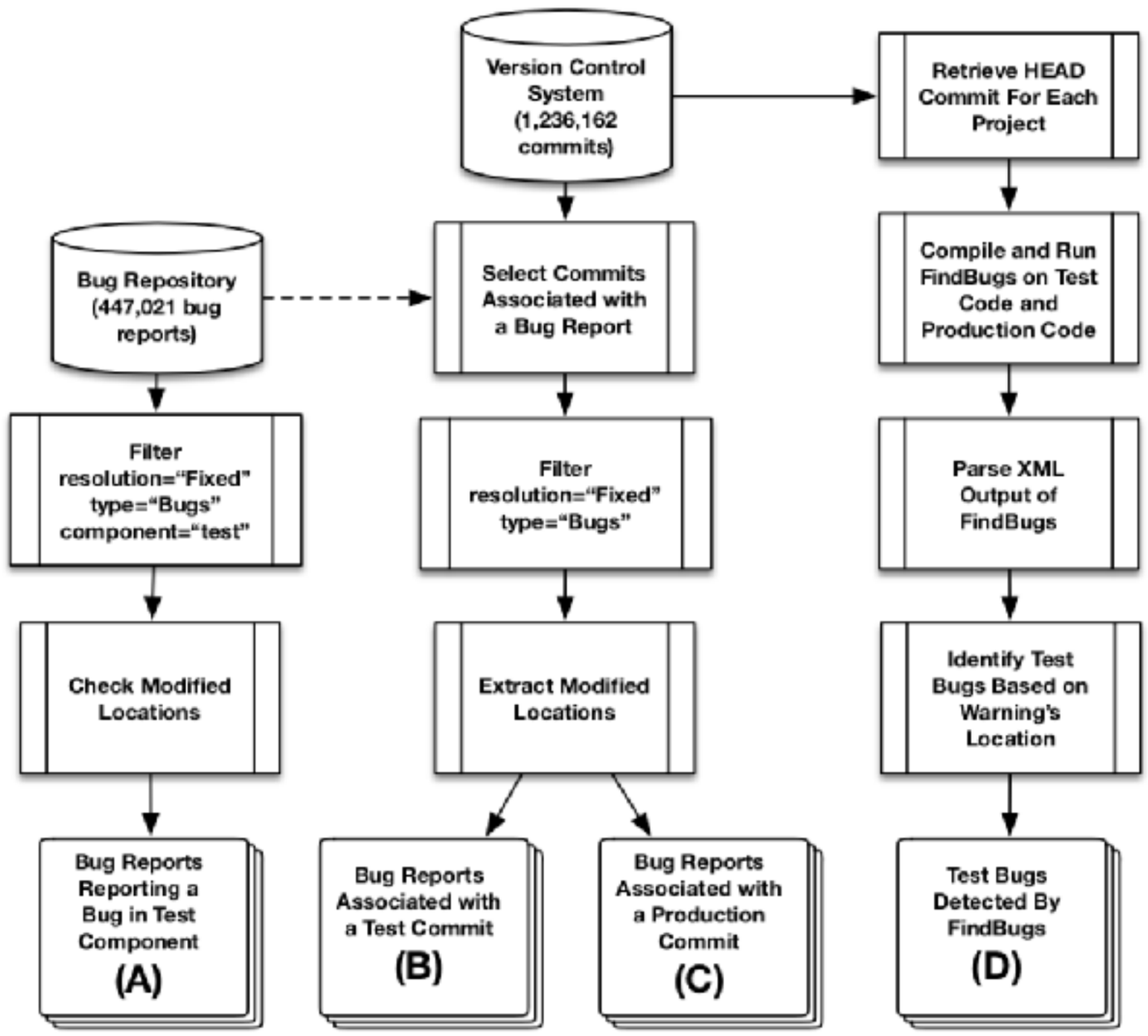


Fig. 2: Overview of the data collection phase.

図について

- A: テストに関するバグをリポジトリマイニングする
- B: バージョン管理システムによってバグレポートを含むコミットを抽出. 変更された場所を特定する. このうちテストコードのコミットに関するもの
- C: バージョン管理システムによってバグレポートを含むコミットを抽出. 変更された場所を特定する. このうちプロダクトコードのコミットに関するもの
- D: 最新バージョンのテストコードとプロダクトコードでFindBugs(ver3.0.0)を実行しテストバグを検出する

