IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 27, 2019, accepted December 19, 2019, date of publication January 7, 2020, date of current version January 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2964587

Resources for Reproducibility of Experiments in Empirical Software Engineering: Topics Derived From a Secondary Study

CARLOS E. ANCHUNDIA[1] AND EFRAÍN R. FONSECA C.[2]
[1]Department of Informatics and Computer Science, Escuela Politécnica Nacional, Quito 170517, Ecuador
[2]Departamento de Ciencias de la Computación, Universidad de las Fuerzas Armadas—ESPE, Sangolquí 171103, Ecuador

Corresponding author: Carlos E. Anchundia (carlos.anchundia@epn.edu.ec)

This work was supported by the Escuela Politécnica Nacional and its Doctoral Program in Computer Science.

ABSTRACT *Background:* Replication is a recurrent issue in empirical software engineering (ESE). Although it is a foundation of science, replication is hard to execute despite the many supporting tools meant to facilitate reproducibility. For example, in an experiment, which is the most used method in ESE, the number of replications is not enough compared to other sciences. *Objective:* In this study, we aim to identify tools that maximize reproducibility in software engineering experiments and how they are applied. *Methods:* We performed a Systematic Mapping Study and complementary strategies to analyze replication from three concerns (communication, knowledge management, and motivation). We analyzed more than 2,600 studies to get 40 primary studies, using a qualitative analytical tool (Atlas.ti) to create semantic maps for synthesizing our results. *Result:* We found that tools and practices depend on the experiment domain. Human-oriented experiments tend to use an informal mechanism that is costly and time-consuming. On the other hand, technology-oriented experiments are automated, domain-centric, and specialized so they require a learning process and are not transferable to other domains. *Conclusion:* Tools and practices still lack acceptation and usability among the ESE research community. Therefore, reproducibility is mostly relegated to internal replication, at which time and costs can be assumed within research groups. A focus on new alternatives should be considered to broaden replication.

INDEX TERMS Tools, replication, reproducibility, empirical software engineering, experiment.

I. INTRODUCTION

Since empirical research has taken an essential role in the field of software engineering (SE), replication of results has become a challenging topic [1]. Replication allows us to support scientific knowledge through the generalization and refutability of findings [2]. In SE, there are constant contributions regarding replication, which describe concepts, classifications, and frameworks, among others [3]. However, there is still no widely-used standard or practice, which adds complexity when trying to understand and executing replication processes. The literature evidences a lack of replication of experiments in SE, which is a problem [3] being one of the most-frequenly-used methods within empiricism [4]. For this reason, the main challenge around replication in SE is to have a simple process that generates the empirical data necessary to support the scientific evidence.

To perform our analysis of replication, it was necessary for us to first have a clear definition of the concept of replication. The concept of *replication* in the literature is diverse. For example, the term is used to define the degree of equivalence between an original study and its replication [5] while other studies use this term as a synonym for reproduction and reanalysis [3], [6], [7]. On the other hand, *replication* is poorly analyzed and tends to be treated as a different process from experimentation [8]. Additionally, we found that the terms *replication* and *reproducibility* tend to be confused. For this research, we will refer to *reproducibility* as the qualities of a study, while *replication* is used to refer to the action of executing, partially or totally, the activities of a previous experiment performed.

A study can reach its maximum reproducibility when all original information is available; however, it is practically

The associate editor coordinating the review of this manuscript and approving it for publication was Mervat Adib Bamiah.

impossible due to the presence of so-called tacit knowledge [9]. Additionally, experiments have an extensive operational diversity that raises many questions to guarantee the availability of information, such as (a) What is important to report? [8], [10], [11]; (b) How to include this information in an article? [8], [10], [11]; (c) How to standardize additional resources? [8], [11]; (d) How and which web platforms or repositories to use? [1], [8], [10]; (e) How to adapt a tool to a specific type of research? [1]. Different supporting tools have been proposed or developed in response to these questions, which include guides, recommendations, and platforms. Such tools focus on communication mechanisms to transmit knowledge and relevant information such as Carver [3] guides or computer systems such as ARREST [7], among others.

This research aims to identify and analyze the extent to which existing proposals have contributed to the reproducibility of experiments in SE. To achieve this objective, we carried out a systematic mapping study (SMS) of which the main result was the identification of replication tools for communicating and transmitting relevant data to the SE community. The analysis of these tools allowed us to identify the main contributions and problems regarding the needs of reproducible research. Among the main contributions of this work, are:

i. Classification of tools proposed focused on achieving the reproducibility of studies,
ii. information on the use and shortcomings of these tools, and
iii. stages of the experiment process in which researchers use these tools.

The remainder of the article is structured as follows: Section 2 explores the background on the measurement of reproducibility. Section 3 describes in detail the SMS process developed in this investigation. Section 4 shows a summary of the results obtained. Section 5 discusses the tools and possible future actions to be developed. Finally, in section 6, the conclusions from the research are presented.

II. BACKGROUND

Taking into account that reproducibility relies on a replication's qualities (as established in Section I), its measurement should be determined objectively.

The Reproducible Research paradigm (RR) is a good start due to the fact that RR empathizes the use of best practices for transferring information[1]. RR's objective is the verification and validation of studies [13]; that is also the aim of some other tools that we have identified in this research. All of them have in common their focus on providing collaborative and communicational structures for researchers concerning all elements involved in the experiments.

[1]Information refers to any element used in experiments, such as raw data, activities carried out, methods of data extraction, parameters used, or reports [12].

On the other hand, many researchers agree it is necessary to share as much information from the original study as possible to achieve the highest degree of reproducibility. However, this information must cover more than just raw data [2], [5], [10]. Therefore, we believe that procedures and times used in training and executions should also be detailed, as well as the practical and methodological considerations that affect the decisions made during the experiment execution [14].

In this sense, a study's level of reproducibility depends on how it is shared and how its relevant information is structured. Therefore, it is necessary to define how a tool influences reproducibility. In a previous study [15], we found that three points of view allow us to analyze reproducibility: (a) communication mechanisms, (b) knowledge management, and (c) motivation. Nevertheless, we believe that these aspects are subjective when being considered as suitable metrics. In this regard, Becker et al. [13] propose a scheme to rate a study's reproducibility by analyzing eight operations of an experiment: (a) Data source, (b) retrieval methodology, (c) raw data, (d) extraction methodology, (e) study parameters, (f) processed dataset, (g) analysis methodology, (h) result dataset. In this way, each operation is scored according to whether these operations are (a) no, (b) partially, or (c) totally described, mentioned, presented, provided, or reported.

However, Roizer and Roizer [2] argue that not all research is likely to be quantified or evaluated due to the data being possibly based on either private information, such as individuals or companies, or qualitative information. This statement is evidenced by Becker [13], as its measurements show that most experimental investigations focus entirely on the latest operations; that is, the analysis methods and the resulting data.

In our study, we conducted an analysis that does not focus on data (as suggested in [2]), but instead, focuses on the mechanisms of communication, knowledge management, and motivations in research. Hence, we analyzed the results based on the evaluation scheme suggested by Becker [13]. Accordingly, our study aims to identify tools that maximize reproducibility in software engineering experiments and how they are applied. As a guide for achieving this goal, we established three research questions following the evidence-based research practices [16] (see Table 1).

TABLE 1. PICO.

Population	Intervention	Context	Outcome
Communication	Mechanism	Replication of experiments in software engineering	Propitiate
Knowledge	Management		Encourage
Motivation	Practices		Transfer

The resulting questions are:

(**RQ1**): What information communication mechanisms are used in experimentation processes in software engineering to promote reproducibility?

