# Design Patterns in Software Maintenace: An Experiment Replication at UPM

Experiences with the RESER'11 Joint Replication Project

Natalia Juristo, Sira Vegas
DLSIIS, Facultad de Informática
Universidad Politécnica de Madrid, Spain
{natalia, svegas}@fi.upm.es

***Abstract*—Replication of software engineering experiments is crucial for dealing with validity threats to experiments in this area. Even though the empirical software engineering community is aware of the importance of replication, the replication rate is still very low. The RESER'11 Joint Replication Project aims to tackle this problem by simultaneously running a series of several replications of the same experiment. In this article, we report the results of the replication run at the Universidad Politécnica de Madrid. Our results are inconsistent with the original experiment. However, we have identified possible causes for them. We also discuss our experiences (in terms of pros and cons) during the replication.**

***Keywords-empirical studies; experimentation; replication.***

## I. INTRODUCTION

The empirical software engineering (ESE) community is conscious of the importance of running replications. There are several reasons why replications of software engineering (SE) experiments should be run. Gómez *et al.* [6], for example, claim that the purpose of replication in SE is to address the different types of validity threats (internal, external, conclusion and construct) to an experiment. On the other hand, Dieste et al. [4] argue that replications have to be run to increase the reliability of the results of experiments with few experimental subjects.

Even so, the replication rate in SE is low [8]. This can be put down to the obstacles that replicating researchers come up against when they embark upon a replication. The most frequently cited problems [8] include the high cost of running a replication, the shortage of detailed information to run the replication or a impossibility to publish the results of replications.

There have been earlier experiences of running replications in SE. However, none are equal to the ambitions of the RESER'11 Joint Replication Project. In our view, this experience is different in several respects:

- No large-scale replications of experiments have ever been run to date. Even the most successful experiments in terms of number of replications have been replicated no more than twenty times.
- The replications run to date were run sequentially. There are no reports of several replications being run (quasi) simultaneously.
- As the replications were run sequentially, the experimental conditions were changed as a result of the evolution of the experiment or the different validity threats. In this case, the experiment to be replicated is the same.

On all these grounds, the RESER'11 approach looks promising.

This paper reports the experiences of running one of the replications for RESER'11, specifically at the Universidad Politécnica de Madrid (UPM). For this purpose, it is organized as follows. Section 2 discusses work related to experiment replication. Section 3 describes the replication run. Section 4 illustrates the experiences gathered as the replication was run in terms of the problems encountered. Section 5 presents the lessons learned. Finally, Section 6 outlines the conclusions of the research.

## II. RELATED WORK: PREVOUS EXPERIENCE

Other researchers have reported problems related to running replications (see for example [10] or [13]). In this article, however, we focus specifically on our experience of running experiment replications in the past. This way, we compare the experience reported here with the complications and problems that we detected in earlier replications.

Our first experience of replication dates back to 2000, when we replicated the experiments run by Kamsties and Lott [9] and Woods et al. [12], which are, in turn, replications of an original experiment run by Basili and Selby [1]. This experiment studies the relative effectiveness and efficiency of three code evaluation techniques: a white box technique, a black box technique and a code review technique. The experiences of this replication were reported in [14].

As a result of the problems encountered, we started to research the topic of replication. This line of research addresses several issues, like understanding the concept of replication [5], [6], formalizing the process of replication [7], [8], or examining how much interaction there should be among groups of researchers when running a replication [14].

This research materialized as a result of the performance of several replications of the above-mentioned experiment in partnership with researchers from other institutions. To date, we have 12 replications of the experiment run at the Universidad Politécnica de Madrid, two replications at the

978-0-7695-4674-2/12 $26.00 © 2012 IEEE
DOI 10.1109/RESER.2011.8

Universidad de Valencia, two replications at Universidad ORT-Uruguay, and one replication at the Universidad de Sevilla, totaling 17 replications of the experiment. Additionally, we are working on replicating more experiments to try to generalize our previous findings.

Thanks to the replications that we have run, we have been able to form a fairly clear idea of the difficulties and problems involved in a replication, which can be summed up as documentation and interaction problems.

