On the Use of Support Mechanisms to Perform Experimental Variables Selection

Lilian P. Scatalon*, Rogério E. Garcia†, and Ellen F. Barbosa*

*University of São Paulo (ICMC-USP), São Carlos-SP, Brazil
†São Paulo State University (FCT-Unesp), Presidente Prudente-SP, Brazil
lilian.scatalon@usp.br, rogerio.garcia@unesp.br, francine@icmc.usp.br

Abstract

The selection of variables in a given experiment is crucial, since it is the theoretical foundation that guides how data should be collected and analyzed. However, selecting variables is an intricate activity, especially considering areas such as Software Engineering and Education, whose studies should also consider human-related variables in the design. In this scenario, we aim to investigate how a support mechanism helps on the variables selection activity of the experiment process. To do so, we conducted a preliminary study on the use of an experimental framework composed by a catalog of variables. We explored the domain of the integration of software testing into programming education. Participants were divided into two groups (ad hoc and framework support) and asked to select variables for a given experiment goal. We analyzed the results by identifying threats to validity in their experimental design drafts. Results show a significant number of threats of type inadequate explication of constructs for both groups. Nonetheless, the framework helped to increase the clarity of concepts selected as variables. The cause of most raised threats, even with the framework support, was an inaccuracy in selecting the values of such variables (i.e. treatments and fixed values).

Keywords: Experimental design, Variables selection, Support mechanisms and Experimental framework.

1 Introduction

The central idea in an experiment is a cause-effect relationship of a given phenomenon. Naturally, the phenomenon of interest may be affected by a sheer number of variables, but the researcher is supposed to select the ones related to the hypothesized cause-effect relationship. Such variables make up much of experimental design, i.e. the plan to conduct the experiment [8, 19].

Researchers usually rely on their personal experience and the empirical literature as a source to help designing their experiments [6]. In this sense, Borges et al. [4] identified several support mechanisms present in the literature of Software Engineering.

Some mechanisms provide help in terms of method, by delineating the experiment process and guidelines on how to conduct its composing activities, such as [19, 8] and [9]. Still, an experimental framework is another kind of support mechanism that provides help in the sense of promoting better study designs.

An experimental framework usually includes models of the domain of interest, providing the basic structure of experiments in such domain [2, 7]. In this sense, it can help to design new studies as a support mechanism to define domain-specific elements.

In this paper we investigated the support provided by an experimental framework to study designing. We explored the domain of software testing integration into programming courses, with a framework that we created in previous works [16], named Step. More specifically, we were interested in evaluating the support of Step while researchers conduct the variables selection activity, which is part of the planning phase in the experimental process.

The remainder of this paper is organized as follows. Section 2 describes the variables selection activity and the experimental framework Step. Section 3 presents other existing frameworks in the literature and similar studies that also investigated research activities. The study protocol and the obtained results are described in sections 4 and 5, respectively. We discuss threats to validity in Section 6. Finally, Section 7 presents conclusions and future work.

DOI reference number: 10.18293/SEKE2020-146

2 Background

The **variables selection** activity involves representing the investigated cause and effect constructs as experiment variables. The cause is represented by *independent variables* and the effect by *dependent variables*.

Juristo and Moreno [8] indicate in details the rationale to select the variables of a given experiment. For all the identified "input" variables representing the cause construct, i.e. *independent* or *context variables*, the researcher has to determine whether each one is a *factor*, *parameter* or *blocking variable*.

Similarly, the identified "output" variables representing the effect construct are *dependent variables*. Such variables hold quantitative result values and should be operationalized by means of a *metric*, if not directly measurable. Still, the *hypothesis* states in a testable way the researcher's guess about how these selected variables will behave during the experiment.

An experimental framework can provide support to identify and properly select such variables [2, 7, 17, 10]. In this scenario, we created an experimental framework, named **Step**, for studies on the integration of **s**oftware **te**sting into **p**rogramming education [16].

Step includes the model depicted in Figure 1, which consists in a catalog of variables on such research domain. The idea is to support researchers in the planning phase of the experiment process [19], more specifically in the activities of context selection, hypothesis formulation and variables selection.