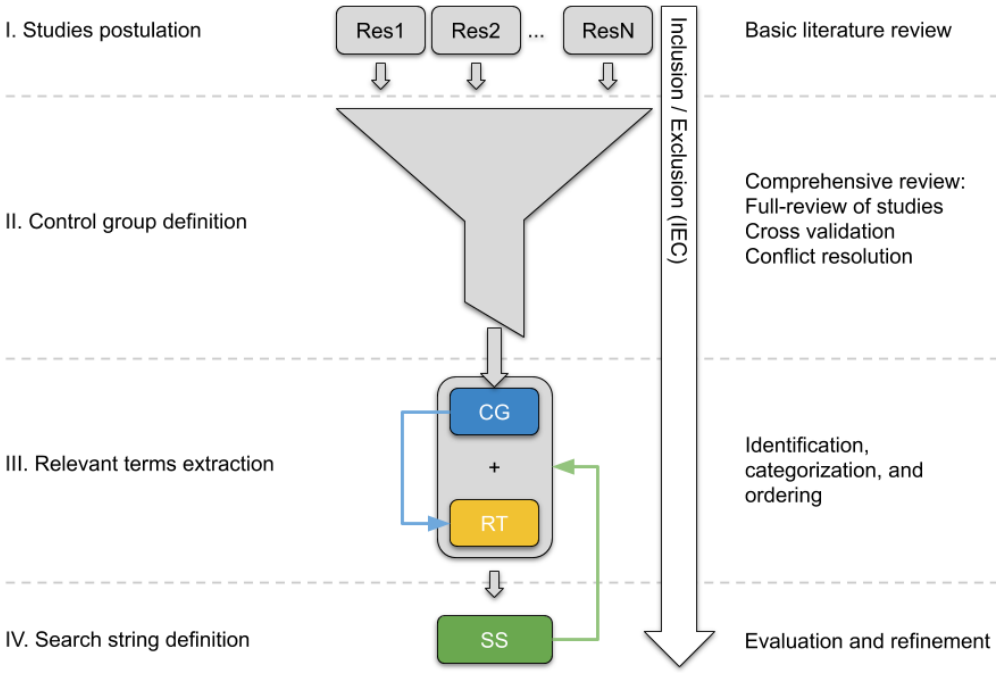


FIGURE 1. Control group support process.

(**RQ1**): How is the knowledge generated in a software engineering experiment managed to be transferred and understood by other researchers?

(**RQ1**): What explicit motivations encourage the replication of experiments in the field of software engineering?

III. RESEARCH METHOD

The research method considered for this research is a Systematic Mapping of Study (SMS), following the guidelines proposed by Kitchenham et al. [17]. The process carried out considers the following phases: (a) search, (b) primary studies selection, (c) quality assessment of primary studies, (d) data extraction, (e) data synthesis, and (f) study limitations.

A. SEARCH

The process used is a proposal to improve the quality of the results we titled as Control Group Support Process (CGSP). Here, it is mandatory to identified inclusion and exclusion criteria (IEC) at the start of the process. However, these could also, be refined throughout the research. We use IEC during the whole process to obtain a selection of studies named Control Group (see Figure 1).

1) INCLUSION AND EXCLUSION CRITERIA

The search process focuses on identifying the most relevant studies thanks to the application of standardized criteria that help researchers decide whether or not to include a study.

The questions and aims of this research allowed us to set the following IEC.

a: INCLUSION

- Studies that mention the use and characteristics of tools to improve the reproducibility of experiments in IS;
- studies describing good practices, procedures, or conditions that facilitate the reproducibility of experiments in IS; and
- studies describing errors, shortcomings or complications to reproduce experiments.

b: EXCLUSION

- Studies that are limited to mentioning tools or procedures without commenting on benefits or difficulties of their use; and
- studies of which the technological or operational infrastructure considered obsolete.

2) CONTROL GROUP

After defining IECs, the next step is to establish a control group (CG). A CG consists of a collection of representative studies that are closely related to the purpose of the research. The main objective of forming the CG is to provide the most relevant terms (RT) to build the search string (SS) [18]. The CGSP consists of four stages shown in Figure 1.

a: STUDY POSTULATION STAGE

In the first stage, researchers proposed a set of 22 studies obtained from an individual literature review, following the IEC. For tracking purposes, we use a codification for all studies following this scheme: CG05-XXXX; where:

TABLE 2. Control group.

File code	Name of study	Year
Accepted		
CG01-ERFC	Investigations about replication of empirical studies in software engineering: A systematic mapping study	2015
CG05-ERFC	A framework for software engineering experimental replication	2008
CG08-ERFC	Analysis of the influence of communication between researchers on experiment replication	2006
CG10-ERFC	Identifying experimental incidents in software engineering replications	2013
CG01-CEAV	Difficulties in running experiments in the software industry: Experiences from the trenches	2015
CG04-CEAV	Replication of software engineering experiments	2013
CG09-CEAV	Supporting and accelerating reproducible empirical research in software evolution and maintenance using TraceLab Component Library	2014
CG10-CEAV	Would wider adoption of reproducible research be beneficial for empirical software engineering research	2017
Rejected		
CG02-ERFC	Replication's role in software engineering	2008
CG03-ERFC	Infrastructure support for controlled experimentation with software testing and regression testing techniques	2004
CG04-ERFC	A web-based support environment for software engineering experiments	2002
CG06-ERFC	Stand on the shoulders of giants: Towards a portal for collaborative experimentation in data mining	2009
CG07-ERFC	Ginger2: An environment for computer-aided empirical software engineering	1999
CG09-ERFC	Replication data management: Needs and solutions	2013
CG11-ERFC	An environment to support large scale experimentation in software engineering	2008
CG12-ERFC	Classifying and analyzing replication packages for software engineering experimentation	2006
CG02-CEAV	Reporting experiments to satisfy professionals information needs	2013
CG03-CEAV	Experimentation with dynamic simulation models in software engineering: planning and reporting guidelines	2015
CG05-CEAV	How practitioners perceive the relevance of software engineering research	2015
CG06-CEAV	A practical guide to controlled experiments of software engineering tools with human participants *(duplicated)*	2013
CG07-CEAV	Investigations about replication of empirical studies in software engineering: A systematic mapping study	2015
CG08-CEAV	Outliers and replication in Software engineering	2014

CG corresponds to the Control Group, 05 is a two-digit ordinal, and XXXX is the initials of the researcher who proposed the particular study.

b: CONTROL GROUP DEFINITION STAGE

In the second stage, we reviewed the 22 studies considering the research objectives, the research questions, and the IEC. The review allowed us to identify eight studies (see Table 2).

c: RELEVANT TERMS EXTRACTION STAGE

It consists of identification, extraction, and classification of RT by researchers. They used two different methods to identify terms. Researcher 1 used a template to write down the words he considered relevant. Researcher 2 used a tool called Atlas.ti [19] to code keywords. We compiled the terms identified using a spreadsheet with the following fields: (i) keyword/phrase, (ii) study code, (iii) frequency of keyword/phrase, (iv) reviewer, and (v) population. The *population* field allocates the different terms under one of the populations specified in Table 1; that is, communication, knowledge, and motivation. For each population, there was a ranking of terms through the *frequency* field.

d: SEARCH STRING DEFINITION STAGE

This is a process to define and evaluate the SS, taking as evaluation criterias (i) the number of studies found is reasonable[2]; (ii) the number of studies from the control group included in the results; and (iii) the studies (titles) from the results are related to the investigation aim. The structure of the SS includes three mandatory elements: tools (communication, knowledge, and motivation), the field of knowledge (empirical software engineering), and the type of research method (experiments) (see Table 3). The ''AND'' connector was used for mandatory elements of the string (populations and contexts); while the ''OR'' connector was used for two purposes: for synonyms of the terms used, and for finding studies that discuss at least one of the search populations specified in the Table 3. At this stage, the final string was obtained after running seven iterations (see Table 4) and tested in the SCOPUS database.

TABLE 3. Structure of search string.

Communication mechanisms	Knowledge management	Motives	Knowledge Field	Research process
OR				
AND				

The string search combination seven was the most optimal, due the articles found was 750, the articles included from the CG was 6, and most titles were consistent with the research aim. The SS resulting is:

```
ALL (("reporting guidelines"
OR "communication mechanism" OR
"report" OR "body of knowledge" OR
"transfer experimental knowledge" OR
```

[2]A number of studies that a person is able to analize in limited period of time that are usually hundreds, but not thousands.

