FOSTER Taxonomy-based Open Science-Related Practices in Software Engineering: Review and Observations

André F. R. Cordeiro
Informatics Department
State University of Maringá
Maringá, PR, Brazil
cordeiroandrefelipe@gmail.com

Edson OliveiraJr
Informatics Department
State University of Maringá (UEM)
Maringá, PR, Brazil
edson@din.uem.br

ABSTRACT

The Open Science (OS) movement, primarily led by UNESCO, advocates for making research and its outcomes accessible to everyone. This movement has explored various issues across different fields, including reproducibility, experimentation, design science methodologies, and education, to provide access to research resources and discoveries. While the adoption of OS principles and practices in Software Engineering (SE) is increasing, it is still in its early stages, prompting further investigation into the topic. This paper aims to identify OS concepts, practices, and challenges related to their implementation in SE. To achieve this, we performed a Scoping Review Study of the literature. The main findings highlighted several OS-related practices in SE, categorized into Open Data, Open Platforms, Open Review, Open Access, Artifact Sharing, and Open Source. Additionally, challenges were noted in applying these practices. These findings culminated in a series of observations intended to support the adoption of OS in SE.

KEYWORDS

Open Science, Software Engineering, Openness, Practices, Open Research, Artifacts, Observations, Scoping Review

1 Introduction

The ISO/IEC/IEEE vocabulary defines Software Engineering (SE) as the application of scientific and technological knowledge to develop software [57]. SE has evolved over the last few decades, driven by scientific and technological research. The high impact and social contributions to our society support this evolution.

The SE area has incorporated various empirical research methods from fields such as Medicine and Social Sciences. These rigorous methods facilitate knowledge sharing in SE research, including systematic reviews [32] and experiments/quasi-experiments [64].

Different artifacts are produced, used, or reused in SE. Examples of artifacts include source code, binary code, scripts, software, configuration files, images, databases, documentation, datasets, and licenses [24, 51, 58]. Additionally, artifacts may include communicability directives [36], tools related to reproducibility [3], program execution models, data flow models [14], containers [10], methodologies [26], models and requirements [46], and conceptual models [22].

Creating or using an artifact is not a trivial task [7, 51]. Issues are observed during the artifact's elaboration or use, such as difficulties in finding specific information, broken links, and a lack of standardization [25, 41]. Different studies in the literature present observations for managing artifacts in SE [7, 25, 28, 34, 36, 37, 41].

The global Open Science (OS) movement, primarily spearheaded by UNESCO [60], promotes the accessibility of research and its outcomes to all individuals. It has examined various issues across different domains, including reproducibility [39], experimentation [6], design science methodologies [16], and education [20]. These efforts aim to enhance and open up research project resources and discoveries.

The OS movement is just gaining traction in SE, as indicated by the growing interest within the community. This interest is evident in the adoption of OS practices by the Empirical Software Engineering Journal and at conferences such as the International Conference on Software Engineering (ICSE), International Symposium on Empirical Software Engineering and Measurement (ESEM), Mining Software Repositories (MSR), International Conference on Software Analysis, Evolution and Reengineering (SANER), International Conference on Software Analysis, Evolution and Reengineering (SANER), and the Brazilian OpenScienSE workshop. Over the past few years, several OS policies for SE have been developed and utilized [17, 40].

The OS movement encompasses various related concepts and practices that aim to make research accessible to all individuals [52]. This paper defines these concepts and practices and identifies the challenges for OS in SE. Since this is a developing area in SE, we conducted a Scoping Review Study (SRS) [31, 44]. Based on the findings of the SRS, we proposed several observations for OS for the coming years.

This paper is organized as follows: Section 2 introduces essential concepts on SE, artifacts, and OS and discusses related work; Section 3 describes the methodology adopted for our SRS; Section 4 presents the SRS and its results based on the established research questions; Section 5 discusses the obtained results and provides several observations based on current and prospective actions of OS for SE; and Section 6 concludes this paper.

2 Background and Related Work

In this section, we present concepts that are essential to understanding our work and the related discussions.

2.1 Open Science

According to UNESCO [60], OS is a set of principles and practices that aim to make scientific research from all fields accessible to everyone for the benefit of scientists and society as a whole.

Different OS-related initiatives are observed in various studies. One initiative described in principles, processes, and badges is presented by Méndez Fernández et al. [42]. Understanding OS at an advanced level requires understanding the concept of openness.

This concept can be discussed in terms of transparency, access, participation, and democracy [54].

When analyzing the OS movement, several initiatives can be mentioned, such as those in Mosconi et al. [43] and Damasceno et al. [13], which focus on the sharing of artifacts.

The literature presents different taxonomies associated with OS [55]. In Pontika et al. [52], a taxonomy is presented with possible subareas of OS research. The European Commission developed the Facilitate OS Training for European Research (FOSTER) taxonomy [49].

Among FOSTER's objectives is the qualification of various stakeholders, including researchers, grantees, and repository managers. Figure 1 presents the FOSTER taxonomy.

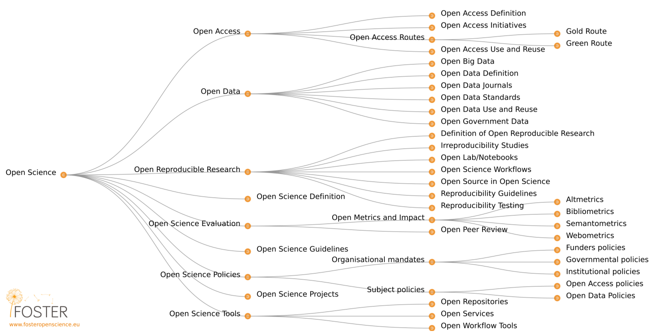

Figure 1: Open Science Taxonomy by FOSTER [23].

Such a taxonomy provides essential branches, including Open Access, Open Data, Open Reproducible Research, OS Evaluation, OS Policies, and OS Tools.

Investigations in the context of OS are getting the attention of the SE community Mendez et al. [40], OliveiraJr et al. [47]. Different aspects are addressed, including justifications for OS adoption, observations, challenges, and potential issues.

2.2 Open Science in Software Engineering

Adopting principles and practices associated with OS in SE is incipiently covered by Mendez et al. [40] and OliveiraJr et al. [47].

The SE community has undertaken various initiatives and studies within the context of OS.

In Méndez Fernández et al. [41], the Empirical Software Engineering Journal (EMSE) initiative involves principles, actions, and badges.

The International Conference on Software Engineering (ICSE) and other international SE top conferences have considered calls for the submission of artifacts for evaluation.

The International Conference on Software Analysis, Evolution, and Reengineering (SANER) features a specific track, called REproducibility Studies and NEgative Results (RENE), aimed at promoting the reproducibility of previous study results and publishing studies with negative or null results.

In Brazil, the Workshop on Open Science Practices for Software Engineering has been held in recent years to exchange experiences on practices and challenges related to OS.

Initiatives of OS also consider organizations and infrastructures, such as OpenAIRE [38], Figshare [21], and Zenodo [19].

2.3 Related Work

Vicente-Saez and Martinez-Fuentes [62] performed a systematic literature review to establish an integrated definition of OS. Seventy-five papers were analyzed, and OS descriptions are associated with knowledge, transparent knowledge, accessible knowledge, shared knowledge, and collaborative-developed knowledge. Such a work complies with the Open Science Framework (OSF)[1] taxonomy in many of the terms presented.

Other studies address reviews on OS-related subjects, but none of them are related to the SE area.

3 Research Methodology

In this work, we conducted a Scoping Review Study (SRS) [31, 44] of the literature.

