Confirmation Bias and Time Pressure: A Family of Experiments in Software Testing

Iflaah Salman, Burak Turhan, *Senior Member, IEEE*, Robert Ramač, and Vladimir Mandić, *Senior Member, IEEE*

***Abstract*—Background: Software testers manifest confirmation bias (the cognitive tendency) when they design relatively more specification consistent test cases than specification inconsistent test cases. Time pressure may influence confirmation bias of testers per the research in the psychology discipline. Objective: We examine the manifestation of confirmation bias of software testers while designing functional test cases, and the effect of time pressure on confirmation bias in the same context. Method: We executed one internal and two external experimental replications concerning the original experimentation in Oulu. We analyse individual replications and meta-analyse our family of experiments (the original and replications) for joint results on the phenomena. Results: Our findings indicate a significant manifestation of confirmation bias by software testers during the designing of functional test cases. Time pressure significantly promoted confirmation bias among testers per the joint results of the family. The different experimental sites affected the results; however, we did not detect any effects of site-specific variables. Conclusion: Software testers should develop an outside-of-the-box thinking attitude to counter the manifestation of confirmation bias. Time pressure can be manoeuvred by centring manual suites on the designing and consequently the execution of inconsistent test cases, while automated testing focuses on consistent ones.**

***Index Terms*—Software testing, experiment, replication, aggregation, cognitive bias, confirmation bias, time pressure.**

I. Introduction

THE cognitive human tendency to seek evidence that confirms prior beliefs instead of seeking disconfirming evidence is referred to as *confirmation bias* [1], [2], [3]. Among thirty-seven cognitive biases investigated in the software engineering (SE) literature[1], confirmation bias is the second most investigated cognitive bias after anchoring/adjustment bias [2]. It is the only investigated cognitive bias in the software testing knowledge area of SE [2]. Confirmation bias is reported to have adverse effects on software testing that ultimately deteriorate the software quality [2]. For example, it leads to higher defect rates and increased post-release defects [4], [5]. Confirmation bias promotes the execution of test cases that validate the correct functioning of the system instead of test cases that reveal failures [6], [7].

The SE literature reports multiple antecedents to confirmation bias in software testing [2], [4], [8], [9]. Çalikli and Bener observed lower levels of confirmation bias among participants who were experienced but inactive in testing or development [4]. Higher confirmation bias levels were observed among participants who lacked training in logical reasoning and mathematical proofreading skills [4]. Teasley et al. found completeness of the specifications to diminish the manifestation of confirmation bias [8]. The literature also investigated the effects of organisational factors on confirmation bias. For example, company culture was found to affect the levels of confirmation bias [10]. However, company size and development methods showed no effects on confirmation bias [9].

Time pressure is a substantial and recognised organisational factor of the software industry [11], [12], [13], [14], [15], [16]. It is observed to have both positive and negative effects on software engineering. In a study by Mantylä et al., time pressure improved the efficiency for test case development and requirements review [11]. On the other hand, time pressure was found as a demotivating attribute for software process improvement [17]. Software testers perceive it negatively for software quality because it is a source of errors when they have insufficient or limited time for testing [14], [15], [16], which leads further to the accumulation of the test debt [13].

In psychology, time pressure is a cognitive load that leads to the manifestation of confirmation bias [18], [19]. It is therefore susceptible, time pressure (an organisational factor) acts as an antecedent to confirmation bias in software engineering. There was no study in software engineering that validated this postulate until we—Salman et al. executed a controlled experiment for its validation in the context of functional software testing [20]. The study also examined the manifestation of confirmation bias by software testers [20]. We found a significant manifestation of confirmation bias by software testers [20]. However, we could not observe time pressure to affect or promote confirmation bias among testers [20].

Manuscript received 3 February 2023; revised 11 September 2023; accepted 23 October 2023. Date of publication 8 November 2023; date of current version 12 December 2023. The work of Iflaah Salman was supported in part by the Infotech Oulu Doctoral grant at the University of Oulu. Recommended for acceptance by K. Blincoe. *(Corresponding authors: Iflaah Salman; Vladimir Mandić.)*

Iflaah Salman is with the School of Engineering Science, Lappeenranta-Lahti University of Technology, FI-15210 Lahti, Finland (e-mail: iflaah.salman@lut.fi).

Burak Turhan is with M3S Research Unit, University of Oulu, FI-90014 Oulu, Finland, and also with the Faculty of IT, Monash University, VIC 3800 Melbourne, Australia (e-mail: burak.turhan@oulu.fi).

Robert Ramač and Vladimir Mandić are with the Faculty of Technical Sciences, University of Novi Sad, 21000 Novi Sad, Serbia (e-mail: ramac.robert@uns.ac.rs; vladman@uns.ac.rs).

Digital Object Identifier 10.1109/TSE.2023.3330400

[1]Refer to a systematic mapping study by Mohanani et al. [2] for a complete list of cognitive biases investigated in SE.

In this study, we aim to investigate the same objective as of our original experimental study [20]: *whether testers manifest confirmation bias while designing functional test cases and how it is affected by time pressure?* We achieve this aim by executing a series of experimental replications of the original experiment. In this respect, we performed three *similar experimental replications*; one internal at Oulu, Finland and two external at Novi Sad, Serbia. This sums up the total number of experiments to four, including the original one executed in Oulu. We refer to them as a *family of experiments* because all experiments are similar replications with the common objective [21], [22], which allows us to investigate whether contextual variables affect the results [23], [24]. Therefore, in addition to reporting the results of individual replications, we also present the joint results of our family of experiments on the manifestation of confirmation bias and the role of time pressure.

The results showed the significant manifestation of confirmation bias by software testers during the design of functional test cases. All four controlled experiments in our family provided this evidence. Time pressure significantly promoted confirmation bias among software testers per the joint, aggregated, analysis of the family. The experience of participants did not affect the results. There are effects of the two different sites, Oulu and Novi Sad, on the results. However, the size of the data could not support to observe the effects of the site-specific contextual variables on the aggregated results.

We contribute to the body of knowledge by:

- Providing empirical evidence of the manifestation of confirmation bias by software testers during functional software testing.
- Providing empirical evidence that time pressure promotes confirmation bias among software testers in the same context with a relatively large sample size, compared to individual experiments on the topic.
- The first family of experiments via experimental replications on confirmation bias and time pressure in functional software testing.
- This family of experiments is the first to present joint results via meta analysis on the topic under investigation.

This study is reported following the replication guidelines proposed by Carver [22]. Section II states the need for replications. The following section - Section III presents background on confirmation bias and time pressure. Section IV introduces the original study followed by Section V elaborating on replication studies. Analyses methods are introduced in Section VI and experimental execution along with data collection is given in Section VII. Results are presented in Section VIII, followed by discussion and comparison of the results with the relevant studies in Section IX. Validity threats are discussed in Section X and we conclude the study in Section XI.

II. The Need for Replications

Experiments in SE are usually isolated events [25]. The results of a single isolated experiment cannot represent the ground reality of the examined phenomenon [26]. In turn, the applicability of such results cannot be generalized [27], [28]. Compared to disciplines like social sciences and medicine, SE experiments often suffer from small sample sizes [27], [28]. This intensifies the risks of generalizability due to a single study [27].

Replication is a core element of experimentation [25], [29]. It helps verify the experimental results by reproducing them under different conditions to strengthen a body of knowledge [25], [29], [30]. The results are beneficial whether they are similar to or different from the original experiment [30]. As the former strengthens the evidence and the latter provides insight into the differentiating, yet influencing factors [30]. Without replications, it is impossible to know whether the results of an experiment pertain to experimental configuration, chance outcome or a reflection of reality [29]. The infancy of experimental knowledge in SE impeded performing replications [29]. Mandić et al. state difficulty in acquiring resources, a lack of realism and the issue of implementing findings in practice are hurdles to replications in SE [31]. However, for quite some time, researchers in SE have been forming replication groups that share the same goal (family of experiments) by executing multiple replications [24], [28]. A family of experiments not only addresses the mentioned limitations of isolated experiments but also improves the reliability and internal validity of joint conclusions [28]. Families facilitate this because researchers who execute them have access to the raw data of every experiment (replication) and can control the changes across the replications [24], [28].

There are five experimental elements that can vary or retain along a replication deeming its purpose: *experimenter*, *site*, *protocol*, *operationalisation* and *population* [25]. When all the specified or basic (protocol, operationalisation) elements remain the same in a replication, compared to the original (baseline) experiment, then the purpose is to *control sampling error*. It is to verify that the results of the original experiment are "not chance outcomes" due to a Type-I error [25], [29]. To verify that the results of the original experiment are independent of the experimenters (*control experimenters independence*), the experimenter element is changed in a replication [29]. Controlling for experimenters independence also accounts for the site element's effect [29]. Whether the results of the original experiment hold for another population (subjects, experimental objects), the population element is varied — *understand population limits* [25]. To *understand operationalisation limits* in terms of cause and effect, a replication can vary treatment application and transmission procedures or measurement procedures [29]. A replication employs a different protocol to *control protocol independence*, which verifies that the results of the baseline experiment are not determined/influenced by experimental design, objects and instruments, but rather evidence of reality [25], [29].

We formed a family of experiments by executing a series of experimental replications of the original experiment [20] from three perspectives: *control sampling error*, *control experimenters independence* and *understand population limits*. The details of the variations to replications respective to the experimental elements and the specific characteristics of the family of experiments are in Section V-A.

III. Background

This section presents brief introductions of concepts; cognitive bias, confirmation bias and time pressure. Confirmation bias and time pressure are briefed from the SE perspective.

Human judgement under uncertain events usually relies on simplifying heuristics rather than relying on thorough logical processing [32]. These simplifying heuristics are often effective in decision making and offer workable solutions, but may lead to systematic errors (cognitive biases) [32], [33]. Formally, cognitive biases are "*cognitions or mental behaviours that prejudice decision quality in a significant number of decisions for a significant number of people*" [34, p. 3]. Tversky and Kahneman introduced the concept of cognitive biases in the early 1970s [33]. Cognitive biases are also referred to as decision or judgement biases [34]. There are two modes of thinking; system-one (intuitive) and system-two (reflective)—the dual process theory [35], [36]. In system-one, thinking for carrying out actions is effortless and unconscious, e.g., walking or brushing teeth, thus it is fast [2], [35]. Whereas, thinking in system-two is effortful and conscious, thus it is slow. System-two is engaged when we deal with tasks that are complex or our interests are vested, e.g., filling a job application [35]. System-one is more vulnerable to cognitive biases because it is proficient in performing a contextual interpretation of our surroundings (along with the visual system and associative memory in action) that it represses alternative explanations [35]. Due to its fast operation, humans are unaware and incapable of recognising their biases [35].

Within SE, Mohanani et al. classified it in the category of *interest* bias — cognitive biases that are manifested due to an individual's interests [2]. *Positive test bias* or *positive test strategy* terms are also used to refer to confirmation bias [20]. According to Salman et al., confirmation bias occurs in software testing when testers design relatively more specification consistent test cases than specification inconsistent test cases [20]. The software testing literature employs two ways to measure confirmation bias [20]. One of the ways is to use psychological instruments to measure confirmation bias, e.g., by using Wason's Rule Discovery and Selection Task [10]. The other way is to objectively detect it from the test artefacts designed by software testers. For example, assessing the type of a test case based on the input data whether from a valid or invalid equivalence class [37].

Time pressure is also found as deadline pressure, schedule pressure and time budget pressure in the SE literature [38], [39]. The Yerkes-Dodson law is a theory that relates an individual's arousal (increased emotional activity with a present state of mind, as an effect of time pressure) and performance [40]. According to this theory, an inverted U-shaped relation exists between performance and arousal [40]. Performance increases up to a certain point by arousal, after which it starts to decrease [39]. In the software development context, deadlines may increase performance, but it begins to decrease when qualifications for deadlines are excessive [41]. There are multiple reasons that create (negative) time pressure in software projects. For example, budget constraints, bottlenecks due to functional dependencies, inadequate time estimations, change of requirements during sprints (agile development method), dependencies on external suppliers, and technical uncertainties are the usual grounds for time pressure [42], [43], [44]. Another factor that constrains the development schedule is the competition in the global market [43]. Due to this, software companies try to deliver rapidly for less financial budgets [43]. The most scientifically studied and compromised phase of software engineering, due to time pressure, is quality assurance [40].

The following section elaborates on how and in what context we examined the relationship between confirmation bias and time pressure.

TABLE I
Experimental Design of the Original Study [20]. *ES* Stands for Experimental Session, *TP* was the Experimental Group and *NTP* was the Controlled Group

	TP	**NTP**
ES	Group 1	Group 2
Object (Task)	MusicFone	

IV. Original Study

Our original study examined the manifestation of confirmation bias among software testers, and how time pressure promotes confirmation bias during the designing of functional test cases [20]. We aimed to examine the mental/psychological approach for designing test cases, that precede their execution. Therefore, the experimental investigation was independent of the test case execution activity [20]. In other words, the participants did not execute the designed functional test cases, instead only the type of test cases was determined to assess the manifestation of confirmation bias. The original study investigated the following research questions [20]:

RQ1: Do testers exhibit confirmatory behaviour when designing functional test cases?

RQ2: How does time pressure impact the confirmatory behaviour of testers when designing functional test cases?

The independent variable of the study was *time pressure* with two levels: time pressure (*TP*) and no time pressure (*NTP*). In order to operationalise the time pressure construct, 30 min were assigned to the TP group and 60 min were assigned to the NTP group for the designing of functional test cases. Additionally, the TP group was reminded of the remaining time on three instances. These reminders were made to psychologically develop the time pressure. On the contrary, there were no reminders of the remaining time to the NTP group after an initial announcement of the task's duration.

