Replication of a MBT Test Cases Discard Reduction using Distance Function for industrial setting at TJPB

Carlos D. Q. Lima
diegoquirino@copin.ufcg.edu.br
Federal University of Campina Grande

Everton L. G. Alves
everton@computacao.ufcg.edu.br
Federal University of Campina Grande

Wilkerson L. Andrade
wilkerson@computacao.ufcg.edu.br
Federal University of Campina Grande

ABSTRACT

Model-Based Testing (MBT) comprises approaches where test cases are derived from a model of the system under test. This peculiar aspect can enhance an industrial software agile team automation test facilities by evolving of test cases from product requirements formal model assets (e.g. UML, state machines, or other modeling languages), that typically change by iteration. But in academy literacy there are several issues related to automatize test cases generation that could be potentially damaging for industrial setting, for instance, the MBT test case discard by minor changes operations (e.g. rename or misplaced text) with the loss of traceability. Therefore, it is important to find at most one study that deals or solve the accounted problem, verifying if their results must be replicated to a real-life context. In this paper, we will replicate the work "Reducing the Discard of MBT Test Cases using Distance Functions" to explore its robustness to deal with the test cases discards to the aim for industrial setting at Court of Justice of the State of Paraiba (TJPB). So, from the empirical study conducted into the original research paper, we will replicate its case study's findings under different conditions, with other requirements assets from existing RGP-Diarias TJPB system. In this way, we expect obtaining similar results as the original study, reinforcing its validity and reliability on the reducing of discard of test cases, and enabling the use of its technique for TJPB agile software development teams.

CCS CONCEPTS

• **Software and its engineering → Formal software verification**.

KEYWORDS

MBT, discard reduction, replication, industrial setting, TJPB

ACM Reference Format:
Carlos D. Q. Lima, Everton L. G. Alves, and Wilkerson L. Andrade. 2023. Replication of a MBT Test Cases Discard Reduction using Distance Function for industrial setting at TJPB. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 3 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Model-Based Testing (MBT) is a software testing approach where test cases are derived from formal or semi-formal models, usually requirements ones, like UML use cases and state machines [4]. Consequently, it is an interesting subject of study to industrial agile software development agile teams, while it can improve with a certain degree of automation some manual laboring and error prone tasks to generate test cases, allowing testers to systematically explore different scenarios and interactions of the system, depending just of the correctness of the requirements specification, that typically change by iteration.

However, several issues related to automatize test cases generation could be potentially damaging for industrial setting, for instance, software development teams may end up with obsolete MBT test cases even though edits do not alter the system's behavior, like pure synthetic updates such as, for instance, the replacement of a word with a synonym. In addition, obsolescence is a serious problem in MBT test cases context as it affects the traceability among updates of the test case artifact once a new test case takes its place (or, test cases substitution) [3].

In this context, the primary goal of any agile software development team is to deliver software with quality to end-user without to lose time-to-market. So, it is important to find at most a good study or a good set of tools that deals or solve the accounted problem. This involves keeping abreast of the latest technological advancements, methodologies, and best practices, as well as understanding the evolving needs and expectations of customers.

The objective of this work is to replicate the study "Reducing the Discard of MBT Test Cases using Distance Functions" [1] to explore its robustness to deal with the test cases discards to the aim for industrial setting at Court of Justice of the State of Paraiba (TJPB). To achieve this, we will replicate its case study's findings under different conditions, with other requirements assets from existing RGP-Diarias TJPB system.

Furthermore, the credibility of scientific claims is confirmed through the presence of evidence supporting their reproducibility using new data. So, as result we expect to be capable to reinforce study's validity and reliability on the way of reducing of discard of test cases to enable the use of the technique, achieving success, for TJPB agile software development teams.

2 METHODOLOGY

The methodology applied to this work is the study's **replication** of academic plus industrial paper "Reducing the Discard of MBT Test Cases using Distance Functions" [1]. In this article, the authors reported a series of empirical studies using real industry projects with the aim of analyzing the efficiency of distance functions to

Conference acronym 'XX, June 03–05, 2023, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

reclassify reusable test cases from a subset that would be discarded by an automated test case generation technique.

2.1 Resume of origin study methodology

Initially, the authors **analyzed 28 test cases in 79 system versions and manually classified 518 performed edits**, categorized into two groups: *Low impact* (e.g., replacing words with synonyms and correcting typing errors) - requiring few updates to the test cases; and, *High impact* (e.g., specifying new features and functionality changes) - requiring more updates to the test cases.

Next, for each performed test case edit, which was already manually categorized, they **calculated the following distance functions**: Sørensen-Dice, Cosine, NGram, Jaccard, LCS, Jaro-Winkler, Jaro, OSA, Levenshtein, and Hamming. The goal was to obtain the distance factor corresponding to the change (on a scale from 0 to 1, factors closer to zero indicating that the distance function identified minimal changes to the use case, and closer to one indicating significant changes). The researchers concluded that distance functions can be used to classify low-impact edits, but they do not apply to high-impact edits, once they surely must obsolescence test cases at all. Additionally, they found that statistically, any of the distance functions listed in the study could be used to classify low-impact edits in use cases.

