On the Harmfulness of Test Smells in Manual System Testing: A Controlled Experiment

Gabriela Soares*, Vanessa Santos*, Márcio Ribeiro*, Luana Martins†, Valeria Pontillo‡‖, Manoel Aranda*, Rohit Gheyi§, Ivan Machado¶, Fabio Palomba†
{gpmrs, vanessa, marcio, mpat}@ic.ufal.br, lalmeidamartins@unisa.it, valeria.pontillo@gssi.it, rohit@dsc.ufcg.edu.br, ivan.machado@ufba.br, fpalomba@unisa.it
*Federal University of Alagoas, Maceió, Brazil
†Software Engineering (SeSa) Lab — University of Salerno, Fisciano, Italy
‡Software Languages (SOFT) Lab —Vrije Universiteit Brussel, Belgium
§Federal University of Campina Grande, Brazil
¶ Federal University of Bahia, Salvador, Brazil
‖ Gran Sasso Science Institute (GSSI), L'Aquila, Italy

***Abstract*—Test smells can pose difficulties during testing activities, such as poor maintainability, non-deterministic behavior, and incomplete verification. Existing research has extensively addressed test smells in automated software tests, but little attention has been paid to smells in natural language tests. While some research has attempted to catalog such test smells, there is a lack of investigation into their impact on the effectiveness of test cases. In this paper, we conduct a controlled experiment with 30 participants from academia and industry to examine the impact of test smells in manual test descriptions. Specifically, we analyze whether the presence of two test smells, *Ambiguous Test* and *Eager Action*, result in (1) increased test execution time, (2) a higher number of steps needed to complete the tests, and (3) high divergency on the perceived success of the tests outcomes. Our findings reveal that an *Ambiguous Test* can increase execution time by up to five times and screen flow by up to seven times. In addition, if the *Eager Actions* are dependent on one another, there is no increase in execution time and screen flow.**

***Index Terms*—Test Smells, Manual System Testing, Controlled Experiment, Empirical Software Engineering.**

I. Introduction

System testing is a key activity in software quality assurance, as it verifies the overall behavior and correctness of a system as a whole, ensuring that all integrated components work together as expected [1]. As software systems evolve and undergo continuous modifications during development, system tests must also be easily adaptable to keep pace with these changes. For this reason, automated system testing has received significant attention in years [2]–[4].

Despite this growing interest, manual system testing remains a prevalent practice in many software development context [5]. In manual testing, human testers execute test cases by interacting with the software directly, based on a set of predefined instructions typically written in natural language. These test cases do not rely on scripts or automation tools to verify behavior but instead require testers to read, interpret, and validate expected outcomes themselves. Manual tests are especially common when automation is not feasible due to time, cost, or tooling limitations [5], [6]. In such contexts, the clarity, readability, and maintainability of test cases become even more crucial, as inconsistencies or ambiguities in the written instructions can lead to divergent interpretations and unreliable test outcomes.

Among the numerous factors that influence test code quality, the concept of test smells has gained importance in recent years. Test smells are considered bad decisions to either design or develop the tests [7], [8]. Their presence in the tests may hinder the testing ability to find bugs due to incomplete verifications [9] and non-deterministic behavior [10]. Researchers have also investigated the effects of test smells from different perspectives, particularly in the context of automated software testing. Some studies investigated the diffusion of test smells and their impact on quality attributes and maintainability [11]–[14]. Other studies explored the test code quality from code and mutation coverage [15], [16], as well as its defect- and change-proneness perspectives [17], [18]. Others have proposed solutions to support developers detecting and removing test smells from the test code [19]–[21].

While much of the research has focused on test smells in automated testing, manual testing remains underexplored [5], [22]. Even with recent advances in Artificial Intelligence for detecting natural language test smells [23], [24], our understanding of their effect on manual test execution and interpretation remains limited. To fill this gap, we empirically investigate whether and how the presence of two prevalent natural language test smells, i.e., *Ambiguous Test* and *Eager Action*, influences test execution effectiveness. The former refers to test cases with unclear steps or expectations, causing testers to interpret instructions inconsistently. The latter involves combining multiple actions into a single step, potentially obscuring the root cause of test failures [25]. We performed a controlled experiment involving 30 participants to measure whether these smells lead to increased test execution time, a higher number of steps performed, and a high divergency on the perceived success of the tests outcomes.

Our results show that an *Ambiguous Test* can increase

979-8-3315-9147-2/25/$31.00 © 2025 IEEE.

execution time by up to five times and screen flow by up to seven times. In addition, if the *Eager Actions* are dependent on one another, there is no increase in execution time and screen flow. Moreover, *Ambiguous Test* led to high variability in participants' perception of test success, suggesting that ambiguity can compromise the clarity and reliability of test outcomes. To sum up, our contributions are:

1) Empirical investigation of the harmfulness of test smells in manual tests, shedding light on their impact on testers' productivity and the reliability of testing activities.
2) Quantitative analysis of the effects of *Ambiguous Test* and *Eager Action* smells on test execution, providing evidence of their impact on execution time and screen flow.
3) Contribution to the broader understanding of how test smells influence manual testing processes, expanding literature that has primarily focused on automated testing.

II. Goal and Research Questions

The *goal* of our study is to understand to what extent the *Ambiguous Test* and the *Eager Action* test smells are harmful when executing manual tests. The *purpose* of the study is to evaluate whether the presence of these test smells in manual test descriptions leads to (1) increased time spent executing the tests, (2) a higher number of steps required to complete the tests, and (3) a higher divergency on the perceived success of the the test outcome. The *perspective* is that of both researchers and practitioners interested in understanding how manual test quality affects the performance of testers, with implications for improving testing efficiency and the overall effectiveness of test documentation.

More specifically, our investigation aims to address the following research questions (**RQs**):

> **RQ$_1$.** *To what extent do Ambiguous Test and Eager Action test smells affect test execution time?*

Test smells can introduce inefficiencies in test execution. In particular, the *Ambiguous Test* might cause delays as the tester takes more time to understand the instructions, and the *Eager Action* might cause failures due to unhandled intermediate states. Therefore, we aim to quantify the impact of these test smells on execution time, helping to determine whether reducing such smells leads to more efficient test execution.

> **RQ$_2$.** *To what extent do Ambiguous Test and Eager Action test smells affect the flow of screens during test execution?*

Certain test smells may unintentionally alter the expected screen transition sequence in tests. For instance, *Ambiguous Tests* might lead to incorrect navigation paths due to unclear instructions. Meanwhile, an *Eager Action* might lead testers to skip steps when executing the test case in the user interface. Therefore, **RQ$_2$** investigates whether test smells disrupt the natural flow of screens during test execution, potentially causing unexpected failures or requiring manual intervention.

> **RQ$_3$.** *To what extent do Ambiguous Test and Eager Action test smells lead to divergencies on the perceived success of the tests outcome?*

Test smells can impact the reliability and accuracy of test results. The *Ambiguous Test* might lead to inconsistent test execution, causing false positives or negatives. Differently, the *Eager Action* might result in premature failures that do not reflect real-world user behavior. By investigating how different test smells influence test results such as pass/fail rates, execution inconsistencies, or the need for reruns, **RQ$_3$** aims to evaluate whether test smells lead to inconsistencies when evaluating the tests outcomes.

III. Experiment Design

This section reports our research method. We followed the empirical software engineering guidelines [26] and the *ACM/SIGSOFT Empirical Standards*[1] for the experiment.