The experimental design of the original study was one factor with two levels between-subjects, which is presented in Table I. As the design shows, the same object (MusicFone) was used for both groups to perform their task.

Forty-two students participated in the study, who were enrolled in the Software Quality and Testing course offered to an international Master's Degree programme in 2015 at the University of Oulu, Oulu, Finland. The participants were trained to design functional test cases before the experimental execution [20].

TABLE II
HYPOTHESES AND RESULTS OF ORIGINAL STUDY

Hypothesis	Null-Hypothesis Result	Effect Size Magnitude
H1: Testers design more consistent test-cases than inconsistent test-cases.	rejected	medium to large
H2: (Dis-)Confirmatory behaviour in software testing differs between the testers under time-pressure and no time-pressure.	failed to reject	medium to large
H3: Testers under time-pressure manifest relatively more confirmation bias than testers under no time-pressure.	failed to reject	small to medium
H4: Testers under time-pressure experience more temporal demand than testers under no time-pressure.	rejected	medium to large

A pre-questionnaire was developed to collect the background information of the participants, i.e., their development and testing experience. NASA-TLX was used as a post-questionnaire to assess subjective workload, and to enhance the conclusion validity of the experiment (briefed in Section V-C) [45]. A realistic object, MusicFone requirements specification was used to design test cases. A conceptual prototype that was a screenshot of the MusicFone application was also provided to the participants, alongside a test case design template. The complete experimental package along with the scripted guidelines to execute the experiment is available online[2].

Table II summarises the results of the original study [20]. Each hypothesis result is mentioned distinctively along with the effect size magnitude observed for the applied treatment, i.e., *time pressure*. The original study provided evidence that testers significantly manifested confirmation bias during the designing of functional test cases ($H1$). The time pressure could not cause the two groups to differ from each other in their confirmation bias manifestation ($H2$). Similarly, time pressure could not promote the manifestation of confirmation bias among the testers ($H3$). However, the magnitude of the effect sizes ($H2$, $H3$) indicates that time pressure might have an impact. The sanity hypothesis ($H4$) showed that the time pressure construct was successfully operationalised because testers in the TP group experienced significantly more temporal demand compared to the NTP group [20].

V. REPLICATIONS

This section elaborates on replications of the original experiment. We use the notation **O-Orig** to refer to the **orig**inal experiment executed in **O**ulu, Finland, from now onwards.

A. Motivation and Changes to O-Orig

We performed three similar **rep**lications of O-Orig, one internal in **O**ulu (**O-Rep**), Finland and two external in **N**ovi **S**ad (**NS-Rep1**, **NS-Rep2**), Serbia.

These experimental replications verify the results of the original experiment for the observed phenomenon of confirmation bias manifestation—*control sampling error*. Additionally, we aim to observe the possible effect of time pressure on the promotion of confirmation bias, as indicated by the effect sizes in Table II of O-Orig. With this set of experiments, including O-Orig, we also aim to provide joint conclusions by aggregating the results.

[2]http://doi.org/10.5281/zenodo.1193955

TABLE III
EXPERIMENTAL ELEMENTS. ''SAME'' INDICATES SAME AS O-ORIG AND ''DIFFERENT'' INDICATES DIFFERENT FROM O-ORIG

	Internal	External	
	O-Rep 2016	NS-Rep1 2017	NS-Rep2 2018
Experimenters	same	different	same as 2017's
Site	same	different	same as 2017's
Experimental Protocol	same	same	same
Construct Operationalisation	same	same	same
Population Properties	same	different	same as 2017's

O-Orig was executed in 2015, the replications were performed afterwards in three successive years: 2016−O-Rep, 2017−NS-Rep1 and 2018−NS-Rep2. Table III compares the five experimental elements (mentioned in Section II) of the three replications with O-Orig. We can see that our set of experiments satisfies the condition to control sampling error because protocol and operationalisation are the same as O-Orig. The *experimenters*, *site* and *population properties* elements are different between Oulu (O-Orig, internal) and Novi Sad's external replications. The external replications were also performed in academic settings with students, but the site was the University of Novi Sad, Novi Sad, Serbia. Moreover, the experimenters in Novi Sad recruited participants who were bachelor's degree students. The joint analysis would help assess the potential effect of the differentiating elements on the results, related to *experimenters*, *site* and *population properties* [23].

The characteristics that qualify our set of experiments as a *family of experiments* are [24]:

- In every experiment an explicit comparison between two treatment levels is made, i.e., *TP* vs *NTP*, and the effect of the treatment is measured on the same response variables.
- We have more than three experiments in our set.
- We have access to the raw data of every experiment.
- We have first-hand knowledge of the experimental setting of every experiment.
- Different sets of participants were recruited for every experiment.

In summary, our family of experiments on confirmation bias and time pressure, which is a result of the replications (from the three perspectives) answers the questions similar to O-Orig:

RQ 1: *Do testers manifest confirmation bias while designing functional test cases?*

RQ 2: *Does time pressure effect the manifestation of confirmation bias while designing functional test cases?*

The answers will leverage to verify the results of O-Orig regarding the manifestation of confirmation bias by software testers. Additionally, to assess the potential effect of varying the site, experimenters and population along with the joint analysis of these factors for our family of experiments.

It is to be noted, the following sections related to experimental elements detail only the necessary information for a coherent reading. In other words, we detail experimental aspects that are necessary from the replications' perspective. The readers are advised to refer to the *original study* [20] regarding the rationales on respective experimental elements.

B. Level of Interaction

We followed the guidelines by Juristo et al. for interacting with the external experimenters for NS-Rep1 (2017) [23]. The experimental package2 was shared with the external experimenters before conducting the *adaptation meeting*. In the two online meetings[3], the O-Orig experimenter briefed the documents in the experimental package to the NS-Rep1 experimenters. We also discussed how to train the (potential) participants before the experimental replication. The training material used for the O-Orig participants was also shared with the external experimenters. The meetings concluded that the replication sample would be drawn from the bachelor's degree students. Additionally, the O-Orig experimental instrumentation and training material, which were in English, were translated into the native (Serbian) language. Other than the language, no changes were made to the O-Orig instrumentation for the external adaptation. For the *querying* step, the O-Orig experimenter was on stand-by (online via Skype) during the NS-Rep1 execution to address any occasional queries. There were no queries from the external experimenters.

The last step, *combination meeting* was held in multiple sessions over distant time-intervals. In the first session, the external experimenters briefed the NS-Rep1 replication's execution to the O-Orig experimenter. It concluded that it was a smooth execution, as per the plan.

The external experimenters ran another similar replication, NS-Rep2, in 2018[4]. The later sessions of the combination meeting were related to data collection (detailed in Section VII-C). In the last session, the results of the O-Orig and replications were compared, which led to decisions regarding joint analysis (detailed in Section VI).

C. Variables and Metrics

The independent variable, *time pressure* is already briefed in the original study's Section IV. There are three dependent variables: number of *consistent* test case (c), number of *inconsistent* test case (ic), and *temporal demand* (TD). A consistent test case refers to a test case that validates the stated required or non-required behaviour of the system as specified in the requirements specification [20]. For example, if requirements specification states,... *the username field does not accept whitespaces.* The test case validating that the username field does not accept whitespaces is a consistent test case. An inconsistent test case validates the behaviour of the system that is not explicitly specified in the requirements, or is outside-of-the-box behaviour but within the domain [20]. For example, if specifications state, *...the phone number field accepts digits...* An inconsistent, outside-of-the-box, test case would exercise the behaviour that the phone number field accepts only '+' from the set of special characters for an international call prefix's purpose. The third dependent variable, temporal demand (TD) is one of the six attributes of NASA-TLX. This attribute assess time pressure as experienced by a subject due to the rate at which a respective activity occurred [46]. In order to measure *TD*, we used the ratings specified by participants on a rating-scale ranging from 0 (low) to 100 (high) on the NASA-TLX sheets.

In the original study, we derived a proxy measure for assessing the manifestation of confirmation bias [20]. It is a rate of change in relative terms, i.e., the difference between the consistent test case coverage and inconsistent test case coverage:

$$z = \frac{c}{C} - \frac{ic}{IC}$$

where c and ic are the number of consistent and inconsistent test cases designed by a participant, respectively. C is the total number of consistent test cases, and IC is the total number of inconsistent test cases. The range of z is $[-1, +1]$. Confirmation bias is manifested when $z > 0$, i.e., a participant has designed relatively more consistent test cases than inconsistent test cases. In order to measure confirmation bias in terms of z, we designed a complete set of test suite comprised of consistent and inconsistent test cases [20]. The suite defines a total number of consistent (*C*) and inconsistent (*IC*) test cases in absolute numbers. This suite serves as a *heuristic-baseline* to enable comparison and perform analysis [20]. The heuristic-baseline comprised $C = 18$ and $IC = 50$ test cases for the O-Orig experiment.

D. Hypothesis Formulation

The performed replications validate the four hypotheses as postulated by O-Orig (Table II). We list them as a set of alternative and the respective null hypotheses.

H1: *Testers design more consistent test cases than inconsistent test cases.*

$$H1_A : \mu(c) > \mu(ic)$$
$$H1_0 : \mu(c) \leq \mu(ic)$$

H2: *(Dis)confirmatory behaviour in software testing differs between testers under time pressure and under no time pressure.*

$$H2_A : \mu([c, ic]_{TP}) \neq \mu([c, ic]_{NTP})$$
$$H2_0 : \mu([c, ic]_{TP}) = \mu([c, ic]_{NTP})$$

H3: *Testers under time pressure manifest relatively more confirmation bias than testers under no time pressure.*

$$H3_A : \mu(z_{TP}) > \mu(z_{NTP})$$
$$H3_0 : \mu(z_{TP}) \leq \mu(z_{NTP})$$

[3]The adaptation discussion also took place via emails.

[4]No interaction was required for the NS-Rep2 because it was an internal replication for the Novi Sad's experimenters.

H4:Testers under time pressure experience more temporal demand than testers under no time pressure.

$$H4_A : \mu(TD_{TP}) > \mu(TD_{NTP})$$
$$H4_0 : \mu(TD_{TP}) \leq \mu(TD_{NTP})$$

$H1$ validates the confirmatory behaviour of testers irrespective of any time pressure. $H2$ validates the effect of time pressure on (dis)confirmatory behaviour in absolute terms, i.e., based on the absolute count of designed (in)consistent test cases. Whereas, ($H3$) detects the manifestation of confirmation bias in relative terms. $H4$ is a posthoc sanity check to assess whether experimenters rightly administered the experimental treatment, i.e., operationalisation of the time pressure construct.

E. Participants

1) Oulu Replication: The recruitment of participants for O-Rep followed the same protocol as O-Orig. We applied convenience sampling to recruit participants from the enrolment of the Software Quality and Testing course offered in 2016 to an international Master's degree programme at the University of Oulu, Oulu, Finland. It was a class announcement to 56 students for the volunteer experiment participation in the introductory lecture of the course. We offered the experimental activity as a regular class activity to all the students, but we considered the data of those who provided written consent for volunteer participation in the experiment. The participants were offered bonus marks as an incentive, which was announced as part of the participation call.

2) Novi Sad Replications: Participants for the replications NS-Rep1 and NS-Rep2 were bachelor students taking the Basics of Software Testing course at the Engineering of Information Systems study program at the University of Novi Sad. We used convenience sampling for recruitment. It was also a class announcement for experiment participation to 59 students in 2017 and 84 in 2018. The call was made in the initial lectures for both implementations of the course and also announced an incentive. Participation in the experiment was not obligatory. The students who voluntarily consented to participate in the experimental activity were rewarded with additional course credits.

It is to be noted that the experimental activity was not itself graded/incentivised. Participation (in the form of consenting the experimenters to use the data) was incentivised across all the experiments in the family.

Similar to O-Orig, we collected participants' background information, using the same O-Orig's pre-questionnaire, for the Oulu and Novi Sad's replications. The background information relates to academic and industrial development and testing experience. The participants marked their experience along the four experience categories: less than 6 months, between 6 months and one year, between 1 and 3 years, and more than 3 years. The experience characteristics of participants are presented in Section VIII-B as a step in analysing the family of experiments.

The replications' participants received theoretical and practical training on performing functional testing before experimental execution. This was in line with the O-Orig training protocol. The training aspects are detailed further in Section VII-A.

VI. Analyses Methods

We perform two types of analyses, individual analyses and meta analysis. The individual analyses are to individually analyse every replication. The meta analysis aggregates the results of our family of experiments. Both are comprised of running statistical significance tests, the preparation of descriptive statistics and visualisations (plots). We set α to 0.05 for the statistical significance testing.

For the individual analyses, we run significance tests of the *t-test* family and *F-test* (Hotelling's T^2) to test our hypotheses. We check for the assumptions of a significance test before running it. In case of a failure to meet the assumptions, we run a non-parametric variant of the respective significance test. For the *t-test* family, we check for the univariate assumption of normality. The two important assumptions for Hotelling's T^2 are: the data from both populations have a common variance-covariance matrix (tested with the Bartlett test) and both populations have a multivariate normal distribution [20], [47]. The other two assumptions for this test are: the subjects from both populations are independently sampled and there are no distinct sub-populations and populations of samples have unique means [20]. We use the Shapiro-Wilk test to check for the univariate and multivariate assumptions of normality. We either report Cohen's d ($0.2 = small$, $0.5 = medium$, $0.8 = large$) or correlation coefficient r ($0.10 = small$, $0.30 = medium$, $0.50 = large$) as an effect size measure with respect to the run (univariate) statistical test and assumptions of the particular effect size measure [48]. Mahalanobis distance ($0.25 = small$, $0.5 = medium$, $> 1 = large$) is reported as an effect size measure for the multivaraite Hotelling's T^2 test. Section VIII-A elaborates on which specific tests of the *t-test* family were executed for the individual analysis.