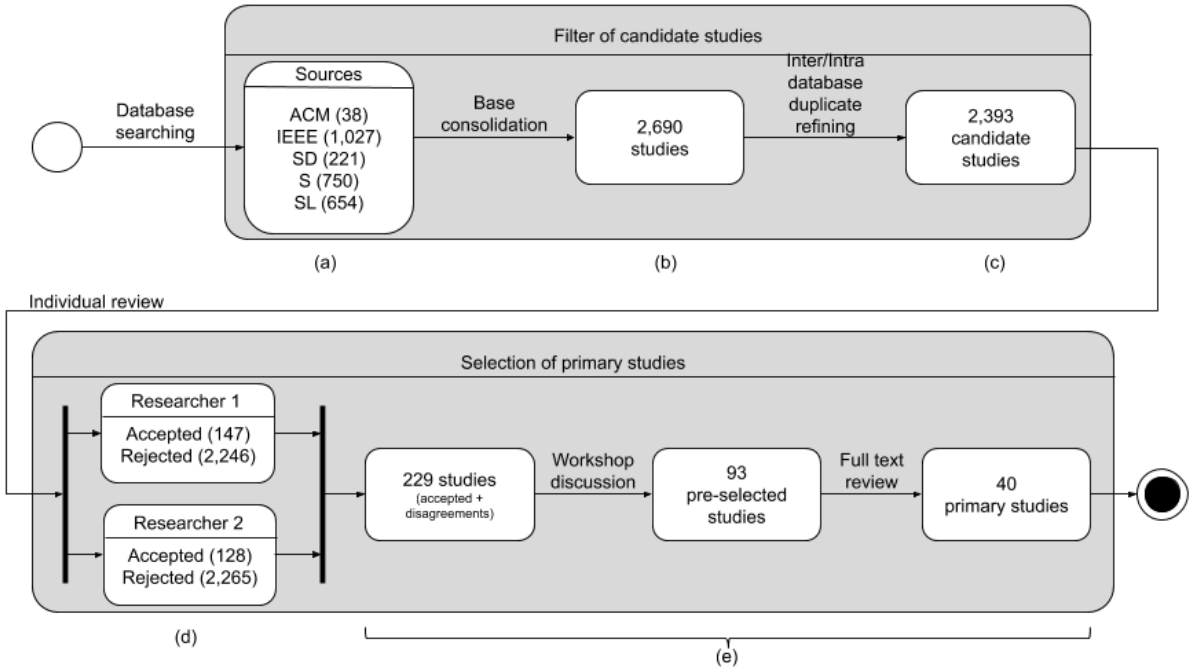


FIGURE 2. State diagram of primary study selection process.

TABLE 4. Search string combinations.

Iteration	Is quantity of studies reasonable?	Number of CG found	Related works?
1	NO (29)	0	YES
2	NO (5567)	0	NO
3	NO (3160)	0	NO
4	NO (212140)	0	NO
5	YES (577)	0	NO
6	YES (389)	3	YES
7	YES (750)	6	YES

```
"knowledge and experience management" OR
"Replication Package") AND ("effort" OR
"generalizable results" OR "different
contexts" OR "detailed description"
OR "new approaches") AND ("empirical
software engineering" OR "software
engineering" OR "software engineering
research") AND ("replication" OR
"reproducibility" OR "reproduction"))
```

B. PRIMARY STUDIES SELECTION PROCESS

1) DIGITAL DATABASES

We used the ACM, IEEE, ScienceDirect, SCOPUS, and SpringerLink databases in this research. These databases were considered due to their affinity with our research area and our access to them. The databases offer various features to search and collect studies; however, we found that not all databases have the same ease of use. For example, ACM, SCOPUS, and IEEE allowed us to collect data automatically by a detailed CVS file. The other digital bases have limited functionalities to collect data, requiring manual intervention, which is a error-prone process and requires extra effort to assure correct transfer of data.

2) FILTERING AND SELECTION OF STUDIES

We used a five-step process for filtering the studies. For this process, we generated five types of artifacts that allow us the traceability of the results, which are described below.

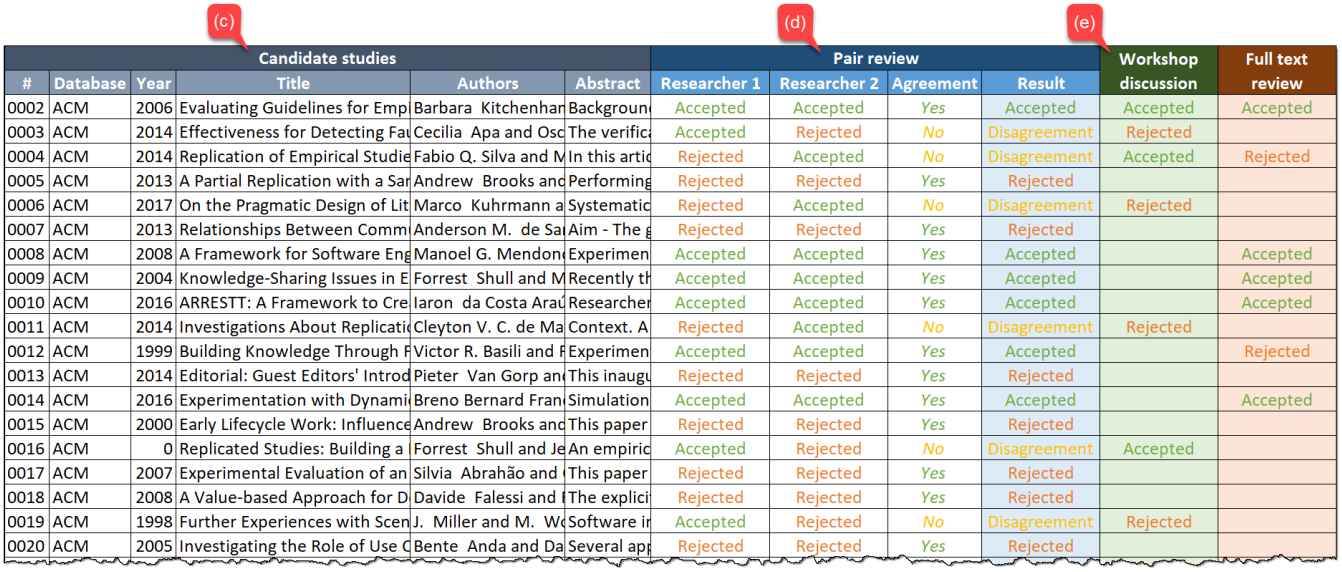
Artifact Type 1: This refers to the files generated or downloaded from digital databases using the search string. These files have CVS or BibTeX (ScienceDirect only) format. Figure 2.a shows the search results for each database.

Artifact Type 2: This consolidates all type 1 artifacts and normalizes the following data: Source, Year, Title, Authors, and Abstract. This artifact also includes an ID field for traceability. As a result, 2,690 studies were consolidated (see Figure 2.b).

Artifact Type 3: This contains 2,393 candidate studies from the filtering of 297 duplicate studies (inter/intra-database) of artifact 1 (see Figure 2.c). Duplicates were identified based on the title and year of publication.

Artifact Type 4: This artifact allows researchers to register acceptance or rejection of a candidate study. There is an artifact for each researcher (see Figure 2.d). The researcher's decision was made based on the analysis of the title, summary, and keywords.

Artifact Type 5: This artifact automatically collects and calculates the decision by study, according to the following categories: (i) Accepted (both researchers accept a study),