The d**ocumentation problems** we have previously experienced are

- In order to understand the experiment, replicators are referred to publications about the experiment. These publications are not specific for replication purposes, but usually are journal or conference papers. But additional details of the experiment are needed when replicating, including for example, a justification for the design/analysis decisions that have been made. As a consequence, the replicating researchers usually lack of enough information to run the replication.
- Very often, it is necessary to reanalyze the data of the original experiment to compare results with the replication. Therefore, raw data needs to be available.
- The replicators lack of guidelines with the necessary steps to run the replication. Every experiment is different and in order to properly replicate it, it is helpful for the replicator to know exactly how (s)he has to proceed. These guidelines should include, among other issues, specific instructions of how to proceed during the experimental operation (e.g. if questions from experimental subjects can be answered, if there should be a previous session in the experiment where its dynamics is presented to the subjects, if there is specific instructions to how to fill in data collection forms, etc.).
- Many experiments in SE involve subjects performing a task, which correctness has to be later examined by the experimenters (detecting faults in software, checking whether a given technique has been properly applied, etc.). Therefore, it is necessary to provide the replicators with a *gold* solution, so that the outcomes of the experiment and replication are measured in the same terms.

The **interaction problems** we have previously experienced consist of:

- Replicating researchers, when left on their own, tend to make unnecessary changes to the original experiment when running *strict* replications. Original experimenters and replicating should meet to guarantee that the replication is really *strict.*
- Impossibility of aggregating the results of the original experiment and replication because of lack of enough knowledge of both contexts. Due to the lack of knowledge of the contextual variables influencing a SE experiment, it is essential the participation of both groups of researchers in the aggregation. Merging their partials views of the context of the experiment/replication is the only way to successfully aggregate the results.

Additionally we have noticed that:

- There is a lack of motivation to run replications in SE because they usually are very costly, in terms of resources, and effort from experimenters.
- Unforeseen events (or incidents) are common in SE experiments. Unfortunately, we have not been able to devise another solution for this issue, but to keep track of them. It is essential to know them while the aggregation of results.
- Special attention has to be taken to get that experimental subjects have the same knowledge in the original experiment and replication. It is very easy to shift to a different population in a replication where this was not intended.

Again we have learned a number of lessons from this series of replications:

- There needs to be an infrastructure (usually known in ESE as replication package or laboratory package) containing all the material of interest for running the experiment (detailed description of the original experiment, etc.). Replicators have to know what they are doing to be able to run a replication.
- Not even the most comprehensive replication package will contain all the information required to run the replication. This is a notorious problem in experiment replication in other fields apart from SE and is known as the tacit knowledge problem [13].
- It is very hard, if not impossible, to find two contexts sharing exactly the same characteristics. In most cases, some aspects of the experiment will have to be modified to adapt it to the contextual characteristics of a new site.
- Researchers participating in the experiment and its replications should meet to aggregate results.

## III. Description of the Replication

The replication is described along the lines in [3].

### A. Original Study

The original study was performed in 1997 by Prechelt *et al.*, and is reported in [11]. It investigates the impact of design patterns on software maintenance. We will refer to this work hereinafter as PatMain.

- *Research question(s)*: Do research patterns improve the performance of subjects doing a maintenance exercise?
- *Participants:* 29 professional software engineers. Before the experiment, the participants had little pattern experience. About half of them had no pattern experience at all.
- *Dependent and independent variables:* The design uses three independent variables: programs and change tasks, program version, and amount of pattern knowledge. The dependent variables are: time (in minutes) to complete each maintenance task, and correctness of the task.
- *Artifacts:* There are four different programs, each with different design patterns. Each program has two or three different maintenance tasks. There are two different, functionally equivalent versions of each program: pattern version (employs one or more design patterns),

and alternative version (uses fewer design patterns or simplified versions of the patterns).