We perform meta analysis per the guidelines by Santos et al. [28]. We aggregate the results by using aggregated data (AD) and stratified individual participant data (IPD-S) in tandem for the joint analysis [28]. With AD using random-effects meta analysis models, we determine the heterogeneity in our results with I^2 and Q statistics [28], [49]. The IPD-S is applied via multilevel/hierarchical modelling (LMM). We use the Shapiro-Wilk test to determine the normality of the residuals of a model. In order to assess the model fit, we use and report a goodness-of-fit measure—Akaike's Information Criterion (AIC) [50]. The smaller the value of AIC, when compared with the AIC of another model, the better is the model fit [50]. Further details related to the application of AD and IPD-S are in the respective sections of Section VIII.

For performing the individual analysis, we used the replication specific z values derived from the replication specific C and IC. Whereas, for the meta analysis, we used the consolidated (final) z value, which was derived from the C and IC of the last replication, i.e., NS-Rep2. The consolidated value of z ensures the consistency of its measure across all the experiments of this

family. The derivation of specific and consolidated values of z is elaborated in *the heuristic-baseline extension* part of the data collection section (Section VII-C).

The RStudio version used to perform individual analyses and the AD of meta analysis is ver. 1.3.1093. We used IBM SPSS ver. 27.0 to perform multilevel modelling.

VII. Experimental Execution

In this section we present the sequential flow of the execution of our family of experiments. Our family is composed of four experiments. Two were executed at the University of Oulu: the original experiment (O-Orig:2015) and an internal replication (O-Orep:2016). The other two were external replications executed at the University of Novi Sad: NS-Rep1:2017 and NS-Rep2:2018. Two of the authors of this study were responsible for experimental activities at the University of Oulu, while the other two conducted activities at the University of Novi Sad.

Fig. 1 depicts the experimental flow. It visualises pre-experimental, experimental and post-experimental activities. The following sections elaborate on these activities for replications. Details related to O-Orig can be found in Salman et al. [20]. In the figure, every experiment is represented by a separate labelled dashed box. The activities specific to experiments are represented by solid labelled boxes within the experiment boxes.

A. Pre-Experimental Activities

The pilot run was part of the O-Orig experimental protocol [20]. There were consent collection and pre-questionnaire filling early-on from the students who were enrolled in the courses (Section V-E), at the two sites, where the experiment was to be executed. For O-Orig, these pre-experimental activities happened in parallel to training as indicated by a separate top most box of O-Orig in the figure. However, for the replications, they occurred before the training - Fig. 1 depicts it as overlapping boxes of replications sharing the same sequence of pre-experimental activities. Students received the necessary training during the courses per the curriculum. The training consisted of lectures on these topics: functional testing techniques (equivalence partitioning, boundary value analysis), test case design techniques (need and classification of techniques, control flow techniques, functional techniques), orthogonal array testing (Novi Sad), and test case specification. It was possible to keep the content of the training similar despite the two different sites and degree levels because the courses/curriculum covered the basics of software testing. However, different from the O-Orig training protocol, the class assignment on functional test case designing was replaced by a home assignment for the internal and external executions. In the figure, it is depicted by a 'Home assignment' labelled box for the training part within the replication boxes. We used the same practice object, whether for the class or home assignment, across all four experiments. It must be noted, similar to O-Orig, all students were trained/taught these topics irrespective of their experiment-participation consent.

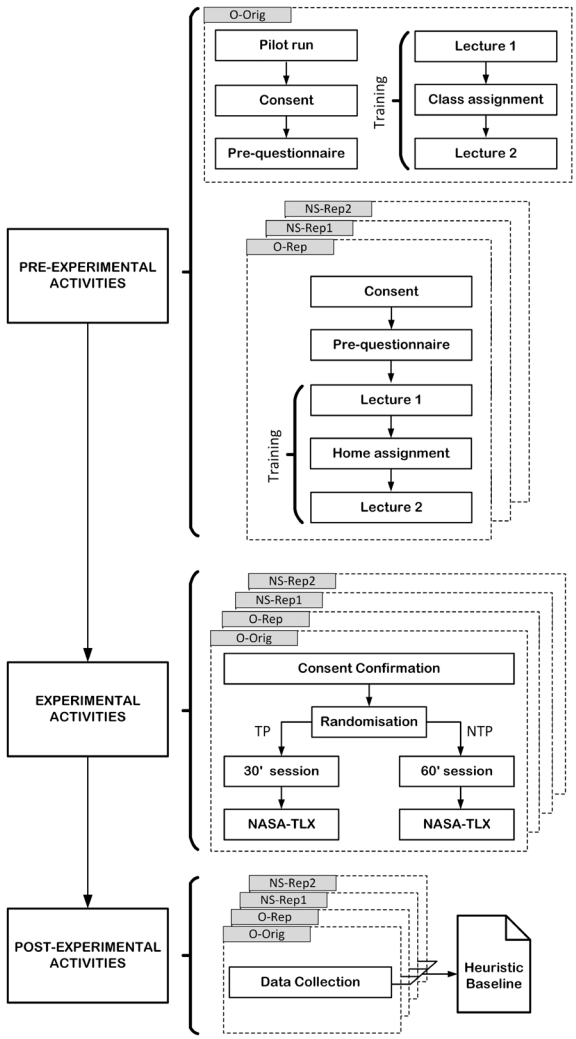
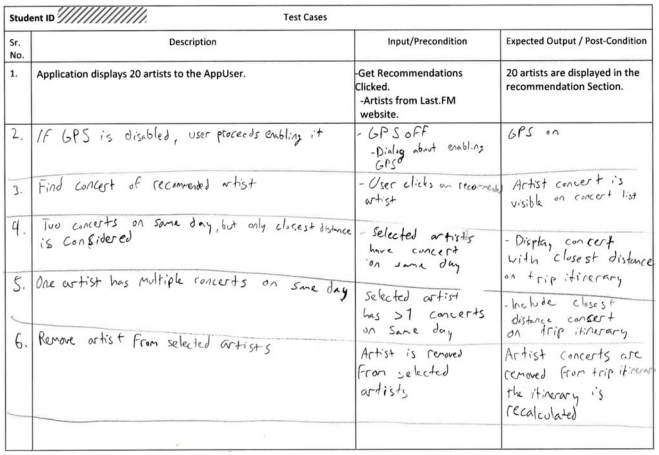


Fig. 1. Experimental process flow.

In total 208 students consented to participate in the experiments: 88 students from Oulu (43 in O-Orig, 45 in O-Rep), and 120 from Novi Sad (53 in NS-Rep1, 67 in NS-Rep2).

B. Experimental Activities

All replications followed the same experimental session's execution protocol as O-Orig. It is indicated by the set of four overlapping boxes of experiments that share the same set of experimental activities in Fig. 1. Experimenters confirmed the consent signing and pre-questionnaire filling before randomly assigning the participants to the control or treatment group (which were in different classrooms). The TP groups were randomly assigned 104 $(22 + 24 + 26 + 32)$ participants and the NTP groups had 104 $(21 + 21 + 27 + 35)$ participants. The TP sessions ran for $30\,min$ and NTP for $60\,min$. The experimenters applied the treatment (time pressure) as explained earlier in Section IV. They executed the experimental sessions (TP, NTP) per the scripted guidelines available in the replication package[2]. The participants in the TP and NTP groups designed test cases on the given template using the MusicFone requirements specification and the conceptual prototype (Section IV).



Student ID ////////// Test Cases

Sr. No.	Description	Input/Precondition	Expected Output / Post-Condition
1.	Application displays 20 artists to the AppUser.	-Get Recommendations Clicked. -Artists from Last.FM website.	20 artists are displayed in the recommendation Section.
2.	If GPS is disabled, user proceeds enabling it	- GPS off -Dialog about enabling GPS	GPS on
3.	Find concert of recommended artist	- User clicks on recommended artist	Artist concert is visible on concert list
4.	Two concerts on same day, but only closest distance is considered	- Selected artists have concert on same day	- Display concert with closest distance on trip itinerary
5.	One artist has multiple concerts on same day	Selected artist has >1 concerts on same day	- Include closest distance concert on trip itinerary
6.	Remove artist from selected artists	Artist is removed from selected artists	Artist concerts are removed from trip itinerary the itinerary is recalculated

Fig. 2. Experimental task: test cases by a participant.

The participants of each group (TP or NTP) performed the activity respective to the time frame of their group. Fig. 2 presents a sample of the experimental task, i.e., test cases designed by a participant. Towards the end of the session, NASA-TLX sheets were handed out to collect post-experimental data.

All students enrolled in the courses performed the actual experimental activity and were exposed to the treatments. However, the randomisation protocol was applied to those who consented to *participate* and to use their data later. In other words, the rest were just sent to either of the classrooms. This was in line with O-Orig.

C. Post-Experimental Activity: Data Collection

The data collection protocol defined for the original experiment [20] was expanded and followed in all replications. Fig. 1 presents it as a single activity labelled 'Data Collection' for all the experiments that also contribute to the heuristic-baseline document (discussed later on) in a similar manner. We extracted data only from the students who consented to be experimental participants.

To extract data, the experimenters at each site read all test cases and labelled them as *consistent*, *inconsistent*, or *dropped* (see Section 4.2 in [20] for the labelling/marking criteria). For the data collection of O-Orig, a substantial agreement of 66% (Randolph's free marginal kappa for inter-rater reliability [51], [52]) existed between the Oulu experimenters [20]. Based on this, the first author continued to mark the test cases for O-Rep. The confusing test cases were resolved to either *c*, *ic* or *dropped* after discussion with the second author to further alleviate subjectivity.

The data collection from external replications (NS-Rep1 and NS-Rep2) was carried out in two phases. The first phase aimed for establishing and enhancing the inter-rater reliability between the data collectors (experimenters) of the two sites. The Oulu experimenters shared the data of randomly chosen 11 participants (set 1) from the O-Orig set. It was a raw data without any markings for test cases. Following this, the Novi Sad experimenters marked the test cases of the set 1 as *c*, *ic* or *dropped* as described in the data collection section of the original study [20]. The two experimenters of Novi Sad first independently marked the data, then, finalized the marking (with a single final label) after discussing their differences. The results of the set 1 were shared with the Oulu experimenters for calculating the inter-rater reliability. The Randolph's free marginal kappa was 46%. It was a low value that called for a session between the Oulu and Novi Sad experimenters to discuss reasons for the low value. After a discussion session, it was decided to share another set of data from the set of O-Orig, i.e. set 2. The set 2 consisted the data of the randomly chosen 10 participants excluding those that appeared in set 1. The Novi Sad experimenters marked the set 2, and the inter-rater reliability improved to 78%—a substantial agreement— between the experimenters of the two sites. This reliability value marked a go-ahead for collecting the data from NS-Rep1 and NS-Rep2.

TABLE IV
DATA COLLECTED: THE NUMBER OF PARTICIPANTS IN EACH GROUP (N), #TC IS THE SUM OF THE COUNT OF CONSISTENT (#C), INCONSISTENT (#IC) AND DROPPED (#DROPPED) TEST CASES

Replication	Group	N	#TC	#c	#ic	#dropped
O-Rep	NTP	21	**217**	160	50	7
	TP	24	**192**	155	31	6
NS-Rep1	NTP	27	**305**	221	39	45
	TP	26	**231**	164	32	35
NS-Rep2	NTP	35	**535**	345	103	87
	TP	31	**337**	280	16	41

In the second phase, the Novi Sad's experimenters collected the data from NS-Rep1 and NS-Rep 2 respectively, independent of each other. They resolved the conflicted marking and confusing test cases with each other. No inter-rater agreement with the Oulu experimenters was calculated before collecting data from NS-Rep2 because experimenters/data collectors were the same as for NS-Rep1. The data collection from NS-Rep2 resulted in exclusion of one participant from the TP group because all of their test cases (10) were discarded due to completely illegible hand writing. It resulted in 66 participants for NS-Rep2. Table IV reports the collected data, i.e., the counts of test cases per replication and per group (NTP vs. TP). In O-Orig, we also excluded a participant because all of their test cases were dropped (not discarded). To *drop* a test case, it must meet the technical criteria defined in Salman et al. [20]. Hence, in total our family of experiments have a data of 206 participants.

The heuristic-baseline extension: It is to be noted that the heuristic-baseline used in the O-Orig was enhanced by adding the test cases designed by the O-Orig participants that were not present in the baseline [20]. This step was taken to improve the validity of the measures. Hence, the counts of $C = 18$ and $IC = 50$ were attained (Section V-C). Similar to the O-Orig, we extended the heuristic-baseline with every replication. Thereby, enhancing the suite towards its *completeness*.

After the data collection from O-Rep, 12 new inconsistent test cases were added to the baseline, which raised the count of IC to 62 inconsistent test cases. Whereas, no new consistent test case were found from the O-Rep data.