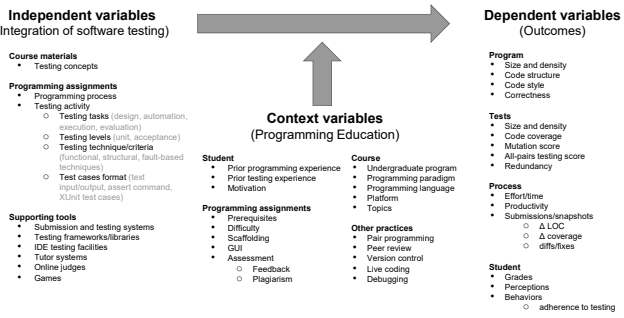

Figure 1. Experimental framework Step

3 Related Work

There are several proposals of frameworks in the Software Engineering literature, all aiming to incorporate domain models. Authors use different names to refer to this kind of frameworks: *organizational framework* [2], *research framework* [7] and *evaluation framework* [18, 11]. Nevertheless, all of them include models of domain-specific elements (e.g. variables) that should be defined when designing an experimental study.

It is worth mentioning that the Computer Science Education area presents an experimental framework for algorithm visualization [12]. In this case, the authors explore one independent variable of the domain, i.e. students' engagement with visualization tools.

The aforementioned frameworks have been used as a reference to design studies in each respective domain, as their authors demonstrate. In this work, we were interested in evaluating the support of this kind of framework while researchers conduct experimental activities.

To this end, we followed a similar approach to Rainer et al. [14], Neto and Conte [13] and Ribeiro et al. [15], which also evaluated researchers conducting research activities, such as applying guidelines, performing validity evaluation and conducting systematic reviews.

4 Method

We conducted an exploratory study on the use of Step by researchers that were not involved with the framework creation. Our goal, expressed using the GQM template [1], is as follows:

Analyze *the use of Step*
for the purpose to *characterize*
with respect to *validity of variables selection*
from the point of view of the *researcher*
in the context of *graduate students selecting variables and formulating a hypothesis for a given research goal*

As stated in the goal, we investigated the use of Step as a support mechanism during the experiment process. We focused on the variables selection activity, providing an experiment goal on our domain of interest (i.e. software testing in programming education) as a starting point to participants.

4.1 Participants

There were seven participants in total, all graduate students that completed the Experimental Software Engineering course at ICMC-USP. Hence, they all had knowledge on the basics of experimentation and the experiment process.

We characterized their background experience both in terms of experimentation and our area of interest (programming education). Firstly, Figure 2 shows how many experiments they have conducted (including definition, planning, execution and analysis). Note that every participant conducted at least one experiment and most were involved in two or more experiments.

Since we aim to evaluate the use of Step during the experiment process, we asked participants what other support

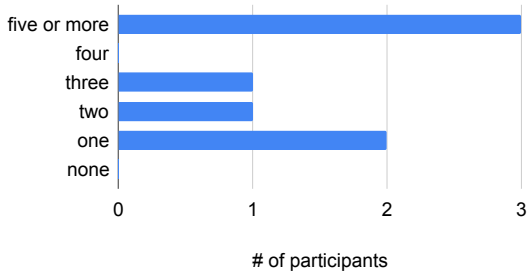

Figure 2. Number of experiments conducted by the participants

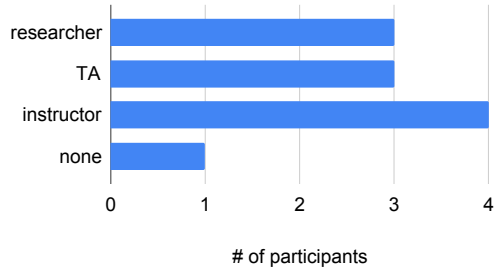

Figure 4. Experience in programming education

mechanisms they usually consult while conducting experiments. Borges et al. [4] identified several mechanisms used by researchers to conduct empirical studies. We presented the ones related to experiments as options to participants, namely: [19], [9], [1], [8], and [2]. Nonetheless, they could indicate other sources of information as well. There was only one mention to another paper [3].

