Empirical Software Engineering (2020) 25:2844–2896
https://doi.org/10.1007/s10664-020-09805-y

On (Mis)perceptions of testing effectiveness: an empirical study

Sira Vegas[1] · Patricia Riofrío[1] · Esperanza Marcos[2] · Natalia Juristo[1]

Published online: 7 May 2020

Abstract

A recurring problem in software development is incorrect decision making on the techniques, methods and tools to be used. Mostly, these decisions are based on developers' perceptions about them. A factor influencing people's perceptions is past experience, but it is not the only one. In this research, we aim to discover how well the perceptions of the defect detection effectiveness of different techniques match their real effectiveness in the absence of prior experience. To do this, we conduct an empirical study plus a replication. During the original study, we conduct a controlled experiment with students applying two testing techniques and a code review technique. At the end of the experiment, they take a survey to find out which technique they perceive to be most effective. The results show that participants' perceptions are wrong and that this mismatch is costly in terms of quality. In order to gain further insight into the results, we replicate the controlled experiment and extend the survey to include questions about participants' opinions on the techniques and programs. The results of the replicated study confirm the findings of the original study and suggest that participants' perceptions might be based not on their opinions about complexity or preferences for techniques but on how well they think that they have applied the techniques.

Keywords Developers perceptions · Testing technique effectiveness · Software testing

1 Introduction

An increasingly more popular practice nowadays is for software development companies to let developers choose their own technological environment. This means that different developers may use different productivity tools (programming language, IDE, etc.). How-

Communicated by: Per Runeson

✉ Sira Vegas
svegas@fi.upm.es

[1] Universidad Politécnica de Madrid, Madrid, Spain

[2] Universidad Rey Juan Carlos, Madrid, Spain

ever, software engineering (SE) is a human-intensive discipline where wrong decisions can potentially compromise the quality of the resulting software.

In SE, decisions on which methods, techniques and tools to use in software development are typically based on developers' perceptions and/or opinions rather than evidence, as suggested by Dybå et al. (2005) and Zelkowitz et al. (2003). However, empirical evidence might not be available, as certain methods, techniques or tools may not have been studied within a particular setting or even at all. Alternatively, developers may simply not be acquainted with such studies, according to Vegas and Basili (2005). On this ground, it is important to discover how well developers perceptions (beliefs) match reality and, if they do not, find out what is behind this mismatch, as noted by Devanbu et al. (2016).

According to Psychology, experience plays a role in people's perceptions. This has also been observed by Devanbu et al. (2016) in SE. However, this research sets out to discover how well matched perceptions are with reality in the absence of previous experience in the technology being used. This makes sense for several reasons: 1) experience is not the only factor affecting developers' perceptions; 2) development teams are usually composed of a mix of people with and without experience; and 3) it is not clear what type of experience influences perceptions. For example, Dieste et al. (2017) conclude that academic rather than professional experience could be affecting the external quality of the code generated by developers when applying Test-Driven Development.

We aim to study whether perceptions about the effectiveness of three defect detection techniques match reality, and if not, what is behind these perceptions. To the best of our knowledge, this is the first paper to empirically assess this issue.

To this end, we conducted an empirical study plus a replication with students. During the original study we measured (as part of a controlled experiment) the effectiveness of two testing techniques and one code review technique when applied by the participants. We then checked the perceived most effective technique (gathered by means of a survey) against the real one. Additionally, we analysed the cost of the mismatch between perceptions and reality in terms of loss of effectiveness. Major findings include:

- Different people perceive different techniques to be more effective. No one technique is perceived as being more effective than the others.
- The perceptions of 50% of participants (11 out of 23) are wrong.
- Wrong perception of techniques can reduce effectiveness 31pp (percentage points) on average.

These findings led us to extend the goal of the study in a replication to investigate what could be behind participants' perceptions. To do this, we examined their opinions on the techniques they applied and the programs they tested in a replication of the controlled experiment. Major findings include:

- The results of the replication confirm the findings of the original study.
- Participants think that technique effectiveness depends exclusively on their performance and not on possible weaknesses of the technique itself.
- The opinions about technique complexity and preferences for techniques do not seem to play a role in perceived effectiveness.

These results are useful for developers and researchers. They suggest:

- Developers should become aware of the limitations of their judgement.

- Tools should be designed that provide feedback to developers on how effective techniques are.
- The best combination of techniques to apply should be determined that is at the same time easily applicable and effective.
- Instruments should be developed to make empirical results available to developers.

The material associated to the studies presented here can be found at https://github.com/GRISE-UPM/Misperceptions.

The article is organised as follows. Section 2 describes the original study. Section 3 presents its validity threats. Section 4 discusses the results. Section 5 describes the replicated study based on the modifications made to the original study. Section 6 presents its validity threats. Section 7 reports the results of this replicated study. Section 8 discusses our findings and their implications. Section 9 shows related work. Finally, Section 10 outlines the conclusions of this work.

2 Original Study: Research Questions and Methodology

2.1 Research Questions

The main goal of the original study is to assess whether participants' perceptions of their testing effectiveness using different techniques are good predictors of real testing effectiveness. This goal has been translated into the following research question:

RQ1: Should participants' perceptions be used as predictors of testing effectiveness?

This question was further decomposed into:

- *RQ1.1: What are participants' perceptions of their testing effectiveness?*
  We want to know if participants perceive a certain technique as most effective than the others.
- *RQ1.2: Do participants' perceptions predict their testing effectiveness?*
  We want to assess if the technique each participant perceives as most effective is the most effective for him/her.
- *RQ1.3: Do participants find a similar amount of defects for all techniques?*
  Choosing the most effective technique can be difficult if participants find a similar amount of defects for two or all three techniques.
- *RQ1.4: What is the cost of any mismatch?*
  We want to know if the cost of not correctly perceiving the most effective technique is negligible and depends on the technique perceived as most effective.
- *RQ1.5: What is expected project loss?*
  Taking into consideration that some participants will correctly perceive their most effective technique (mismatch cost 0), and others will not (mismatch cost greater than 0), we calculate the overall cost of (mis)match for all participants in the empirical study and check if it depends on the technique perceived as most effective.

2.2 Study Context and Ethics

We conducted a controlled experiment where each participant applies three defect detection techniques (two testing techniques and one code review technique) on three different programs. For testing techniques, participants report the generated test cases, later run a set of test cases that we have generated (instead of the ones they created), and report the failures found.[1] For code reading they report the identified faults. At the end of the controlled experiment, each participant completes a questionnaire containing a question related to his/her perceptions of the effectiveness of the techniques applied. The course is graded based on their technique application performance (this guarantees a thorough application of the techniques).

The study is embedded in an elective 6 credits Software Verification and Validation course. The regular assessment (when the experiment does not take place) is as follows: students are asked to write a specification for a program that can be coded in about 8 hours. Specifications are later interchanged so that each student codes a different program from the one (s)he proposed. Later, students are asked to perform individually (in successive weeks) code reading, and white-box testing on the code they wrote. At this point, each student delivers the code to the person who wrote the specification, so that each student performs black-box testing on the program (s)he proposed. Note that this scenario requires more effort from the student (as (s)he is asked to write first a specification and then code a program, and these tasks do not take place when the study is run). In other words, the students workload during the experiment is smaller than the workload of the regular course assessment. The only activity that takes place during the experiment that is not part of the regular course is answering the questionnaire, which can be done in less than 15 minutes. Although the study causes changes in the workflow of the course, its learning goals are not altered.

All tasks required by the study, with the exception of completing the questionnaire, take place during the slots assigned to the course. Therefore, there is no additional effort for the students but attending lectures (which is mandatory in any case).

Note that the students are allowed to withdraw from the controlled experiment, but this would affect their score in the course. But this also happens when the experiment is not run. If a student misses one assignment, (s)he would score 0 in that assignment and his/her course score would be affected consequently. However, they are allowed to withdraw from the study without penalty in their score, as the submission of the questionnaire is completely voluntary. No incentives are given to those students who submit the questionnaire.

The fact of submitting the questionnaire implies giving consent for participating in the study. Students are aware this is a voluntary activity aiming for a research, but they can also get feedback. Those students who do not submit the questionnaire, are not considered in the study in any way, as they are not giving consent to use their data. For this reason, they will not be included in the quantitative analysis of the controlled experiment (even though their data is available for scoring purposes).

The study is performed in Spanish, as it is the participants' mother tongue. Its main characteristics are summarised in Table 1.

2.3 Constructs Operationalization

Code evaluation technique is an experiment **factor**, with three treatments (or levels): equivalence partitioning (EP)—see Myers et al. (2004), branch testing (BT)—see Beizer (1990), and code reading by stepwise abstraction (CR)—see Linger (1979).

[1]This has been done for learning purposes, as we have noticed that students sometimes do not report failures that are exercised by test cases. Since this is a learning goal of the course, not relevant for the study, we measure it separately, and do not use it here.

Table 1 Description of the experiment

Aspect	Value
Factor	Code evaluation techniques
Treatments	Equivalence partitioning
	Branch testing
	Code reading by stepwise abstraction
Response variables	Technique effectiveness
	Perception of effectiveness
	Mismatch cost
Design	3 period crossover for 3 treatments
	First training then operation
	6 training sessions of 2 hours each
	4 hours of individual practice with techniques
	3 experimental sessions of 4 hours each
Experimental Objects	3 (cmdline, nametbl, ntree)
	C programming language
	2 versions
	7 faults injected
Participants	32
	Fifth-(final) year undergraduate CS students
	Experienced in programming
	Experienced with C
	Trained in SE
	Little or none professional experience
	No testing experience

The **response variables** are *technique effectiveness*, *perception of effectiveness* and *mismatch cost*. Technique effectiveness is measured as follows:

- For EP and BT, it is the percentage of faults exercised by the set of test cases generated by each participant. In order to measure the response variable, experimenters execute the test cases generated by each participant.[2]
- For CR, we calculate the percentage of faults correctly reported by each participant (false positives are discarded).

Note that dynamic and code review techniques are not directly comparable as they are different technique types (dynamic techniques find failures and code review techniques find faults). However, the comparison is fair, as:

- Application time is not taken into account, and participants are given enough time to complete the assigned task.
- All faults injected are detectable by all techniques. Further details about faults, failures and their correspondence is given in Section 2.5.

[2]Note that that it is not possible to take measurements on the failures reported by participants, as they do not run their own test cases, but the ones we have given them.

Table 2 Measuring mismatch cost

	Observed effectiveness			Perceived		
Participant	CR	BT	EP	Most effective	Match	Mismatch cost
P_i	57.14%	0%	83.33%	CR	No	26pp
P_j	42.86%	85.71%	100%	EP	Yes	0pp
P_k	57.14%	85.71%	85.71%	BT	Yes	0pp

Perception of effectiveness is gathered by means of a questionnaire with one question that reads: *Using which technique did you detect most defects?*[3]

Mismatch cost is measured, for each participant, as the difference between the effectiveness obtained by the participant in the technique (s)he perceives as most effective and the most effective in reality for him/her. Note that participants neither know the total amount of seeded faults, nor which techniques are best for their colleagues or themselves. This operationalization imitates the reality of testers –who lack such knowledge in real projects. Therefore, the perception is fully subjective (and made in relation with the other two techniques).

Table 2 shows three examples that show how mismatch cost is measured. Cells in grey background show the technique for which highest effectiveness is observed for the given participant.

The first row shows a situation where the participant perceives as most effective CR, but the most effective for him/her is EP. In this situation, there is a mismatch (misperception) and the associated cost is calculated as the difference in effectiveness between CR and EP. The second row shows a situation where the participant correctly perceives EP as the most effective technique for him/her. In this situation there is a match (correct perception) and therefore, the associated mismatch cost is 0pp. The third row shows a situation where the participant perceives BT as the most effective technique for him/her, and BT and EP are tied as his/her most effective technique. In this situation we consider that there is a match (correct perception), and therefore, the associated mismatch cost is 0pp.

2.4 Study Design

Testing techniques are applied by human beings, and no two people are the same. Due to the dissimilarities between the participants already existing prior to the experiment (degree of competences achieved in previous courses, innate testing abilities, etc.), there may exist variability between different participants applying the same treatment. Therefore, we opted for a *crossover* **design**, as described by Kuehl (2000) (a within-subjects design, where each participant applies all three techniques, but different participants apply the techniques in a different order) to prevent dissimilarities between participants and technique application order from having an impact on results. The design of the experiment is shown in Table 3.

The **experimental procedure** takes place during seven weeks, and is summarised in Table 4. The first three weeks there are training sessions in which participants learn how to apply the techniques and practice with them. Training sessions take place twice a week

[3] During the training, definitions for the terms error, fault and failure are introduced. Additionally, participants are explained that the generic term *defect* is used to refer to both faults and failures indistinctly.

Table 3 Experimental design

Program		Ntree			Cmdline			Nametbl		
Period		Day 1			Day 2			Day 3		
Technique	N	CR	BT	EP	CR	BT	EP	CR	BT	EP
Group 1	6	X	–	–	–	X	–	–	–	X
Group 2	5	X	–	–	–	–	X	–	X	–
Group 3	5	–	X	–	–	–	X	X	–	–
Group 4	5	–	X	–	X	–	–	–	–	X
Group 5	5	–	–	X	X	–	–	–	X	–
Group 6	6	–	–	X	–	X	–	X	–	–

(Tuesdays and Thursdays) and each one lasts 2 hours. Therefore, training takes 12 hours (2 hours/session x 2 sessions/week x 3 weeks). Participants are first taught the code review technique, then white-box and finally black-box. The training does not follow any particular order, but the one we have found best to meet the learning objectives of the course.

The following week there are no lectures, and students are asked to practice with the techniques. For this purpose, they are given 3 small programs in C (that contain faults), and are asked to apply a given technique on each program (all students apply the same technique on the same training program). The performance on these exercises is used for grading purposes.

The other three are experiment execution weeks. Each experiment execution session takes place once a week (Fridays) and lasts four hours. This is equivalent to there being no time limit, as participants can complete the task in less time. Therefore, experiment execution takes 12 hours (4 hours/session x 1 session/week x 3 weeks). Training sessions take place during lecture hours and experiment execution sessions take place during laboratory hours. Those weeks in which there are lectures, there is no laboratory and vice versa. The time used for the controlled experiment is the corresponding one assigned to the course in which the study is embedded. No extra time is used.

In each session, participants apply the techniques and, for equivalence partitioning and branch testing, run test cases too. They report application of technique, and generated test cases and failures (for the testing techniques) or faults (for the code review technique).

At the end of the last experiment execution session (after applying the last technique), participants are surveyed about their perceptions of the techniques that they applied. They must return their answer before the following Monday, to guarantee that they remember as much as possible about the tasks performed.

