Automated Support for Controlled Experiments in Software Engineering: A Systematic Review

Marília Freire,[a,b] Daniel Costa,[a] Edmilson Campos,[a, b] Tainá Medeiros,[a]
Uirá Kulesza,[a] Eduardo Aranha,[a] Sérgio Soares,[c]
[a] Federal University of Rio Grande do Norte, Campus Universitário, 59.078-970, Natal-RN, Brasil
[b] Federal Institute of Rio Grande do Norte, Av. Salgado Filho, 1559, Tirol, 59015-000, Natal-RN, Brasil
[c] Federal University of Pernambuco, Av. Prof. Moraes Rego, 1235, C. Universitária, 50670-901, Recife-PE, Brasil
{marilia.freire, daniel.calencar , edmilsoncampos, tainamedeiros}@ppgsc.ufrn.br, {eduardo, uira}@dimap.ufrn.br,
scbs@cin.ufpe.br

***Abstract— Context*: There is an increasing need to perform controlled experiments in software engineering. *Objective*: This systematic review (SR) shows the current state of the art on the tools and infrastructures that provide automated support for controlled experiments in software engineering. *Method*: We performed manual searches in journals and conferences proceedings for papers describing supporting tools and environments to conduct controlled experiments. *Results*: We found and reviewed 25 primary studies according to inclusion and exclusion criteria, resulting in 15 relevant studies. *Conclusion*: There are few supporting environments for conducting controlled experiments, despite of the increasing demand for this kind of study in software engineering. We also highlight many limitations of these tools, which configures great potential for future research.**

Experimental software engineering; automated support; controlled experiment.

I. INTRODUCTION

In recent years, the software engineering research community has given more attention and importance to the development and reporting of experimental studies, considering that simple proof of concept is no longer acceptable in the assessment of new proposed methods. One important type of empirical study for the research in software engineering is the controlled experiment [1].

A controlled experiment allows testing research hypotheses and cause and effect relationship between variables involved in the study. The process of a controlled experiment is typically composed of the following phases: definition, planning, execution, analysis, and packaging [2]. Each one of these phases is associated with the execution of different activities that consume and/or produce artifacts related to the experiment. Controlled experiments require great care in planning so they can provide useful and significant results [3]. However, the process of planning, conducting, and reporting the various activities involved in a controlled experiment is very complex.

Despite the growing need to run controlled experiments in software engineering, their development is still very complex. Furthermore, controlled studies need to be replicated because a single controlled experiment may be insufficient and their results are limited in terms of conclusions' generalization [4]. Conduction and replication of large-scale experimental SE studies is even more complex. One factor that contributes to that is the lack of automated and integrated tools supporting the experiment process phases.

This paper focus in presenting the results of a systematic review [5] [6] that analyzes the current state of the art on the tools and infrastructures developed to provide automated support to conduct controlled experiments in software engineering. Our study has found seven tools specifically developed or adapted for conducting experiments in software engineering. Our systematic review aims to provide findings to appropriately points out new research efforts and opportunities related to development of automated support for the software engineering experimental process.

The remainder of this paper is organized as follows. Section II explains our review method. Section III presents our results..The limitations of our study as stated in Section IV. Section VI presents some discussions based on the data we got in our systematic review, and, finally, Section VII presents the conclusions and possible future works.

II. REVIEW METHOD

This section describes the protocol used to conduct this systematic review, which was defined based on specific guidelines [6]. The process was performed in the early 2012. Due to space restrictions some contents related to the protocol is not presented in this paper, for additional details please refer to: http://bit.ly/10zA6FY.

A. Research Questions

Our research was guided by questions about the empirical support tools and infrastructures that provide (semi) automated support to the phases of controlled experiments in software engineering. The four main research questions (RQ) investigated in the systematic review were:

- **RQ.1** What tools have been proposed to support controlled experiments in software engineering?
- **RQ.2** Which stages of controlled experiments the proposed tools are supporting?
- **RQ.3** What are the main features supported by the proposed tools?
- **RQ.4** Who has been developing tools to support controlled experiments? When?