- *Context variables:* Experiment run on paper.
- *Design:* It follows a within-subject design, with a pretest (without pattern knowledge) and a posttest (with pattern knowledge) stage where subjects are divided into four groups. In each stage, each group maintained a different version of two different programs in a different order. Overall, each subject worked on all four programs.
- *Experimental operation.* The experiment is run in four sessions, lasting two full days: first day, pretest plus first part of pattern course, second day, second part of pattern course and posttest.
- *Summary of the results:* (1) It is usually, but not always, useful to use a design pattern if there are simpler alternatives. (2) Software engineering common sense should be used to find the exceptions where a simpler solution should be preferred, even if a design pattern solution could easily be applied. (3) Even where this common sense suggests that using a pattern might not be a good idea, it is sometimes right to use it. (4) A thorough understanding of specific design patterns is often helpful for program maintenance, even if these programs are neither very large nor very complicated.

### B. Other Replications

PatMain was then replicated by researchers at Simula with 44 paid professional subjects from various consultancy companies [15]. This replication was nearly identical in its setup, except that the subjects worked in a real programming environment.

### C. UPM Replication

The UPM replication was run within the framework established by the RESER'11 Joint Replication Project throughout April 2011. The details for the replication can be found in [16].

- *Motivation for conducting the replication*. The arguments supporting the RESER'11 Joint Replication Project described in the introduction of this paper.
- *Level of communication (documentation and interaction) with original experimenters*. We were referred to the journal paper where the original experiment is described [11] and web page where the replication is explained [16]. Interaction is implemented through a specific mailing list created for the Joint Replication Project.
- *Description of the replication* (in terms of changes to the original experiment):
  - *Research question(s)*: Same as original experiment.
  - *Participants:* 8 master students. Before the experiment, all the participants, but one, have some pattern experience. They all have solid experience in JAVA.
  - *Dependent and independent variables:* The design uses only two independent variables: programs and change tasks, and program version. Amount of pattern knowledge is not used for this replication. The dependent variable is time (in minutes) to complete each maintenance task. Task correctness is not explored.
  - *Artifacts:* Only two out of the four programs used in the original experiment are used in the replication. The remaining characteristics of artifacts are the same as in the original experiment. JAVA and C++ versions of programs are available.
  - *Context variables:* The replication is run using the tool specifically designed for this purpose [17], [18]. The replication is fully automated. Subjects have to implement solutions on a computer.
  - *Design:* This applies to the pretest stage only. The rest of the design is the same as in the original experiment.
  - *Experimental operation.* The subjects did the experiment asynchronously, although they all completed the replication within four days.
- *Results obtained*. TABLE I shows the raw data of the replication. It shows only the relevant data for the results reported in this paper.

  It is worth mentioning that both subjects of group A failed to complete the replication. Subject 92689 missed the internet connection in the last submit (where other comments were asked), but all the data was lost. Subject 94345 abandoned the replication because of lack of patterns knowledge.

  Figure 1 shows the professional experience of subjects measured in number of years. Figure 2 shows the experience subjects have with JAVA, in terms of lines of code (LOC) written. From these figures, we can see that groups are not balanced. Subjects in group D have much more professional experience than subjects in other groups, while subjects in group C have more experience with JAVA. Subjects in group B reported not have written any single line of code in JAVA. This suggests a possible error in the data.

TABLE I. RAW DATA OF THE REPLICATION.

| Group | User ID | LOC written in JAVA | Years of professional experience | Abstract Factory | Composite | Decorator | Time communication task 1 | Time communication task 2 | Time graphics task 1 | Time graphics task 2 |
|---|---|---|---|---|---|---|---|---|---|---|
| **A** | 92689 | - | - | - | - | - | - | - | - | - |
| **($CO_A$-$GL_P$)** | 94345 | - | - | - | - | - | - | - | - | - |
| **B** | 10354 | 0 | 0 | 3 | 3 | 3 | 40.63 | 0.00 | 0.03 | 10.32 |
| **($GL_A$-$CO_P$)** | 15350 | 0 | 4 | 2 | 2 | 2 | 25.40 | 15.78 | 24.60 | 10.65 |
| **C** | 22591 | 10000 | 0 | 3 | 2 | 2 | 29.92 | 36.97 | 23.48 | 7.52 |
| **($CO_P$-$GL_A$)** | 23719 | 60000 | 1 | 4 | 4 | 4 | 27.67 | 16.38 | 19.92 | 11.30 |
| **D** | 38048 | 3000 | 3 | 2 | 2 | 2 | 12.30 | 5.62 | 0.08 | 10.28 |
| **($GL_P$-$CO_A$)** | 80744 | 5000 | 10 | 2 | 1 | 1 | 37.27 | 5.72 | 0.13 | 7.67 |

