

# ログイン操作を考慮した修正差分確認手法の拡張

橋浦研究室

122D067 佐野申和

122D123 藤原将也

## 1 はじめに

GitHub [1] を用いたソフトウェア開発では、Issue を通じて欠陥報告や改善提案が共有されているが、記述の不足により欠陥の再現が困難となる場合がある。この問題に対し、高橋ら [2] は、報告者の操作を再現手順ファイルとして記録・共有することで欠陥再現を支援する手法を提案した。TAKAPACHI [2] は、修正前後の Web アプリケーションを二画面で比較することで修正差分の確認を支援する一方、同一のブラウザセッション上で動作するため、ログイン操作を含む欠陥に対しては修正差分確認が困難であるという課題を有している。

## 2 研究目的

本研究の目的は、TAKAPACHI [2] の二画面再現環境におけるログイン処理を含む操作の再現性向上させるための手法を実装した Web アプリケーションテストツールである NOBUKAZU を提案し、その有効性を評価することである。本研究では以下の RQ を設定する。

RQ1: 提案手法はログインを伴う修正差分の確認に有効か

RQ2: 提案手法は既存の Web アプリケーションテストツールと比較して、修正差分確認において広い欠陥検知率を提供できるか

## 3 提案手法

本研究では、修正前後の Web アプリケーションに対して、互いに干渉しない独立したブラウザセッションを用い、同一の再現手順を同時に実行することで、修正差分確認を行う手法を提案する。ここで独立したブラウザセッションとは、認証状態を含む Cookie 等が修正前後で共有されない状態を指す。

欠陥報告者は、Web アプリケーション上で欠陥を再現する操作を行い、その操作を再現手順ファイルとして記録・報告する。NOBUKAZU は、報告された再現手順ファイルを入力として、修正前後の Web アプリケーションに対して共通に利用可能な操作手順へ変換する。この際、異なるポート番号およびブラウザプロファイルを割り当てることで、独立したブラウザセッション上で同時実行を実現する。変換後の 2 画面再現用ファイルを実行することで、修正前後の Web アプリケーションが独立したセッションで起動される。記録された操作の実行後、修正前バージョンにおける欠陥要素を可視化し、修正差分の確認を支援する。ここで図 1 は、お問い合わせ内容に含まれる改行が修正前の画面ではタグとして解釈されている一方で、修正後の画面では改行として正しく解釈され、複数行に分割されて表示されていることが確認できる。

修正前		修正後
お問い合わせ	お問い合わせ	お問い合わせ
イシュー テスト イシュー テスト イシュー テスト	イシュー テスト イシュー テスト イシュー テスト	イシュー テスト

図 1: 2 画面再現用ファイルを用いた修正差分確認の例

## 4 実験対象

本実験では、OSS として開発されている EC-CUBE [3] において報告・クローズされた Issue 約 500 件の中から、欠陥の再現が確認できた 144 件を対象とする。このうち、ログイン操作を伴う欠陥は 95 件である。

対象欠陥は、修正ファイルの内容に基づいてカテゴリ分けを行った。欠陥カテゴリの内訳を表 1 に示す。なお、1 つの欠陥が複数のカテゴリに該当する場合がある。

表 1: 欠陥カテゴリとその内訳

#	欠陥カテゴリ	カテゴリ定義	対象件数
1	データベース	データベース構造などに関連するもの	15
2	開発環境	composer など環境設定に関連するもの	1
3	ロジック	Web アプリの機能や制御に関連するもの	73
4	外観	見た目や表示に関連するもの	90
5	運用	品質保証やテストに関連するもの	38
6	実行環境	稼働環境やサーバー設定に関連するもの	9
7	その他	上記カテゴリに当てはまらないもの	0

## 5 実験方法と評価

実験対象は、EC-CUBE [3] のクローズ済 Issue 500 件から画面表示に関する欠陥を抽出し、最終的に 144 件（うちログインを伴う欠陥 95 件）を評価した。

### 5.1 実験方法

本実験は、以下の 2 段階に分けて実施した。

1. NOBUKAZU および TAKAPACHI [2] により生成した二画面差分用ファイルを用いた修正差分確認
2. AShot [4] および BackstopJS [5] を用いたスクリーンショット取得による修正差分確認

実験 1 では、事前に生成した二画面差分用ファイルを用いて、修正前後の Web アプリケーション上で同一の操作を実行した。実験 2 では、欠陥が確認できるページの URL を入力として画面を取得した。ログイン操作を伴う欠陥については、事前に取得した Cookie 情報を各ツールに設定した。

### 5.2 評価指標

- RQ1: ログイン操作を含む欠陥（95 件）を対象とし、欠陥単位で修正差分確認に到達したかどうか（欠陥検知の成否）を評価する
- RQ2: 全欠陥（144 件）を対象とし、欠陥カテゴリ別に欠陥検知率を算出することで、提案手法と既存手法の適用可能な欠陥範囲の違いを分析する。

ここで、欠陥検知率とは、対象欠陥に対して修正差分確認が成功した割合を指す。

## 6 実験結果と評価

### 6.1 RQ1への回答

ログイン操作を伴う欠陥 95 件を対象とした評価結果を以下に示す。表 2 より、NOBUKAZU はログイン操作を伴う欠陥の一部において修正差分確認を可能とした一方、TAKAPACHI [2] では同条件下での成立事例は確認されなかった。