Table 4 Experimental procedure

Type	Training			Free	Operation		
Week	W1	W2	W3	W4	W5	W6	W7
Activity	CR	BT	EP	Exercises	ntree	cmdline	nametbl
N. Sessions	2	2	2	–	1	1	1
Session duration	2h	2h	2h	–	4h	4h	4h
Total time	4h	4h	4h	4h	4h	4h	4h

2.5 Experimental Objects

Program is a **blocking variable**. It is not a factor, because the goal of the experiment is not to study the programs, but the code evaluation techniques. However, it is a blocking variable, because we are aware that programs could be influencing results. The experiment has been designed to cancel out the influence of programs. Every participant applies each technique in a different program, and each technique is applied on different programs (by different participants). Additionally, the program by technique interaction is later analysed.

The experiment uses three similar *programs*, written in C (used in other empirical studies about testing techniques like the ones performed by Kamsties and Lott 1995 or Roper et al. 1997):

- *cmdline*: parser that reads the input line and outputs a summary of its contents. It has 239 executable LOC and a cyclomatic complexity of 37.
- *nametbl*: implementation of the data structure and operations of a symbol table. It has 230 executable LOC and a cyclomatic complexity of 27.
- *ntree*: implementation of the data structure and operations of an n-ary tree. It has 215 executable LOC and a cyclomatic complexity of 31.

Appendix A shows a complete listing of the metrics gathered by the PREST[4] tool (Kocaguneli et al. 2009) on the correct programs (before faults were injected). Although the purpose of the programs is different, we can see that most of the metrics obtained by PREST are quite similar, except Halstead metrics, which are greater for ntree. At the same time, cmdline is slightly larger and more complex than the other two.

Each program has been seeded with seven faults (some, but not all, are the same faults as used in previous experiments run on these programs), and there are 2 versions of each faulty program. All faults are conceptually the same in all programs (eg., a variable initialisation is missing). Some faults occurred naturally when the programs were coded, whereas others are typical programming faults. All faults:

- Cause observable failures.
- Can be detected by all techniques.
- Are chosen so that the programs fail only on some inputs.
- No fault conceals another.[5]
- There is a one-to-one correspondence between faults and failures.

Note, however, that it is possible that a participant generates two (or more) test cases that exercise the same seeded fault, and therefore produce the same failure. Participants have been advised to report these failures (the same failure exercised by two or more different test cases) as a single one. For example, there is a fault in program ntree in the function in charge of printing the tree. This causes the failure that the tree is printed incorrectly. Every time a participant generates a test case that prints the tree (which is quite often, as this function is useful to check the contents of the tree at any time), the failure will be shown.

Some examples of the seeded faults and their corresponding failures are:

- Variable not initialised. The associated failure is that the number of input files is printed incorrectly in cmdline.

[4] Available at http://code.google.com/p/prest/

[5] One of the versions in one of the programs contains only six faults. Due to a mistake we made, one of the failures was concealed by another.

– Incorrect boolean expression in a decision. The associated failure is that the program does not output error if the second node of the “are siblings” function does not belong to the tree.

2.6 Participants

The 32 **participants** of the original study were fifth-(final)year undergraduate computer science students taking the elective Software Verification and Validation course at the Universidad Politécnica de Madrid. The students have gone through 2 courses on Software Engineering of 6 and 12 credits respectively. They are trained in SE, have strong programming skills, have experience programming in C, have participated in small size development projects,[6] and have little or no professional experience. So, they should not been considered unexperienced in programming, but good proxys of junior programmers.

They have not formal training in any code evaluation techniques (including the ones involved in the study), as this is the course in which they are taught them. Since they have had previous coding assignments, they might have done testing previously but informally. As a consequence, they might have acquired some intuitive knowledge on how to test/review programs (developing their own techniques or procedures that could resemble the techniques), but they have never learned the techniques formally. They have never been required to do peer-reviews in coding assignments, or write test cases in the projects where they have participated. They could possibly have used assertions or informal input validation, but on their own (never under request, and have not previously been taught how to do it).

All participants have a homogeneous background. The only differences could be due to the level of achievement of learning goals in previous courses, or innate ability for testing. The former could have been determined by means of scores in previous courses (which was not possible). The latter was not possible to measure. Therefore, we have not deemed necessary to do some kind of blocking, and just performed simple randomisation.

Therefore, the sample used represents developers with little or no previous experience on code evaluation techniques (novice testers). The use of our students is appropriate in this study on several grounds:

– We want to rule out any possible influence of previous experience on code evaluation techniques. Therefore, participants should not have any preconceived ideas or opinions about the techniques (including having a favourite one).
– Falessi et al. (2017) suggest that it is easier to induce a particular behaviour among students. More specifically, reinforce a high level of adherence to the treatment by experimental subjects applying the techniques.
– Students are used to make predictions during development tasks, as they are continually undergoing assessment on courses related with programming, SE, networking, etc.

Having said that, since our participants are not practitioners, their opinions are not based on any previous work experience on testing, but on their experience on informally testing programs for some years (they are in 5th year of a 5-year CS bachelor). Additionally, as part of the V&V training, our participants are asked to practice in small programs with the techniques used in the experiment.

[6]They have participated in development projects in teams (as in the Artificial Intelligence, Compiler and Operating Systems courses).

Table 5 Statistical tests used to answer research questions

RQ	Statistical test	Test description	Checks
1.1	χ^2 goodness-of-fit	How well a theoretical distribution fits a empirical distribution	All 3 techniques are equally frequently perceived as most effective
1.2	Cohen's kappa	Agreement for categorical variables when 2 raters classify different objects	Match between perceptions and reality
	Stuart-Maxwell	Changes in the proportion among raters' agreement (marginal homogeneity)	Bias in perceptions
	McNemar-Bowker	Symmetry of the associated contingency table	Bias in perceptions
1.3	Krippendorff's alpha	Agreement for variables when N raters classify different objects	Participants find similar amount of defects for techniques/programs
1.4	1-way ANOVA or	Compare three or more groups	Different techniques show same
1.5	Kruskall-Wallis		mismatch cost/project loss

According to Falessi et al. (2017), we (SE experiments) tend to forget practitioners' heterogeneity. Practitioners have different academic backgrounds, SE knowledge and professional experience. For example, a developer without a computer science academic background might not have knowledge about testing techniques. We assume that for this exploratory study, the characteristics of the participants are a valid sample for developers that have little or no experience on code evaluation techniques and are junior programmers.

2.7 Data Analysis

The analyses conducted in response to the research questions, are explained below.[7] Table 5 summarises the statistical tests used to answer each research question. First we report the analyses (descriptive statistics and hypothesis testing) of the controlled experiment.

To examine *participants' perceptions* (RQ1.1), we report the frequency of each technique (percentage of participants that perceive each technique as the most effective). Additionally, we determine whether all three techniques are equally frequently perceived as being the most effective. We test the null hypothesis that the frequency distribution of the perceptions is consistent with a discrete uniform distribution, i.e., all outcomes are equally likely to occur. To do this, we use a chi-square (χ^2) goodness-of-fit test.

To examine if *participants' perceptions predict their testing effectiveness* (RQ1.2), we use Cohen's kappa coefficient along with its 95% confidence interval—calculated using bootstrap. Cohen's kappa coefficient (κ) is a statistic that measures agreement for qualitative (categorical) variables when 2 raters are classifying different objects (units). It is calculated on the corresponding contingency table generated. Table 6 shows an example of a contingency table. Cells contain the frequencies associated to each pair of classes.

[7] All analyses are performed using IBM SPSS v26.

Table 6 Example of contingency table

		Rater 1		
		Class A	Class B	Class C
Rater 2	Class A	X_{AA}	X_{AB}	X_{AC}
	Class B	X_{BA}	X_{BB}	X_{BC}
	Class C	X_{CA}	X_{CB}	X_{CC}

It is used to calculate Kappa, and perform Stuart-Maxwell's and McNemar-Bowker's tests

Table 7 Interpretation of kappa values

Kappa	Landis and Koch (1977)	Altman (1991)	Fleiss and Levin (2003)
0	Poor	Poor	Poor
0.01–0.20	Slight		
0.21–0.40	Fair	Fair	
0.41–0.60	Moderate	Moderate	Fair to good
0.61–0.75	Substantial	Good	
0.76–0.80			Excellent
0.81–1.00	Almost perfect	Very good	

Negative values are interpreted like positive values, but meaning disagreement instead of agreement

Table 8 Example of collapsed contingency table

		Rater 1	
		Class A	Other
Rater 2	Class A	X_{AA}	$X_{AB}+X_{AC}$
	Other	$X_{BA}+X_{CA}$	$X_{BB}+X_{BC}+X_{CB}+X_{CC}$

It is used to calculate partial kappa

Kappa is generally thought to be a more robust measure than simple percent agreement calculation, since it takes into account the agreement occurring by chance. It is not the only coefficient that can be used to measure agreement. There are others, like Krippendorff's alpha, which is more flexible, as can be used in situations where there are more than 2 raters, or the response variable is in an interval or ratio scale. However, in our particular situation, where there are 2 raters, data in nominal scale and no missing data, Kappa behaves similarly to Krippendorff's alpha (Banerjee et al. 1999; Zapf et al. 2016).

Kappa is a number from -1 to 1. Positive values are interpreted as agreement, while negative values are interpreted as disagreement. There is still some debate about how to interpret kappa. Different authors have categorised detailed ranges of values for kappa that differ with respect to the degree of agreement that they suggest (see Table 7). According to scales by Altman (1991) and Landis and Koch (1977), 0.6 is the value as of which there is considered to be agreement. Fleiss and Levin (2003) lower this value to 0.4. Each branch of science should establish its kappa value. As there are no previous studies that specifically address the issue of which is the most appropriate agreement scale and threshold for SE, and different studies in SE have used different scales,[8] we use Fleiss et al.'s more generous scale as our baseline.

We measure the agreement between the technique perceived as most effective by a participant, and the most effective technique for that participant for all participants. Therefore, we have 2 raters (perceptions and reality), three classes (BT, EP and CR), and as many units to be classified as participants.

Since there could be agreement for some but not all techniques, we also measure kappa for each technique separately (kappa per category), following the approach described in Everitt (2000). It consists of collapsing the corresponding contingency table. Table 8 shows the collapsed contingency table for Class A from Table 6. Note that a collapsed table is always a 2x2 table.

In the event of disagreement, we also study the type of mismatch between perceptions and reality—whether the disagreement leads to some sort of bias in favour of any of the techniques. To do this, we use the respective contingency table to run Stuart-Maxwell's test of marginal homogeneity (testing the null hypothesis that the distribution of preferences match reality) and the McNemar-Bowker test for symmetry (testing the null hypothesis of symmetry) as explained in Everitt (2000). The hypothesis of marginal homogeneity corresponds to equality of row and column marginal probabilities in the corresponding contingency table. The test for symmetry determines whether observations in cells situated symmetrically about the main diagonal have the same probability of occurrence. In a 2x2 table, symmetry and marginal homogeneity are equivalent. In larger tables, symmetry implies marginal homogeneity, but the reciprocal is not true.[9]

Since we have injected only 7 defects in each program, there exists the possibility that if no agreement is found between perceptions and reality, it could be due to the fact that *participants find a similar amount of defects for all three (or pairs of) techniques* (RQ1.3). If this is the case, then it would be difficult for them to choose the most effective technique. To check this, we will run agreement on the effectiveness obtained by participants using different techniques. Therefore we have 3 raters (techniques) and as many units as participants. This will be done with all participants, and with participants in the same experiment group, for every group; for all techniques, and for pairs of techniques. Note that kappa can

[8] For example, Octaviano et al. (2015) use Landis & Koch, but Massey et al. (2015) use Fleiss et al. as we do.

[9] For this reason, we need to check both.

no longer be used, as we are seeking agreement on interval data. For this reason, we will use Krippendorff's alpha (Hayes and Krippendorff 2007) along with its 95% confidence interval—calculated using bootstrap, and the KALPHA macro for SPSS.[10]

To examine the *mismatch cost* (RQ1.4) and *project loss* (RQ1.5), we report the cost of the mismatch (when it is greater than zero for RQ1.4 and in all cases for RQ1.5), associated with each technique as explained in Section 2.3. To discover whether there is a relationship between the technique perceived as being the most effective and the mismatch cost and project loss, we apply a one-way ANOVA test or a medians Kruskall-Wallis test for normal and non-normal distributions, respectively along with visual analyses (scatter plots).

3 Original Study: Validity Threats

Based on the checklist provided by Wohlin et al. (2014), the relevant threats to our study are next described.

3.1 Conclusion Validity

1. *Random heterogeneity of participants*. The use of a within-subjects experimental design ruled out the risk of the variation due to individual differences among participants being larger than the variation due to the treatment.

3.2 Internal Validity

1. *History and maturation*:
   - Since participants apply different techniques on different artefacts, learning effects should not be much of a concern.
   - Experimental sessions take place on different days. Given the association of grades to performance in the experiment, we expect students will try to do better on the following day, causing that the technique applied the last day gets a better effectiveness. To avoid this, different participants apply techniques in different orders. This way we cancel out the threat due to order of application (avoiding that a given technique gets benefited from the maturation effect). In any case, an analysis of the chosen techniques per day is done to study maturation effect.
2. *Interactions with selection*. Different behaviours in different technique application groups are ruled out by randomly assigning participants to groups. However, we will check it analysing the behaviour of groups.
3. *Hypothesis guessing*. Before filling in the questionnaire, participants in the study were informed about the goal of the study only partially. We told them that we wanted to know their preferences and opinions, but they were not aware of our research questions. In any case, if this threat is occurring, it would mean that our results for perceptions are the best possible ones, and therefore would set an upper bound.
4. *Mortality*. The fact that several participants did not give consent to participate in the study has affected the balance of the experiment.

[10]Retrieved from: http://afhayes.com/spss-sas-and-mplus-macros-and-code.html

Table 9 Balance before and after submitting the questionnaire in the original study

Group	G1	G2	G3	G4	G5	G6
Initial	6	5	5	5	5	6
Final	5	4	5	3	4	2

5. *Order of Training*. Techniques are presented in the following order: CR, BT and EP. If this threat had taken place, then CR would be the most effective (or their favourite).

3.3 Construct Validity

1. *Inadequate preoperational explanation of cause constructs*. Cause constructs are clearly defined thanks to the extensive training received by participants on the study techniques.
2. *Inadequate preoperational explanation of effect constructs*. The question being asked is totally clear and should not be subject to possible misinterpretations. However, since the perception is subjective, there exists the possibility that the question asked is interpreted differently by different participants, and hence, perceptions are related to how the question is interpreted. This issue should be further investigated in future studies.

3.4 External Validity

1. *Interaction of setting and treatment*. We tried to make the faults seeded in the programs as representative as possible of reality.
2. *Generalisation to other subject types*. As we have already mentioned, the type of subjects our sample represents are developers with little or none previous experience in testing techniques and junior programmers. The extent to which the results obtained in this study can be generalised to other subject types needs to be investigated.

Of all threats listed, the only one that could affect the validity of the results of this study in an industrial context is the one related to generalisation to other subject types.