A. Planning and Instrumentation

To answer our research questions, we conducted a controlled experiment following a between-groups design [27]. A summary of the main steps to execute the experiment is depicted in Figure 1. In particular, we executed the experiment one participant at a time. In the first step, we asked each participant to sign a consent form (*Pre-survey*). We informed them that their names would not be disclosed and that they could withdraw from the experiment at any time. In the second step, we gave a document containing the definition and examples of manual tests to all participants (*Introduction*). In the third step, we provided four manual tests for each participant and asked them to execute the tests on the `Ubuntu` operating system. Half of the participants received four tests containing smells, while the other half received tests without smells. We recorded the screens to allow us to collect the *execution time* and *screen flow* metrics. After completing the tests, participants filled out a form to indicate whether each test had passed or failed. We also collect data regarding their testing experience and whether their primary occupation is in an industry or academic context.

B. Experimental Materials

Questionnaires. After inviting the participants to take part in the experiment, we sent them an online *pre-survey* via email to assess the participants' knowledge of each technology, enabling us to assign them to appropriate groups. At the end of the experiment, participants completed a *post-survey* questionnaire to gather their feedback.

Hardware and Software. During the experiment, participants accessed the same workstation to carry out the tasks. We used a workstation running Ubuntu Version 24.04 LTS. To monitor and record the experiment, we used the Eye of Gnome 45.3[2] for basic viewing effects (such as zooming

[1] Available at https://github.com/acmsigsoft/EmpiricalStandards.
[2] Available at https://wiki.gnome.org/Apps/EyeOfGnome

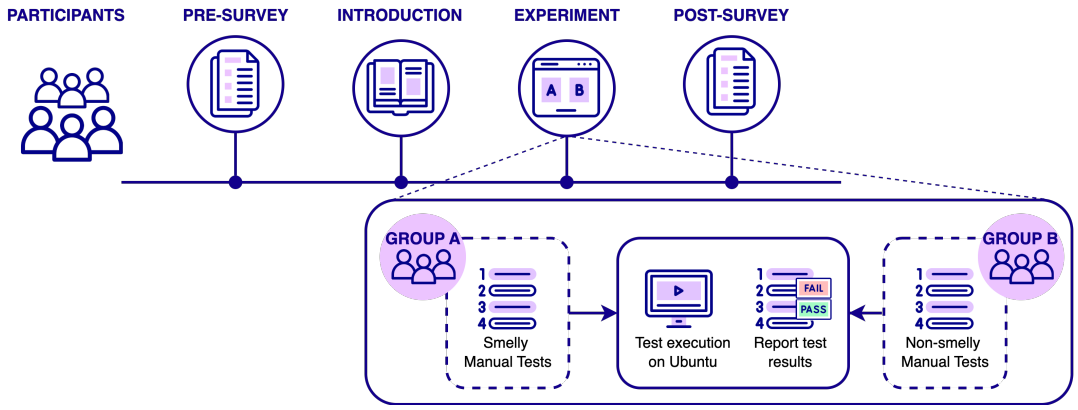

Fig. 1. Controlled experiment design.

and tracking mouse pointer) and `OBS Studio`[3] to record the screen for further analysis.

System under analysis. For our experiment, we focused on a `Linux`-based environment given its open-source nature, which allows full control over the system configuration and facilitates reproducibility. We selected `Ubuntu` because of its widespread adoption, with 79% of Linux users choosing it.[4] Its prominence in both academia and industry ensures familiarity for most participants, minimizing potential bias. Moreover, Ubuntu provides a public repository of manual test cases written in natural language, which served as a realistic and consistent basis for our experimental tasks.[5]

Test smells under analysis. Although several types of test smells have been identified and cataloged [5], [12], [25], [28], we chose test cases containing the *Ambiguous Test* and *Eager Action* as they are highly diffused in manual test cases [22]. They are described as follows.

- ***Ambiguous Test:*** First proposed by Hauptmann et al. [5], this smell refers to an unclear test that allows for multiple interpretations, hindering the understanding of the actions to be performed. A key indicator of this test smell is the use of vague and non-specific terms such as "some," "other", or "any" [5], [22], [25].
- ***Eager Action:*** This test smell occurs when a single step or action encompasses multiple actions. A failure in a test with this smell complicates the developer's ability to understand the cause of the failure [12], [22], [25].

C. Participants

We conducted the experiment in September 2024, individually and in person with each participant. We recruited 30 participants through convenience sampling, including Computer Science students from the Federal University of Alagoas (UFAL), as well as practitioners from the Industrial Foundation of Alagoas (FIEA) and the Federal Justice of Alagoas (JFAL). We selected participants from both academia and industry to enhance the diversity of the sample to capture a broader audience. Although we did not analyze performance by experience level, we ensured that both groups included participants with varying expertise.

Table I shows a summary of the participants' profiles. Of these participants, 67.6% were from academia, and 32.4% were from the industry. The majority of the participants have a B.Sc. degree (88%), while only 12% of the participants have a higher education level. In addition, we also collected the participants' experience with manual test cases and development activities. The answers are categorized in: 0 (no experience); $0 \vdash 1$ (up to 1 year of experience); $1 \vdash 2$ (1 to 2 years of experience); $2 \vdash 3$ (2 to 3 years of experience), and 3+ (3 years or more experience). The participants also play different roles in their jobs, including Data Analyst (DA), Data Engineer (DE), Developer (D), Infrastructure Analyst (InfA), Systems Analyst, Database Admin (DA), Lecturer (L), Project Manager (PM), Product Owner (PO), Researcher (R), and Requirements Analyst (RA), Software Engineer (SE), Student (S).

D. Tasks

The task assigned to participants consisted in the execution of four manual test cases, each provided in two versions: one affected by a test smell and one without it. For the sake of space, we report an example for each smell in the paper — our online appendix [29] contains all four manual test cases used for the experiment in both versions (smelly and non-smelly).

Figure 2 shows a manual test case with the *Ambiguous Test* smell (actions highlighted in red) and its version smell-free (actions highlighted in green). In the smelly version, the action "*Open an OpenDocument presentation (.odp) file containing special characters in your language*" is considered ambiguous, as it creates uncertainty by failing to specify which document the tester should open. This lack of specificity can lead to different testing paths and, consequently, divergent results. To eliminate this ambiguity, the smell-free version of the test specifies the action as "*Open the file '/Documents/-fileTest.odp' containing special characters in your language*"

[3]Available at https://obsproject.com/
[4]Available at: https://canonical.com/blog/ubuntu-20-04-survey-results
[5]Available at: https://launchpad.net/ubuntu-manual-tests

TABLE I
PARTICIPANTS OF OUR EXPERIMENT.

ID	Working on	Experience w/ tests	Experience w/ development	Educational level	Role	Group
P2	Industry	3+	3+	MSc	SE	A
P3	Academia	No experience	$0 \vdash 1$	B.Sc.	SA	B
P5	Academia	$1 \vdash 2$	$1 \vdash 2$	B.Sc.-S	PO, S	B
P6	Academia	$0 \vdash 1$	$1 \vdash 2$	B.Sc.-S	InfA, S	A
P7	Academia	No experience	$0 \vdash 1$	B.Sc.-S	D, S	B
P8	Academia	$0 \vdash 1$	$1 \vdash 2$	B.Sc.-S	D, S	A
P9	Academia	$0 \vdash 1$	3+	B.Sc.	D	B
P10	Industry	No experience	$1 \vdash 2$	B.Sc.-S	DA, S	A
P11	Academia	$1 \vdash 2$	$1 \vdash 2$	B.Sc.	D	B
P12	Academia	$0 \vdash 1$	3+	B.Sc.-S	D, S	A
P13	Academia	3+	3+	MSc	D	B
P14	Industry	$0 \vdash 1$	3+	B.Sc.-S	D, S	A
P15	Industry	No experience	$1 \vdash 2$	B.Sc.-S	DE, S	B
P16	Academia	$1 \vdash 2$	3+	B.Sc.-S	D, S	A
P17	Industry	$0 \vdash 1$	3+	B.Sc.	C	B
P18	Industry	$1 \vdash 2$	3+	B.Sc.	DE	A
P19	Academia	3+	3+	B.Sc.-S	SE, S	B
P20	Academia	$0 \vdash 1$	$1 \vdash 2$	B.Sc.-S	D, S	A
P21	Academia	No experience	No experience	B.Sc.-S	S	B
P22	Industry	$0 \vdash 1$	$0 \vdash 1$	MSc	R	A
P24	Academia	No experience	3+	B.Sc.-S	D, S	A
P25	Academia	No experience	$1 \vdash 2$	B.Sc.	S	B
P26	Industry	3+	3+	B.Sc.-S	D, R, PM, S	A
P27	Industry	$0 \vdash 1$	3+	B.Sc.	D	B
P28	Academia	No experience	No experience	B.Sc.	R	A
P29	Industry	3+	3+	B.Sc.	D	B
P31	Academia	$0 \vdash 1$	$0 \vdash 1$	B.Sc.-S	S	B
P32	Academia	No experience	$1 \vdash 2$	B.Sc.-S	S	A
P34	Academia	No experience	$0 \vdash 1$	B.Sc.-S	R, S	A
P35	Academia	No experience	$0 \vdash 1$	B.Sc.	R	B

