Reviewing Reproducibility in Software Engineering Research

André F. R. Cordeiro[a] and Edson Oliveira Jr[b]
Informatics Department, State University of Maringá, Colombo Avenue - 5790, Maringá, Brazil

Keywords: Reproducibility, Research Opportunities, Review, Software Engineering Research, Systematic Mapping.

Abstract: This paper presents a Systematic Mapping Study (SMS) on reproducibility in Software Engineering (SE), analyzing definitions, procedures, investigations, solutions, artifacts, and evaluation assessments. The research explores how reproducibility is defined, applied, and investigated, identifying several approaches and solutions. The final set of studies considered 25 primary studies, grouping the definitions of reproducibility into categories such as method, repeatability, probability, and ability. The application procedures are categorized into method, architecture, container, technique, framework, environment, notebook, toolkit, and benchmark. The investigation of reproducibility is analyzed through workflows, approaches, prototypes, methods, measures, methodologies, technologies, case studies, and frameworks. Solutions to reproducibility problems include environments, tools, benchmarks, initiatives, methodologies, notebooks, and containers. The artifacts considered include tools, environments, scenarios, datasets, models, diagrams, notebooks, algorithms, codes, representation structures, methodologies, containers, repositories, sequences, and workflows. Reproducibility assessment is performed using methods, experiments, measurements, processes, and factors. We also discuss future research opportunities. The results aim to benefit SE researchers and practitioners by providing an overview of organizing and providing reproducible research projects and artifacts, in addition to pointing out research opportunities related to reproducibility in the area.

1 INTRODUCTION

Software Engineering (SE) involves the use of technical knowledge, methods, and experience in a systematic way to design, implement, test, and document software(Garousi and Mäntylä, 2016).

Empirical research takes into account different problems related to experimentation, such as selection of empirical methods (Easterbrook et al., 2008; Zhang et al., 2018), guidelines for reporting experiments (Jedlitschka et al., 2008), samples and participants (Baltes and Ralph, 2022), and recorded research reports (Ernst and Baldassarre, 2023).

To address or minimize these issues, it is important to implement methods that support the achievement and validation of scientific discoveries (Juristo and Vegas, 2010). Reproducibility is one example and represents the ability to obtain a measurement performed by another research group, considering the same measurement procedure used to collect the original value under the same original operating conditions (Gonzalez-Barahona and Robles, 2023; ACM, 2023).

In the SE context, reproducibility suggests that an independent group can obtain the same results from the artifacts made available by the author of the original study (ACM, 2023). Given this, this paper presents a Systematic Mapping Study (SMS) to describe the state-of-the-art reproducibility in SE.

This paper is organized as follows: Section 2 presents the essential background; Section 3 presents the methodology Section 4 presents the results; Section 5 discusses the results obtained; Section 6 presents a set of research opportunities; and Section 7 concludes this paper.

2 BACKGROUND AND RELATED WORK

This section presents the theoretical foundation necessary to understand this study.

[a] https://orcid.org/0000-0001-6719-2826
[b] https://orcid.org/0000-0002-4760-1626

Cordeiro, A. F. R. and Oliveira Jr, E.
Reviewing Reproducibility in Software Engineering Research.
DOI: 10.5220/0013456000003929
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 27th International Conference on Enterprise Information Systems (ICEIS 2025) - Volume 2*, pages 364-371
ISBN: 978-989-758-749-8; ISSN: 2184-4992
Proceedings Copyright © 2025 by SCITEPRESS – Science and Technology Publications, Lda.

2.1 Reproducibility in Software Engineering Research

Different methods for verifying experimental findings can be considered in the context of SE, such as repetition, replication, and reproduction. Such methods are associated with the characteristics of repeatability, replicability, and reproducibility, respectively (ACM, 2023; Gonzalez-Barahona and Robles, 2023). Reproducibility is associated with the ability of a research group to obtain the same results as an original study, using the same artifacts considered in such a study (ACM, 2023).

Considering the literature about reproducibility, it is possible to observe studies structured in different ways to explore this characteristic. Reproducibility can be considered an evaluation criterion, as in the study of Rodríguez-Pérez et al. (2018). The study presents a Case Study based on a review related to the SZZ algorithm [1].