Interpretation of pattern columns: 1: Never heard of it, 2: Have only heard of it, 3: Understand it roughly, 4: Understand it well

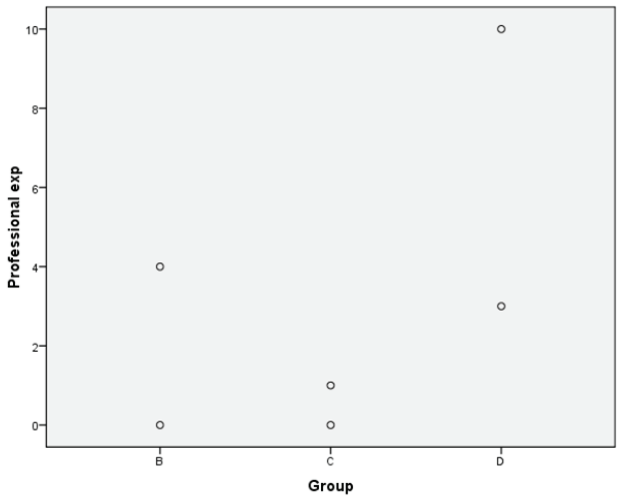


Figure 1. Years of professional experience of subjects per group.

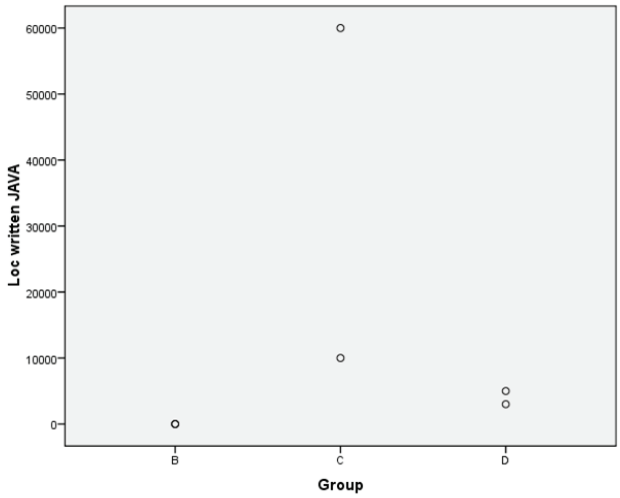


Figure 2. LOC written in JAVA by subjects per group.

According to TABLE I, subjects have only theoretical experience with patterns. The knowledge is varied, although have subjects have good pattern knowledge. However, they have never used them in a real environment.

For data analysis purposes, we run an analysis of variance (ANOVA). The model is the same as in the original experiment, but removing those variables that do not apply to this replication. Therefore, the variables in the model are: ID, order, task and task*version.

The residuals meet the normality criterion required by the ANOVA, as shows Figure 3. However, they do not meet the homoscedasticity criterion, as shows the funnel-shaped Figure 4. Some subjects reported that they had interruptions while performing the tasks. This could be the reason for which variance increases with mean. However, several authors report that F-test is robust to lack of homoscedasticity. In any case, we will take this as a limitation for the results obtained.

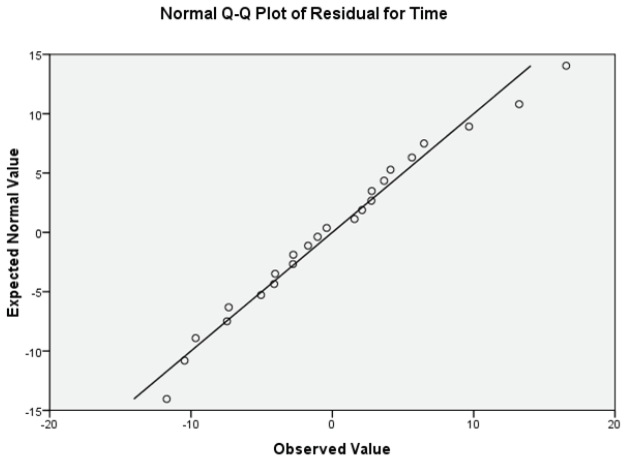


Figure 3. Normal probability plot of residuals.