4 Original Study: Results

Of the 32 students participating in the experiment, nine did not complete the questionnaire[11] and were removed from the analysis. Table 9 shows the balance of the experiment before and after participants submitted the questionnaire. We can see that G6 is the most affected group, with 4 missing people.

Appendix B shows the analysis of the experiment. The results show that program and technique are statistically significant (and therefore are influencing effectiveness), while group and the technique by program interaction are not significant.

As regards the techniques, EP shows a higher effectiveness, followed by BT and then by CR. These results are interesting, as all techniques are able to detect all defects. Additionally, more defects are found in ntree compared to cmdline and nametbl, where the same amount of defects are found. Note that ntree is the program applied the first day,

[11] Meaning they were not giving consent to participate in the study.

Table 10 Participants' perceptions of technique effectiveness in the original study

N	CR	BT	EP	Result
23	17.39%	43.48%	39.13%	CR=BT=EP

has the highest Halstead metrics, and it is not the smallest program or the one with lowest complexity.

These results suggest that:

- There is no maturation effect. The program where highest effectiveness is obtained is the one used the first day.
- There is no interaction with selections effect. Group is not significant.
- Mortality does not affect experimental results. The analysis technique used (Linear Mixed-Effects Models) is robust to lack of balance.
- Order of training could be affecting results. The highest effectiveness is obtained in the last technique taught, while the lowest effectiveness is obtained in the first technique taught. This suggests that techniques taught last are more effective than techniques taught first. This could be due to participants remembering better last techniques.
- Results cannot be generalised to other subject types.

4.1 RQ1.1: Participants' Perceptions

Table 10 shows the percentage of participants that perceive each technique to be the most effective. We cannot reject the null hypothesis that the frequency distribution of the responses to the questionnaire item (*Using which technique did you detect most defects?*) follows a uniform distribution[12] (χ^2(2,N=23)=2.696, p=0.260). This means that the number of participants perceiving a particular technique as being more effective cannot be considered different for all three techniques. **Our data do not support the conclusion that techniques are differently frequently perceived as being the most effective**.

4.2 RQ1.2: Comparing Perceptions with Reality

Table 11 shows the value of kappa along with its 95% confidence interval (CI), overall and for each technique separately. We find that all values for kappa with respect to the questionnaire item (*Using which technique did you detect most defects?*) are consistent with lack of agreement (κ <0.4, poor). Although the upper bound of the 95% CIs show agreement, 0 belongs to all 95% CI, meaning that agreement by chance cannot be ruled out. Therefore, **our data do not support the conclusion that participants correctly perceive the most effective technique for them**.

It is worth noting that agreement is higher for the code review technique (the upper bound of the 95% CI in this case shows excellent agreement). This could be attributed to participants being able to remember the actual number of defects identified in code reading whereas for testing techniques they only wrote the test cases. On the other hand, participants do not know the number of defects injected in each program.

As lack of agreement cannot be ruled out, we examine whether the perceptions are biased. The results of the Stuart-Maxwell test show that the null hypothesis of existence

[12] In a uniform distribution, 33.3% of participants should choose each technique.

Table 11 Agreement between perceived and real technique effectiveness in the original study (N=23)

	Kappa value	95% Confidence interval	
		Lower bound	Upper bound
Overall	0.245	–0.072	0.557
CR	0.395	–0.131	0.832
BT	0.175	–0.232	0.585
EP	0.225	–0.146	0.566

of marginal homogeneity cannot be rejected (χ^2(2,N=23)=1.125, p=0.570). This means that we cannot conclude that perceptions and reality are differently distributed. Taking into account the results reported in Section 4.1, this would suggest that, in reality, techniques

Table 12 Agreement between percentage of defects found with each technique in the original study

Sample	N		Krippendorff's Alpha value	95% Confidence interval Lower bound	Upper bound
All	23	CR-BT	–0.2837	–0.7285	0.1711
		CR-EP	–0.4352	–1.0000	0.2524
		EP-BT	–0.1078	–0.9040	0.5512
		CR-BT-EP	–0.2203	–0.6120	0.0978
G1	5	CR-BT	–0.1313	–0.7848	0.4218
		CR-EP	–0.4062	–1.0000	0.5784
		EP-BT	–0.1730	–0.9193	0.5734
		CR-BT-EP	–0.1170	–0.7287	0.3989
G2	4	CR-BT	–0.1666	–1.0000	0.5461
		CR-EP	–0.0722	–0.4052	0.2483
		EP-BT	0.4862	–0.0031	0.9755
		CR-BT-EP	–0.0151	–0.4875	0.4199
G3	5	CR-BT	–0.2956	–0.7273	0.1017
		CR-EP	–0.4540	–1.0000	0.6506
		EP-BT	–0.1368	–1.0000	0.7738
		CR-BT-EP	–0.2289	–0.8099	0.2888
G4	3	CR-BT	–0.1600	–1.0000	1.0000
		CR-EP	–0.1600	–1.0000	1.0000
		EP-BT	Error	Error	Error
		CR-BT-EP	–0.0943	–1.0000	0.8490
G5	4	CR-BT	–0.4789	–1.0000	0.6056
		CR-EP	–0.6448	–1.0000	–0.3768
		EP-BT	–0.4260	–1.0000	0.7408
		CR-BT-EP	–0.3095	–0.8496	0.2306
G6	2	CR-BT	–0.1029	–1.0000	0.9559
		CR-EP	–0.2931	–1.0000	0.9483
		EP-BT	–0.1029	–1.0000	0.9559
		CR-BT-EP	–0.1437	–1.0000	0.9447

Table 13 Agreement between percentage of defects found with each program in the original study (N=23)

	Krippendorff's Alpha value	95% Confidence interval Lower bound	Upper bound
cmdline-nametbl	–0.3301	–1.0000	0.3137
cmdline-ntree	–0.1801	–0.7331	0.2787
nametbl-ntree	–0.1808	–0.9570	0.4400
cmdline-nametbl-ntree	–0.2203	–0.5933	0.1300

cannot be considered the most effective a different number of times.[13] Additionally, the results of the McNemar-Bowker test show that the null hypothesis of existence of symmetry cannot be rejected (χ^2(3,N=23)=1.286, p=0.733). This means that we cannot conclude that there is directionality when participants' perceptions are wrong. These two results suggest that participants are not differently mistaken about one technique as they are about the others. **Techniques are not differently subject to misperceptions**.

4.3 RQ1.3: Comparing the Effectiveness of Techniques

We are going to check if misperceptions could be due to participants detecting the same amount of defects with all three techniques, and therefore being impossible for them to make the right decision. Table 12 shows the value and 95% CI of Krippendorff's α, overall and for each pair of techniques, for all participants and for every design group (participants that applied the same technique on the same program) separately, and Table 13 shows the value and 95% CI of Krippendorff's α, overall and for each program/session. For values with all participants, we can rule out agreement, as the upper bound of the 95% CIs are consistent with lack of agreement (α <0.4), except for the case of EP-BT and nametbl-ntree for which the upper bound of the 95% CIs are consistent with fair to good agreement. However, even in this two cases, 0 belongs to the 95% CIs, meaning that agreement by chance cannot be ruled out. This means that **participants do not obtain similar effectiveness values when applying the different techniques (testing the different programs) so as to be difficult to discriminate among techniques/programs**.

Furthermore, kappa values are negative, which indicates disagreement. This is good for the study, as it means that **participants should be able to discriminate among techniques, and lack of agreement cannot be attributed to a problem of being impossible to discriminate among techniques**.

As regards the results for groups, although α values are negative,[14] the 95% CIs are too wide to show reliable results (due to small sample size). Note that in most of the cases they range from existence of disagreement in the lower bound (α <-0.4) to the existence of agreement in the upper bound (α >0.4).

[13]Note that the fact that all three techniques are classed as the most effective the same number of times is not incompatible with there being techniques that are more effective than others.

[14]Except in the case of Group 2 where there is agreement for the EP-BT techniques. Since this is the only agreement found we think it could be spurious.

Table 14 Observed reduction in technique effectiveness for mismatch

Technique	No. mismatches	Mismatch cost	Cost		
			Mean	Median	Std. deviation
CR	2(4)	26pp; 43pp	35pp	35pp	12
BT	6(10)	2pp; 14pp; 17pp; 29pp; 57pp; 86pp	34pp	23pp	31
EP	3(9)	14pp; 14pp; 33pp	21pp	14pp	11
TOTAL	11(23)		31pp	26pp	24

Column 2 shows the number of mismatches out of the total number of participants who perceived the technique as being most effective. Column 3 shows the cost for each mismatch. Columns 4-6 shows the mean and median (in percentage points), and standard deviation for mismatch cost

4.4 RQ1.4: Cost of Mismatch

Table 14 and Fig. 1 show the cost of mismatch. We can see that the EP technique has fewer mismatches compared to the other two. Additionally, the mean and median mismatch cost is smaller. On the other hand, the BT technique has more mismatches, and a higher dispersion. The results of the Kruskal-Wallis test reveal that we cannot reject the null hypothesis of techniques having the same mismatch cost (H(2)=0.685, p=0.710). This means that we cannot claim a difference in mismatch cost between the techniques. The estimated mean mismatch cost is 31pp (median 26pp).

These results suggest that **the mismatch cost is not negligible** (31pp), **and is not related to the technique perceived as most effective.** However, note that the existence of very high mismatches and few datapoints could be affecting these results.

4.5 RQ1.5: Expected Loss of Effectiveness

Table 15 shows the average loss of effectiveness that should be expected in a project, where typically different testers participate, and therefore, there would be both matches and mis-

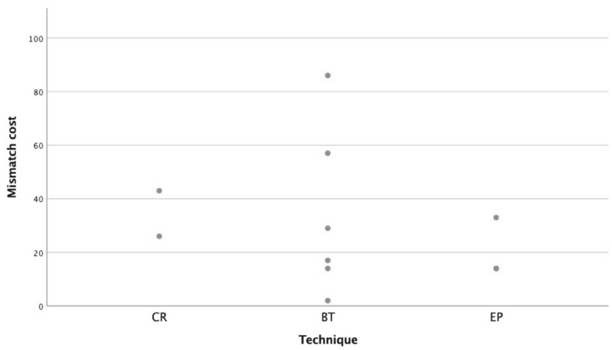

Fig. 1 Scatterplot for observed mismatch cost in the original study. Datapoints correspond to the mismatch cost in Table 14

Table 15 Observed reduction in technique effectiveness when considering matches and mismatches

			Cost		
Technique	N	(Mis)match cost	Mean	Median	Std. deviation
CR	4	0pp; 0pp; 26pp; 43pp	17pp	13pp	21
BT	10	0pp; 0pp; 0pp; 0pp; 2pp; 14pp; 17pp; 29pp; 57pp; 86pp	21pp	8pp	29
EP	9	0pp; 0pp; 0pp; 0pp; 0pp; 0pp; 14pp; 14pp; 33pp	7pp	0pp	12
TOTAL	23		15pp	0pp	22

Column 2 shows the number of datapoints. Column 3 shows the cost for each (mis)match. Columns 4-6 show the mean and median (in percentage points), and std. deviation for the reduction in technique effectiveness

matches.[15] Again, the results of the Kruskal-Wallis test reveal that we cannot reject the null hypothesis of techniques having the same expected reduction in technique effectiveness for a project (H(2)=1.510, p=0.470). This means we cannot claim a difference in project effectiveness loss between techniques. The mean expected loss in effectiveness in the project is estimated as 15pp.[16]

These results suggest that **the expected loss in effectiveness in a project is not negligible** (15pp), **and is not related to the technique perceived as most effective.** However, we must note again that the existence of very high mismatches for BT and few datapoints could be affecting these results.

4.6 Findings of the Original Study

Our **findings** are:

- Participants should not base their decisions on their own perceptions, as their perceptions are not reliable and have an associated cost.
- We have not been able to find a bias towards one or more particular techniques that might explain the misperceptions.
- Participants should have been able to identify the different effectiveness of techniques.
- Misperceptions cannot be put down to experience. The possible drivers of these misperceptions require further research.

Note that these findings cannot be generalised to other types of developers rather than those with the same profile as the ones used in this study.

5 Replicated Study: Research Questions and Methodology

We decide to further investigate the results of the original study in search of possible drivers behind misperceptions. Psychology considers that people's perceptions can be affected by personal characteristics as attitudes, personal interests and expectations. Therefore, we

[15] Note that the mismatch cost is 0 when there is a match.

[16] Note that the median here is not very informative. In this particular case it is 0pp. This happens when there are more matches than mismatches.

decide to examine participants' opinions by conducting a differentiated replication of the original study (Shull et al. 2008) that extends its goal as follows:

1. The survey of effectiveness perception is extended to include questions on programs.
2. We want to find out whether participants' perceptions might be conditioned by their opinions. More precisely: their preferences (favourite technique), their performance (the technique that they think they applied best) and technique or program complexity (the technique that they think is easiest to apply, or the simplest program to be tested).

Therefore, the replicated study reexamines RQ1 stated in the original study (this time the survey taken by participants also includes questions regarding programs), and addresses the new following research questions:

- *RQ1.6: Are participants perceptions related to the numb1er of defects reported by participants?*
  We want to assess if participants perceive as the most effective technique the one with which they have reported more defects.
- *RQ2: Can participants' opinions be used as predictors for testing effectiveness?*
  - *RQ2.1: What are participants' opinions about techniques and programs?*
    We want to know if participants have different opinions about techniques or programs.
  - *RQ2.2: Do participants' opinions predict their effectiveness?*
    We want to assess if the opinions that participants have about techniques (or programs) predict which one is the most effective for them.
- *RQ3: Is there a relationship between participants' perceptions and opinions?*
  - *RQ3.1: Is there a relationship between participants' perceptions and opinions?*
    We want to assess if the opinions that participants have about techniques (or programs) are related to their perceptions.
  - *RQ3.2: Is there a relationship between participants' opinions?*
    We want to assess if a certain opinion that participants have about techniques are related to other opinions.

To answer these questions, we replicate the original study with students of the same course in the following academic year. This time we have 46 students. The changes made to the replication of the experiment are as follows:

- The questionnaire to be completed by participants at the end of the experiment is extended to include new questions. The information we want to capture with the opinion questions is:
  - *Participants performance on techniques*. With this question we are referring to process conformance. Best applied technique is the technique each participant thinks (s)he applied more thoroughly. It corresponds to OT1: Which technique did you apply best?
  - *Participants preferences*. We want to know the favourite technique of each participant. They one (s)he felt more comfortable with when applied. It corresponds to OT2: Which technique do you like best?
  - *Technique complexity*. We want to know the technique each participant thinks was easiest to get process conformance. It corresponds to OT3: Which technique is the easiest to apply?
  - *Program testability*. We want to know the program it was easier to test. This is, the program in which process conformance could be obtained more easily. It corresponds to OP1: Which is the simplest program?

