A Conceptual Framework and Recommendations for Open Data and Artifacts in Empirical Software Engineering

Runeson, Per; Söderberg, Emma; Höst, Martin

Published in:
WSESE '24: Proceedings of the 1st IEEE/ACM International Workshop on Methodological Issues with Empirical Studies in Software Engineering

DOI:
10.1145/3643664.3648206

2024

Document Version:
Publisher's PDF, also known as Version of record

A Conceptual Framework and Recommendations for Open Data and Artifacts in Empirical Software Engineering

Per Runeson
per.runeson@cs.lth.se
Lund University
Lund, Sweden

Emma Söderberg
emma.soderberg@cs.lth.se
Lund University
Lund, Sweden

Martin Höst
martin.host@mau.se
Malmö University
Malmö, Sweden

ABSTRACT

Background. Open science aims to improve research accessibility, replicability, and consequently its quality. Empirical software engineering entails both data and artifacts, which may be shared more or less openly, to support transparency. However, the trade-offs involved in balancing the openness against integrity and secrecy concerns need methodological guidance. **Aim.** We aim to derive such advice, based on our own experiences from a research project, in the field of gaze-assisted code reviews – the Gander case. **Method.** We draw on literature about open data and artifacts in socio-technical research. Next, we describe our case project and derive a conceptual framework of steps in research data analysis and artifact development, using our data and artifacts as illustrating examples. **Results.** The conceptual framework contains 1) a categorization of humans involved as participants and their concerns, 2) four steps for data analysis, each resulting in corresponding data and meta-data, and 3) three steps of artifact distribution, matching different levels of openness. We derive a preliminary set of recommendations for open science practices for data and artifacts. **Conclusion.** The conceptual framework has proven useful in summarizing and discussing data and artifacts in the studied case project. We envision that the framework and recommendations will provide a foundation for further advancement of open science research practices in empirical, socio-technical software engineering.

CCS CONCEPTS

• **Software and its engineering** → *Integrated and visual development environments*;

KEYWORDS

Open science, Open research, Open data, FAIR data, Open artifacts, Socio-technical software engineering

ACM Reference Format:
Per Runeson, Emma Söderberg, and Martin Höst. 2024. A Conceptual Framework and Recommendations for Open Data and Artifacts in Empirical Software Engineering. In *International Workshop on Methodological Issues with Empirical Studies in Software Engineering (WSESE '24), April 16, 2024, Lisbon, Portugal.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3643664.3648206

WSESE '24 , April 16, 2024, Lisbon, Portugal

ACM ISBN 979-8-4007-0567-0/24/04
https://doi.org/10.1145/3643664.3648206

1 INTRODUCTION

Open science is brought forward as a means to increase research quality and efficiency, e.g., by making it easier to reproduce and replicate studies, and to democratize research. Open science or *open scientific knowledge*, as defined by UNESCO[1], includes open scientific publications, open research data, open educational resources, open source software, and open hardware. In open science, the transparency is expected to increase reviewability, reproducibility and replicability, and thereby the quality of research. The open access and data is expected to contribute to democratization and to increase efficiency by avoiding double work. This development is, for example, manifested in the Empirical Software Engineering journal open science initiative [21] and the ACM artifact evaluation policy[2]. In empirical software engineering, experimental material have been made available as replication or laboratory packages [30] to some extent.

Software engineering (SE) is a research and practice domain, which is fundamentally socio-technical [31]. This implies that research and practice involves humans, and data generated by and about humans. As many SE research challenges come with scaling, it is ideally being conducted in real-world industrial contexts [2], involving real-world products and business. These characteristics lead to a series of ethical and legal concerns for SE research, which conflict with the aims of open science, at first sight. Not only has research to protect *personal data and integrity*, *company data and secrets* have also to be safeguarded. Further, since researchers and companies may have commercial interests in software tools, they may be reluctant to sharing their artifacts.

However, open does not mean out of control. According to the European Horison 2020 Program Guidelines on FAIR Data[3], data should be "as open as possible and as closed as necessary" – open in order to foster the reusability and to accelerate research, but at the same time they should be as closed as needed to safeguard the privacy of the subjects as well as legitimate secrecy concerns for commercial entities. Recently sharpened legislation on personal data protection – e.g. the GDPR in Europe[4] – and ethical approval – e.g. changed interpretation of the ethical approval act in Sweden – clearly illustrate these conflicting interests.

Mendez et al. discuss open science in SE as a means to improve repeatability, replicability, and reproducibility of research [19]. They

[1] https://doi.org/10.54677/UTCD9302
[2] https://www.acm.org/publications/policies/artifact-review-and-badging-current
[3] FAIR – Findable, Accessible, Interoperable, Reusable Data https://ec.europa.eu/research/participants/data/ref/h2020/grants_manual/hi/oa_pilot/h2020-hi-oa-data-mgt_en.pdf
[4] The General Data Protection Regulation (EU 2016/679, GDPR) is a European Union regulation on information privacy in the European Union (EU) and the European Economic Area (EEA). https://en.wikipedia.org/wiki/General_Data_Protection_Regulation

acknowledge the above mentioned concerns, namely personal data and company data protection, and "the conflict between anonymity and confidentiality on one side, and openness on the other" [19, p.493]. However, they offer little practical guidance for research projects in this balancing act. Further, based on observation of the citation of method guidelines for empirical software engineering (e.g., [14, 24, 33]), authors seem to prefer guidelines tailored to SE, rather than using their general counterparts.