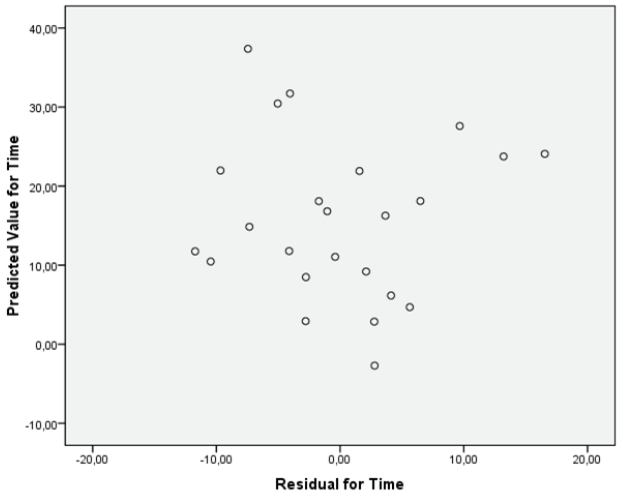


Figure 4. Dispersion diagram of residuals against predicted values.

TABLE II shows that only Task is significant (p-value of 0.043). But most interesting, it shows that the model we are using is not valid, as its significance is above 0.05 (p-value of 0.167). This means that the model we are using does not explain the variability in the replication. There are variables that are having an influence in the response variable and we are missing them. Additionally, the observed power for the model is low (p-value of 0.529). This could be another reason for which the model is not significant. We have very few data points, and we are missing data from one whole group.

With respect to the tasks, Tukey's multiple comparison test indicates that Task 1 in communication channels is significantly different from Task 2 in graphics library, taking more time to subjects Task 1 of communication channels.

TABLE II. TESTS OF BETWEEN-SUBJECTS EFFECTS.

| Source | Type III Sum of Squares | df | Mean Square | F | Sig. | Observed Power[b] |
|---|---|---|---|---|---|---|
| Corrected Model | 2365.644[a] | 12 | 197.137 | 1.812 | 0.167 | 0.529 |
| Intercept | 5454.043 | 1 | 5454.043 | 50.138 | 0.000 | 1.000 |
| ID | 339.075 | 4 | 84.769 | 0.779 | 0.561 | 0.179 |
| Task | 1239.603 | 3 | 413.201 | 3.798 | 0.043 | 0.654 |
| Order | 9.781 | 1 | 9.781 | 0.090 | 0.770 | 0.059 |
| Task * Version | 191.962 | 3 | 63.987 | 0.588 | 0.635 | 0.135 |
| Error | 1196.589 | 11 | 108.781 | | | |
| Total | 9567.505 | 24 | | | | |
| Corrected Total | 3562.233 | 23 | | | | |

a. R Squared = 0.664 (Adjusted R Squared = 0.298)
b. Computed using alpha = 0.05

Finally, Figure 5 shows the results per task and version. Although this interaction has turned out not to be statistically significant, we wanted to check its trend against the original experiment. Tasks GL2 and CO2, follow the same trend as in the original experiment. However, this is not the case for tasks GL1 and CO1.

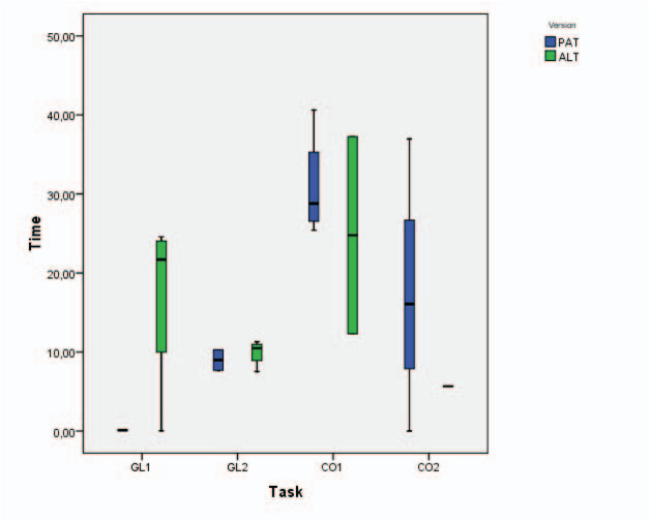


Figure 5. Results per task and program version.