Our goal is to identify primary studies that have reported or directly investigated OS-related practices, challenges, topics, and contexts in SE. We based our SRS on explicit and implicit practices from the main literature sources presented in Section 2.1 and the FOSTER Taxonomy.

We opted for a Scoping Review because we aim to broadly characterize OS and related practices as an emerging research topic in SE. The association between OS and SE is evident in studies that address one or more subareas of OS, as described in the FOSTER taxonomy.

4 The Scoping Review Study

This section presents the results of the performed Scoping Review Study (SRS).

4.1 Aim and Research Questions

This SRS is **aimed at analyzing** OS, **with the purpose** of characterizing the current implicit and explicit OS-related practices in the SE area, **from the point of view of** SE researchers, **in the context of** primary studies considering OS and SE. Therefore, we defined the following research questions:

- **RQ1: What explicit or implicit practices associated with OS have been observed in SE?** It aims to identify practices used in the context of OS present in SE and understand the benefits associated with these practices.
- **RQ2: What are the observed challenges in applying OS practices in SE?** We aim to examine the current challenges associated with implementing the identified OS practices in SE. Our interest in these challenges stems from the understanding that they could impede or even render the adoption of OS practices in SE projects challenging or unworkable.
- **RQ3: What OS topics have been most considered in SE?** We aim to identify the OS subareas most explored in SE. This information may contribute to the evolution of future SE studies based on OS practices.
- **RQ4: In what contexts it is observed practices associated with OS in SE?** It aims to understand the contexts that have facilitated the implementation of practices related to OS in SE. Based on the research conducted in OS, we

[1] https://osf.io/

recognize that these practices are relevant to both software development and research in SE.

For this SRS, we requested that two experienced SE researchers and research artifacts evaluation experts assess the protocol, research questions, and procedures. They act as artifact reviewer in the ICSE Artifact Evaluation track.

4.2 Search Process

We selected and tuned the string terms based on the FOSTER taxonomy and the work of Mendez et al. [40], thus choosing the following terms:

- **software engineering**; and
- **open science** - "open access", "open data", "open reproducible research", "open science evaluation", "open science policies", and "open science tools".

We composed and applied the terms in selected digital database sources (ACM, Springer, ScienceDirect, Scopus, Engineering Village, Web of Science, IEEE, and Wiley), using our final general search string:

"software engineering" **AND** ("open science" **OR** "open access" **OR** "open data" **OR** "open reproducible research" **OR** "open science evaluation" **OR** "open science policies" **OR** "open science tools")

Table 1 summarizes the search results per source. The initial set comprises 504 studies.

Table 1: Summary of the Search Process

Sources	Returned Studies
ACM (Digital Library)	398
Springer	90
ScienceDirect	5
Scopus	4
Engineering Village	3
Web of Science	3
IEEE	1
Wiley	0
Total	**504**

4.3 Selection Process

The selection process began by defining inclusion (I) and exclusion (E) criteria for the primary studies. Subsequently, we established the following criteria: inclusion - (**I.1**) mention of one or more practices or concepts associated with OS in SE, and (**I.2**) explicit mention of one or more challenges associated with OS in SE; exclusion - (**E.1**) study that is not in the field of SE; (**E.2**) non-explicit mention of one or more practices associated with OS in SE; (**E.3**) non-explicit mention of one or more challenges associated with OS in SE; (**E.4**) study not written in English, thus aiming at the dissemination and reproduction of this study; (**E.5**) study with four pages or less; (**E.6**) study not published in conferences or journals; (**E.7**) duplicate study; (**E.8**) study unavailable, even after contacting the authors; and (**E.9**) secondary or tertiary study.

We screened all 504 primary studies by applying the inclusion and exclusion criteria to their titles, keywords, and abstracts. A summary of the resulting papers from this process is shown in Table 2.

Table 2: Summary of the Selection Process.

Sources	Returned	Selected	Duplicated	Rejected
ACM	398	8	2	390
IEEE	1	0	0	1
Scopus	4	0	1	3
Springer	90	3	0	87
ScienceDirect	5	2	0	3
Wiley	0	0	0	0
Engineering Village	3	1	0	2
Web of Science	3	0	0	3
Total	**504**	**14**	**3**	**487**

from the search process (Section 4.2).

We also performed forward and reverse snowballing [5, 63]. We identified a second set of studies for analysis. We applied the same selection criteria to this second set, initially retrieving 80 studies without duplication. After applying the criteria, only one study met the selection criteria. Finally, we selected 15 primary studies, listed in Table 3.

Table 3: Final Set of Studies.

ID	Ref.	Title	Pub. Venue	Year
S01	[12]	Investigating Open Data Portals Automatically: A Methodology and Some Illustrations	dg.o	2018
S02	[1]	From Mashup Applications to Open Data Ecosystems	OpenSym	2014
S03	[11]	A Layered Architecture for Open Data: Design, Implementation and Experiences	ICEGOV	2018
S04	[59]	Open Data Framework for Sustainable Assessment in Software Forges	WIMS	2013
S05	[56]	Open Data-Driven Usability Improvements of Static Code Analysis and Its Challenges	EASE	2021
S06	[35]	IDEs Need Become Open Data Platforms (as Need Languages and VMs)	TOPI	2012
S07	[8]	Brazilian Government Open Data: Implementation, Challenges, and Potential Opportunities	dg.o	2014
S08	[15]	Exploiting Open Data to Improve the Business Intelligence	BDCA	2017
S09	[18]	Understanding peer review of software engineering papers	EMSE	2021
S10	[27]	A framework for systematic analysis of open access journals and its application in software engineering and information systems	Scientometrics	2014
S11	[58]	Understanding and improving artifact sharing in software engineering research	EMSE	2021
S12	[53]	A framework for software process deployment and evaluation	IST	2015
S13	[4]	Volunteering for Linked Data Wrapper maintenance: A platform perspective	Inf. Sys.	2019
S14	[30]	Enhancing software engineering curricula by incorporating open, data-driven planning methods	ASEE	2016
S15	[48]	From Opinions to Data-Driven Software R&D: A Multi-case Study on How to Close the 'Open Loop' Problem	EURO-MICRO	2014

4.4 Extraction Process

We started the extraction process based on the 15 primary studies previously selected.

We read all the papers and extracted the following metadata: title, source, publication venue, and publication year. Additionally, we identified practices, challenges, topics related to OS, and context.

4.5 Results

This section presents the results obtained with our SRS (Table 3).

4.5.1 Metadata Results. The publications span the years 2012 through 2021. In 2014, 2021, and 2018, we have four, three, and two studies, respectively. Each of the remaining years has one study. This fact indicates a lack of studies in this period, which is expected for such a new topic in SE.

Table 4 summarizes the contribution of each study per research question.

Table 4: Contributor Studies per Research Questions.

ID	RQ1	RQ2	RQ3	RQ4	Count
S01	**X**	**X**	**X**	**X**	04
S02	**X**	**X**	**X**	**X**	04
S03	**X**	**X**	**X**	**X**	04
S04	**X**	—	**X**	**X**	03
S05	**X**	**X**	**X**	**X**	04
S06	**X**	**X**	**X**	**X**	04
S07	**X**	**X**	**X**	**X**	04
S08	**X**	—	**X**	**X**	03
S09	**X**	—	**X**	**X**	03
S10	**X**	**X**	**X**	**X**	04
S11	**X**	**X**	**X**	**X**	04
S12	**X**	**X**	**X**	**X**	04
S13	**X**	**X**	**X**	**X**	04
S14	**X**	—	**X**	**X**	03
S15	**X**	—	**X**	**X**	03
Count	**15**	**10**	**15**	**15**	—

It is noted that all 15 studies contribute to RQ1 on OS practices. Additionally, they all contribute to RQ3, which is related to OS topics, and to RQ4, which is related to contexts. As for RQ2, it is worth noting that five studies do not explicitly address the challenges of applying OS practices in SE.