Table 16 Questions of replicated study questionnaire

ID	Question	Aspect
PT1	Using which technique did you detect most defects?	Perceptions
PP1	In which program did you detect most defects?	
OT1	Which technique did you apply best?	
OT2	Which technique do you like best?	Opinions
OT3	Which technique is the easiest to apply?	
OP1	Which is the simplest program?	

Table 16 summarizes the survey questions. We have chosen these questions because we need to ask simple questions, that can be easily understood by participants, being at the same time meaningful. We do not want to overwhelm participants with complex questions that have lots of explanations. A complex questionnaire might discourage students to submit it.

- The program faults are changed. The original study is designed so that all techniques are effective at finding all defects injected. We choose faults detectable by all techniques so the techniques could be compared fairly. The replicated study is designed to cover the situation in which some faults cannot be detected by all techniques. Therefore, we inject some faults that techniques are not effective at detecting.

  For example, BT cannot detect a non-implemented feature (as participants are required to generate test cases from the source code only). Likewise, EP cannot find a fault whose detection depends on the combination of two invalid equivalence classes. Therefore, in the replicated study, we inject some faults that can be detected by BT but not by EP and some faults that can be detected by EP but not by BT into each program (each program is seeded with six faults). Note that the design is balanced: we inject the same number of faults that BT can detect, but not EP, that the opposite –EP can detect, but not BT). This change is expected to affect the effectiveness of EP and BT, which might be lower than in the original study. It should not affect the effectiveness of CR.
- We change the program application order to further study maturation issues. The order is now: cmdline, ntree, nametbl. This change should not affect the results.
- Participants run their own test cases. It could be that the misperceptions obtained in the original study are due to the fact that participants are not running their own test cases.
- There are not two versions anymore but one. Faults and failures are not the goal of this study. This helps to simplify the experiment.

Table 17 shows a summary of the changes made to the study.

Table 17 Changes made to the original study

Change	Purpose
New questions	Extend scope of study
Program faults	Cover all possible scenarios
Program application order	Further study maturation issues
Participants run their own test cases	Improve perceptions
Just one program version	Simplify experiment

To measure technique effectiveness we proceed in the same way as in the original study. We do not rely on the reported failures, as participants could:

1. Report false positives (non-real failures).
2. Report the same failure more than once (although they were asked not to do so).
3. Miss failures corresponding to faults that have been exercised by the technique, but for some reason have not been seen.

We measure the new response variable (reported defects) by counting the number of faults/failures reported by each participant.

We analyse RQ2.1 in the same manner as RQ1.1, and RQ1.6, RQ2.2, RQ3.1 and RQ3.2 like RQ1.2. Table 18 summarises the statistical tests used to answer each research question.

6 Replicated Study: Validity Threats

The threats to validity listed in the original study apply to this replicated study. Additionally, we have identified the following ones:

6.1 Conclusion Validity

1. *Reliability of treatment implementation*. The replicated experiment is run by the same researchers that performed the original experiment. This assures that the two groups of participants do not implement the treatments differently.

6.2 Internal Validity

1. *Evaluation Apprehension*. The use of students and associating their performance in the experiment with their grade in the course might explain that participants consider that their performance and not the weaknesses of the techniques explain the effectiveness of a technique.

6.3 Construct Validity

1. *Inadequate preoperational explanation of effect constructs*. Since opinions are hard constructs to operationalize, there exists the possibility that the questions appearing in the questionnaire are not interpreted by participants the way we intended to.

Table 18 Statistical tests used to answer new research questions of the replicated study

RQ	Statistical test	Test description	Checks
2.1	χ^2 goodness-of-fit	How well a theoretical distribution fits a empirical distribution	All 3 techniques/programs are equally frequently perceived/opined as most effective
1.6	Cohen's kappa	Agreement for categorical	Match between perceptions or
2.2		variables when 2 raters classify	opinions and reality and
3.1		different objects	among them
3.2	Stuart-Maxwell	Changes in the proportion among raters' agreement (marginal homogeneity)	Bias in perceptions/opinions
	McNemar-Bowker	Symmetry of the associated contingency table	Bias in perceptions/opinions

Table 19 Balance before and after dropouts in the replicated study

Group	G1	G2	G3	G4	G5	G6
Initial	8	8	8	8	7	7
Final	6	7	7	8	5	6

6.4 External Validity

1. *Reproducibility of results*. It is not clear to what extent the results obtained here are reproducible. Therefore, more replications of the study are needed. The steps that should be followed are:
   (a) Replicate the study capturing the reasons for the answers given by participants.
   (b) Perform the study with practitioners with the same characteristics as the students used in this study (people with little or no experience in software testing).
   (c) Explore and define what types of experience could be influencing the results (academic, professional, programming, testing, etc.).
   (d) Run new studies taking into consideration increasing levels of experience.

Again, of all threats affecting the replicated study, the only one that could affect the validity of the results of this study in an industrial context is the one related to generalisation to other subject types.

7 Replicated Study: Results

Of the 46 students participating in the experiment, seven did not complete the questionnaire[17] and were removed from the analysis. Table 19 shows the changes in the experimental groups due to students not participating in the study. Balance is not seriously affected by mortality—although it would have been desirable that Group 5 had at least one more participant.

Additionally, another four participants did not answer all the questions and were removed from the analysis of the respective questions.

7.1 RQ1: Participants' Perceptions as Predictors

7.1.1 RQ1.1-RQ1.5: Comparison with Original Study Results

Appendix C shows the analysis of the experiment. Program is the only statistically significant variable (group, program and the program by technique interaction are not significant). In this replication, fewer defects are found in cmdline compared to nametbl and ntree, where the same amount of defects are found. Some results are in line with those obtained in the original study:

– There is no interaction with selections effect. Group is not significant.
– Mortality does not affect experimental results. The analysis technique used (Linear Mixed-Effects Models) is robust to lack of balance.

[17] Meaning they were not giving consent to participate in the study.

Table 20 Participants' perceptions for technique effectiveness in the replicated study

Question	N	CR	BT	EP	Result
PT1	37	37.84%	18.92%	43.24%	CR=BT=EP

– Results cannot be generalized to other subject types.

But others contradict those obtained in the original study, and therefore need further investigation:

– Maturation effect cannot be ruled out. The program where lowest effectiveness is obtained is the one used the first day.
– Order of training does not seem to be affecting results. All techniques show the same effectiveness.

Table 20 shows the results of participants' perceptions for techniques. The results are the same as in the original study (χ^2(2,N=37)=3.622, p=0.164). **Our data do not support the conclusion that techniques are differently frequently perceived as being the most effective.**.

Our data do not support the conclusion that participants correctly perceive the most effective technique for them. The overall and per technique kappa values and 95% CI reported in Table 21 are in line with those in the original study. This suggests that the hypothesis we elaborated in the original experiment would not be correct. For some reason, perceptions are more accurate with the CR technique.

Again as in the original study, **we have not been able to observe bias in perceptions** (Stuart-Maxwell outputs (χ^2(2, N=37)=3.103,p=0.212), and McNemar-Bowker outputs (χ^2(3,N=37)=3.143, p=0.370)).

Table 22 shows the value of Krippendorff's α and 95% CI, overall and for each pair of techniques, for all participants and for every design group (participants that applied the same technique on the same program) separately, and Table 23 shows the value of Krippendorff's α and 95% CI, overall and for each program/session. Again, the results obtained are the same as in the original study. **Participants do not obtain similar effectiveness values when applying the different techniques (testing the different programs) so as to be difficult to discriminate among techniques/programs**.

Table 24 and Fig. 2 show the cost of mismatch. As in the original study, the **mismatch cost is not related to the technique perceived as being the most effective,** (Kruskal-Wallis

Table 21 Agreement between technique effectiveness perceptions and reality in replicated study (PT1, N=37)

		95% Confidence interval	
	Kappa value	Lower bound	Upper bound
Overall	0.193	–0.023	0.430
CR	0.364	0.078	0.628
BT	–0.025	–0.253	0.311
EP	0.142	–0.176	0.446

Table 22 Agreement between techniques for percentage of defects found in the replicated study

Sample	N		Krippendorff's Alpha value	95% Confidence interval Lower bound	Upper bound
All	39	CR-BT	0.0462	–0.2975	0.3384
		CR-EP	–0.0217	–0.4304	0.3444
		EP-BT	–0.0074	–0.4577	0.3363
		CR-BT-EP	0.0083	–0.1988	0.2029
G1	6	CR-BT	0.0782	–0.9665	0.8771
		CR-EP	–0.0447	–1.0000	0.6313
		EP-BT	0.0833	–0.7188	0.6563
		CR-BT-EP	0.0368	–0.5652	0.5184
G2	7	CR-BT	0.0000	–0.5000	0.5000
		CR-EP	0.2593	–0.1493	0.6680
		EP-BT	0.2642	–0.5698	0.9019
		CR-BT-EP	0.1499	–0.1969	0.4855
G3	7	CR-BT	0.2500	–0.1875	0.6875
		CR-EP	0.1727	–0.4182	0.7045
		EP-BT	0.1096	–0.6918	0.6884
		CR-BT-EP	0.1958	–0.1713	0.5105
G4	8	CR-BT	–0.2069	–0.8966	0.3103
		CR-EP	–0.1290	–1.0000	0.6237
		EP-BT	0.1916	–0.4371	0.6407
		CR-BT-EP	–0.0758	–0.5395	0.3137
G5	5	CR-BT	–0.3125	–1.0000	0.8125
		CR-EP	–0.5672	–1.0000	0.5522
		EP-BT	–0.6211	–1.0000	0.2733
		CR-BT-EP	–0.3515	–0.9926	0.2723
G6	6	CR-BT	–0.3750	–1.0000	0.7250
		CR-EP	–0.2222	–0.8333	0.3889
		EP-BT	0.0833	–0.3750	0.5417
		CR-BT-EP	–0.1168	–0.7993	0.4416

(H(2)=2.979,p=0.226)). Also, there are about the same proportion of mismatches as in the original study (48% of mismatches in the original study versus 51% in the replicated study.

Table 23 Agreement between programs for percentage of defects found in the replicatd study (N=39)

	Krippendorff's Alpha value	95% Confidence interval Lower bound	Upper bound
cmdline-nametbl	–0.0240	–0.4017	0.3040
cmdline-ntree	–0.0398	–0.4049	0.3095
nametbl-ntree	0.0304	–0.3716	0.3851
cmdline-nametbl-ntree	0.0083	–0.2082	0.2092

Table 24 Observed reduction in technique effectiveness for mismatch

Technique	No. mismatches	Mismatch cost	Cost		
			Mean	Median	Std. deviation
CR	3(14)	17pp; 17pp; 17pp	17pp	17pp	0
BT	6(7)	17pp; 50pp; 17pp; 17pp; 17pp; 33pp	25pp	17pp	14
EP	10(16)	17pp; 50pp; 17pp; 17pp; 17pp; 33pp; 17pp; 50pp; 33pp; 17pp	27pp	17pp	14
TOTAL	19(37)		25pp	17pp	13

Column 2 shows the number of mismatches out of the total number of participants who perceived the technique as being most effective. Column 3 shows the cost for each mismatch. Columns 4-6 shows the mean and median (in percentage points), and standard deviation for mismatch cost

However, there are some differences with respect to the original study:

– While CR had the greatest number of mismatches in the original study, now it has the smallest. The number of mismatches for BT and EP has increased with respect to the original study.
– In the replicated study, the mismatch cost is slightly lower (25pp compared with 31pp in the original study). The mismatch cost is smaller when CR is involved.

This could be due to the change in the seeded faults or just to natural variation. It should be further checked. However, it is a fact that the effectiveness of EP and BT has decreased in the replicated study, while CR has a similar effectiveness as in the original study. This suggests that **the mismatch cost could be related to the faults that the program contains**. However, this issue needs to be further investigated, as we have few data points. Note that, as in the original study, the existence of few datapoints could be affecting these results.

Table 25 shows the average loss of effectiveness that should be expected in a project due to mismatch. The expected loss in effectiveness in a project is similar to the one observed in the original study (13pp), but this time it is related to the technique perceived as most

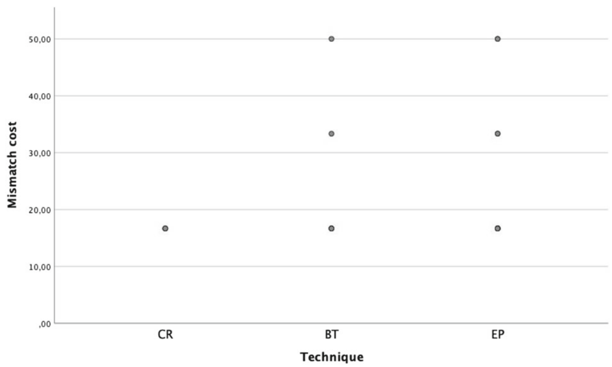

Fig. 2 Scatterplot for observed mismatch cost in the replicated study

Table 25 Observed reduction in technique effectiveness when considering matches and mismatches

Technique	N	(Mis)match Cost	Cost		
			Mean	Median	Std. deviation
CR	14	0pp; 0pp; 0pp; 0pp; 0pp; 0pp; 0pp; 0pp; 0pp; 0pp; 0pp; 17pp; 17pp; 17pp	4pp	0pp	7
BT	7	0pp; 17pp; 50pp; 17pp; 17pp; 17pp; 33pp	21pp	17pp	16
EP	16	0pp; 0pp; 0pp; 0pp; 0pp; 0pp; 17pp; 17pp; 17pp; 17pp; 17pp; 17pp; 33pp; 33pp; 50pp; 50pp	17pp	17pp	17
TOTAL	37		13pp	17pp	15

Column 2 shows the number of datapoints. Column 3 shows the cost for each (mis)match. Columns 4-6 show the mean and median (in percentage points), and std. deviation for reduction in effectiveness in the project

effective (Kruskal-Wallis (H(2)=9.691, p=0.008)). This means that some mismatches are more costly than others. The misperception of CR as being the most effective technique has a lower associated cost (4pp) than for BT or EP (18pp). This suggests that participants' who think CR is the most effective might be allowed to apply this technique, as, even if they are wrong, the loss of effectiveness would be negligible. However, **participants should not rely on their perceptions** even in this case, since **fault type could have an impact** on this result and they will never know what faults there are in the program beforehand. Note that again the existence of few datapoints could be affecting these results. Therefore, this issue needs to be further researched.

The **findings of the replicated study** are:

– They confirm the results of the original study.
– A possible relationship between fault type and mismatch cost should be further investigated.

Since the results of both studies are similar, we have pooled the data and performed joint analyses for all research questions to overcome the problem of lack of power due to sample size. They are reported in Appendix D. The results confirm those obtained by each study individually. This allows us to gain confidence in the results obtained.

7.1.2 RQ1.6: Perceptions and Number of Defects Reported

One of the conclusions of the original study was that perceived technique effectiveness could match the technique with the highest number of defects reported. Table 26 shows the value of kappa and its 95% CI, overall and for each technique separately. We find that all values for kappa with respect to the *perceived most effective technique and technique with greater number of defects reported* are consistent with lack of agreement (κ <0.4, poor). However, the upper bound of all 95% CIs show agreement, and the lower bound of all 95% CIs but BT are greater than zero. This means that **although our data do not support the conclusion that participants correctly perceive the most effective technique for them, it should not be ruled out**.