The heuristic-baseline's extension from the Novi Sad replications followed multiple steps for the identification of new

TABLE V
INDIVIDUAL ANALYSES RESULTS. * STATISTICALLY SIGNIFICANT

Experiment	Hypothesis	Results	Effect Size (ES)
O-Rep $N = 45$	$H1_0$	pooled: p-value = 1.682e-08*, df = 44	r = -0.595
		TP: p-value = 1.531e-05*, df = 23	r = -0.624
		NTP: p-value = 1.133e-4*, df= 20	r = -0.595
	$H2_0$	T^2 = 4.543, F(2,42) = 2.218, p-value = 1.213e-1, η^2 = 0.095	Mahal. D = 0.636
	$H3_0$	p-value = 8.157e-1, df = 43	d = -0.277
	$H4_0$	p-value = 1.706e-4*, df = 43	r = -0.535
NS-Rep1 $N = 53$	$H1_0$	pooled: p-value = 1.199e-9*, df = 52	r=-0.591
		TP: p-value = 9.707e-9*, df = 25	r=-0.613
		NTP: p-value = 1.134e-05*, df = 26	r=-0.597
	$H2_0$	T^2 = 5.239, F(2,50) = 2.568, p-value = 8.674e-2, η^2 = 0.093	Mahal. D = 0.628
	$H3_0$	p-value = 9.675e-1, df = 51	d=-0.503
	$H4_0$	p-value = 8.938e-3*, df = 51	r=-0.325
NS-Rep2 $N = 66$	$H1_0$	pooled: p-value = 7.138e-12*, df = 65	r=-0.596
		TP: p-value = 9.198e-07*, df = 30	r=-0.623
		NTP: p-value = 9.81e-07*, df = 34	r=-0.585
	$H2_0$	T^2 = 42.556, F(2, 63) = 20.945, p-value = 1.062e-07*, η^2 = 0.090	Mahal. D = 1.608
	$H3_0$	p-value = 6.248e-1, df = 64	d=-0.079
	$H4_0$	p-value = 8.109e-3*, df = 64	r=-0.295

test cases with an objective to enhance reliability among the experimenters. With the Novi Sad replications, we went through a thorough process only for the identification of new inconsistent test cases because: (i) the definition of an inconsistent test case leverages a maximum coverage for testing any implicit and out-of-the-box behaviour, thus, it is impossible to determine the absolute number of inconsistent test cases [20], and (ii) the previous experiments could not find any new consistent test cases.

The first step towards the baseline extension was a pilot run for the Novi Sad experimenters to identify unique inconsistent test cases from the data. They randomly chose 15 inconsistent test cases from the NS-Rep1 dataset. They shared the same set with the Oulu experimenters[5]. Both sites independently identified unique test cases from the set. This step was followed by a meeting between the two sites to discuss their results and resolve discrepancies. After developing a shared understanding, as a next step, the Novi Sad experimenters identified new inconsistent test cases from the NS-Rep1 dataset with comparison to the baseline. This resulted in the addition of 3 new inconsistent test cases to the baseline suite after a discussion between the Oulu[5] and Novi Sad experimenters. The Novi Sad experimenters, then, continued with the identification of new inconsistent test cases from the NS-Rep2 dataset with comparison to the baseline suite. They, then, shared the new test cases set with the Oulu experimenters[5]. An afterwards discussion between the two sites lead to the addition of 11 new inconsistent test cases to the baseline suite from the NS-Rep2 dataset. Finally, the total number of inconsistent test cases (*IC*) in the heuristic-baseline suite after four experiments is 76.

For the identification of potential new consistent test cases, the Novi Sad experimenters checked the completeness of the existing consistent test cases in the baseline suite. They did this by comparing them with the MusicFone requirements specification. The Novi Sad experimenters could not find any new consistent test cases for adding to the suite. In other words, the consistent test cases in the suite gave *complete coverage*[6] to the requirements specification. Furthermore, the steps for the identification of new consistent and inconsistent test cases by the Novi Sad experimenters indirectly corroborated the baseline suite, i.e., every test case in the suite validates a unique behaviour of MusicFone (the object).

The raw data is available online[7].

VIII. RESULTS

This section reports the results of individual replications for the stated hypotheses (Section V-D), followed by the meta-analysis of our family of experiments per the guidelines by Santos et al. [28].

A. Individual Analysis

The results of individual analysis for O-Orig are already given in Table II. The analyses procedures of replications are in consistent with the analyses procedures of O-Orig because all replications implemented the same experimental design.

In order to test $H1_0$, we first test it for the pooled data of TP and NTP groups, for every replication. Secondly, again for every replication, we test it separately for the TP and NTP groups. The c data in all replications, for pooled and separate groups, was normally distributed. Whereas, the ic data for all the three cases of every replication failed to meet the assumption of normality. Therefore, we executed the Wilcoxon signed-rank test for $H1_0$ for every replication. We also applied Bonferroni adjustment for $H1_0$, $\alpha = 0.016$ to the pooled and separate TP and NTP testing. The results ($p - val$ and ES) for $H1_0$ significance testing can be see in Table V. $H1_0$ is rejected with a *large* effect size for every case in every replication. To test $H2_0$, we first tested for its assumptions. In order to

[5]The Novi Sad experimenters shared the data with Oulu experimenters after translating it into English language because it was in Serbian (Section V-B).

[6]Per the definition of a consistent test case, it is possible to determine and design consistent test cases with complete coverage with respect to the requirements specification [20].

[7]https://doi.org/10.5281/zenodo.8330331

meet the multivariate normality assumption for the TP and NTP groups, we normalised the ic data of every replication. The c data of every replication was normally distributed. The Bartlett test revealed that the data from both populations, for every replication, have a common variance-covariance matrix. The last two assumptions for Hotelling's T^2 also hold for all the replications. After executing the test, we failed to reject $H2_0$ for O-Rep and NS-Rep1. However, we rejected the null hypothesis for NS-Rep2 with a large ES (Mahal.D) and variance $\eta^2 = 0.090$ (Table V). To test $H3_0$, we first tested the z data of the TP and NTP groups for the assumption of normality. The z data met the assumption of normality for both groups for every replication. We, thus, executed two sample *t-test* for every replication for $H3_0$. We failed to reject $H3_0$ for every replication as can be seen in Table V. For testing $H4_0$, we assessed the normality of the TD—temporal demand—data of the TP and NTP groups for every replication. The TD data of the TP groups was not normally distributed; this result held for every replication. Hence, we run the Mann-Whitney U test to test $H4_0$. We rejected $H4_0$ for every replication with either *medium to large* ES or *large* ES (Table V). The results of the assumptions testing, hypothesis wise and replication wise, are available online[8] in Appendix A.

The results of replications in Table V show that they are in consistent with the results of the O-Orig. The participants exhibited confirmation bias by designing significantly more consistent test cases compared to inconsistent test cases ($H1_0$). Time pressure could not be observed to increase confirmation bias among participants in relative terms ($H3_0$). Similarly, time pressure could not cause a difference in confirmation bias in absolute terms ($H2_0$) for O-Rep and NS-Rep1. However, $H2_0$ of NS-Rep2 is rejected with $p - val = 1.062e - 07$ and ES is large. It indicates that time pressure affected the manifestation of confirmation bias between the groups, in absolute terms. The results also indicate a possible practical effect of time pressure in terms of medium-to-large ($d = -0.504$) ES for $H3_0$ of NS-Rep1. It must be noted that the individual analyses for $H3$ were carried out with respect to the replication's specific z value (explained in Section VII-C) calculated with $IC = 62$ for O-Rep, $IC = 65$ for NS-Rep1 and $IC = 76$ for NS-Rep2.

The results for $H4_0$ indicate that all three replications successfully operationalized the time pressure construct, i.e., participants in the *TP* groups experienced significantly more time pressure compared to participants in the *NTP* groups.

We can see that the results of H1 are consistent among all experiments, including the O-Orig. There are no variations either in the statistical significance testing or practical significance (effect size) terms. In other words, we verified and validated the results of O-Orig from the perspective of the manifestation of confirmation bias by software testers. For H2, we observe variation in the statistical and practical significance terms due to NS-Rep2. While for H3, there is no variation in the statistical significance terms, but for practical significance terms due to NS-Rep1 and O-Orig. We, therefore, perform meta analysis of our family of experiments to provide joint results from the perspective of time pressure's effect on confirmation bias.

[8]https://doi.org/10.5281/zenodo.7599720

TABLE VI
DESCRIPTIVE STATISTICS FOR PARTICIPANT CHARACTERISTICS: ACADEMIC DEVELOPMENT EXPERIENCE (ADE), ACADEMIC TESTING EXPERIENCE (ATE), INDUSTRIAL DEVELOPMENT EXPERIENCE (IDE) AND INDUSTRIAL TESTING EXPERIENCE (ITE); MEDIAN = M, RANGE = R, IQR = I, MODE = MO

Experiment	ADE				ATE				IDE				ITE			
	m	r	i	mo	m	r	i	mo	m	r	i	mo	m	r	i	mo
O-Orig	3	3	2	4	1	2	0	1	1	3	2	1	1	3	0	1
O-Rep	3	3	1	3	1	2	0	1	1	3	1	1	1	3	0	1
NS-Rep1	3	2	2	3	1	2	0	1	1	3	0	1	1	1	0	1
NS-Rep2	3	3	1	3	1	2	0	1	1	2	0	1	1	1	0	1

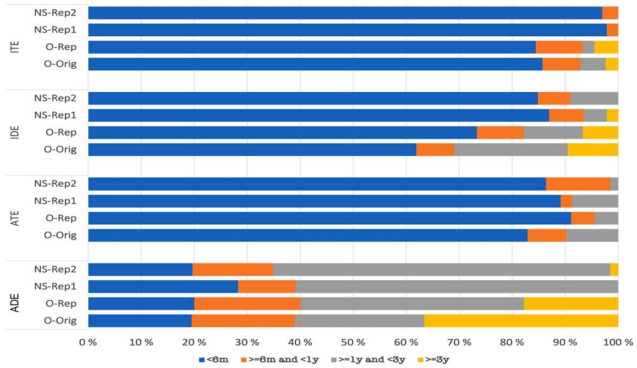


Fig. 3. Stacked bar chart for participants' experience.

B. Meta Analysis Step 1: Participants Description

The first step to performing meta-analysis is to describe the participants to understand the characteristics of the sample population, the population to which the results have to be generalised for and observe the patterns of characteristic variability among the family of experiments. Table VI provides summary statistics for the experience of the participants of all experiments. These statistics are calculated by assigning 1, 2, 3 and 4 ranks respectively to the experience categories (Section V-E). The results show that the participants of Oulu's experiments are more experienced than the participants of Novi Sad's experiments.

Fig. 3 shows the stacked bar chart for the participants' experience across the family of experiments. It further informs that the participants of O-Orig are more experienced than the participants in other experiments except for ITE in O-Rep. The Oulu participants have more industrial development (IDE) and testing (ITE) experience compared to Novi Sad's participants. The lower experience for Novi Sad pertains to the fact that Oulu's executions employed Master's students, whereas, Bachelor's students were recruited for NoviSad's replications. It is also visible that every experiment has participants in $> 1y$ *and* $> 3y$ category who are experienced in development and/or testing, except for ITE for Novi Sad's replications.

We further characterise our participants' experience in terms of the R^3 (Real, Relevant, Recent) scheme proposed by Falessi et al. in the context of our experimental objectives [53].

TABLE VII
DESCRIPTIVE STATISTICS FOR z: NTP VS TP, N IS NUMBER OF PARTICIPANTS AND SD IS STANDARD DEVIATION

Experiment	Group	N	Mean	SD	Median
O-Orig	NTP	21	0.436	0.159	0.431
	TP	21	0.381	0.125	0.388
O-Rep	NTP	21	0.391	0.192	0.375
	TP	24	0.341	0.142	0.357
NS-Rep1	NTP	27	0.435	0.189	0.434
	TP	26	0.348	0.139	0.349
NS-Rep2	NTP	35	0.526	0.200	0.571
	TP	31	0.511	0.186	0.500

In our case, *real* refers to the industrial and academic experience (testing and/or development), which we can map onto our already defined experience categories. Considering the *relevance* aspect, the academic and industrial development experience are not relevant to the objectives of our study, but only the testing experience. In our case, more than 80% of the participants have $< 6m$ of relevant experience. *Recent* refers to the recency of the relevant experience. The training given to the participants has not only made their knowledge relevant but also recent, which makes them suitable for the experimental objectives. Participants with more recent experience tend to perform better than participants with higher but lapsed experience [53]. With reference to the contextual (of the experimental objectives) characterisation and Salman et al., we, therefore refer to our participants as the proxies for novice professionals [54].

C. Meta Analysis Step 2: Individual Replications

In this step we analyse individual replications. We provide summary statistics and visualisations of our response variables c, ic and the proxy confirmation bias measure, z—rate of change. It is to be noted, we re-calculated z with the latest (NS-Rep2) values of *C* and *IC* for all the previous experiments, i.e., O-Orig, O-Rep and NS-Rep1. The heuristic baseline was enhanced with every replication as explained in Section VII-C.

Table VII shows the descriptive statistics of the response variable z for the control and treatment groups for this family. The descriptive statistics for c and ic are present online[8] in Appendix B.1.

It can be seen in Table VII that mean and median values of z for the NTP groups across all experiments are higher than the TP groups. This suggests a relatively higher manifestation of confirmation bias in NTP groups. The descriptive statistics of c and ic show that participants in the NTP groups designed a higher number of consistent and inconsistent test cases compared to TP groups. In the online[8] Appendix B.1, Tables 5 and 6 show that the mean and median of NS-Rep2 for c is higher than the other experiments. In case of ic for NS-Rep2, the NTP group's mean is higher compared to the mean of ic in other experiments. However, the median of NS-Rep2 for ic is not the highest value compared to the other experiments. The mean of the TP group for ic of NS-Rep2 is smallest compared to the mean of ics of the TP groups of the other experiments in the family. However, the median value (0) for ic for TP of NS-Rep2 is the same as for O-Orig and NS-Rep1.