Figure 3 provides an overview of responses. We refer to the options as the first author's name, distinguishing the repeated ones by adding the main subject next to it. The book of Wohlin et al. [19] is the most consulted one, followed by the GQM model [1]. On the other hand, it is interesting to note that nobody indicated the book of Juristo and Moreno [8], especially considering that such book provides detailed rationale on designing experiments.

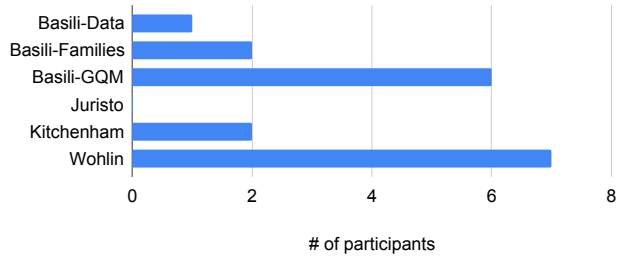

Figure 3. Support mechanisms used by the participants to conduct experiments

Regarding their background on our domain of interest, we asked participants about their experience in programming education. We provided options in terms of the roles in which they could have performed activities in this area, namely instructor, teaching assistant (TA), researcher and none. As Figure 4 shows, most (85.71% – 6) had some kind of experience in the area, whether in practice, as instructor or TA of programming courses, or in theory, as researchers.

We also asked about their familiarity with the integration of software testing into programming education. All had some kind of familiarity on the domain, since the option "none" was not selected by anyone. Some (42.86% – 3) even have conducted empirical studies on the domain.

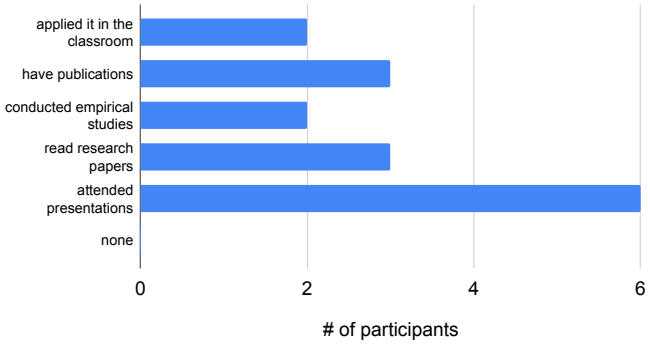

Figure 5. Familiarity with the integration of software testing into programming education

4.2 Procedures and materials

The study involved participants performing variables selection in our domain of interest with two different approaches. They were thus divided into two groups: one using an ad hoc approach (G1) and the other consulting Step (G2).

Firstly, participants filled out a consent form and then received training on study designing, to recall basic concepts such as independent, context and dependent variables.

Moreover, we reinforced the rationale to select which input variables should be factors, parameters or blocking variables and which output variables should be the investigated dependent variables in a given experiment. Only participants in G2 received additional training on the experimental framework Step.

We handed out the training materials on experiment designing to both groups and the overview of Step only to G2. Then, we asked participants to fill out the characterization form, which provided us information on participants' background, as discussed in Section 4.1.

Finally, participants undertook the study task, generating experimental design drafts. We asked them to perform variables selection and formulate a hypothesis for the following experiment goal:

Analyze *progressive assignments*
for the purpose to *evaluate*
with respect to *student's testing performance*
from the point of view of the *researcher*
in the context of *introductory programming courses of Computer Science at ICMC-USP*.

We discussed the underlying scenario with participants, highlighting the motivation to explore this particular goal, which is the following. If students had a greater incentive to ensure quality in their programs, maybe they would feel the need to conduct software testing. One way to lead students in this direction is conducting progressive assignments [5].

A sequence of assignments could be formulated in such a way to configure the progression, i.e. all assignments, except the first one, should have as a prerequisite the solution of the previous one. Students would have to maintain their code from previous solutions, instead of starting them all from scratch.