Candidate studies						Pair review				Workshop discussion	Full text review
#	Database	Year	Title	Authors	Abstract	Researcher 1	Researcher 2	Agreement	Result		
0002	ACM	2006	Evaluating Guidelines for Emp	Barbara Kitchenhar	Backgroun	Accepted	Accepted	Yes	Accepted	Accepted	Accepted
0003	ACM	2014	Effectiveness for Detecting Fa	Cecilia Apa and Osc	The verifica	Accepted	Rejected	No	Disagreement	Rejected	
0004	ACM	2014	Replication of Empirical Studie	Fabio Q. Silva and M	In this artic	Rejected	Accepted	No	Disagreement	Accepted	Rejected
0005	ACM	2013	A Partial Replication with a Sar	Andrew Brooks and	Performing	Rejected	Rejected	Yes	Rejected		
0006	ACM	2017	On the Pragmatic Design of Lit	Marco Kuhrmann a	Systematic	Rejected	Accepted	No	Disagreement	Rejected	
0007	ACM	2013	Relationships Between Comm	Anderson M. de Sa	Aim - The g	Rejected	Rejected	Yes	Rejected		
0008	ACM	2008	A Framework for Software Eng	Manoel G. Mendon	Experimen	Accepted	Accepted	Yes	Accepted		Accepted
0009	ACM	2004	Knowledge-Sharing Issues in E	Forrest Shull and M	Recently th	Accepted	Accepted	Yes	Accepted		Accepted
0010	ACM	2016	ARRESTT: A Framework to Cre	Iaron da Costa Arau	Researcher	Accepted	Accepted	Yes	Accepted		Accepted
0011	ACM	2014	Investigations About Replicati	Cleyton V. C. de Ma	Context. A	Rejected	Accepted	No	Disagreement	Rejected	
0012	ACM	1999	Building Knowledge Through F	Victor R. Basili and F	Experimen	Accepted	Accepted	Yes	Accepted		Rejected
0013	ACM	2014	Editorial: Guest Editors' Introd	Pieter Van Gorp an	This inaugu	Rejected	Rejected	Yes	Rejected		
0014	ACM	2016	Experimentation with Dynami	Breno Bernard Fran	Simulation	Accepted	Accepted	Yes	Accepted		Accepted
0015	ACM	2000	Early Lifecycle Work: Influence	Andrew Brooks and	This paper	Rejected	Rejected	Yes	Rejected		
0016	ACM	0	Replicated Studies: Building a	Forrest Shull and Je	An empiric	Accepted	Rejected	No	Disagreement	Accepted	
0017	ACM	2007	Experimental Evaluation of an	Silvia Abrahão and	This paper	Rejected	Rejected	Yes	Rejected		
0018	ACM	2008	A Value-based Approach for D	Davide Falessi and I	The explici	Rejected	Rejected	Yes	Rejected		
0019	ACM	1998	Further Experiences with Scen	J. Miller and M. We	Software ir	Accepted	Rejected	No	Disagreement	Rejected	
0020	ACM	2005	Investigating the Role of Use C	Bente Anda and Da	Several app	Rejected	Rejected	Yes	Rejected		

FIGURE 3. Screenshot of artifact 5.

(ii) Rejected (both researchers reject a study), and (iii) Discrepancy (researchers do not agree on the decision about a study). Discrepancies were resolved through a discussion during a workshop (see Figure 2.e). As a result, we obtained 93 pre-selected studies. Finally, the pre-selected studies were reviewed in full (full-text review) to determine the primary studies, which in this case were 40 (see appendix). Figure 3 shows an extract of artifact 5.

C. QUALITY ASSESSMENT

Since SMS is a qualitative study, the search and selection phases depend mostly on the experience of researchers, which tends to induce subjectivity [20]. Therefore, we implemented strategies to evaluate the activities carried out in the research process objectively.

1) SEARCH STRATEGIES

We focused particularly on the process of shaping the search string since it represents the most important input in the search process. Therefore, our strategy consisted of the use of a set of initial studies called the control group (CG). The CG was used to: (a) Extract and rank explicit terms referring to the subject, instead of being obtained from the knowledge of an expert; (b) Conform different search strings based on the hierarchy and classification of the terms provided by the control process instead of random strings based on the researcher's criteria; and, (c) Validate the results of the different SS, increasing an additional evaluation criterion.

2) SOURCE SELECTION STRATEGIES

We established that studies only from reliable sources[3] would be considered in both the initial studies proposed by each researcher, as well as in the selected primary studies. For instance, sources included journals, such as *Information and Software Technology*, *Empirical Software Engineering*; proceedings or conferences such as CIBSE, EASE, or ESEM, and symposiums, such as ACM/IEEE, all of them specialized on empirical software engineering.

[3]Understood as reliable sources to those studies from congresses, journals, and workshops related to the research field and having any tradition or recognition in the community, such as through the H-index or impact factor.

TABLE 5. Kappa coefficient.

		Researcher 2		
		Accepted	Rejected	rm
Researcher 1	Accepted	51	96	147
	Rejected	77	2,169	2,246
	cm	128	2,265	2,393

3) STUDY SELECTION STRATEGIES

We apply two inter-rater agreement metrics [17] in the study selection process: (i) Percent agreement, and (ii) Cohen's Kappa coefficient. These were based on data from artifact 5 and served to verify the consistency of the results. The first metric was calculated by comparing the number of studies in which both researchers had the same criteria (2,220 studies), regardless of whether they accepted or rejected a study, compared to the number of studies reviewed (2,393 studies). We obtained 93% of agreement in this metric. For the second metric, the values shown in the confusion matrix (see Table 5) were used in the equations 1 and 2. A "fair agreement" [17] was obtained due to the k being 0.33; that is, the researchers reached a level of close agreement.

$$Pr(e) = \frac{\left(\frac{cm_1 \times rm_1}{n}\right) + \left(\frac{cm_2 \times rm_2}{n}\right)}{n} \quad (1)$$

$$k = \frac{Pr(a) - Pr(e)}{1 - Pr(e))} \quad (2)$$

where:

$Pr(a)$	is actual agreement
$Pr(e)$	is expected agreement
$cm1$	represents column 1 marginal
$cm2$	represents column 2 marginal
$rm1$	represents row 1 marginal
$rm2$	represents row 2 marginal, and
n	is the number of observations (not raters)

As noted, both metrics reflect that there was agreement; however, its interpretation shows somewhat conflicting results. On the one hand, there is a high value of agreement among the researchers, while the Cohen's Kappa coefficient presents a value considered as a ''fair'' agreement. In this regard, the Cohen's Kappa coefficient may exhibit a problem in studies where category assignment is presented [21], [22]. It is the case of the present study since the relationship between primary studies vs. candidate studies is very low (2,169: 51 studies).

4) ADDITIONAL STRATEGIES

We considered additional strategies to increase the quality of this research. We use the PICO method (see Table 1) not only to describe the research questions elements, but also to structure the search string, categorize relevant terms, and create semantic diagrams in the analysis phase.

5) LIMITATIONS

Despite our strategies to maintain quality, our study is susceptible to other threats to validity in the extension of our search strategy and interpretations.

a: LANGUAGE

This limitation affects the range of studies we can cover in our research. We focused on English-written studies as it is the most representative language in proceedings and journals. As a result, we can omit valuable information written in other languages; however, authors around the world opt to publish their studies in English.

b: TIME PERIOD

It is usual that researches limit the studies covered in the analysis. We decided not to include the time period as an exclusion criterion because we expected to collect practices applied during the experiment process. Praxis is not time-dependent, however, operations are supported by technological tools that we wanted to explore. As a result, we excluded studies based on what we consider as obsolete technology. We expected that our pair review strategy may alleviate this limitation.

c: INTERPRETATION

As this was a qualitative research, experts' interpretation of the information collected from the primary studies could be biased by their own experiences. We expected to reduce bias by the use of a qualitative analytical tool (Atlas.ti), a pair review, and workshops.

D. DATA EXTRACTION PROCESS

For the data extraction process, we used the quantitative analysis tool Atlas.ti [19]. This tool uses codes and semantic networks to analyze the information.

1) CODES

The first step is to identify terms or sentences that represent a relevant idea and label them through one or more codes. This task could result in an unmanageable amount of codes, so it is necessary to perform a refinement through the combination, deletion, or division of codes. Two code examples could be: (i) ''Replication package''; or (ii) ''A package faces two problems, the first is about the integrity and traceability of versions; and the second, difficulty to handle materials''. Figure 4 shows Atlas.ti screenshot of codes in a portion of a study.

A portion of the text from a study may be related to multiple codes. Having many codes allowed us a greater understanding of the relationship between different codes when creating a semantic network.

2) SEMANTIC NETWORKS

Based upon our research questions raised in the Section II, several semantic networks were developed:

(i) the ''body of knowledge'' in replications, (ii) how ''communication'' is used in research, (iii) explore ''motives'' described for research, (iv) how ''replication'' is performed from an experiment, (v) what ''reproducibility'' means or works, and (vi) which ''tools, methods, or proposal'' are used by researchers.

The creation of the semantic network begins with the manual addition of the most representative codes to each network. As a reference, we used the same relevant terms extracted from the control group. Afterward, we performed an automatic import of ''neighbor'' codes, which is a functionality of the tool. This task allowed us to graphically discover all existing relationships between the codes. Finally, we created explicit and meaningful relationships between the codes of each network. Figure 5 shows the network ''Tools, methods, and proposals'' characterized by colors (used for categorization) and their respective relationships.

