Confounding Factors When Conducting Industrial Replications in Requirements Engineering

David Callele
Business Development
Telecommunications Research
Laboratories (TRLabs)
Saskatoon, Canada
dcallele@trlabs.ca

Krzysztof Wnuk
Department of Computer Science
Lund University
Lund, Sweden
Krzysztof.Wnuk@cs.lth.se

Markus Borg
Department of Computer Science
Lund University
Lund, Sweden
Markus.Borg@cs.lth.se

***Abstract*—Despite the widely recognized importance of replications in software engineering, industrial replications in software engineering are still rarely reported. Although the literature provides some evidence about the issues and challenges related to conducting experiments and replications the practitioner's view of the issues and challenges has not been fully explored. This paper reports an industrial practitioner's review of a replicated experiment on linguistic tool support for consolidation of requirements from multiple sources. The review identified potential confounding factors from a perspective that differed significantly from that of the designers of the experiment. The results suggest that industrial practice may focus upon specific process aspects that are not necessarily reflected in academic practice.**

***Index Terms*—Requirements engineering, replication, confounding factors, experience report.**

I. Introduction

Replications play an important role in software engineering, furthering our knowledge about which results or observations hold and under what conditions [1]. Despite the widely recognized importance of this type of research study, replications in software engineering are still rarely reported; Sjøberg *et al.* [2] reported that replication studies constituted only 18% of the surveyed experiments, that only 9% of the subjects in the reviewed experiments were practitioners and that undergraduate students are used much more often than graduate students. The comprehensive software engineering experimentation literature that identifies threats to validity (*e.g.* [11]) does not discuss whether the reported threats are as valid for industrial replications as they are for academic replications.

In this paper we report our experiences when replicating an experiment in automatic support for finding and recording similar requirements that originate from different customers [3]. The automatic support tool used the cosine correlation measure to find and present lexically similar requirements to the human analyst. The experiment used students as subjects and investigated whether a tool with linguistic similarity functionality can help the subjects in the requirements consolidation process. Our main research question for this study is: "*Are there additional confounding factors that should be taken into consideration when replicating an experiment in industry?*"

An industrial practitioner review of a replicated experiment [3] identified confounding factors relevant to an industrial replication of the experiment.

II. Replication Context

The reviewed study assumes that incoming requirements and changes to existing requirements are inevitable throughout the entire development process [5][6] and that requirements originate from multiple sources [3][4][5][7]. Having many customers (sources of requirements) creates a risk that some incoming requirements will be similar to already implemented requirements or requirements considered, but not yet implemented. Large companies, operating globally, have numerous requirements sources and the stream of incoming requirements can easily overwhelm the capacity of the requirements analysts to check every incoming requirement, identifying whether or not the requirement was already investigated or implemented. One possible way to assist in this analysis is to find and record similarities while making traceability links, a task reported as time consuming and frustrating [4].

Requirements sources (customers and subcontractors) do not necessarily have knowledge of what requirements may already have been received or knowledge of what requirements have already been implemented. The goal is to rationalize these incoming requirements, identifying their sources, their similarities, whether they have been analyzed already (possibly in a similar but not identical form) and whether they have already been implemented. An automatic method for analyzing similarity between incoming requirements could significantly decrease the amount of time needed to perform this task. Both the original experiment, reported in [4] and the replicated study analyzed in this paper [3] attempt to assess the benefits of using a linguistic similarity method against a manual method for finding and linking similar requirements.

The linguistic similarity method used in both experiments, the ReqSimile tool [9], measures lexical similarity between requirements, ranking candidate requirements for linking then presenting the user with the most relevant of those requirements. ReqSimile utilizes the cosine correlation measure, where each requirement is represented by a vector of linguistic terms with the respective number of occurrences of each term [7].

The alternative method differed between the two experiments. In the original experiment [4] the alternative method, also called the *manual method*, was represented by a

modified version of the ReqSimile tool [7][9] where the linguistic similarity functionality was disabled and participants were constrained to using simple keyword searching to identify similarities. In the replicated experiment [3], the manual method was represented by searching and filtering functionality provided by the Telelogic DOORS tool [10]. In addition to keyword searching, DOORS allows the user to select the attributes included in the search or to use UNIX-style regular expressions. Search results may also be filtered by attribute and content and simple filters may be combined to create more complex filter operations. The final significant difference between the original and the replication was that participants worked individually in the original experience and in pairs during the replicated experiment.

