AGEST

ミューテーションテスト

単体テストの品質を見える化して改善する

2023年3月9日

株式会社AGEST

坂本 竜一

自己紹介

AGEST

名前

坂本 竜一

所属

株式会社AGEST（AGEST, Inc.）

- お客様の品質課題を解決する会社
- QAコンサルタント
- ソースコード解析等をしている

目次

1. 単体テストの重要性

2. 単体テストの品質とは

3. ミューテーションテスト

4. 課題と対策

5. まとめ

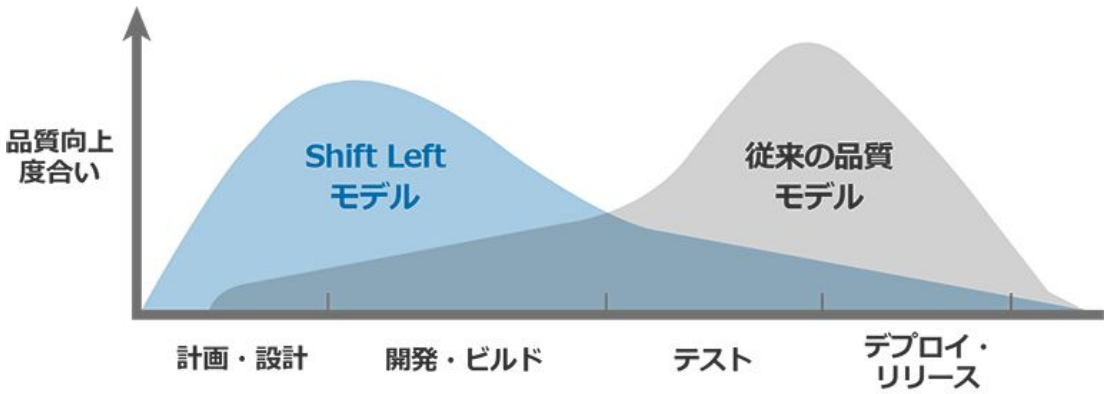
単体テストの重要性

世はアジャイル時代

スピードが重要
- 市場ニーズの多様化
- 新技術・サービスの続出

テスト工程での品質担保では間に合わない
- 手戻りの発生
- テスト工数肥大によるリリース遅延

⇒開発工程での品質が重要
- シフトレフト
- 開発工程の品質向上



単体テストの重要性

AGEST

単体テストケース漏れは市場不具合につながる

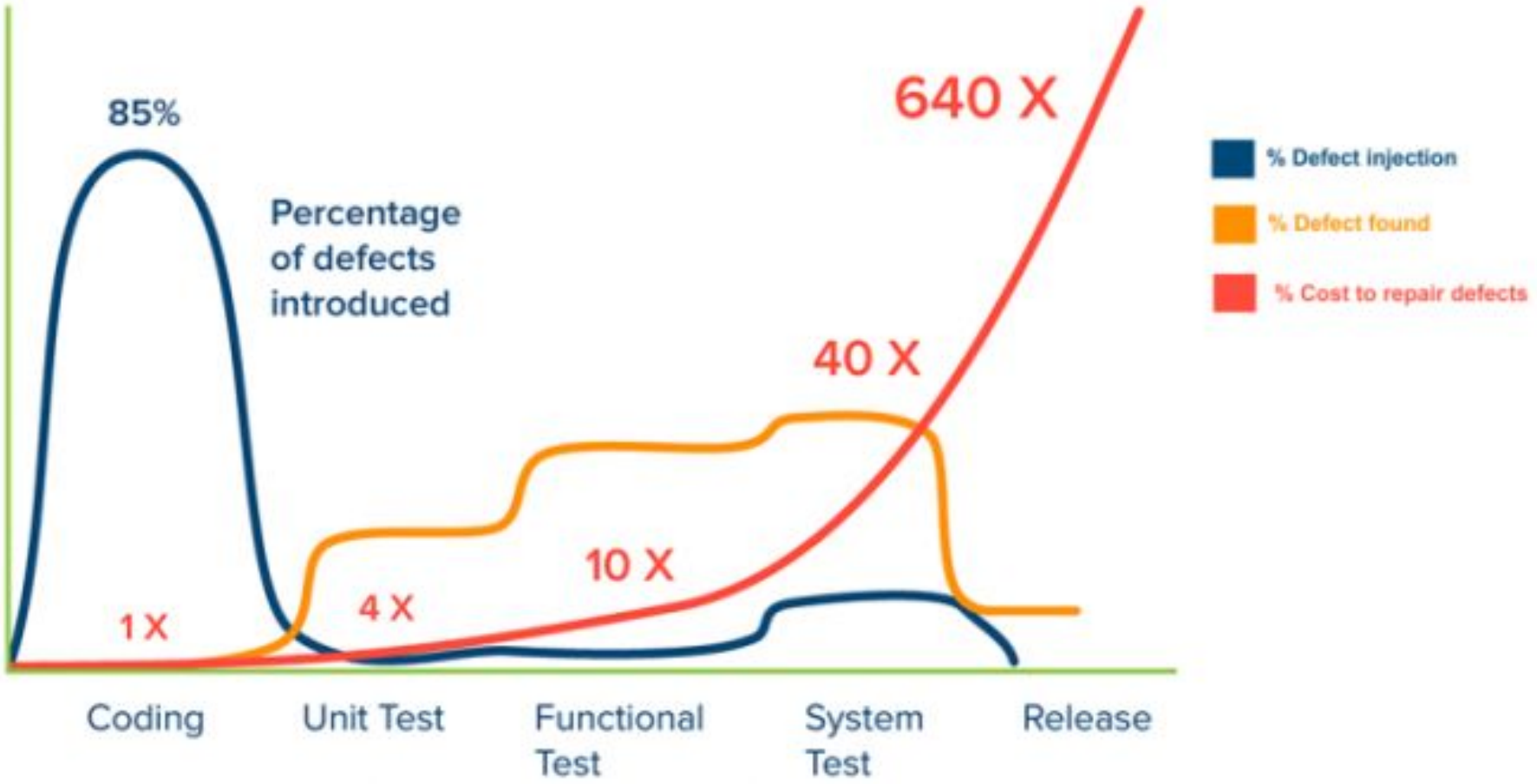
- システムテストで単体テストレベルのバグを見つけることは難しい

システムテストでの不具合検出・修正はリスクが高い

- 納期
- 修正コスト（影響範囲が大きい）
- 修正による新たなバグ

⇒単体テストの品質の向上が重要

単体テストの重要性



Capers Jones, Applied Software Measurement: Global Analysis of Productivity and Quality

単体テストの品質とは

単体テストの品質とは

コードカバレッジ（網羅率）

- 全体のコードに対し、テストによって実行された命令の割合を表す

種類

- ステートメントカバレッジ（命令網羅／C0カバレッジ）
  - プログラム内の命令文をどの程度テストしたかを表す
- ブランチカバレッジ（分岐網羅／C1カバレッジ）
  - コード内の判定条件（分岐）をどの程度テストしたかを表す
- コンディションカバレッジ（条件網羅／C2カバレッジ）
  - 条件内に複数条件があるときそれぞれの真／偽をどの程度テストしたかを表す
- MC/DC(Modified Condition/Decision Coverage)
  - C1とC2をどの程度テストしたかを表す

単体テストの品質とは

AGEST

カバレッジが高ければ品質が高い？

- 命令実行していても下記の問題が存在するケースがある
  - 境界値テストがなく不具合を検出できない
    - 月入力チェック機能で入力値0,1,12,100でテスト⇒if (1 <= in && in <= 13)
  - 不適切なテスト（EXPECT_EQ(0, add(0, 0)) 等）