IV. RESULTS

After the extraction process, we present our findings to the research questions.

A. COMMUNICATION MECHANISMS USED IN SOFTWARE ENGINEERING EXPERIMENT PROCESS TO PROMOTE REPRODUCIBILITY AND THEIR ISSUES

Communication is mainly focused on internalizing and externalizing information [14]. Internalization is used inside research groups to share or teach procedures, plans, actions, and problems. Less-formal tools are generally used in

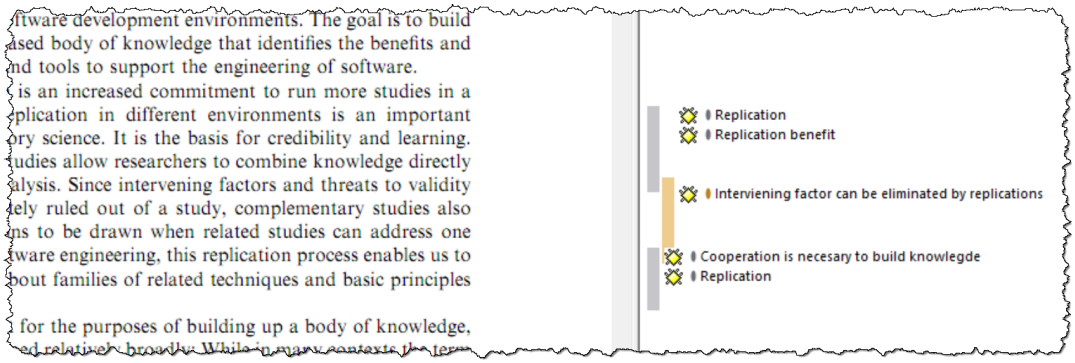


FIGURE 4. Print screen of codes in Atlas.ti.

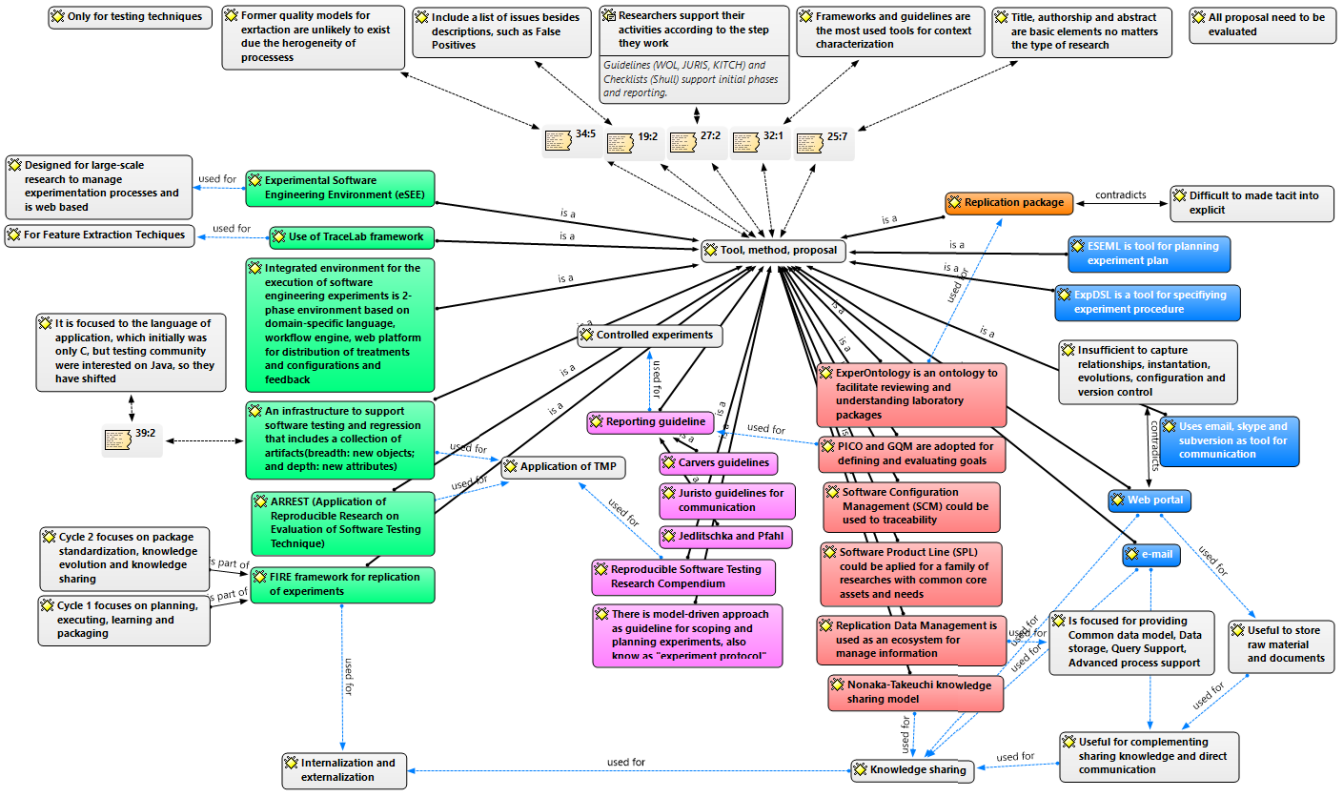


FIGURE 5. Print Screen of a semantic network in Atlas.ti.

internalization such as email, face to face conversations, meetings, and shared folders (in a public or third-party cloud repositories such as Google Drive) [23]. More mature groups formalize meetings based on agile principles (e.g., scrum meetings) [3]. In addition, this kind of group uses subversion tools (e.g., GitHub) to control changes and share code or files [23]. On the other hand, externalization share information with researchers outside the research group. This type of communication is usually done through reports and papers; therefore, reporting guidelines and communication guidelines are commonly used. The most used guidelines are Juristo's [10], [24], Carver's [24]–[27], Wholing's [10], [24], [25], [27]. Additionally, websites and portals are used to complement the information that is included in the reports, as they are limited in extension and functionality [14]. Another widely-used tool is laboratory packages [1], [8], [11], which purposes to include files and other essential artifacts in research.

The most commonly-reported issues are that reports and packages are neither sufficient to capture all information (e.g., raw material) nor to share tacit knowledge (e.g., procedures and decision making processes) [8], [12], [13], [28]. As a consequence, researchers use complementary mechanisms to mitigate these problems:

- External groups use direct communication tools (e.g., video conferences) during externalization to better understand a research [14], [23].
- PICO and GQM are adopted for guidelines to help researchers to identify, define, and evaluate research goals [16], [26].
- Another approach to support guidelines scoping and planning is the use of model-driven for ''experiment protocols'' [16].
- For improving the understanding and reviewing of replication packaging, ontologies such as ExpertOntology are used [16].
- Finally, the newest proposals include the use of Software Configuration Management, Software Product Lines, Data Management tools for Replications, and Knowledge Sharing Models such as Nonaka-Takeuchi [1], [5], [14].

B. KNOWLEDGE MANAGEMENT GENERATED DURING AN SOFTWARE ENGINEERING EXPERIMENT

Knowledge is what should be communicated from an SE experiment; thus, it should be treated somehow to be stored, classified, and transmitted for internalization or externalization purposes. At present, knowledge is managed using software applications or web-oriented platforms [3], [29]. As knowledge refers to more complex information, it needs more complex tools that enable tracking changes and time [30]. In this way, several proposals have been developed, for example: FIRE (Framework for replication of experiments) [8], eSEE (Experimental Software Engineering Environment) [16], [31], ARREST (Application or Reproducible Research on Evaluation of Software Testing Techniques) [7], and TraceLab framework (for feature extraction techniques) [32]. In general, all these tools are integrated environments for domain-oriented software engineering experiments; therefore, they include the entire research workflow, a collection of artifacts, allow instantiation and distribution of experiments, and they are specialized [24].

On the other hand, other activities may be used beside software applications, in particular, to manage tacit knowledge. These activities focus on extracting knowledge directly from original experiments' researchers, or on collecting self-experience prior to a research execution [3], [8], [26]. However, even though those activities are an effective way to perform a replication, they are also the most expensive in terms of time and cost [12], [33]. For example, it is recommended to plan regular face-to-face workshops, execution of pilot studies, or simulation-based studies. Researchers report that through direct communication, it is easy to acquire detailed procedures, description of activities, and to understand essential decisions that affect the development of research, which is not included in a typical report. Pilot studies help researchers to identify gaps in knowledge so that they could be solved through interviews or meetings.