Table 26 Agreement between perceived most effective technique and technique with greater number of defects reported in the replicated study (N=37)

	Kappa value	95% Confidence interval	
		Lower bound	Upper bound
Overall	0.289	0.059	0.518
CR	0.347	0.051	0.624
BT	0.081	–0.229	0.416
EP	0.371	0.065	0.645

This means that **participants perceptions about technique effectiveness could be related to reporting a greater number of defects with that technique**.

As lack of agreement cannot be ruled out, we examine whether the perceptions are biased. The results of the Stuart-Maxwell test show that the null hypothesis of existence of marginal homogeneity cannot be rejected (χ^2(2,N=37)=2.458, p=0.293). This means that we cannot conclude that perceptions and reported defects are differently distributed. Additionally, the results of the McNemar-Bowker test show that the null hypothesis of existence of symmetry cannot be rejected (χ^2(3,N=37)=2.867, p=0.413). This means that we cannot conclude that there is directionality when participants' perceptions do not match the technique with highest defects reported.

The lack of a clear agreement could be due to the fact that participants do not remember exactly the number of defects found with each technique.

7.1.3 RQ1.1-RQ1.2: Program Perceptions

Table 27 shows the results of participants' perceptions for the *program in which the participants detected most defects*. We found that the same phenomenon applies to programs as to techniques. All three programs cannot be considered differently frequently perceived as being the ones where most defects were found, as we cannot reject the null hypothesis that the frequency distribution of the responses follows a uniform distribution (χ^2(2,N=37)=2.649, p=0.266). **Our data do not support the conclusion that programs are differently frequently perceived as having a higher percentage of defects found than the others**. This contrasts with the fact that cmdline has a slightly higher complexity and number of LOC, and that ntree shows highest Halstead metrics. We expected cmdline and/or ntree should be perceived less frequently as having a higher detection rate.

However, the values for kappa in Table 28 show that there seems to be agreement overall and for cmdline and ntree (κ >0.4, fair to good and agreement by chance can be ruled out, since 0 does not belong to the 95% CI), but not so for the nametbl program (κ=0.292, poor and agreement by chance cannot be ruled out, as 0 belongs to the 95% CI). This means that **participants do tend to correctly perceive the program in which they detected most defects**. This is striking, as it contrasts with the disagreement for techniques. Pending the

Table 27 Participants' perceptions of program effectiveness in the replicated study (PP1)

Question	N	cmdline	nametbl	ntree	Result
PP1	37	43.24%	35.14%	21.62%	Cm=Na=Nt

Table 28 Agreement between program effectiveness perceptions and reality in the replicated study (PP1, N=37)

	Kappa value	95% Confidence interval	
		Lower bound	Upper bound
Overall	0.401	0.186	0.621
cmdline	0.469	0.208	0.711
nametbl	0.292	-0.011	0.617
ntree	0.460	0.137	0.769

analysis of the mismatch cost, it suggests that participants' perceptions on the percentage of defects found may be reliable. This is interesting, as cmdline has a higher complexity.

Since there is agreement, we are not going to study mismatch cost.

Misperceptions do not seem to affect participants' perception of how well they have tested a program.

7.2 RQ2: Participants' Opinions as Predictors

7.2.1 RQ2.1: Participants' Opinions

Table 29 shows the results for participants' opinions with respect to techniques.

With regard to the *technique participants applied best* (OT1), we can reject the null hypothesis that the frequency with which they perceive that they had applied each of the three techniques best is the same (χ^2(2,N=38)=10.947, p=0.004). **More people think they applied EP best, followed by both BT and CR** (which merit the same opinion).

In the case of the *technique participants liked best* (OT2), the results are similar. We can reject the null hypothesis that participants equally as often regard all three techniques as being their favourite technique (χ^2(2,N=38)=22.474, p=0.000). **Most people like EP best, followed by both BT and CR** (which merit the same opinion).

Finally, as regards the *technique that participants found easiest to apply* (OT3), the results are exactly the same as for the preferred technique (χ^2(2,N=38)=22.474, p=0.000). **Most people regard EP as being the technique that is easiest to apply, followed by both BT and CR** (which merit the same opinion).

Table 30 shows the results for participants' opinions for programs. We cannot reject the null hypothesis of all programs equally frequently being viewed as *the simplest*. (χ^2(2,N=38)=1.474, p=0.479). Therefore, our data do not support the conclusion that **all three programs are differently frequently perceived as being the simplest**. This result suggests that both the differences in complexity and size of cmdline and the highest Halstead metrics of ntree

Table 29 Participants' opinions for techniques in the replicated study

Question	N	CR	BT	EP	Result
OT1	38	26.32%	15.79%	57.89%	EP>(BT=CR)
OT2	38	23.68%	7.89%	68.42%	EP>(BT=CR)
OT3	38	23.68%	7.89%	68.42%	EP>(BT=CR)

Table 30 Participants' opinions for programs in the replicated study

Question	N	cmdline	nametbl	ntree	Result
OP1	38	26.32%	42.11%	31.58%	Cm=Na=Nt

are small. This result suggest that participants could be interpreting differently this question. Another possibility could be that the question that has been used to operationalize the corresponding construct is vague, and participants are not interpreting it correctly.

7.2.2 RQ2.2: Comparing Opinions with Reality

The technique that participants think they *applied best* (OT1) **is not a good predictor of technique effectiveness**. The overall and per technique kappa values in the fourth column of Table 31 are consistent with lack of agreement (in all cases (κ <0.4, poor), and although the upper bound of the 95% CIs show agreement, 0 belongs to most 95% CIs, meaning that agreement by chance cannot be ruled out). However, we find that there is a bias, as the Stuart-Maxwell and McNemar-Bowker tests can reject the null hypotheses of marginal homogeneity (χ^2(2,N=38)=10.815, p=0.004) and symmetry (χ^2(3, N=38)=12.067, p=0.007), respectively. Looking at the light and dark grey cells in the corresponding contingency table represented in Table 32, we find that the cells placed under the diagonal have higher values than those positioned above the diagonal. In other words, there are rather more participants that consider that they applied EP best, despite achieving better effectiveness with CR and BT (9 and 5), than participants who consider that they applied CR or BT best, despite being more effective using EP (1 in both cases). This suggests that **there is a bias towards EP**. This bias is much more pronounced with respect to CR. These results are consistent with the ones found in the previous section. There are several possible interpretations for these results: 1) we do not know if the opinion on the best applied technique is accurate (meaning that it is really the best applied technique); 2) possibly due to

Table 31 Agreement between opinions and reality for techniques in the replicated study

				95% Confidence interval	
Question	N	Technique	Kappa value	Lower bound	Upper bound
OT1	38	Overall	0.257	0.060	0.461
		CR	0.348	0.091	0.598
		BT	0.166	–0.181	0.539
		EP	0.216	–0.022	0.474
OT2	38	Overall	0.166	–0.023	0.360
		CR	0.232	–0.009	0.486
		BT	0.101	–0.145	0.457
		EP	0.134	–0.072	0.365
OT3	38	Overall	0.166	–0.016	0.367
		CR	0.232	–0.019	0.491
		BT	0.101	–0.145	0.469
		EP	0.134	–0.077	0.351

Table 32 Contingency table for best applied technique (OT1) in the replicated study

Best applied	Reality			
technique	CR	BT	EP	Total
CR	9	0	1	10
BT	3	2	1	6
EP	9	5	8	22
Total	21	7	10	38

the change in faults, technique performance is worse in this replication than in the original study; and 3) it could be that participants have misunderstood the question. Interviewing participants or asking them in the questionnaire about the reasons for their answers, would have helped to clarify this last issue.

As regards *participants' favourite technique* (OT2), **the results are similar**. This opinion does not predict technique effectiveness, since all kappa values in the fourth column of Table 31 denote lack of agreement (in all cases (κ <0.4, poor), and although the upper bound of the 95% CIs show agreement, 0 belongs to all 95% CIs, meaning that agreement by chance cannot be ruled out). Again, we find there is bias, as the Stuart-Maxwell and the McNemar-Bowker tests can reject the null hypotheses of marginal homogeneity (χ^2(2,N=38)=11.931, p=0.003) and symmetry (χ^2(3,N=38)=11.974,p=0.007), respectively. Looking at the light and dark grey cells in Table 33, we again find **that there is bias towards EP**. There are rather more participants that think that they applied EP better, despite being more effective using CR and BT (12 and 5), than participants that considered that they applied CR or BT better, despite being more effective using EP (1 in both cases). Note that the bias between CR and EP is more pronounced. Note that it is very unlikely that participants have not properly interpreted this question. It just seems that the technique they most like is not typically the most effective.

Finally, with respect to the *technique that is easiest to apply* (OT3), **we find that the results are exactly the same as for their preferred technique**. However, as we have seen in OT2, their preferred technique is not a good predictor of effectiveness (see third row of Table 31), and there is bias towards EP (see light and dark grey cells in Table 33). These results are in line with a common claim in SE, namely, that **developers should not base the decisions that they make on their opinions, as they are biased**. Again, it should be noted that participants might not be interpreting the question as we expected. Further research is necessary.

Table 33 Contingency table for favourite (OT2) and easiest to apply (OT3) technique in the replicated study

Preferred /	Reality			
Easiest to apply	CR	BT	EP	Total
CR	7	1	1	9
BT	1	1	1	3
EP	12	5	9	26
Total	20	7	11	38

Table 34 Agreement between opinions and reality for programs in the replicated study (OP1, N=38)

	Kappa value	95% Confidence interval	
		Lower bound	Upper bound
Overall	0.225	–0.014	0.449
cmdline	0.066	–0.213	0.371
nametbl	0.197	–0.124	0.501
ntree	0.372	0.034	0.662

As far as the *simplest program* is concerned, we find, as we did for the techniques, that **it is not a good predictor of the program in which most defects were detected** (the values for overall and per program kappa in Table 34 denote lack of agreement (in all cases (κ <0.4, poor), and although the upper bound of the 95% CIs show agreement, 0 belongs to 95% CIs—except ntree, meaning that agreement by chance cannot be ruled out). Unlike the opinions on techniques, **we were not able to find any bias** this time, as neither the null hypothesis of marginal homogeneity (χ^2(2,N=38)=1.621, p=0.445) nor symmetry (χ^2(3,N=38)=3.286, p=0.350) can be rejected. This result suggests that **the programs that participants perceive to be the simplest are not necessarily the ones where most defects have been found**. Again, it should be noted the problem of participants interpreting the simplest program in a different way as we expected.

7.2.3 Findings

Our **findings** suggest:

- Participants' opinions should not drive their decisions.
- Participants prefer EP (think they applied it best, like it best and think that it is easier to apply), and rate CR and BT equally.
- All three programs are equally frequently perceived as being the simplest.
- The programs that the participants perceive as the simplest are not the ones where the highest number of defects have been found.

These results should be understood within the validity limits of the study.

7.3 RQ3: Comparing Perceptions and Opinions

7.3.1 RQ3.1: Comparing Perceptions and Opinions

In this section, we look at whether participants' perceptions of technique effectiveness are biased by their opinions about the techniques.

According to the results for kappa shown in the fourth column of Table 35 (PT1-OT1), we find that **results are compatible with agreement** (overall and per technique, except for BT in which lack of agreement cannot be ruled out) **between the *technique perceived to be the most effective* and the *technique participants think they applied best*** (in all cases (κ >0.4, fair to good), and in all cases but BT, 0 does not belong to 95% CIs, meaning that agreement by chance can be ruled out). This is an interesting finding, as it suggests that **participants think that technique effectiveness is related to how well the technique is applied**. Technique performance definitely decreases if they are not applied properly. It is

Table 35 Agreement between technique perceptions and opinions in the replicated study

Question	N	Technique	Kappa value	95% Confidence interval	
				Lower bound	Upper bound
PT1-OT1	37	Overall	0.561	0.340	0.767
		CR	0.513	0.216	0.801
		BT	0.406	–0.045	0.768
		EP	0.684	0.479	0.891
PT1-OT2	37	Overall	0.325	0.100	0.541
		CR	0.197	–0.121	0.510
		BT	0.323	–0.078	0.684
		EP	0.432	0.180	0.682
PT1-OT3	37	Overall	0.325	0.103	0.543
		CR	0.197	–0.117	0.508
		BT	0.323	–0.088	0.684
		EP	0.432	0.174	0.674

no less true, however, that techniques have intrinsic characteristics that may lead to some defects not being detected. In fact, the controlled experiment includes some faults that some techniques are unable to detect. A possible explanation for this result could be that the evaluation apprehension threat is materializing.

On the other hand, the kappa values in the fourth column of Table 35 (PT1-OT2) **reveal a lack of agreement for CR and BT between the *preferred technique* and the *technique perceived as being most effective*** (in all cases (κ <0.4, poor), and although the upper bound of the 95% CIs show agreement, 0 belongs to 95% CIs, meaning that agreement by chance cannot be ruled out), **whereas overall lack of agreement cannot be ruled out** ((κ <0.4, poor), the upper bound of the 95% CI shows agreement, and 0 does not belong the to 95% CI, meaning that agreement by chance can be ruled out). Finally, **there is agreement** (κ >0.4, fair to good) **in the case of EP**. This means that, **in the case of EP, participants tend to associate their favourite technique with the perceived most effective technique, contrary to the findings for CR and BT**. This is more likely to be due to EP being the technique that many more participants like best (and the chances for there being a match are higher compared to the other techniques) than to there actually being a real match.

With respect to directionality whenever there is disagreement, the results of the Stuart-Maxwell and the McNemar-Bowker tests show that the null hypotheses of marginal homogeneity (χ^2(2,N=37)=8.355,p=0.015) and symmetry (χ^2(3,N=37)=8.444, p=0.038) can be rejected. Looking at the light grey cells in Table 36, we find that there are more participants claiming to have applied CR best that prefer EP than vice versa (8 versus 1). This means that **the mismatch between the technique that participants like best and the technique that they perceive as being most effective can largely be put down to participants who like EP better perceiving CR to be more effective**.

The results for the agreement between the *technique that is easiest to apply* and the *technique that is perceived to be most effective* are exactly the same as for the preferred technique (see third row of Table 35). This means that, for EP, the participants equate the technique that they find easiest to apply with the one that they regard as being most effective. This does not hold for the other two techniques. Likewise, the mismatch between the

Table 36 Contingency table for perceived most effective vs. preferred technique (PT1-OT2) and perceived most effective vs. easiest to apply technique (PT1-OT3)

Perceived	Preferred/easiest to apply			
most effective	CR	BT	EP	Total
CR	5	1	8	14
BT	3	2	2	7
EP	1	0	15	16
Total	9	3	25	37

technique that is easiest to apply and the technique perceived as being most effective can be largely put down to participants who applied EP best perceiving CR to be more effective (see Table 36).