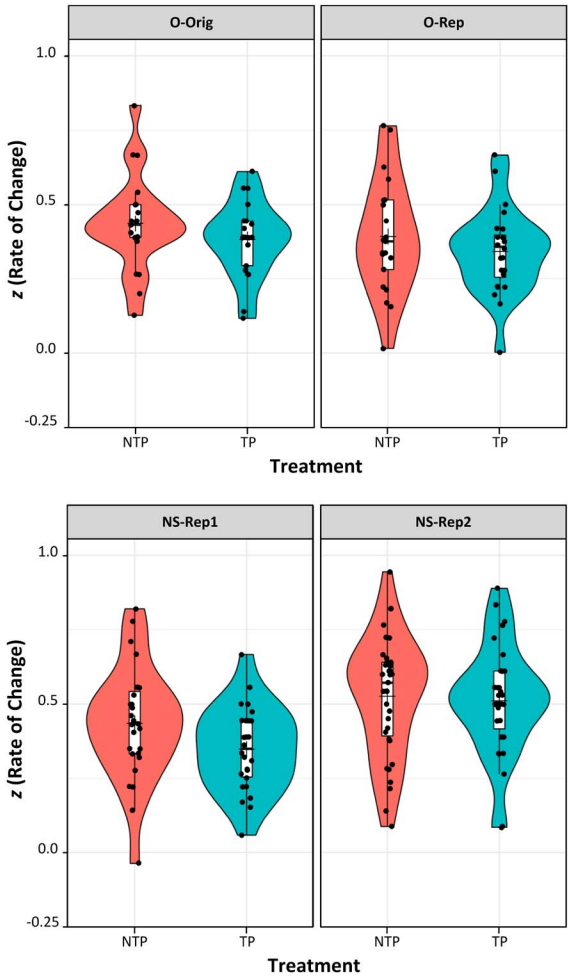


Fig. 4. Box plot and Violin plot for z.

Fig. 4 presents the box and violin plots for z across all experiments. It can be seen that most of the data for NTP groups is scattered around the median except for the NTP of NS-Rep2, for which it is around the third quantile. The z values for the TP groups (vs NTP groups) have overall a lesser range of distribution. It is also observable that both groups in this family manifested confirmation bias because all the z values are greater than 0. Only one participant from NS-Rep1 (NTP group) did not manifest confirmation bias because z is lesser than 0.

A profile plot for z for this family of experiments is in Fig. 5. It complements the observations based on summary statistics and box-and-violin plots. A similar relation can be seen between z and the treatment groups for the Oulu (O-Orig, O-Rep) experiments because of the same slopes. However, the relationship varies across sites (i.e., different slopes) and also between NoviSad's (NS-Rep1, NS-Rep2) replications. Therefore, the plot suggests a possible heterogeneity across the sites and/or experiments. The visualisations for c and ic are in Appendix B.1[8].

This step of meta analysis also requires performing statistical testing following the same analysis procedures. This is to ensure

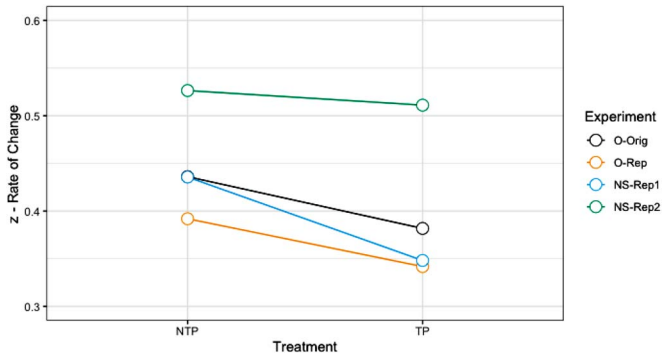


Fig. 5. Profile plot for z: NTP vs TP.

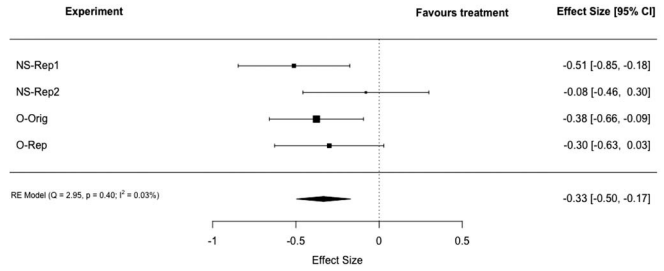


Fig. 6. Forest plot: Effect of TP in terms of z.

that differences in the results of the family of experiments do not pertain to inconsistent statistical methods [28]. Section VIII-A already reports the individual analyses that were carried out with consistent statistical analyses procedures.

D. Meta Analysis Steps 3 and 4: Joint Analysis

In this step, we aggregate the results of our family of experiments by applying AD and IPD-S to provide a joint analysis.

For AD, in terms of z metric, we first calculate effect size for all the experiments. We computed Cohen's d because the data for z was normally distributed for the TP and NTP groups in all experiments. Additionally, we computed the variance for every experiment. Then, the effect sizes and variances were pooled together via a random-effects meta-analysis model.

Fig. 6 presents the forest plot of the meta-analysis for z. We can see that for none of the experiments, time pressure could significantly promote confirmation bias because all the estimates are on the left side of 0. The random-effects meta-analysis shows low heterogeneity ($I^2 = 0.03\%$, $Q = 2.95$, $p = 0.40$) among the results of the experiments [49]. We can also see in Fig. 6 that the upper end of the confidence interval for NS-Rep2 shows that it may have favoured the treatment. The summary effect: $M = -0.33$ (*small to medium*), and 95% CI $[-0.50, -0.17]$ (black diamond) shows no effect of time pressure on confirmation bias when assessed as the relative rate-of-change.

The next step for joint analysis is the application of the IPD-S via multilevel/hierarchical modelling (LMM) [28]. Multilevel modelling enables a statistical analysis when data consists of/originates from a hierarchical structure [50], [55]. Our

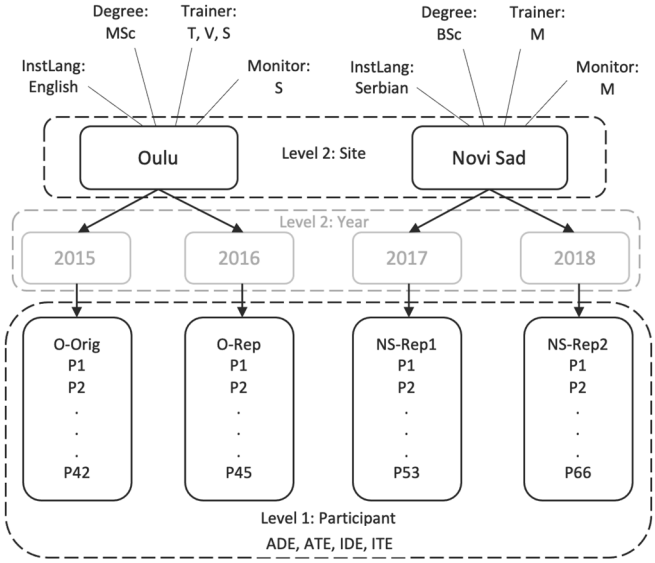


Fig. 7. Multilevel hierarchical structure of our family of experiments.

family of experiments presents a three-level hierarchy; participants belong to four distinct experiments conducted in successive years at two different sites: Oulu and Novi Sad (Fig. 7). It is essentially a three-level hierarchy but presents two levels: *site* and *participant*. Level 2, *year*, is deprecated as explained later.

The response variable, z, is measured at the lowest (participant) level; it tends to be more homogeneous for its own unit compared to the participants from different years and sites. In other words, due to the hierarchical structure, z may have an effect of contextual variables (e.g., language of instrumentation) or their interactions occurring either at the same or other levels of hierarchy [50], [55]. Additionally, the results may also be influenced by participant level variables or their interaction with the treatment, e.g., experience of the participants. The LMMs leverage exploratory analysis for the effect on the response variable due to the contextual and participant level variables [28], [50], [55]. The exploratory analyses using the LMMs is referred to as **Step 4** by Santos et al. [28].

Fig. 7 depicts the identified contextual variables. The identification is based on the differences indicated for replication elements (Table III) between the internal and external replications. On the site level, the differences occurred for experimenters: *Trainer* and *Monitor*, population properties: *Degree*. The instrumentation language, *InstLang*, also differed between the two sites (Section V-B), though the experimental protocol and construct operationalisation were the same. We deprecate the middle level—*Level 2: Year*—in Fig. 7 because no contextual characteristic was exclusive for this level. Hence, the final hierarchy has only two levels. The participant level variables are the experience characteristics (Section VIII-B).

For building multilevel models, we follow the approach by Field [50], [56]. It is to start from a simple model to more complex models. Table VIII presents the results of the analysis of multilevel models, their assessment of normality, the fit (AIC) and significance. Model 1 is the basic model with treatment as a fixed factor, and it does not account for any hierarchical

TABLE VIII
RESULTS OF MULTILEVEL MODELS; THE BEST FIT MODEL IS IN **BOLD**; F = FIXED EFFECTS, R = RANDOM EFFECTS; THE NOTATIONS TO REPRESENT THE MODELS ARE BASED ON IBM SPSS

	Model	AIC	Normality	Comments
	Step 3			
1	F= Treatment	-115.626	Yes	$p - value = 3.1e - 2$
2	**F= Treatment; R= INTERCEPT | Site**	**-117.405**	**Yes**	
	Step 4			
3	F= Treatment ADE ADE*Treatment; R= INTERCEPT | Site	-105.735	Yes	
4	F=Treatment ATE ATE*Treatment; R= INTERCEPT | Site	-111.435	Yes	
5	F=Treatment IDE IDE*Treatment; R= INTERCEPT | Site	-102.099	Yes	
6	F=Treatment ITE ITE*Treatment; R= INTERCEPT | Site	-103.939	Yes	
7	F=Treatment Trainer; R= INTERCEPT | Site	-120.428	Yes	Convergence Issues
8	F=Treatment Degree; R= INTERCEPT | Site	-120.428	Yes	Convergence Issues
11[9]	F=Treatment; R= INTERCEPT Treatment | Site	-116.556	Yes	
12	F=Treatment Trainer; R= INTERCEPT Treatment | Site	-120.428	Yes	Convergence Issues
16	F=Treatment Treatment*Trainer; R= INTERCEPT Treatment | Site	-116.194	Yes	Convergence Issues

structure of the data. Model 1 has a significant effect of treatment (time pressure) on z with $p - val = 3.1e - 2$. The next model, Model 2, expands Model 1 by varying the intercept, i.e., it accounts for the two level hierarchical structure that the data is coming from two different sites. We can see that the fit of the model improved because the AIC decreased to -117.405. This indicates that the model must account for the site level differences between two different sites.

The models under Step 4 in Table VIII are built for the exploratory analyses. These models analyse the effects of contextual and participant level variables—moderators [28]. The models from 3 to 6 expand the previous best model (Model 2) using forward selection by adding the participant level (Level 1) variables as fixed effects to the model. None of these (expanded) models could improve the fit of the model. This indicates that the model should not account for the participant level variables. We now expand the best model (Model 2) using forward selection with the site level (Level 2) contextual variables as fixed effects. Table VIII shows only two models 7 and 8 that were respectively built with *Trainer* and *Degree* predictors. Models 9 and 10 are in the online[8] Appendix B.2. These models show an (apparent) improvement in the fit, but the results are not reliable due to convergence issues.

The next step in multilevel modelling is to vary the slope, i.e., to let the effect of treatment to vary across sites. Thus, we introduce random slope to our last best Model 2. The AIC of Model 11 is still higher than Model 2. This suggests, the effect of treatment may not be varying across the sites. Adding predictors (Level 2 variables) per forward selection as fixed effect to the random intercept and random slope models (Model 12 and the rest in Appendix B.2[8]) did not improve the model fit. This also introduced convergence issues, making the results unreliable. Introducing possible cross-level interactions to the best model, which is another step in multilevel modelling, could not improve the fit and introduced convergence issues—Model 16.

[9]We built the models with five covariance matrices (CV): scaled identity, diagonal, compound symmetry, unstructured and AR(1). The fit of the models with the scaled identity was the best. Thus the table reports the AIC of the scaled identity CV model.

TABLE IX
TYPE III TESTS OF FIXED EFFECTS

Source	Numerator df	Denominator df	F	Sig.
Intercept	1	2.940	241.675	.001
Treatment	1	204.121	4.467	.036

Convergence (non convergence and hessian matrix is not positive definite) issues indicate that our models are complex and our data is not enough to support the defined model. Hence, the analysis results of our final (best fit) model—Model 2 is in Table IX.

Our final model—Model 2—consists of Treatment as a fixed factor and Site as a random factor. It does not include any participant level variables, site level variables, and same or cross level interactions. The effect of treatment (time pressure) is significant ($p = 0.036$) on confirmation bias measured as a relative rate-of-change (z). The participants experience's characteristics neither suggest to impact confirmation bias nor have interaction with the treatment because adding those could not improve the model fit. We do not have enough data neither to observe the effect of the contextual level (of the site) variables on confirmation bias nor their interaction with the treatment.

We now, complement the contextual level exploratory analysis performed via IPD-S (multilevel modelling) with AD [28]. We perform AD sub-group meta analysis to assess the effect of the sites on the results. Fig. 8 shows no heterogeneity ($I^2 = 0.0\%$, $Q = 0.12$, $p = 0.73$) for Oulu's experiments. Whereas, medium to large heterogeneity ($I^2 = 64.4\%$, $Q = 2.81$, $p = 0.09$) for Novi Sad, and not statistically significant because the 95% CI $[-0.73, 0.12]$, includes 0. This affirms the results of Model 3 that there are differences on the site level, and they are due to Novi Sad's experiments. Another sub-group meta analysis was performed that compared the instrument language difference (*InstLang*) between the sites (see Fig. 3 in Appendix B.2[8]).The summary effects for English and Serbian groups were the same as for Oulu and Novi Sad, respectively, in Fig. 6. This indicates that there is a difference on the *Site* level, however, the results of multilevel modelling suggests that our data is not enough.