We therefore share our experiences and considerations on open science in socio-technical SE from a research project on *gaze-driven assistance in code review* as a case study – the Gander case. The project covers interview and survey data from industry practitioners, eye-tracking data from human subjects, and open source tools for experimentation. From our experiences, as well as literature on data sharing and software reuse, we derive a conceptual framework, which aims to structure analysis and communication about data and artifact openness and thereby guide future open science in SE. We also provide our own project data and artifacts as an illustrating example and derive preliminary recommendations.

We first discuss relevant literature on open data in socio-technical research in Section 2. Then, we introduce the case project in Section 3 and analyze open science aspects that appeared in the project, rendering our conceptual framework in Section 4. In Section 5 we map our project to the conceptual framework and present recommendations for SE researchers. Section 6 concludes the paper.

2 OPEN DATA AND ARTIFACTS IN SOCIO-TECHNICAL RESEARCH

2.1 Outcomes from Research Studies

Empirical data collection techniques in software engineering field studies were categorized by Lethbridge et al. into three degrees: (1) direct involvement of software engineers – human-enacted inquisitive and observational techniques (e.g. interviews), (2) indirect involvement of software engineers – technically enabled observations (e.g. eye-tracking), and (3) study of work artifacts only (e.g. code) [17]. All the three categories have similar concerns with respect to both the personal and company data protection needs. For example, eye-tracking and the code may reveal inefficient work practices by the individual, and interviews and commercial code may reveal company secrets. Generally, companies involved in empirical studies are concerned with protection of their data, preventing researchers from opening the data to the research community.

A special type of SE contexts is open source software (OSS) development. Then, software artifacts are open by default, as well as personal data in the form of contributors' names or pseudonyms and contact information. As a consequence of the easy access, SE research on OSS projects is popular, although only some aspects are comparable for corporate SE [23]. Also, there is a risk of revealing personal identities and data if interviewees are selected via open source software, for example, from an OSS community [20].

A special branch of empirical software engineering is the mining of software repositories (MSR) studies, where data is collected from open development repositories. González-Barahona and Robles proposed a method for assessing the reproducibility of MSR studies [10], which they validated a decade later [11]. Since the raw data in MSR studies come from open sources, their method primarily focuses on the transparency of analysis methods and procedures.

In the research field of Open data ecosystems [25], the concern about sharing data between commercial actors is addressed, as well as open data from governmental and other public sources. Based on a survey of the literature, Enders et al. explored factors to take into account when deciding the degree of openness for data, i.e. selective revealing of data [6]. These factors are (1) Coreness (closeness to the core business), (2) Currentness (how recent is the data), (3) Extent (volume of data), (4) Granularity (level of detail), (5) Interoperability (e.g, standardized formats), and (6) Quality (fit for purpose).

Since the data collected in SE research varies across these degrees and factors, a conceptual framework and recommendations for open data must take them into account.

2.2 Qualitative and Quantitative Data

Socio-technical research in SE embraces both quantitative and qualitative data. "Quantitative data is more exact, while qualitative data is 'richer' in what it may express." [24, p.15]. Quantitative data are easier to summarize, e.g. through means and dispersion measures, or statistical distributions. They may also be easier anonymized, since the identities are mostly connected to the meta data and context descriptions, rather than the data itself. In some cases, removing or changing the scale of data may help addressing secrecy issues. Also quantitative data is more often encoded and does not reveal opinions, work tasks, etc. that make it possible to identify the data source.

There is, as far as we have found, no discussion on open qualitative research data in the literature specifically for SE, beyond open science policies for journals and conferences. However, the topic is discussed in social science and psychology, including a multitude of perspectives on the feasibility of open qualitative research data.

For example, Chauvette et al. [3] are critical to open data for epistemological, methodological and ethical reasons. *Epistemologically*, qualitative data are tightly linked with the context, so changing context make them meaningless, they claim. *Methodologically*, the reflexivity in the qualitative analysis requires the researcher to be part of the data collection to be close enough to the studied phenomenon. *Ethically*, issues related to confidentiality and anonymity prevent open data.

Other researchers, e.g. Field et al. [7], are more neutral regarding open, qualitative data. They acknowledge the above mentioned problems, but weigh them against potential benefits. Among this, participants may want to share their data for the greater good, and improved research efficiency may speak for open data. They continue nuancing the issue, by proposing to share codebooks with "a list of codes with associated definitions, examples and descriptions" that may add to the transparency and replicability of research.

Finally, Joyce et al. [13], are proponents for openness and argue that "there are several notable benefits to sharing qualitative research data". They claim that most concerns can be addressed by good policies and practices of data repositories. This position is supported by DuBois et al. [5] who report that data sharing has been an established practice in the research field of Conversation Analysis for over three decades. They also offer practical guidelines for sharing qualitative data [4].

There are conceptual differences, related to open data, between qualitative and quantitative data. However, there are also differences between research domains in their principles and practices for open data. SE consequently has to find a position as a community.

2.3 Ethical and Legal Aspects

We hypothesize that the ethical and legal conditions for open data in socio-technical SE research differs between data collection methods.

Survey data are first degree data, primarily *quantitative* or semi-quantitative (e.g. Likert scales). This implies that respondents may give consent to openness and that the opportunities for anonymization are good, particularly for large populations and samples.

Interview data are also first degree data, but primarily *qualitative*. Thus, researchers are in direct contact with respondents and can get their consent. However, the richness and strong connection to context in qualitative data makes it harder to anonymize data to enable openness. Similar conditions hold for *focus group data*.