B. Inclusion/exclusion criteria

The inclusion/exclusion criteria defined to assess the primary studies in our systematic review were: (i) only studies written in English were considered; (ii) studies that did not present supporting environments for conducting controlled experiments in software engineering were excluded; and (iii) only studies that were related to specifics domains of SE, such as software testing, were included; (iv) studies that had insufficient information to perform the quality evaluation stage were also discarded.

C. Decision Procedure

In a systematic review, it is important to define how to solve possible conflicting situations. These conflicts may happen during the study selection, quality evaluation, or data extraction. Therefore, we defined our process to support decision-making and consensus as follows: two members from our team read selected studies assigned in a random way. Any disagreement among researchers was resolved by a third reviewer (one professor).

D. Data Sources and Search Strategy

Our search strategy was established in two steps: (i) a manual search across the main publication vehicles in the Experimental Software Engineering area; and (ii) a reference search where the reference section of all selected primary studies were analyzed searching for new research work related to our systematic review questions. For the first step the following conferences and journals were considered: Journal of Empirical Software Engineering, Journal of Systems and Software, ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM) (and the previous METRICS and ISESE), Experimental Software Engineering Latin American Workshop (ESELAW), and International Conference on Software Engineering (ICSE). It is important to say that for publication sources as ICSE, Journal of Empirical Software Engineering, and Journal of Systems and Software, we limited the search period from January/2002 to December/2011, because these events are already made long ago and we would not have time to check them out. However, for the analysis of referred papers we do not consider any restriction regarding publication venues and dates.

E. Study Selection

The study selection process was realized in two steps as follows: (i) titles and abstracts of papers found during the manual search were read and irrelevant papers were removed; (ii) the complete reading of the selected primary studies was performed to assess whether they address the inclusion/exclusion criteria.

F. Data Extraction Process and Synthesis

As usual, each research question motivated some data extraction (see Table I). In addition, general information was extracted from the studies, such as title, authors, publisher, and publication year. The data extraction was also performed with the aid of spreadsheets containing forms to extract portions of the selected studies.

TABLE I. EXTRACTED DATA

Research Question	Attribute	Data
RQ.1	Tool	study title, tool name, origin (academic or industrial), tools compared
RQ.2	Process	study stages supported
RQ.3	Features	functionalities supported
RQ.4	Mapping	first author's affiliation and country

III. RESULST AND FINDINGS

In this section we show the results of our study addressing the research questions presented in Section II. Our systematic review was interested in primary studies that present (semi) automated solutions to support software engineering experiments.

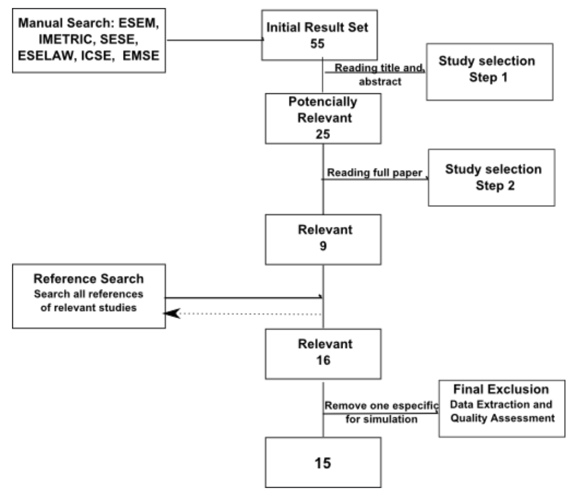

Figure 1. Studies Search and Selection Process

Figure 1 illustrates the systematic review process showing the primary studies that were found and selected. During the manual search 55 papers were selected from the venues previously specified. For this selection, the titles and abstracts of the paper studies were read resulting in 25 papers. These 25 papers were completely read in order to discard irrelevant studies. After this selection, we had 9 studies. After that, a search on the reference sections of the papers was accomplished aiming the selection of other relevant studies. During this process we found other 7 primary studies. In the final step we removed studies that were not specific to a single domain of SE. The final number of relevant studies was 15.