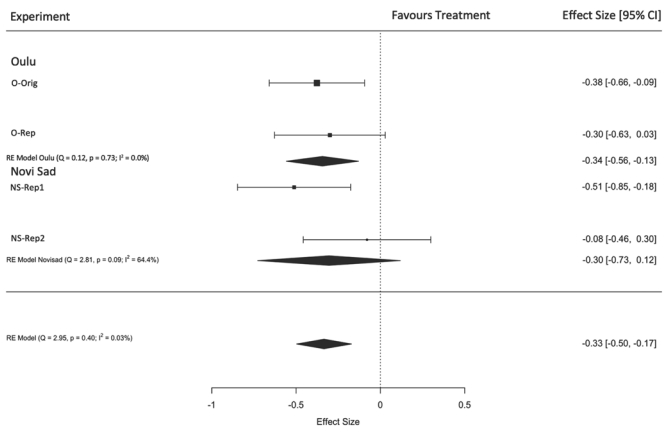


Fig. 8. Forest plot: Oulu vs. Novi Sad.

We performed meta analysis for the univariate case, i.e., for the single response variable z. Our data could not support the complex models, and hence, the effect of contextual variables is inconclusive. We, therefore, refrain from performing meta analysis for multivariate response variables (c, ic); this analysis suggests more complex models with insufficient data.

IX. Discussion

In this section, we discuss our findings related to the manifestation of confirmation bias and the effect of time pressure in this regard. We also compare our results with the findings of relevant studies, and discuss implications of our results.

A. The Manifestation of Confirmation Bias

The replications affirm that junior and novice testers—student participants—exhibit confirmation bias irrespective of time pressure. The confirmatory behaviour is observed not only in absolute terms but also for the relative rate-of-change measure. It is a general behaviour of testers to first exhibit confirmatory behaviour before exhibiting disconfirmatory behaviour [12]. The designing of consistent test cases helps them identify inconsistent test cases [12]. The manifestation of confirmation bias is also irrespective of the relative experience of our participants. Per Fig. 3, O-Orig had the most experienced participants on all scales, yet they significantly manifested confirmation bias. In literature, experience is accounted for promoting disconfirmatory behaviour or decreasing the level of confirmation bias among software testers [4], [8], [9], [12], [37]. Our observation for *experience* is attributed to inexperienced participants of our family of experiments because they are proxies for novice professionals [54].

In our family of experiments, we used a realistic experimental object—MusicFone. The limited experience of our participants could have hindered them in designing inconsistent test cases for the realistic and complex object. It is evident from the data of our experiments that they also could not fully design consistent test cases (per the heuristic-baseline) despite the provision of a conceptual prototype (Section IV). With such a combination of experience and realistic object, it could be impossible not to manifest confirmation bias whether assessed in absolute or the relative rate-of-change terms. Unfamiliarity with the domain may also apprehend the manifestation of disconfirmatory behaviour [12], [16]. Salman et al. refer to the familiarity with domain (*project experience*) as one of the sub-categories of experience in relation to software testing and confirmation bias [12]. Therefore, this factor could also have promoted the manifestation of confirmatory behaviour among our participants.

It may also be impossible for an experienced tester not to manifest confirmation bias in terms of the relative rate-of-change in certain cases. For example, if requirements specification are either ambiguous, incomplete or minimal [12], then, the number of inconsistent test cases may considerably increase relative to consistent test cases. Hence, giving complete coverage to inconsistent test cases could conceptually and practically be impossible. When assessed in terms of the relative rate-of-change, against a heuristic-baseline, the low coverage of inconsistent test cases would always result in favour of the manifestation of confirmation bias. The number of inconsistent test cases would continue to increase whenever a tester or test suite reviewer would evaluate the test suite/baseline designed by consulting ambiguous, incomplete or minimal specifications. The practical relevance of this scenario is supported by our phase of data collection; every replication increased the count of inconsistent test cases in the heuristic-baseline. Yet, neither the requirements specification was ambiguous or minimal [20] nor our participants were experienced testers. If confirmation bias is assessed only in absolute terms, it is certainly possible not to detect confirmation bias. However, it is a compromise on the coverage aspect with respect to the requirements specification.

It is important to note that the completeness of requirements specification is a critical aspect. A complete set of requirements explicitly elicits all the required and non-required behaviours of the application. When a tester designs test cases referring to such specifications, then, it is the manifestation of confirmation bias with a full coverage. All the test cases are consistent test cases because they are designed to validate/confirm the stated behaviours of the application. This is a scenario when confirmation bias is not detrimental for software testing because all the required and non-required behaviours are validated with (a theoretically) complete coverage.

B. The Effect of Time Pressure on Confirmation Bias

In our original experiment (O-Orig), we could not observe time pressure to significantly promote confirmation bias neither in absolute terms nor in the relative rate-of-change (z metric) terms. However, the results of individual replications (Table V) report mixed results for the effect when observed in absolute terms. The effect of time pressure on confirmation bias is statistically significant for NS-Rep2 with a large effect size. However, in terms of the relative rate-of-change, the results of individual analyses are in line with the results of O-Orig from the statistical significance and magnitude of effect size perspectives. A possible reason could be a small sample size in every experiment. We acknowledged and referred to this limitation, for O-Orig, in the context of a possible Type-II error in Salman et al. [20]. The meta analysis via multilevel modelling leveraged

to address this limitation. The first model (Model 1, Table VIII) did not account for any hierarchical structure. In other words, it is similar to an analysis performed for when the data is from a single experiment. Thus, 206 (42+45+53+66) data points from a *technically single* experiment that formed a relatively large sample size compared to each individual experiment in our family. We, then, observed time pressure to significantly effect confirmation bias in terms of the relative rate-of-change.

Experience also plays a role in determining the (dis)confirmatory behaviour of a software tester under time pressure [12]. Experienced (around $6yr$ of industrial experience) software testers may only manifest disconfirmatory behaviour under time pressure, i.e., they may choose not to execute consistent test cases [12]. Additionally, they may also manifest both behaviours but with limited coverage amid time pressure [12]. If our participants were experienced, we could have observed diverse results. Nonetheless, with the factors like experienced participants and familiarity with the domain, suitable time duration must be chosen for the participants to experience time pressure, in turn to observe its effect on confirmation bias.

The site level difference: It is clear from the results of AD (sub-group meta analysis) and IPD-S that the two different experimental sites have an effect on the final results. The difference and effect is pertained to NS-Rep2. It is also supported by the statistically significant results of NS-Rep2. The exploratory analysis (via multilevel modelling) could reveal the site/contextual level variables that caused differences. However, the results yielding the potential differentiating variables and their interactions with the treatment were hampered by the insufficient amount of data for the complex models. It must be noted that the AD and IPD-S analyses were carried out for a univariate case, i.e., z. Yet, the statistically significant result of NS-Rep2 is yielded from a multivariate analysis (Section VIII-A). We could possibly observe the effects of differentiating contextual variables by running multivariate analysis assessing the effects on confirmation bias in absolute terms (c, ic). This may not, however, eliminate the problem of insufficient data for complex multilevel models.

The contextual level variable—*monitor*—could be the result-differentiating factor between the two sites, especially between NS-Rep1 and NS-Rep2. The psychological building of time pressure (Section IV) could be influenced by the monitor despite following the scripted guidelines and the same person acting as a monitor during experimental sessions. In other words, how the reminders are made could still be influenced by the monitor's tone or body language. Apart from the potential influence of *monitor*, it could be the NS-Rep2 participants who experienced more time pressure compared to the TP participants of NS-Rep1 and/or Oulu's TP participants.

C. Comparison With the Related Work

Teasley et al. and Leventhal et al. experimentally observed the presence of confirmation bias in functional software testing [8], [57]. They found that the participants manifested confirmation bias irrespective of the effect of expertise level, level of detail of specifications and error feedback factors [8], [57]. Causevic et al. made a similar observation, participants manifested confirmation bias irrespective of the development approach (test driven development vs test-last) they used [58]. Our results support these findings because our participants significantly manifested confirmation bias, also irrespective of time pressure; whether they belonged to the TP or NTP groups.

We found a significant effect of time pressure on our dependent variables in contrast to the quantitative findings by Mäntylä et al. and Topi et al. [11], [59]. Topi et al. experimentally investigated the relation between task complexity and time availability [59]. The authors did not find time availability to affect task performance [59]. Mäntylä et al. also couldn't find their independent variable - time pressure to negatively effect the effectiveness of software testing [11]. However, they observed that time pressure improved the efficiency of test case development [11]. Mäntylä and Itkonen also observed a positive effect of time pressure in the context of software testing. [60]. They observed that multiple testers under time pressure were better in defect detection compared to non-time-pressured individuals [60]. Yet, we report the negative effects of time pressure on confirmation bias in software testing. Several other studies also report that time pressure deteriorates software quality and is a source of errors by affecting testers performance and development [15], [61], [62], [63], [64], [65].

Qualitative studies by Baddoo and Hall in software process improvement, Shah et al. in global software testing and Willson and Hall in software quality report time pressure as a deteriorating factor [14], [17], [66]. Our study quantitatively (via statistical significance tests) supports the qualitative findings of those studies. The results of our study provide quantitative evidence in support of the qualitative findings by Salman et al. that time pressure promotes confirmation bias among software testers [12]. Salman et al. found time pressure as an antecedent to the confirmatory behaviour of software testers [12].

D. Implications for Research and Practice

In general, there is a need to develop a *code-breaking* attitude or disconfirmatory behaviour among software testers by enhancing their *outside-of-the-box* thinking ability. Experience is a vital factor in this respect, but it is acquired only with the passage of time. Practitioners can specifically take care of the following:

- The outside-of-the-box thinking ability is required when dealing with ambiguous or incomplete requirements. These types of requirements are attributed to companies that implement agile software development [67], [68].
- Novice software testers should be trained in the project domain or involved earlier in the software development lifecycle [12], [16]. This may broaden their perspective, and thus, enable them to give more coverage to inconsistent test cases.
- The practice of test suite reviews especially by the members of the same project or team may also apprehend the manifestation of confirmation bias [12]. This recommendation relates to the heuristics-baseline extension step.

Data from every replication increased the count of inconsistent test cases in the baseline suite (Section VII-C).

- Harnessing multiple modes of testing may help lessen confirmation bias. For example, automated testing may help suppress the effects of time pressure on confirmation bias. This recommendation particularly targets testing on higher levels, e.g., integration testing and system testing.

According to Salman et al., automated testing is an antecedent to confirmation bias because it is difficult to automate inconsistent test cases when the output is an uncertain value, condition, or event [12]. However, automated testing may leverage time for software testers to focus on the designing and/or execution of difficult-to-automate test cases, especially inconsistent ones, via manual testing [12]. In other words, performing manual and automated testing in a complementary fashion may help manoeuvre time pressure. This is an indirect way to promote and leverage disconfirmatory behaviour among testers with the ultimate goal of improving software quality.

There is a need to develop mitigation techniques to prevent the manifestation of confirmation bias. Such techniques are referred to as debiasing techniques [2]. We implicate the following to researchers:

- Debiasing for confirmation bias can adopt a *proactive* approach. Proactive debiasing refers to inhibiting the antecedents to confirmation bias. For example, time pressure is an antecedent to confirmation bias, therefore creating a time pressure free or non-time pressured environment may inhibit the promotion of confirmation bias.
- Debiasing can also adopt a *reactive* approach. A reactive debiasing approach would be to develop or introduce an intervention that mitigates the adverse effects of confirmation bias after its manifestation. For example, a practice of test suite reviews may increase the coverage of inconsistent test cases, thus, possibly diminishing the manifested confirmation bias [12].
- Is the choice to develop/implement proactive or reactive debiasing techniques contextual? Or, a tandem approach that applies both proactive and reactive debiasing is well suited?
- Could there be absolute or robust debiasing techniques for confirmation bias in software testing?

We encourage more replications. Variations to the following elements would leverage more insight into the phenomenon [29] of confirmation bias and time pressure:

- Recruit experienced participants. Training or knowledge of functional (black-box) software testing is the core requirement whether a replication is run with novice (students or inexperienced professionals) or experienced participants. The training topics mentioned in Section VII-A are ample for establishing the necessary foundation for the participants.
- Modify the experimental protocol (experimental design, experimental object, material, different or multiple metrics to measure confirmation bias) [29].
- Choose different geographical locations (other than Europe) that may reflect the influence of respective educational systems and cultural backgrounds.
- Vary the setting to industry. *Caveat:* The industrial setting could be fraught with challenges considering the treatment application procedures. We built time pressure psychologically as part of the operationalisation construct (Section IV). The implementation of the same construct may be impossible in the industry due to multiple uncontrollable factors, e.g., interruptions by non-participants, emergency call-ups for the participants, etc. Therefore, replication can be achieved by considering different treatment applications and transmission procedures to understand operationalisation limits.

A step-by-step guide by Vegas et al. can be followed further to proceed with the aggregated analyses on the subject, especially when experimental protocols and operationalisation differ among experiments [69].