To complete the study task, participants were supposed to fill out a form, resulting in an experimental design draft. The form was composed by the elements required in the experiment activities we explored: hypothesis, independent and context variables (factors with respective treatments, parameters with respective values, blocking variables with respective blocks) and dependent variables with respective metrics/description.

We evaluated each experimental design draft by means of the number of identified threats to validity. To do so, we used the threats to validity presented by Wohlin et al. [19] as a checklist. However, we only considered threats due to the activities that participants performed, namely hypothesis formulation and variables selection.

Furthermore, the selected threats have to do with how well experiment variables represent the theory constructs (construct validity) and whether the investigated relationship is indeed causal (internal validity).

The remaining types are related to representativeness of subjects and objects (external validity) and issues of the statistical analysis (conclusion validity). Such threats are raised due to decisions of other activities outside the scope of this study (i.e. selection of subjects, instrumentation, execution, hypothesis testing and so on).

5 Results

Table 1 shows occurrences of threats to validity for individual participants. Each type of threat is labeled with an id (T1, T2, and so on). A dash in a cell indicates that the corresponding threat in the row had no occurrence for the participant in the column. Similarly, each value indicates the number of threat occurrences for a given participant.

These same results are summarized in Table 2 with the average (Avg.) and standard deviation (SD) for each group. Considering all threats to validity (i.e. total in the last row), participants using Step presented less threats in average than the ones selecting variables ad hoc, respectively 3.25 against 5.66.

Looking at each threat that had occurrences, T1 was the most frequent one, for both groups. In particular, participants s3 and s5, from distinct groups, presented high values for this threat. Still, in average, Group 2 (2.75) presented less T1 threats than Group 1 (5.00).

One example of such threat found in s2's dependent variables, "student program quality", whose description/metric was "student programs will be assessed by instructors' test cases". It is possible to note that it is not clear how quality is going to be measured.

Another example, now on Group 2, from s6's context variables, "student previous knowledge" was selected as a blocking variable, with blocks defined as "different levels of knowledge in Java development". However, it is not clear what levels are these.

Next, T3 had one occurrence in Group 1 for s2, whose selection of dependent variables included only one variable, thus configuring mono-method bias. Again, the selected variable was "student program quality", without defining how quality should be assessed.

All participants in Group 2 selected more than one dependent variable. Moreover, s4, s5 and s6 selected many dependent variables (more than five) without necessarily having a direct relation with the hypothesis they formulated.

Threat T4 was the second most frequent one. For Group 1, s3 indicated that "C++ or Java" would be a context variable, when in fact the correct construct would be "programming language" instead.

For Group 2, s4 indicated "prior programming experience" as a blocking variable, whose blocks would be "up to four completed programming courses" and "more than four completed programming courses". Such division seems arbitrary, since, until completing four programming courses, students can present very different levels of programming experience.

Table 1. Occurrences of threats to validity

		Group 1 Ad hoc			Group 2 Step			
id	**Threat**	s1	s2	s3	s4	s5	s6	s7
T1	Inadequate preoperational explication of constructs	4	4	7	1	7	3	-
T2	Mono-operation bias	-	-	-	-	-	-	-
T3	Mono-method bias	-	1	-	-	-	-	-
T4	Confounding constructs and levels of constructs	-	-	1	1	-	1	-
T5	Interaction of testing and treatment	-	-	-	-	-	-	-
T6	Restricted generalizability across constructs	-	-	-	-	-	-	-
T7	Ambiguity about direction of causal influence	-	-	-	-	-	-	-
	Total	4	5	8	2	7	4	0

Table 2. Summary of threats to validity occurrences

		Group 1 Ad hoc		Group 2 Step	
id	**Threat**	**Avg.**	**SD**	**Avg.**	**SD**
T1	Inadequate preoperational explication of constructs	5	1.73	2.75	3.09
T2	Mono-operation bias	-	-	-	-
T3	Mono-method bias	0.33	0.57	-	-
T4	Confounding constructs and levels of constructs	0.33	0.57	0.5	0.57
T5	Interaction of testing and treatment	-	-	-	-
T6	Restricted generalizability across constructs	-	-	-	-
T7	Ambiguity about direction of causal influence	-	-	-	-
	Total	5.66	2.08	3.25	2.98

6 Threats to validity

Regarding the external validity of our preliminary study, we had a small sample size and the study task may not represent how researchers conduct study designing in real-world conditions.