⇒網羅していても不具合を検出できないことがある

より多くの不具合を検出できる単体テストの品質が高い

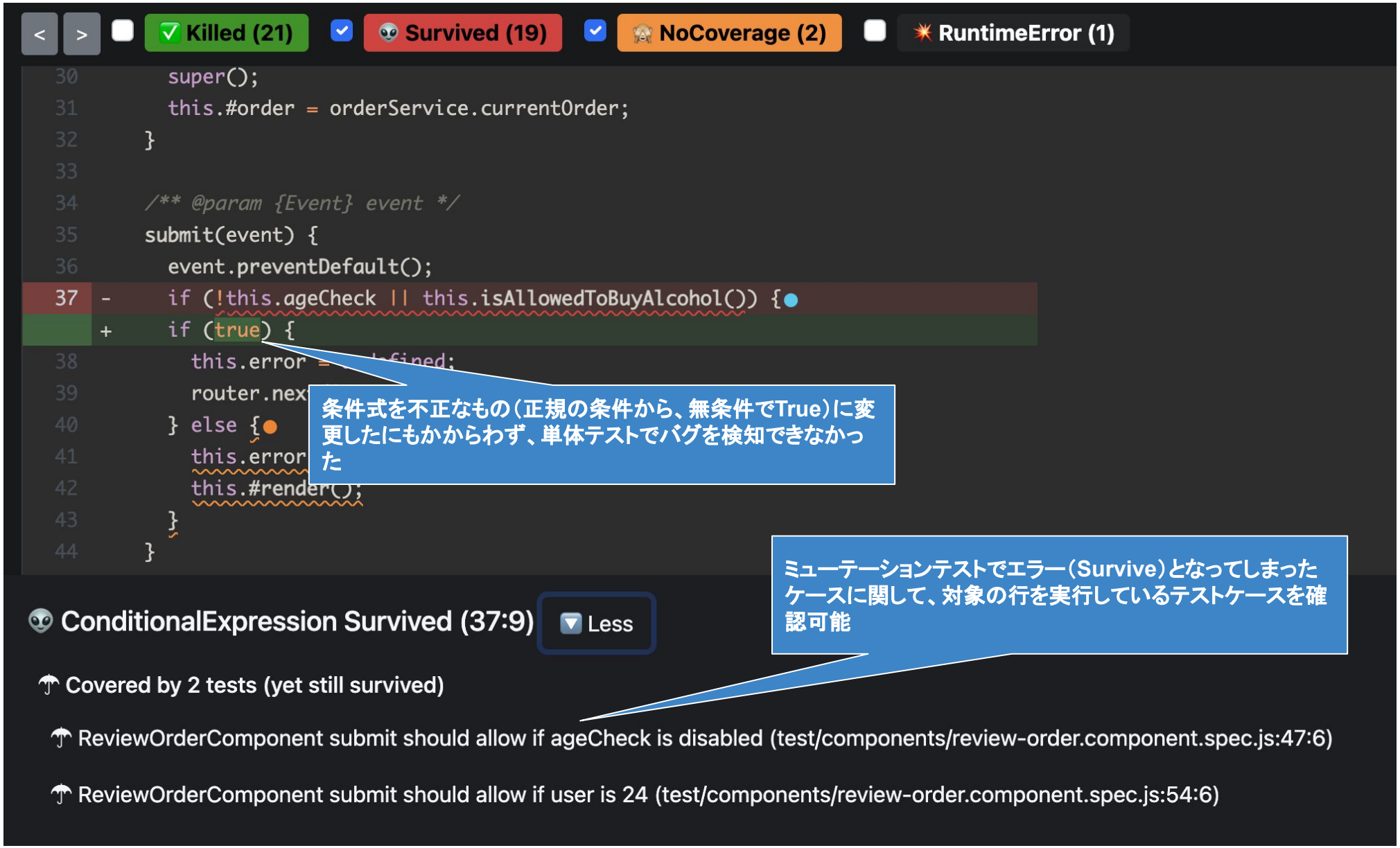
⇒カバレッジが高くても単体テストの品質が高いとは言えない

ミューテーションテスト

ミューテーションテストとは

- 単体テストの品質を見える化する手法
- テスト対象のソースコードの一部に誤りを加え、その誤りをテストで検出できるかを確認する
- 誤りを加えられたコードをミュータントという
- 大量に生成されるミュータントの誤りをどの程度検出できたかで、そのテストの有効性を計測することができる。この有効性の指標をミューテーションスコア(カバレッジ)という。

ミューテーションテストとは



ミューテーションテストのメリット

AGEST

単体テストの品質を見える化することが可能

- カバレッジだけではできなかった単体テスト品質の説明が可能になる

不具合を検出する機会の増加

- コードカバレッジで判別できないテストパターン漏れを検出
  - 漏れているケースを実装・実施する事により新たな不具合を検知

テストコードの品質向上

- 開発者がミューテーションテストに慣れてくると最初からスコアの高い（＝テストパターンの漏れがない）テストを書けるようになる

単体テストコードレビューの効率・問題検知率向上

- 重点的に確認が必要なポイントを絞ることができる
- 手動でのテストコードレビューでは漏れが発生しやすい問題を検知できる

課題と対策

ミューテーションテストの課題と対策

AGEST

課題

- 確認コストが高い
  - 無意味（殺す価値のない、殺せない）なミュータントの判別
    - ログメッセージ等、そもそもテスト対象外としている箇所のミュータント
  - 量が膨大
  - 確認難易度が高い
    - キルされていないことが問題なのか、一目で判断できないものもある
    - プログラミング能力や対象システムへの知識が必要となることもある
- テスト実行時間の増大
  - ミュータントの数だけビルド・テスト実行が必要
  - ミュータントは一行につき複数生成され、結果として膨大な数となる

ミューテーションテストの課題と対策

AGEST

対策

- ミュータントを減らす
  - テストでカバーされている箇所や追加・変更された行にのみミュータント生成
    - そもそもカバーされていない範囲は必ず殺せない
    - 2回目以降のミューテーションテストについては変更箇所のみ(増分)で十分
  - 1行にひとつのミュータント
    - 多少テスト不足数の検出率は下がるが、十分有効
- ミュータントの適用方法を変更する
  - ミュータントスキーマ
    - 一度にすべてのミュータントを埋め込み、実行ごとにフラグで実行するミュータントを決定する。ビルド回数の削減が可能。

まとめ

まとめ

スピードが求められる現在のソフトウェア開発ではより単体テストの評価が重要になる
⇒単体テストの品質を定量的に測定し、改善する必要がある

カバレッジだけではテストの品質は見れない

ミューテーションテストの導入で、実施されているテストの品質を表すことができる、カバレッジだけでは見えない問題を発見する事ができる

AGESTで提供している関連サービス

ミューテーションテスト

テスト自動化

DevOps/アジャイルテスト支援