4.5.2 RQ1 - Practices Related to Open Science. Table 5 presents the observed practices in the final set of primary studies.

Several practices are observed in the context of **Open Data**. Regarding the software portal (S01, S07), open data is stored once it is established and can be accessed by different applications. The practice of open data portals for software development with different data sets suggests that development portals can consider an architecture based on three main elements: extractor, analyzer, and presenter. The extractor retrieves all the necessary data for the software from various sources. The analyzer evaluates the retrieved information and selects only the most important data for users. The presenter generates informative pages and presents them to users.

Table 5: Open Science-related Practices in Software Engineering.

OS Practices	Solution	Study ID	Practice ID
Open Data	software portal	S01	Pr.01
	open data mashups for data-as-a-service approach	S02	Pr.02
	architectural model based on open data	S03	Pr.03
	open data model	S04	Pr.04
	reuse of open data; usage of open collaborative data	S05	Pr.05
	open data portals for software development with different data sets	S07	Pr.06
	Linked Open Data and Ontologies	S08	Pr.07
	Linked Open Data for integrating data hosted by different tools	S12	Pr.08
	Linked Open Data for use or development of Linked Data Wrappers (LDWs)	S13	Pr.09
	software development based on data	S15	Pr.10
Open Platforms	uniform data structures; open architectures	S06	Pr.11
	open data clouds (DBPedia, Wikitology, Yago, WordNet, and others)	S08	Pr.12
Open Review	exemplar process guideline	S09	Pr.13
Open Access	framework for the analysis of open access journals	S10	Pr.14
Artifact Sharing	a plan for creating and sharing artifacts	S11	Pr.15
Open Source	leverage of Open Source Software communities	S14	Pr.16

The concept of open data mashups for the data-as-a-service approach is introduced in S02. Mashups can be developed as services. Various access technologies, such as Web Feeds, Web Interfaces, Representational State Transfer (REST), Simple Object Access Protocol (SOAP), and Application Programming Interfaces (APIs), can be considered.

An architectural model based on open data (S03) considers the layers of open data, open information, and open knowledge. Each layer is associated with specific operations, which can be accessed via service APIs. Operations are assigned to each layer. The data layer handles storage, catalog generation, and queries. The open information layer facilitates discussions, visualization, and co-creation of knowledge. The open knowledge layer focuses on sharing operations.

The open data model emphasizes the importance of considering key entities within their respective contexts. In the scenario outlined in S04, the model considered various entities, such as projects, datasets, users, and metrics. The reuse of open data and the utilization of open, collaborative data can be viewed from multiple angles, as discussed in Section 5. Software development based on data can be structured using a hypothesis-based model. The model presented in S15 encompasses activities such as resource backlog generation, resource selection and specification, implementation and instrumentation, gap analysis, hypothesis generation and selection, and alternative implementation.

The level of detail for using or developing Linked Data Warehouses (LDWs) (S08, S12, S13) involves considering various technologies, including the Resource Description Framework (RDF) and the RDF Query Language (SPARQL). Combining Linked Open Data (LOD) with ontologies establishes a cloud-based structure based on an ontology-driven open data architecture. This architecture acts as an intermediary between OD resources and data mining tools. Furthermore, when integrating data hosted by different tools, LOD considers that OD is linked and accessed using specific technologies, such as RDF.

We observe the use of "uniform data structures and open architectures" and "open data clouds" regarding **Open Platforms**. It is essential to prioritize standardization and readability to achieve uniform data structures and open architectures (S06). Regarding open data clouds (S08), a middleware-based architecture is utilized to leverage open data. The middleware is structured into layers that are related to Data Coordination, Processing, and API. The coordination layer handles the sending and receiving of data from a cloud structure for open data based on LOD and ontologies. In the processing layer, various calculations and operations are carried out. The API layer is responsible for utilizing the middleware.

In the context of **Open Review**, the "exemplar process guideline" (S09) aims to illustrate important aspects to consider during a review. It covers application, specific attributes, general quality criteria, anti-standards, acceptable deviations, and invalid and exemplary criticisms.

Open Access (S10) is related to the practice of a *framework for analyzing open access journals*, built on a set of attributes (bibliographic information, activity metrics, economics, accessibility, and predatory issues). Finally, observations are presented to readers of open-access journals.

Artifact Sharing (S11) is considered by the practice *a plan for creating and sharing artifacts*. The plan must consider different perspectives on the creators, mentors, reviewers, and organizers. For each perspective, observations were presented.

When leveraging **Open-Source** Software (S14) communities, the following actions can be taken to take advantage of their benefits: identify key software project repositories; search repositories for software projects based on specific attributes; select five open source projects similar in size, scope, and complexity; review the developer and discussion forums related to each of the five projects to determine experiences related to time, cost, and scope.

4.5.3 RQ2 - Challenges Related to Open Science in Software Engineering. Table 6 outlines the observed challenges.

Table 6: Observed Challenges.

OS Challenges	Challenge	Study ID	Challenge ID
Open Data	data set update	S01	Ch.01
	enable the programming at an end-user level for inexperienced developers; minimize maintenance efforts	S02	Ch.02
	deal with licensing, governance, and privacy	S05	Ch.03
	tools for dealing with open data	S12	Ch.04
	linked data wrapper maintenance	S13	Ch.05
Open Access	deal with data security (separation for private and public access)	S03	Ch.06
	preconceptions related to open access	S10	Ch.07
Artifact Sharing	changing of a cultural paradigm, related to data sharing	S06	Ch.08
	deal with multiple and decentralized data sources; lack standards for data publishing	S07	Ch.09
	deal with portability, maintenance and hosting questions; lack of standards and guidelines	S11	Ch.10

Regarding **Open Data**, the challenge of dataset update (S01) involves managing multiple data sets on a software portal. This might be complicated due to different management strategies, such as organizing different dataset versions into a single package using, for instance, the Comprehensive Knowledge Archive Network (CKAN), or managing data sets individually with Socrata.

When practitioners consider enabling programming at an end-user level (S02), it is essential to assess the potential for user profiles to utilize applications in unforeseen ways.

In addition to security, it is also essential to address licensing, governance, and privacy (S05). Licensing mainly relates to the use of open-source tools. Governance and privacy are connected to data sharing, even in the context of OD on software development (static code analyzers). In S05, the authors discuss the challenges of sharing development data and balancing the privacy of business information.

In S12, the authors emphasize the importance of having tools to handle OD. The study demonstrates the combined use of the tools Abreforjas (S04), EasyData/Rails, EasyData/Django, and Data adapter for publishing data from Enterprise Architect.

To handle the maintenance of linked data wrappers (LDW) (S13), it is suggested that when publishing an LDW, a maintenance strategy should be put in place to assure consumers of a commitment to the wrapper's evolution. The authors emphasize the importance of adaptive maintenance for new environments. It is crucial to ensure the evolution of APIs and Ontologies and update LOD to meet new requirements in new environments.

Regarding **Open Access**, when dealing with data security (separation for private and public access) (S03), different approaches can be observed in a layer-based architectural model. One of them is to remove or transfer one or more layers to an intranet to limit the audience for that layer. Another approach involves considering access to different layers based on the user's needs through an access control mechanism.

