A process for managing interaction between experimenters to get useful similar replications

Natalia Juristo [a], Sira Vegas [a,*], Martín Solari [b], Silvia Abrahão [c], Isabel Ramos [d]

[a] Universidad Politécnica de Madrid, Campus de Montegancedo s/n, 28660 Madrid, Spain
[b] "Universidad ORT Uruguay", Cuareim 1451, Montevideo, Uruguay
[c] Universidad Politécnica de Valencia, Camino de Vera s/n, 46022 Valencia, Spain
[d] Universidad de Sevilla, Avda. Reina Mercedes s/n, 41012 Sevilla, Spain

Keywords:

Empirical studies Experimentation

Replication

Combination of experiment results

A B S T R A C T

Context: A replication is the repetition of an experiment. Several efforts have been made to adopt replication as a common practice in software engineering. There are different types of replications, depending on their purpose. Similar replications keep the experimental conditions as alike as possible to the original ones. External similar replications, where the replicating experimenters are not the same people as the original experimenters, have been a stumbling block. Several attempts at combining the results of replications have resulted in failure. Software engineering does not appear to be well suited to such replications, because it works with complex experimentally immature contexts. Software engineering settings have a large number of variables, and the role that many of them play is unknown. A successful (or useful) similar replication helps to better understand the phenomenon under study by verifying results and/or identifying contextual variables that could influence (or not) the results, through the combination of experimental results.
Objective: To be able to get successful similar replications, there needs to be interaction between original and replicating experimenters. In this paper, we propose an interaction process for achieving successful similar replications.
Method: This process consists of: *an adaptation meeting*, where experimenters tailor the experiment to the new setting; *querying*, to settle occasional inquiries while the experiment is being run; and *a combination meeting*, where experimenters meet to discuss the combination of replication outcomes with previous results. To check its effectiveness, the process has been tested on three different replications of the same experiment.
Results: The proposed interaction process has helped to identify new contextual variables that could potentially influence (or not) the experimental results in the three replications run. Additionally, the interaction process has helped to uncover certain problems and deviations that occurred during some of the replications that we would have not been aware of otherwise.
Conclusions: There are signs that suggest that it is possible to get successful similar replications in software engineering experimentation, when there is appropriate interaction among experimenters.

1. Introduction

Single experiments are liable to yield fortuitous results. The repetition of an experiment to verify its results is called replication. Experiment replication, then, is a key feature of experimentation. Replications output new data, which, compared with the outcomes of earlier experiments, help to understand the reliability of the results.

* Corresponding author.
E-mail addresses: natalia@fi.upm.es (N. Juristo), svegas@fi.upm.es (S. Vegas), martin.solari@ort.edu.uy (M. Solari), sabrahao@dsic.upv.es (S. Abrahão), iramos@us.es (I. Ramos).

There are different types of replications, each playing a role in corroborating results [6,7]. An experiment can be repeated by the same experimenters in the same setting (to check whether the results were a one-off chance occurrence). Or a replication can be run by other experimenters at a different site (to check whether the results are independent of the experimenters and the setting). Replications can be similar or differentiated. In similar replications the experimental conditions are reproduced as closely as possible to the original setting (to verify results, find out the range of conditions under which the results hold, and the effects of new variables on the results [11]). Differentiated replications pursue the same goal without complying with the same experimental protocol (to identify whether the experimental procedures biased the results).

Most of the results of similar replications of SE experiments run by other researchers (to verify results) differ from the ones of the original experiment [9,17,18,22,23]. The only studies having similar results have been internal replications ([15,20,25]). Whenever the results of the replication differ from the outcomes of the original experiment, the experimenters consider the replication to have been a failure, because they are unable to combine the results. The replicating experimenters mainly put this failure down to variations in the experimental conditions of the experiment and the replication [9,17,18,22,23].

Software engineering (SE) experiments have a highly complex context, involving numerous variables (about developers, techniques, projects, etc.), many of which are still unknown. When the setting is changed, there is a risk of the results not being comparable to the earlier outcomes. This may materialize if too many changes have made to the experimental conditions. Another possibility is that some experimental conditions have been unintentionally changed. These inadvertent changes may lead to differences in the results. Since the variables changed have been overlooked, results cannot be explained.

We believe that similar replications are of use for advancing SE knowledge. A successful similar replication helps to: (1) verify results and identify contextual variables that might not have an influence on the results, in case the results of the original experiment hold, or (2) identify contextual variables that might have an influence on the results, in case the results of the original experiment do not hold.

The goal of this paper is to analyze whether it is possible to obtain successful similar replications, if the appropriate mechanisms are applied during replication. The mechanisms traditionally applied when replicating experiments involve replication packages or publications on the experiment, where a detailed description of the original experiment is given (design, data analysis, etc.). In this paper we propose incorporating to the traditional mechanisms, an interaction process among experimenters. The interaction will allow: (1) keep the changes to the original experiment to the minimum required to adapt the experiment to the new context, and (2) verify results and/or identify contextual variables that might (or not) influence the results.

The article is organized as follows. Section 2 presents the interaction types commonly used in SE replications. Section 3 presents our proposal of interaction to output successful similar replications. With the aim of evaluating this proposal, we have run three replications of the same experiment. Section 4 describes the organization of the evaluation. Section 5 illustrates, by means of one of these replications, how this interaction transpires. Section 6 describes the results of the other two replications. Section 7 discusses the results of the whole evaluation. Finally, Section 8 presents the conclusions of this research.

2. Interaction types for replicating software engineering experiments

To run similar replications, the experimenters that are going to run the replication should have as many details as possible about the baseline experiment. For this purpose, some sort of communication is required among the experimenters that ran the baseline (original) experiment, and the experimenters running the replication.

The context of a SE experiment is very complex due to the very many variables involved in the phenomenon under examination. So, a lot of information about the experiment is needed to run a similar replication. Software engineering experimenters have tried out different levels of communication when replicating, as we will see later ([1,5,14,21,23,24]). Communication aims to transmit enough details of the experiment for it to be reproduced as accurately as possible. Communication between experimenters consists of documentation and interaction between the groups.

Usually, the documentation interchanged between experimenters consists of an experimental package or laboratory package. At present, there is no agreement about what contents an experimental package should have. The contents of existing experimental packages vary. For a detailed description of the different types of packages, see [24,26,27]. In other cases, the documentation interchanged consists of publications of the original experiment. The contents of these publications may also vary. Guidelines on how to report experiments have been proposed recently [8].

Although documentation is a key factor to be able to run a similar replication, we believe that it is not enough using replication packages and/or publications about the experiment, no matter how detailed they are. The interaction among experimenters is just as important. Some authors of this paper have replicated experiments using only documentation, and they have suffered problems of different nature: not understanding the rationale for certain design decisions, missing information about how some specific experimental tasks have to be done, impossibility to combine results of the replication with previous results because more changes than strictly necessary were made, etc. Other authors agree that it is very difficult that, even a very well described and justified experiment is able to transfer all experimental know-how needed to run a replication [24].