As noted above, the independent variable is the method used by participants in the experiment. The controlled variable is the experience of the participants, evaluated by a questionnaire. The dependent variables considered are: (1) time used for the consolidation, (2) number of analyzed requirements, (3) number of correct links, (4) number of incorrect links, (5) number of correctly not linked and (6) number of missed links (incorrectly not linked).

The hypotheses remained unchanged from the original experiment [4]:

H_1: The assisted method results in the same number of requirements analyzed per minute, as the manual method.

H_2: The assisted method results in the same share of correctly linked requirements as the manual method.

H_3: The assisted method results in the same share of missed requirements links as the manual method.

H_4: The assisted method results in the same share of incorrectly linked requirements as the manual method.

H_5: The assisted method is as precise as the manual method.

H_6: The assisted method is as accurate as the manual method.

The replicated experiment used a different set of participants, drawn from a similar population – a course in Requirements Engineering. There were 45 subjects participating in the replicated experiment and 44 participating in the original experiment. Two questionnaires were used, before and after conducting the experiment to record the subjects' skills in reading and writing English and their industrial experience in software development. The participants in both experiments used two requirements sets, one written in case style (139 requirements) and the other in feature style (160 requirements). The participants were asked to analyze 30 randomly selected requirements from the first set against all 160 requirements from the second set, and to create links where necessary.

III. Results Trigger Review

The results from hypotheses testing in both experiments are summarized in Table I. 'Significant' means that there was a statistically significant difference (with a 95% confidence level) between the manual and the assisted method. For non-significant results, we report interpretations from the replicated experiment as the original experiment reported few interpretations [4]. The results for H1 were contradictory and the results for H_2 and H_3 were significant in both the original and replicated experiments and could be interpreted in favor of the assisted method (ReqSimile). Neither the original nor the replicated experiments provided statistically significant results regarding the number of incorrect links (H_4), precision (H_5) and accuracy (H_6).

Table 1. The results from hypotheses testing in both experiments. "!significant" indicates that the results from hypothesis testing were *not* significant.

Hypothesis	Result original experiment	Result replicated study	Possible interpretation (see also [3])
H_1: Speed	significant	!significant	Observed wide variation in results, possibly due to participant motivation
H_2: Correctness	significant	significant	
H_3:Missed links	significant	significant	
H_4:Incorrect links	!significant	!significant	Uncovered confounding factors.
H_5: Precision	!significant	!significant	
H_6: Accuracy	!significant	!significant	

Triggered by the replicated lack of statistical significance regarding hypotheses H_4, H_5, H_6, and possible additional factors affecting the results regarding H_1, we performed an independent review by an industrial practitioner who reviewed the experimental design and results from both experimental runs. The goal of the review was twofold:

(1) to search for additional confounding factors that may be important when replicating the experiment in industry

(2) to seek further explanations for the lack of statistically significant results regarding H_4, H_5 and H_6.

IV. Result Review by Industrial Practitioner

The first author reviewed the design of, and the results from, the experiment (after its publication) identifying the following items as potential confounding factors.

A. Task Analysis

Participants in the study were required to perform a series of analysis tasks [2] with aspects that are mechanical (such as reading through lists, selecting from lists and creating links between elements) and aspects that are cognitive (such as interpreting statements, remembering statements and correlating between sets of statements).

We present three illustrative subtasks that have the potential to be significant confounding factors when interpreting the results of the experiment.

1) Complexity of generating the search terms: there is a significant difference in the number of operations necessary to generate the search results between the two evaluated methods. In the assisted method, the work is performed by the underlying tool [7][9]. However, in the manual case, the requirement in question must be analyzed by the participants and appropriate search terms must be generated. Records of

the search terms were not kept, nor were the number of attempts made by the participants to generate the final working set recorded. Even if the participants in the manual study were able to generate their final working set on their first attempt, the complexity of the task is much higher than the assisted method. Given the time constraints for the experiments, does the effort for manual generation of the search terms overwhelm the other results? We do not see direct evidence to confirm or deny that this confounding factor actually occurred in these experiments. This confounding factor is related to the inadequate preoperational explication of construct threat [11] but is not discussed in the recent literature study [2]. This factor could influence the results regarding incorrect links (H_4) precision (H_5) and accuracy (H_6).