The reproducibility of studies in the area of Software Repository Mining is discussed in González-Barahona and Robles (2012). Elements that can be considered in the reproducibility of studies in the area are presented. A methodology for evaluating the reproducibility of studies is presented and illustrated. The methodology was re-evaluated in the study of Gonzalez-Barahona and Robles (2023).

2.2 Related Work

The literature presents secondary studies related to reproducibility in Evidence-Based Software Engineering (EBSE), reproducibility and replicability in Deep Learning (DL), tools, and practices to maximize experiments in SE.

The study of Li (2021) presents an investigation whose collected results suggest important problems related to reproducibility in EBSE. The problems are related to the reuse of search strings, non-reproduction of search activities, and non-reproduction of automatic searches described in the studies. A sample of 311 studies was considered. Based on the sample, different analytics were conducted to find possible causes for the problems raised.

The aspects of reproducibility and replicability were investigated in the study of Liu et al. (2021). The investigation was carried out to understand the importance of both characteristics during the application of Deep Learning (DL) in studies developed in SE. The study of Liu et al. (2021) reinforces the need for suitable artifacts to reproduce studies involving DL models. Among the results observed is the finding that only approximately 10% of the studies investigated have any relationship with reproduction or replication. It was also observed that approximately 62% of studies do not share artifacts that favor replication or reproduction.

An investigation was conducted to identify tools that maximize reproducibility in SE experiments in the study of Anchundia et al. (2020). Understanding the application of tools is also considered. Based on the results observed, the authors understand that reproducibility may be restricted to internal replication. Given the results, the authors suggest focusing on new alternatives to expand reproduction.

The SMS presented in this article seeks to understand reproducibility in a broader context, considering the SE area.

3 RESEARCH METHODOLOGY

The SMS followed the guidelines by Kitchenham et al. (2015) and Petersen et al. (2015).

This SMS is **aimed at** characterizing the reproducibility scenario in SE, in terms of definition, application, investigation, solutions, artifacts, and evaluation methods,**with the purpose** of evaluating the state of the art, **from the point of view of** SE researchers, **in the context of** primary studies.

We defined the following research questions related to the SE area:

- **RQ1.** How is reproducibility defined in the Software Engineering literature?
- **RQ2.** How is reproducibility applied in Software Engineering?
- **RQ3.** How is reproducibility investigated in Software Engineering?
- **RQ4.** What solutions are presented in the literature for the reproducibility issue in Software Engineering?
- **RQ5.** What artifacts are considered by the solutions to the reproducibility issue in Software Engineering?
- **RQ6.** How is reproducibility assessed in Software Engineering?

3.1 Search Process

We defined the following terms: **software engineering** and **reproduction** or **reproducibility** used to compose the final search string: **("software engineering") AND ("reproduction" OR "reproducibility")**, applied to following sources: IEEE, ACM Digi-

[1]https://github.com/topics/szz-algorithm

tal Library, SpringerLink, ScienceDirect, and Scopus. Results are summarized in Table 1.

Table 1: Summary of the Search Process.

Sources	**Returned Studies**
IEEE	487
ACM (Digital Library)	2
Springer	134
ScienceDirect	1
Scopus	8
Total	**632**

3.2 Selection Process

We initiated the selection process with the definition of inclusion (I) and exclusion (E) criteria for the primary studies, which are:

- (**I.1**:) explicit description of application procedures associated with reproducibility in software engineering;
- (**I.2**:) explicit description of investigation procedures associated with reproducibility in software engineering;
- (**E.1**:) study that is not in the field of software engineering;
- (**E.2**:) non-explicit citation of one or more application procedures associated with reproducibility in software engineering;
- (**E.3**:) non-explicit citation of one or more investigation procedures associated with reproducibility in software engineering;
- (**E.4**:) study not written in English (it difficult the dissemination and reproducibility);
- (**E.5**:) studies that have not been published in conferences or journals;
- (**E.6**:) duplicate studies;
- (**E.7**:) studies unavailable, even after contacting the authors; and
- (**E.8**:) secondary studies.

We screened all 632 primary studies by applying the inclusion and exclusion criteria. Table 2 presents a summary of papers from this process. Initially, 17 studies were selected for full reading. After reading, 11 studies (column "Selected") were selected for extraction.