SE experimenters have used different types of interaction:

- The simplest interaction is just interchanging documentation. This interaction was used, for example, in [21]. There is no additional interaction aside from the mere transfer of the documentation (it could be a replication package – the contents of which may range from very basic to very thorough, publications about the experiment or both).
- At the next step, there is more interaction and earlier experimenters answer the replicating experimenters' queries. This was used, for instance, in [14].
- On the next rung up, there is occasional cooperation among experimenters. For example, the replicating experimenters visit the earlier experimenters while they are performing the replication, or earlier experimenters analyze the data collected by the replicating experimenters. This was used, for example, in [1,5,23].
- At the top end of the ladder, much closer interaction has been used in some replications. In [24], Shull and colleagues describe several replications of an experiment run by different groups of experimenters. The interaction in this case is composed of different types of workshops (virtual and on-site), e-mail, web portals and a knowledge repository. The cooperation takes place among all the groups of experimenters.

Because researcher interaction is resource consuming, and consequently expensive, we are looking to keep interaction to the minimum, while, at the same time, assuring that the replication will be useful.

3. A proposal of an interaction process for similar replications

We propose a process of interaction for experimenters that makes it easier to tailor the replication to the new context with help from earlier experimenters. It aims to adapt the experiment to the new context without setting it so far apart from the baseline experimental conditions as to prevent comparison of results. A similar replication run in line with our process might help to provide new useful information: whether the results are independent

of the experimenters and the site, whether the results hold in this similar context, or what effect the new variables have on the response variable.

The interaction process we propose for similar replications is organized as follows.

3.1. Adaptation meeting

During the replication definition and planning phase, the two groups of experimenters meet to study and tailor the experiment to the context of the new setting.

At this meeting, researchers analyze the context of the new setting in which the new replication is to take place, and context-induced changes are made to the experiment. For the adaptation meeting to be successful, it should deal with at least the following points: resource-related issues (like, time, space, computers, etc., available for running the replication), and subject-related issues (like number, experience, knowledge, necessary training, etc.).

In the event that the replicating experimenters are acquainted with the experiment to be replicated, they could do the adaptation themselves. The adaptation meeting could be substituted with a telephone or e-mail discussion among the two groups of experimenters about critical tailoring issues (available time, subjects' prior knowledge, training, etc.). Owing to the time and travelling they involve, meetings are the most expensive part of the interaction. Cutting out unnecessary meetings could result in less resource-consuming ways of interacting with equally successful results.

Additionally, during the meeting the experimenters might decide not to run the replication for different reasons. For example:

- The contexts are too different, and too many changes have to be made to the original experiment.
- It is not understood the effect the changes will have on the results.
- The changes imply increasing too much the threats to the validity of the replication.
- Etc.

3.2. Querying

Provision should be made for the possibility of making telephone or e-mail inquiries to settle occasional queries while the experiment is being run.

Details that have not been discussed thoroughly enough during the adaptation meeting are always liable to crop up, and new issues may turn up as one goes along.

3.3. Combination meeting

The experimenters should meet again when the data of the replication has been analyzed in order to combine the replication outcomes with previous results.

Before they meet, the replicating and/or original experimenters will have previously compared the results of the replications. Therefore, they will have already identified inconsistencies in the results and have individually examined the context and experimental conditions of the replication to find variables that could explain the differences in the results. The comparison of the replication results with the earlier results is a separate step from the combination meeting that does not require interaction.

At the meeting, they will again review the context and experimental conditions of the replication in search of any change that might have happened during the experiment operation and may have previously been overlooked. The important point about this meeting is that it reviews how the replication was run. So, it is absolutely vital that it should be attended by the experimenters present during the operation of the baseline experiment and the replication. These experimenters should be well acquainted with the context and settings of each replication.

In order to apply the interaction proposed here, the following constraints or requirements must be met:

- The original experimenters must be accessible and willing to cooperate with the replicating experimenters.
- The replicating experimenters must be motivated to run the replication.
- The two groups of experimenters must have the resources required to conduct the meetings.

4. Application study of the proposed interaction process

The goal of this paper is to analyze whether the proposed interaction process allows successful similar replications. To check the suitability of this process, we ran three replications of the same experiment in three different settings. We set out to answer to the following research questions:

- **RQ1**: Is the proposed interaction process useful for running successful similar replications? (It is able to verify results and identify contextual variables that both do or do not influence the results).
- **RQ2**: Is each and every step of the interaction process necessary?
- **RQ3**: What are the limitations of the process?

The sources for data collection are: the adaptation meeting minutes of each replication (there were two people taking notes of what transpired during the meeting), emails exchanged or notes on telephone conversations during querying, and, finally, the recordings of the combination meeting. The data were gathered by examining all the available material.

The experiment replicated here is one originally run by Basili and Selby [2,3]. The goal of this experiment is to examine the effectiveness and efficiency of different code evaluation techniques. This experiment has been replicated several times, for example, by Kamsties and Lott [13,14], or Roper et al. [21], Wood et al. [28]. The reason we chose this series of experiments is that they are especially well documented compared with others.

Two authors of this paper (Juristo and Vegas) ran five internal replications of this experiment [10,12]. In this paper our replications will be referred to as UPM replications. Even though the UPM replications are not the baseline experiment of this series of replications, they serve this purpose for the three replications run. In the following, we summarize the key features of the UPM replications. For more details, see [10,12].

The aim of the UPM replications is to evaluate the relative effectiveness (defects detected by the test cases generated), efficiency (time taken and number of generated test cases), and defect visibility (defects reported by subjects) of three code evaluation techniques: equivalence partitioning [19], decision coverage [4] and code reading by stepwise abstraction[1] [16]. To do this, we used a factorial design with three programs (acting as blocking variables). The replications were run in three sessions. In each session, the subjects individually applied one technique to one program. All three techniques were exercised during each session, and subjects exercised only one technique. Each session was dedicated to a single program. At the end of the three sessions of the replications, each

[1] Note that for this technique, no test cases are generated, and therefore the variable *number of generated test cases* cannot be reported. For this same reason, the value for the effectiveness variable is the same as for the defect visibility variable.