2) Number of search results in relation to the 'quality' of the search terms: each approach attempts to constrain the search for requirements links from an n*n search space (for n requirements) to a space m*n (m<<n). We are not able to determine the 'quality' of the search terms employed by each team and the number of search results presented to each participant pair could be significantly different. As a result, any differences in the measured results may have been caused by a significant imbalance between the numbers of requirements that were presented to the participants. There is a risk that the experiment is measuring each team's ability to generate effective and efficient search terms and this factor may dominate other results. This factor is related to the reliability of used measures threat [11] and might have been one of the factors affecting the number of incorrect links (H_4), precision (H_5) and accuracy (H_6).

3) Complexity of interpreting search results: we assume that the assisted method returned the same result set to each team. Therefore, differences in results for each team may be attributed to other factors. However, we do not have the same degree of control over the manual method. Also, the fact that a list of highly similar requirements sorted by their similarity degree was presented to subjects may potentially increase the number of incorrect links (H_4) as more false positives are generated, which may negatively impact precision (H_5) and accuracy (H_6).

Let us assume that the requirements are of varying levels of complexity and with varying levels of linkage to other requirements. Given this assumption, and under the knowledge that the experiment is time-constrained, then the order in which participants addressed the requirements can have a significant impact upon the results.

The reported time for running the experiments was 45 minutes, or 90 seconds per requirement – far less than in industrial practice. To facilitate industrial adoption of the results, practitioner time-constraints should have been removed to simulate the effort deemed acceptable to industrial practice. This factor conflicts with the history and maturation threats to internal validity [11] and could influence the results regarding performance (H_1). Further confounding factors include:

4) Using students as practitioner proxies: this factor has been listed among the threats to internal and external validity by Wohlin et al. [11] and discussed by several researchers e.g. [12][13]. In the first author's practice, none of the replication subjects would likely be considered to have sufficient experience to participate in a requirements effort, except as support staff "in training." The most experienced participants claimed approximately two years of experience, none of which included a focus on requirements. It was not identified whether that experience was two consecutive years or two cumulative years. The analysis of the possible influence of industrial experience on results revealed that, in most cases, this factor negatively affected results [3] and these results may only be applicable to practitioners with little or no experience.

5) Reading speed: the reading speed of the participants can be a significant factor in a timed evaluation [14] and could influence performance results (H_1). Adult reading speeds, with significant comprehension, are widely reported from about 100 words per minute to approximately 1000 words per minute. A practitioner with high-speed reading skills would be expected to perform elements of the task at rates of up to an order of magnitude more quickly than their slowest counterparts. If the experimental results do not compensate for this aspect of the environment, they can be dominated by this one factor alone. This factor isn't mentioned by Wohlin et al. [11]. This confounding factor could also influence the results regarding correctness (H_2) and accuracy (H_6).

6) Solution strategies undertaken by subjects: the solution strategy undertaken by each team may not be a linear scan through the presented alternatives. Strategies that may have been employed by teams to improve their performance over a linear scan include, but are not limited to the following:

- Partition the requirements between the partners, with each partner working independently. This approach is particularly effective for generating the search terms in the manual approach.
- Partition the result sets between the partners, one partner working from the bottom of the result set toward the top, the other from the top down.
- Partition the result sets by length of requirements. Scan and evaluate all short requirements first, then proceed to the longer requirements.
- Process all requirements stated using a simple sentence structure first, then move on to compound and complex sentence structure requirements.

These strategies could influence the results regarding performance (H_1) precision (H_5) and accuracy (H_6). If the participants employed any of these techniques (or others not explicitly mentioned here) and did not accurately report upon the technique used in the post-experiment questionnaire then the results may be biased. This factor isn't explicitly mentioned by Wohlin *et al.* [11].

7) Personalities: personality may have played a significant role in the experiment. Pairs of dominant personalities may have clashed, pairs of passive personalities may have dithered and mixing a dominant and a passive

personality may have been a waste of the passive participant resource. This factor was not considered in the analyzed experiment and is often not considered in industrial environments despite the significant potential for the noted issues to occur.