We considered the selected studies for the snowballing process. The forward and reverse modes were performed (Wohlin, 2014). For the studies returned from the snowballing, we applied the inclusion/exclusion criteria. From 30 studies, we selected 14 (S12 through S25). Thus, a final set of studies was established for the extraction process, comprising 25 studies (Table 3).

Table 2: Summary of the Selection Process.

Sources	**Returned***	**Selected**	**Rejected**	**Duplicated**
IEEE	487	9	471	7
ACM	2	0	1	1
Springer	134	0	134	0
ScienceDirect	1	0	1	0
Scopus	8	2	5	1
Total	**632**	**11**	**612**	**9**

**from the search process (Section 3.1).*

Table 3: Final Set of Studies.

ID	**Title**	**Venue**	**Year**
S01	An Initiative to Improve Reproducibility and Empirical Evaluation of Software Testing Techniques	ICSE	2015
S02	1-2-3 Reproducibility for Quantum Software Experiments	SANER	2022
S03	Investigating The Reproducibility of NPM Packages	ICSME	2020
S04	Restoring Reproducibility of Jupyter Notebooks	ICSE	2020
S05	A Large-Scale Study About Quality and Reproducibility of Jupyter Notebooks	MSR	2019
S06	Variability and Reproducibility in Software Engineering: A Study of Four Companies that Developed the Same System	IEEE Trans. Soft. Eng.	2009
S07	Reproducibility of Environment-Dependent Software Failures: An Experience Report	ISSRE	2014
S08	Assessing and Restoring Reproducibility of Jupyter Notebooks	ASE	2020
S09	DOS Middleware Instrumentation for Ensuring Reproducibility of Testing Procedures	IEEE Trans. Instr. Measur.	2007
S10	On the reproducibility of empirical software engineering studies based on data retrieved from development repositories	Emp. Soft. Eng.	2012
S11	Using docker containers to improve reproducibility in software engineering research	ICSE	2016
S12	An introduction to docker for reproducible research	SIGOPS Op. Syst. Review	2015
S13	Analyzing multicore dumps to facilitate concurrency bug reproduction	ASPLOS	2010
S14	Automated localization for unreproducible builds	ICSE	2018
S15	Computing environments for reproducibility: Capturing the "Whole Tale"	Future Gen. Comp. Syst.	2019
S16	Constructing a reproducible testing environment for distributed Java applications	ICQS	2003
S17	Experiences from replicating a case study to investigate reproducibility of software development	RESER	2010
S18	Provbook: Provenance-based semantic enrichment of interactive notebooks for reproducibility	ISWC	2018
S19	QAOAKit: A Toolkit for Reproducible Study, Application, and Verification of the QAOA	QCS	2021
S20	QBugs: A Collection of Reproducible Bugs in Quantum Algorithms and a Supporting Infrastructure to Enable Controlled Quantum Software Testing and Debugging Experiments	Q-SE	2021
S21	Replication, reproduction, and re-analysis: three ways for verifying experimental findings	RESER	2010
S22	Reprozip: Using provenance to support computational reproducibility	TaPP	2013
S23	Root Cause Localization for Unreproducible Builds via Causality Analysis over System Call Tracing	ASE	2019
S24	State-based reproducible testing for CORBA applications	PDSE	1999
S25	Using Docker containers to improve reproducibility in software and web engineering research	ICWE	2016

3.3 Extraction Process

For each study, the following data were extracted: title, database, conference/journal, year, definition of reproducibility, application procedures, investigation procedures, study context, SE subarea, solution for the reproducibility problem, artifacts considered, and reproducibility assessment.

3.4 Study Data Availability

Data from this paper is available at https://zenodo.org/records/12510403.

4 RESULTS

This section presents the obtained results of our SMS.

4.1 RQ1 - Definitions of Reproducibility

Considering that the terms repetition and replication can be used as synonyms for reproduction (Anchundia et al., 2020), the premise is that different definitions are considered.

Table 4: Grouping of definitions of reproducibility.