Test name: packages/1423_LibreOffice (9)

#	Actions	Verifications
1	Open a OpenDocument presentation (.odp) file containing special characters in your language.	Special characters in your language should display correctly.
2	Select 'Slide Show' and then click on 'Start from First Slide'	The presentation is displayed.

Test name: packages/1423_LibreOffice (9)

#	Actions	Verifications
1	Open the file "~/Documents/fileTest.odp" containing special characters in your language.	Special characters in your language should display correctly.
2	Select 'Slide Show' and then click on 'Start from First Slide'	The presentation is displayed.

Fig. 2. Test #1 with an *Ambiguous Test* in the smelly version (highlighted in red) and without test smells (highlighted in green)

thereby clearly indicating which file should be opened and removing any potential for ambiguity.

Figure 3 shows the manual test case with the *Eager Action* smell (actions highlighted in red) and its version smell-free (actions highlighted in green). In the smelly version of the test, multiple actions are combined into a single step, such as "*Click on the 'Show Apps' icon and launch backup by entering 'backups'*." Combining several actions in one step may cause the tester to lose track during execution, and in the case of a failure, it is difficult to pinpoint which specific action caused the failure. The smell-free version separates each action into individual steps with its own verification.

E. Metrics

Since each test case includes actions that are shared between the smelly and non-smelly versions, we introduce the concept of a *Critical Zone* to focus our analysis. A *Critical Zone* refers specifically to the steps within a test case that exhibit the targeted test smell. For example, in Figure 2, only the first action (line 1) is affected by the *Ambiguous Test* smell; thus, metrics such as execution time and interactions were calculated solely for that step. In contrast, Figure 3 includes multiple smelly actions (lines 2 and 4) interleaved with a non-smelly one (line 3). In this case, we aggregated the metrics associated with the smelly steps only, excluding those related to non-smelly actions. This selective approach ensures a focused and fair comparison between the two versions of each test case.

We calculated the following metrics:

1) **Execution Time** counts the time a participant spent in the *critical zone* to complete the test.
2) **Screen Flow** counts the screen transitions navigated by the participant in the critical zone to complete the test.
3) **Number of Interactions** counts the interactions with elements within the critical zone to complete the test.
4) **Perceived Success** evaluates whether the participant considered the outcome of the test case to be successful.

Test name: packages/1414_Deja-Dup (0)		
#	**Actions**	**Verifications**
1	Create a test folder called 'Test' in your home directory.	Folder “Test” is created.
2	Click on the “Show Apps” icon and launch backup by entering “backups”.	Backup should launch.
3	Click “Preferences”.	The “General” window opens by default.
4	Click 'Folders' tab. Remove all folders from 'Folders to backup' list, then add your test folder to “Folders to backup” list.	'Folders to backup' list contains your test folder.

Test name: packages/1414_Deja-Dup (0)		
#	**Actions**	**Verifications**
1	Create a test folder called 'Test' in your home directory.	Folder “Test” is created.
2	Click on the "Show Apps" icon.	The “Show Apps” window opens.
3	Launch backup by entering ”backups”.	Backup should launch.
4	Click “Preferences”.	The “General” window opens by default.
5	Click "Folders" tab.	The “Folders” window opens.
6	Remove all folders from “Folders to backup” list.	“Folders to backup” list is empty.
7	Add your test folder to “Folders to backup” list.	“Folders to backup” list contains your test folder.

Fig. 3. Test #4 with an *Eager Action* in the smelly version (highlighted in red) and without test smells (highlighted in green)

F. Variables of the Study

Independent variables: The experiment analyzes two independent variables corresponding to the presence of specific test smells. Specifically, test cases are categorized into two groups: those that contain test smells, such as *Ambiguous Test* and *Eager Action*, and those that do not contain any test smells. The experiment aims to analyze how the presence of the specific test smells affects the dependent variables of our study. It is worth highlighting that each test case was intentionally designed to exhibit only one test smell at a time. This decision aligns with the primary goal of the study, which is to understand the individual impact of the *Ambiguous Test* and *Eager Action* smells on the execution of manual tests. In this way, we ensure a focused and controlled analysis, without confounding results from the interaction of multiple smells.

Dependent Variables: The experiment analyzes three dependent variables: (i) *execution time*, (ii) *screen flow*, and (iii) *test execution output*. For $\mathbf{RQ}_1$, the variable is the *execution time*, which is measured as the duration that participants took to execute each test case. This allows us to assess the efficiency of test execution under the influence of test smells, specifically the time taken from start to finish in performing the test activities. For $\mathbf{RQ}_2$, the variable is the *screen flow*, which refers to the sequence and smoothness of transitions between different screens or steps during the test execution. This metric is crucial for understanding how test smells might impact the logical progression and user experience during test execution, possibly indicating delays, confusion, or disruptions in the flow of actions. As for $\mathbf{RQ}_3$, the variable is the *test execution output*, i.e., pass or fail. This variable captures the final result of each test case, helping to measure the effect of test smells on the effectiveness and correctness of the tests. The outcome could reveal whether the presence of test smells leads to divergencies when interpreting the tests outcomes.

Each metric was recorded individually for each task and participant, enabling a detailed analysis of their performance.

G. Experiment design

We employed a between-groups design [27] with two groups: **Group A** is the treatment group that received the smelly test cases, and **Group B** is the control group that received no variable treatment and is used as a reference. Based on that, we assigned participants to the groups, each consisting of 15 participants — the group assignment of each participant is reported in Table I. To balance the groups, we took into account the participants' background experience, making sure that each group had both experienced participants in testing and development (such as P2, P13, P19, P26, and P29) and non-experienced participants (such as P21 and P28).

H. Pilot study

We conducted a two-round pilot study involving a total of 25 pilot testers. The objective of the first round was to assess whether testers, who were not part of the Ubuntu development team, could understand and execute the original smelly test cases. We distributed test cases containing the *Ambiguous Test* and *Eager Action* test smells, and after the test cases were successfully executed, we refactored these test cases to remove the test smells and performed the second round of the pilot study. This round aimed to determine whether the refactored test cases remained executable by the testers and to ensure that the applied refactorings preserved the original intent and functionality of the test cases.

I. Analysis Procedure

We first formulated the working hypotheses that we statistically assessed. As for $\mathbf{RQ}_1$, given the *execution time* metric and the set of test smells (Ts_i) considered in the study, our null hypothesis was the following:

Hn1. There is *no significant difference* in terms of execution time between test cases affected and not by Ts_i.