In the context of cultural paradigms, it is important to address preconceptions related to open access (S10). It is believed that these biases can be minimized by enhancing the transparency and quality of presentation on journal websites and services associated with open access.

For **Artifact Sharing** it is important to focus on efforts that contribute to changing the cultural paradigm related to data sharing to address the challenges observed in the context of open source in SE. In S06, the authors present evidence suggesting that data sharing is more of a cultural problem than a technical problem. In this context, it is essential to raise awareness about the fact that such data can be valuable to various groups of users, including developers and non-developers alike.

Regarding the challenge of dealing with multiple and decentralized data sources and the lack of standards for data publishing, it is believed that the integration between databases may not be adequate, despite the need for it. For example, in S07, the authors explain that there is inadequate integration between different databases. Regarding standardization, the authors comment on the lack of standards, especially for data sources and formats. Due to the absence of standards, different situations are observed. For instance, data about the same hospitals as those used by Web Services in Rio de Janeiro and Comma-Separated Values (CSV) in Recife may be published.

Other important considerations include addressing portability, maintenance, and hosting issues, as well as the lack of established standards and guidelines (S11). Portability issues may arise when the artifact cannot run on other computers due to incomplete experimental packages, unresolved dependencies, and source code issues.

Maintenance can become challenging when the original creator is no longer involved. Hosting problems may arise when finding storage for large artifacts becomes challenging, necessitating repositories with high storage capacities. The absence of standards and guidelines becomes evident when creators and reviewers are uncertain about how artifacts should be packaged and what standards should be followed.

4.5.4 RQ3 - Topics Related to Open Science. In addition to practices and challenges, it is important to understand the OS topics addressed in the selected studies.

Table 7 provides specific details for each study topic.

Table 7: Open Science Topics.

OS Topics	Studies
Open Data	S01; S02; S04; S06; S12
Open Government	S03; S07; S08
Open Source Software	S05; S14
Open Review	S09
Open Access	S10
Open Reproducible Research	S11
Open Platform	S13
Open Feedback	S15

Analyzing the studies in Table 3, we observed that they were directly related to at least one branch in the OS Taxonomy.

In the context of **Open Data**, various structures are associated with the findings presented in the studies. Examples include platforms, frameworks, and mashups. In S01, open data platforms such as CKAN, Socrata, ArcGIS Open Data, and OpenDataSoft are observed. The study presents a methodology that explores the possibility of approaching open data portals as software platforms. In S06, practices are suggested to promote the openness of software platforms, such as Integrated Development Environments (IDEs) and Virtual Machines (VMs), in terms of shared data and metadata availability.

In S04, an open data framework is proposed as part of an assessment approach that focuses on learning in collaborative software development projects. The framework involves working with data sources and representation models, taking assessment criteria into account. Furthermore, the framework described in S12 facilitates the creation of automated evaluation processes at the outset of each software project. This framework is explained in terms of its activities and models.

The mashup concept is introduced in S02 for use in software ecosystems with open data. An ecosystem based on mashups involves users, authors, and service providers as key actors. The implementation of a mashup application is described in terms of engines, clients, and technology characteristics.

In the context of **Open Government**, our selected studies offer solutions for architecture and repositories. In S03, an architectural model for designing software platforms is presented, focusing on publishing open data, information, and knowledge. This model defines various operations. In S08, a middleware-based architecture is introduced. This architecture is an intermediary between open data resources and data mining tools. It is possible to utilize the middleware through requests and responses.

Two Brazilian initiatives are described in S07, detailing the creation of repositories to store information about the cities of Rio de Janeiro and Recife. The study discusses the results achieved, challenges, and difficulties, emphasizing the potential for developing applications from the repository.

Regarding **Open Source**, in S05, the authors conducted a study on using and sharing static code analyzers and associated data. The results highlighted the need for developing solutions to share data, reduce risks, and facilitate the opening of tools like analyzers. Additionally, S14 presented an evaluation of practices that can be integrated into the study of SE to enhance student learning. They also introduced an approach for planning SE projects influenced by open-source software.

As for **Open Review**, a survey to comprehend review approaches is presented in S09. The survey focused on reviewers of SE studies. Issues concerning the acceptance, conduct, and reporting of review results were discussed.

Considering **Open Access**, a framework for analyzing open-access journals in SE and information systems is presented in S10. The framework was developed based on attributes such as bibliographic information, accessibility, and predatory issues. Additionally, the study provides readers with valuable observations.

Concerning **Open Reproducible Research**, one crucial aspect of OS is managing artifacts, which promotes open and reproducible research. A study examining the challenges of creating, sharing, and using artifacts is discussed in S11. The study's methodology involved collecting and analyzing primary and secondary data, and it also provided observations for creators, advisors, reviewers, and event organizers.

Considering **Open Platforms**, they can contribute to various aspects of the OS, such as open data and open research. In S13, a platform based on LDWs was constructed. An experimental framework was used to evaluate the platform.

Regarding **Open Feedback**, in S15, a model for data-driven development is developed based on open feedback. The model was developed after conducting a case study, which identified various issues, including the development of features without confirmed customer value, a lack of clarity in feature content, and a misalignment of the product with customer needs.

4.5.5 RQ4 - Contexts Related to Open Science and Software Engineering. Table 8 lists studies related to the contexts observed, including research, experience reports, development, and education.

Table 8: Contexts related to the Selected Studies.

OS Contexts	Studies
Research	S01; S02; S09; S10; S11
Experience Report	S03; S04; S06
Development	S05; S07; S08; S12; S13; S15
Education	S14

In five studies, the **research** context was addressed. A methodology was developed in S01 to investigate open data platforms. It involves searching for data portals in a specific area, developing a repository of data portals, identifying platforms, and collecting metadata. In S02, mashups are investigated, and the implementation of a mashup application is described as a result. S09 describes a survey conducted with reviewers of studies in SE. The survey aimed to document review practices. The construction of a framework is presented in Section 10, which is situated in an open-access context. Additionally, S11 addresses artifact management, considering the involvement of various actors who handle and oversee artifacts, including creators, advisors, and reviewers.

Three studies considered **experience reports**. In S03, an architectural model for software platforms that considers open data publishing is presented. The model establishes levels associated with open data, information, and knowledge and defines a set of operations for each level. Additionally, the study describes three application instances after defining the model. S04 outlines a learning assessment methodology and an open data framework for collaborative software development projects. It includes an analysis of evaluation examples and a case study. Additionally, in S06, two experience reports are presented, describing projects that enhance the openness of software platforms such as IDEs and VMs.

Six studies focused on the **development** context. In S05, the author investigates the use and sharing of static code analyzers in various scenarios, including program analyzers and the data generated by these analyzers. Another study, S07, describes two Brazilian initiatives related to OD repositories, which record city characteristics. These repositories can utilize data from various sources, and applications can extract and present different information to citizens and tourists in these cities. A middleware-based architecture is introduced in S08, which incorporates ontologies and LOD. The architecture provides an API for use. Another study, S12, discusses the creation of a framework to build automated evaluation mechanisms for software projects. The framework's evaluation is based on a case study. S13 presents the development of Linked Data wrappers (LDWs) and assesses their maintenance. Data-driven development is explored in S15, which involves creating a model to guide development based on a case study.

S14 addressed the **education** context by evaluating practices that can be integrated into the study of SE to promote student learning. It presented an approach to project planning.

4.6 Threats to Validity

Threats from this SRS are discussed based on the works of Ampatzoglou et al. [2] and Verdecchia et al. [61].