テストバグの抽出

今回の研究ではテストコードのコミットに関連するが
プロダクトコードのコミットには関連しないバグレポートのみ
を考慮する

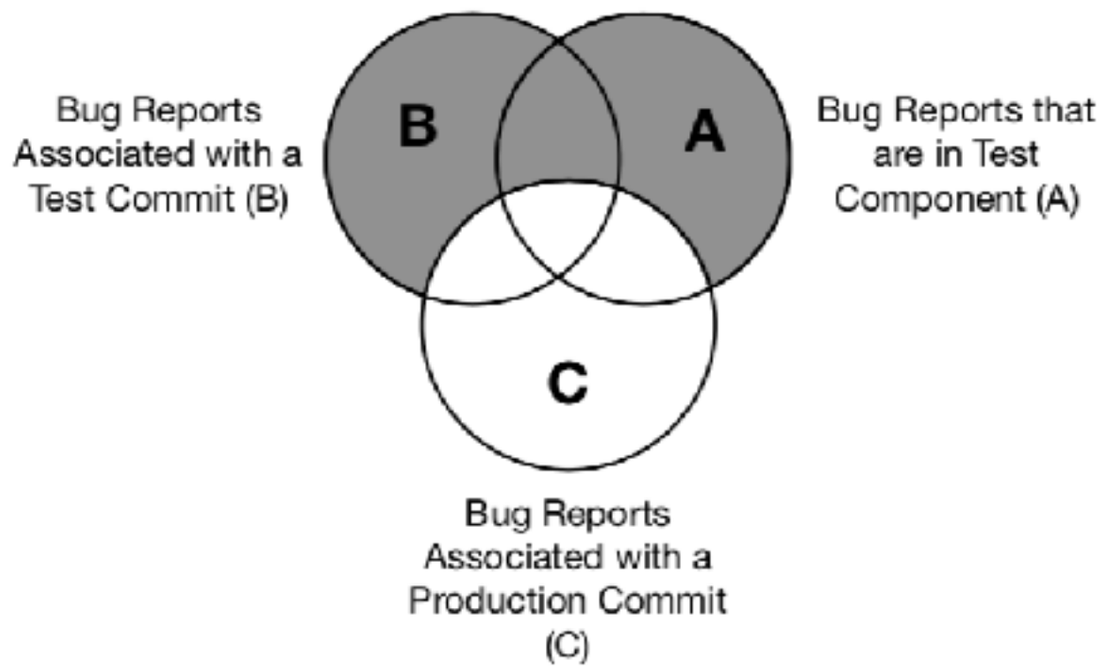


Fig. 3: Bug reports collected from bug repositories and version control systems. $|(A \cup B) - C| = 5,556$ test bug reports in total ($|B - A - C| = 3,849$, $|A - C| = 1,707$).

分析結果

RQ1:
テストバグはどのくらいの頻度で発生するのか？

- 448プロジェクト中211プロジェクト(47%)でテストバグが発見された

  * 5,556のバグレポートが発見された((Ａ ∪ Ｂ) - Ｃ)

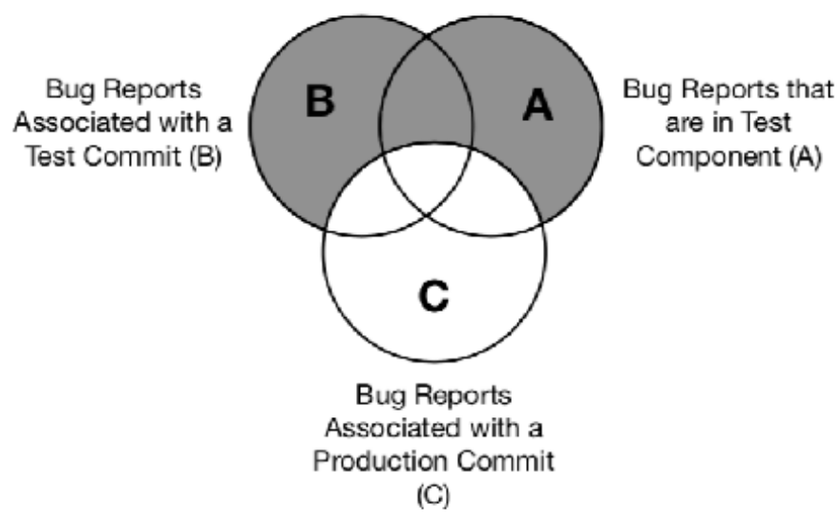


Fig. 3: Bug reports collected from bug repositories and version control systems. $|(A \cup B) - C| = 5,556$ test bug reports in total ($|B - A - C| = 3,849$, $|A - C| = 1,707$).