As for $\mathbf{RQ}_2$, given the number of steps in the *screen data flow* metric and the set of test smells (Ts_i) considered in the study, our null hypotheses were the following:

Hn2. There is *no significant difference* in the screen data flow for test cases affected and not by Ts_i.

Hn3. There is *no significant difference* in the number of interactions for test cases affected and not by Ts_i.

As for **RQ**$_3$, given the number of *test execution output* metric and the set of test smells (Ts_i) considered in the study, our null hypothesis was the following:

Hn4. There is *no significant difference* in test output for test cases affected and not by Ts_i.

If one of the null hypotheses is statistically rejected, we will accept the corresponding alternative hypothesis, namely:

An1. There is a *significant difference* in test execution time for test cases affected and not by Ts_i.

An2. There is a *significant difference* in the screen data flow for test cases affected and not by Ts_i.

An3. There is a *significant difference* in the number of interactions for test cases affected and not by Ts_i.

An4. There is a *significant difference* in test output for test cases affected and not by Ts_i.

Then, we applied statistical tests to verify the working hypotheses, hence accepting or rejecting them.

Statistical tests for RQ$_1$ and RQ$_2$. The *execution time* and *screen flow* metrics measured in these RQs return discrete numerical data. Initially, we performed the *Shapiro-Wilk* [30] to verify the data normality with a significance level of 5%. If the *p-value* returned by the test is less than 5%, the data is not normally distributed. As the data does not follow a normal distribution for any of the metrics, we applied the *Mann-Whitney* test [31] to analyze the hypotheses with a significance level of 5%. If the *p-value* returned by the test is less than 5%, we refute the null hypotheses.

Statistical tests for RQ$_3$. The *output of the test case* metric returns a categorical data, i.e., test pass or fail. We applied the *Chi-Square* test [32] to check if the proportion of successes and failures differs significantly between the tests, with a significance level of 5%. If the *p-value* returned by the test is less than 5%, we refute the null hypotheses.

J. Publication of generated dataset

All artifacts are made available in an online repository [29].

IV. Results

This section presents the results for each **RQ**.

A. Test smells impact on execution time (RQ$_1$)

We collected and analyzed the time spent by the participants in the critical zone to execute each test case. Participants with the smelly version of test cases finish their tasks in 120.38 seconds on average with a standard deviation (sd) of 127.92, indicating a high variability in execution times across participants. Differently, participants with the non-smelly version of test cases finished the execution of their tests in 47.20 seconds on average with a sd of 33.04. In general, test smells can increase execution time by up to 2.5 times.

Figure 4 presents the execution time metric per test case. Regarding the smelly version, while test cases with *Ambiguous Test* present an average execution time of 189.10 seconds with a sd of 147.03, the test cases with *Eager Action* present an

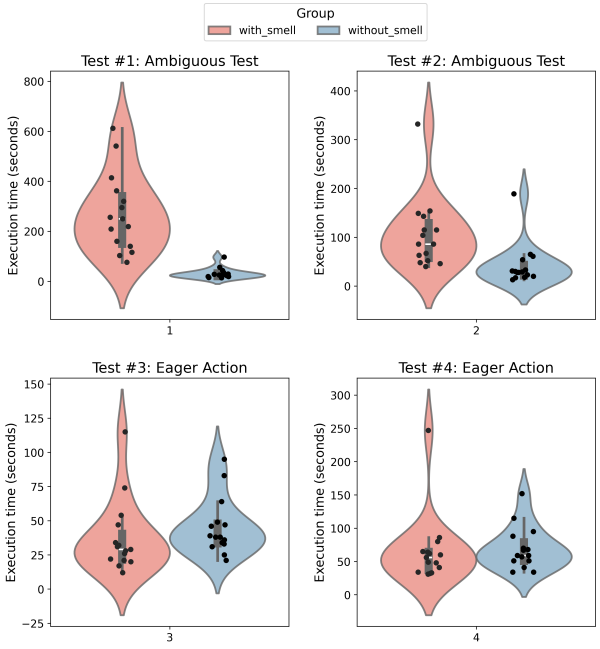

Fig. 4. Violin Plots for the test cases execution with and without test smells.

average execution time of 51.67 seconds with a sd of 43.65. When comparing Test #1 and #2 for the *Ambiguous Test*, the former is more complex, requiring an execution time of 271.53 seconds (sd = 157.63) against 106.67 seconds (sd = 73.27). When comparing Test #3 and #4 for the *Eager Action*, the latter is more complex, requiring an execution time of 65.87 seconds (sd = 52.95) against 37.47 seconds (sd = 26.69). Regarding the non-smelly version, while test cases with *Eager Action* present an average execution time of 57.33 seconds (sd = 29.14), test cases with *Ambiguous Test* present an average execution time of 37.07 seconds (sd = 34.05). Therefore, the smelly version of the test cases with *Ambiguous Test* can increase execution time by up to five times. In contrast, the execution time for *Eager Action* is similar in both versions.

In addition, we investigated whether there was a statistically significant difference in the execution of each test case. The results of the *Mann-Whitney* test show a significant difference for Test #1 (p-value = 4.1e-06) and Test #2 (4.2e-04). Differently, the statistical test does not indicate a significant difference for Test #3 (p-value = 6.8e-02) and Test #4 (2.4e-01). Thus, we refute the null hypothesis (*Hn1*) for the *Ambiguous Test* and we accept it for the *Eager Action*, answering our **RQ**$_1$.

> **Summary**$_1$. *We refute the null hypothesis (**Hn1**) for test cases affected by the Ambiguous Test. In particular, the smelly version of the test cases with Ambiguous Test can increase execution time by up to five times. For Eager Action smell, we do not observe statistically significant differences in the execution time.*

B. Test smells impact on screen flow (RQ$_2$)

We mapped all actions performed by the participants while executing the test case in order to establish a consistent way to

compare the results. For example, Figure 2 shows the Test #1 versions with and without the *Ambiguous Test*. We mapped 31 actions taken by the participants to execute Test #1. We mapped all the actions and established the screen flows as reported in Table II. Similarly, we created ten labels to describe the participants' actions to execute the Test #4 described in Figure 3, which translates to the screen flows in Table III.

TABLE II
MAPPING OF ACTIONS TO DEFINE THE SCREEN FLOWS FOR TEST #1.

ID	Flow	Count	Group	ID	Flow	Count	Group
P2	1, 2, 3, 1, 2, 4, 1, 5, 1, 2, 6, 5	12	A	P3	1, 2, 7	3	B
P6	1, 4, 8, 9, 1, 2	6	A	P5	1, 2, 7	3	B
P8	10, 5, 1, 5, 11, 1, 2, 12, 13, 11, 2, 12	12	A	P7	1, 2, 7	3	B
P10	1, 9, 1, 2	4	A	P9	1, 2, 7	3	B
P12	10, 1, 2, 4, 14, 15, 1, 16, 17, 14, 18, 19, 10, 20, 21, 22, 23, 24	18	A	P11	1, 2, 7	3	B
P14	10, 5, 11	3	A	P13	1, 2, 7	3	B
P16	1, 10, 1, 16, 1, 2, 1, 2, 1, 14, 15, 25, 4, 2, 1, 10, 3, 12, 12, 4, 18, 15, 25, 4, 26	25	A	P15	1, 2, 7	3	B
P18	1, 2, 4, 27, 14, 15, 17, 2, 1, 16	10	A	P17	1, 2, 7	3	B
P20	10, 5, 8, 5	4	A	P19	1, 2, 7	3	B
P22	3, 10, 5, 1, 2, 16, 4, 5	8	A	P21	1, 2, 7	3	B
P24	1, 2, 4, 14, 15, 16, 28, 1, 3, 1, 5, 21, 2, 4, 17	15	A	P25	1, 2, 7	3	B
P26	1, 2, 4, 14, 15, 16, 17, 26, 3, 21, 2, 4, 17, 1, 3, 1	16	A	P27	1, 2, 7	3	B
P28	1, 5, 4, 2, 16, 8, 5, 8, 5, 11	10	A	P29	1, 2, 7	3	B
P32	1, 16, 4, 2, 28, 17, 14, 26, 10, 5, 21, 8, 27, 5, 11	15	A	P31	1, 2, 7	3	B
P34	30, 31, 20, 3, 12	5	A	P35	1, 2, 7	3	B