Confirmation bias may interact, overlap, reinforce or be reinforced by other cognitive biases [70]. It is, therefore, difficult to determine the direction of the cause-and-effect relationship between such cognitive biases [2]. This concept is referred to as *biasplex*, and is not specific to confirmation bias [70], [71]. Confirmation bias along with the other cognitive biases; the bandwagon effect, miserly information processing and status-quo bias form the *inertia* biasplex [70]. Which one of these biases is possibly at interplay with confirmation bias in the software testing context is yet to be explored. The phenomenon of confirmation bias is not limited to software testing. It also occurs in other knowledge areas of SE, e.g., construction, design, maintenance and requirements [2]. These phases of development (knowledge areas) still need investigation as to how confirmation bias may impact the umbrella process of software quality.

X. Threats to Validity

Wohlin et al. [72], recommend to discuss the following threats for experimental studies.

A. Construct Validity

We measure confirmation bias not only in the relative terms (z) but also in absolute terms. The construct of time pressure (and no-time pressure) was operationalised after determining the duration from a pilot run that was part of the original experiment's protocol. Moreover, during the data extraction phase, we discussed and resolved the confusing test cases - Section VII-C. These steps inhibit the threats of inadequate preoperational explication of constructs and mono-method bias. Our experimental design is limited to one object, which could have introduced the mono-operation bias threat. In our opinion, performing meta-analysis with multilevel modelling has leveraged the prevention of confounding constructs and level of constructs threat. The experiments are not prone to the interaction of testing and treatment threat because the participants were not aware of the treatment, and all students performed the experimental task irrespective of their consent for participation. We added additional guidance in the scripted guidelines to tackle the human-specific problems that could compromise our operationalisation construct. In this respect, we anticipated the

participants querying about the remaining time. The guidelines instructed to openly announce the remaining time if a participant in the TP group asked about it. For the NTP group, the experimenters would inform of the remaining time only to that particular participant.

B. Internal Validity

Similar to the original experiment, the participants were taught and trained together followed by their random assignment either to the control or treatment groups. Hence, none of the replications are prone to the selection-maturation interaction threat. The joint teaching and training setup also prevented the compensatory equalisation of treatments threat. The degree level differed between the Oulu and Novi Sad's experiments. This could cause selection-maturation threat because of their experience characteristics. However, the results via multilevel modelling neither support the occurrence of this threat nor the selection-history threat. The experimental executions for the control and treatment groups were run in parallel (in different rooms) for all four experiments, which dismisses the imitation of treatments, compensatory rivalry and resentful demoralisation threats. As mentioned in Section V-E, only the consented participation was incentivised, otherwise all students performed the experimental activity. Therefore, the bonus marks or additional course credits for participation in the respective experiments are not an internal validity threat because it was neither coercive nor constituted undue influence [73]. The cultural differences among participants (international degree programme: mixed multiple nationalities in Oulu, 90% Serbians in Novi Sad) are a possible threat to our results despite considering the differences in language and degree in our analysis.

C. External Validity

We recruited students as proxies for novice professionals in our family of experiments [54]. Instead of using conventional simplistic labelling as students or professionals, we characterised their experience with respect to our experimental objectives (Section VIII-B). This lessens the interaction of selection and treatment threat. Yet, their limited (novice) experience in combination with the realistic object (MusicFone) may not rule out the interaction of selection and treatment threat to our study. The time pressure was operationalised in controlled academic settings, i.e., the other factors that could have influenced the application of treatment were not present. This controlled environment, despite its necessity, is not representative of an industrial setup, which makes our study prone to the interaction of setting and treatment threat. As mentioned earlier, there are multiple additional factors present in the industrial environment (e.g., phone call disturbances) that may add to the manifestation of confirmation bias. The use of pen and paper for performing the task does not exacerbate the interaction of setting and treatment threat because we focused only on the designing of test cases which leads the execution of test cases. Moreover, the use of the realistic object further alleviates this particular threat.

D. Conclusion Validity

We addressed the threat of violated assumptions of statistical tests by ensuring that every respective test met its assumptions before its execution. For example, we ran non-parametric tests (independent sample: Mann-Whitney, dependent sample: Wilcoxon signed-rank) for the *t-test* family when the assumption of normality was violated. The data was applied normality transformations to meet the assumptions of multivariate normality for statistical test of the *F-test* family. We also report those effect-size measures that correspond to the run statistical test - Section VI. Details related to multilevel modelling are reported in Section VI and Section VIII-D. In order to address the error rate threat that relates to the significance level $-\alpha$, we applied the Bonferroni type adjustment as mentioned in Section VIII-A. Objectively addressing the threat of violated assumptions via statistical interventions has also mitigated other threats in analyses. For example, a threat may have occurred due to different measurers[10] and analysts across two sites. Moreover, following the interactions guidelines (Section V-B) has further alleviated analyses' threats. We performed multiple steps to ensure the reliability of measures. For example, the experimental instrumentation that was used in the replications was improved as a result of a pilot run. In order to alleviate subjectivity in the identification of (in)consistent test cases, multiple interactive sessions between Oulu and Novi Sad's experimenters (measurers and analysts) were held as per detailed in Section VII-C. We ensured the reliability of treatment implementation between the two sites (among four experiments) by: 1) developing and following the replication package; 2) following the process for managing interactions between the experimenters to get useful similar replications [23] - Section V-B. Additionally, we ensured the reliability of treatment for each experiment by validating the sanity check hypothesis. Despite these cautions, cultural aspects specific to human characteristics could still have implications for applying the treatment. For example, the potential effect of the tone or body language of the experimenters to make reminders — already discussed in Section IX-B.

XI. Conclusion and Future Directions

We executed a series of three similar experimental replications; one internal (with respect to the original experimental site) in Oulu, Finland and two external in Novi Sad, Serbia. The aim was to examine the same objective as of our original experiment — Salman et al. [20]; whether software testers manifest confirmation bias while designing functional test cases. How does time pressure effect the manifestation of confirmation bias in this regard? With this aim, we verified and validated the results of the original experiment. Additionally, we performed a joint analysis of our family of experiments (1 original $+3$ replications) to provide joint results on the manifestation of confirmation bias and the role of time pressure. The joint

[10] *Measure* and *analyst* are the other two types of experimenters in addition to *Trainer* and *Monitor* [29].

results enabled us to observe the plausible effects of contextual variables related to two different experimental sites within this family.

In our family of experiments, we observed that testers (student participants: proxies for novice professionals) significantly manifest confirmation bias while designing the functional test cases. Additionally, time pressure promoted the confirmatory behaviour (a manifestation of confirmation bias) among testers when designing the test cases, per the joint results of this family. The characteristics of participants were not observed to effect the results. The two different experimental sites affected the results, i.e., the manifestation of confirmation bias as an effect of time pressure. However, our data was not enough to enable the observation of the site specific (contextual) variables on the results. Conclusively, we verified and validated the results on the manifestation of confirmation bias, and extended them with a perspective on time pressure, in this context, by employing the joint analysis.

Practitioners are recommended to develop a disconfirmatory/code breaking attitude with an outside-of-the-box thinking capability. This would leverage a quality testing when either dealing with incomplete requirements specification, ambiguous specifications or lack of experience. The effect of time pressure can be contained by the complementary designing and execution of manual and automated testing. The complementary approach in which manual testing focuses on the designing of inconsistent test cases, and automated on consistent test cases.

Future directions of this work include the development of contextual and non-contextual debiasing strategies for confirmation bias. Additionally, the examination of the plausible effects of confirmation bias related biasplex. It is important to find ways to manoeuvre or contain time pressure, especially when industrial setups lack automated testing resources. Thus, time pressure may not promote confirmation bias during software testing. It is worthy to also examine the relation between confirmation bias and time pressure for non-functional testing. Nonetheless, how does confirmation bias may manifest in non-functional testing, is a yet-to-be-examined direction.

ACKNOWLEDGMENT

The authors acknowledge the guidance by Dr. Sira Vegas (Universidad Politecnica de Madrid, Madrid, Spain) for performing multilevel modelling.

REFERENCES

[1] D. Arnott, "Cognitive biases and decision support systems development: A design science approach," *Inf. Syst. J.*, vol. 16, no. 1, pp. 55–78, 2006.

[2] R. Mohanani, I. Salman, B. Turhan, P. Rodriguez, and P. Ralph, "Cognitive biases in software engineering: A systematic mapping study," *IEEE Trans. Softw. Eng.*, vol. 46, no. 12, pp. 1318–1339, Dec. 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8506423/

[3] R. Mohanani, P. Ralph, B. Turhan, and V. Mandic, "How templated requirements specifications inhibit creativity in software engineering," *IEEE Trans. Softw. Eng.*, vol. 48, no. 10, pp. 4074–4086, Oct. 2022.

[4] G. Calikli and A. Bener, "Empirical analyses of the factors affecting confirmation bias and the effects of confirmation bias on software developer/tester performance," in *Proc. 6th Int. Conf. Predictive Models Softw. Eng. (PROMISE)*, 2010, pp. 1–11. Accessed: Oct. 31, 2015. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1868328.1868344

[5] G. Calikli, A. Bener, T. Aytac, and O. Bozcan, "Towards a metric suite proposal to quantify confirmation biases of developers," in *Proc. Int. Symp. Empirical Softw. Eng. Meas.*, 2013, pp. 363–372.

[6] L. M. Leventhal, B. Teasley, D. S. Rohlman, and K. Instone, "Positive test bias in software testing among professionals: A review," in *Proc. Int. Conf. Human-Comput. Interact.*, 1993, pp. 210–218.

[7] W. Stacy and J. MacMillan, "Cognitive bias in software engineering," *Commun. ACM*, vol. 38, no. 6, pp. 57–63, 1995.

[8] B. E. Teasley, L. M. Leventhal, C. R. Mynatt, and D. S. Rohlman, "Why software testing is sometimes ineffective: Two applied studies of positive test strategy," *J. Appl. Psychol.*, vol. 79, no. 1, pp. 142–155, 1994.

[9] G. Calikli and A. Bener, "Empirical analysis of factors affecting confirmation bias levels of software engineers," *Softw. Qual. J.*, vol. 23, pp. 695–722, Dec. 2015. [Online]. Available: http://link.springer.com/10.1007/s11219-014-9250-6

[10] G. Calikli, A. Bener, and B. Arslan, "An analysis of the effects of company culture, education and experience on confirmation bias levels of software developers and testers," in *Proc. ACM/IEEE 32nd Int. Conf. Softw. Eng.*, vol. 2, 2010, pp. 187–190.

[11] M. V. Mäntylä, K. Petersen, T. O. A. Lehtinen, and C. Lassenius, "Time pressure: A controlled experiment of test case development and requirements review," in *Proc. 36th Int. Conf. Softw. Eng. (ICSE)*, 2014, pp. 83–94. Accessed: Nov. 18, 2015. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2568225.2568245

[12] I. Salman, P. Rodriguez, B. Turhan, A. Tosun, and A. Gureller, "What leads to a confirmatory or disconfirmatory behaviour of software testers?" *IEEE Trans. Softw. Eng.*, vol. 48, no. 4, pp. 1351–1368, Apr. 2022.

[13] R. Ramač et al., "Prevalence, common causes and effects of technical debt: Results from a family of surveys with the IT industry," *J. Syst. Softw.*, vol. 184, Feb. 2022, Art. no. 111114.

[14] H. Shah, M. J. Harrold, and S. Sinha, "Global software testing under deadline pressure: Vendor-side experiences," *Inf. Softw. Technol.*, vol. 56, no. 1, pp. 6–19, 2014.

[15] M. Cataldo, "Sources of errors in distributed development projects: Implications for collaborative tools," in *Proc. ACM Conf. Comput. Supported Cooperative Work*, 2010, pp. 281–290. Accessed: Mar. 5, 2018. [Online]. Available: http://portal.acm.org/citation.cfm?id=1718971

[16] F. P. Seth, O. Taipale, and K. Smolander, "Organizational and customer related challenges of software testing: An empirical study in 11 software companies," in *Proc. IEEE 8th Int. Conf. Res. Challenges Inf. Sci. (RCIS)*, 2014, pp. 1–12.

[17] N. Baddoo and T. Hall, "De-motivators for software process improvement: An analysis of practitioners' views," *J. Syst. Softw.*, vol. 66, no. 1, pp. 23–33, 2003.

[18] I. Hernandez and J. L. Preston, "Disfluency disrupts the confirmation bias," *J. Exp. Social Psychol.*, vol. 49, no. 1, pp. 178–182, 2013. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S002210311200176X

[19] K. Ask and P. A. Granhag, "Motivational bias in criminal investigators' judgments of witness reliability," *J. Appl. Social Psychol.*, vol. 37, no. 3, pp. 561–591, 2007.

[20] I. Salman, B. Turhan, and S. Vegas, "A controlled experiment on time pressure and confirmation bias in functional software testing," *Empirical Softw. Eng.*, vol. 24, no. 4, pp. 1727–1761, Dec. 2018. [Online]. Available: http://link.springer.com/10.1007/s10664-018-9668-8

[21] A. Santos et al., "A family of experiments on test-driven development," *Empirical Softw. Eng.*, vol. 26, no. 3, pp. 1–53, 2021. [Online]. Available: http://arxiv.org/abs/2011.11942

[22] J. Carver, "Towards reporting guidelines for experimental replications: A proposal," in *Proc. 1st Int. Workshop Replication Empirical Softw. Eng.*, vol. 1, 2010, pp. 1–4. Accessed: Mar. 12, 2017. [Online]. Available: http://carver.cs.ua.edu/Papers/Conference/2010/2010_RESER.pdf

[23] N. Juristo, S. Vegas, M. Solari, S. Abrahão, and I. Ramos, "A process for managing interaction between experimenters to get useful similar replications," *Inf. Softw. Technol.*, vol. 55, no. 2, pp. 215–225, 2013, doi: 10.1016/j.infsof.2012.07.016.