一方で、提案手法においても、全てのログイン欠陥に対して修正差分確認が可能であったわけではない。修正差分確認に到達できなかった事例には、記録手法の制約に起因する再現不成立や、修正前後の DOM 構造差異により操作対象を特定できなかった事例が含まれている。これらのうち、DOM 構造差異に起因する不成立は提案手法固有の問題ではなく、操作対象を DOM 要素として特定する操作再現型手法全般に内在する制約である。TAKAPACHI [2] ではログイン操作を含む再現が成立しなかったため、これらの制約が結果として顕在化しにくかったといえる。

以上より、提案ツールはログイン操作を含む欠陥に対して一律に修正差分確認を可能とするものではないが、再現手順ファイルの実行が成立する条件下においては、従来手法では到達が困難であったログイン操作を含む欠陥に対しても、修正差分確認を実行可能とする場合があると回答する。

### 6.2 RQ2への回答

評価対象全 144 件に対し、既存の修正差分確認手法の比較結果を表 3 に示す。表 3 より、NOBUKAZU

表 3: 欠陥検知率の比較

#	手法	対象数	差分確認数	欠陥検知率 (%)
1	NOBUKAZU	144	56	38.89
2	TAKAPACHI [2]	144	10	6.94
3	AShot [4]	144	24	16.67
4	BackstopJS [5]	144	24	16.67

は他手法と比較して多くの欠陥に対応していることが確認できる。一方で、いずれの手法においても、全ての欠陥に対して修正差分確認が可能であったわけではなく、手法ごとに到達可能な欠陥の範囲が異なることがわかる。

次に、欠陥カテゴリごとの傾向を把握するため、各手法の欠陥検知率を図 2 に示す。図 2 より、NOBUKAZU はロジックおよび外観カテゴリにおいて、TAKAPACHI [2] と比較して多くの欠陥に対して修正差分確認に到達していることが確認できる。これは、ログイン操作や操作手順を前提とする欠陥に対して、再現手順を用いた自動遷移が有効に機能したためである。一方、実行環境カテゴリでは、記録手法に起因する制約により、修正差分確認に到達できなかった

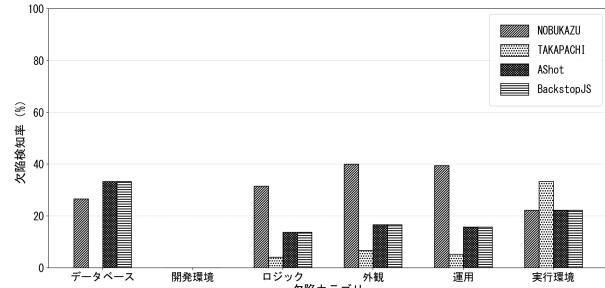


図 2: 欠陥カテゴリごとの欠陥検知率  
事例が含まれている。これらの不成立は、操作対象を DOM 要素として特定する操作再現型手法全般に内在する制約によるものである。

以上より、提案手法 NOBUKAZU は、ログイン操作や特定の操作手順を前提とする欠陥を含め、従来手法では評価が困難であった欠陥に対しても修正差分確認を可能とする場合がある。一方で、記録手法および DOM 構造に依存する制約により、全ての欠陥に適用可能であるわけではない。したがって、提案手法は既存手法と比較して、修正差分確認が可能な欠陥の適用範囲を拡張する手法であると回答する。

## 7 関連研究

Bo ら [6] は、LLM を用いて、バグレポートに含まれる情報の欠落を推測・補完する手法を提案している。一方、本研究の提案ツール NOBUKAZU は、LLM による推論ではなく、欠陥報告者の実際のブラウザ操作を記録することで、実行可能な再現手順を提供する点でアプローチが異なる。

## 8 まとめ

本研究では、Web アプリケーションの修正差分確認におけるセッション干渉の問題に対し、ブラウザプロファイル分離に基づく 2 画面再現手法を提案した。実験の結果、提案手法は、再現手順ファイルの実行が成立する条件下において、従来手法では困難であったログイン操作を含む欠陥に対しても修正差分確認を可能とする場合があることを示した。また、欠陥の性質に応じて、修正差分確認手法を使い分ける必要性を示すとともに、修正差分確認の適用範囲を拡張する一手法を提示した。

### 参考文献

- [1] GitHub, “GitHub,” <https://github.com>, Apr. 2025. (Accessed on 2025/07/10)
- [2] 高橋黎, 横山淳雄, 橋浦弘明, “Web アプリケーションに対する欠陥再現支援手法,” ソフトウェアエンジニアリングシンポジウム 2023 論文集, Vol.2023, pp.39–47, 08 2023.
- [3] EC-CUBE, “EC-CUBE,” <https://github.com/EC-CUBE/ec-cube>, Jul. 2025. (Accessed on 2025/07/10)
- [4] pazone, “ashot,” <https://github.com/pazone/ashot>, Sep. 2025. (Accessed on 2025/10/02)
- [5] garris, “BackstopJS,” <https://github.com/garris/BackstopJS>, Sep. 2024. (Accessed on 2025/10/02)
- [6] Lili Bo, Wangjie Ji, Xiaobing Sun, Ting Zhang, Xiaoxue Wu, and Ying Wei, “ChatBR: Automated assessment and improvement of bug report quality using ChatGPT,” 2024 39th IEEE/ACM International Conference on Automated Software Engineering (ASE) (online), pp.1472–1483, Oct. 2024. <<https://ieeexplore.ieee.org/document/10764843>>