C. MOTIVATION FOR ENCOURAGE SOFTWARE ENGINEERING REPLICATIONS

Motivation to encourage SE replications is a less-explored concept in the literature. Guidelines recommend incorporating a motivation section where researchers give a brief insight into the research purpose, as a means of exploring justifications [3], [12]. Motivations could be as general as to corroborate or to validate results. Some studies analyze factors that demotivate or discourage the execution of replication activities, in which we can mention a large amount of effort to obtain all necessary data, the resources, and materials that may not be adequately acceded or allocated [12].

However, our exploration of motives was focused on finding the mechanism used to ''sell'' a research. In our previous work, we found that motivation in SE and ESE studies should include not only an academic point of view but also a social motivation [15]. Dittrich [34] mentions that many implications are involved in motivation when a project is formulated (e.g., research, academic, and industrial). However, in this regard, there are few explicit studies on the literature. For example, Juristo et al. [12] state that there is an evident lack of industry participation in empirical studies in SE, and Jedlitschka et al. [35] state that replications do not satisfy professional needs.

D. OVERALL RESULTS

The replication of empirical studies in SE is mainly supported by how researchers manage information and communicate knowledge to others. Specialized software can help researchers to manage specific application domains by the implementation of structured databases. Also, researchers can use more generic and informal mechanisms such as unstructured written documentation. However, as the information gets less structured, researchers lean on more direct ways of communication, such as interviews. In this manner, researchers (internally or externally) get flexibility for collecting specific information inexpensively. Additionally, researchers employ workshops for transferring practical experiences, especially in internal research teams.

However, the cost of conducting replications is still a problem. Although there are tools focused on specific domains, these usually require researchers to incur a learning process that not all of them are willing to face. The cost of direct extraction from researchers, used to gather information from interviews, is also a time-consuming activity. If direct extraction is not possible, researchers typically extract data from documentation or repositories, that usually are not well structured, adding complexity to replications. Of course, experienced research teams may overcome these difficulties, but they may focus on internal replications rather than external ones.

In summary, studies tend to cover the communication issues deeper than other aspects. Knowledge management is covered mostly in automated environments. Motivations are not explored at all; however, it is considered implicitly by the

TABLE 6. Summary of studies by research questions/aspect.

Aspects	%	Studies
Communication	80	LD0002, LD0008, LD0009, LD0010, LD0014, LD0031, LD0052, LD0073, LD0111, LD0161, LD0187, LD0189, LD0207, LD0379, LD0475, LD0511, LD0745, LD0758, LD0890, LD0912, LD0992, LD1014, LD1047, LD1126, LD1284, LD1291, LD1332, LC1392, LD1395, LD1477, LD1479, LD1546, LD1595, LD1663, LD1665, LD1698, LD1708, LD1875, LD1961
Knowledge	54	LD0002, LD0008, LD0009, LD0010, LD0014, LD0031, LD0052, LD0073, LD0111, LD0161, LD0187, LD0189, LD0207, LD0379, LD0475, LD0511, LD0745, LD0758, LD0890, LD0912, LD0992, LD1014, LD1047, LD1126, LD1284, LD1291, LD1332, LC1392, LD1395, LD1477, LD1479, LD1546, LD1595, LD1663, LD1665, LD1698, LD1708, LD1875
Motivation	49	LD0002, LD0008, LD0009, LD0010, LD0014, LD0031, LD0052, LD0073, LD0111, LD0161, LD0187, LD0189, LD0207, LD0379, LD0475, LD0511, LD0745, LD0758, LD0890, LD0912, LD0992, LD1014, LD1047, LD1126, LD1284, LD1291, LD1332, LC1392, LD1395, LD1477, LD1479, LD1546, LD1595, LD1663, LD1665, LD1698, LD1708, LD1875, LD1961, LD1966

use of standardized tools, technologies, and collaboration in both external replications and industry. Table 6 indicates the studies that cover each aspect.

V. DISCUSSION AND FUTURE WORKS

Numerous contributions and proposals in ESE have tried to improve and extend research reproducibility. Such proposals are intended to facilitate the transfer of knowledge in one or more research operations (data sources, retrieval methodology, raw data, extraction methodology, study parameters, processed dataset, analysis methodology, result data set). However, we found that beyond tools or methods, there are issues (e.g., communication, knowledge, motivation) that can impact on reproducibility. For example, research using a platform to support an experiment may distribute their procedures and results to others by a simple instantiation, while others may find difficulties in adopting this platform.

A. COMMUNICATION

Documentation is the most common resource to transfer information; however, it is not the most effective. Researchers rely on more direct and thus more informal, ways to transfer information such as face-to-face conversations and shared-folders. These mechanisms are costly and time-consuming but useful to acquire relevant information and detailed procedures. Several guides recommend how to include relevant information, but in some cases, researchers do not follow these guides adequately or do not accept them widely. We consider that information presented in documents is not enough to cover all points of interest for all researchers. Also, researchers do not have enough time to share information every time someone else needs it. These two statements may be why most replication of experiments tends to be internal. We believe that log is an excellent way to record every course of action and the decision-making process during an experiment execution, and guides focus solely on reports, which, for us, is just one deliverable of the experiment process.

B. KNOWLEDGE

Knowledge is the fundamental element produced in any research. It somehow has to be managed and stored. However, knowledge is diverse (quantitative or qualitative) so that setting automated procedures to analyze it is challenging. In this regard, knowledge-based systems are an integrated component in most experiment support platforms. However, platforms focus on managing either raw data or processed data. One bit of information that is reported as commonly omitted yet essential is tacit knowledge. As mentioned, the information is diverse in the field of research, though. As a result, we believe that it is not possible to build conventional and structure data without modifications and adaptations. We believe that a more dynamic and lose structure should be used to store specialized information and tacit knowledge, such as XML or JSON. Moreover, those schemes are compatible with most modern databases and NoSQL databases.

C. MOTIVATIONS

In contrast, motivation is the less-explored issue in the literature. The literature points to pressure for publishing original research since that is seen as more prestigious in the research community [36]. For this reason, we believe that motivation should be treated as more than a philosophical component in a report. Instead, motivation should be utilized to explore external needs (industry, academic, social) and call others to replicate and collaborate. In this sense, we argue that motivation should be used to explore both tangible and intangible benefits, suggest collaborative schemas of work, and identify strategies for addressing industry needs.

We believe that the needs we found in ESE such as managing knowledge and communication (internally or externally) are comparable to the software development processes needs. In the field of software development, widely-used tools and practices that have proven to be helpful are applied both in industry applications and open-source communities. These tools help developers to reproduce environments, conditions, or bugs to improve code and experiences. Based on our findings, we are confident that it is possible to address replication problems by incorporating or adapting tools used for the software development process. Research could improve communication, knowledge, and motivations by using tools that allow researchers to keep a log, in a lose structured way based on extensive markup language (XML), and by storing the information following an

TABLE 7. List of primary studies.