Firstly, they worked alone only in a limited part of the experiment process, when usually researchers conduct experiments in a holistic and collaborative way. Nevertheless, we believe it was thereby possible to isolate the use of Step.

Another threat to validity that could be raised is limiting researchers to the concepts present in the experimental framework. However, participants in Group 1 (ad hoc) presented design drafts that tended to be more incomplete than Group 2 (Step), what can be verified by the higher number of threats T1 (inadequate preoperational explication of constructs) in Group 1.

Also, it is important to highlight that threats to validity are expected in a human-centered experiment. The presence of a threat does not necessarily invalidate an experimental design. Hence, only the number of threats may not fully capture its quality. Nonetheless, we focused on two types of validity (construct and internal), both related to theory representation, whose corresponding threats should be carefully analyzed and, whenever possible, mitigated.

Finally, only one person, i.e. the first author, analyzed the experimental design drafts and identified the threats to validity. Therefore, we cannot address inter-rater reliability nor discard the presence of bias in the results. Despite this, the findings shed light on how researchers use an experimental framework, along with possible benefits and drawbacks in doing so.

7 Conclusion

In this paper we reported an exploratory study on the use of the framework Step as a support mechanism to select experimental variables. Despite preliminary, we can point out some interesting findings. The overall number of threats to validity in average was lower for participants using Step, what suggests that it does help to perform variables selection while designing an experiment.

However, there was a considerable amount of threats T1 (inadequate preoperational explication of constructs) for both groups. Looking at specific occurrences of this threat, it is possible to observe different trends for each group.

Group 1 (ad hoc) tended to present less clarity when selecting both the concepts for the variables and their respective values (i.e. treatments, values, blocks and metrics). On the other hand, Group 2 (Step) tended to only present difficulties on selecting the latter. Indeed, Step does not provide much support towards defining variables' values.

Another aspect that drew attention in results is that some participants in Group 2 (Step), namely s5, s6 and s7, seem to have aimlessly selected several variables from Step, in an attempt to form a experimental design. However, not every selected variable in this way had a clear relation with their formulated hypothesis or the provided goal.

Hence, results suggest that Step helps to select variables on the domain with more clarity, but does not provide enough support to select their corresponding values and metrics. Also, we observed a possible side effect of novice researchers using an experimental framework, which is the selection of unnecessary variables, in a kind of trial and error behavior.

The purpose of Step is to help researchers to have an overview of what has been done in the domain research, allowing them to borrow useful concepts and to explore "new" ones. More importantly, the organized overview of concepts can help the researcher to clearly see the boundaries of the study being conducted in terms of domain concepts. In this sense, the results on the use of Step allowed us to observe how researchers not involved with its creation would use it and their resulting experimental designs.

As future work, we intend to further investigate how experimental designing is done in practice, especially by novices. Also, we aim to propose a mechanism in terms of method to support the conduction of variables selection and validity evaluation in parallel.

Acknowledgments

We would like to thank the study participants and the paper reviewers. This work was supported by FAPESP (São Paulo Research Foundation) grants 2014/06656-8 and 2018/26636-2.

References

[1] V. R. Basili, G. Caldiera, and H. D. Rombach. Goal question metric paradigm. In *Encyclopedia of Software Engineering*, pages 528–532. John Wiley & Sons, 1994.

[2] V. R. Basili, F. Shull, and F. Lanubile. Building knowledge through families of experiments. *IEEE Transactions on Software Engineering*, 25(4):456–473, 1999.

[3] V. R. Basili and D. M. Weiss. A methodology for collecting valid software engineering data. *IEEE Transactions on Software Engineering*, SE-10(6):728–738, Nov 1984.