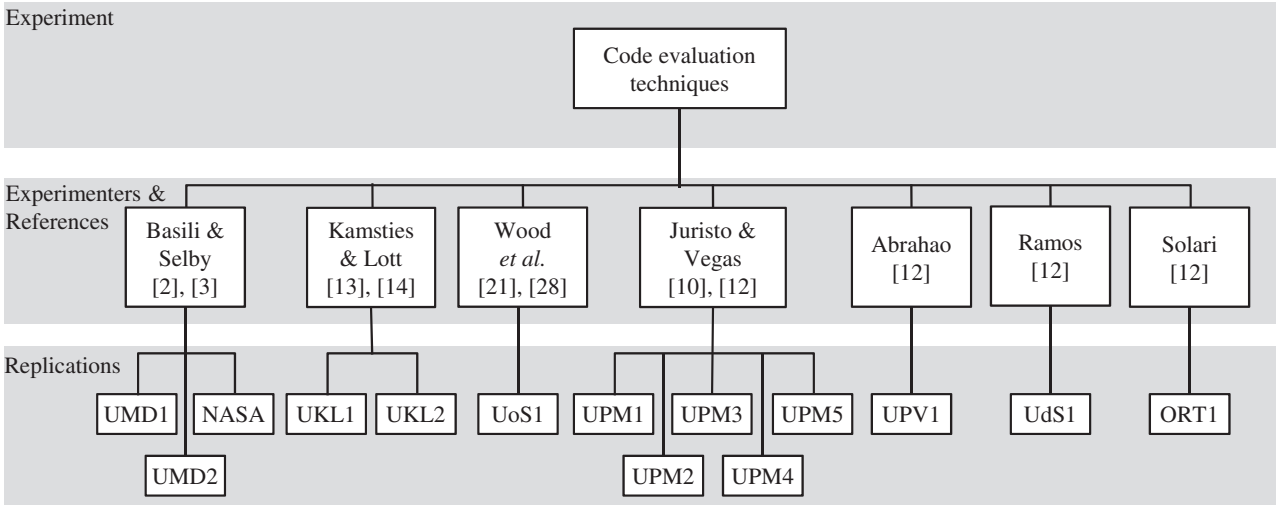

Fig. 1. History of the experiment.

subject will have applied each of the three techniques under study, one on each of the three programs (one technique per day and program). Each session has a 4-h duration. An average student, who has followed the training and studied the techniques as proposed in the training, is able to complete the task in a shorter period. In each of the three sessions, the subjects will apply the techniques and, in the case of the dynamic techniques (equivalence partitioning and decision coverage), will also have to execute the test cases. The subjects do not know the techniques before attending this course[2]. They were given several hours of training to learn how they should be applied before the experiment was run.

We opted to run replications at native Spanish-speaking universities, because, thanks to the similarity of some aspects of the setting, we could control some subject-related variables, like previous training or native language. The replications take place in the context of a Software Engineering Experimentation Network partially funded by the Spanish government. This means that the participants are highly motivated. The participants are: UPM, Universidad Politécnica de Valencia, Universidad de Sevilla and Universidad ORT-Uruguay. From now on, we will refer to these replications as the UPV replication, UdS replication and ORT replication respectively. The UPV and UdS replications were run simultaneously, whereas the ORT replication was executed after the other two had been completed. Fig. 1 summarizes the history of the experiment.

Apart from interaction, running a replication involves the use of documentation, as mentioned in Section 2. The experimental package we used is an extension of Wood and colleagues' package [29]. In actual fact, the replication package we used for the described replications contains:

- A document describing the UPM replications. This document was built on the contents of Wood and colleagues' [29] (which contained the experimental material). The UPM document includes a main body with the definition and planning of the experiment, the experimental operation, and a series of annexes including:
  - A script describing the tasks the experimenters are to perform.
  - Source code with and without defects for experiment programs.
  - Description of the faults in the experiment programs and the failures they caused when executed.
  - Experiment program specification.
  - Instructions sheets for subjects per program and technique.
  - Data collection forms.
  - Examples of how to fill in the data collection forms.
- Material used for training. The main body is a 100-page document containing lecture notes. It also contained the following annexes:
  - Slides and literature references.
  - Training program source code with and without defects.
  - Description of the faults in the training programs and the failures they caused when executed.
  - Training program specification.
  - Solutions for each of the training programs.
- Electronic version of the material to run the UPM experiment.
- A document describing the new (UPV/UdS/ORT) replication. As the replication setting was different from the UPM replication, another document was drafted with the same contents as mentioned above, but specific for the new setting.
- Electronic version of the material to run the new replication.

We have identified the following threats to the validity of this study:

- The researchers proposing and applying the interaction process are the same. Consequently, they may be especially motivated, resulting in the interaction having better results. However, we think that it makes sense to have an initial evaluation where we are involved, before asking somebody else to apply the interaction process.
- The three examined replications are of the same experiment. Consequently, the results cannot be generalized. There are inherent characteristics of the experiment in question (such as, for example, the context in which it is run, the treatments examined in the experiment and programs) that might influence the performance of the interaction process.

5. Illustration of how interaction develops

In this section, we show how interaction flows. To do this, we will describe the UPV replication. This replication was run by experimenters from the Universidad Politécnica de Valencia.

[2] However, they might have some experience in informal testing, as they have attended programming courses and have programming experience. Even though these programming courses do not deal with testing issues, subjects may have tested their own programs.

5.1. Adaptation meeting

During the adaptation meeting, the experimenters identified a number of changes that needed to be made to the UPM experiment to account for the differences in the UPV context. The main differences between the UPM setting and the UPV setting were:

- The subjects already knew how to use the techniques, as they had already been taught in earlier SE course units. Therefore, students did not have to be taught the techniques beforehand as they did in the UPM experiment. To adapt the experiment to this new condition, the subjects' training was changed. In the UPM experiment, where students were unfamiliar with the techniques, they were given lectures. In the UPV replication, the training was reduced to a refresher tutorial on the techniques in the shape of one of the practical exercises in the replication package. This change should guarantee that the subjects in both replications are similarly knowledgeable about the techniques.
- Due to subject teaching constraints, the experiment operation sessions would have to be interleaved with the training sessions. It is not possible to give all training first and then run the experiment as was done at UPM. To adapt the experiment to this new condition, one technique and all three programs (instead of one program and all three techniques) were used in each experimental session. This means there could be a sequence effect, and therefore, this replication does not deal with two validity threats:
  - The subjects can now discuss the defects they found in the programs after the session. Therefore, the technique applied on the second day may benefit from the copying effect. We tried to discourage this behavior by evaluating subjects on how well they had applied the techniques rather than the defects they had found.
  - By the second day students will be acquainted with the experimental procedure. Therefore, the technique applied on the second day may benefit from the experimental procedure learning effect, and behave better than in the baseline experiment. Maybe not effectiveness (techniques are very different), but yes time and defect visibility.
- The maximum duration of each session would have to be two instead of the 4 h allowed at UPM. To adapt the experiment to this new condition, each 4 h UPM session is divided into two 2 h sessions at UPV. The subjects generate the test cases in the first session and execute them in the second.
- The time required to run the experiment needed to be reduced. Two changes were made to adapt the experiment to this new condition:
  - The code review technique was left out, and just equivalence partitioning and branch coverage were exercised.
  - Test cases were executed for just one of the two programs that the UPV subjects worked with.

This means that there is a first session focusing on the application of the structural technique, a second session on the application of the functional technique, and a third session concerned with the running of the test cases for a program.

Table 1 shows the changes made to the UPM experiment for each difference in the UPV setting.

The decisions on the changes to be made to the experiment were taken jointly by experimenters from UPM and UPV during the adaptation meeting. We (UPM and UPV) accepted only the changes strictly required by the new setting in an attempt to keep the changes to the bare minimum.