Labels. 1. Home, 2. Documents, 3. Libreoffice writter, 4. Downloads, 5. Libreoffice impress, 6. Libre office draw, 7. Open filetest.odp, 8. Firefox, 9. Search folder .odp, 10. Show apps, 11. Save document, 12. Open document, 13. Edit document, 14. Videos, 15. Pictures, 16. Recent, 17. Trash, 18. Other locations, 19.Ubuntu, 20. LibreOffice, 21. Open file, 22. Templates, 23. Recent documents, 24. Remote files, 25. Music, 26. Desktop, 27. Search file, 28. Starred, 30. Terminal, 31. Create document.

In general, participants performed a flow composed of 8.05 screens on average, with a standard deviation (sd) of 4.50 for the test cases with test smells, and a flow with 3.75 screens on average, with an sdn of 1.56 screens for test cases without test smells. Concerning test cases with the *Ambiguous Test*, participants took 10.87 screens on average (sd = 6.20) for executing Test #1 and 9.4 screens in average (sd = 2.72) for executing Test #2. In contrast, the participants took an average of 3 screens for executing Test #1 and 2 screens for executing Test #2 without the *Ambiguous Test* (sd = 0). Concerning the test cases with *Eager Action*, participants took, on average, 4.93 screens (sd = 1.53) for Test#3 and 7.00 screens (sd = 3.87) for Test #4. In contrast, the participants took an average of 4 screens (sd = 0.9) for executing Test #3 and 6 screens (sd = 0) for executing Test #4 without the *Eager Test*. Figure 5 shows the screen flow length per test case.

Figure 6 presents the multidimensional scaling plot considering the similarities between the screen flows. Note that the axes represent a mathematical projection that preserves the original distances between data points, but they have no direct interpretation. In particular, Tests #1 and #2 present highly dispersed values for participants executing the version of the test cases with *Ambiguous Test*. Differently, the participants executing the same test cases without test smells followed the exact same screen flow. When analyzing the *Eager Action*, we have an interesting scenario. In Test #3, participants executing the test cases with and without test smells are highly dispersed.

TABLE III
MAPPING OF ACTIONS TO DEFINE THE SCREEN FLOWS FOR TEST #4.

ID	Flow	Count	Group	ID	Flow	Count	Group
P2	1, 2, 3, 4, 5, 6	6	A	P3	1, 2, 3, 4, 5, 6	6	B
P6	1, 2, 3, 4, 5, 6	6	A	P5	1, 2, 3, 4, 5, 6	6	B
P8	1, 2, 3, 10, 6, 10, 1, 2, 3, 10, 1, 2, 3, 5, 6, 7, 7, 8, 9, 8, 9	21	A	P7	1, 2, 3, 4, 5, 6	6	B
P10	1, 2, 3, 4, 5, 6	6	A	P9	1, 2, 3, 4, 5, 6	6	B
P12	1, 2, 3, 4, 5, 6	6	A	P11	1, 2, 3, 4, 5, 6	6	B
P14	1, 2, 3, 4, 5, 6	6	A	P13	1, 2, 3, 4, 5, 6	6	B
P16	1, 2, 3, 4, 5, 6	6	A	P15	1, 2, 3, 4, 5, 6	6	B
P18	1, 2, 3, 4, 5, 6	6	A	P17	1, 2, 3, 4, 5, 6	6	B
P20	1, 2, 3, 4, 5, 6	6	A	P19	1, 2, 3, 4, 5, 6	6	B
P22	1, 2, 3, 4, 5, 6	6	A	P21	1, 2, 3, 4, 5, 6	6	B
P24	1, 2, 3, 4, 5, 6	6	A	P25	1, 2, 3, 4, 5, 6	6	B
P26	1, 2, 3, 4, 5, 6	6	A	P27	1, 2, 3, 4, 5, 6	6	B
P28	1, 2, 3, 4, 5, 6	6	A	P29	1, 2, 3, 4, 5, 6	6	B
P32	1, 2, 3, 4, 5, 6	6	A	P31	1, 2, 3, 4, 5, 6	6	B
P34	1, 2, 3, 4, 5, 6	6	A	P35	1, 2, 3, 4, 5, 6	6	B

Labels. 1. show apps, 2. Type to search, 3. Backup, 4. Folders, 5. Delete folder, 6. Select folder, 7. Forward, 8. Install packages, 9. Authentication, 10. Create your first backup.

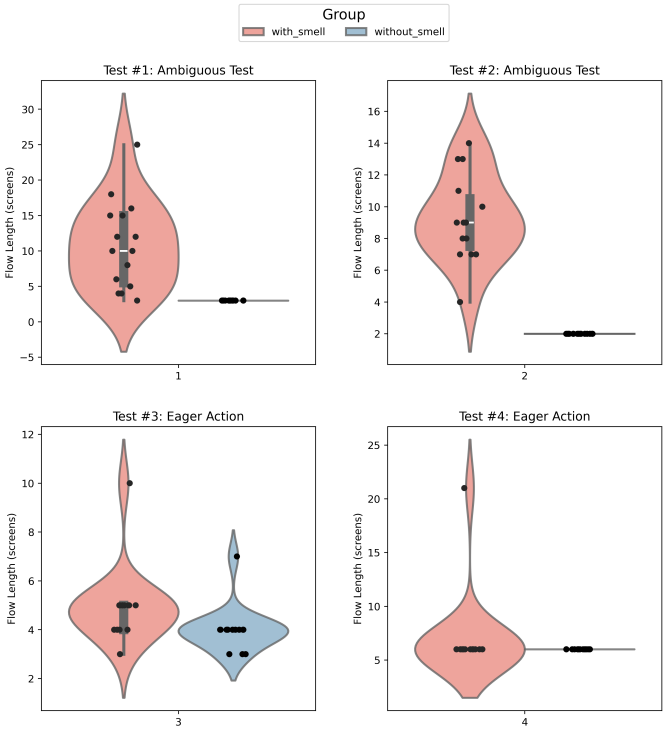

Fig. 5. Violin plots for the screen flow length of test cases with and without test smells.

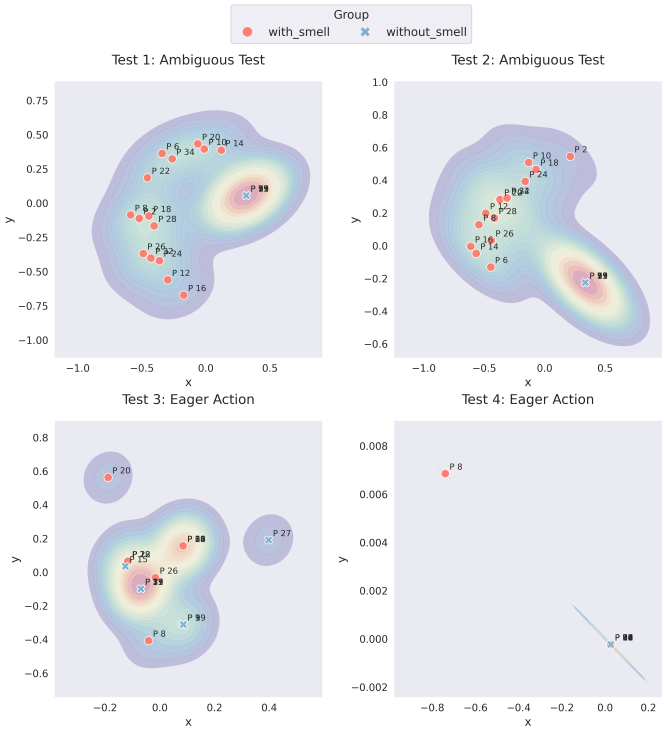

Fig. 6. Multidimensional scaling plot for the screen flow of test cases with and without test smells.