Regarding the **study selection validity**, we conducted multiple tests on the ACM and IEEE databases to mitigate the threat of constructing search strings. These databases are well-known in the field of SE and offer query mechanisms that can handle complex searches using various logical operators. Nevertheless, we acknowledge that omitting certain databases or filters could impact the selection of primary studies. We also decided to use only the term "Software Engineering" as the first cut of our general search string. This was done as a deliberate design aspect to limit the amount of literature. Therefore, this might have affected the validity and coverage of our SRS.

For **data validity**, we acknowledge that the sample size of the SRS may pose a potential threat to our work, particularly when identifying OS practices. However, the limited number of primary studies can be attributed to the unique characteristics of this SRS, which focuses on the intersection of OS-related practices and concepts and SE, a relatively new research topic in SE. The intersection was established from one or more OS subareas that were considered by studies carried out in the context of SE. The selected SRS data for extraction may compromise the validity of the SRS data. However, we tried to mitigate this by conducting a prior analysis of the published papers on the subject. This helped us select pertinent research questions to address. Publication bias could be considered a potential threat because we only conducted automatic searches in digital sources. To mitigate this, we also performed a manual snowballing process, which allowed us to identify an additional primary study. We attempted to mitigate a potential threat in the classification schemes by utilizing a well-established OS taxonomy to analyze primary studies in terms of possible subareas presented by the same.

Concerning the **research validity**, we aimed to ensure the repeatability of the SRS by following a standard protocol widely referenced in the literature [31, 33, 44, 45, 50]. The lack of comparable studies can be attributed to the nature of our SRS, which aims to link OS with SE, a recent topic in SE research. Generalization is a potential threat to this SRS. However, we attempted to mitigate this by testing and validating the main search string with researchers who have experience in OS/SE and secondary studies.

5 Discussion of Results and Observations

This section discusses the SRS results from Section 4.4 regarding OS-related practices, challenges, topics, and contexts. Based on these discussions, we prepared a set of observations to aid with the adoption of OS in SE. These observations aim to develop research and practice solutions to address the challenges identified.

5.1 Discussion and Observations Related to Practices

The practices listed in Table 5 were categorized according to OS topics, specifically Open Data, Open Platform, Open Review, Open Access, Artifact Sharing, and Open Source.

The practices related to Open Data are linked to models, LOD, data-driven software development, and portals. There is a specific model for Open Data (S04) and an architectural model based on Open Data (S03). Various options exist for utilizing LODs. One approach involves developing and maintaining LOD warehouses (LDWs) (S13), incorporating ontologies (S08), and integrating data stored in different locations (S12). In the context of data-driven development and portals, the reuse of Open Data is observed, as well as the use of collaborative open data (S05). In a service-oriented approach, mashups can be used to present data as a service (S02). When working with portals, one can use open data portals for software development, incorporating different datasets (S07). Another option is to develop a software portal (S01).

When dealing with Open Platforms, it is important to consider various levels of abstraction. This includes thinking about data structures, the platform architecture model (S06), and cloud structures for managing OD (S08). Regarding Open Review practices, guidelines associated with the review process were introduced

(S09). About OA, the analysis of OA journals was approached from a particular framework (S10).

The practice associated with artifact sharing involves creating a plan for sharing artifacts (S11). The practice associated with open source involves disseminating and using open-source communities, especially for teaching SE.

In the results section, we found that with the increasing availability of datasets, it is essential to explore and create solutions that can combine them to achieve specific goals. For example, a software ecosystem using the mashup concept could be a potential solution for this issue (S02). Based on this situation, we observe the following:

> **Obs.01:** *Different open data sources from various software projects should be combined using the Mashup concept.*

Most of the various artifacts [40] consist of data. Their adoption could benefit from standards already established in SE development, such as JavaScript Object Notation (JSON) and eXtensible Markup Language (XML). These standards could be used to describe specific data sets. As a result, we describe the following observation:

> **Obs.02:** *Data standards and data life cycle models should be analyzed to ensure adherence to different data sets and facilitate data curation.*

Open data is the artifact most cited in our SRS. Open Data Models are being developed as presented in Traverso-Ribón et al. [59]. Therefore, data models are essential for representing and organizing data sets. However, it is noted that such models are not enough to describe all the activities that can be performed with data. In addition to activities related to representation and organization, various other activities can be undertaken. For example, removing or concealing sensitive information could be considered [28]. This reflection influenced the following observation:

> **Obs.03:** *Open Data Models are crucial for organizing data. To simplify curation activities, they should be thoroughly researched and implemented in software engineering research data.*

The reuse of open data can be considered in SE. In addition to models and activities, it is essential to consider standards for open data (S05). Thus, we understand that it is not sufficient to consider representation, organization, and sharing. Planning for future data reuse in the medium and long term is necessary. Therefore, our observation is:

> **Obs.04:** *When defining and using open data, whether collaboratively or not, it is essential to consider models or standards that promote data reuse, thereby emphasizing data provenance. Reuse is a crucial aspect when creating and assessing open data models.*

Open data gives rise to various ideas for developing solutions. Different structures, such as architectures and platforms (S06), can be explored as artifacts during a software project. These structures can be utilized in solutions that promote software openness across different levels of abstraction [54]. This potential inspired us to make the following observation:

> **Obs.05:** *During software projects, abstractions that may favor the openness of software systems can be considered, such as open-platform architectural models.*

Depending on the activity or purpose of the software, it can be beneficial to incorporate data into the process. This can aid in enhancing the developed solutions or evaluating the steps involved in the development. The data can be stored on portals (S07). In this context, the following observation has been prepared:

> **Obs.06:** *Open data should be deposited in portals and repositories to facilitate their sharing and preservation, particularly in data-driven development.*

The interoperability of data and datasets needs further investigation (S08). This is particularly crucial in Distributed Systems, which involve geographically distant elements. In this context, the following observation is provided:

> **Obs.07:** *Establishing interoperability between data or data sets should be prioritized, especially when they are stored in different or geographically distant repositories. This can be facilitated by implementing procedures related to data provenance.*

In an SE project, various artifacts are used. Some are reused, while others are created from scratch. Documents, such as research projects or experiment plans, can outline both scenarios. According to Timperley et al. [58], such a document can be described as a plan for creating and sharing artifacts, as is done in one of the preregistration types of OSF. Given the potential advantages of this plan, the following suggestion is proposed:

> **Obs.08:** *The plan for creating and sharing artifacts and the long-term strategy for sharing and evaluating them should be established as a section of the research project.*

In addition, the mashups as described in Aaltonen et al. [1], it has been observed that open-linked data can assist in data integration (S12). Depending on the context, a combined solution between mashups and linked data can be considered to address the integration problem. Based on this understanding, the following observation was made:

Obs.09: *The use of Linked Open Data and mashups might facilitate the integration of data from different sources, such as repositories or tools.*

5.2 Discussion and Observations Related to Challenges

The challenges listed in Table 6 were organized by OS topics, specifically Open Data, Open Access, and Artifact Sharing.

The challenges of Open Data can be understood from a management perspective at various levels, including maintenance and updating, other aspects, and tools. Maintenance involves efforts used with LDWs (as discussed in Azpeitia et al. [4]) and minimizing effort in scenarios that aim to facilitate development for inexperienced professionals (as mentioned in Aaltonen et al. [1]). It is important to address the challenge of updating datasets to keep them relevant to a given context (as noted in Correa et al. [12]). As for other aspects, licensing, governance, and privacy are important considerations (as indicated in Söderberg et al. [56]). When it comes to tools, there's a need to develop and use solutions that facilitate the manipulation of Open Data (as described in Ruiz-Rube et al. [53]).