During the search on the reference sections, we found papers from other conferences: IEEE TSE (IEEE Transactions on Software Engineering), Advances in Computing Journal, NJC (Nordic Journal of Computing), NWPE (Nordic Workshop on Programming and Software Development Tools and Techniques), JIISIC (Jornadas IberoAmericanas en Ingenieria del Software e Ingenieria del Conocimiento), and ICECCS (International Conference on Electronics, Communication and Computer Science). The results found for each research question are discussed in the following sections.

A. *Tools to support controlled experiments (RQ.1)*

The following seven environments – tools, infrastructure – were found: Simula Experiment Support Environment (SESE), experimental Software Engineering Environment (eSEE), value-based empirical research (VBER), Ginger2, Experiment Manager framework, Framework for Improving the Replication of Experiments (FIR), and Mechanical Turk. Following, we have a short summary of the seven tools and the respective ID according to the Table 2:

SESE [7] [8] [9] [10]: It is a web-based tool that supports participants management, capturing the time spent during the experiment, enable the work product collection, and participants activities monitoring. Its weaknesses are the data collection and analysis (PS1, PS2, PS3, PS4).

eSEE [11] [12] [13] [14] [15] [16]: An environment to manage several kinds of empirical software engineering studies. It works in three levels of knowledge organization about the experimentation process: knowledge for any kind of experimental study (meta level), knowledge for each experimental study (configuration level), and knowledge for a specific experimental study (instance level). It has a prototype and an initial set of tools to populate the eSEE infrastructure has been built (PS5, PS6, PS7, PS8, PS9, PS10).

VBER [17]: A value-based framework to the planning phase. It assists the stakeholders to compare the benefits and risks of potential empirical study variants (PS3).

Mechanical Turk [18]: A crowdsourcing tool adapted to support empirical studies in the experimental software engineering context. It offers facilities to access and manage a large pool of study participants and enables recruiting the right type and number of subjects to assess a software engineering technique or tool (PS11).

Ginger2 [19]*:* An experimental environment constructed based on the CAESE framework. Although the CAESE covers the complete processes, Ginger2 is restricted to the execution and analysis phase. Its strength is the variety of low-level detail collected (PS15).

FIRE [20]: A Framework for Improving Replication of Experiments that focuses on knowledge sharing issues to allow cooperation between research groups. Fire is a framework with seven steps that assumes researchers are collaborating closely using a variety of communication mechanisms. Its weakness is that it gives only a conceptual support (PS13).

Experiment Manager Framework [21]: This framework is an integrated set of tools to support software engineering experiments. It was used only in high performance computing (HPC) experiments. It helps the subjects by applying heuristics to infer programmer activities. Its analysis tools are simple (PS14). Table 2 presents the 15 studies included in this review, including the study ID, title, publisher year, name of related supporting environment (tool, framework or infrastructure), university or industry that developed the environment, and the publisher source.

For the studies that are part of a collaborative work we fill in the table with the first author information. Another point to clarify is that the Feedback-collecting tool described in PS2 is implemented as part of the web-based Simula Experiment Support Environment (SESE).

Although PS15 was published in 1999, we included it in our revision after the reference search because it presents the Ginger2 that is a tool based on the CAESE (Computer-Aided Empirical Software Engineering) framework. This framework defines a complete solution for conducting experiments but, according to the study, Ginger2 have been implemented to only support the data collection and analysis stages. A major feature of these tools is allowing the collection of several empirical data as mouse clicks and keystrokes, eye traces, three-dimensional movement, skin resistance level, and video-taped data. We were not able to find more details of the current development stage of these solutions. The eSEE tool is the only one that reports a web site, although we were not able to find out updated information from it.

B. *Supported Stages of Controlled Experiments (RQ.2)*

This question aims to point the main experimental process stages that are supported by the investigated tools. A typical controlled experiment process has the following stages [2]:

1) Definition: In this phase the study has to be characterized in terms of problem, objective and goals. It determines the foundation for the experiment.

2) Planning: It prepares for how the experiment is conducted. It comprises the hypothesis formulation, variables selection (dependent and independent), selection of subjects and, design of experiment determination. It also considers the threats to experiment evaluation.

3) Operation (execution): It follows from the design. It comprises: (i) the study configuration (preparation), where participants are chosen and the materials are prepared; and (ii) the execution that collects the data that should be analyzed.