#	Code	Database	Year	Title
1	LD0002	ACM	2006	Evaluating Guidelines for Empirical Software Engineering Studies
2	LD0008	ACM	2008	A Framework for Software Engineering Experimental Replications
3	LD0009	ACM	2004	Knowledge-Sharing Issues in Experimental Software Engineering
4	LD0010	ACM	2016	ARRESTT: A Framework to Create Reproducible Experiments to Evaluate Software Testing Techniques
5	LD0014	ACM	2016	Experimentation with Dynamic Simulation Models in Software Engineering: Planning and Reporting Guidelines
6	LD0031	ACM	2005	Replicating Software Engineering Experiments: A Poisoned Chalice or the Holy Grail
7	LD0052	IEEE	2008	An Environment to Support Large Scale Experimentation in Software Engineering
8	LD0073	IEEE	2012	A TraceLab-based solution for creating, conducting, and sharing feature location experiments
9	LD0111	IEEE	2007	Extracting Information from Experimental Software Engineering Papers
10	LD0161	IEEE	2007	Modeling the Experimental Software Engineering Process
11	LD0187	IEEE	2017	Describing What Experimental Software Engineering Experts Do When They Design Their Experiments - A Qualitative Study
12	LD0189	IEEE	2002	Experience from replicating empirical studies on prediction models
13	LD0207	IEEE	2004	Infrastructure support for controlled experimentation with software testing and regression testing techniques
14	LD0379	IEEE	2014	Reproducibility, correctness, and buildability: The three principles for ethical public dissemination of computer science and engineering research
15	LD0475	IEEE	2013	Replication Data Management: Needs and Solutions – An Initial Evaluation of Conceptual Approaches for Integrating Heterogeneous Replication Study Data
16	LD0511	IEEE	2015	An Initiative to Improve Reproducibility and Empirical Evaluation of Software Testing Techniques
17	LD0745	IEEE	2005	Genetic algorithms to support software engineering experimentation
18	LD0758	IEEE	2013	Identifying Experimental Incidents in Software Engineering Replications
19	LD0890	IEEE	2002	Replicating software engineering experiments: addressing the tacit knowledge problem
20	LD0912	IEEE	2005	Reporting guidelines for controlled experiments in software engineering
21	LD0992	IEEE	2008	On the difficulty of replicating human subjects studies in software engineering
22	LD1014	IEEE	2011	Design Patterns in Software Maintenance: An Experiment Replication at UPM - Experiences with the RESER'11 Joint Replication Project
23	LD1047	IEEE	2011	Design Patterns in Software Maintenance: An Experiment Replication at Brigham Young University
24	LD1126	ScienceDirect	2018	Content and structure of laboratory packages for software engineering experiments
25	LD1284	Scopus	2018	Empirical studies omit reporting necessary details: A systematic literature review of reporting quality in model based testing
26	LD1291	Scopus	2017	Describing What Experimental Software Engineering Experts Do When They Design Their Experiments-A Qualitative Study
27	LD1332	Scopus	2017	A comparative study of model-driven approaches for scoping and planning experiments
28	LD1392	Scopus	2016	ARRESTT-A framework to create reproducible experiments to evaluate software testing techniques
29	LD1395	Scopus	2016	An External Replication on the Effects of Test-driven Development Using a Multi-site Blind Analysis Approach
30	LD1477	Scopus	2015	Investigations about replication of empirical studies in software engineering: A systematic mapping study
31	LD1479	Scopus	2015	Mechanisms to characterize context of empirical studies in software engineering
32	LD1546	Scopus	2014	A process-oriented environment for the execution of software engineering experiments
33	LD1595	Scopus	2013	Guidelines for reporting productivity studies - A review of the reproducibility of data envelopment analysis in the service sector
34	LD1663	Scopus	2012	Using configuration management and product line software paradigms to support the experimentation process in software engineering
35	LD1665	Scopus	2012	Design patterns in software maintenace: An experiment replication at UPM: Experiences with the RESER'11 joint replication project
36	LD1698	Scopus	2010	Replication of defect prediction studies: Problems, pitfalls and recommendations
37	LD1708	Scopus	2010	On the effectiveness of screen mockups in requirements engineering: Results from an internal replication
38	LD1875	Springer Link	2005	Supporting Controlled Experimentation with Testing Techniques: An Infrastructure and its Potential Impact
39	LD1961	Springer Link	2014	Reporting experiments to satisfy professionals' information needs
40	LD1966	Springer Link	2003	Empirical Studies in ESERNET

organizational framework guided by research needs. Other tools and approaches, such as GIT or KANBAN may be used to control and manage the decision-making process with no attachment to a particular technology. Our results describe the issues present in replication; hence, we could further explore the adaptation of software development tools in a future study.

VI. CONCLUSION

In this research, we analyzed the reproducibility of experiments in software engineering from a fresh perspective. Reproducibility is an aspect pursued in all scientific fields, including software engineering. ESE focuses on following strategies that propitiate reproducibility; however, there are constraints in reproduction tasks. Other studies evaluate data

characteristics or reproduction processes; but we analyze three mechanisms that should be present in an experiment: communication, knowledge, and motivation.

Communication involves all procedures, tools, and practices to internalizing and externalizing information. These can be formal or informal, but informal communication is the most-frequently used way to transmit information between researchers. Formal communication is usually limited by formats in reports, or are mainly focused on data and technical artifacts like in laboratory packages. In this regard, experience and decision-making processes are no supported by formal mechanisms. Informality makes the communication process expensive and time-consuming.

Knowledge management (referred to throughout this paper simply as knowledge) refers to the organization and storage of information and data for technology-oriented and human-oriented experiments. Robust platforms support most of the process of technology-oriented experiments; however, these are so specialized that they are not transferable to other domains. Hence, they are typically domain-centric and technology-centric and usually used in closed experiment groups. Human-oriented experiments do not rely on such platforms, and their activities are not automated. In both cases, informal communication mechanisms help researchers to collect tacit knowledge (such as experience and decision-making process) due it possibly not being stored correctly or organized.

Motivation examines ways to encourage experimenters to replicate. Motivations are the least-explored topic in the literature. Most studies allocate motivations to a section in a paper in which researchers explain why they executed an experiment. Motivations should be used to explore ways to encourage others to replicate, including social and industrial motivations in addition to academic ones.

Replication is still a significant concern; hence, many studies propose strategies and tools to improve reproducibility. We found that, despite the existence of many proposals, most of those focus on addressing the problems by interfering with the research process itself. Many platforms look promising but lack flexibility, which highlights the need for operational diversity of experiments. Our perspective argues that future proposals should focus on crosscutting concerns such as communication concerns, knowledge concerns, and motivational concerns.

APPENDIX

See Table. 7.

REFERENCES

[1] E. G. E. Gallardo, "Using configuration management and product line software paradigms to support the experimentation process in software engineering," in *Proc. Sixth Int. Conf. Res. Challenges Inf. Sci. (RCIS)*, May 2012, pp. 1–6.

[2] K. Y. Rozier and E. W. D. Rozier, "Reproducibility, correctness, and buildability: The three principles for ethical public dissemination of computer science and engineering research," in *Proc. IEEE Int. Symp. Ethics Sci. Technol. Eng.*, May 2014, pp. 1–13.

[3] C. V. de Magalhães, F. Q. da Silva, R. E. Santos, and M. Suassuna, "Investigations about replication of empirical studies in software engineering: A systematic mapping study," *Inf. Softw. Technol.*, vol. 64, pp. 76–101, Aug. 2015.

[4] L. Zhang, J.-H. Tian, J. Jiang, T.-J. Lui, M.-Y. Pu, and T. Yue, "Empirical research in software engineering—A literature survey," *J. Comput. Sci. Technol.*, vol. 33, no. 5, pp. 876–899, Sep. 2018.

[5] S. Biffl, E. Serral, D. Winkler, O. Dieste, N. Juristo, and N. Condori-Fernández, O. Dieste, and N. Juristo, "Replication data management: Needs and solutions—An initial evaluation of conceptual approaches for integrating heterogeneous replication study data," in *Proc. ACM/IEEE Int. Symp. Empirical Softw. Eng. Meas.*, Oct. 2013, pp. 233–242.

[6] F. G. D. Oliveira Neto, R. Torkar, and P. D. L. Machado, "An initiative to improve reproducibility and empirical evaluation of software testing techniques," in *Proc. IEEE/ACM 37th IEEE Int. Conf. Softw. Eng.*, May 2015, pp. 575–578.

[7] I. Da C. Araújo, W. O. Da Silva, J. B. De Sousa Nunes, and F. O. Neto, "ARRESTT: A framework to create reproducible experiments to evaluate software testing techniques," in *Proc. 1st Brazilian Symp. Syst. Automat. Softw. Test. (SAST)*, Sep. 2016, p. 1.

[8] M. G. Mendonça, J. C. Maldonado, M. C. D. Oliveira, J. Carver, S. C. Fabbri, F. Shull, G. H. Travassos, E. N. Höhn, and V. R. Basili, "A framework for software engineering experimental replications," in *Proc. 13th IEEE Int. Conf. Eng. Complex Comput. Syst. (ICECCS)*, Mar./Apr. 2008, pp. 203–212.

[9] M. Polanyi, *The Tacit Dimension*. Chicago, IT, USA: Univ. Chicago Press, 2009.