Group	Definition	Study ID
Method	observed when a different team, with a different experimental setup, can confirm published results related to a previous experiment	S02
Repeatability	confirmation process related to a fact or of the conditions under which the same fact can be observed	S06
	process to establish a fact or conditions under which we can observe the same fact	S11
	process of establishing a fact, or of the conditions under which the same fact can be observed	S17
	process for establishing a fact or the conditions under which we can observe the same fact	S25
Probability	recurrence of a failure after repeated workload submissions	S07
Ability	start one study to be reproduced, in whole or in part, by an independent research team	S10
	recreate identical binaries without predefined build environments	S14
	conduct one test or experiment to be accurately reproduced by another person or team, working independently	S21

Only nine studies present reproducibility definitions. These definitions can be grouped by **Method, Repeatability, Probability, and Ability**. In terms of method, the study **S02** presents a definition that considers different teams and experimental configurations as conditions for reproduction to occur.

For the definitions based on repeatability, four studies consider this characteristic. For instance, in the study **S06**, reproducibility is described as the repeatability of the fact-finding process. A definition of reproducibility associated with the probability concept is observed in **S07**. In it, reproducibility is defined as the probability of recurrence of a failure. More specifically, the reproducibility of an experiment is defined as the ability to reproduce it by another person or independent team in **S21**.

4.2 RQ2 - Procedures for Reproducibility

Table 5 presents procedures focusing on reproducibility in SE.

Table 5: Grouping of procedures associated with reproducibility.

Group	Procedure	ID
Method	standardize, share methods and artifacts to evaluate software testing techniques; tools to support running experiments	S01
Architecture	distributed testing architecture with middleware instrumentation	S09
Container	use of containers to enable reproducibility in software engineering research	S11
	use of docker containers for reproducibility of research artifacts	S25
Technique	reproduction for programs running on multi-core platforms	S13
Framework	used to locate problematic files in non-reproducible builds	S14
Environment	deal with the reproducibility problem	S15
	reproducing a test on concurrent and distributed systems	S16
	creation of experiments in specific environments	S22
Notebook	a jupyter notebook extension to handle provenance over time	S18
Toolkit	a python toolkit for exploratory research	S19
Benchmark	used to facilitate the evaluation of new research and the reproducibility of previously published results	S20

Table 5 shows that of the 25 studies selected, 12 studies provided answers to this question. The application procedures can be grouped by **Method, Architecture, Container, Technique, Framework, Environment, Notebook, Toolkit, and Benchmark**. Study **S1** considers the application based on the sharing and standardization of methods and artifacts used to evaluate software techniques.

The application of reproducibility can be achieved through specific technologies and tools. In study **S11**, researchers utilized containers to enable reproducibility in SE research. The application of specific environments is one way to achieve reproducibility. In **S15**, a specific environment is described for achieving reproducibility.

4.3 RQ3 - Reproducibility Investigation

Table 6 presents the investigation procedures adopted in the studies.

Among the selected studies, 13 of them described research procedures related to reproducibility. The investigations included propositions, definitions, and applications to investigate reproducibility. These procedures can be grouped by **Workflow, Approach,**

Table 6: Grouping of reproducibility investigation procedures.

Group	Investigation Procedure	Study ID
Workflow	considered for reproducibility engineering	S02
Approach	procedure based on data tracking, version reconstruction, version comparison, and manual inspection activities	S03
	the proposition of a reproducible testing for components	S24
Prototype	design and implementation of a tool to restore the reproducibility of Jupyter notebooks	S04
Method	procedure based on data collection and analysis activities	S05
	definition of an experimental method based on specific factors	S07
	reproducibility study based on data collection activities, the configuration of the execution environment, experimental study, and recording of causes associated with non-reproducibility	S08
	to verify experimental findings in software engineering	S21
Measure	application of measures to assess reproducibility	S06
Methodology	proposition of a methodology for evaluating reproducibility	S10
Technology	use of docker technology to address technical challenges related to reproducibility	S12
Case Study	case study replications to the reproducibility of the software development process	S17
Framework	proposition of a framework for finding problems observed in non-reproducible builds	S23

Prototype, Method, Measure, Methodology, Technology, Case Study, and Framework.

Studies **S03** e **S24** present investigation procedures based on approaches. Study **S03** presents manual inspections and version comparisons. The implementation of a prototype to restore the reproducibility of Jupyter Notebooks is described in study **S04**.

Study **S05** outlines a method for investigating reproducibility. The method addresses specific questions, data collection, and analysis procedures. **S07** defines an experimental method that investigates reproducibility based on specific factors, such as memory occupancy, disk usage, and level of competition.