The topic related to Open Access involves both cultural and technical challenges. Culturally, it is important to address prejudices associated with Open Access (S10). From a technical standpoint, it is crucial to ensure data security to establish a clear separation between public and private access (S03).

The challenges related to sharing artifacts are considered from cultural and management perspectives. It is important to change the cultural mindset around sharing artifacts, particularly data (S06). Regarding artifact management, there is a need to enhance processes that involve handling decentralized data sources. The absence of data publication standards can negatively impact this regard (S07. Addressing issues such as portability, maintenance, and hosting, as well as the lack of standards and guidelines, presents another notable challenge (S11).

As shown in the results section (Table 6), numerous studies describe significant challenges for OS in SE.

One of the challenges identified is updating data (S11). Updating data is another important aspect to consider during the data life cycle [9, 29] definition. Updating activities and representation, organization, and sharing activities should be considered. These activities should be clearly described in documents. Studying this specific challenge leads us to provide the following observation:

Obs.10: *Establish data update procedures that should be clearly outlined in a research Data Management Plan (DMP).*

Managing data is an ongoing challenge that needs to be addressed in defining the data life cycle (S11). Upon further consideration, it becomes apparent that cleaning up and updating data are specific maintenance tasks. It is important to include these activities in a Data Management Plan. This realization has led us to make the following observation:

Obs.11: *Consider data maintenance efforts from the early stages of research design. Actions related to these efforts should be clearly outlined in a Data Management Plan.*

Specific security measures were determined based on the type of data or data set, including its structure, recording, and evaluation (S11). These measures should also be included in a Data Management Plan:

Obs.12: *Include detailed assessment procedures for the security level of each data set in the Data Management Plan.*

When managing research data or artifacts, it is important to consider challenges such as licensing, governance, and privacy (S11). These aspects should be addressed in an Artifact Management Plan, as recommended below:

Obs.13: *Consider including licensing, governance, and privacy aspects in an Artifact Management Plan.*

Artifact sharing standardization presents a significant challenge that should be addressed in Artifact Management Plans (S11):

Obs.14: *Research artifact sharing standardization procedures and techniques should be explored to facilitate adoption and reuse.*

5.3 Discussion and Observation Related to Topics

In our analysis of the topic of **open data**, we found that platforms are used, mashup applications are developed to integrate users, authors, and service providers, methodology and framework are established for collaborative software development, data and metadata sharing is necessary, and automated evaluation is considered in software projects. Platforms can be structured to provide open data portals (S01). Mashup applications can be developed to integrate users, authors, and service providers (S02). Methodology and framework are used in the project for developing collaborative software (S04). The need for sharing data and metadata was noted during software development (S06). A framework in software projects considered automated evaluation (S12).

In the context of **Open Government**, solutions related to models, use of repositories, and middleware are explored. One architectural model was developed for designing software platforms that prioritize the open publication of data (S03). Additionally, open data repositories were created during the construction of applications to make municipal data and information accessible (S07). A specific middleware was implemented to construct architectures for intermediary actions between open data sources and data mining tools (S08).

Solutions related to data sharing and project planning in the realm of **Open Source Software** were found. Data sharing was viewed as an alternative practice to be integrated into the early stages of software development, making tasks such as program analysis easier (S05). Project planning was also recognized as an approach to studying SE (S14).

Concerning the **Open Review**, specific guidelines have been established for reviewing studies in SE (S09).

In terms of **Open Access**, a framework was developed to analyze Open Access journals in SE and Information Systems (S10).

Open Reproducible Research observations on sharing artifacts are presented in Timperley et al. [58], such as sharing management, to assist artifact creators and reviewers.

Regarding the **Open Platform** topic, the development of solutions for data integration was verified. LDWs were developed from LODs and Ontologies (S13).

As for **Open Feedback**, a model was developed to assist in the development of data-driven software (S15).

The solutions outlined in studies related to OS within the context of SE have influenced the formulation of the following observation:

> **Obs.15:** *Further explore how Open Science topics should benefit various research projects and artifacts, particularly Open Reproducible Research and protocols, as these topics are crucial to the field.*

5.4 Discussion and Observation Related to Contexts

The results identified various contexts in which OS has been practiced, including research, Experience Reports, Development, and Education.

In the **research** context, we observe the presentation of methodology, models, interviews and surveys, frameworks, and scenarios for sharing and use of artifacts, as follows:

- open science methodologies have considered utilizing open data platforms. The methodology involves searching for data portals, developing a data portal repository, identifying platforms, and collecting metadata (S01);
- a mashup model was first developed and then implemented (S02);
- an interview and a survey were conducted to understand the review approaches used in evaluating studies in SE (S09);
- a framework is constructed and applied to the analysis of open access journals (S10);
- understanding of sharing and using artifacts was developed through collecting, analyzing data, and making observations (S11).

Studies related to the context of **experience reports** present the application of a model, a case study, and reports on development projects as follows:

- an architectural model is presented and utilized in the development of software platforms for open data publishing (S03);
- a case study carried out is related to an evaluation methodology and an open data framework, used in collaborative software development projects (S04);
- reports on development projects were prepared based on the experiences gained with the development of the plugin Codemap and the Moose platform. Such developments showed the need to work with open data (S06).

In the context of **development**, exploratory research, an application associated with repositories, consideration of different structures, a case study, solution validation with experienced professionals, and a guidance model were identified as follows: exploratory research was carried out on the development of static code analyzers (S05); a study on open data repositories which provides an extractor and a parser (S07); an application based on different structures including a middleware, an ontology, and LOD (S08); a case study carried out to evaluate a framework for software process deployment and evaluation, during its use (S12); a solution validation was performed for the maintenance of LDWs (S13); and a guidance model is defined for data-driven development (S15).

In the **education** context, an exploratory and documentary research was carried out to evaluate SE course curricula (S14).

The practices described in various contexts have influenced the presentation of the following observation:

> **Obs.16:** *Consult Open Science practices already used in the context of Software Engineering when planning the methodology of a study.*

6 Final Remarks

The paper addressed the growing importance of Open Science within the field of Software Engineering by conducting a comprehensive scoping review of existing literature.

Based on the insights gained from the scoping review, we formulated a set of observations aimed at fostering the adoption of OS principles and practices within the SE community. These observations provide valuable guidance for researchers and practitioners seeking to enhance the transparency, accessibility, and reproducibility of their work.

The study acknowledges the incipient nature of OS adoption in SE, suggesting avenues for future research to further explore and address the identified challenges, as well as investigate how different OS topics can benefit various research projects and artifacts, particularly Open Reproducible Research.

Ultimately, this work contributes to the ongoing dialogue surrounding OS in SE and provides a foundation for its continued growth and integration within the discipline.

ARTIFACT AVAILABILITY

Artifacts are available at **https://doi.org/10.5281/zenodo.16749465**, licensed under the **Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License**.

ACKNOWLEDGMENTS

The authors thank the support of PROAP/CAPES, Brazil, code 001.

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

REFERENCES

[1] Timo Aaltonen, Tommi Mikkonen, Heikki Peltola, and Arto Salminen. 2014. From mashup applications to open data ecosystems. In *Proceedings of The International Symposium on Open Collaboration*. ACM, https://dl.acm.org/doi/abs/10.1145/2641580.2641599, 1–8.

[2] Apostolos Ampatzoglou, Stamatia Bibi, Paris Avgeriou, Marijn Verbeek, and Alexander Chatzigeorgiou. 2019. Identifying, categorizing and mitigating threats to validity in software engineering secondary studies. *Information and Software Technology* 106 (2019), 201–230.