4) Analysis and Interpretation: Responsible by the compilation of collected study data. It comprises descriptive statistics, data set reduction and hypothesis testing.

5) Presentation and Package: In this stage the information about the study is presented and the package is generated. It is essential for the study replication.

Each stage defines activities to be accomplished and specific work products. Table 3 presents the stages covered for each experimental environment found in our systematic review. Almost all empirical environments selected in our study give some kind of support to the controlled experiment definition, planning, and execution phases. The analysis stage is only supported by two of them. Moreover, four of the studied environments give support to the packaging step, which is important for replication, but none of them defines an explicit format or pattern to package experiments. CAESE framework is the only that mentioned supporting the complete process but the Ginger2 does not implement all these phases

C. *SE Empirical Tools Functionalities Supported (RQ.3)*

It is fundamental to understand the level of support provided by each different tool for each different stage of the process. Table 4 shows some features described by the primary studies.

TABLE II. SELECTED DATA

ID	Title	Year	Reported Tool	University/Industry	Publish Source
PS1	Conducting realistic experiments in software engineering	2002	Web-based Simula Experiment Support Environment (SESE)	Simula Research Laboratory	ISESE
PS2	Collecting Feedback During Software Engineering Experiments	2005	Feedback-collecting tool	Simula Research	ESEM
PS3	A Web-based Support Environment for Software Engineering Experiments	2002	Simula Experiment Support Environment (SESE)	Simula Research Laboratory + KompetanseWeb AS	NJC
PS4	SESE – an Experiment Support Environment for Evaluating Software Engineering Technologies	2002	Simula Experiment Support Environment (SESE)	Simula Research Laboratory + KompetanseWeb AS	NWPER
PS5	Infrastructure for SE Experiments Definition and Planning	2004	experimental Software Engineering Environment (eSEE)	COPPE/UFRJ	ESELAW
PS6	eSEE: a Computerized Infrastructure for Experimental Software Engineering	2004	experimental Software Engineering Environment (eSEE)	COPPE/UFRJ	ESELAW
PS7	A computerized infrastructure for supporting experimentation in software engineering	2005	experimental Software Engineering Environment (eSEE)	COPPE / UFRJ	ESELAW
PS8	Supporting Meta-Description Activities in Experimental Software Engineering Environments	2005	Meta-configurator from experimental Software Engineering Environment (eSEE)	COPPE/UFRJ	ESELAW
PS9	An environment to support large scale experimentation in software engineering	2008	experimental Software Engineering Environment (eSEE)	COPPE/UFRJ	ICECCS
PS10	Towards a Computerized Infrastructure for Managing Experimental Software Engineering Knowledge	2004	experimental Software Engineering Environment (eSEE)	COPPE/UFRJ	JIISIC
PS11	Exploring the use of crowdsourcing to support empirical studies in software engineering	2010	Mechanical Turk	University of Nebraska	ESEM
PS12	Value-Based Empirical Research Plan Evaluation	2007	value-based empirical research (VBER) planning framework	Vienna Univ. of Technol.	ESEM
PS13	A Framework for Software Engineering Experimental Replications	2008	FIRE - Framework for Improving the Replication of Experiments	Salvador University	ICECCS
PS14	An Environment for Conducting Families of Software Engineering Experiments	2008	Experiment Manager framework	University of Nebraska	Advances in Computers
PS15	Ginger2: An Environment for Computer-Aided Empirical Software Engineering	1999	CAESE Framework and Ginger2	Nara Institute of Science and Technology	IEEE TSE

TABLE III. EXPERIMENTAL PROCESS STAGES SUPPORTED PER TOOL

	(1)	(2)	(3)	(4)	(5)
SESE	■	■	■		■
eSEE	■	■	■		■
VBER		■			
Mechanical Turk	■	■	■		
FIRE	■	■	■		■
Ginger2	■	■	■	■	■
Experiment Manager framework			■	■	