5.2. Querying

UPV experimenters did not make any queries during this step.

5.3. Comparison of results

Table 2 summarizes UPM and UPV results. Note that there are 5 UPM replications that have showed consistent results.

A first attempt at combining UPM and UPV results was made. The comparison of the results indicates that the relative behavior of the techniques is equivalent in both replications for all the examined variables (effectiveness, application time, number of generated test cases and visibility of defects). However, the values of these variables are not the same. Effectiveness has fallen equally for both techniques at UPV. The functional technique application time is less at UPV, whereas the structural technique application time is unchanged. Fewer test cases are generated by both techniques at UPV, and the drop is sharper for the functional technique. Defects are less visible at UPV, and this is more pronounced for cosmetic defects of commission.

A first attempt at interpreting these results was made by UPM researchers. We analyze the possible sequence effects (learning and copying). UPV subjects apply first the structural technique and then the functional. This could cause an improvement of the functional technique. However, no such improvement seems to occur as regards effectiveness (the functional technique is as effective as the structural). Although efficiency does seem to improve (it takes less time to apply the functional technique than the structural technique compared to UPM subjects). This means that in case there is a sequence (learning and/or copying) effect, it seems to influence technique efficiency only.

However, sequence effects do not seem to explain the other discrepancies found in results. Therefore, we make a first attempt at coming up with possible explanations for these discrepancies. We look for other possible variables that might have been altered by unintended changes.

The functional technique generates a lot fewer test cases at UPV than at UPM. One might think, in principle, that subjects are doing a worse job of applying the technique, as the decrease in the number of test cases could lead to a drop in effectiveness.

Subject training is a variable that could explain some differences for the functional technique. Let us hypothesize that UPV subjects received worse training than UPM subjects. Worse training would result in the subjects applying the functional technique poorly, and would therefore explain the drop in the variable values

Table 1
Changes to the UPM experiment for the UPV replication.

New condition	Change to experiment
Subjects already acquainted with techniques	4-h refresher tutorial rather than 16 h of lectures on the same material
Training and operation not sequential	One technique applied after receiving training on the technique (each subject to a different program out of three). First structural, then functional
2 h per session	Each UPM session is split into two: test case generation performed in one session and test case execution in another session
Less time available	Code review technique left out Test cases executed for just one of the programs, after all techniques have been applied

Table 2
UPM and UPV results.

Aspect	Replication	Technique behavior	Mean values
Effectiveness	UPM	$(F = S) > CR$	*F*: 82.84% *S*: 81.82% CR: 46.53%
	UPV	$F = S$	*F*: 70.73% *S*: 70.33%
Time	UPM	$F < S < CR$	*F*: 62′ *S*: 96′ CR: 167′
	UPV	$F < S$	*F*: 56′ *S*: 95′
Number of test cases	UPM	$S > F$	*F*: 14 *S*: 19
	UPV	$S > F$	*F*: 9 *S*: 15
Defect visibility	UPM	*F*(cc) < others	*F*(cc): 30% Others: 87%
	UPV	*F*(cc) < others	*F*(cc): 4% Others: 75%

F: functional; CR: code review; *S*: structural; *F*(cc); cosmetic, commission defects.

(effectiveness, application time, number of generated test cases and visibility of defects). As the subjects would not do a good job of applying the technique, it would not take them as long, they would generate fewer test cases, and, as a result, the technique would be less effective. This hypothesis could even explain the fact that the subjects would find fewer defects. This effect may, however, be offset by the sequence effect of the experimental procedure, causing subjects not to take as long to apply the functional technique, and causing a smaller drop in effectiveness.

Yet, the training hypothesis is unable to explain the behavior of the structural technique. The subjects take just as long to apply this technique in both replications and, compared with UPM values, actually generate more test cases than for the functional technique. If our hypothesis were true, the results for the effectiveness of the structural technique should have been closer to UPM values.

Additionally, we expected training in UPV to be equivalent to UPM: the experimental package contained the all the training material used at UPM (slides, lecture notes, training exercises, etc.), the UPV trainer is a senior testing professor, and the techniques are commonly taught in testing courses.

The training, sequence and copying hypotheses, as well as the uncertainty about what is happening with the structural technique, was taken to the combination meeting.

5.4. Combination meeting

During combination, we confirmed that training could have been influencing the results. At the adaptation meeting, it had been agreed that, as UPV subjects had already taken a course unit on the techniques earlier and should know how to use them, they would be set a practical exercise to simply remind them of how to use the techniques. UPM subjects had received several training lectures instead. At the end of the training, the subjects of both replications were expected to have a similar knowledge of the techniques.

This change was designed to tailor the training to the experimental subjects' previous knowledge. It did not have the desired effect, however. We found, during the combination meeting, that UPV experimenters did not use the training material included in the experimental package at all, since the material provided was not tailored to UPV conditions. The experimental package contained all the training material, whereas the training programs alone would have been sufficient for UPV. Failure to use the material provided in the experimental package meant that the practical exercise used was simpler than the ones in the experimental package; therefore it is very likely that UPV subjects were not as well acquainted with the techniques as UPM subjects.

The combination meeting also helped to uncover another possible cause for the behavior of the techniques. It was discovered that the subjects were not equally motivated during the execution of the experiment. The UPM replications served to pass or fail the course unit, whereas participation in the replication had hardly any effect on the final grade at UPV. This unintended change could mean that subjects did not make an effort to get good results. If this were the case, time differences, induced by lack of subject motivation during the experiment, could possibly be influencing the behavior of the techniques at UPV.

These two factors (training and motivation) might not be affecting the two techniques equally. The subjects might have learned one technique better than the other or have been more motivated 1 day than the other.

Additionally, during the combination meeting, one new possible explanation came up: the subjects did not have time to fully apply the structural technique. UPV experimenters realized during discussions with UPM experimenters that there had been a scheduling error. The subjects had 2 h to apply the technique. From UPM experiment experience, 2 h seemed to be enough time for subjects to apply the technique. But we overlooked the start-up time. UPV experimenters needed time during the first session to present the experiment (between 15 and 30 min). This was the session in which the structural technique was applied. Consequently, the subjects were working under time pressure, as 90 min does not appear to be enough time to apply the structural technique. This new variable could explain (along with motivation) the fact that, having generated comparatively more test cases than the functional technique, the test cases behaved equally in terms of effectiveness. Realizing that they were not going to have enough time to finish the task, the subjects made every effort to get it done, probably generating test cases at random rather than using the technique.

The question of how much time subjects would be given to apply the techniques, which, unlike the UPM experiment, was limited by the context, was incorrectly addressed at the adaptation meeting. Perhaps if the issue had been more thoroughly dealt with then, we would have been able to find a solution to the problem (or maybe not, as it was a context-dependent issue).