4.4 RQ4 - Solutions of Reproducibility

Table 7 lists various solutions for reproducibility considered by the selected studies, organized into similar groups.

Table 7: Solutions related to the reproducibility problem.

Solutions	Studies	Count
Environments, Tools and Benchmarks	S09; S15; S16; S19; S20; S22	6
Initiatives, Schemes, Methodologies, Techniques, and Approaches	S01; S02; S10; S13; S24	5
Jupyter Notebooks	S04; S08; S18	3
Containers	S11; S12; S25	3
Frameworks	S14; S23	2
Sets of Specific Results	S03; S05	2

Different studies present solutions regarding **Environments, Tools, and Benchmarks**. Study **S09** describes an environment for conducting tests in distributed applications. Study **S19** describes a toolkit built in Python to deal with important algorithms for the area of Quantum Approximate Optimization.

Regarding **Initiatives, Schemes, Methodologies, Techniques, and Approaches**, Study **S01** presents a solution based on a reproducible research **initiative** for evaluating software testing techniques. About **Jupyter Notebooks**, studies **S04** and **S08** introduce an automated approach to managing dependencies between elements. Study **S05** presents a set of good practices to promote the reproducibility of notebooks.

The use of **Containers** is also noted in the context of the presented solutions, specifically in studies **S11**, **S12**, and **S25**. Study **S11** describes the potential of containers to enhance the reproducibility of research in SE. Meanwhile, study **S12** explains how containers can facilitate reproducible research.

4.5 RQ5 - Considered Artifacts

Given the broad reproducibility context, it is crucial to determine which artifacts are applicable. In Table 8, eleven groups of artifacts are listed. These artifacts were observed alone or combined with other data.

Table 8: Artifacts considered.

Artifacts	Studies	Count
Tools, Environments, and Scenarios	S01; S06; S07; S14; S16; S22; S23; S24	8
Sets	S02; S09; S10; S12; S14; S17; S21	7
Models and Diagrams	S01; S15; S16; S24; S25	5
Data and Datasets	S01; S02; S10; S19	4
Notebooks	S04; S05; S08; S18	4
Algorithms and Codes	S13; S19	2
Representation Structures	S03; S18; S23	3
Methodologies	S01; S10	2
Containers	S11; S12	2
Repositories	S15; S20	2
Sequences and Workflows	S15; S24	2

Artifacts related to **Tools, Environments, and Scenarios** are observed in the studies **S01**, **S06**, **S07**, **S14**, **S16**, **S22**, **S23**, and **S24**. The study **S01** presents a **tool** for executing and analyzing experiments in the context of Software Testing. Different types of **sets** of reproducibility are explained by the studies **S02**, **S09**, **S10**, **S12**, **S14**, **S17**, and **S21**.

Notebooks are considered by the studies **S04**, **S05**, **S08**, and **S18**. A set of practices to improve the reproducibility rate of notebooks is proposed in study **S05**. **S18** presents a **Jupyter Notebook** extension to visualize provenance. **Algorithms and Codes** are presented in the studies **S13**, and **S19**.

The **Representation Structures** were considered by the studies **S03**, **S18**, and **S23**. The study **S03** details a framework for rebuilding NPM package versions. About **Methodologies**, the studies **S01** and **S10** present this kind of artifact. Study **S01** describes the methodology considered by a tool that runs and analyzes experiments.

Containers were used by the studies **S11**, and **S12**, which detail how containers can help with the reproducibility of SE studies. **Repositories** were used by the studies **S15**, and **S20**.

Sequences and Workflows were considered by the studies **S15**, and **S24**. Study **S15** presents a workflow based on an architectural model.

4.6 RQ6 - Reproducibility Assessment

Table 9 outlines the assessment procedures related to reproducibility. These procedures are categorized into **Method, Experiment, Process, Measurement, Process, and Factors**.

Table 9: Reproducibility assessment of the studies.