### D. *Comparison of Results to Original*

Our results are not consistent with the original experiment. However, there are several variables that could be having an influence on the results. Some of them are related to changes in the original experiment (programming language, and subject profile). Others are related to the nature of our replication (few data points). Finally, others are related to incidences that occurred during the replication (data from one group is missing, and in some cases, the reported times are not accurate, as the subjects had interruptions while doing the tasks).

### E. *Drawing Conclusions across Studies*

Although we intended to run a strict replication, it seems we have not been able to get it. There have been some changes that could provoke differences in the results. Additionally, the data suffers from important threats to its validity (inaccuracies and missing data points).

## IV. EXPERIENCES WITH THE REPLICATION

Clearly, this replication has a number of **benefits** over previous replications in which we have participated. Most of these pluses are related to the use of the replication portal:

- **B1**: Few resources are required of the replicating researchers. The time it takes to do the replication is reduced practically to the duration of the experimental session. The subjects do not need any specialized knowledge apart from the programming language to be used. No computer suites or computers are necessary. Subjects can do the experiment at home using their own computer, as all they need is Internet access.
- **B2**: The replication is easy for the experimenter to run. The experiment operation is fully automated thanks to the portal. Therefore, the experimenter is relieved of preparing the material for the experimental operation. This task is usually with one of the heaviest workload in an experiment.
- **B3**: No changes are allowed. This stops the replicating researchers from making unnecessary changes to the experiment. It provides some control over the replication.
- **B4**: Data collection forms are automated. This should avoid possible problems of how to fill them in.

However, we also ran into some trouble during the replication. In the following, we describe the problems encountered by both the replicating researchers and the experimental subjects. The problems encountered by the replicating researchers have been typed as follows: problems related to the documentation used in the replication and interaction problems.

The **documentation problems** that we identified are as follows:

- **D1**: The replicating researcher is referred to a journal article to learn about the original experiment.
  Such publications have space limits, making some aspects of the original experiment, such as the experimental design or analysis, unclear.
- **D2**: The replicating researcher is referred to a web site to look up the description of the replication, where there is a no more than half-page description of the changes made to the original experiment.
  The replication is not described in enough detail, having some information unavailable, such as what the design groups are, or what happened with Communications channel task3. Additionally, it makes some unjustified decisions, such as why there have to be four groups of subjects, why a new programming language is added or why subjects implement solutions on a computer, unclear.
- **D3**: The guidelines stating the steps that the replicator has to take in order to perform the replication are confined to replication portal user instructions.
  These guidelines are not enough to run the replication, making it unclear what information or instructions on how to perform the experiment should (or should not) be given to the subjects before the experiment.
- **D4**: There are no guidelines concerning the correctness of tasks performed by the experimental subjects.
  One of the response variables examined in the original experiment was the correctness of the solutions proposed by the subjects. This variable has three values (high/medium/low). It was unclear how to rate the solutions proposed the subjects, and therefore this response variable was not examined.
- **D5**: There are no guidelines about how the data should be analyzed. Should we run again an ANOVA? What should be the model?
  The replication involves the analysis of the gathered data. It was unclear what the best type of analysis was.

Additionally, we identified the following **interaction problems**. They could all be summed up in one point, namely, the lack of direct interaction between the replicating and the original researchers

- **I1**: The synthesis of the results is an essential part of running a replication. One part is to compare the results of the replication with the results output in the original experiment to determine whether or not they are consistent. The other is to analyze, irrespective of the whether or not there are inconsistencies, the changes made to the replication to determine which might have and which might not have had an effect on the replication results. A meeting should be held with the original experimenters, who are the people that are best acquainted with the results of the original experiment.
- **I2**: It is impossible to identify what, save the variables that represent changes to the original experiment, could be influencing the results.