Additionally, at the combination meeting, UPV experimenters stated that they would have liked to have had a description of their expected attitude to the experimental subjects during the experimental operation. They had doubts about how they were to treat the subjects during the experiment. For example, were they to answer questions about the techniques, programming language or programs? Were subjects to be allowed to talk or use their notes while the experiment was being run? Similarly, the subjects were not sure how to fill in the data collection forms. Although the experimental package included examples of how to fill in the forms, there was nothing to say that this information was to be explained to the subjects and was not for the experimenters' use only.

6. Evaluation of the proposed interaction process

In order to evaluate the proposed interaction, two more replications were run, apart from the one described in the previous section: the UdS and ORT replications.

6.1. UdS replication

During the adaptation meeting of the UdS replication, the experimenters also identified a number of changes that needed

Table 3
Changes to the UPM experiment for the UdS replication.

New condition	Change to experiment
Subjects already acquainted with techniques	4-h refresher tutorial rather than 16 h of lectures on the same material
Training and operation not sequential	One technique applied after receiving training on the technique (each subject to a different program out of three). First code review, then structural, then functional
There are not enough computers	Pair work
2 h per session	Each UPM session is split into two: test case generation performed in one session and test case execution in another session
Less time available	Test cases executed for just one of the programs, after all techniques have been applied

to be made to the UPM experiment to account for the differences in the UdS context. Table 3 shows the changes made to the UPM experiment for each difference in the UdS setting.

Like the UPV replication, this replication suffers from the sequencing (learning and/or copying) effects, as subjects apply first code review, then the structural technique, and then the functional technique. Also pair work could influence (increase) technique effectiveness or affect (increase or decrease depending on whether the subjects collaborate or divide up the task) efficiency.

It is worth noting that, as in the UPV replication, the decisions on the changes to be made to the experiment were taken jointly by experimenters from UPM and UdS during the adaptation meeting. We (UPM and UdS) accepted only the changes strictly required by the new setting in an attempt to keep the changes to the bare minimum.

UdS experimenters formulated several queries related to experiment operation during the querying step. Table 4 summarizes the results for UdS compared with UPM. When comparing the UPM–UdS results we observed that the results for effectiveness, number of generated test cases and dynamic technique application time were worse for UdS than for UPM and were comparable with the UPV results. Taking into consideration UPV findings, we hypothesized that training and motivation could again explain these results (although motivation and training may not be affecting all techniques equally). Additionally the effects of sequence (learning and/or copying) could be to some extent counteracting the negative effects of training and motivation. Deficient training and/or less motivation in a given techniques could result in the subjects applying the technique poorly. If the subjects were applying the technique badly, it would not take them as long, they would generate fewer test cases and, as a result, the technique would be less effective (these effects being counteracted in the case of the functional and structural techniques by the sequence effects of learning and/or copying). The combination meeting helped to confirm that both training and motivation could possibly be influencing the results. As at UPV, the experimental subjects were already acquainted with the techniques and were to be set a practical exercise simply as a reminder of how to use the techniques. Also as at UPV, the replicating experimenters again failed to use the training material included in the experimental package, because it was not tailored to their context. Once again, the experiment had hardly any effect on the final grade at UdS, a factor that might discourage subjects from making the effort to get good results.

Additionally, all the results (effectiveness, number of generated test cases, dynamic technique application time and number of test cases generated) are consistently better at UdS than at UPV, although they only lead to statistically significant differences in some cases (technique application time and visibility of cosmetic defects of commission). Pair work could explain this. It could also explain why the static technique at UdS (not studied at UPV) is equally as effective as at UPM, but its application time is halved. In this case, pair work could appear to be improving both its effectiveness and its efficiency, and even offsetting the possible influence of motivation and training (there no sequence effect in this case, as the code review technique is the first being applied).

Table 4
UPM, UPV and UdS results.

Aspect	Replication	Behavior	Mean values
Effectiveness	UPM	$(F = S) > CR$	*F*: 82.84% *S*: 81.82% CR: 46.53%
	UPV	$F = S$	*F*:70.73% *S*: 70.33%
	UdS	$(F = S) > CR$	*F*: 67.99% *S*: 76.48% CR: 41.80%
Time	UPM	$F < S < CR$	*F*: 62′ *S*: 96′ CR: 167′
	UPV	$F < S$	*F*: 56′ *S*: 95′
	UdS	$F < (S = CR)$	*F*: 48′ *S*: 68′ CR: 64′
Number of test cases	UPM	$S > F$	*F*: 14 *S*: 19
	UPV	$S > F$	*F*: 9 *S*: 15
	UdS	$F = S$	*F*: 11 *S*: 12
Defect visibility	UPM	*F*(cc) < others	*F*(cc): 30% Others: 87%
	UPV	*F*(cc) < others	*F*(cc): 4% Others: 75%
	UdS	*F*(cc) < others	*F*(cc): 34% Others: 77%

F: functional; CR: code review; *S*: structural; *F*(cc); Cosmetic, commission defects.

However, some irregularities were observed by UPM researchers in UdS data. These irregularities would increase the threats to validity.

- The UdS experimenters were given two options in the experiment design: pairs should either remain unchanged or never be repeated throughout the replication. The final design is a hybrid of both alternatives, where some pairs are repeated several times and others never come together again.
- The experimental groups defined in the final design were not balanced. This means some groups had fewer pairs.
- There is a subject randomization error, meaning that there are pairs that use a program more than once.
- Finally, some of the subjects did not turn up for some replication sessions. This led to some subjects working alone.

As regards these matters, the UdS experimenters stated during the combination meeting that the design was not totally instantiated for their context. They had to further specify the design, and particularly the part referring to pair formation and experimental group assignment. When the UdS experimenters proceeded to fur-

Table 5
Changes to the UPM experiment for the ORT replication.

New condition	Change to experiment
Junior subjects without programming language experience	Junior subjects
Computer room not available	Test case execution left out (no results on defect visibility)
Less time available	Code review technique left out Experiment in one session One of the programs left out

ther instantiate the design, and due to their inexperience in experimentation, they made the mistakes mentioned earlier.

Additionally, the UdS experimenters also stated at the combination meeting that they missed a guide for preparing the material to be handed out to the experimental subjects. Two mistakes were made during the experiment operation at UdS. First, the subjects never received the supplementary sheet. Second, the code specification for the structural and code review techniques was handed out to students before it should have been. This had no effect on the results, though. The supplementary sheet was most useful for the structural and code review people, because they did not have the specification. Since UdS students were given the code specification beforehand, the effect of the errors was cancelled out. Finally, as with UPV replication, the replicating experimenters said that they would have liked to have had a description of their expected attitude to the experimental subjects during the experimental operation.

6.2. ORT replication

Regarding the ORT replication, no adaptation meeting was performed. It was impossible for the two groups of experimenters to have a face-to-face meeting. At any rate, we decided to go ahead with the replication to check whether the adaptation meeting was really necessary. The adaptation meeting was, therefore, switched for e-mail responses to queries. In any case, the experimenters also identified a number of changes that needed to be made to the UPM experiment to account for the differences in the ORT context. Table 5 shows the changes made to the UPM experiment for each difference in the ORT setting.