Group	Reproducibility Assessment	Study ID
Method	use of one method based on score	S01, S10
Experiment	experiment in which the tool receives as input a set of Jupyter notebooks selected at random. Each notebook is checked whether it is possible to reproduce it. In the end, the ratio between the reproductions performed and the total number of attempts is calculated. From this result, it is possible to obtain the percentage of reproductions carried out	S04, S08
Measurement	reproducibility rate for notebooks	S05
	reproducibility extent	S06
Process	process consisting of the following activities: repeating the standard package construction process and comparing the version of the generated package with the version of the published package	S03
Factors	assessment related to the reproducibility of failures	S07

The studies **S01** and **S10** present a method that seeks to establish a score for the degree of reproducibility of an empirical study. Experiments are also considered as assessment procedures, more specifically by the studies **S04**, and **S08**. Measurements are considered by the studies **S05** and **S06**. The evaluation of factors associated with reproducibility, described in study **S07**, can be considered an assessment procedure.

5 DISCUSSION OF RESULTS

This section discusses the results obtained in our study.

5.1 Discussion on Reproducibility Definition

In our analysis of the definitions of reproducibility in SE, the term “repetition” or “repeatability” is frequently used to explain reproduction. Only one definition explicitly refers to different experimental configurations. It is also important to consider specific definitions within the context of a study. Therefore, there appears to be some confusion regarding the terms repetition, replication, and reproduction.

In this context, **the most accurate definition of reproducibility is the ability to achieve the same measurement by a different team under identical operational conditions, following the same measurement procedure and using the same measurement system**ACM (2023); Gonzalez-Barahona and Robles (2023).

5.2 Discussion on Procedures for Reproducibility

The analysis of the procedures for reproducibility allowed us to group them into nine categories: Method, Architecture, Container, Technique, Framework, Environment, Notebook, Toolkit, and Benchmark. In addition, **several studies emphasize the need to share and standardize methods and artifacts and to use technologies such as containers to ensure the consistency of experiments**.

5.3 Discussion on Reproducibility Investigation

Various investigation procedures can be identified, including the definition and implementation of workflows, measures, methods, and processes. Inspections, implementations, and framework proposals are among the other identified procedures. Future research can be done to **simplify application and investigation processes, enhancing their reusability**.

5.4 Discussion on Artifacts Considered

Various artifacts were documented, including tools, environments, and scenario descriptions. Observations of these artifacts were made across different studies, encompassing data, datasets, and notebooks. The artifacts observed in the studies were categorized as shown in Figure 1, which illustrates the percentage associated with each group of artifacts.

The analysis of the frequency of artifacts reveals that the category of tools, environments, and scenarios was the most frequently cited, making up 19% of the total occurrences. Following this, data sets accounted for 16.7% of the artifacts.

Based on our findings, we recognize that several artifacts can affect reproducibility. These ar-

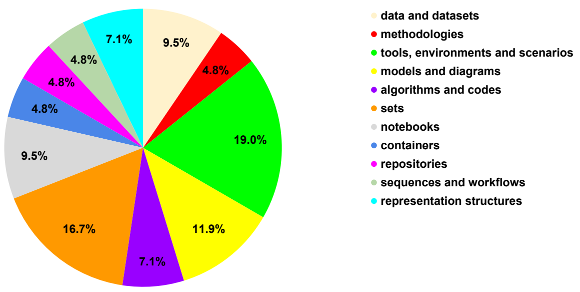

Figure 1: Reproducibility artifacts.

tifacts may arise from both the application and investigation procedures. We also emphasize **the importance of identifying artifacts that promote reproducibility early in the research process**.

5.5 Discussion on Reproducibility Assessment

A limited number of studies have evaluated reproducibility in SE. The findings suggest that reproducibility can be assessed through various methods, experiments, measurements, and processes.

This analysis emphasizes that **the assessment of reproducibility in SE is an emerging area** of study. It indicates **the necessity for more comprehensive and context-specific evaluation methods**.

6 OUTLINING RESEARCH OPPORTUNITIES

We discuss primary research opportunities based on the outcomes of our study. This discussion is particularly relevant, as we recognize that reproducibility must be considered a fundamental requirement for all SE research. Therefore, we list and discuss such opportunities as follows:

- **Opp.1.** Investigation of the reproducibility in subareas of SE, especially encompassing empirical studies;
- **Opp.2.** Investigation regarding the relationship between reproducibility and the Open Science (OS) movement in the context of SE.

