

# 後方互換性の損失に伴うテストコード変更内容の分類

前川 大樹<sup>†</sup>      伊原 彰紀<sup>‡</sup>      大森 楓己<sup>§</sup>      才木 一也<sup>¶</sup>  
和歌山大学<sup>†</sup>      和歌山大学<sup>‡</sup>      和歌山大学<sup>§</sup>      和歌山大学<sup>¶</sup>

## 1 はじめに

ソフトウェア開発では、開発効率向上のためにライブラリと呼ばれるプログラムを再利用することが多い。新機能追加、バグ修正のためにライブラリの新バージョンがリリースされた場合、ライブラリ利用者は適宜ライブラリのバージョン更新を余儀なくされる。ライブラリの更新には後方互換性を損失する変更（破壊的変更）を含むことがあり、破壊的変更は利用者のプログラムの実行エラーにつながる。そのため、ライブラリ開発者は、適切なバージョン名によって破壊的変更の有無を利用者に伝える。しかし、ライブラリ開発者が破壊的変更を含むことを見逃し、意図せず利用者に被害を与えることも少なくない。

従来研究 [1] では、破壊的変更を含むライブラリの更新では、プログラムの更新と合わせてテストコードも修正すると考え、テストコードの変更有無による破壊的変更の検出手法を提案した。 **TODO: [この文章を書くとき前段落の意図せず（テスト変更しない）を対応できないような...]** しかし、テストコードの変更には、テストの誤り修正や実行手順の修正など、ライブラリ変更とは無関係にテストコードを変更も考えられ、そのような変更は破壊的変更を誤検出する。本研究では、従来手法の誤検出を減らす

ために、テストコード変更内容の分類に取り組み、破壊的変更検出に寄与するテスト変更を明らかにする。

## 2 分析

### 2.1 データセット

本研究では、松田らのデータセット [1] を使用する。データセットには、npm から収集した、人気度が高くテストが整備されているライブラリバージョン 2,111 組が含まれている。このデータセットから、npm に公開され、ライブラリテストに変更があり、テスト変更の diff サイズが 1024\*2 バイト以下であるもの 1,027 組を使用する。対象データ 1,027 組から実際に破壊的変更が含まれるバージョンと含まれないバージョンを 50 組ずつランダムに抽出し計 100 組を分析した。

本研究では、従来研究と同様に、ライブラリ利用者のプログラムがライブラリの新しいバージョンを利用することで、テストが失敗した場合に、ライブラリの新しいバージョンに破壊的変更が加えられたと判断する。

### 2.2 分析手法

100 件のデータについて、2 段階に分けて分析する。

1. テスト変更パターンを分類し出現回数を集計する
2. 1. をもとに、決定木を用いて、破壊的変更に関係するテスト変更パターンについて分析する

1. について、テストプログラムは、複数のテストケースからなるテストスイートが組み

---

A classification of test code changes accompanied by a lack of backward compatibility

<sup>†</sup> Daiki Maekawa, Wakayama University

<sup>‡</sup> Akinori Ihara, Wakayama University

<sup>§</sup> Fuki Omori, Wakayama University

<sup>¶</sup> Kazuya Saiki, Wakayama University

合わさって構成されており、個々のテストケースには、前提となる条件、期待する結果などが記述されている。これらをもとに、ライブラリテストの変更内容を、テストケースの追加、削除、期待する結果の変更など、計 19 種類に分類した。分類をもとに、100 件のライブラリバージョンについて、それぞれの出現回数を集計した。

2. について、破壊的変更の有無に影響を与えるテスト変更パターンを探索するために、実際の破壊的変更の有無を目的変数とし、テスト変更内容 19 種類を説明変数とする決定木分析を行った。

### 2.3 結果

表 1 は、テスト変更パターンと、それぞれの合計件数、実際の破壊的変更の有無を示す。本論文では、19 種類のテスト変更パターンから、決定木分析によって特に破壊的変更有無に影響が高いとわかった 3 件を示す。

#### 等価な変数や関数に置換:

決定木分析のルートノードであり、破壊的変更と関係のある変更という結果となった。しかし、テスト自体の挙動は変わらない変更のため、破壊的変更とは関係のないテスト変更と言える。原因として、データ数が少ないこと、他のテスト変更との組み合わせによるものが考えられる。等価な変数や関数に置換するテスト変更と同時に発生しやすいテスト変更を調査したところ、破壊的変更ありの 11 件中 7 件、テストケースの追加と同時に行為されており、破壊的変更不明の 1 件はテストケースの追加が行われていなかった。テストケースの追加と同時に、等価な変数や関数に置換された時、破壊的変更に関するテスト変更であると言える。

#### テストケースの追加:

テストケースの追加は、全体数が多く、破壊的変更であるものの方が少し多いという結果となった。破壊的変更と関係のある場合は、テスト対象の関数の挙動が変更されたため、それに

対応するテストケースを追加する必要があったと考えられる。関数の振る舞いの変更という破壊的変更に関するテスト変更と言える。破壊的変更と関係のない場合として、テストケースが不足しており、テスト品質向上のため追加したということが考えられる。自動分類の方法として、変更前後でテストケースを記述する関数 (it や test) が追加されているかどうかで捉えることが考えられる。

#### 期待する結果の変更:

期待する結果の変更例として,equals 文や assert 文など、アサーション関数の引数の変更がある。テスト対象の関数の出力内容が変更されたことで、変更前の形式で出力を検証していたテストにも変更が加えられたと考えられる。期待する結果の変更は、関数の出力内容の変更という破壊的変更に関するテスト変更と言える。自動分類の方法として、変更前後の AST を比較することによって引数の変更を捉えることが考えられる。

## 3 おわりに

本研究は、テスト変更の内容分類に取り組み、テスト変更パターンと破壊的変更有無の関係性を決定木により分析した。分析の結果、ライブラリテストの変更パターンとして、テストケースの追加、期待する結果の変更が含まれると、破壊的変更が含まれる可能性が高いとわかった。今後は、テスト変更パターンの自動分類に取り組むとともに、テスト変更内容に基づく破壊的変更の検出手法の精度向上を目指す。

## 謝辞

**TODO:** [謝辞]

## 参考文献

- [1] 松田和輝, 伊原 彰紀, 才木 一也. ライブラリのテストケース変更に基づく後方互換性の実証的分析. 第 28 回ソフトウェ

表 1: ライブラリテスト変更パターンの分類

テスト変更パターン	破壊的変更あり	破壊的変更不明	合計件数
等価な変数や関数に置換	11 件	1 件	12 件
テストケースの追加	57 件	28 件	85 件
期待する結果の変更	27 件	6 件	33 件

ア工学の基礎ワークショップ論文集, pp.  
139-144, 2021.