We find a priori that two of the changes may lead to differences in the results with respect to UPM:

- ORT subjects are junior students, some of whom had hardly any programming language experience at all. UPM students have good programming skills. The structural technique can be expected to behave worse in this replication. When applying the structural technique, the subjects use the source code to generate test cases for the program. However, when applying the functional technique, they use the program specification and never see the source code. The change enables us to study the effect of subject programming language experience.
- The ORT experiment is run in only one session, in which the subjects were given unlimited time to apply the two techniques to two programs. It takes a lot of concentration and effort to apply a technique. Therefore, subjects are likely to be tired by the time they come to apply the second technique. We expect the effectiveness of the second technique applied to be lower as a consequence of tiredness. However, not all the subjects use the techniques in the same order, which means that the overall effectiveness might decrease.

The ORT experimenters did not make any query during the querying step. Table 6 summarizes the results for the ORT replication compared to UPM. In view of the results, and taking into account what happened at UPV and UdS, we initially considered the possibility of the fact that ORT subjects were inexperienced with the programming language influencing the results. If this were the case, we would expect the functional technique to behave better and take less time to apply, as no source code is required in this case. Additionally, training, motivation and tiredness do not appear to be influencing the results. The combination meeting helped to confirm that neither training nor motivation should be influencing the results: both were similar to the UPM context.

However, this does not explain why the techniques generated a different number of test cases with different programs. There is no apparent reason for the relative behavior of the technique to vary in terms of the technique. In the other three cases, the techniques had behaved equally, irrespective of the program. The subjects appear not to be applying the structural technique as thoroughly as they should for just one of the programs. During the combination meeting, ORT experimenters considered that a possible explanation could be that one of the programs was used as a training program. This may have caused the unexpected improvement in the results, as the program used for training was more like the one for which the technique performed best in terms of test case generation.

It was at the combination meeting that some of the effects of switching the adaptation meeting for e-mail responses to queries became apparent. ORT experimenters cut back the amount of interaction with UPM experimenters, because they only queried changes about which they were unsure. This means that not all the changes made by ORT experimenters were discussed with UPM experimenters, as they had been at UPV and UdS. ORT experimenters only consulted UPM experimenters about what they considered to be major changes: for instance, leaving out the static technique. They made other changes on their own initiative and without previous consultation. As a result, UPM experimenters did not find out about all the changes made to the replication until

Table 6
UPM, UPV, UdS and ORT results.

Aspect	Replication	Behavior	Mean values
Effectiveness	UPM	$(F = S) > CR$	*F*: 82.84% *S*: 81.82% CR: 46.53%
	UPV	$F = S$	*F*:70.73% *S*: 70.33%
	UdS	$(F = S) > CR$	*F*: 67.99% *S*: 76.48% CR: 41.80%
	ORT	$F > S$	*F*: 80.29% *S*: 70.33%
Time	UPM	$F < S < CR$	*F*: 62′ *S*: 96′ CR: 167′
	UPV	$F < S$	*F*: 56′ *S*: 95′
	UdS	$F < (S = CR)$	*F*: 48′ *S*: 68′ CR: 64′
	ORT	$F < S$	*F*: 66′ *S*: 111′
Number of test cases	UPM	$S > F$	*F*: 14 *S*: 19
	UPV	$S > F$	*F*: 9 *S*: 15
	UdS	$F = S$	*F*: 11 *S*: 12
	ORT	P1: $F = S$ P2: $F < S$	*F*: P1: 9; *P*2: 17 *E*: P1:17; P2: 24

F: functional; CR: code review; *S*: structural; *F*(cc); cosmetic, commission defects.

Table 7
Variables identified in the replications.

Replication	Identified variables	Response variable	Direction of the effect	Technique
UPV	Training	Effectiveness	↑	Functional
	Motivation	Effectiveness	↑	Structural
	Time pressure	Effectiveness	↓	Structural
	Sequencing	Effectiveness	↑	Functional
		Application time	↓	
UdS	Training	Effectiveness	↑	Functional
	Motivation	Effectiveness	↑	Structural
	Pair work	Application time	↓	Code review
		Effectiveness	↑	Code review
	Sequencing	Effectiveness	↑	Functional
		Application time	↓	Structural
ORT	Tiredness	Effectiveness	No influence	Functional Structural
	Programming experience	Effectiveness	↑	Structural
		Application time	↓	

↑: Increases; ↓: decreases.

Table 8
Summary of the results of the interaction process.

Replication	Deviations	Problems
UPV	*Training not tailored* Scheduling problem	*Experimenters' attitude to subjects* *Queries on subject form filling*
UdS	*Incomplete design* *Training not tailored*	*Experimenters' attitude to subjects* *Experimental material preparation guide not detailed enough*
ORT	More changes than necessary	*Experimenters' attitude to subjects* *Missing e-forms*

the combination meeting, finding more changes than expected (and, in this case, more than necessary).

Additionally, at the combination meeting, ORT experimenters stated (as was the case in UPV and UdS replications) that they would have liked to have had a description of their expected attitude to the experimental subjects during the experimental operation. They also missed an electronic version of the forms required to run the experiment, as they had to take them from the annex of the document describing the replication.

7. Discussion of results

The results of the three replications run will be used to answer the research questions.

7.1. Is the proposed interaction process useful for running successful similar replications?

The three replications run – following the proposed process – were useful for confirming some experimental results and discovering possible variables influencing (or not) the results.

To be precise, we were able to confirm the following outcomes:

- The functional and structural techniques appear to be equally effective and could both be more effective than code review.
- There are signs that it takes subjects less time to apply the functional technique than the structural technique, and less again to apply the structural technique than code review.
- The functional technique appears to generate fewer test cases than the structural technique.
- The number of test cases generated by dynamic techniques appears to vary depending on the program.
- Cosmetic defects of commission seem to be less visible than others.

The identified variables that could be influencing the results are:

- Technique effectiveness could increase in proportion to *subject motivation*.
- *If subjects have programming language experience*, the structural technique effectiveness could increase, and the structural technique application time decrease.
- *The better trained the subjects* are in a technique, the more effective the technique is likely to be.
- *Pair work* could decrease technique application time and increase code review technique effectiveness.
- *Work under pressure* could decrease structural technique effectiveness, although it does not necessarily mean that fewer test cases are generated.
- *The sequence effect* could decrease the time it takes the subject to do the experiment. It also appears to increase technique effectiveness.
- *Tiredness* does not seem to have an influence.

Table 7 summarizes the variables identified in the replications.

Additionally, the proposed interaction process has helped us to identify: (1) deviations of the replication operation from what was originally planned, and (2) Problems that the replicating experimenters had when running the experiments. Some of the deviations and problems of the UPV replication appear again in the UdS replication. They could not have been avoided in UdS, as both replications were run in parallel. Table 8 summarizes the deviations and problems identified thanks to the proposed interaction process.