Human-enacted *observational data* are also first degree data, mostly constituted of qualitative data. In contrast to interviews and focus group data, observed participants are (by design) less aware of the researcher's presence. Consequently, the participants have less control over the data they provide. Such data must be more actively cleared from sensitive and irrelevant information.

Technology-based *observational data* are second degree data and mostly of quantitative character. Thus, the data may be more easily separated from the context compared to qualitative data. However, the interpretation of the quantitative data is highly related to the context, and thus the value is reduced by anonymization. A special case is *instrumented data for learning*, where the data is collected by technical instruments, used to train machine learning algorithms. There are several ethical and legal concerns raised regarding these technologies, e.g. in relation to Microsoft's CoPilot[5].

Finally, *archival data* or work artifacts are third degree data. Since they are derived for other primary purposes, the openness must be considered in relation to the original contributors. Company internal artifacts are rarely possible to open, while OSS artifacts are open by definition. However, open data may still be personal, e.g. defect reports and commits, and ethical and legal concerns in relation to individuals and their data must be properly handled.

Depending on the legal and ethical conditions for each data collection method, these must be reflected in how data is handled with respect to openness.

2.4 Artifacts

Artifact evaluation has emerged as an open research practice in computer science during the last decade[6]. It aims towards improved quality and transparency, by supporting reproducible research. ACM has established a policy with three separate badge categories for papers published with ACM, related to artifact review, namely Artifacts Evaluated (Functional, Reusable), Artifacts Available, and Results Validated (Reproduced, Replicated)[7].

For software reuse in a broader sense, it is acknowledged that software reuse without any strategy or support is difficult and software must be prepared to make it reusable. For example, Belfadel et al. [1] propose an Enterprise Architecture Capability Framework, with the aim to increase reuse of software by upgrading technical components to match end-user's requirements. However, software can be made available for future reuse in several ways. Publishing software as Open Source is another way of using software available to a broader audience and therefore used in more systems.

While open data includes aspects related to human subjects from which empirical data is collected, artifact openness is primarily an issue on the researcher side. Researchers have to decide what degree of openness they apply with respect to their intellectual property and with respect to the effort it takes to make the artifacts openly available, support their usage, etc.

3 THE GANDER CASE

We present the Gander research project as a case to identify a multitude of research data and related open data concerns. The objective of the Gander project was to gain an increased understanding of code review in practice and to use this understanding to inform the design of new code review tooling. The project had two main components; 1) an empirical component with focus on problem conceptualization, providing input to the tool design – conducted as a mixed-method study with practitioners in industry [29, 32], and 2) a development component focused on development of an experimental code review platform incorporating eye-tracking [9, 28]. A key aspect in the project was to explore how gaze data from eye-tracking can be used to trigger assistance in code review tools. The Gander project was a continuation on a line of research started by the second author in two earlier studies connecting to code review, carried out in an industrial setting [26, 27].

The empirical code review study [29, 32] in the Gander project included a series of 12 semi-structured interviews with practitioners at two companies, about the experience of code review (below referred to as *Data and artifact set 1*). The interviews were recorded after informed consent and transcribed for thematic analysis. They were followed by a survey, based on the interview results, that rendered replies from 78 practitioners (*Data and artifact set 2*). The survey results were gathered, coded, and summarized for reporting. Neither the qualitative nor the quantitative data gathered in this study have been shared as supplementary material to any publication, but the protocols were shared during the review process.

One aim with the platform development in the Gander project was to build an experimental code review setup that would allow for more realistic code review experiments closer to practitioners [16], i.e., outside of the lab environment and with realistic data. With this goal in mind, the platform strives to provide a similar look-and-feel as well-used code review environments like GitLab or GitHub (with regard to the textual diff view) and it should be easy to populate the platform with realistic samples from open-source. The latter resulted in a connection to GitHub to facilitate the experimental setup process. Finally, the platform should be easy to run outside the lab, which means it has been developed using portable eye-trackers used with a laptop to increase the mobility of the setup. Figure 1

[5]https://www.sdxcentral.com/articles/news/github-copilot-raises-ownership-ethical-concerns/2022/07/
[6]https://artifact-eval.org
[7]https://www.acm.org/publications/policies/artifact-review-and-badging-current

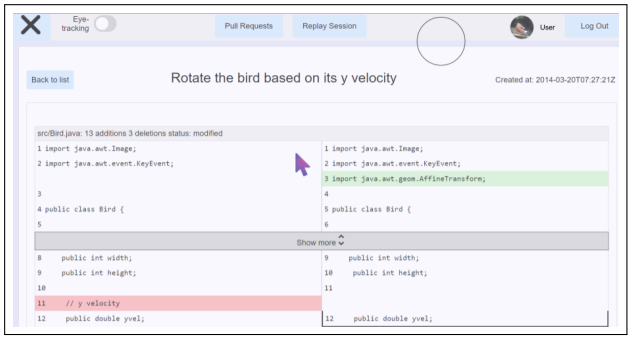

Figure 1: A screenshot of the Gander platform showing a textual diff view in replay mode, where the interaction of a session is replayed from logged interaction and eye-tracking data.

shows a screenshot of the Gander platform where the textual diff is populated with data from the FlappyBird project on GitHub.

The connection to eye-tracking and processing of gaze data is central to the design of the platform. The architecture is structured around the needs of processing eye-tracking data, for replay of sessions with participants and for exploration of real-time gaze-based assistance during a code review session. Gaze data is analyzed in real-time to detect fixation points which can be connected to programming language elements which may correspond to areas of interest. This data processing can be used to trigger assistance in response to certain gaze behavior in relation to the content of the code being reviewed.