[10] J. Miller, "Replicating software engineering experiments: A poisoned chalice or the holy grail," *Inf. Softw. Technol.*, vol. 47, no. 4, pp. 233–244, Mar. 2005.

[11] M. Solari, "Identifying experimental incidents in software engineering replications," in *Proc. ACM/IEEE Int. Symp. Empirical Softw. Eng. Meas.*, Oct. 2013, pp. 213–222.

[12] N. Juristo and S. Vegas, "Design patterns in software maintenance: An Experiment replication at UPM–Experiences with the RESER'11 joint replication project," in *Proc. 2nd Int. Workshop Replication Empirical Softw. Eng. Res.*, Sep. 2011, pp. 7–14.

[13] J. Becker, D. Beverungen, D. Breuker, H. A. Dietrich, and H. P. Rauer, "Guidelines for reporting productivity studies–a review of the reproducibility of data envelopment analysis in the service sector," *Int. J. Services Oper. Manage.*, vol. 16, no. 3, pp. 407–425, 2013.

[14] F. Shull, M. G. Mendoncça, V. Basili, J. Carver, J. C. Maldonado, S. Fabbri, G. H. Travassos, and M. C. Ferreira, "Knowledge-sharing issues in experimental software engineering," *Empirical Softw. Eng.*, vol. 9, no. 1/2, pp. 111–137, Mar. 2004.

[15] C. Anchundia, "The replication of experiments in software engineering, a dilemma associated with knowledge generation," in *Proc. 2nd Ibero-Amer. Conf. Softw. Eng.*, 2017.

[16] W. Ferreira, M. T. Baldassarre, S. Soares, B. Cartaxo, and G. Visaggio, "A comparative study of model-driven approaches for scoping and planning experiments," in *Proc. 21st Int. Conf. Eval. Assessment Softw. Eng. (EASE)*, 2017.

[17] B. A. Kitchenham, D. Budgen, and P. Brereton, *Evidence-based Software Engineering and Systematic Reviews*. Boca Raton, FL, USA: CRC Press, 2016.

[18] H. Zhang, M. A. Babar, and P. Tell, "Identifying relevant studies in software engineering," *Inf. Softw. Technol.*, vol. 53, no. 6, pp. 625–637, Jun. 2011, doi: 10.1016/j.infsof.2010.12.010.

[19] S. Friese, J. Soratto, and D. Pires, "Carrying out a computer-aided thematic content analysis with ATLAS.ti," MMG, Göttingen, Germany, MMG Working Papers 18-02, Apr. 2018, vol. 18. [Online]. Available: https://www.mmg.mpg.de/62130/wp-18-02

[20] B. Cartaxo, A. Almeida, E. Barreiros, J. Saraiva, W. Ferreira, and S. Soares, "Mechanisms to characterize context of empirical studies in software engineering," in *Proc. ESELAW*, 2015, p. 281.

[21] M. L. McHugh, "Interrater reliability: The kappa statistic," *Biochemia Medica: !Casopis Hrvatskoga društva Medicinskih Biokemi!cara/HDMB*, vol. 22, pp. 276–282, Oct. 2012.

[22] A. R. Feinstein and D. V. Cicchetti, "High agreement but low Kappa: I. the problems of two paradoxes," *J. Clin. Epidemiol.*, vol. 43, no. 6, pp. 543–549, Jan. 1990.

[23] F. Ricca, G. Scanniello, M. Torchiano, G. Reggio, and E. Astesiano, "On the effectiveness of screen mockups in requirements engineering: Results from an internal replication," in *Proc. ACM-IEEE Int. Symp. Empirical Softw. Eng. Meas. (ESEM)*, Sep. 2010, p. 17.
[24] M. U. Khan, S. Iftikhar, M. Z. Iqbal, and S. Sherin, "Empirical studies omit reporting necessary details: A systematic literature review of reporting quality in model based testing," *Comput. Standards Inter.*, vol. 55, pp. 156–170, Jan. 2018.
[25] B. Kitchenham, H. Al-Khilidar, M. A. Babar, M. Berry, K. Cox, J. Keung, F. Kurniawati, M. Staples, H. Zhang, and L. Zhu, "Evaluating guidelines for reporting empirical software engineering studies," *Empirical Softw. Eng*, vol. 13, no. 1, pp. 97–121, Feb. 2008.
[26] B. B. N. De França and G. H. Travassos, "Experimentation with dynamic simulation models in software engineering: Planning and reporting guidelines," *Empirical Softw. Eng*, vol. 21, no. 3, pp. 1302–1345, Jun. 2016.
[27] L. S. D. S. Fonseca, C. B. Seaman, and S. C. B. Soares, "Describing what experimental software engineering experts do when they design their experiments–a qualitative study," in *Proc. ACM/IEEE Int. Symp. Empirical Softw. Eng. Meas. (ESEM)*, Nov. 2017, pp. 211–216.
[28] F. Shull, V. Basili, J. Carver, J. Maldonado, G. Travassos, M. Mendonca, and S. Fabbri, "Replicating software engineering experiments: Addressing the tacit knowledge problem," in *Proc. Int. Symp. Empirical Softw. Eng.*, Jun. 2003, pp. 7–16.
[29] M. Freire, G. Sizílio, E. Campos, U. Kulesza, and E. Aranha, "A process-oriented environment for the execution of software engineering experiments," in *Proc. Int. Conf. Product-Focused Softw. Process Improvement*, Dec. 2014, pp. 290–293.
[30] D. Cruzes, M. Mendonca, V. Basili, F. Shull, and M. Jino, "Extracting information from experimental software engineering papers," in *Proc.14th Int. Conf. Chilean Soc. Comput. Sci. (SCCC)*, Nov. 2007, pp. 105–114.
[31] G. H. Travassos, P. S. M. Dos Santos, P. G. Mian, P. G. M. Neto, and J. Biolchini, "An environment to support large scale experimentation in software engineering," in *Proc. 13th IEEE Int. Conf. Eng. Complex Comput. Syst. (ICECCS)*, Mar. 2008, pp. 193–202.
[32] B. Dit, E. Moritz, and D. Poshyvanyk, "A tracelab-based solution for creating, conducting, and sharing feature location experiments," in *Proc. 20th IEEE Int. Conf. Program Comprehension (ICPC)*, Jun. 2012, pp. 203–208.
[33] H. Do, S. Elbaum, and G. Rothermel, "Infrastructure support for controlled experimentation with software testing and regression testing techniques," in *Proc. Int. Symp. Empirical Softw. Eng. (ISESE)*, Aug. 2004, pp. 60–70.
[34] Y. Dittrich, "What does it mean to use a method? Towards a practice theory for software engineering," *Inf. Softw. Technol.*, vol. 70, pp. 220–231, Feb. 2016.
[35] A. Jedlitschka, N. Juristo, and D. Rombach, "Reporting experiments to satisfy professionals' information needs," *Empirical Softw. Eng.*, vol. 19, no. 6, pp. 1921–1955, Dec. 2014.
[36] M. Galster, D. Weyns, A. Tang, R. Kazman, and M. Mirakhorli, "From craft to science: The road ahead for empirical software engineering research," in *Proc. 40th Int. Conf. Softw. Eng. New Ideas Emerg. Results (ICSE-NIER)*, 2018, pp. 77–80

CARLOS E. ANCHUNDIA received the B.S. and M.S. degrees from Escuela Politécnica del Ejercito, Sangolquí, Ecuador, in 2007 and 2009, respectively, and the master's degree in business administration and the master's degree in information Technology from La Trobe University, Melbourne, Australia, in 2010 and 2011, respectively. He is currently pursuing the Ph.D. degree with Escuela Politécnica Nacional, Quito, Ecuador.

At present, he works at Escuela Politécnica Nacional as an Associated Professor of the Department of Informatics and Computer Science. His professional experience covers software development experience and management positions in industries related to laboratory and agronomy systems. His research interests include empirical software engineering, software development, and management methodologies.

EFRAÍN R. FONSECA C. received the Ph.D. degree, in 2014. He has ten years of IT industry experience as a Consultant. He is currently a Full Professor with the Universidad de las Fuerzas Armadas—ESPE, Ecuador. Among his research interests include the research process in empirical software engineering, research methods in empirical software engineering, object-oriented analysis, design and development of ontological representations in software engineering, information security, and new emerging technologies, such as the Internet of Things (IoT).