Since the replications helped us to confirm some results, and identify new variables possibly (not) influencing the results, we could conclude that the interaction process has helped us to obtain

successful replications. But, we did not expect deviations to take place during interaction or the replicating experimenters to have trouble running the experiments. This could question the usefulness of the proposed interaction process. However, we were able to know that the deviations and the problems existed thanks to the interaction process. More precisely, thanks to the combination meeting. If we had not met, we would not have been aware of them.

Additionally, some deviations and/or problems turned out to produce changes in some context variables (e.g. working under pressure, training). The changes allowed us to explore these variables.

Finally, it is worth noting that the process does not guarantee that all contextual variables that could influence the results are identified. This all depends on how accurate the interpretation of the results made by the experimenters is. Additionally, the effect of these variables can be guessed, but never determined. It is essential to run new experiments, in order to test the real effect of the discovered variables.

7.2. Is each and every step of the interaction process necessary?

The interaction process proposed consists of the following steps: an *adaptation meeting*, where experimenters tailor the experiment to the new setting; *querying*, to settle occasional inquiries while the experiment is being run; and *a combination meeting*, where experimenters meet to discuss the combination of replication outcomes with previous results.

We have found the **adaptation meeting** essential for adapting the experiment to the new setting in the three replications we have run. These replications appear to confirm how difficult it is to find a context that is exactly the same as the context in which the baseline experiment was run, even if at first glance they look as if they might be. For this reason, it is very likely that changes will have to be made to the baseline experiment to be able to adapt it to the new setting. It is essential that the changes are kept to the minimum.

At some point, we thought that this meeting could be substituted with a telephone or e-mail discussion, provided the replicating experimenters were well acquainted with the experiment to be replicated. However, the experience with ORT suggests that this meeting seems to be necessary. In the ORT replication, the two groups of experimenters were not obliged to meet, and therefore, the replicating experimenters got the feeling that they alone were responsible for tailoring the experiment to the new setting and running the replication. This led them into thinking that they only needed to consult the earlier experimenters when they had a problem that they did not know how to solve. Interaction slowed down, fell below the absolute minimum, and collaborative work, which appeared to be one of the keys to successful similar replication, broke down. Since the adaptation meeting was left out, the overlap between the replications shrank more than was called for by the differences in the context and more changes were made than are strictly necessary to tailor the experiment to the new conditions. Additionally, this led to unnecessarily increase the validity threats.

In case a physical meeting is not possible, we suggest a virtual one (teleconference, etc.).

The replications have revealed that some people tend to not use the **querying** step. UPV and ORT experimenters did not formulate any query during this step, although UdS experimenters did make use of this mechanism. One might think that there would be fewer deviations and/or problems in the replication if the experimenter queries were resolved than if the experimenters had no queries (or vice versa), but this was not the case. Deviations and problems occurred in all the replications. Consequently, the use of this step is not a reliable indicator. As there are experimenters that do make use of this step, we do not believe that it should be omitted.

We have also found that the **combination meeting** is equally as necessary. Not only has it helped us to confirm candidate variables to explain the discrepancies between the results of the different replications, but it has also helped us to identify new variables (see Table 7), discover deviations in the replication from what originally planned (see Table 8, column 2), or find out about problems the replicating experimenters had during the experiments (see Table 8, column 3). Without the combination meeting, we would have been unable to discover the problems and deviations, and to identify some variables (motivation, pressure and training).

7.3. What are the limitations of the process?

In order to discover the limitations of the process and possible improvements, we are going to analyze the problems and deviations presented in Table 8.

The problems and deviations in italics in Table 8 indicate defects in the experimental package, although they could have been resolved through interaction. In actual fact, the deviations and all the problems, except for UPV replication, could have been satisfactorily solved in time if the replicating experimenters had contacted the original experimenters with their queries. For some reason, the replicating experimenters did not make use of that possibility (as has very often happened in other replications). The success of the adaptation and combination meetings, and the consequences of leaving the adaptation meeting out of the ORT replication, suggest that any interaction that does not entail direct contact among experimenters tends to be less effective. This could be remedied by holding a pre-execution meeting just before the experiment is operated (when the replicating experimenters have the replication ready) to settle any queries. Note, importantly, that this improvement does not imply any real change to the interaction process, which is essentially the same. All we want to do is force the replicating experimenters to voice their last-minute doubts and/or problems, because, otherwise, they appear not to do so.

The problems not set in italics in Table 8 deserve a special mention. We should stress that there is no way that we could have improved the interaction to prevent the **scheduling problem**. In the UPV replication, the subjects had limited time to finish the task, and this was conditioned by the context. Even if we had known that the subjects would be short of time, nothing could have been done to prevent this. The alternative would have been not to run the replication.

On the other hand, the fact that **more changes** were made to ORT replication **than strictly necessary** confirms that it would have been highly advisable to check the decisions taken with the original experimenters. When discussing the changes at the combination meeting, UPM experimenters took the view that at least one (related to the use of one of the experiment programs as a training program) and perhaps a second (application of the two techniques consecutively in the same session) were unnecessary changes that could, however, have a big influence on results. Unnecessary changes were made beyond what were strictly necessary for adaptation to the context. The explanation for what happened is to be found in the interaction used. The adaptation meeting for jointly defining and planning the experiment cannot be left out, as, without it, changes were made that had effects on the replication preventing combination. With hindsight, we found that some, if not all, of the changes (mainly the above two) would possibly not have been made if the adaptation meeting had been held. The adaptation meeting obliges the experimenters to analyze and evaluate all the changes to the replication. This assures that the number of changes introduced is reduced to a bare minimum.

Finally, let us thoroughly analyze the problem related to the **incomplete design** of the UdS replication. Although this problem can be attributed to the experimental package, it is striking that it only occurred in one of the replications. UPV and ORT replications have a conspicuous difference from the UdS replication that could explain this point. The person that attended the adaptation meeting on behalf of UdS was the person in charge of coordinating the course unit on which the experiment was run but not for running the replication or actually training the subjects. Consequently, the person in attendance may, unconsciously, not have been as involved in the meeting as she should have been, resulting in the design that was passed on to UdS experimenters perhaps not being as detailed as the one supplied to UPV. The solution to this problem would be to have any one directly involved in running the replication attend the meetings.

8. Conclusions

The general hypothesis underlying the research reported here was that, in SE, it is possible to run successful similar replications with appropriate interaction among the involved groups of original and replicating experimenters. To do this, we proposed what we consider to be sufficient interaction, composed of an adaptation meeting to tailor the experiment to the replication setting, querying during replication operation and a combination meeting to combine the replication results with the outcomes of previous replications. The goal of keeping interaction down to a minimum is to encourage the running of replications.

To evaluate this proposal of interaction, we ran three replications, carried out by other experimenters in different settings. Thanks to the interaction, we were able to confirm results and identify new variables possibly (not) influencing the results. Additionally, the interaction helped us to identify deviations and problems that we would not have been aware of otherwise.