[24] A. Santos, O. Gomez, and N. Juristo, "Analyzing families of experiments in SE: A systematic mapping study," *IEEE Trans. Softw. Eng.*, vol. 46, no. 5, pp. 566–583, May 2020.

[25] N. Juristo and O. S. Gómez, "Replication of software engineering experiments," in *Empirical Software Engineering and Verification: LASER 2008-2010. Lecture Notes in Computer Science*, B. Meyer and M. Nordio, Eds., Berlin, Germany:

Springer, vol. 7007, 2012, pp. 60–88. [Online]. Available: http://link.springer.com/10.1007/978-3-642-25231-0_2

[26] N. Juristo and S. Vegas, "Using differences among replications of software engineering experiments to gain knowledge," in *Proc. 3rd Int. Symp. Empirical Softw. Eng. Meas. (ESEM)*, 2009, pp. 356–366.

[27] M. G. Mendonça et al., "A framework for software engineering experimental replications," in *Proc. IEEE Int. Conf. Eng. Complex Comput. Syst., (ICECCS)*, 2008, pp. 203–212.

[28] A. Santos, S. Vegas, M. Oivo, and N. Juristo, "A procedure and guidelines for analyzing groups of software engineering replications," *IEEE Trans. Softw. Eng.*, vol. 47, no. 9, pp. 1742–1763, Sep. 2019.

[29] O. S. Gomez, N. Juristo, and S. Vegas, "Understanding replication of experiments in software engineering: A classification," *Inf. Softw. Technol.*, vol. 56, no. 8, pp. 1033–1048, 2014.

[30] F. J. Shull, J. C. Carver, S. Vegas, and N. Juristo, "The role of replications in Empirical Software Engineering," *Empirical Softw. Eng.*, vol. 13, no. 2, pp. 211–218, Apr. 2008.

[31] V. Mandić, J. Markkula, and M. Oivo, "Towards multi-method research approach in empirical software engineering," in *Proc. 10th Int. Conf. Product-Focused Softw. Process Improvement*, 2009, pp. 96–110.

[32] T. Gilovich, D. Griffin, and D. Kahneman, *Heuristics and Biases*, 8th ed. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[33] A. Tversky and D. Kahneman, "Judgement under uncertainty: Heuristics and biases," Oregon Res. Inst. Res. Bull., Tech. Rep., 1973.

[34] D. Arnott, "A taxonomy of decision biases," School Inf. Manag. Syst., Monash Univ., Melbourne, Australia, Tech. Rep., 1998. Accessed: Nov. 18, 2015. [Online]. Available: http://www.sims.monash.edu.au/staff/darnott/biastax.pdf

[35] D. Kahneman, D. Lovallo, and O. Sibony, "Before you make that big decision...," *Harvard Bus. Rev.*, pp. 51–60, Jun. 2011. Accessed: Jun. 8, 2019. [Online]. Available: http://website.aub.edu.lb/units/ehmu/Documents/before-you-make-that-big-decision.pdf

[36] D. Arnott and S. Gao, "Behavioral economics for decision support systems researchers," *Decis. Support Syst.*, vol. 122, Feb. 2019, Art. no. 113063. doi: 10.1016/j.dss.2019.05.003.

[37] L. M. Leventhal, B. E. Teasley, and D. S. Rohlman, "Analyses of factors related to positive test bias in software testing," *Int. J. Human-Comput. Stud.*, vol. 41, pp. 717–749, 1994.

[38] I. Salman, "The effects of confirmation bias and time pressure in software testing," Ph.D. dissertation, Univ. Oulu, Oulu, Finland, 2019.

[39] M. Kuutila, M. Mäntylä, U. Farooq, and M. Claes, "Time pressure in software engineering: A systematic review," *Inf. Softw. Technol.*, vol. 121, May 2020, Art. no. 106257. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0950584920300045?via%3Dihub

[40] M. Kuutila, M. Mantyla, U. Farooq, and M. Claes, "What do we know about time pressure in software development?" *IEEE Softw.*, vol. 38, no. 5, pp. 32–38, Sep./Oct. 2021.

[41] M. Kuutila, M. V. Mantyla, M. Claes, and M. Elovainio, "Reviewing literature on time pressure in software engineering and related professions: Computer assisted interdisciplinary literature review," in *Proc. IEEE/ACM 2nd Int. Workshop Emotion Awareness Softw. Eng. (SEmotion)*, 2017, pp. 54–59.

[42] O. Hazzan, O. Hazzan, Y. Dubinsky, and Y. Dubinsky, "The software engineering timeline : A time management perspective," in *IEEE Int. Conf. Softw.-Sci., Technol. Eng. (SwSTE)*, 2007, pp. 95–103.

[43] N. Nan and D. E. Harter, "Impact of budget and schedule pressure on software development cycle time and effort," *IEEE Trans. Softw. Eng.*, vol. 35, no. 5, pp. 624–637, Sep./Oct. 2009.

[44] S. Linßen, D. Basten, and J. Richter, "Antecedents and consequences of time pressure in Scrum projects: Insights from a qualitative study," in *Proc. 51st Hawaii Int. Conf. Syst. Sci.*, 2018, pp. 4835–4844.

[45] NASA, "NASA Task Load Index," *Human Mental Workload*, vol. 1, no. 6, 2006, pp. 21–21. Accessed: Mar. 18, 2016. [Online]. Available: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=16243365

[46] Human Performance Group at NASA's Ames Research Center, "NASA Task Load Index (TLX)," 1987. Accessed: May 4, 2022. [Online]. Available: https://humansystems.arc.nasa.gov/groups/TLX/publications.php

[47] "7.2.6—Model assumptions and diagnostics assumptions—STAT 505." Accessed: Jan. 6, 2023. [Online]. Available: https://online.stat.psu.edu/stat505/lesson/7/7.2/7.2.6

[48] C. O. Fritz, P. E. Morris, and J. J. Richler, "Effect size estimates: Current use, calculations, and interpretation." *J. Exp. Psychol. General*, vol. 141, no. 1, pp. 2–18, 2012. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/21823805

[49] M. Borenstein, L. V. Hedges, J. P. Higgins, and H. R. Rothstein, *Introduction to Meta-Analysis*, 1st ed. Hoboken, NJ, USA: Wiley, Jan. 2009.

[50] A. Field, J. Miles, and Z. Field, *Discovering Statistics Using R*. Newbury Park, CA, USA: Sage, 2012.

[51] J. J. Randolph, "Free-marginal multirater kappa: An alternative to Fleiss fixed-marginal multirater kappa," in *Proc. Joensuu Univ. Learn. Instruction Symp.*, 2005, pp. 1–20.

[52] J. J. Randolph, "Online kappa calculator," 2008. Accessed: Feb. 8, 2018. [Online]. Available: http://justusrandolph.net/kappa/#dInfo

[53] D. Falessi et al., "Empirical software engineering experts on the use of students and professionals in experiments," *Empirical Softw. Eng.*, vol. 23, no. 1, pp. 452–489, 2018.

[54] I. Salman, A. T. Misirli, and N. Juristo, "Are students representatives of professionals in software engineering experiments?" in *Proc. Int. Conf. Softw. Eng.*, vol. 1, 2015, pp. 666–676.

[55] J. Hox, "Multilevel modeling: When and why," in *Classification, Data Analysis, and Data Highways*. Berlin, Germany: Springer-Verlag, 1998, pp. 147–154.

[56] A. Field, *Discovering Statistics Using IBM SPSS Statistics*, 5th ed. Newbury Park, CA, USA: Sage, 2018.

[57] L. M. Leventhal, B. E. Teasley, and D. S. Rohlman, "Analyses of factors related to positive test bias in software testing," *Int. J. Human-Comput. Stud.*, vol. 41, no. 5, pp. 717–749, Nov. 1994. Accessed: Apr. 29, 2017. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1071581984710792

[58] A. Causevic, R. Shukla, S. Punnekkat, and D. Sundmark, "Effects of negative testing on TDD: An industrial experiment," in *Proc. Int. Conf. Agile Softw. Develop.*, 2013, pp. 91–105. Accessed: Oct. 31, 2015. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-38314-4_7

[59] H. Topi, J. S. Valacich, and J. A. Hoffer, "The effects of task complexity and time availability limitations on human performance in database query tasks," *Int. J. Human Comput. Stud.*, vol. 62, no. 3, pp. 349–379, 2005.

[60] M. V. Mäntylä and J. Itkonen, "More testers—The effect of crowd size and time restriction in software testing," *Inf. Softw. Technol.*, vol. 55, no. 6, pp. 986–1003, 2013.

[61] I. Salman and B. Turhan, "Effect of time-pressure on perceived and actual performance in functional software testing," in *Proc. Int. Conf. Softw. Syst. Process (ICSSP)*, 2018, pp. 130–139.

[62] R. Baskerville, L. Levine, J. Pries-Heje, and S. Slaughter, "How Internet software companies negotiate quality," *Computer*, vol. 34, no. 5, pp. 51–57, May 2001.

[63] J. Verner, J. Sampson, and N. Cerpa, "What factors lead to software project failure?" in *Proc. 2nd Int. Conf. Res. Challenges Inf. Sci.*, Piscataway, NJ, USA: IEEE Press, Jun. 2008, pp. 71–80.

[64] A. Deak, T. Stålhane, and G. Sindre, "Challenges and strategies for motivating software testing personnel," *Inf. Softw. Technol.*, vol. 73, pp. 1–15, May 2016.

[65] M. Cataldo and J. D. Herbsleb, "Factors leading to integration failures in global feature-oriented development," in *Proc. 33rd Int. Conf. Softw. Eng.*, New York, NY, USA: ACM, May 2011, pp. 161–170.

[66] D. N. Wilson and T. Hall, "Perceptions of software quality: A pilot study," *Softw. Qual. J.*, vol. 7, no. 1, pp. 67–75, 1998. [Online]. Available: http://link.springer.com/article/10.1023/B:SQJO.0000042060.88173.fe

[67] F. Paetsch, A. Eberlein, and F. Maurer, "Requirements engineering and agile software development," in *Proc. 12th IEEE Int. Workshops Enabling Technol. Infrastructure Collaborative Enterprises (WETICE)*, 2003, pp. 308–313.

[68] O. Albayrak, H. Kurtoglu, and M. Biçakçi, "Incomplete software requirements and assumptions made by software engineers," in *Proc. 16th Asia-Pacific Softw. Eng. Conf.*, 2009, pp. 333–339.

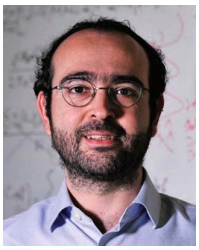
[69] S. Vegas, I. Salman, P. Riofrío, and N. Juristo, "A method for aggregating families of experiments in software engineering a step by step guide," *Empirical Softw. Eng.*, early access, 2023.

[70] P. Ralph, "Possible core theories for software engineering," in *Proc. 2nd SEMAT Workshop General Theory Softw. Eng. (GTSE)*, Piscataway, NJ, USA: IEEE Press, 2013, 2013, pp. 35–38.

[71] P. Ralph, "Toward a theory of debiasing software development," in *Proc. Lecture Notes Bus. Inf. Process.*, vol. 93, 2011, pp. 92–105. [Online]. Available: http://link.springer.com/10.1007/978-3-642-25676-9

[72] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*. Berlin, Germany: Springer-Verlag, 2012.
[73] National Bioethics Advisory Commission, *Ethical and Policy Issues in International Research: Clinical Trials in Developing Countries*. Washington, DC, USA: NBAC, 2012. Accessed: Oct. 22, 2018. [Online]. Available: https://bioethicsarchive.georgetown.edu/nbac/clinical/Vol1.pdf

Iflaah Salman received the M.Sc. and Ph.D. degrees in information processing science from the University of Oulu, Oulu, Finland. She is a Postdoctoral Researcher with the School of Engineering Science, Lappeenranta-Lahti University of Technology (LUT), Finland. Her research interests include empirical software engineering, human aspects in software engineering, artificial intelligence, and software testing. She has industrial experience working as a Software Developer and Software Quality Assurance Engineer at Lahore, Pakistan. For more information please visit: https://www.linkedin.com/in/iflaahsalman/ and follow on https://www.researchgate.net.



Burak Turhan (Senior Member, IEEE) received the Ph.D. degree from Boğaziçi University. He is a Professor with the M3S Research Unit at the University of Oulu, Oulu, Finland. His research focuses on empirical software engineering, artificial intelligence, quality assurance and testing, human factors, and (agile) development processes. He has published over 120 articles in international journals and conferences, received several best paper awards, and secured funding for several large scale research projects. He has served on the editorial boards of several software engineering journals, as a Co-Chair for PROMISE'13, ESEM'17, PROFES'17, and EASE'23, and as a steering committee member for PROMISE and ESEM. He is a member of the ACM and ACM SIGSOFT. For more information please visit: https://turhanb.net.

Robert Ramač received two M.Sc. degrees in information technologies and engineering management from the University of Novi Sad. He is a Software Developer with TIAC ltd. Currently, he is working toward the Ph.D. degree with the Faculty of Technical Sciences at the University of Novi Sad. His areas of interest are empirical software engineering, software development, technical debt, and the improvement of the software development process.

Vladimir Mandić (Senior Member, IEEE) received the M.Sc. degree in EE from the University of Novi Sad, Serbia, and the Ph.D. degree in information processing science and software engineering from the University of Oulu, Finland. He is an Associate Professor in software engineering with the Faculty of Technical Sciences, University of Novi Sad, Novi Sad, Serbia. His areas of interest are software process improvement, empirical software engineering, software quality, goal-driven measurement approaches, technical debt, and value-based software engineering. He is a member of ACM.