As mentioned earlier, we found that participants have a correct perception of the program in which they detected most defects. Table 37 shows that participants do not associate *simplest program* with *program in which most defects were detected* (PP1-OP1). This is striking as it would be logical for it to be easier to find defects in the simplest program. As illustrated by the fact that the null hypotheses of marginal homogeneity (χ^2(2,N=37)=3.220,p=0.200) and symmetry (χ^2(3,N=37)=4.000, p=0.261) cannot be rejected, we were not able to find bias in any of the cases where there is disagreement. A possible explanation for this result is that participants are not properly interpreting what simple means.

7.3.2 RQ3.2: Comparing Opinions

Finally, we study the possible relation between the opinions themselves. Looking at Table 38, we find that **participants equate *the technique they applied best* with *their favourite technique* and with *the technique they found easiest to apply*** (overall and per technique (κ >0.4, fair to good), and 0 does not belong to 95% CIs, meaning that agreement by chance can be ruled out). It makes sense that the technique that participants found easiest to apply should be the one that they think they applied best and like best. Typically, people like easy things (or maybe we think things are easy because we like them). In this respect, we can conclude that participants' opinions about the techniques all have the same directional effect.

7.3.3 Findings

Our **findings** suggest:

Table 37 Agreement between Program perceptions and opinions in the replicated study (PP1-OP1, N=37)

		95% Confidence interval	
	Kappa value	Lower bound	Upper bound
Overall	0.189	–0.057	0.433
cmdline	0.308	–0.029	0.595
nametbl	0.043	–0.265	0.359
ntree	0.228	–0.097	0.590

Table 38 Agreement among technique opinions in the replicated study

				95% Confidence interval	
Question	N	Technique	Kappa value	Lower bound	Upper bound
OT1-OT2	38	Overall	0.652	0.411	0.858
		CR	0.649	0.350	0.887
		BT	0.627	0.001	1.000
		EP	0.665	0.410	0.892
OT1-OT3	38	Overall	0.652	0.405	0.865
		CR	0.649	0.319	0.907
		BT	0.627	0.000	0.907
		EP	0.665	0.408	0.887
OT2-OT3	38	Overall	1.000	1.000	1.000
		CR	1.000	1.000	1.000
		BT	1.000	1.000	1.000
		EP	1.000	1.000	1.000

- Participants' perceptions of technique effectiveness are related to how well they think they applied the techniques. They tend to think it is they, rather than the techniques, that are the obstacle to achieving more effectiveness (a possible evaluation apprehension threat has materialized).
- We have not been able to find a relationship between the technique they like best and find easiest to apply, and perceived effectiveness. Note however, that the technique participants think they have applied best is not necessarily the one that they have really best applied.
- Participants do not associate the simplest program with the program in which they detected most defects. This could be due to participants not properly interpreting the concept "simple".
- Opinions are consistent with each other.

Again, these results are confined to the validity limits imposed by the study.

8 Discussion

Next, we summarize the findings of this study and analyse their implications. Note that the results of the study are restricted to junior programmers with little testing experience, and defect detection techniques.

8.1 Answers to Research Questions

- *RQ1.1: What are participants' perceptions of their testing effectiveness?*
  The number of participants perceiving a particular technique/program as being more effective cannot be considered different for all three techniques/programs.
- *RQ1.2: Do participants' perceptions predict their testing effectiveness?*
  Our data do not support that participants correctly perceive the most effective technique for them. Additionally, no bias has been found towards a given technique. However, they tend to correctly perceive the program in which they detected most defects.

- *RQ1.3: Do participants find a similar amount of defects for all techniques?*
  Participants do not obtain similar effectiveness values when applying the different techniques.
- *RQ1.4: What is the cost of any mismatch?*
  Mismatch cost is not negligible (mean 31pp), and it is not related to the technique perceived as most effective.
- *RQ1.5: What is expected project loss?*
  Expected project loss is 15pp, and it is not related to the technique perceived as most effective.
- *RQ1.6: Are participants perceptions related to the number of defects reported by participants?*
  Results are not clear about this. Although our data do not support that participants correctly perceive the most effective technique for them, it should not be ruled out. Further research is needed.

Therefore, the answer to *RQ1: Should participants' perceptions be used as predictors of testing effectiveness?* is that participants should not base their decisions on their own perceptions, as they are not reliable and have an associated cost.

- *RQ2.1: What are participants' opinions about techniques and programs?*
  Most people like EP best, followed by both BT and CR (which merit the same opinion). There is no difference in opinion as regards programs
- *RQ2.2: Do participants' opinions predict their effectiveness?*
  They are not good predictors of technique effectiveness. A bias has been found towards EP.

Therefore, the answer to *RQ2: Can participants' opinions be used as predictors for testing effectiveness?* is that participants should not use their opinions, as they are not reliable and are biased.

- *RQ3.1: Is there a relationship between participants' perceptions and opinions?*
  Participants' perceptions of technique effectiveness are related to how well they think they applied the techniques. We have not been able to find a relationship between the technique they like best and find easiest to apply, and perceived effectiveness. Participants do not associate the simplest program with the program in which they detected most defect.
- *RQ3.2: Is there a relationship between participants' opinions?*
  Yes. Opinions are consistent with each other.

Therefore, the answer to *RQ3: Is there a relationship between participants' perceptions and opinions?* is positive for some of them.

8.2 About Perceptions

Participants' perceptions about the effectiveness of techniques are incorrect (50% get it wrong). However, this is not due to some sort of bias in favour of any of the three techniques under review. These misperceptions should not be overlooked, as they affect software quality. We cannot accurately estimate the cost, as it depends on what faults there are in the software. However, our data suggest a loss of from 25pp to 31 pp. Perceptions about programs appear to be correct, although this does not offset the mismatch cost.

Our findings **confirm** that:

- Testing technique effectiveness depends on the software faults.

Additionally, they **warn** developers that:

- They should not rely on their perceptions when rating a defect detection technique or how well they have tested a program.

Finally, they suggest the need for the following **actions**:

- Develop tools to inform developers about how effective the techniques that they applied are and the testing they performed is.
- Develop instruments to give developers access to experimental results.
- Conduct further empirical studies to learn what technique or combination of techniques should be applied under which circumstances to maximize its effectiveness.

8.3 About Opinions

Participants prefer EP to BT and CR (they like it better, think they applied it better and find it easier to apply). Opinions do not predict real effectiveness. This failure to predict reality is partly related to the fact that a lot of people prefer EP but are really more effective using BT or CR. Opinions do not predict real effectiveness with respect to programs either.

These findings **warn** developers that:

- They should not be led by their opinions on techniques when rating their effectiveness.

Finally, they suggest the need for the **action**:

- Further research should be conducted into what is behind developers' opinions.

8.4 About Perceptions and Opinions

The technique that participants believe to be the most effective is the one that they applied best. However, they are capable of separating their opinions about technique complexity and preferences from their perceptions, as the technique that they think is most effective is not the one that they find easiest to apply or like best.

Our findings **challenge** that:

- Perceptions of technique effectiveness are based on participants' preferences.

They also **warn** developers that:

- Maximum effectiveness is not necessarily achieved when a technique is properly applied.

Finally, they suggest the need for the following **actions**:

- Determine the best combination of techniques to apply that is at the same time easily applicable and effective.
- Continue to look for possible drivers to determine what could be causing developers' misperceptions.

9 Related Work

In recent years, several *experiments on defect detection technique effectiveness (static techniques and/or test-case design techniques) have been run with and without humans*. Experiments without human compare the efficiency and effectiveness of specification-based, code-based, and fault-based techniques, as for example the ones conducted by

Bieman and Schultz (1992), Hutchins et al. (1994), Offut et al. (1996), Offut and Lee (1994), Weyuker (1984) and Wong and Mathur (1995). Most of the experiments with humans evaluate static techniques, as for example the ones run by Basili et al. (1996), Biffl (2000), Dunsmore et al. (2002), Maldonado et al. (2006), Porter et al. (1995) and Thelin et al. (2004). Experiments evaluating test-case design techniques studied the efficiency and effectiveness of specification-based and control-flow-code-based techniques applied by humans, as the ones run by Basili and Selby (1987), Briand et al. (2004), Kamsties and Lott (1995), Myers (1978) and Roper et al. (1997). These experiments focus on strictly quantitative issues, leaving aside human factors like developers' perceptions and opinions.

There are *surveys that study developers' perceptions and opinions* with respect to different testing issues, like the ones performed by Deak (2012), Dias-Neto et al. (2016), Garousi et al. (2017), Gonçalves et al. (2017), Guaiani and Muccini (2015), Khan et al. (2010) and Marsden and Pérez Rentería y Hernández (2014). However, the results are not linked to quantitative issues. In this regard, some studies *link personality traits to preferences according to the role of software testers*, as for example Capretz et al. (2015), Kanij et al. (2015) and Kosti et al. (2014). However, there are no studies looking for a relationship between personality traits and quantitative issues like testing effectiveness.

There are some *approaches for helping developers to select the best testing techniques to apply under particular circumstances*, like the ones made by Cotroneo et al. (2013), Dias-Neto and Travassos (2014) or Vegas et al. (2009). Our study suggests that this type of research needs to be more widely disseminated to improve knowledge about techniques.

Finally, there are several ways in which developers can make decisions in the software deveelopment industry. The most basic approach is the classical perceptions and/or opinions, as reported in Dybå et al. (2005) and Zelkowitz et al. (2003). Other approaches suggest using classical decision-making models (Aurum and Wohlin 2002). Experiments can also be used for industry decision-making, as described by Jedlitschka et al. (2014). Devanbu et al. (2016) have observed the use of past experience (beliefs). More recent approaches advocate automatic decision-making from mining repositories (Bhattacharya 2012).

10 Conclusions

The goal of this paper was to discover whether developers' perceptions of the effectiveness of different code evaluation techniques are right in absence of prior experience. To do this, we conducted an empirical study with students plus a replication. The original study revealed that participants' perceptions are wrong. As a result, we conducted a replication aimed at discovering what was behind participants' misperceptions. We opted to study participants' opinions on techniques. The results of the replicated study corroborate the findings of the original study. They also reveal that participants' perceptions of technique effectiveness are based on how well they applied the techniques. We also found that participants' perceptions are not influenced by their opinions about technique complexity and preferences for techniques.

Based on these results, we derived some recommendations for developers: they should not trust their perceptions and be aware that correct technique application does not assure that they will find all the program defects.

Additionally, we identified a number of lines of action that could help to mitigate the problem of misperception, such as developing tools to inform developers about how effective their testing is, conducting more empirical studies to discover technique applicability conditions, developing instruments to allow easy access to experimental results, investigating other possible drivers of misperceptions or investigating what is behind opinions.

Future work includes running new replications of these studies to better understand their results.

Acknowledgments This research was funded by Spanish Ministry of Science, Innovation and Universities research grant PGC2018-097265-B-I00, the Regional Government of Madrid, under the FORTE-CM project (S2018/TCS-4314) and the Spanish Ministry of Economy and Business, under the MADRID project (TIN2017-88557-R).

Appendix A: Program Metrics

Table 39 shows the metrics collected for each program with the PREST tool. Note that all three programs show similar results for all metrics, except ntree that shows higher Halstead metrics. The size and complexity of cmdline is slightly higher compared to the other two programs.

Table 39 Metrics obtained with PREST

Metric	cmdline	nametbl	ntree
Total loc	289	289	247
Blank LOC	44	54	27
Comment LOC	6	5	5
Code and Comment LOC	0	0	0
Executable LOC	239	230	215
Unique Operands	91	121	89
Unique Operators	29	19	21
Total Operands	401	486	609
Total Operators	560	594	698
Halstead Vocabulary	120	140	110
Halstead Length	961	1080	1307
Halstead Volume	4600	5336	6143
Halsted Level	0.02	0.03	0.01
Halstead Difficulty	50.0	33.33	100.0
Halstead Effort	230000.0	177866.67	614300.0
Halstead Error	1.53	1.78	2.05
Halstead Time	12777.78	9881.48	34127.78
Branch Count	108	84	94
Decision Count	54	42	47
Call Pairs	41	57	35
Condition Count	51	40	39
Multiple Condition Count	16	16	17
Cyclomatic Complexity	37	27	31
Cyclomatic Density	0.15	0.12	0.14
Decision Density	1.06	1.05	1.21
Design Complexity	41	57	35
Design Density	1.11	2.11	1.13
Normalized Cyclomatic Complexity	0.13	0.09	0.13
Formal Parameteres	0	0	0
Risk Level	False	False	False

Appendix B: Analysis of the Original Experiment

Figure 3 shows the boxplot, and Table 40 shows the descriptive statistics for observed technique effectiveness.

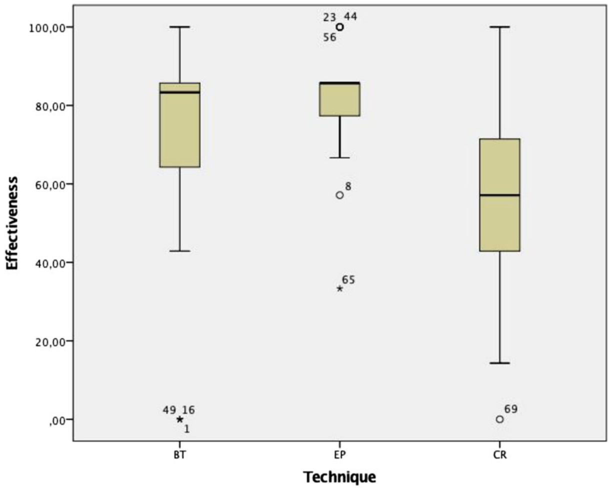

Fig. 3 Boxplot for observed technique effectiveness in the original study

We find that the mean and median effectiveness of BT is highest, followed by EP and then by CR. Additionally, EP has a lower variance. The 95% confidence interval suggests that EP is more effective than CR and that BT is as effective as EP and CR. This is an interesting result, as all faults injected in the code could be detected by all techniques. Additionally, it could indicate that code reading is more dependent on experience that you cannot acquire in a 4-hour training.

All three techniques have outliers corresponding to participants that have performed exceptionally bad: 3 in the case of BT (all 3 participants scored 0), 2 in the case of EP and 1 in the case of CR (also scoring 0). Additionally, EP has 3 outliers corresponding to participants that have performed exceptionally well (all scoring 100). In none of the cases the values belong to the same participant (which could suggest that outliers correspond to participants that performed exceptionally bad with one technique, but not all three). Additionally, CR shows a higher variability, which could indicate that it is more dependent on the person applying the technique.