As a proof-of-concept, the Gander platform was used to develop a simple gaze assistant that triggers visualisation of use-declaration relationships in the code based on gaze fixation point on, for instance, variable names. This assistant was tested in a user study with eight participants. During the study, participants were given a number of tasks to solve on the Gander platform with the gaze assistant enabled and then they were interviewed about their experience (*Data and artifact sets 3a and b*). The study included both quantitative data gathering, in the form of interaction logs and eye-tracking data (*Data and artifacts sets 3c and d*), and qualitative data in the form of interviews recorded after informed consent. For this study, the quantitative data has been shared as a data set, both serving as supplementary material to the publication about the platform [28] and as a test set for how to use the replay function in the platform which has been released as open-source [9].

In releasing the platform as open-source, licenses of any system used in the project, e.g., the JastAdd[8] project and the ExtendJ[9] project, had to be considered. During this review, it became clear that one of the used projects (for gaze data analysis) was available online but did not have a licence. However, after reaching out to the author of that project a licence file was added (the MIT license) and the use of the project remained unchanged. After considering the interaction of licenses in used projects and after discussion within the contributor team, a BSD license was selected for the open-source project.

[8]https://jastadd.org
[9]https://extendj.org

The project is conducted at Lund University, Sweden, which like many higher research institutes and funding agencies has an increasing focus on open science. Open science is one of the prioritised issues in the university's Research strategy 2023–26, although there is not yet any mandatory prescriptions. The Swedish Research Council, which is one of the funding agencies of this work, requires a data management plan (DMP) to be created and maintained for all projects funded 2019 or later, while the other funding agencies for the research do not yet have any requirements on open science. A data management plan was created in Lund university's DMP system, but it is not public. Creating a DMP is a first step towards fostering open data sciences, although there are no specified requirements on openess, neither from the university nor the funding agencies.

4 A CONCEPTUAL FRAMEWORK

To guide the analysis and discussion on open science in socio-technical software engineering, we define a conceptual framework of the execution of a research study, and the analysis and generalization of research artifacts. The framework emerged during our post-mortem analysis of the research projects, its actors, data, and artifacts, by iterating over the following principal steps.

(1) Identify participants and other stakeholders
(2) Identify data collected, and analyze types of data and corresponding analysis processes
(3) Identify artifacts developed for and in the project

We identified specific instances of the Gander projects and then abstracted the framework elements towards more general concepts.

The framework has three main concepts: participants, data and artifacts, as shown in Figure 2. The participants have different roles and relations to the research endeavour, while the data and artifacts are refined and evolved in separate pipelines.

4.1 Participants

We identify three typical categories of *participants* in socio-technical software engineering research, namely (1) employees or other stakeholders of software development organizations (marked with round cap in Figure 2), (2) students or other beneficiaries of the university involved (marked with square cap), and (3) independent participants (without cap). The categorization is conducted based on legal and ethical concerns in the relation between researchers and participants, and consequently the openness of data and artifacts emerging from the research. For example, employees or students may feel pressure to participate in a study, even if participation should be fully voluntary. Further, employees may participate under certain secrecy conditions bound by their employment contract.

4.2 Research Data

Regarding our research project *data* we observe that the analysis process of research data at a general level resembles a *data pipeline*. In a typical research project the researcher starts with detailed raw data, analyse that data in a sequence of steps and ends up with a set of findings. The data from the last step, i.e., the findings, typically consist of data that is "open" in the sense of being published, while the results from the steps prior to that are increasingly more difficult to make publicly available, due to privacy and secrecy concerns.