TABLE IV. TOOLS FEATURES

Functionalities	Tools
Define the experiment (questionnaires, tasks descriptions, artifacts and roles)*	SESE, eSEE, Ginger2
Allow the researcher defining the kind and number of subjects that should take part in the experiment	SESE, eSEE
Allow the user (subject) fills in questionnaires and downloads task descriptions, and other required documents. User carries out the tasks, answers questions along the way and uploads the finished documents.	SESE, eSEE, Mechanical Turk
Do continuous timestamp for activities	SESE, Mechanical Turk, Experiment Manager framework, Ginger2
Monitor experiment (the progress of each subject)	SESE, eSEE
Collect and storage data	SESE, Ginger2, Experiment Manager framework
Store and publish the experimentation process in a process model repository	eSEE
Specify and visualize the Experimental Plan trough well-defined process	eSEE
Elaborate the documents produced/consumed throughout the Experimentation Process	eSEE
Make the experimental tasks available	eSEE, Ginger2, Mechanical Turk
Control the experiment	eSEE
Register lessons learned	eSEE
Gather feedback from subjects	Feedback-collecting tool (SESE)
Characterize the study (using GQM)	Ginger2, VBER
Support Experiment Design	Ginger2
Support Data Analysis	Ginger2, Experiment Manager framework
Integrate data among different tools, integrate different control tools, integrate analysis tools	Ginger2
Support Packaging	eSEE, SESE, Ginger2, FIRE
Manage payment (to subjects)	Mechanical Turk
Address knowledge sharing issues both at the intra-group (internal replications) and inter-group (external replications)	FIRE
Frequent interaction among groups through e-mail and phone calls	FIRE
Execution of pilot studies	FIRE
Help identifying potential conflicts that indicate project risks	VBER
Elicit empirical study principal stakeholders	VBER

(industry and academia) and their key value propositions expected	

In an experimental process, one needs to choose the experimental study design. This choice determines how to organize (participants, experimental material, and treatments), to run the experiment, and to analyze the experiment collected data using a specific statistical analysis method. However, no tool details what support is given to set up the experiment design during the planning.

We also realized that analysis tools that are part of investigated infrastructures are relatively simple, except the Ginger2 that explicitly mentions the existence of internal tools to support observational and computational analysis. Another issue that is not addressed by the environments is how to define metrics to be collected during the experiment execution. Ginger2 mentions a Statistical Metrics Tool for data analysis that computes and returns various statistical values and metrics that have been defined by experimenters but it does not detail how it works.

Finally, although Ginger2 and SESE enable the experiment definition, they have a predetermined process that cannot be adjusted according to the needs of each new experiment.

D. *Developing Tools to Conduct SE Experiments (RQ.4)*

In our systematic review study, we have initially restricted to only select papers published between 2002 and 2010, when searching for the papers from the investigated conferences and journals. In the second step of our study, we included the studies found while searching the reference section of the primary studies even if they are outside of the established initial period, as established by our protocol. Our main aim was to try to capture a wide set of related research work. Our study results demonstrated that research on automated execution of SE controlled experiments was performed mainly over the last decade.

Table 5 shows that the selected studies were originated from five different countries. It is important to emphasize that some studies have researchers coming from different countries but we have only considered the first author affiliation and country. Among the selected studies, 47% comes from Brazil (mainly from COPPE/UFRJ) and other 27% comes from the Simula Research Laboratory/ Norway.

TABLE V. DISTRIBUTION OF STUDIES OVER AFFILIATION/COUNTRIES

Country	**Affiliation**	**Studies Distribution**
Japan	Nara Institute of Science and Technology	1
USA	University of Nebraska	2
Norway	Simula Research Laboratory	4
Austria	Vienna University of Technology	1
Brazil	COPPE / UFRJ	6
Brazil	Salvador University	1

IV. STUDY LIMITATIONS

Limitations are most related to our search strategy. The first plan was to perform both manual and automatic searches. When we started to define our search string we realized that we are in a very large scope due to the diversity of our research. Many research works in software engineering present frameworks, tools, environments and infrastructures for other different contexts than empirical software engineering. In addition, there are also many research works that describe experimental studies and controlled experiments. Because of those reasons, it was extremely difficult to perform an automated search without resulting in a large number (thousands) of studies not related to the purpose of our systematic review. As a result we decided to execute only manual searches. We agree that there is a significant effort to examine many irrelevant studies when submitting general automatic searches, but on the other hand we can ensure the gathering of relevant studies when choosing specific conferences and journals. Similar strategies have been adopted by other existing systematic reviews [22].