The first author posits that the inexperienced participants may also have been afraid of failure or appearing incompetent in front of their peers. As a result, their attentiveness was boosted, as was their attention to detail. The relatively experienced participants may have been preconditioned by their (meager) experience and they may have been overconfident. These factors aren't explicitly mentioned by Wohlin *et al.* [11].

8) The nature of the requirements sample: the population of 160 requirements was sampled to create a working set of 30 requirements. The contents of this working set may or may not have been representative of the greater population. The requirements themselves were not analyzed to determine what effect changing the working set would have upon the experimental results. Confounding factors include the complexity of the requirement statements, the nature of the requirements (e.g. functional vs. non-functional) and the requirements domain (well-understood or not).

V. CONCLUSIONS

This paper has reflected upon some challenges associated with performing replications of empirical software research experiments in an industrial setting. The analysis indicates that industrial practice may focus upon specific aspects of processes that are not necessarily reflected in academic practice. For example, human factors such as reading speed and personalities, or process optimizations such as solution strategies may be ignored when designing academic experiments as they are not explicitly listed in the experimentation literature [2][11].

However, these factors may need to be addressed to justify applying the research results in an industrial environment or to obtain investment in equivalent industrial experiments. We note that our analysis is based on the opinion of a single practitioner and there exists a risk that our findings are specific to the reviewed experiment.

Future work is planned to focus on investigating which of the identified factors were experiment specific and which could potentially be generalized to other types of experiments in software engineering. Moreover, it would be valuable to repeat the analysis with other practitioners. Finally, we plan to conduct a systematic review for collecting and analyzing all the confounding factors that have been reported when conducting replications in requirements and software engineering.

REFERENCES

[1] F. J. Shull, S. Vegas, N. Juristo, "The role of replications in empirical software engineering", Empirical Software Engineering, vol. 13, pp. 211–218, April 2008.

[2] D. I. K. Sjøberg, J. E. Hannay, O. Hansen, V. B. Karahasanovic, A. Liborg, N. K. Rekdal, "The survey of controlled experiments in software engineering," IEEE Trans Softw Eng, vol 31, Sep. 2005, pp. 733–753.

[3] K. Wnuk, M. Höst, B. Regnell, "Replication of an Experiment on Linguistic Tool Support for Consolidation of Requirements from Multiple Sources," Empirical Software Engineering, vol. 17, pp. 305-344, June 2012.

[4] J. Natt och Dag, T. Thelin, B. Regnell, "An experiment on linguistic tool support for consolidation of requirements from multiple sources in market-driven product development," Emp Soft Eng, vol. 11, pp. 303–329, Jun. 2006, .

[5] G. Kotonya and I. Sommerville, Requirements Engineering: Processes and Techniques , Wiley, 1998.

[6] N. Fogelström, T. Gorschek, M. Svahnberg and P. Olsson, "The Impact of Agile Principles on Market-driven Software Product Development", Softw Maint and Evol – Research and Practice, Vol. 22, pp. 53-80, February 2010.

[7] J. Natt och Dag, Managing natural language requirements in large-scale software development. PhD thesis, Lund University, Sweden, 2006.

[8] C. D. Manning, H. Schuetze, Foundations of Statistical Natural Language Processing. MIT Press, 2002.

[9] The Reqsimile is available at http://reqsimile.sourceforge.net/

[10] IBM Rational Doors is available at http://www-01.ibm.com/software/awdtools/doors/productline/

[11] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, A. Wesslén, Experimentation in Software Engineering, Springer, 2012.

[12] T. Gorschek, M. Svahnberg, M. Borg, J. Börstler, M. Eriksson, A. Lonconsole and K. Sandahl, "A Controlled Empirical Evaluation of a Requirements Abstraction Model", Inf and Soft Techn, vol. 49, pp. 790-805, July 2007.

[13] M. Höst, B. Regnell, C. Wohlin, "Using Students as Subjects – A Comparative Study of Students and Professionals in Lead-Time Impact Assessment," Empirical Software Engineering, vol. 5, pp. 201-214, Nov. 2000.

[14] R. F. Hudson, H. B. Lane and P. C. Pullen, "Reading fluency assessment and instruction: What, why, and how?", The Reading Teacher, Vol.58, pp. 702-714, May 2005.