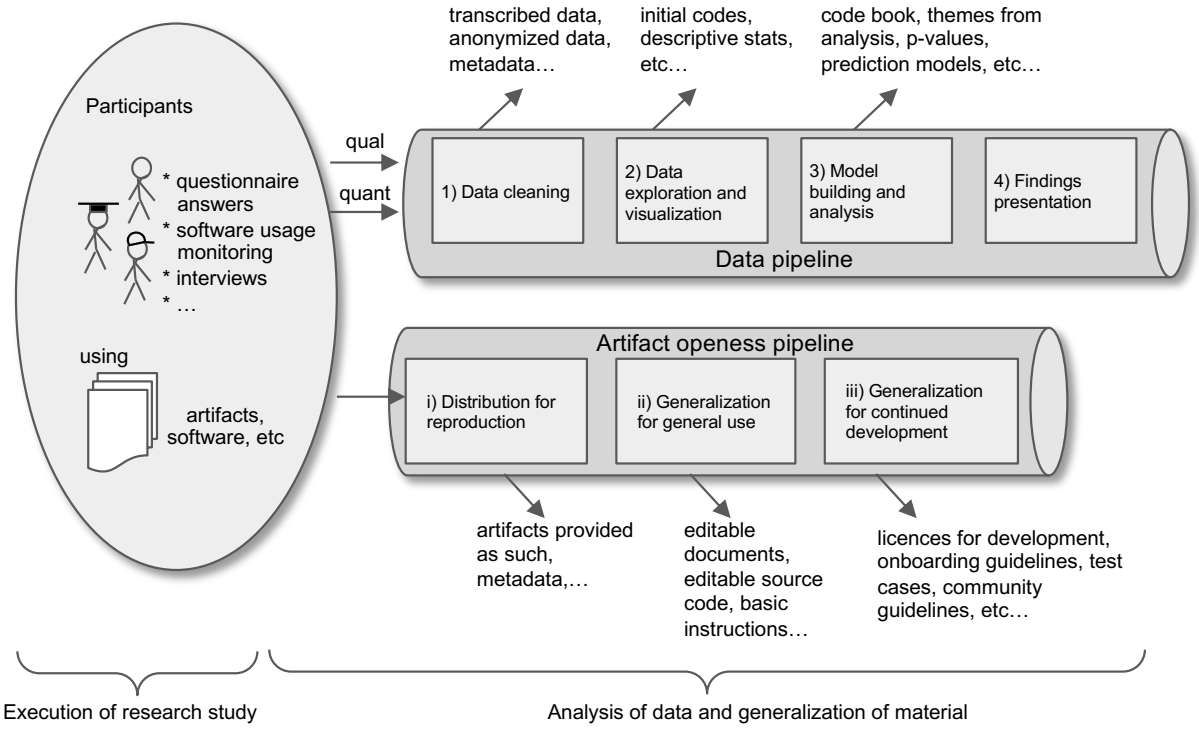

Figure 2: Graphical representation of the conceptual framework with steps of data analysis and material generalization. Data sources to the left, the data pipeline top right and artifact pipeline bottom right.

For example, in a qualitative research project with interview data, it is common to publish the general findings of an interview, but the audio recordings from the interview are rarely openly available nor are the full transcripts. In terms of openness factors, discussed by Enders et al. [6] (see Section 2.1), audio recordings and transcripts are of finer *granularity* than the abstracted findings, and thus harder to make open than the course grained findings. Consequently, not only the raw data, but the full analysis pipeline contains relevant concepts for open science.

The conceptual data pipeline, illustrated in the top right part of Figure 2, is divided into four steps which are inspired by Majeed and Hwang [18] from the field of data science, and illustrated by typical study examples below.

(1) Data cleaning:
In a *quantitative* study this involves transforming the data into a readable form for statistical tools, which may, for example, involve coding of the data based on a pre-defined scheme. It may also include anonymizing the data.
In a *qualitative* study, this typically includes transcribing the data, as well as anonymization.
(2) Data exploration and visualization:
In a *quantitative* study this includes investigating descriptive statistics and visualising data. This is also a natural step for identifying outliers.
In a *qualitative* study this would include getting a first understanding of the data and probably a first idea of procedures for coding.
(3) Model building and analysis:
In a *quantitative* study this would mean statistical analysis, e.g., building prediction models, hypothesis testing, etc.
In a *qualitative* study this would involve defining a set of codes, coding, obtaining findings, and potentially theory building, in an iterative manner.
(4) Findings presentation:
Findings are commonly presented in journal/conference publications, technical reports, or similarly, including data and analyses to support the findings.

Generally, outcomes from the later stages of the data pipeline are less sensitive to share openly, both with respect to participants' privacy and potential company secrets.

4.3 Research Artifacts

Research artifacts may emerge from a study, like questionnaires and other tools for data collection, software script for analysing qualitative data, software tools for illustration of the research conducted (e.g., a tool for managing code reviews in a code review experiment) or the studied code itself.

This type of research artifacts can be used in studies where results are validated by other research groups repeating the studies. In ACM terminology a study may be *reproduced* (different team, same

experiment setup) or *replicated* (different team, different experiment setup)[10]

In these situations artifacts can either be input, or serve as background information. For example, in one case exactly the same research is conducted by another group, and in another case studies build on the research, but develop or adapt artifacts and introduce them in a new context. Notice that these terms are used in different ways by different researchers. In Empirical Software Engineering, the term replication can mean conducting new studies similar to previously conducted studies (see, e.g., Gómez et al. [12]).

Inspired by Belfadel et al. [1], we identify three steps of artifact generalization towards increasing reuse:

i) Publication for reproduction, resulting in artifacts in original state (non-editable documents, executable code, etc).
ii) Generalization for general use, resulting in artifacts in editable state (editable documents, editable source code, etc).
iii) Generalization for continued development, resulting in artifacts released with guidance how to adapt it, e.g., by inviting to a community.

Each of these steps require additional investments in making the artifacts openly available. The first step (i) focuses on transparency through accessibility, connecting to the goals of the ACM artifact badges. To advance the research, access to editable artifacts are needed (step ii). To build a community (step iii) around tools or other research artifacts, requires governance effort, like for any open source software.

These three dimensions constitute our conceptual framework, and is used next to analyze data and artifacts in the Gander case.

5 ANALYSIS AND DISCUSSION OF GANDER DATA AND ARTIFACTS

To demonstrate potential use of the conceptual framework from Figure 2, we extract data sets and artifacts shared in the Gander case and list them in Table 1. Further, based on trade-offs in our case with respect to different data sets, we propose recommendations for open data practices, summarized in framed boxes below. When possible, we have also indicated which ACM badge level we believe that the recommendation supports.

The data sets and artifacts are split into three sections; the semi-structured interviews of the empirical part of the project (1), the survey part of the same empirical part (2), and the user study connected to the development part of the project (3).

The data gathered in the empirical part of the Gander project, data set (1a) and (2a), went through the pipeline of Figure 2, i.e., data cleaning, data exploration and visualisation, model building and analysis, and were then shared in anonymized and summarized form in the presentation of the results [29, 32]. The protocols from the empirical part were shared as metadata (Step i) for inspection during the review process.