As regards the process, the adaptation and combination meetings played a key role and turned out to be indispensable. Adaptation meetings are essential for keeping the number of changes to the experiment to a minimum, and combination meetings identify or confirm possible causes for inconsistencies between results. Even though there are researchers that do not make use of the mechanism, the querying step should not be omitted because it is vital for resolving last-minute queries. However, as an improvement to the process, we suggest an obligatory meeting before the replication is run for experimenters to discuss last-minute issues.

Analyzing the problems and deviations that occurred during the replications, we discovered that most of them did not have anything to do with the interaction itself, but other issues, such as motivation of the replicating experimenters or misunderstandings among experimenters.

Finally, the results obtained are subjects to the threats to the validity of the study presented in Section 4: over-motivation of experimenters, and replications of the same experiment.

Acknowledgment

This work has been performed under research Grant TIN2011-23216 of the Spanish Ministry of Science and Innovation.

References

[1] V.R. Basili, S. Green, O. Laitenberger, F. Lanubile, F. Shull, S. Soerumgaard, M.V. Zelkowitz, The empirical investigation of perspective-based reading, Empirical Software Engineering 1 (2) (1996) 133–164.

[2] V.R. Basili, R.W. Selby, Comparing the Effectiveness of Software Testing Strategies, Department of Computer Science, University of Maryland. Technical Report TR-1501, College Park, May 1985.

[3] V.R. Basili, R.W. Selby, Comparing the effectiveness of software testing strategies, IEEE Transactions on Software Engineering SE-13 (12) (1987) 1278–1296.

[4] B. Beizer, Software Testing Techniques, second ed., International Thomson Computer Press, 1990.

[5] M. Ciolkowski, C. Differding, O. Laitenberger, J. Muench, Empirical investigation of perspective-based reading: a replicated experiment. ISERN Technical, Report, ISERN-97-13, 1997.

[6] O.S. Gómez, N. Juristo, S. Vegas, Replication types in experimental disciplines, in: International Symposium on Empirical Software Engineering and Measurement (ESEM'10), September 16–17, 2010, Bolzano, Italy, pp. 19–28.

[7] O. Gómez, N. Juristo, S. Vegas, Replication, reproduction and re-analysis: three ways for verifying experimental findings in software engineering, in: 1st International Workshop on Replication in Empirical Software Engineering Research (RESER'10), May 4, Cape Town, South, Africa, 2010.

[8] A. Jedlitschka, M. Ciolkowski, D. Pfahl, Reporting controlled experiments in software engineering, in: F. Shull, J. Singer, D. Sjoberg (Eds.), Guide to Advanced Empirical Software Engineering, Springer, 2008. Chapter 8.

[9] N. Juristo, A.M. Moreno, S. Vegas, Reviewing 25 years of testing technique experiments, Empirical Software Engineering 9 (1) (2004) 7–44.

[10] N. Juristo, S. Vegas, Functional testing, structural testing and code reading: What fault type do they each detect? Empirical Methods and Studies in Software Engineering- Experiences from ESERNET. Springer-Verlag, vol. 2785, 2003, pp. 235–261 (Chapter 12).

[11] N. Juristo, S. Vegas, The role of non-exact replications in software engineering experiments, Empirical Software Engineering 16 (3) (2011) 295–324.

[12] N. Juristo, S. Vegas. M. Solari, S. Abrahao, I. Ramos, Equivalence partitioning, branch testing and code review: comparing their effectiveness applied by subjects, in: Proceedings of the Fifth International Conference on Software Testing, Verification and Validation (ICST'12). Montreal, April 17–21, 2012.

[13] E. Kamsties, C. Lott, An empirical evaluation of three defect detection techniques, Technical Report ISERN 95–02, Dept. Computer Science, University of Kaiserslautern, May 1995.

[14] E. Kamsties, C.M. Lott, An empirical evaluation of three defect-detection techniques, in: Proceedings of the Fifth European Software Engineering Conference. Sitges, Spain, September 1995.

[15] O. Laitenberger, H.D. Rombach, (Quasi-)Experimental Studies in Industrial Settings: Lecture Notes on Empirical Software Engineering, World Scientific Publishing, 2003.

[16] R.C. Linger, Structured Programming: Theory and Practice (The Systems Programming Series), Addison-Wesley, 1979.

[17] J. Lung, J. Aranda, S.M. Easterbrook, G.V. Wilson, On the difficulty of replicating human subjects studies in software engineering, in: Proceedings of the 30th International Conference on Software Engineering (ICSE'08), May 10–18, 2008, Leipzig, Germany.

[18] J. Miller, Applying meta-analytical procedures to software engineering experiments, Journal of Systems and Software 54 (1) (2000) 29–39.

[19] G.J. Myers, T. Badgett, C. Sandler, The Art of Software Testing, Wiley-Interscience, Second edition, 2004.

[20] A. Porter, P. Johnson, Assessing software review meetings: results of a comparative analysis of two experimental studies, IEEE Transactions on Software Engineering 23 (3) (1997) 129–145.

[21] M. Roper, M. Wood, J. Miller, An empirical evaluation of defect detection techniques, Information and Software Technology 39 (1997) 763–775.

[22] P. Runeson, C. Andersson, T. Thelin, A. Amschler-Andrews, T. Berling, What do we know about defect detection methods?, IEEE Software 23 (3) (2006) 82–90

[23] F. Shull, J. Carver, G.H. Travassos, J.C. Maldonado, R. Conradi, V.R. Basili, Replicated studies: building a body of knowledge about software reading techniques, Lecture Notes on Empirical Software Engineering, World Scientific, 2003, pp. 39–84 (Chapter 2).

[24] F. Shull, M. Mendoça, V. Basili, J. Carver, J.C. Maldonado, S. Fabbri, G.H. Travassos, M.C. Ferreira, Knowledge-sharing issues in experimental software engineering, Empirical Software Engineering 9 (1–2) (2004) 111–137.

[25] D. Sjoberg, J. Hannay, O. Hansen, V. Kampenes, A. Karahasanovic, N. Liborg, A.C. Rekdal, A survey of controlled experiments in software engineering, IEEE Transactions on Software Engineering 31 (9) (2005) 733–753.

[26] M. Solari, S. Vegas, Classifying and analyzing replication packages for software engineering experimentation, in: Proceedings of the 4th International Workshop Series on Empirical Software Engineering (WSESE'06), Amsterdam, June, 2006, pp 19–24.

[27] S. Vegas, N. Juristo, A.M. Moreno, M. Solari, P. Letelier, Analysis of the Influence of Communication between Researchers on Experiment Replication, Proceedings of the International Symposium on Empirical Software Engineering (ISESE'06), Río de Janeiro, Brazil, September, 2006, pp. 28–37.

[28] M. Wood, M. Roper, A. Brooks, J. Miller, Comparing and combining software defect detection techniques: a replicated empirical study, in: Proceedings of the 6th European Software Engineering Conference, Zurich, Switzerland, September 1997.

[29] Available upon request from any of the authors involved in the replication.