To further identify the optimal configuration for each distance function to correctly and automatically categorize an edit in use cases, three well-known metrics for binary classification were used in several repeated exploratory executions: **Precision**, **Recall**, and **Accuracy** (see details on [1]). **The optimal configuration value for each metric was defined as the intersection point between the precision and recall curves, reflecting a scenario with the lowest number of classification errors** (see Figure **??** with Levenshtein example). *Since all distance functions performed adequately for low-impact edits, a case study was conducted* ***using one of them (Levenshtein)*** *on use cases of a real system to attest to its efficiency in reducing the discard of test cases automatically generated via MBT.*

Finally, the authors **explored out of a total of 1477 MBTs**, 333 test cases were considered new (23%), 724 obsolete (49%), and 420 reused (28%). Considering that in this universe of 724 obsolete test cases, 109 test cases were automatically classified as low impact by using Levenshtein distance function. So, this research concluded that its use could reduce the number of discarded test cases by almost 15%.

2.2 Detailed current work methodology

The replication proposed in this work aims to repeat the research conducted by the researchers (see section 2.1), selecting a new set of data from versions of specified use cases for the RGP-Diarias TJPB system (details on section 3). Therefore, the following steps will be carried out to execute the same experiment design as described in the article:

(1) Translate for CLARET notation [2] all RGP-Diarias current use cases, version by version, with the support of GitHub Tags to detect differences (*diffs*) between claret files in each version.
(2) Manually classify the performed use case edits into low and high impact categories.

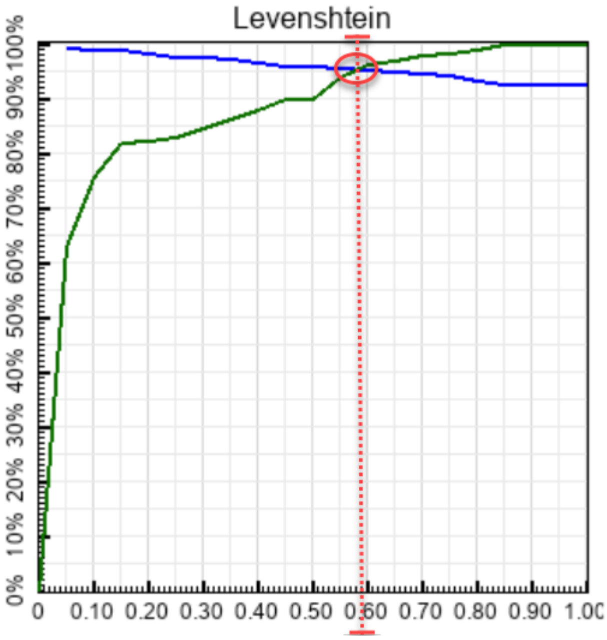

Figure 1: Levenshtein impact threshold from study for precision and recall result of [1]

(3) Perform Levenshtein distance function calculation for each use case edit.
(4) Calculate the metrics Precision, Recall, and Accuracy based on the optimal impact threshold obtained in the origin study for the Levenshtein distance function
(5) Assess the percentage of test cases classified as low impact using the distance functions compared to those deemed obsolete by MBT generation.

Finally, by following these steps, we will be able to evaluate the measures of impact of using distance functions to prevent test cases from being discarded to the aims of the spread use of the technique for TJPB agile software development teams, comparing the current results with those ones widely discussed in the origin study.

3 REPLICATION

To reinforce validity and reliability of the conclusions present in the original study in [1], we considered requirements assets from existing RGP-Diarias TJPB system. RGP-Diarias is an industrial software developed and maintained for TJPB that controls the payment of daily allowances for employees and judges of the court.

Firstly, we translated for CLARET notation [2] all RGP-Diarias's requirements repository and collected all versions of its use case documents and their edits. Table 1 summarizes the collected data.[1]

Then, we manually classified all edits between low and high impact to serve as validation for the automatic classification. Once all distance functions behave similarly in origin study [1], we replicate the use of Levenshtein distance function calculation for each edit. In

[1] Complete tag diffs in branch "reproduction01_distancefunctions" of the open-source repository in https://github.com/diegoquirino/openscience

	Use Cases	Versions	Edits
RGP-Diarias	14	11	57

Table 1: Summary of the artifacts for RGP-Diarias

Figure 2 we compiled totals of high and low impacts classifications from manual and Levenshtein distance function.

