A Reproducibility Study on Predicting Fragility in End-to-End Web Tests

Regina Letícia Santos Felipe
Universidade Federal de Campina Grande
reginaleticia@copin.ufcg.edu.br

I. Introduction

Automated end-to-end (E2E) testing plays a fundamental role in ensuring the functionality and reliability of modern web applications [1]. These tests simulate user interactions through scripts that rely on DOM locators to identify and interact with web elements [2]. However, even minor structural changes in web pages may break these locators, leading to a phenomenon known as test fragility [3].

While several studies in the literature address test fragility by focusing on the automatic repair of broken tests or the creation of more robust locators, the study reproduced in this work, proposed by Di Meglio and Starace, adopts a different approach [4]. The authors aim to define a fragility score, i.e., a metric that quantifies how likely a test is to fail when executed on a future version of the application.

Here's the English translation of the provided text:

—

II. The Proposed Approach

The authors propose an approach that should be easily integrated into any E2E test library/framework. In this work, an implemented integration for the Selenium platform was presented.

The approach estimates test fragility using a **fragility score** that reflects the probability of test failure in future application versions. The approach utilizes an instrumented WebDriver to execute the tests, which includes code activated when a locator identifies an element on a web page. This code analyzes each WebDriver instruction involving a locator to find weaknesses, focusing on Locator Analysis and Web Page Analysis.

A. Locator Analysis

The fragility of E2E tests is closely linked to the robustness of the locators used to identify elements on the page during test execution. The analysis calculates the fragility score of the locators, determined by the number of similar alternative paths in the DOM that a locator can traverse. The formula used to calculate this fragility score is:

$$\text{locator\_score}(P, L) = 1 - \frac{\text{match}(P, L)}{\text{match}(P, \text{clip}(L))}$$

Here, $\text{match}(P, L)$ represents the number of elements on page P that match locator L. $\text{clip}(L)$ is a more generic version of locator L, which removes the first level of the original locator's hierarchy, broadening the selection possibilities.

B. Web Page Analysis

The complexity of the web page is also a determining factor in test fragility. To measure this complexity, the approach considers three main aspects: (i) the depth of the DOM tree; (ii) the branching factor; and (iii) the navigation complexity of the page. The page complexity score is calculated as the average of these three indicators:

$$\text{page\_score}(P) = \frac{\text{depth}(P) + \text{branch}(P) + \text{nav}(P)}{3}$$

In this formula, $\text{depth}(P)$ represents the depth of the page's DOM tree, $\text{branch}(P)$ is the average branching factor of the tree, and $\text{nav}(P)$ is the navigation complexity, calculated by the ratio of links to the total number of elements on the page.

C. Test Fragility Calculation

The fragility of the E2E test, composed of multiple locators, is calculated using the harmonic mean of the fragility scores of all locators present in the test. The final fragility score of a test T is given by the formula:

$$\text{fragility}(T) = \frac{|\text{statements}(T)|}{\sum_{S \in \text{statements}(T)} \frac{1}{\text{score}(S)}}$$

The harmonic mean is used for aggregation because it gives greater weight to statements with lower scores, making the overall fragility more influenced by the weakest links in the test, i.e., the locators with higher fragility. In this formula, $\text{score}(S)$ refers to the fragility score of an individual statement S, and $|\text{statements}(T)|$ represents the number of statements in test T.

III. Methodology

This section details the reproduction of the study, using the authors' provided package [5]. This package contains web application versions, Selenium E2E tests, fragility score algorithms, and result analysis algorithms.

A. Web Application Execution

The web application used was Addressbook, an open-source application implemented in PHP. The different versions of the application were provided in the reproduction package, along with Dockerfile, docker-compose.yml files, and a script for database pre-configuration. These files enabled the local execution of the application for testing. The environment setup

required Docker version 23.0.6 and docker-compose version 1.29.2. During the first attempt at execution, failures arose which were corrected through adjustments, such as adding a command to the Dockerfile for creating missing folders and manually executing the database configuration script. The fixed reproduction package and the detailed execution procedure, including the corrections, are available in the repository: *https://github.com/reginaLeticia/replication_package*.