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

[4] Iker Azpeitia, Jon Iturrioz, and Oscar Díaz. 2020. Volunteering for Linked Data Wrapper maintenance: A platform perspective. *Information Systems* 89 (2020), 101468.

[5] Deepika Badampudi, Claes Wohlin, and Kai Petersen. 2015. Experiences from using snowballing and database searches in systematic literature studies. In *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering*. ACM, https://dl.acm.org/doi/abs/10.1145/2745802.2745818, 1–10.

[6] Sebastian Baltes and Paul Ralph. 2022. Sampling in software engineering research: A critical review and guidelines. *Empirical Software Engineering* 27, 4 (2022), 94.

[7] Victor R Basili, Marvin V Zelkowitz, Dag IK Sjøberg, Philip Johnson, and Anthony J Cowling. 2007. Protocols in the use of empirical software engineering artifacts. *Empirical Software Engineering* 12, 1 (2007), 107–119.

[8] Kellyton S. Brito, Marcos Antônio da Silva Costa, Vinicius Cardoso Garcia, and Silvio Romero de Lemos Meira. 2014. Brazilian government open data: implementation, challenges, and potential opportunities. In *Proceedings of the 15th annual international conference on digital government research*. ACM, https://dl.acm.org/doi/abs/10.1145/2612733.2612770, 11–16.

[9] Digital Curation Centre. 2023. Digital Curation Centre Data Lifecycle Model. https://www.dcc.ac.uk/.

[10] Jürgen Cito and Harald C Gall. 2016. Using docker containers to improve reproducibility in software engineering research. In *Proceedings of the 38th international conference on software engineering companion*. ACM, https://dl.acm.org/doi/abs/10.1145/2889160.2891057, 906–907.

[11] Gennaro Cordasco, Delfina Malandrino, Donato Pirozzi, Vittorio Scarano, and Carmine Spagnuolo. 2018. A layered architecture for open data: Design, implementation and experiences. In *Proceedings of the 11th International Conference on Theory and Practice of Electronic Governance*. ACM, https://dl.acm.org/doi/abs/10.1145/3209415.3209466, 371–381.

[12] Andreiwid Sheffer Correa, Pär-Ola Zander, and Flavio Soares Correa Da Silva. 2018. Investigating open data portals automatically: a methodology and some illustrations. In *Proceedings of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age*. ACM, https://dl.acm.org/doi/abs/10.1145/3209281.3209292, 1–10.

[13] Carlos Diego Nascimento Damasceno, Isotilia Costa Melo, and Daniel Struber. 2021. Towards Multi-Criteria Prioritization of Best Practices in Research Artifact Sharing. *arXiv preprint arXiv:2109.02304* 1 (2021), 1–5.

[14] Jack B Dennis. 2017. Principles to Support Modular Software Construction. *Journal of Computer Science and Technology* 32 (2017), 3–10.

[15] Oussama El Bacha, Othmane Jmad, Younes El Bouzekri El Idrissi, and Nabil Hmina. 2017. Exploiting Open Data to Improve the Business Intelligence & Business Discovery Experience. In *Proceedings of the 2nd international Conference on Big Data, Cloud and Applications*. ACM, https://dl.acm.org/doi/abs/10.1145/3090354.3090382, 1–6.

[16] Emelie Engström, Margaret-Anne Storey, Per Runeson, Martin Höst, and Maria Teresa Baldassarre. 2020. How software engineering research aligns with design science: a review. *Empirical Software Engineering* 25 (2020), 2630–2660.

[17] N. A. Ernst and M. T. Baldassarre. 2023. Registered reports in software engineering. *Empirical Software Engineering* 28 (2023), 1–11. https://doi.org/10.1007/s10664-022-10277-5

[18] Neil A Ernst, Jeffrey C Carver, Daniel Mendez, and Marco Torchiano. 2021. Understanding peer review of software engineering papers. *Empirical Software Engineering* 26 (2021), 1–29.

[19] European Organization For Nuclear Research and OpenAIRE. 2013. Zenodo. https://doi.org/10.25495/7GXK-RD71

[20] Andrea Farnham, Christoph Kurz, Mehmet Ali Öztürk, Monica Solbiati, Oona Myllyntaus, Jordy Meekes, Tra My Pham, Clara Paz, Magda Langiewicz, Sophie Andrews, et al. 2017. Early career researchers want Open Science. *Genome biology* 18 (2017), 1–4.

[21] Figshare. 2024. Figshare. https://figshare.com/

[22] Shayne Flint. 2009. A Conceptual Model of Software Engineering Research Approaches. In *2009 Australian Software Engineering Conference*. IEEE, IEEE, https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5076645, 229–236.

[23] Foster. 2022. Foster Open Science Taxonomy. https://www.fosteropenscience.eu/foster-taxonomy/open-science?page=6.

[24] E. Frachtenberg. 2022. Research artifacts and citations in computer systems papers. *PeerJ Comput Sci* 7 (2022), e887. https://doi.org/10.7717/peerj-cs.887

[25] Parisa Ghazi and Martin Glinz. 2017. Challenges of working with artifacts in requirements engineering and software engineering. *Requirements Engineering* 22, 3 (2017), 359–385.

[26] Jesús M González-Barahona and Gregorio Robles. 2012. On the reproducibility of empirical software engineering studies based on data retrieved from development repositories. *Empirical Software Engineering* 17, 1 (2012), 75–89.

[27] Daniel Graziotin, Xiaofeng Wang, and Pekka Abrahamsson. 2014. A framework for systematic analysis of open access journals and its application in software engineering and information systems. *Scientometrics* 101 (2014), 1627–1656.

[28] Mark Grechanik, Collin McMillan, Tathagata Dasgupta, Denys Poshyvanyk, and Malcom Gethers. 2014. Redacting sensitive information in software artifacts. In *Proceedings of the 22Nd International Conference on Program Comprehension*. ACM, https://dl.acm.org/doi/abs/10.1145/2597008.2597138, 314–325.

[29] Sarah Higgins. 2008. The DCC curation lifecycle model. *International Journal of Digital Curation* 1, 1 (2008), 1–7.

[30] John Lalit Jagtiani and Neal Lewis. 2016. Enhancing Software Engineering Curricula By Incorporating Open, Data-Driven Planning Methods. In *2016 ASEE Annual Conference & Exposition*. ASEE PEER, https://peer.asee.org/enhancing-software-engineering-curricula-by-incorporating-open-data-driven-planning-methods, 1–11.

[31] Hanan Khalil and Andrea C. Tricco. 2022. Differentiating between mapping reviews and scoping reviews in the evidence synthesis ecosystem. *Journal of Clinical Epidemiology* 149 (2022), 175–182. https://doi.org/10.1016/j.jclinepi.2022.05.012

[32] Barbara Kitchenham and Stuart Charters. 2007. Guidelines for performing systematic literature reviews in software engineering.

[33] Barbara Ann Kitchenham, David Budgen, and Pearl Brereton. 2015. *Evidence-based software engineering and systematic reviews*. Vol. 4. CRC Press, CRC Press.

[34] Shriram Krishnamurthi. 2013. Artifact evaluation for software conferences. *ACM SIGSOFT Software Engineering Notes* 38, 3 (2013), 7–10.

[35] Adrian Kuhn. 2012. Ides need become open data platforms (as need languages and vms). In *2012 Second International Workshop on Developing Tools as Plug-Ins (TOPI)*. IEEE, IEEE, https://ieeexplore.ieee.org/abstract/document/6229807, 31–36.