Finally, the **experimental subjects** identified the following **problems**:

- **S1**: Loss of Internet connection
  This happened to one of the subjects towards the end of the experiment. He was unable to complete the last step, and there is no provision for retrieving the saved state of the experiment if disconnected.
- **S2**: Some pattern knowledge is required
  One subject was unable to do the replication and another person had difficulty on the grounds of their pattern knowledge.
- **S3**: Difficulty with understanding the tasks
  Several subjects reported problems related to English language proficiency. One person reported having had trouble understanding a task, which was unrelated to his English language proficiency.
- **S4**: Interruptions in experimental task performance
  The subjects reported two sources of wasted time during experiment performance. One was that nobody had Eclipse (most use NetBeans); the other was that they experienced some sort of external interruption. The web portal automatically records times, which the subjects cannot modify, leading to inaccurate measurements in almost all cases.

## V. LESSONS LEARNED

We analyze each of the benefits and problems identified in the last section and how they could have been optimized. If we have experienced a difficulty previously, we explain how we propose to solve the problem.

As regards the **benefits**:

- **B1, B2:** Although having a tool that automates part of the experimenter's workload does not solve any of the problems we identified earlier; it does address comments or complaints raised by replicating researchers with whom we have worked. This could be an incentive for replication and help to increase the rate of replication in SE.
- **B3:** The provision of a set replication beforehand (irrespective of whether it or not it offers options) can prevent a problem that we have met before. Replicating researchers sometimes make unnecessary changes to the original experiment when they run the replication.
- **B4:** Having a tool that automates the experimental operation is a way of tackling a problem that we have met before. Subjects are sometimes unsure about how to fill in the forms to be completed during the experiment and experimenters are on occasions unsure about the order in which to hand out the experimental material.

As regards the **documentation problems**:

- **D1**: The need to report the original experiment in detail for the purposes of replication is a problem that we have met before. The solution that we proposed is to describe the original experiment in a self-contained document without a page limit, specifying all the details of the experiment.
- **D2**: The need to report the replication in detail is not a problem that we have met before, because, as a matter of course, we drafted a self-contained document fully describing the replication (it is not necessary to read the aims and scope of the original experiment to understand

the aims and scope of the replication) and specifying where it differs from the original experiment.

- **D3**: The need for guidelines containing the steps to be taken by the replicator to run the replication is a problem that we have met before. The solution that we proposed is to draft a document containing these instructions. Drafting such detailed guidelines is not, by any means, a straightforward task, as illustrated by the fact that the problem recurred in later replications, despite the use of such guidelines.
- **D4**: The need for a correct solution to each experimental task is a problem that we have met before. The solution that we proposed to this problem is to have the original experimenters provide a correct solution for use as a benchmark or guideline for the replicating researchers.
- **D5**: The lack of guidelines on how to conduct the analysis is not a problem that we have had to deal with before. When we played the role of replicators, we managed to conduct the analysis based on the available information about the original experiment. When other researchers have replicated one of our experiments, we have analyzed the data.

As regards **interaction problems**:

- **I1, I2**: The impossibility of aggregating the results because we know absolutely nothing about the context in which either the original experiment or the replication was run is a problem that we have met before. To solve this problem, the replicating researchers and original researchers met after running the replication.

Finally, as regards the **problems encountered by the experimental subjects**:

- **S1, S3, S4**: We have experienced similar problems in the past. They refer to unforeseen events that take place unexpectedly in the replication and are, of course, hard to solve. The only solution that we can think of is for there to be a constant communication channel between replicators and original experimenters in case any of these problems could be solved on the fly.
- **S2**: We have experienced problems related to what subjects need to know to be able to run the replication properly and how to acquire this knowledge in the past. We have discovered that this is a far from trivial problem and is not easy to solve. We have proposed that this problem should be discussed specifically during conversations between the original replicators and the replicating researchers.

Looking at the problems identified during the replication it is clear that we have not come across any problem that: either we have not met before when running a replication, or it was already addressed by the replication in a natural way. In other words, replication always appears to come up against the same problems.

There is a series of recurrent problems for which solutions have already been identified, but which, for some reason, we in the ESE community apparently still try to ignore, since nothing has been done to remedy the difficulties.