RQ1:
テストバグはどのくらいの頻度で発生するのか？

- テストバグが多かったASFプロジェクト

TABLE II: Top 10 ASF projects sorted by the number of reported test bugs.

Project	Production Code KLOC	Test Code KLOC	# Test Bug Reports
Derby	386	370	614
HBase	587	195	440
Hive	836	124	295
Hadoop HDFS	101	57	286
Hadoop Common	1249	380	279
Hadoop Map/Reduce	60	24	231
Accumulo	405	78	187
Qpid	553	93	152
Jackrabbit Content Repository	247	107	145
CloudStack	1361	228	111

RQ2:
テストバグはどのような種類が一般的なのか？

- 手作業で499個のバグレポートのサンプルを調査
  - ５６件は十分に報告されていなかったため残りの443件を分類
- このうち97%はテストが失敗, 3%が成功した
  - つまり97%はFalse Alarmsで，３%は Silent Horrors である

RQ2:
テストバグはどのような種類が一般的なのか？

- False Alarmsのカテゴリ分け

TABLE III: Test bug categories for false alarms.

Category	Subcategory	Description
Semantic Bugs	**S1.** Assertion Fault	Fault in the assertion expression or arguments of a test case.
	S2. Wrong Control Flow	Fault in a conditional statement of a test case.
	S3. Incorrect Variable	Usage of the wrong variable.
	S4. Deviation from Test Requirement and Missing Cases	A missing step in the exercise phase, missing a possible scenario, or when test case deviates from actual requirements.
	S5. Exception Handling	Wrong exception handling.
	S6. Configuration	Configuration file used for testing is incorrect or test does not consider these configurations.
	S7. Test Statement Fault or Missing Statements	A statement in a test case is faulty or missing.
Environment	**E1.** Differences in Operating System	Tests in this category pass on one OS but fail on another one.
	E2. Differences in third party libraries or JDK versions and vendors	Failure is due to incompatibilities that exist between different versions of JDK or different implementations of JDK by different vendors, or different versions of third party libraries.
Resource Handling	**I1.** Test Dependency	Running one test affects the outcome of other tests.
	I2. Resource Leak	A test does not release its acquired resources properly.
Flaky Tests	**F1.** Asynchronous Wait	Test failure is due to an asynchronous call and not waiting properly for the result of the call.
	F2. Race Condition	Test failure is due to non-deterministic interactions of different threads.
	F3. Concurrency Bugs	Concurrency issues such as deadlocks and atomicity violations.
Obsolete tests	**O1.** Obsolete Statements	Statements in a test case are not evolved when production code has evolved.
	O2. Obsolete Assertions	Assertion statements are not evolved as production code evolves.