B. Test Execution

The execution of the study involved conducting tests across various versions of the Addressbook application. The provided replication package contained a project with E2E Selenium test suites (including the original tests for version 1.8.0 and author-corrected tests for version 1.8.6), the instrumented WebDriver, and the classes responsible for implementing the fragility calculation. The project contains a main class that ran the tests using the instrumented WebDriver, along with a configuration file that required specification of the classes for fragility calculation, test suites, and the application versions to be tested.

As outlined in the original experiment described in the article, four test executions were carried out: the 8.1.0 test suite was executed for versions v8.1.0 and v8.1.6, and the 8.1.6 test suite was executed for versions v8.1.6 and v8.1.7. However, several adaptations were necessary during the execution of my reproduction. The first modification addressed the application's language settings. The tests had been originally written in Italian, and the application was configured to automatically select the language. To resolve this, modifications to the application's source code were required, specifically in the addressbook/include/translations.inc.php class, to explicitly set the language to Italian. Another issue encountered was that the configuration file referenced a non-existent class in the *pageAndSelectorScoreStrategy* field. To resolve this, a search of the repository was conducted, and a suitable class implementing the *IPageAndSelectorScoreStrategy* interface was selected.

C. Statistical Calculation Execution

The reproduction package also provides a directory containing the resources necessary for reproducing the statistical calculations. This directory includes a Python file with the algorithm for calculating the point-biserial correlation, as well as files with scores and information generated after the executions, suggesting prior execution by the authors. For the calculation to be performed, the algorithm uses CSV files generated for the test classes, which contain the fragility scores, and the CSV file generated after a pair of executions, indicating which tests passed in each application version.

I carefully selected the score files from a test suite only after its execution for the corresponding application version. For instance, for the 8.1.0 suite, the score files generated after execution for Addressbook version v8.1.0 were chosen, and the same procedure was followed for the 8.1.6 version tests. This precaution was taken to prevent any errors during executions from influencing the fragility score calculation.

After executing the *ease_pointbisserial_correlation.py* file, two CSV files are generated containing the results of the point-biserial correlation, the p-value, the function used for calculating the mean of the scores (harmonic mean), and the weights applied to the selector, page, and page & selector scores, which are 0, 0.5, and 0.5, respectively.

IV. Results

This section presents the findings of the empirical evaluation conducted, including both the original study and my reproduction.

A. Original Study Results

The previous evaluation results, as reported in the original paper, are summarized in Table I.

TABLE I
POINT-BISERIAL CORRELATION COEFFICIENTS AND P-VALUES FROM THE ORIGINAL STUDY

Version pair	Num. of Broken tests	Correlation coeff.	p-value
8.1.0→8.1.6	8	0.64	0.0003
8.1.6→8.1.7	2	0.59	0.0013

For the 8.1.0 → 8.1.6 version pair, 8 tests broke, corresponding to approximately 30% of the complete test suite. The point-biserial correlation analysis confirmed a positive correlation of 0.64. In the second version pair (8.1.6 → 8.1.7), only 2 tests broke, and a point-biserial correlation coefficient of 0.59 was observed. In both scenarios, the computed p-value was significantly low (0.0003 and 0.0013, respectively), indicating statistical significance. These results suggest that tests with higher fragility scores generally correspond to instances of observed breakages.