Table 40 Descriptive statistics for observed technique effectiveness in the original study

Technique	Mean	Std. Deviation	95% Confidence interval Lower Bound	95% Confidence interval Upper Bound
CR	52.795	24.904	42.025	63.564
BT	66.251	33.691	51.682	80.820
EP	82.089	15.921	75.204	88.973

Table 41 Type III tests of fixed effects in the original study

Source	Numerator df	Denominator df	F	Sig.
Intercept	1	57.182	338.655	0.000
Group	5	45.470	0.259	0.933
Technique	2	54.522	16.648	0.000
Program	2	51.485	6.852	0.002
Technique * Program	4	61.073	0.583	0.676

The experimental data has been analysed with a Linear Mixed-Effects Model (SPSS v26 MIXED procedure). Group, program, technique and the program by technique interaction are fixed effects and subject is a random effect. Eleven models were tried, choosing the one with the lowest AIC:

- One pure random effects model and no repeated measures.
- Five models specifying program as repeated measures effect.
- Five models specifying technique as repeated measures effect.

The five models differed in the covariance structures used (identity, diagonal, first-order autoregressive, compound symmetry and unstructured).

In this first analyses the MIXED procedure did not achieve convergence. We then decided to relax the model by removing subject from the random effects list. We re-run analyses, but this time we found severe departures from normality, leading to non-reliable results. Next step consisted on data transformation. The chosen transformation, square, solved the normality issues, and the MIXED procedure converged.

Table 41 shows the results for the model chosen (program is a repeated measures effect, and the covariance structure is Diagonal). Figure 4 shows residuals normality.

Table 41 shows that technique and program are both statistically significant. Group and the technique by program interaction are not significant. The Bonferroni multiple comparisons tests show that:

- All three techniques show different effectiveness (EP>BT>CR).
- ntree shows a higher defect detection rate than the other two programs (ntree>(cmdline=nametbl)).

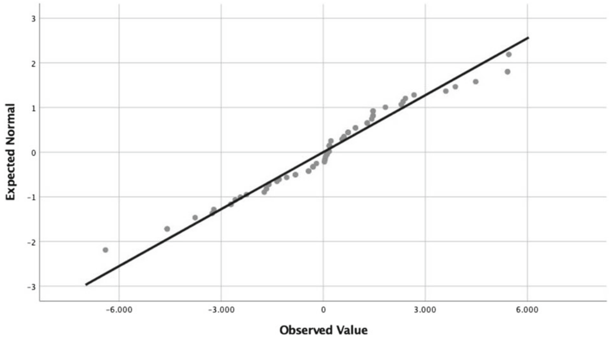

Fig. 4 Normal Q-Q plot of residuals in the original study

Table 42 Estimated marginal means for technique in the original study

				95% Confidence interval	
Technique	Mean	Std. error	df	Lower bound	Upper bound
BT	74.307	3.309	53.279	67.199	80.793
EP	85.486	2.864	57.329	79.477	91.099
CR	57.502	4.118	53.212	48.096	65.572

Table 43 Estimated marginal means for program in the original study

				95% Confidence interval	
Program	Mean	Std. error	df	Lower bound	Upper bound
cmdline	66.562	4.655	21.854	55.969	75.686
nametbl	69.642	2.271	23.153	64.886	74.093
ntree	82.798	3.117	21.900	76.201	88.907

Tables 42 and 43 show the estimated marginal means for both technique and program. Note that the mean, std. error and 95% confidence interval bounds values have been untransformed.

Finally, Fig. 5 shows the profile plot with error bars for the program by technique interaction. Although the interaction is not statistically significant, the profile plot suggests that nametbl could be behaving differently for CR. This could mean that lack of significance of the interaction in the analysis could be due to sample size.

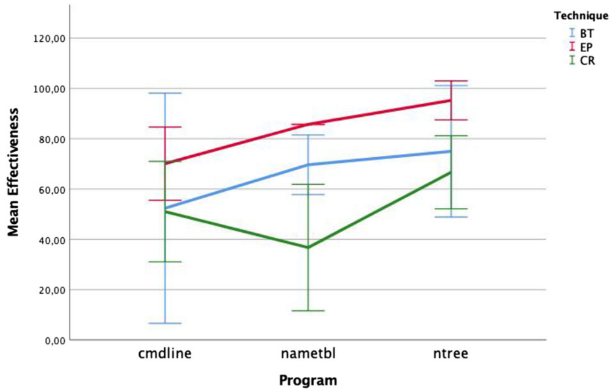

Fig. 5 Profile plot for program by technique interaction in the original study

Appendix C: Analysis of the Replicated Experiment

Figure 6 shows the boxplot for observed technique effectiveness. All three techniques show a similar median, and the same range. Compared to the original study, the behaviour of the techniques is much more homogeneous.

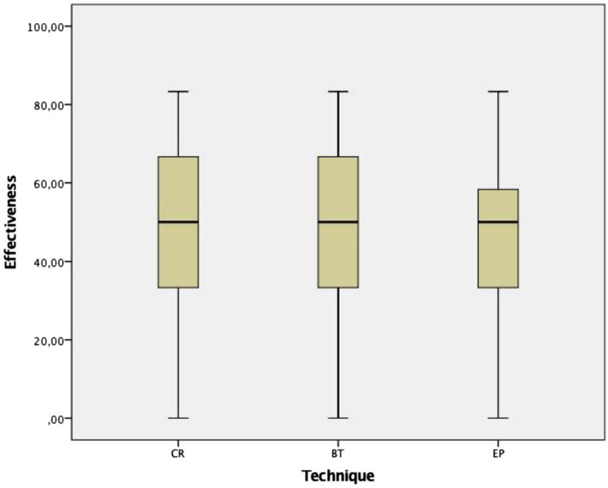

Fig. 6 Boxplot for observed technique effectiveness in the replicated study

Table 44 shows the descriptive statistics for observed technique effectiveness. We find that the effectiveness of all three techniques is similar. CR has a similar effectiveness as in the original study suggesting again that perhaps 4 hours of training are not enough for learning the code review technique). However, the mean effectiveness for BT and EP has dropped from 66% and 82%, respectively, in the original study to 46%. Note that in this study, the nature of the faults has been changed. While in the original study all defects can be detected by all techniques, in this study some defects cannot be exercised by testing techniques. Therefore, a possible explanation for the change in effectiveness of testing techniques could be the faults seeded in the programs. However, this is just a mere hypothesis that need to be tested.

Figure 7 shows the boxplot for the percentage of defects found in each program. It is interesting to see that the median detection rate for nametbl and ntree is higher than for cmdline. These results could be attributed to cmdline having a slightly higher size and

Table 44 Descriptive statistics for observed technique effectiveness in the replicated study

Technique	Mean	Std. Deviation	95% Confidence interval	
			Lower bound	Upper bound
CR	52.564	22.793	45.175	59.953
BT	46.153	17.297	40.546	51.760
EP	46.581	17.597	40.876	52.285

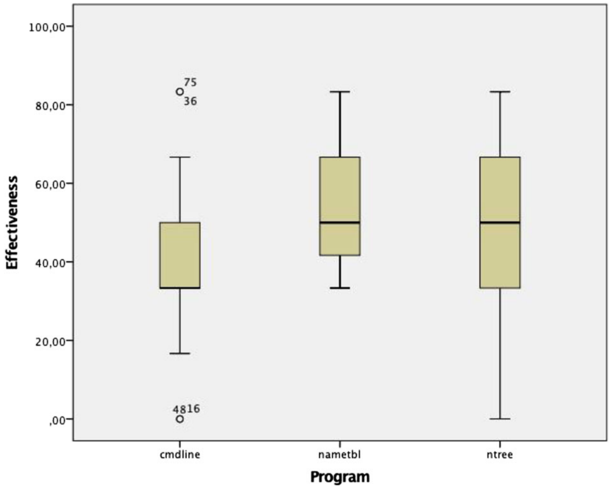

Fig. 7 Boxplot for observed program detection rate in the Replicated Study

Table 45 Descriptive statistics for program defect detection rate in the replicated study

Program	Mean	Std. Deviation	95% Confidence interval Lower bound	95% Confidence interval Upper bound
cmdline	39.315	20.407	32.700	45.931
nametbl	53.419	14.400	48.751	58.087
ntree	52.564	20.065	46.059	59.068

complexity. Additionally, cmdline shows 4 outliers, reflecting 4 people who performed exceptionally well, and 2 people who performed exceptionally bad (scoring 0). Finally, ntree shows a higher range compared to the other two programs (note that ntree shows higher Halstead metrics). This is an unexpected result, as nametbl and ntree are very similar in terms of complexity.

Table 45 shows the descriptive statistics for defect detection rate in programs. It is higher for nametbl and ntree than for cmdline. This result could be due to cmdline having a slightly higher complexity than the other two programs.

Table 46 Type III tests of fixed effects in the replicated study

Source	Numerator df	Denominator df	F	Sig.
Intercept	1	42.788	858.926	0.000
Group	5	64.916	1.930	0.101
Technique	2	76.721	1.817	0.169
Program	2	57.386	8.308	0.001
Technique * Program	4	77.253	0.796	0.532

We started the analysis of the replicated study data with the same model as in the original study. This time we had neither convergence nor normality issues. The model chosen was: group, program, technique and the program by technique interaction are fixed effects and subject is a random effect. Program is a repeated measures effect, and the covariance structure is Diagonal. Table 46 shows the results of the analysis of the best model. Figure 8 shows residuals normality.

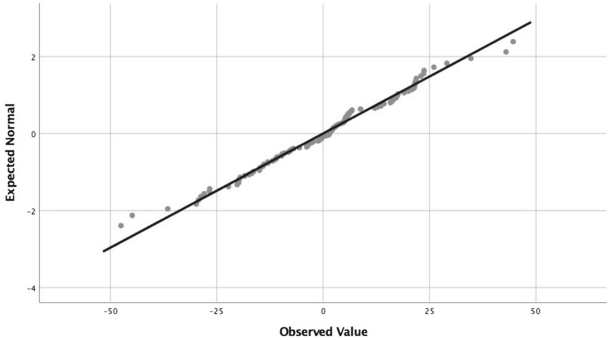

Fig. 8 Normal Q-Q plot of residuals in the replicated study

Table 46 shows that program is statistically significant. Group, technique and the technique by program interaction are not significant. The Bonferroni multiple comparisons tests show that cmdline shows a lower defect detection rate than the other two programs (cmdline<(cmdline=nametbl)).

Tables 47 and 48 show the estimated marginal means for both technique and program.

Table 47 Estimated marginal means for technique in the replicated study

Technique	Mean	Std. error	df	95% Confidence interval Lower bound	Upper bound
CR	52.871	2.782	106.039	47.356	58.386
BT	47.041	2.765	106.927	41.560	52.523
EP	46.020	2.817	101.380	40.432	51.608

Table 48 Estimated marginal means for program in the replicated study

Program	Mean	Std. error	df	95% Confidence interval Lower bound	Upper bound
cmdline	39.379	3.100	36.670	33.096	45.662
nametbl	53.962	2.065	38.280	49.783	58.142
ntree	52.591	3.075	38.282	46.368	58.814

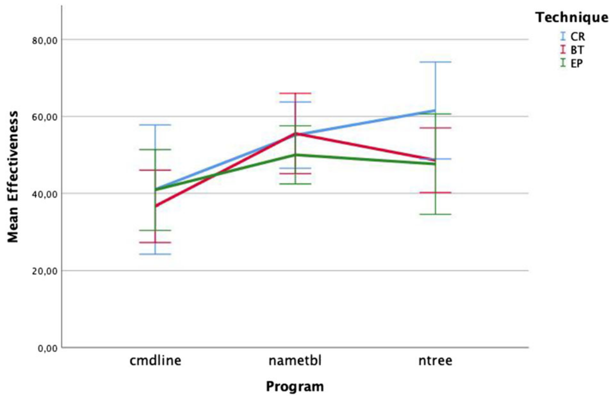

Fig. 9 Profile plot for program by technique interaction in the replicated study

Finally, Fig. 9 shows the profile plot with error bars for the program by technique interaction. In this case, the profile plot suggests that no interaction exists, which means the non-significance of the interaction in the analysis could not be due to sample size.

Appendix D: Joint Analyses

D.1 RQ1.1: Participants' Perceptions

Table 49 shows the percentage of participants that perceive each technique to be the most effective. We cannot reject the null hypothesis that the frequency distribution of the responses to the questionnaire item (*Using which technique did you detect most defects?*) follows a uniform distribution ($\chi^2(2,N=60)=1.900$, $p=0.387$). This means that the number of participants perceiving a particular technique as being more effective cannot be considered different for all three techniques. **Our data do not support the conclusion that techniques are differently frequently perceived as being the most effective**.

Table 49 Participants' perceptions of technique effectiveness in the joint analysis

N	CR	BT	EP	Result
60	30.00%	28.33%	41.67%	CR=BT=EP

D.2 RQ1.2: Comparing Perceptions with Reality

Table 50 shows the value of kappa and its 95% CI, overall and for each technique separately. We find that all values for kappa with respect to the questionnaire item (*Using which technique did you detect most defects?*) are consistent with lack of agreement, except for CR (κ <0.4, poor). This means that **our data do not support the conclusion that participants correctly perceive the most effective technique for them**.

Table 50 Agreement between perceived and real technique effectiveness in the joint analysis

	Kappa value	95% Confidence interval	
		Lower bound	Upper bound
Overall	0.241	0.059	0.444
CR	0.420	0.170	0.648
BT	0.116	–0.136	0.393
EP	0.168	–0.092	0.430

As lack of agreement cannot be ruled out, we examine whether the perceptions are biased. The results of the Stuart-Maxwell test show that the null hypothesis of existence of marginal homogeneity cannot be rejected (χ^2(2,N=60)=2.423, p=0.298). Additionally, the results of the McNemar-Bowker test show that the null hypothesis of existence of symmetry cannot be rejected (χ^2(3,N=60)=2.552, p=0.466). This means that we cannot conclude

Table 51 Agreement between percentage of defects found with each technique in the joint analysis

Sample	N		Krippendorff's Alpha value	95% Confidence interval	
				Lower bound	Upper bound
All	61	CR-BT	–0.1074	–0.4575	0.1887
		CR-EP	–0.0735	–0.4893	0.2888
		EP-BT	0.2276	–0.1096	0.5574
		CR-BT-EP	0.0264	–0.1893	0.2108
G1	11	CR-BT	–0.1005	–0.8416	0.5812
		CR-EP	0.4410	0.0021	0.7586
		EP-BT	–0.1573	–1.0000	0.5615
		CR-BT-EP	0.0204	–0.6032	0.4626
G2	11	CR-BT	0.0843	–0.3277	0.4351
		CR-EP	0.2121	–0.1489	0.5262
		EP-BT	0.4109	–0.1718	0.8474
		CR-BT-EP	0.2015	–0.0429	0.4289
G3	12	CR-BT	–0.1142	–0.8008	0.4459
		CR-EP	-0.2242	–1.0000	0.5142
		EP-BT	0.2978	–0.4059	0.7313
		CR-BT-EP	0.0397	–0.3692	0.3948
G4	11	CR-BT	–0.1320	–0.9523	0.4963
		CR-EP	-0.1295	–1.0000	0.6278
		EP-BT	0.7483	0.5245	0.9161
		CR-BT-EP	0.1263	–0.3618	0.5116
G5	4	CR-BT	–0.4382	–1.0000	0.5625
		CR-EP	–0.2669	–0.8214	0.2538
		EP-BT	0.1131	–0.4175	0.6056
		CR-BT-EP	–0.1471	–0.5263	0.2249
G6	2	CR-BT	–0.3766	–1.0000	0.7864
		CR-EP	–0.5468	–1.0000	0.8683
		EP-BT	0.7689	0.6768	0.9060
		CR-BT-EP	–0.1187	–1.0000	0.7590

that there is directionality when participants' perceptions are wrong. These two results suggest that participants are not differently mistaken about one technique as they are about the others. **Techniques are not differently subject to misperceptions**.