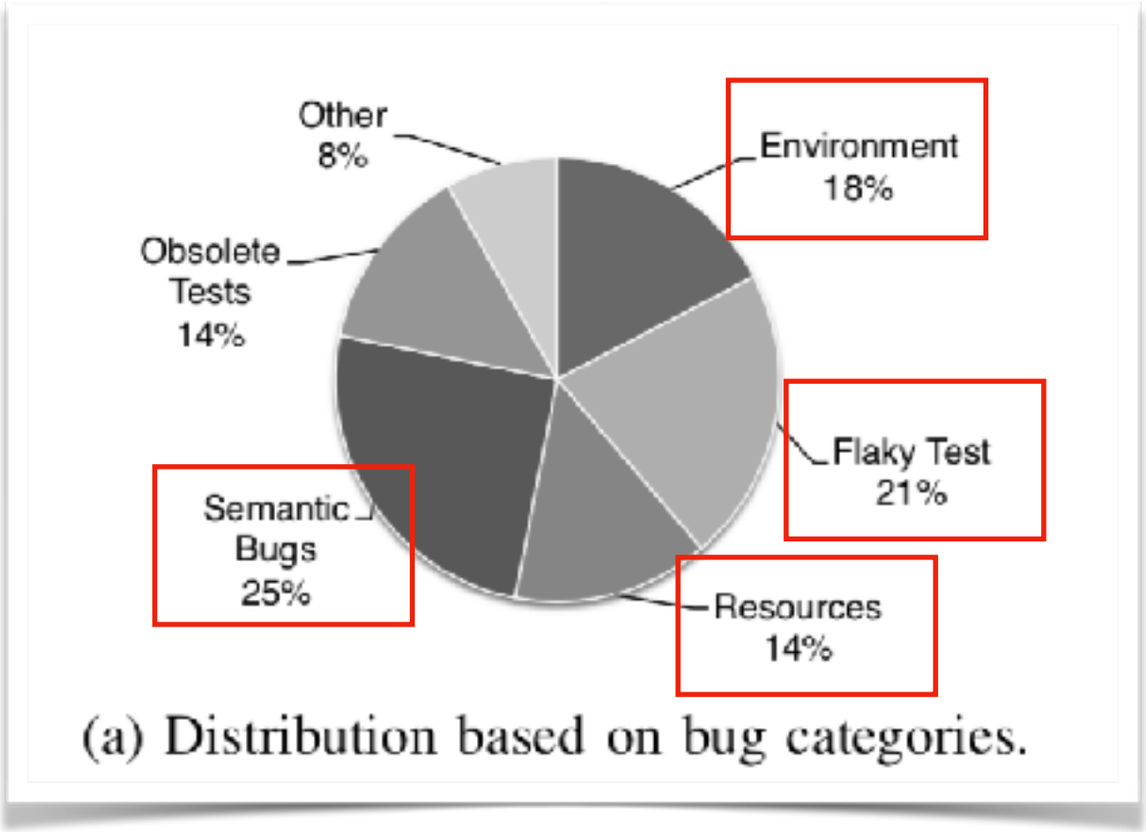
RQ3:
テストバグの一般的なカテゴリは何か？

* セマンティックバグ(25%): 仕様, プロダクトコード, 及びテストコードで矛盾が生じたものを反映する

* フレークテスト(21%): 同じコードで成功, 失敗両方の結果が出てしまうこと

* 環境(18%)

* リソース(14%)



(a) Distribution based on bug categories.

RQ4:
バグ検出ツールはテストバグを発見できるのか？

- FindBugsで発見されたテストバグ(129プロジェクトのうち20のプロジェクトで報告されたもの(平均1.6個のバグ))

TABLE IV: Test code warnings detected by FindBugs.

Bug Description	Bug Category	Percentage
Inconsistent synchronization	Flaky	29.8%
Possible null pointer dereference in method on exception path	Semantic	17.6%
Using pointer equality to compare different types	Semantic	8.8%
Possible null pointer dereference	Semantic	7.3%
Class defines field that masks a superclass field	Semantic	3.9%
Nullcheck of value previously dereferenced	Semantic	2.9%
An increment to a volatile field isn't atomic	Flaky	2.9%
Method call passes null for nonnull parameter	Semantic	2.4%
Incorrect lazy initialization and update of static field	Flaky	2.4%
Null value is guaranteed to be dereferenced	Semantic	2.0%

RQ4:
バグ検出ツールはテストバグを発見できるのか？

- 以前の研究では静的コード解析ツールが5〜15%のバグが検出されたことを示した[3][4]
  - 今回はどのように実行されるのかを調査
- 623件のバグレポートのうち, 50件のバグレポートをサンプリング
- しかし, FindBugsは50のテストバグを1つも検出できなかった
  - 例: 誤検知, バグレポートに直接関係していない, など

[3]C. Couto, J. a. E. Montandon, C. Silva, and M. T. Valente. Static correspondence and correlation between field defects and warnings reported by a bug finding tool. *Software Quality Control*, pages 241–257, 2013.
[4]M. G. Nanda, M. Gupta, S. Sinha, S. Chandra, D. Schmidt, and P. Balachandran. Making defect-finding tools work for you. In Proceedings of the International Conference on Software Engineering *(ICSE)*, pages 99–108. ACM, 2010.

RQ4の結果に対しての考察

- FinfBugsのテストコードに対してのバグ検出パターンが, 全424パターンのうち6パターンしかない

  * プロダクトコードに対して重点的にパターン化されていて, テストコードに対しては手薄な状態

- テストコード内の環境バグ, フレークテスト、およびリソースバグのパターンは, 現在のバグ検出ツールはテストコードでそれらを検出できない

- 今回はテストコードのパターンがわずかに変更されたために, FindBugsは検出することができなかった

結論と所感

結論

- テストコードにバグが混入している例が存在することがわかった
  - 実際にテストバグが様々なプロジェクトで見られることが判明した
- 大半のテストバグはFalse Alarmsである
  - セマンティックバグやフレークテスト
- FindBugsはテストバグの検出に向いていない
  - 現在の検出ツールにはテストバグを発見する能力がない

所感

- データ収集の方法は自身の研究に役立ちそう
  - 今後この方法を参考にしてデータ収集したい
- テストバグが十分に検出できない状況は, 企業のテスターにとって不安要素の1つになってしまうのではないか
  - テスターが安心してテストコードを書けないのは大問題
- 満足なテストを書くために, テストコードが正しいことを保証できるようなツールを作りたい