[4] A. Borges, W. Ferreira, E. Barreiros, A. Almeida, L. Fonseca, E. Teixeira, D. Silva, A. Alencar, and S. Soares. Support mechanisms to conduct empirical studies in software engineering. In *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ESEM '14, pages 50:1–50:4, New York, NY, USA, 2014. ACM.

[5] H. B. Christensen. Systematic Testing Should Not Be a Topic in the Computer Science Curriculum! In *Proceedings of the 8th Annual Conference on Innovation and Technology in Computer Science Education*, ITiCSE '03, pages 7–10, New York, NY, USA, 2003. ACM.

[6] L. Fonseca, S. Soares, and C. Seaman. Describing what experimental software engineering experts do when they design their experiments: A qualitative study. In *Proceedings of the 11th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ESEM '17, pages 211–216, Piscataway, NJ, USA, 2017. IEEE Press.

[7] H. Gallis, E. Arisholm, and T. Dyba. An initial framework for research on pair programming. In *International Symposium on Empirical Software Engineering (ISESE 2003)*, pages 132–142, Sept 2003.

[8] N. Juristo and A. M. Moreno. *Basics of Software Engineering Experimentation*. Springer Publishing Company, Incorporated, 1st edition, 2010.

[9] B. A. Kitchenham, S. L. Pfleeger, L. M. Pickard, P. W. Jones, D. C. Hoaglin, K. E. Emam, and J. Rosenberg. Preliminary guidelines for empirical research in software engineering. *IEEE Transactions on Software Engineering*, 28(8):721–734, 2002.

[10] P. Morrison. A security practices evaluation framework. In *Proceedings of the 37th International Conference on Software Engineering - Volume 2*, ICSE '15, pages 935–938, Piscataway, NJ, USA, 2015. IEEE Press.

[11] P. Morrison. *A Security Practices Evaluation Framework*. PhD thesis, North Carolina State University, 2017.

[12] T. L. Naps, G. Rossling, V. Almstrum, W. Dann, R. Fleischer, C. Hundhausen, A. Korhonen, L. Malmi, M. McNally, S. Rodger, and J. A. Velazquez-Iturbide. Exploring the role of visualization and engagement in computer science education. In *Innovation and Technology in Computer Science Education (ITiCSE*, pages 131–152, New York, NY, USA, 2002. ACM.

[13] A. A. Neto and T. Conte. Identifying threats to validity and control actions in the planning stages of controlled experiments. In *26th International Conference on Software Engineering and Knowledge Engineering*, 2014.

[14] A. Rainer, T. Hall, and N. Baddoo. A preliminary empirical investigation of the use of evidence based software engineering by under-graduate students. In *Proceedings of the 10th International Conference on Evaluation and Assessment in Software Engineering*, EASE'06, pages 91–100, Swindon, UK, 2006. BCS Learning & Development Ltd.

[15] T. V. Ribeiro, J. Massollar, and G. H. Travassos. Challenges and pitfalls on surveying evidence in the software engineering technical literature: An exploratory study with novices. *Empirical Software Engineering*, 23(3):1594–1663, 2018.

[16] L. P. Scatalon. *A framework for experimental studies on the integration of software testing into programming education*. PhD thesis, University of São Paulo (ICMC-USP), 2019.

[17] L. Williams, W. Krebs, L. Layman, A. Anton, and P. Abrahamsson. Toward a framework for evaluating extreme programming. In *Assessment in Software Engineering (EASE)*, 2004.

[18] L. Williams, L. Layman, and P. Abrahamsson. On establishing the essential components of a technology-dependent framework: A strawman framework for industrial case study-based research. In *Proceedings of the 2005 Workshop on Realising Evidence-based Software Engineering*, REBSE '05, pages 1–5, New York, NY, USA, 2005. ACM.

[19] C. Wohlin, P. Runeson, M. Host, M. Ohlsson, B. Regnell, and A. Wesslen. *Experimentation in Software Engineering*. Springer Publishing Company, Incorporated, 2 edition, 2012.