D.3 RQ1.3: Comparing the Effectiveness of Techniques

We are going to check if misperceptions could be due to participants detecting the same amount of defects with all three techniques, and therefore being impossible for them to make the right decision. Table 51 shows the value and 95% CI of Krippendorff's α overall and for each pair of techniques, for all participants and for every design group (participants that applied the same technique on the same program) separately, and Table 52 shows the value and 95% CI of Krippendorff's α overall and for each program/session. For values with all participants, we can rule out agreement ($\alpha < 0.4$) except for the case of EP-BT and nametbl-ntree for which the upper bound of the 95% CIs are consistent with fair to good agreement. However, even in this two cases, 0 belongs to the 95% CIs, meaning that agreement by chance cannot be ruled out. This means that **participants do not obtain similar effectiveness values when applying the different techniques (testing the different programs) so as to be difficult to discriminate among techniques/programs**. As regards the results for groups, the 95% CIs are too wide to show reliable results.

Table 52 Agreement between percentage of defects found with each program in the joint analysis (N=61)

	Krippendorff's	95% Confidence interval	
	Alpha value	Lower bound	Upper bound
cmdline-nametbl	–0.0942	–0.5398	0.2782
cmdline-ntree	0.0551	–0.2394	0.3211
nametbl-ntree	0.0685	–0.3615	0.4256
cmdline-nametbl-ntree	0.0242	–0.1831	0.2178

D.4 RQ1.4: Cost of Mismatch

Table 53 and Fig. 10 show the cost of mismatch. We can see that the CR technique has fewer mismatches compared to the other two. Although the BT and EP techniques have the same

Table 53 Observed reduction in technique effectiveness for mismatch

		Cost		
Technique	No. mismatches	Mean	Median	Std. deviation
CR	5(18)	24pp	17pp	11
BT	12(17)	30pp	17pp	24
EP	13(25)	25pp	17pp	13
TOTAL	30(60)	27pp	17pp	18

Column 2 shows the number of mismatches out of the total number of participants who perceived the technique as being most effective. Columns 3-5 show the mean, median and standard deviation for mismatch cost (in percentage points)

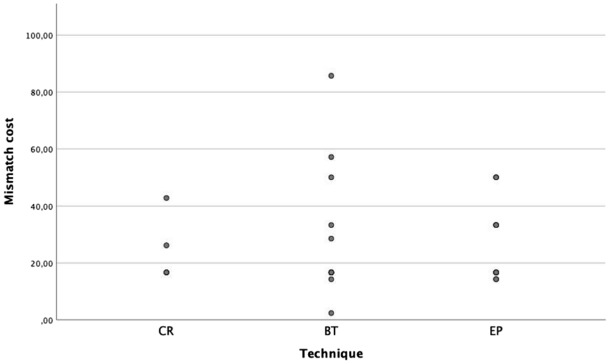

Fig. 10 Scatterplot for observed mismatch cost in the original study

number of mismatches, BT shows a higher dispersion. The results of the Kruskal-Wallis test reveal that we cannot reject the null hypothesis of techniques having the same mismatch cost (H(2)=0.034, p=0.983). This means that we cannot claim a difference in mismatch cost between the techniques. The estimated mean mismatch cost is 27pp (median 17pp).

These results suggest that **the mismatch cost is not negligible** (27pp), **and is not related to the technique perceived as most effective.**

D.5 RQ1.5: Expected Loss of Effectiveness

Table 54 shows the average loss of effectiveness that should be expected in a project. Again, the results of the Kruskal-Wallis test reveal that we cannot reject the null hypothesis of techniques having the same expected reduction in technique effectiveness for a project (H(2)=5.680, p=0.058). This means we cannot claim a difference in project effectiveness loss between techniques. The mean expected loss in effectiveness in the project is estimated as 13pp.

These results suggest that **the expected loss in effectiveness in a project is not negligible** (15pp), **and is not related to the technique perceived as most effective.**

Table 54 Observed reduction in technique effectiveness in a software project

		Cost		
Technique	N	Mean	Median	Std. deviation
CR	18	7pp	0pp	12
BT	17	21pp	17pp	24
EP	25	13pp	14pp	16
TOTAL	60	13pp	1pp	18

Column 2 shows the number of (mis)matches. Columns 3-5 show the mean, median and std. deviation for the reduction in effectiveness in the project (in percentage points)

References

Altman D (1991) Practial statistics for medical research. Chapman and Hall

Aurum A, Wohlin C (2002) Applying decision-making models in requirements engineering. In: Proceedings of requirements engineering for software quality

Banerjee MV, Capozzoli M, McSweeney L, Sinha D (1999) Beyond kappa: a review of interrater agreement measures. Can J Stat 27:3–23

Basili V, Selby R (1987) Comparing the effectiveness of software testing strategies. IEEE Trans Softw Eng 13(2):1278–1296

Basili V, Green S, Laitenberger O, Lanubile F, Shull F, Sorumgard S, Zelkowitz M (1996) The empirical investigation of perspective based reading. Empir Softw Eng 1(2):133–164

Beizer B (1990) Software testing techniques, 2nd edn. International Thomson Computer Press

Bhattacharya P (2012) Quantitative decision-making in software engineering. Ph.D. thesis University of California Riverside

Bieman J, Schultz J (1992) An empirical evaluation (and specification) of the all-du-paths testing criterion. Softw Eng J, 43–51

Biffl S (2000) Analysis of the impact of reading technique and inspector capability on individual inspection performance. In: 7th Asia-Pacific software engineering conference, pp 136–145

Briand L, Penta M, Labiche Y (2004) Assessing and improving state-based class testing: a series of experiments. IEEE Trans Softw Eng 30(11):770–793

Capretz L, Varona D, Raza A (2015) Influence of personality types in software tasks choices. Comput Hum Behav 52:373–378

Cotroneo D, Pietrantuono R, Russo S (2013) Testing techniques selection based on odc fault types and software metrics. J Syst Softw 86(6):1613–1637

Deak A (2012) Understanding socio-technical factors influencing testers in software development organizations. In: 36th Annual computer software and applications conference (COMPSAC'12), pp 438–441

Devanbu P, Zimmermann T, Bird C (2016) Belief & evidence in empirical software engineering. In: Proceedings of the 38th international conference on software engineering, pp 108–119

Dias-Neto A, Travassos G (2014) Supporting the combined selection of model-based testing techniques. IEEE Trans Softw Eng 40(10):1025–1041

Dias-Neto A, Matalonga S, Solari M, Robiolo G, Travassos G (2016) Toward the characterization of software testing practices in south america: looking at Brazil and Uruguay. Softw Qual J, 1–39

Dieste O, Aranda A, Uyaguari F, Turhan B, Tosun A, Fucci D, Oivo M, Juristo N (2017) Empirical evaluation of the effects of experience on code quality and programmer productivity: an exploratory study. Empirical Software Engineering. https://doi.org/10.1007/s10664-016-9471-3

Dunsmore A, Roper M, Wood M (2002) Further investigations into the development and evaluation of reading techniques for object-oriented code inspection. In: 24th International conference on software engineering, pp 47–57

Dybå T, Kitchenham B, Jorgensen M (2005) Evidence-based software engineering for practitioners. IEEE Softw 22(1):58–65

Everitt B (2000) The analysis of contingency tables. In: Monographs statistics and applied probability, vol 45. Chapman & Hall/CRC

Falessi D, Juristo N, Wohlin C, Turhan B, Münch J, Jedlitschka A, Oivo M (2017) Empirical software engineering experts on the use of students and professionals in experiments. Empirical Software Engineering. https://doi.org/10.1007/s10664-017-9523-3

Fleiss J, Levin BMP (2003) Statistical methods for rates and proportions, 3rd edn. Wiley

Garousi V, Felderer M, Kuhrmann M, Herkiloğlu K (2017) What industry wants from academia in software testing?: hearing practitioners' opinions. In: Proceedings of the 21st international conference on evaluation and assessment in software engineering, EASE'17, pp 65–69

Gonçalves W, de Almeida C, de Araújo LL, Ferraz M, Xandú R, de Farias I (2017) The influence of human factors on the software testing process: the impact of these factors on the software testing process. In: 2017 12th Iberian conference on information systems and technologies (CISTI), pp 1–6

Guaiani F, Muccini H (2015) Crowd and laboratory testing, can they co-exist? An exploratory study. In: 2nd International workshop on crowdsourcing in software engineering (CSI-SE), pp 32–37

Hayes A, Krippendorff K (2007) Answering the call for a standard reliability measure for coding data. Commun Methods Meas 1:77–89

Hutchins M, Foster H, Goradia T, Ostrand T (1994) Experiments on the effectiveness of dataflow- and controlflow-based test adequacy criteria. In: Proceedings of the 16th international conference on software engineering, pp 191–200

Jedlitschka A, Juristo N, Rombach D (2014) Reporting experiments to satisfy professionals' information needs. Empir Softw Eng 19(6):1921–1955
Kamsties E, Lott C (1995) An empirical evaluation of three defect-detection techniques. In: Proceedings of the Fifth European software engineering conference, pp 84–89
Kanij T, Merkel R, Grundy J (2015) An empirical investigation of personality traits of software testers. In: 8th International workshop on cooperative and human aspects of software engineering (CHASE'15), pp 1–7
Khan T, Pezeshki V, Clear F, Al-Kaabi A (2010) Diverse virtual social networks: implications for remote software testing teams. In: European, mediterranean & middle eastern conference on information systems
Kocaguneli E, Tosun A, Bener A, Turhan B, Caglayan B (2009) Prest: an intelligent software metrics extraction, analysis and defect prediction tool, 637–642
Kosti M, Feldt R, Angelis L (2014) Personality, emotional intelligence and work preferences in software engineering: an empirical study. Inf Softw Technol 56(8):973–990
Kuehl R (2000) Design of experiments: statistical principles of research design and analysis, 2nd edn. Duxbury Thomson Learning
Landis J, Koch G (1977) The measurement of observer agreement for categorical data. Biometrics 33:159–174
Linger R (1979) Structured programming: theory and practice (the systems programming series). Addison-Wesley
Maldonado J, Carver J, Shull F, Fabbri S, Dória E., Martimiano L, Mendonça M, Basili V (2006) Perspective-based reading: a replicated experiment focused on individual reviewer effectiveness. Empir Softw Eng 11(1):119–142
Marsden N, Pérez Rentería y Hernández T (2014) Understanding software testers in the automotive industry a mixed-method case study. In: 9th International conference on software engineering and applications (ICSOFT-EA), pp 305–314
Massey A, Otto P, Antón A (2015) Evaluating legal implementation readiness decision-making. IEEE Trans Softw Eng 41(6):545–564
Myers G (1978) A controlled experiment in program testing and code walkthroughs/inspections. Commun ACM 21(9):760–768
Myers G, Badgett T, Sandler C (2004) The art of software testing, 2nd edn. Wiley-Interscience
Octaviano F, Felizardo K, Maldonado J, Fabbri S (2015) Semi-automatic selection of primary studies in systematic literature reviews: is it reasonable? Empir Softw Eng 20(6):1898–1917
Offut A, Lee S (1994) An empirical evaluation of weak mutation. IEEE Trans Softw Eng 20(5):337–344
Offut A, Lee A, Rothermel G, Untch R, Zapf C (1996) An experimental determination of sufficient mutant operators. ACM Trans Softw Eng Methodol 5(2):99–118
Porter A, Votta L, Basili V (1995) Comparing detection methods for software requirements inspection: a replicated experiment. IEEE Trans Softw Eng 21(6):563–575
Roper M, Wood M, Miller J (1997) An empirical evaluation of defect detection techniques. Inf Softw Technol 39:763–775
Shull F, Carver J, Vegas S, Juristo N (2008) The role of replications in empirical software engineering. Empir Softw Eng 13:211–218
Thelin T, Runeson P, Wohlin C, Olsson T, Andersson C (2004) Evaluation of usage-based reading—conclusions after three experiments. Empir Softw Eng 9:77–110
Vegas S, Basili V (2005) A characterisation schema for software testing techniques. Empir Softw Eng 10(4):437–466
Vegas S, Juristo N, Basili V (2009) Maturing software engineering knowledge through classifications: a case study on unit testing techniques. IEEE Trans Softw Eng 35(4):551–565
Weyuker E (1984) The complexity of data flow criteria for test data selection. Inf Process Lett 19(2):103–109
Wohlin C, Runeson P, Höst M., Ohlsson MC, Regnell B, Wesslén A (2014) Experimentation in software engineering: an introduction, 2nd edn. Springer
Wong E, Mathur A (1995) Fault detection effectiveness of mutation and data-flow testing. Softw Qual J 4:69–83
Zapf A, Castell S, Morawietz L, Karch A (2016) Measuring inter-rater reliability for nominal data – which coefficients and confidence intervals are appropriate? BMC Med Res Methodol, 16(93)
Zelkowitz M, Wallace D, Binkley D (2003) Experimental validation of new software technology. Series Softw Eng Knowl Eng 12:229–263

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Sira Vegas has been associate professor of software engineering with the School of Computer Engineering at the Technical University of Madrid, Spain, since 2008. Sira belongs to the review board of IEEE Transactions on Software Engineering, and is a regular reviewer of the Empirical Software Engineering Journal. She was program chair for the International Symposium on Empirical Software Engineering and Measurement in 2007.

Patricia Riofrío received her BSc in Computer Science and her MSc in Software and Systems from the Polytechnic University of Madrid, Spain. She is a PhD student at the Polytechnic University of Madrid. Her main research interest is empirical software engineering.

Esperanza Marcos is a full professor at the Universidad Rey Juan Carlos of Madrid, where she leads the Kybele R&D group. She received her Ph.D. from the Technical University of Madrid in 1997 and her current research interests include Service Science, Model-Driven engineering and human aspects of Software Engineering.

Natalia Juristo has been full professor of software engineering with the School of Computer Engineering at the Technical University of Madrid, Spain, since 1997. She was awarded a FiDiPro (Finland Distinguished Professor Program) professorship at the University of Oulu, from January 2013 until June 2018. Natalia belongs to the editorial board of EMSE and STVR. In 2009, Natalia was awarded an honorary doctorate by Blekinge Institute of Technology in Sweden.