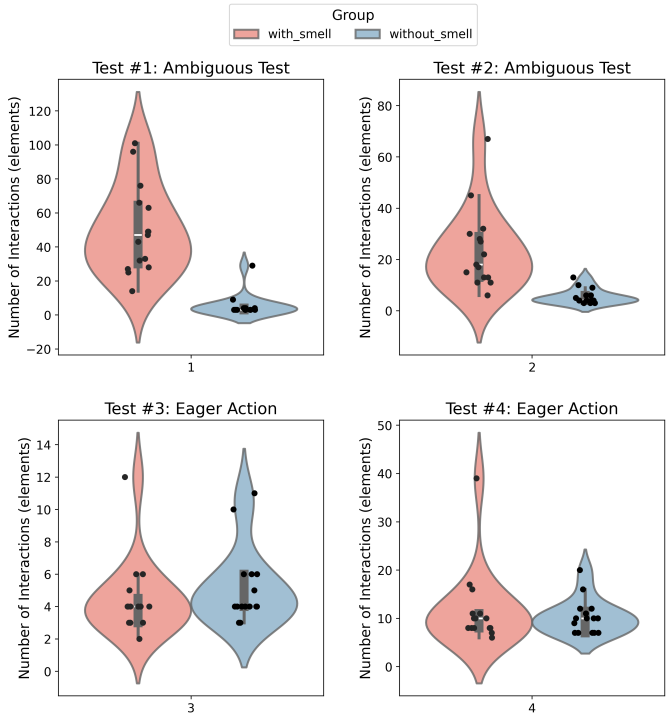

Fig. 7. Violin plots for the interaction of elements for test cases with and without test smells.

In Test #4, with the exception of one participant, all the others followed the same screen flow.

In addition, we investigated whether there is a statistically significant difference in the execution of each test case. Since our data did not have a normal distribution, we performed the *Mann-Whitney* test. The test indicated a significant difference for Test #1 (p-value = 2.5e-06), Test #2 (p-value = 6.6e-07), and Test #3 (p-value = = 4.2e-03). However, there is no significant difference for Test #4 (p-value = = 3.5e-01).

Taking into account the number of interactions within the critical zone, participants interact with 22.46 elements on average (sd = 23.22) for the smelly test cases, and with 6.65 elements on average (sd = 4.66) for non-smelly test cases. More specifically, participants interact with 49.93 elements on average (sd = 25.95) for Test #1, with 23.67 elements on average (sd = 15.78) for Test #2, with 4.47 elements on average (sd = 2.35) for Test #3, and with 11.80 elements in average (sd = 8.13) for Test #4 with test smells. When interacting with test cases without test smells, the participants interact with 5.42 elements on average for Test #1, Tets #2, and Test #3 (sd = 4.33). However, for Test #4, participants interact with 10.33 elements on average (sd = 3.68). Figure 7 summarizes the number of interactions in the critical zone.

Finally, we investigated whether there is a statistically significant difference in the number of interactions within the critical zone for the execution of each test case. The result obtained by the *Mann-Whitney* test indicated a significant difference for Tests #1 (p-value = 5.8e-06) and #2 (p-value = 1.3e-05), i.e., there is a significant difference in the number of interactions for *Ambiguous Test*. Differently, there is no significant difference for Tests #3 (p-value = 2.3e-01) and #4 (p-value = 9.3e-01).

> **Summary$_2$.** *We refute the null hypotheses (**Hn2** and **Hn3**) for test cases affected by the Ambiguous Test. When executing the non-smelly version of the test cases participants followed the same flow, but the flow becomes five times longer when the test case has test smell. In addition, the participants interacted with three times more elements when executing an ambiguous test.*

C. Test smells impact on test outcomes (**RQ$_3$**)

The participants were asked to evaluate whether they perceived the outcome of each test case as successful or not. Most participants considered the execution output of the non-smelly versions of the test cases as successful: 93% of the test cases without the *Ambiguous Test* and 100% of the test cases without the *Eager Action* were perceived as successful. It means that, for the non-smelly test cases, there is low divergency on the perceived success among participants. In contrast, when evaluating the smelly versions of the tests, 44% of the participants executing Test #1 and 60% of the participants executing Test #2 considered the test successful in the presence of the *Ambiguous Test*. When executing test cases with the *Eager Action*, 93% of participants perceived them as successful. It indicates a high divergency on the perceived success for the *Ambiguous Test* but a low divergency for the

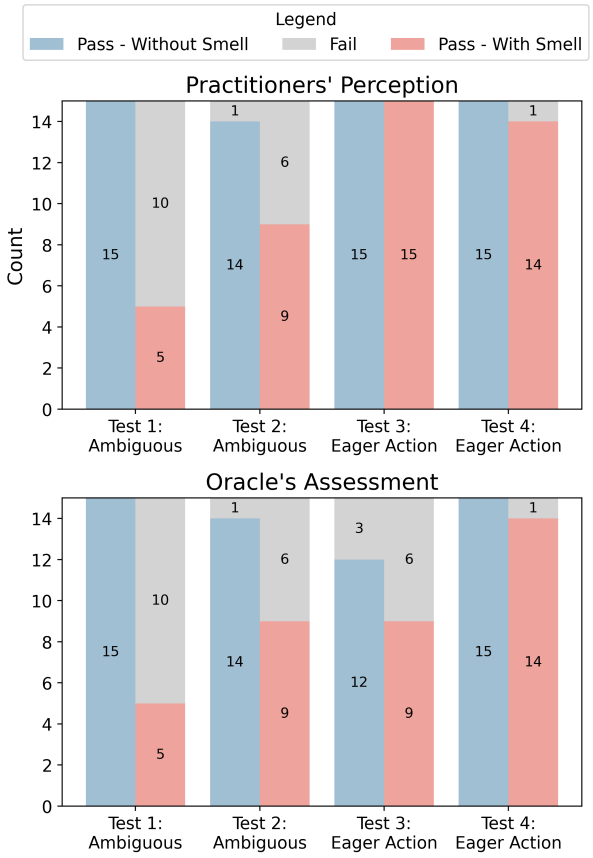

Fig. 8. Stack plot for pass/fail test cases with and without test smells.

Eager Action. Figure 8 shows, in the left side, the participants' perceptions concerning the tests outcomes.

In addition, we investigated whether there is a statistically significant difference in the execution of test case. Since our data is categorical, we performed the *Chi-Square* test. The test indicate a significant difference when considering the *Ambiguous Test* (p-value = 9.3e-0.5), but not for the *Eager Action* (p-value = 0.42). Therefore, we can suppose that the nature of the smell in a test case, particularly the ambiguity, has a stronger influence on how participants evaluate the success of the test, whereas the eagerness factor is not impactful.

Given the subjective nature of the evaluation of test cases success, we analyzed whether the participants' perspective converges to our perspective. This could help in understanding whether test smells might introduce interpretational inconsistencies and not just execution inefficiencies in real-world testing scenarios. Figure 8 shows, in the right side, our evaluation of the the tests outcomes. As we can see, the participants' perception and our evaluation converge for the *Ambiguous Test*. However, we observe a reduction of 20% of passing tests without test smells (Test #3), and a a reduction of 40% of passing tests with test smells (Test #4). We applied again the *Chi-Square* test to verify possible statistically significance differences in the execution of test cases. The results indicate a significant difference for the *Eager Action* (p-value = 0.017).