[36] Adriana Lopes, Tayana Conte, and Clarisse Sieckenius de Souza. 2020. Exploring the Directives of Communicability for Improving the Quality of Software Artifacts. In *19th Brazilian Symposium on Software Quality*. ACM, https://dl.acm.org/doi/abs/10.1145/3439961.3439974, 1–10.

[37] Adriana Lopes Damian, Clarisse Sieckenius de Souza, and Tayana Conte. 2021. Directives of Communicability for Software Artifacts. In *XX Brazilian Symposium on Software Quality*. ACM, https://dl.acm.org/doi/abs/10.1145/3439961.3439974, 1–9.

[38] Paolo Manghi, Alessia Bardi, Claudio Atzori, Miriam Baglioni, Natalia Manola, Jochen Schirrwagen, Pedro Principe, Michele Artini, Amelie Becker, Michele De Bonis, et al. 2019. The OpenAIRE research graph data model. *Zenodo* 1, 1 (2019), 1–23.

[39] Wolfgang Mauerer and Stefanie Scherzinger. 2022. 1-2-3 reproducibility for quantum software experiments. In *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, IEEE, https://ieeexplore.ieee.org/abstract/document/9825822, 1247–1248.

[40] Daniel Mendez, Daniel Graziotin, Stefan Wagner, and Heidi Seibold. 2020. Open science in software engineering. In *Contemporary empirical methods in software engineering*. Springer, New York, 477–501.

[41] Daniel Méndez Fernández, Wolfgang Böhm, Andreas Vogelsang, Jakob Mund, Manfred Broy, Marco Kuhrmann, and Thorsten Weyer. 2019. Artefacts in software engineering: a fundamental positioning. *Software & Systems Modeling* 18, 5 (2019), 2777–2786.

[42] Daniel Méndez Fernández, Martin Monperrus, Robert Feldt, and Thomas Zimmermann. 2019. The open science initiative of the Empirical Software Engineering journal. *Empirical Software Engineering* 24, 3 (2019), 1057–1060.

[43] Gaia Mosconi, Qinyu Li, Dave Randall, Helena Karasti, Peter Tolmie, Jana Barutzky, Matthias Korn, and Volkmar Pipek. 2019. Three gaps in opening science. *Computer Supported Cooperative Work (CSCW)* 28, 3 (2019), 749–789.

[44] Zachary Munn, Micah DJ Peters, Cindy Stern, Catalin Tufanaru, Alexa McArthur, and Edoardo Aromataris. 2018. Systematic review or scoping review? Guidance for authors when choosing between a systematic or scoping review approach. *BMC medical research methodology* 18, 18 (2018), 1–7. https://doi.org/10.1186/s12874-018-0611-x

[45] Elisa Yumi Nakagawa, Kátia Romero Felizardo Scannavino, Sandra Camargo Pinto Ferraz Fabbri, and Fabiano Cutigi Ferrari. 2017. *Revisão sistemática da literatura em engenharia de software: teoria e prática*. Elsevier Brasil, https://repositorio.usp.br/item/002944453.

[46] Joaquín Nicolás and Ambrosio Toval. 2009. On the generation of requirements specifications from software engineering models: A systematic literature review. *Information and Software Technology* 51, 9 (2009), 1291–1307.

[47] Edson OliveiraJr, Fernanda Madeiral, Alcemir Rodrigues Santos, Christina von Flach, and Sergio Soares. 2024. A Vision on Open Science for the Evolution of Software Engineering Research and Practice. In *ACM International Conference on the Foundations of Software Engineering*. ACM, New York, NY, USA, 512–516. https://doi.org/10.1145/3663529.3663788

[48] Helena Holmström Olsson and Jan Bosch. 2014. From opinions to data-driven software r&d: A multi-case study on how to close the'open loop'problem. In *2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications*. IEEE, IEEE, https://ieeexplore.ieee.org/abstract/document/6928783, 9–16.

[49] Foster Open Science. 2022. Foster Open Science. https://www.fosteropenscience.eu.

[50] Kai Petersen, Sairam Vakkalanka, and Ludwik Kuzniarz. 2015. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and software technology* 64 (2015), 1–18.

[51] Rolf-Helge Pfeiffer. 2020. What constitutes software? An empirical, descriptive study of artifacts. In *Proceedings of the 17th International Conference on Mining Software Repositories*. ACM, https://dl.acm.org/doi/abs/10.1145/3379597.3387442, 481–491.

[52] Nancy Pontika, Petr Knoth, Matteo Cancellieri, and Samuel Pearce. 2015. Fostering open science to research using a taxonomy and an eLearning portal. In *Proceedings of the 15th international conference on knowledge technologies and data-driven business*. ACM, https://dl.acm.org/doi/abs/10.1145/2809563.2809571, 1–8.

[53] Iván Ruiz-Rube, Juan Manuel Dodero, and Ricardo Colomo-Palacios. 2015. A framework for software process deployment and evaluation. *Information and Software Technology* 59 (2015), 205–221.

[54] Daniel Schlagwein, Kieran Conboy, Joseph Feller, Jan Marco Leimeister, and Lorraine Morgan. 2017. “Openness” with and without Information Technology: a framework and a brief history. , 297–305 pages.

[55] Lúcia da Silveira, Priscila Machado Borges Sena, Nivaldo Calixto Ribeiro, Jorge Polanco Cortés, Remedios Melero, Juliana Fachin, Andrea Mora-Campos, Julio Santillán-Aldana, Daniel Fernando Piraquive-Piraquive, Fabiano Couto Corrêa da Silva, et al. 2023. Taxonomy of Open Science: revised and expanded. *Encontros Bibli* 28 (2023), e91712.

[56] Emma Söderberg, Luke Church, and Martin Höst. 2021. Open Data-driven Usability Improvements of Static Code Analysis and its Challenges. In *Evaluation and Assessment in Software Engineering*. ACM, https://dl.acm.org/doi/abs/10.1145/3463274.3463808, 272–277.

[57] ISO/IEC/IEEE International Standard. 2017. ISO/IEC/IEEE International Standard - Systems and software engineering–Vocabulary. *ISO/IEC/IEEE 24765:2017(E)* 1, 1 (2017), 1–541. https://doi.org/10.1109/IEEESTD.2017.8016712

[58] Christopher S Timperley, Lauren Herckis, Claire Le Goues, and Michael Hilton. 2021. Understanding and Improving Artifact Sharing in Software Engineering Research. *Empirical Software Engineering* 26 (2021), 1–41.

[59] Ignacio Traverso-Ribón, Iván Ruíz-Rube, Juan Manuel Dodero, and Manuel Palomo-Duarte. 2013. Open data framework for sustainable assessment in software forges. In *Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics*. ACM, https://dl.acm.org/doi/10.1145/2479787.2479817, 1–8.

[60] United Nations Educational, Scientific and Cultural Organization UNESCO. 2022. UNESCO Recommendation on Open Science. https://en.unesco.org/science-sustainable-future/open-science/recommendation.

[61] Roberto Verdecchia, Emelie Engström, Patricia Lago, Per Runeson, and Qunying Song. 2023. Threats to validity in software engineering research: A critical reflection. *Information and Software Technology* 164 (2023), 107329. https://doi.org/10.1016/j.infsof.2023.107329

[62] Ruben Vicente-Saez and Clara Martinez-Fuentes. 2018. Open Science now: A systematic literature review for an integrated definition. *Journal of business research* 88 (2018), 428–436.

[63] Claes Wohlin. 2014. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering*. ACM, https://dl.acm.org/doi/abs/10.1145/2601248.2601268, 1–10.

[64] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in software engineering*. Springer Science & Business Media, Germany.