Similarly, for the development part of the Gander project, the interview data gathered during the user study, data set (3a), were shared in anonymized and summarized form in the presentation of the results [28]. The protocols for the interviews were shared as metadata (Step i) both during the review process and also later. In addition, the session data gathered during the user study, in the form of interaction data and gaze data, data set (3c), was shared as anonymized data (Data step 1) along side the experimental setup used, the Gander platform (Artifact step iii). The purpose of sharing the Gander platform is to contribute to the research community and to enable and facilitate further research into gaze-assisted code review tooling. With this goal in mind, care was taken to select an appropriate license for the project and to review dependencies with regard to licenses. There was one project among the dependencies that was shared without a license (matching artifact step ii) but after discussion with the Gander team the project added a license (matching artifact step iii).

Giving open access to *research artifacts*, like interview and survey protocols (artifacts 1b, 2b, and 3b) is non-controversial and mostly a matter of practical procedures for their publication and sustained accessibility. In the Gander case they were provided for peer review in the first two cases, but then not published with the papers, while published with the platform artifact in the third case. Given the space constraints of conference papers, authors are reluctant to use the space by adding such protocols as appendices. However, conferences and journals may offer online publication of supplementary material with the main publication. Alternatively, artifacts may be given persistent digital object identifiers (DOI) on their own right, although this adds to the bureaucracy burden for researchers. Providing access through a university's persistent storage, like in our third case [8], is convenient for the researchers, although not optimal from a traceability point of view.

> **R1**. *We therefore advice open research artifacts be given persistent DOI to enable traceability, independently of storage solution – as long as it is persistent enough. [ACM Available]*

Providing research platform artifacts as *open source software* is a highly recommended practice. The Gander platform builds on other open source projects, which helped speed up the development. However, the licensing issues reported in Section 3 demonstrates clearly, that the artifact step ii is not sufficient to build further research on. This is both due to the unclear licensing situation, and lack of community that might respond to questions and improvement proposals. In our case, the issue was sorted out in dialogue with the originating author, but that is not a scalable solution.

There might be conflicting interests with opening research artifacts, if the originator aims to commercialise the material or some services or products build thereon. However, we still advocate for open source solutions, which actually may be compatible with business models, such as freemium [22] or servitization [15].

> **R2**. *We advice research software be made open source with an appropriate license. We further advice research institutes and funding agencies to cover costs related to governing OSS communities for such software. [ACM Artifacts Evaluated – Functional]*

Access to *research data* is more sensitive and is in the Gander case published only in synthesized form in the publications. Both data sets (1a) and (2a), i.e. qualitative interview data and quantitative survey data, were collected within the same two multi-national companies.

[10] https://www.acm.org/publications/policies/artifact-review-badging Notice that ACM has recently swapped the meaning of the two terms after discussion with the National Information Standards Organization.

Table 1: Data and artifacts made open from the Gander case. * Shared as part of the peer review process but not after.

Data/Artifact	Class	Kind	Participants	Figure 2 step	Purpose
1a) Semi-structured interview data (12 participants)	First	Qual	Industry	4 [29, 32]	Conceptualization
1b) Semi-structured interview protocol	-	Artifact	-	i*	
2a) Survey data (78 practitioners)	First	Quant	Industry	2, 3, 4 [29]	Conceptualization
2b) Survey protocol	-	Artifact	-	i*	
3a) User study interview data (8 participants)	First	Qual	Students	4 [28]	Validation
3b) User study protocol	-	Artifact	-	i	
3c) User study gaze and interaction data	Second	Quant	Students	1	
3d) User study software (the Gander platform)	-	Artifact	-	iii	

The risks related to sharing the *qualitative interview data* are multifold: firstly, information related to the company that is not relevant for the focus of the study might be mentioned in the interview, e.g. information about the physical design of an embedded product to come. Secondly, the information about the company is relevant, but has to be filtered before publication, e.g. a critical event for the company in relation to security that both interviewer and interviewee knew about, but still was not in the public communication. Thirdly, the interviewee could mention facts or opinions that are sensitive with respect to their own future in the company, i.e. criticising a manager for certain actions, or lack thereof. Any of these factors on their own prevents from publishing the raw interview data, both with respect to the information as such, and that it is impossible to anonymize individuals among a such small set of interviewees. On top of that, fourthly, the general criticism with respect to epistemological concerns, raised by e.g. Chauvette et al. [3], that the lack of connection to the context makes data useless. We share these concerns partially in our case, since we have a long research collaboration track record with the companies in the study, which means that the shared understanding of the software engineering practice is significant. Transcripts of the interviews might be hard to understand without the knowledge of the context, e.g. code review practices of the company.

This discussion leads us *not* to recommend sharing qualitative data from companies openly. However, researchers could consider publishing code books from the qualitative analysis (data step 3), as proposed by Field et al. [7] and DuBois et al. [4], as well as transparently reporting evolution of codes, conceptual model and theory, for example, as done by Runeson et al. [25]. Regarding qualitative data from students, the participants' integrity must be protected although there are no company secrets to protect, which leads to a similar recommendation as for company participants.

R3. *We advice not to openly publish qualitative research data, but to publish study and analysis artifacts, such as study protocols, interview guides, interviewee descriptions, and code books from thematic analysis. [4, 7]*

The *quantitative survey data* is somewhat easier to share more openly. Firstly, there are more participants – finding a person in a large pool is harder than in a small one, although modern data analyses are very powerful in finding a "nail in a haystack". Secondly, the opinions shared in Likert scale responses are not as rich and detailed as qualitative survey or interview responses, unless the questions are asked to shame the company. Finally, the statistical analysis methods and tools enable more standardized analyses, which reduce the need for transparency in terms of open data, unless there are suspicions of fake data which should be checked.