It is a shame that the experiment chosen for replication was not mature enough to prevent these problems from occurring. Otherwise, we could have discovered new problems of real interest to the community.

## VI. CONCLUSIONS

The RESER'11 Joint Replication Project aims to tackle some of the most pressing problems in ESE today, such as the low rate of replication. This paper reports the replication that was run at the Technical University of Madrid.

The paper reports the results of the replication, and our experience with the replication.

In our view, the RESER replication came up against problems that are far from new in SE replications, ranging from a deficient one-to-one relationship between the original experimenter and the replicator to the shortage of proper documentation to support the replication.

## ACKNOWLEDGMENT

This work was funded by Spanish Ministry of Science and Innovation research grant TIN2008-00555.

## REFERENCES

[1] V.R. Basili, R.W. Selby. Comparing the Effectiveness of Software Testing Strategies. *IEEE Transactions on Software Engineering*. SE-13(12):1278–1296. 1987.

[2] V. Basili, F. Shull, F. Lanubile. Building Knowledge through Families of Experiments. *IEEE Transactions on Software Engineering*. 25(4): 456-473. 1999.

[3] J. Carver. Towards Reporting Guidelines for Experimental Replications: A Proposal. *1st International Workshop on Replication in Empirical Software Engineering Research (RESER'10)*. May 4. Cape Town, South Africa. 2010.

[4] O. Dieste, E. Fernández, R. García, N. Juristo. Hidden Evidence Behind Useless Replications. *1st International Workshop on Replication in Empirical Software Engineering Research (RESER'10)*. May 4. Cape Town, South Africa. 2010.

[5] O.S. Gómez, N. Juristo, S. Vegas. Replication Types in Experimental Disciplines. *International Symposium on Empirical Software Engineering and Measurement (ESEM'10)*. September 16-17. Bolzano, Italy.

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

[7] N. Juristo, S. Vegas. Using Differences among Replications of Software Engineering Experiments to Gain Knowledge. *3rd International Symposium on Empirical Software Engineering and Measurement (ESEM'09)*. Orlando, USA. 2009.

[8] N. Juristo, S. Vegas. The Role of Non-Exact Replications in Software Engineering Experiments. *Empirical Software Engineering*. 16(3):295–324. 2011.

[9] E. Kamsties, C.M. Lott. An Empirical Evaluation of Three Defect-Detection Techniques. *Proceedings of the Fifth European Software Engineering Conference (ESE'95)*. Sitges, Spain. 1995.

[10] M.G. Mendonça, J.C. Maldonado, M.C.F. de Oñoveroa, K.C.S. Fabbri, F. Shull, G.H. Travassos, E.N. Höhn, V.R. Basili. A Framework for Software Engineering Experimental Replications. In *Proceedings of the 13th International Conference on Engineering of Complex Computer Systems*. Belfast, Northern Ireland. 2008.

[11] L. Prechelt, B. Unger, W.F. Tichy, P. Brössler, L.G. Votta. A Controlled Experiment in Maintenance Comparing Design Patterns to

Simpler Solutions. *IEEE Transactions on Software Engineering*. 27(12):1134-1144, December 2001.

[12] M. Roper, M. Wood, J. Miller. An Empirical Evaluation of Defect Detection Techniques. Information Software Technology. (39):763–775. 1997.

[13] F. Shull, M.G. Mendoncça, V. Basili, J. Carver, J.C. Maldonado. Knowledge-Sharing Issues in Experimental Software Engineering *Empirical Software Engineering*. 9(1-2):111-137. 2004.

[14] 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)*. pp 28–37, 2006, Rio de Janeiro, Brazil.

[15] M. Vokáč, W. Tichy, D.I.K. SjØberg, E. Arisholm, M. Aldrin. A Controlled Experiment Comparing the Maintainability of Programs Designed with and without Design Patterns - A Replication in a Real Programming Environment. *Empirical Software Engineering*. 9(3):149-195, 2004

[16] https://www.inf.fu-berlin.de/w/SE/PatmainReplicationInfo

[17] http://replication.inf.fu-berlin.de/replication/explogin

[18] http://replication.inf.fu-berlin.de/replication/userlogin