B. Replication Study Results

My reproduction obtained the following results for the point-biserial correlation analysis:

TABLE II
POINT-BISERIAL CORRELATION COEFFICIENTS AND P-VALUES FROM THE REPRODUCTION STUDY

Version pair	Num. of Broken tests	Correlation coeff.	p-value
8.1.0→8.1.6	8	-0.64	0.00027
8.1.6→8.1.7	21	-0.41	0.03078

For the 8.1.0 → 8.1.6 version pair, 8 broken tests were observed, with a correlation coefficient of approximately -0.64 and a p-value of 0.00027. For the 8.1.6 → 8.1.7 version pair, the number of broken tests was 21, resulting in a correlation coefficient of approximately -0.41 and a p-value of 0.03078. The results indicate that the higher the fragility score, the lower the likelihood of these tests failing in future versions.

V. Discussion

Significant discrepancies were observed when comparing the results. The obtained correlation coefficients displayed opposite signs, and the second execution set showed a difference in the number of errors. For the pair 8.1.0 $\rightarrow$ 8.1.6, the original study indicated a positive correlation of 0.64, while the reproduction found a value of -0.64. Similarly, for the pair 8.1.6 $\rightarrow$ 8.1.7, the original study reported 0.59, whereas the reproduction identified -0.41. In both cases, the p-value indicates statistical significance.

Several hypotheses can be raised to justify these discrepancies. The reproduction package provided by the authors had an inconsistent structure with various limitations that hampered experiment execution, possibly impacting the reproducibility of results. Furthermore, a crucial file for calculating statistical tests was missing and had to be replaced, which may have introduced additional variations.

Another relevant aspect was found when examining the results files provided by the authors in the replication package. It was observed that they contained the same number of errors reported in the original article for each version pair, as well as correlation coefficients of -0.64 and -0.59, corresponding to the values presented in the study but with inverted signs. This suggests the possibility that the author interpreted negative correlations as positive associations between the fragility score and the probability of future failures. To validate this possibility, an analysis of the algorithm used for calculating the correlations was performed. After analyzing the algorithm, it was observed that, given its structure, a negative correlation indicates that the higher the fragility score, the lower the incidence of failures in subsequent versions. This is a relationship that contradicts the logic advocated by the authors of the original study.

These factors present substantial challenges to reproducing the reported results and raise questions about the reliability of the conclusions presented by the original study's authors.

VI. Conclusion

This study aimed to reproduce the work of Di Meglio and Starace, which proposed a metric to predict the fragility of E2E tests in web applications by calculating a fragility score. While the original findings suggested a positive correlation between higher fragility scores and the likelihood of test failures in future versions of the application, my reproduction revealed results with opposite signs, indicating that higher fragility scores were, in fact, associated with fewer failures in subsequent versions.

Several factors may have contributed to these discrepancies, including inconsistencies and gaps in the provided replication package, as well as possible misinterpretations of the correlation signs in the original work. The statistical analysis performed on the replication data showed statistically significant negative correlations, challenging the conclusions presented by the authors of the original study.

These results highlight the importance of rigorous reproducibility efforts in software engineering. Furthermore, they suggest that additional investigations are needed to clarify the true relationship between the calculated fragility scores and the stability of E2E tests as application versions evolve.

References

[1] A. Corazza, S. Di Martino, A. Peron, and L. L. L. Starace, "Web application testing: Using tree kernels to detect near-duplicate states in automated model inference," in *Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2021, pp. 1–6.

[2] M. Niranjanamurthy, R. Arun Kumar, and M. R. Sahana Srinivas, "Research study on web applicationtesting using selenium testing framework," *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 10, pp. 121–126, 2014.

[3] M. Nass, E. Alégroth, R. Feldt, M. Leotta, and F. Ricca, "Similarity-based web element localization for robust test automation," *ACM Transactions on Software Engineering and Methodology*, vol. 32, no. 3, pp. 1–30, 2023.

[4] S. Di Meglio and L. L. L. Starace, "Towards predicting fragility in end-to-end web tests," in *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering*, 2024, pp. 387–392.

[5] S. D. Meglio and L. L. Starace, "Replication package for the paper titled "towards predicting fragility in end-to-end web tests"," 2024. [Online]. Available: https://zenodo.org/records/10165245