R4. *We recommend quantitative data be shared openly, if and only if, the data is anonymized sufficiently to protect the identity of the individual or company (if requested).*

Finally, we have a set of data (3c) which is collected through "human-enacted inquisitive and observational techniques", i.e. eye-tracking data. In the case where raw image data is collected, the data is by definition personal and non-anonymizable, since the eye iris is possible to uniquely identifiable to persons. Thus, in such cases anonymization of eye-tracking data must take place to an abstracted level, e.g. eye movement positions. In the case of the Gander project, the eye-tracker used does not collect iris images, but rather details such as left and right gaze positions and pupil diameter.

R5. *In case of data that can be directly or indirectly traceable to individuals, it cannot be open. Transforming such data into anonymized forms may enable publication.*

We derived this conceptual framework from one line of research in software engineering, in the context of our experience of many years of empirical software engineering research. The project contains a multitude of data and artifacts, and our collective experience is extensive. Still we do not claim this conceptual framework is generally applicable nor in a final state. We therefore invite the research community to validate and further extend the framework and its recommendations for practice.

6 CONCLUSIONS

To support the transition of SE research towards open science, we have derived a conceptual framework, based on our experiences with a multitude of data and artifacts in a socio-technical software engineering project, that entails participants, data and artifacts. We unfold the variation in these concepts across our project, and discuss openness practices in relation to those.

Based on the guiding principles for FAIR data – "as open as possible and as closed as nescessary" – we recommend that research artifacts, such as survey and interview instruments are always made open access. Research platforms should also be made open, but need governance, e.g. licence and community, to reach its full potential. Quantitative data may be more open, due to its standardization and

pure volume, while opening qualitative data comes with several risks and challenges. At the end of the day, participants' integrity and companies secrecy concerns are essential to respect, also while advocating the benefits of open science.

The framework and recommendations align with open science and FAIR data principles, as well as artifact evaluation policies. Our contribution is to interweave these with our experiences from a concrete research project and to generalize for a broader range of software engineering projects.

We advise that our conceptual framework be used to guide trade-offs between openness and closeness. We hope that the preliminary recommendations become a starting point for the research community on open science practices for empirical software engineering. We further invite the community to validate and evolve the guidelines to be more comprehensive.

ACKNOWLEDGEMENT

The authors would like to thanks the co-workers in the Gander project. This work has been partially supported by the Swedish Foundation for Strategic Research (grant no. FFL18-0231), the Swedish Research Council (grant no. 2019-05658), ELLIIT – the Swedish Strategic Research Area in IT and Mobile Communications, and the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

OPEN DATA AND ARTIFACTS

The following data and artifacts are openly available for use.

- The Gander platform [9]
- Supplementary data to the Gander user study and platform (protocols and session data)[8]

REFERENCES

[1] Belfadel, A., Amdouni, E., Laval, J., Cherifi, C.B., Moalla, N., 2022. Towards software reuse through an enterprise architecture-based software capability profile. Enterprise Information Systems 16, 29 – 70. doi:10.1080/17517575.2020.1843076.

[2] Briand, L.C., Bianculli, D., Nejati, S., Pastore, F., Sabetzadeh, M., 2017. The case for context-driven software engineering research: Generalizability is overrated. IEEE Software 34, 72–75. doi:10.1109/MS.2017.3571562.

[3] Chauvette, A., Schick-Makaroff, K., Molzahn, A.E., 2019. Open data in qualitative research. International Journal of Qualitative Methods 18, 160940691882386. doi:10.1177/1609406918823863.

[4] DuBois, J.M., Mozersky, J., Parsons, M., Walsh, H.A., Friedrich, A., Pienta, A., 2023. Exchanging words: Engaging the challenges of sharing qualitative research data. Proceedings of the National Academy of Sciences 120. doi:10.1073/pnas.2206981120.

[5] DuBois, J.M., Strait, M., Walsh, H., 2018. Is it time to share qualitative research data? Qualitative Psychology 5, 380–393. doi:10.1037/qup0000076.

[6] Enders, T., Wolff, C., Satzger, G., 2020. Knowing what to share: Selective revealing in open data, in: European Conference on Information Systems (ECIS) Research-in-Progress Papers, p. 11. URL: https://aisel.aisnet.org/ecis2020_rip/11.

[7] Field, S.M., van Ravenzwaaij, D., Pittelkow, M.M., Hoek, J.M., Derksen, M., 2021. Qualitative open science – pain points and perspectives, in: OSF preprints, Center for Open Science. doi:10.31219/osf.io/e3cq4.

[8] Gander Contributors, 2023a. Gander: a platform for exploration of gaze-driven assistance in code review - supplementary material. https://doi.org/10.5281/zenodo.10527122.

[9] Gander Contributors, 2023b. The Gander open source platform. https://gitlab.com/lund-university/gander.

[10] González-Barahona, J.M., Robles, G., 2012. On the reproducibility of empirical software engineering studies based on data retrieved from development repositories. Empiririal Software Engineering 17, 75–89. doi:10.1007/S10664-011-9181-9.

[11] González-Barahona, J.M., Robles, G., 2023. Revisiting the reproducibility of empirical software engineering studies based on data retrieved from development repositories. Information and Software Technology 164, 107318. doi:10.1016/J.INFSOF.2023.107318.