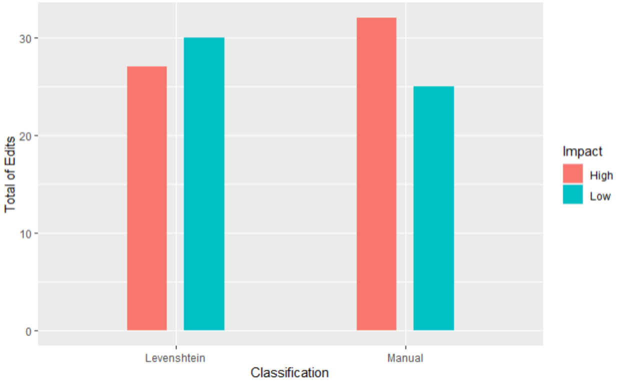

Figure 2: Total of low and high impact Levenshtein and Manual edits

In Figure 3, we summarize compiled results for Levenshtein distance function calculation[2] distribution, using its optimal impact threshold (presented previously in Figure 1).

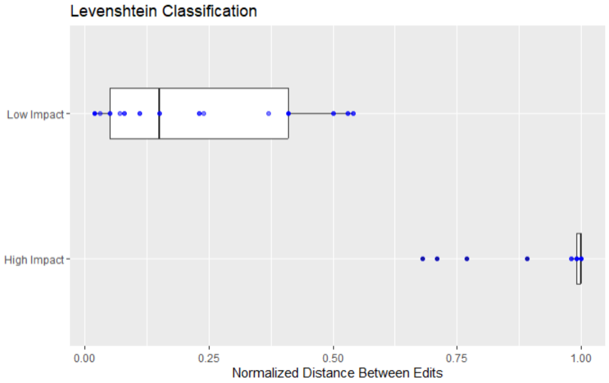

Figure 3: Box-Plot for low and high impact Levenshtein distance values

Consecutively, we calculated Precision, Recall and Accuracy values for the configuration (see compiled results on Table 2), to verify the quality attributes of distribution.

Next step, we evaluate how origin study [1] approach could help reducing test discards through automation. A total of 399 MBT test cases were initially collected from RGP-Diarias TJPB system, where 77 where found NEW (19.3%), 170 OBSOLETE (42.6%), and 152 REUSABLE (38.1%). This counting was a manual task and estimated throughout analysis of each edition, because RGP-Diarias did not originally have its test cases generated through CLARET, over the versions and times.

	Impact Threshold	Precision	Recall	Accuracy
Levenshtein	0.59	73.34%	88.00%	80.71%

Table 2: RGP-Diarias - Evaluating the use of the found impact threshold for Levenshtein function and respective precision, recall and accuracy values

Finally, we analyzed test cases. Typically, "obsolete" test cases - 170 samples (42.6% of total) - would be discarded throughout the development cycles. Once our focus remains on reducing of discard of test cases, now we go into these samples and filtered results considering those test cases that had updated steps classified by automated strategy using Levenshtein distance function as "low impact". From this analysis, 12 test cases where *low impacted*. Although this number seems low (7%), those test cases would be wrongly discarded when in fact they could be easily turned into reusable one.

4 CONCLUSIONS

The use of MBT Test Cases in the software industry context can be a competitive advantage in automating the systematic generation of test cases from requirements. In this regard, this work explored the replication of the results from an academic research [xxxxx] which, as a broad outcome, showed a reduction in the number of test case discards, around 15%, when combined with distance function calculation techniques for each detected step alteration between system versions.

After executing the replicated steps of the methodology, it was possible to obtain slightly lower results, around 7%, but still showing a decrease in the number of automatically discarded test cases. One possible explanation may lie in the data set used, as it involved a conversion of historical test cases, version by version, which reduced the precision, recall, and accuracy rates, respectively.

Finally, With this replication, we reinforce study's validity and reliability on the way of reducing of discard of test cases. Furthermore, we enable the use of the technique for TJPB agile software development teams once it achieved success.

REFERENCES

[1] Thomaz Diniz, Anderson G.F. Silva, Everton L.G. Alves, and Wilkerson L. Andrade. 2019. Reducing the discard of MBT test cases using distance functions. *ACM International Conference Proceeding Series*, 337–346. https://doi.org/10.1145/3350768.3350790

[2] Dalton N. Jorge, Patricia D.L. Machado, Everton L.G. Alves, and Wilkerson L. Andrade. 2018. Integrating requirements specification and model-based testing in agile development. *Proceedings - 2018 IEEE 26th International Requirements Engineering Conference, RE 2018*, 336–346. https://doi.org/10.1109/RE.2018.00041

[3] Anderson G.F. Silva, Wilkerson L. Andrade, and Everton L.G. Alves. 2018. A study on the impact of model evolution in MBT suites. *ACM International Conference Proceeding Series*, 49–56. https://doi.org/10.1145/3266003.3266009

[4] Mark Utting, Alexander Pretschner, and Bruno Legeard. 2012. A taxonomy of model-based testing approaches. *Software Testing, Verification and Reliability* 22 (8 2012), 297–312. Issue 5. https://doi.org/10.1002/stvr.456

Received DD Mmm YYY; revised DD Mmm YYY; accepted DD Mmm YYY

[2] We reused open-source implementations of the distance functions in https://github.com/luozhouyang/python-string-similarity