> **Summary$_3$.** *We refute the null hypothesis (**Hn4**) for test cases affected by the Ambiguous Test and Eager Action. The ambiguity in test cases has a stronger influence on how participants evaluate the success of the test. In addition, Eager Action can lead to inconsistencies among practitioners when interpreting the tests outcomes.*

V. Discussion

Our results suggest that the *Ambiguous Test* smell has a statistically significant negative impact on manual test. In particular, from our study we observed that participants working with smelly versions of these tests exhibited higher execution times, performed more interactions, and navigated significantly longer screen flows. Moreover, the participants perceived the the test outcome as failure, reinforcing the negative effect of the test smell. These results highlight the critical role of clarity in manual tests and suggest that addressing ambiguous instructions should be a priority when improving test quality.

In contrast, the impact of the *Eager Action* smell is less pronounced. We did not observe statistically significant differences in execution time or participants' perceived success of test cases, and the effects on screen flow and interactions were inconsistent. This suggests that the harmfulness of this smell may depend on contextual factors, such as the complexity or interdependence of actions. Further studies should investigate under which conditions this smell becomes more problematic.

Another point of discussion concerns the individual and combined effects of test smells on manual tests. Our study focused exclusively on the individual effects of each test smell, analyzing one smell per test case. As such, we did not evaluate the potential combined or interaction effects of multiple smells. Future research could explore how multiple co-occurring smells influence test effectiveness.

Finally, the high variability observed in participants' performance, especially when experimenting with smelly test cases, highlighted the subjective nature of test smells and their interpretation. This suggests that the impact of certain smells may vary based on the background and experience of individual practitioners. As a result, even when test smells do not consistently degrade objective performance metrics, they may still introduce inconsistencies in how testers perceive and execute the tests, ultimately affecting the reliability and reproducibility of manual testing.

VI. Threats to Validity

This section discusses the potential threats that may affect the validity of our empirical study plan [33].

Construct validity. A first threat concerns the criteria used to select the subject system and manual test cases. We selected test cases from the `Ubuntu Manual Tests` repository, a publicly available and widely adopted resource, ensuring realistic and representative scenarios for manual testing. Another threat refers to the set of test smells evaluated in our study. We selected *Ambiguous Test* and *Eager Action* because they are among the most common and empirically validated test

smells in the context of manual test descriptions [22], ensuring alignment with the literature. Nonetheless, further study should investigate and evaluate different test smells in manual tests.

Internal Validity. We mitigated the *Learning effect* threat by applying a between-group design, i.e., participants only perform one of the tasks. With this design, we could also mitigate *Fatigue* threat as the participants spent half of the time performing only one of the tasks instead of the two tasks. Finally, a threat concerns the *Human error* as we intervened to assign participants to tasks and balance the set of participants, which could have introduced bias or error into the study.

External Validity. Our conclusions are based on a sample of 30 participants, equally split between two groups. On the one hand, the number of participants involved in our controlled experiment is comparable to other empirical studies in the software engineering literature [34], and aligns with commonly adopted sample sizes for controlled experiments in this field [33]. On the other hand, our participants have the characteristics described in Section III-C, representing a diverse group from both academia and industry, with varying levels of experience in software development and testing. Their profiles reflect typical roles involved in manual testing activities, such as developers, analysts, and software engineers. As such, we believe that our findings can be meaningfully applied to similar testing contexts, particularly where manual test execution remains prevalent. Nonetheless, replications of our study would strengthen the validity of our conclusions. For this reason, we have made all experimental materials publicly available in an online appendix [29].

Additionally, the study was conducted in a controlled environment, which may not reflect the real-world situations developers face while working on software projects. This may limit the applicability of the findings to real-world scenarios. Regarding complexity, the tasks used in the study may not accurately represent the complexity and diversity of test smells that developers encounter in real-world projects, which may limit the external validity of the findings.

To ensure the collection of all relevant information, we followed the guidelines provided by Charness et al. [27] to define our between-group study. Additionally, we conducted a pilot study with 25 participants to fix possible biases and flaws before conducting the real study.

Conclusion validity. A potential threat of this category concerns the choice of the statistical tests employed in the study. The choice of non-parametric tests was guided by the results of the *Shapiro-Wilk* test [30], which indicated that our data did not follow a normal distribution. Therefore, we selected *Mann-Whitney* and *Chi-Square* [31], [32].

VII. Related Work

Bavota et al. [11] conducted a controlled experiment involving twenty master's students, with the aim of analyzing whether the presence of test smells affects the comprehensibility of source code during software maintenance. The results show that most test smells have a strong negative impact on the understandability of test suites and production code. In a later study, Bavota et al. [12] conducted an experiment in which participants performed different program comprehension tasks on test suites with and without test smells and measured participants' performance using both correctness and time spent performing a task. The results showed that test comprehension is 30% better in the absence of test smells. In our work, we also carried out a controlled experiment to see how harmful test smells are, but unlike most work that focuses on automated software, our analysis focused on manual tests written in natural language.

When focusing on manual tests, Hauptman et al. [5] introduced a set of test smells for natural language tests and defined metrics to automatically detect these smells in natural language tests using static analysis. In addition, the authors performed an empirical study to validate the metrics for natural test smells detection, and to quantify the extent of the smells in real-world test suites. In a similar work, Soares et al. [22] performed an exploratory study in manual tests to propose a catalog of test smells for natural language and their respective identification rules. Then, the authors validated the catalog with 24 in-company test engineers. Based on the previous catalogs of natural language test smells, Rajkovic and Enoiu [35] presented a tool called NALABS to detect smells in requirements and test specifications. In contrast, our work aims to investigate the effects of proposed natural language test smells from literature rather than listing new ones.

VIII. Concluding Remarks

In this paper, we investigated the impact of two natural language test smells, *Ambiguous Test* and *Eager Action*, on the effectiveness of manual system testing by analyzing the number of interactions, screen flow, and the execution time. Our findings show that *Ambiguous Test* can significantly increase execution time and screen flow, highlighting the importance of clear, unambiguous test instructions in ensuring efficient and accurate testing. Additionally, while each *Eager Action* is independent of another, they still impact execution time, as developers may skip some steps or fail to follow the intended sequence, leading to inefficiencies in test execution.

As future work, we plan to conduct experiments with diverse participant groups to validate and expand upon our findings. Additionally, other potential test smells could be explored to further understand their impact on manual testing.

Acknowledgment

This study was financed by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) grants 403719/2024-0, 310313/2022-8, 404825/2023-0, 443393/2023-0, 312195/2021-4, 315840/2023-4, 403361/2023-0, Fundação de Amparo a Pesquisa do Estado de Alagoas (FAPEAL) grant 60030.0000002340/2022, Fundação de Apoio à Pesquisa do Estado da Paraíba (FAPESQ-PB) grant 268/2025, Fundação de Amparo a Pesquisa do Estado da Bahia (FAPESB) grant PIE0002-2022.

REFERENCES

[1] M. Pezzè and M. Young, *Software testing and analysis: process, principles, and techniques*. John Wiley & Sons, 2008.

[2] E. Borjesson and R. Feldt, "Automated system testing using visual gui testing tools: A comparative study in industry," in *2012 IEEE Fifth International Conference on Software Testing, Verification and Validation*. IEEE, 2012, pp. 350–359.

[3] M. Hanna, A. E. Aboutabl, and M.-S. M. Mostafa, "Automated software testing framework for web applications," *International Journal of Applied Engineering Research*, vol. 13, no. 11, pp. 9758–9767, 2018.