The investigation of reproducibility in SE subareas (**Opp.1**), with special attention to empirical studies, can favor the reuse of knowledge. Certain subareas are more machine and algorithm-dependant thus they might have different methodological procedures from other subareas.

The relationship between Reproducibility and Open Science (**Opp.2**) in SE should be widely investigated. We understand that reproducibility is directly related to open science practices, especially for preservation, curation, and provenance. In addition, registered reports directly benefit reproducibility as they are concerned with the studies' protocols, previous to data collection (Ernst and Baldassarre, 2023). Studies have been initially carried out in this context, such as the study of Mendez et al. (2020), and OliveiraJr et al. (2024).

7 FINAL REMARKS

This paper analyzed the reproducibility landscape in SE, focusing on definitions, application and investigation procedures, solutions, artifacts, and evaluation methods. The analysis was based on 25 primary studies, allowing a comprehensive understanding of reproducibility in SE.

As directions for future work, we suggest investigating reproducibility in SE subareas, especially in empirical studies; and investigate the relationship between reproducibility and the Open Science movement in the context of SE.

ACKNOWLEDGMENTS

Edson OliveiraJr thanks CNPq/Brazil Grant #311503/2022-5.

REFERENCES

ACM (2023). Artifact review and badging. https://www.acm.org/publications/policies/artifact-review-and-badging-current.

Anchundia, C. E. et al. (2020). Resources for reproducibility of experiments in empirical software engineering: Topics derived from a secondary study. *IEEE Access*, 8:8992–9004.

Baltes, S. and Ralph, P. (2022). Sampling in software engineering research: A critical review and guidelines. *EMSE*, 27(4):94.

Easterbrook, S., Singer, J., Storey, M.-A., and Damian, D. (2008). Selecting empirical methods for software engineering research. *Guide to advanced empirical software engineering*, pages 285–311.

Ernst, N. A. and Baldassarre, M. T. (2023). Registered reports in software engineering. *Empirical Software Engineering*, 28(2):55.

Garousi, V. and Mäntylä, M. V. (2016). Citations, research topics and active countries in software engineering: A bibliometrics study. *CSR*, 19:56–77.

González-Barahona, J. M. and Robles, G. (2012). On the reproducibility of empirical software engineering studies based on data retrieved from development repositories. *EMSE*, 17(1):75–89.

Gonzalez-Barahona, J. M. and Robles, G. (2023). Revisiting the reproducibility of empirical software engineering studies based on data retrieved from development repositories. *IST*, 164:107318.

Jedlitschka, A., Ciolkowski, M., and Pfahl, D. (2008). Reporting experiments in software engineering. In *Guide to advanced empirical software engineering*, pages 201–228. Springer, New York.

Juristo, N. and Vegas, S. (2010). Replication, reproduction and re-analysis: Three ways for verifying experimental findings. In *RESER*.

Kitchenham, B. A., Budgen, D., and Brereton, P. (2015). *Evidence-based software engineering and systematic reviews*, volume 4. CRC press.

Li, Z. (2021). Stop building castles on a swamp! the crisis of reproducing automatic search in evidence-based software engineering. In *ICSE-NIER*, pages 16–20. IEEE.

Liu, C., Gao, C., Xia, X., Lo, D., Grundy, J., and Yang, X. (2021). On the reproducibility and replicability of deep learning in software engineering. *TOSEM*, 31(1):1–46.

Mendez, D., Graziotin, D., Wagner, S., and Seibold, H. (2020). Open science in software engineering. In *Contemporary empirical methods in software engineering*, pages 477–501. Springer, New York.

OliveiraJr, E., Madeiral, F., Santos, A. R., von Flach, C., and Soares, S. (2024). A vision on open science for the evolution of software engineering research and practice. In *FSE*, page 512–516. ACM.

Petersen, K., Vakkalanka, S., and Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *IST*, 64:1–18.

Rodríguez-Pérez, G., Robles, G., and González-Barahona, J. M. (2018). Reproducibility and credibility in empirical software engineering: A case study based on a systematic literature review of the use of the szz algorithm. *IST*, 99:164–176.

Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *EASE*, pages 1–10.

Zhang, L., Tian, J.-H., Jiang, J., Liu, Y.-J., Pu, M.-Y., and Yue, T. (2018). Empirical research in software engineering—a literature survey. *JCST*, 33:876–899.