V. DISCUSSION

After performing the systematic review we have identified some weaknesses and opportunities for future improvements. In this section, we present these new perspectives based on the results of the systematic mapping.

A. *Environment customizations based on the experiment design*

The proposal of the investigated tools is to facilitate planning and conducting an experiment, minimizing threats to validity, and reducing the time spent in preparation, execution, and analysis of a controlled experiment. However, they do not mention how to set up the experimental design or how to organize the execution according to a statistic experimental design, even the most known, such as completely randomized design (CRD), randomized complete block design (RCBD), or Latin square (LS). We believe that providing assistance in how controlled experiments will actually be arranged according to statistical design can not only reduce the effort of skilled researchers on experimental software engineering, but also encouraging researchers that are no experts to perform such kinds of experiments.

B. *Improved analysis capabilities*

Although the analysis phase has been supported by two of the found approaches, all the environments exhibit weaknesses that should be addressed, such as: (i) help to set up the experimental design; (ii) automatic workflow generation of the execution procedure for each experiment participant in order to facilitate the automatic collection of their specific data; and (iii) finally, analysis capabilities that facilitates the production of graphics and data that help the analysis of the study according to the chosen experiment design.

C. *Guidance and automatic data collection*

The effort to run and manage the great volume of information collected in the experiment is substantial. In this context, actions are necessary to minimize the manual data collection effort, and the time consumed to run the experiment, in other words, such environments should enable subjects to keep updated about their current activities through guidance and automatic data collection. It can simplify the process, since the participant would not have to collect data such as time for each activity performed, and to follow their activities through of a systematic and customized workflow of your duties. We believe these represent existing deficiencies of existing environment.

D. Future improvements for experimental software engineering environments

We have observed in our systematic review that current some environments for empirical software engineering are adaptable and extensible for specific needs of certain experiments. We identified that those environments should address the following requirements: (i) flexibility for integration with external tools, as the execution of controlled experiment involves a wide range of external tools that have to be integrated and monitored to support the complete experiment, such as process management tools, integrated development environments (IDE), testing tools, and statistic tools; and (ii) flexibility to extend the environment – to address the variety of requirements of experiments from different domains, it is also fundamental to promote the extensibility of the SE experimental environment to support new experimental study design, collected metrics, strategies to collect information from subject, and so on.

VI. CONCLUSIONS AND FUTURE WORK

This paper reported a systematic review study of automated support to conduct experiments in software engineering. The results indicate a restricted number of existing environments, infrastructures, tools or frameworks (total of seven). Moreover, there are few empirical studies reporting the usage of these tools. Potential future improvements for the development of experimental software engineering environments are the support to their customization to address specific needs of experiments to give more flexibility to extend their basic functionality, and to allow the integration with external tools. Inspired on the results and illustrated challenges of this systematic review, we are developing a customizable model-driven environment for supporting and conducting controlled experiments in software engineering.

ACKNOWLEDGMENT

This work was partially supported by the National Institute of Science and Technology for Software Engineering (INES, www.ines.org.br), funded by CNPq under grants 573964/2008-4, 560256/2010-8, and 552645/2011-7, and by FAPERN.

REFERENCES

[1] D. Sjoeberg, J. Hannay, O. Hansen, V. Kampenes, A. Karahasanovic, N.-K. Liborg and A. Rekdal, "A survey of controlled experiments in software engineering," IEEE Transactions on Software Engineering, Vols. %1 de %231 , Issue: 9, pp. 733 - 753, Sept. 2005.

[2] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell and A. Wesslén, Experimentation in Software Engineering: An Introduction, Boston/Dordrecht/London: Kluwer Academic Publishers, 2000.

[3] S. L. Pfleeger, "Experimental design and analysis in software engineering: Part 2: how to set up and experiment," ACM SIGSOFT Software Engineering Notes, vol. 20, pp. 22-26, jan 1995.

[4] V. Basili, F. Shull and F. Lanubile, "Building Knowledge through Families of Experiments," IEEE Trans. Software Engineering, vol. 25 (4), pp. 456-473, July/August 1999.