[4] C. U. Baytar, "Model proposal for testing websites in multiple browsers: Case of selenium test tool," *Topkapı Sosyal Bilimler Dergisi*, vol. 1, no. 2, pp. 105–119, 2022.

[5] B. Hauptmann, M. Junker, S. Eder, L. Heinemann, R. Vaas, and P. Braun, "Hunting for smells in natural language tests," in *ICSE 2013*, 2013, pp. 1217–1220.

[6] M. D. L. C. Peixoto, D. de Medeiros Baia, N. Nascimento, P. Alencar, B. Fonseca, and M. Ribeiro, "On the effectiveness of llms for manual test verifications," 2024. [Online]. Available: https://arxiv.org/abs/2409.12405

[7] E. Soares, M. Ribeiro, R. Gheyi, G. Amaral, and A. Santos, "Refactoring test smells with junit 5: Why should developers keep up-to-date?" *IEEE Transactions on Software Engineering*, vol. 49, no. 3, pp. 1152–1170, 2023.

[8] E. Soares, M. Ribeiro, G. Amaral, R. Gheyi, L. Fernandes, A. Garcia, B. Fonseca, and A. Santos, "Refactoring test smells: A perspective from open-source developers," in *SAST 2020*, 2020, p. 50–59.

[9] D. Spadini, M. Schvarcbacher, A.-M. Oprescu, M. Bruntink, and A. Bacchelli, "Investigating severity thresholds for test smells," in *MSR 2020*, 2020, p. 311–321.

[10] G. Meszaros, *XUnit Test Patterns: Refactoring Test Code*. USA: Prentice Hall PTR, 2006.

[11] G. Bavota, A. Qusef, R. Oliveto, A. D. Lucia, and D. Binkley, "Are test smells really harmful? an empirical study," *Empirical Software Engineering*, vol. 20, no. 4, pp. 1052–1094, May 2014. [Online]. Available: https://doi.org/10.1007/s10664-014-9313-0

[12] G. Bavota, A. Qusef, R. Oliveto, A. De Lucia, and D. Binkley, "Are test smells really harmful? an empirical study," *Empirical Software Engineering*, vol. 20, pp. 1052–1094, 2015.

[13] F. Palomba, D. Di Nucci, A. Panichella, R. Oliveto, and A. De Lucia, "On the diffusion of test smells in automatically generated test code: An empirical study," in *Proceedings of the 9th International Workshop on Search-Based Software Testing*. New York, NY, USA: ACM, 2016, p. 5–14.

[14] L. Martins, H. Costa, and I. Machado, "On the diffusion of test smells and their relationship with test code quality of java projects," *Journal of Software: Evolution and Process*, vol. 36, no. 4, p. e2532, 2024.

[15] T. Virginio, L. Martins, L. Rocha, R. Santana, A. Cruz, H. Costa, and I. Machado, "Jnose: Java test smell detector," in *Proceedings of the 34th Brazilian Symposium on Software Engineering*. New York, NY, USA: ACM, 2020, p. 564–569.

[16] L. Martins, V. Pontillo, H. Costa, F. Ferrucci, F. Palomba, and I. Machado, "Test code refactoring unveiled: where and how does it affect test code quality and effectiveness?" *Empirical Software Engineering*, vol. 30, no. 1, pp. 1–39, 2025.

[17] D. Spadini, F. Palomba, A. Zaidman, M. Bruntink, and A. Bacchelli, "On the relation of test smells to software code quality," in *2018 IEEE international conference on software maintenance and evolution (ICSME)*. New York, NY, USA: IEEE, 2018, pp. 1–12.

[18] A. Qusef, M. O. Elish, and D. Binkley, "An exploratory study of the relationship between software test smells and fault-proneness," *IEEE Access*, vol. 7, pp. 139 526–139 536, 2019.

[19] G. Lopes, D. Romão, E. Soares, M. Ribeiro, G. Amaral, R. Gheyi, and I. Machado, "A road to find them all: Towards an agnostic strategy for test smell detection," in *Proceedings of the XXIII Brazilian Symposium on Software Quality*, ser. SBQS '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 231–241. [Online]. Available: https://doi.org/10.1145/3701625.3701662

[20] L. Martins, H. Costa, M. Ribeiro, F. Palomba, and I. Machado, "Automating test-specific refactoring mining: A mixed-method investigation," in *SCAM 2023*, 2023, pp. 13–24.

[21] R. Melo, P. Simões, R. Gheyi, M. d'Amorim, M. Ribeiro, G. Soares, E. Almeida, and E. Soares, "Agentic slms: Hunting down test smells," 2025. [Online]. Available: https://arxiv.org/abs/2504.07277

[22] E. Soares, M. Aranda, N. Oliveira, M. Ribeiro, R. Gheyi, E. Souza, I. Machado, A. Santos, B. Fonseca, and R. Bonifácio, "Manual tests do smell! cataloging and identifying natural language test smells," in *ESEM 2023*, 2023, pp. 1–11.

[23] K. Lucas, R. Gheyi, E. Soares, M. Ribeiro, and I. Machado, "Evaluating large language models in detecting test smells," 2024. [Online]. Available: https://arxiv.org/abs/2407.19261

[24] K. Lucas, R. Gheyi, M. Ribeiro, F. Palomba, L. Martins, and E. Soares, "Investigating the performance of small language models in detecting test smells in manual test cases," 2025. [Online]. Available: https://arxiv.org/abs/2507.13035

[25] M. Aranda, N. Oliveira, E. Soares, M. Ribeiro, D. Romão, U. Patriota, R. Gheyi, E. Souza, and I. Machado, "A catalog of transformations to remove smells from natural language tests," in *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering*, ser. EASE '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 7–16. [Online]. Available: https://doi.org/10.1145/3661167.3661225

[26] A. Jedlitschka, M. Ciolkowski, and D. Pfahl, "Reporting experiments in software engineering," *Guide to advanced empirical software engineering*, pp. 201–228, 2008.

[27] G. Charness, U. Gneezy, and M. A. Kuhn, "Experimental methods: Between-subject and within-subject design," *Journal of economic behavior & organization*, vol. 81, no. 1, pp. 1–8, 2012.

[28] L. Martins, T. A. Ghaleb, H. Costa, and I. Machado, "A comprehensive catalog of refactoring strategies to handle test smells in java-based systems," *Software Quality Journal*, vol. 32, no. 2, pp. 641–679, 2024.

[29] G. Soares, V. Santos, M. Ribeiro, L. Martins, V. Pontillo, M. Aranda, R. Gheyi, I. Machado, and F. Palomba. (2025) Dataset – on the harmfulness of test smells in manual tests. [Online]. Available: https://doi.org/10.6084/m9.figshare.28769603

[30] H. Thornburg. (2001, Mar.) Introduction to bayesian statistics. [Online]. Available: http://ccrma.stanford.edu/~jos/bayes/bayes.html

[31] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," in *The annals of mathematical statistics*, vol. 18. Michigan: JSTOR, 1947, pp. 50–60.

[32] R. A. Fisher, "Statistical methods for research workers," *biometrics*, p. 110, 1925.

[33] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. New York, NY: Springer Science & Business Media, 2012.

[34] H. Frluckaj, L. Dabbish, D. G. Widder, H. S. Qiu, and J. HERBSLEB, "Gender and participation in open source software development," *Proceedings of the ACM on Human-Computer Interaction*, vol. 6, no. CSCW2, pp. 1–31, 2022.

[35] K. Rajkovic and E. P. Enoiu, "Nalabs: Detecting bad smells in natural language requirements and test specifications," Mälardalen Real-Time Research Centre, Mälardalen University, Tech. Rep., February 2022. [Online]. Available: http://www.es.mdu.se/publications/6382-