[12] Gómez, O.S., Juristo, N., Vegas, S., 2014. Understanding replication of experiments in software engineering: A classification. Information and Software Technology 56, 1033–1048. doi:https://doi.org/10.1016/j.infsof.2014.04.004.

[13] Joyce, J.B., Douglass, T., Benwell, B., Rhys, C.S., Parry, R., Simmons, R., Kerrison, A., 2022. Should we share qualitative data? Epistemological and practical insights from conversation analysis. International Journal of Social Research Methodology 0, 1–15. doi:10.1080/13645579.2022.2087851.

[14] Kitchenham, B.A., Budgen, D., Brereton, P., 2015. Evidence-Based Software Engineering and Systematic Reviews. Routledge.

[15] Kowalkowski, C., Gebauer, H., Kamp, B., Parry, G., 2017. Servitization and deservitization: Overview, concepts, and definitions. Industrial Marketing Management 60, 4–10. doi:10.1016/j.indmarman.2016.12.007.

[16] Kuang, P., Söderberg, E., Niehorster, D., Höst, M., 2023. Toward gaze-assisted developer tools, in: Proceedings of the 45th IEEE/ACM International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER). doi:10.1109/ICSE-NIER58687.2023.00015.

[17] Lethbridge, T.C., Sim, S.E., Singer, J., 2005. Studying software engineers: Data collection techniques for software field studies. Empirical Software Engineering 10, 311–341. doi:10.1007/s10664-005-1290-x.

[18] Majeed, A., Hwang, S.O., 2023. Data-centric artificial intelligence, preprocessing, and the quest for transformative artificial intelligence systems development. Computer 56, 109–115. doi:10.1109/MC.2023.3240450.

[19] Mendez, D., Graziotin, D., Wagner, S., Seibold, H., 2020. Open science in software engineering, in: Felderer, M., Travassos, G.H. (Eds.), Contemporary Empirical Methods in Software Engineering. Springer International Publishing, Cham, pp. 477–501. doi:10.1007/978-3-030-32489-6_17.

[20] Munir, H., Linåker, J., Wnuk, K., Runeson, P., Regnell, B., 2017. Open innovation using open source tools: A case study at Sony Mobile. Empirical Software Engineering 23, 186–223. doi:10.1007/s10664-017-9511-7.

[21] Méndez Fernández, D., Monperrus, M., Feldt, R., Zimmermann, T., 2019. The open science initiative of the empirical software engineering journal. Empirical Software Engineering 24, 1057–1060. doi:10.1007/s10664-019-09712-x.

[22] Niculescu, M.F., Wu, D.J., 2014. Economics of free under perpetual licensing: Implications for the software industry. Information Systems Research 25, 173–199. doi:10.2139/ssrn.1853603.

[23] Robinson, B., Francis, P., 2010. Improving industrial adoption of software engineering research: a comparison of open and closed source software, in: Succi, G., Morisio, M., Nagappan, N. (Eds.), Proceedings of the International Symposium on Empirical Software Engineering and Measurement (ESEM), ACM. doi:10.1145/1852786.1852814.

[24] Runeson, P., Höst, M., Rainer, A., Regnell, B., 2012. Case Study Research in Software Engineering – Guidelines and Examples. Wiley. doi:10.1002/9781118181034.

[25] Runeson, P., Olsson, T., Linåker, J., 2021. Open data ecosystems – an empirical investigation into an emerging industry collaboration concept. Journal of Systems and Software 182, 111088. doi:10.1016/j.jss.2021.111088.

[26] Sadowski, C., Söderberg, E., Church, L., Sipko, M., Bacchelli, A., 2018. Modern code review: A case study at google, in: Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice, ACM. pp. 181–190. doi:10.1145/3183519.3183525.

[27] Sadowski, C., Van Gogh, J., Jaspan, C., Soderberg, E., Winter, C., 2015. Tricorder: Building a program analysis ecosystem, in: 37th IEEE International Conference on Software Engineering, IEEE. pp. 598–608. doi:10.1109/ICSE.2015.76.

[28] Saranpää, W., Apell Skjutar, F., Heander, J., Söderberg, E., Niehorster, D.C., Mattsson, O., Klintskog, H., Church, L., 2023. Gander: A platform for exploration of gaze-driven assistance in code review, in: Proceedings of the 2023 Symposium on Eye Tracking Research and Applications, ACM. doi:10.1145/3588015.3589191.

[29] Söderberg, E., Church, L., Börstler, J., Niehorster, D., Rydenfält, C., 2022. Understanding the experience of code review: Misalignments, attention, and units of analysis, in: Proceedings of the International Conference on Evaluation and Assessment in Software Engineering (EASE), ACM. doi:10.1145/3530019.3530037.

[30] Solari, M., Vegas, S., Juristo, N., 2018. Content and structure of laboratory packages for software engineering experiments. Information and Software Technology 97, 64–79. doi:10.1016/j.infsof.2017.12.016.

[31] Storey, M.A., Ernst, N.A., Williams, C., Kalliamvakou, E., 2020. The who, what, how of software engineering research: a socio-technical framework. Empirical Software Engineering 25, 4097–4129. doi:10.1007/s10664-020-09858-z.

[32] Söderberg, E., Church, L., Börstler, J., Niehorster, D.C., Rydenfält, C., 2022. What's bothering developers in code review?, in: IEEE/ACM 44th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), pp. 341–342. doi:10.1145/3510457.3513083.

[33] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A., 2012. Experimentation in Software Engineering. Springer. doi:10.1007/978-3-642-29044-2.