[5] B. A. Kitchenham, T. Dyba and M. Jorgensen, "Evidence-based software engineering, In ICSE '04," n. ICSE, p. 273–281, 2004.

[6] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering," 2007.

[7] E. Arisholm, D. I. K. Sjøberg, G. J. Carelius and Y. Lindsjørn, "A Web-based Support Environment for Software Engineering Experiments," Nordic Journal of Computing, n. 9(4), pp. 231-247, 2002.

[8] E. Arisholm, D. I. Sjøberg, G. J. Carelius and Y. Lindsjørn, "SESE – an Experiment Support Environment for Evaluating Software Engineering Technologies," Nordic Workshop on Programming and Software Development Tools and Techniques, pp. 81-98, 2002.

[9] A. KARAHASANOVIC, B. ANDA, E. ARISHOLM, S. E. HOVE, M. JØRGENSEN, D. I. K. SJØBERG and R. WELLAND, "Collecting Feedback During Software Engineering Experiments," Empirical Software Engineering, n. Springer Science + Business Media, p. 113–147, 2005.

[10] D. I. Sjøberg, B. Anda, E. Arisholm, T. Dybå, M. Jørgensen, A. Karahasanovic, E. F. Koren and M. Vokác, "Conducting Realistic Experiments in Software Engineering," International Symposium on Empirical Software Engineering , 2002.

[11] A. C. Dias Neto, R. F. Barcelos, P. S. M. Santos, S. N. Mafra and G. H. Travassos, "Infrastructure for SE Experiments Definition and Planning," ESELAW'04, Brasília, Brazil, 2004.

[12] P. G. MIAN, G. H. TRAVASSOS, A. R. C. ROCHA and A. C. C. NATALI, "Towards a Computerized Infrastructure for Managing," Jornadas Iberoamericanas em Ingeniería del Software e Ingeniería del Conocimiento, 2004.

[13] P. Mian, G. Travassos and A. Rocha, "eSEE: a Computerized Infrastructure for Experimental Software Engineering," Experimental Software Engineering Latin American Workshop (ESELAW'04), 2004.

[14] P. G. Mian, G. H. Travassos and A. R. C. Rocha, "A computerized infrastructure for supporting experimentation in software engineering," Experimental Software Engineering Latin American Workshop, 2005.

[15] G. H. Travassos, P. S. M. Santos, P. G. Mian, A. C. Dias Neto and J. Biolchini, "An environment to support large scale experimentation in software engineering," 13th IEEE International Conference on Engineering of Complex Computer Systems, pp. 193-202, 2008.

[16] W. A. Chapetta, P. Santos and G. H. Travassos, "Supporting Meta-Description Activities in Experimental Software Engineering Environments," ESELAW'05, Brazil, 2005.

[17] S. Biffl e D. Winkler, "Value-Based Empirical Research Plan Evaluation," International Symposium onEmpirical Software Engineering and Measurement, September 2007.

[18] K. T. Stolee and S. Elbaum, "Exploring the use of crowdsourcing to support empirical studies in software engineering," 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, 2010.

[19] K. Torii, K. Nakakoji, Y. Takada, S. Takada and K. Shima, "Ginger2: An Environment for Computer-Aided Empirical Software Engineering," IEEE Transactions on Software Engineering, vol. (25)4, p. 474–492., July 1999.

[20] M. G. Mendonça, J. C. Maldonado, M. C. d. Oliveira, J. Carver, S. C. P. F. Fabbri, F. Shull, G. H. Travassos, E. N. Höhn and V. R. Basili, "A Framework for Software Engineering Experimental Replications," IEEE International Conference on Engineering of Complex Computer Systems, 2008.

[21] L. Hochstein, T. Nakamura, F. Shull, N. Zazworka, V. R. Basili and M. V. Zelkowitz, "An Environment for Conducting Families of Software Engineering Experiments," Advances in Computers, vol. 74, p. 175–200, 2008.

[22] V. Alves, . N. Niu, C. Alves and G. Valença, "Requirements engineering for software product lines: A systematic literature review," Information & Software Technology , vol. 52(8, pp. 806-820, 2010.