A Systematic Analysis of Problems in Open Collaborative Data Engineering

PHILIP HELTWEG and DIRK RIEHLE, Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Collaborative workflows are common in open-source software development. They reduce individual costs and improve the quality of work results. Open data shares many characteristics with open-source software as it can be used, modified, and redistributed by anyone, for free. However, in contrast to open-source software engineering, collaborative data engineering on open data lacks a shared understanding of processes, methods, and tools.

This article presents a systematic literature review of collaboration processes, methods, and tools in data engineering as performed by open data users. An additional interview study with practitioners confirms and enhances the findings and strengthens the resulting insights.

We find an ecosystem with heterogeneous participants and no standardized processes, methods, and tools. Participants face a variety of technical and social challenges during their work. Our work provides a structured overview of collaboration systems in open collaborative data engineering, enabling further research. Additionally, we contribute preliminary guidelines for successful open collaborative data engineering projects and recommendations to increase its adoption for open data ecosystems.

CCS Concepts: • **Human-centered computing** → **Collaborative and social computing systems and tools**; *Empirical studies in collaborative and social computing*; • **General and reference** → Surveys and overviews.

Additional Key Words and Phrases: collaboration, data engineering, open data

1 INTRODUCTION

Open data is data that can be used, modified, and shared, free of charge. However, using open data can be challenging for many reasons, including poor quality of data sources, uncommon and undefined formats and schemata, and lack of well-understood workflows and processes. Data engineering, the process of extracting, preparing and transforming the data into a usable format, is a labor-intensive and hence costly engineering activity [18, 26].

Open-source development has shown that collaborating on shared software artifacts can lower individual costs and improve quality. Similarly, sharing intermediate artifacts between data-projects could allow contributors to collectively increase the quality of the data they use, as demonstrated by Infomediaries that add value to data by preprocessing it for other consumers [31]. We make the natural assumption, that, due to its similarities with open-source software, open data can be improved in open collaborative workflows in which self-organizing, meritocratic and egalitarian communities of users contribute to a shared artifact [20]. The collaborative effort can be driven by the users' own motivation to improve the data for their reuse, eliminating the need to rely solely on data publishers, who may lack incentives to provide it in a well-structured format.

Authors' address: Philip Heltweg, philip@heltweg.org; Dirk Riehle, dirk@riehle.org, Friedrich-Alexander-Universität Erlangen-Nürnberg, Martensstr. 3, Erlangen, Bavaria, Germany, 91058.

2469-7818/2023/10-ART $15.00
https://doi.org/10.1145/3629040

However, while open collaboration is common in open-source software development, data engineering is typically project-specific and done by small teams. These teams often reuse tools and workflows from collaborative software engineering like GitHub that, while passable, lack features to specifically support collaborative data engineering [4].

Tools and processes made specifically for collaborative data engineering would be needed for best results. However, it is unclear which challenges they need to address. For other stages of the data science workflow, using more specific tools and workflows has shown promising effects [19, 22, 23].

It is therefore important to understand how data engineers collaborate. To do so, it is key to not only consider specific software, but also who participates in the collaboration, their workflows and how they interact. In this article, we investigate the larger context and refer to the combination of participants, their workflows, and tools as well as the artifacts they create as collaboration systems. Especially for open data, collaborative data engineering is a complex activity with participants from various backgrounds working together [7]. Social systems and their interactions with technical infrastructure are a large factor that creates challenges for data users. So far, academic research has focused on data publishers and the technical challenges of publishing good quality data. A comprehensive theory of collaborative data engineering by data users is missing.

To move towards such a theory, we answer the following research questions:

Research Question 1: *Which elements of collaboration systems for data engineering by open data users are described in literature?*

Research Question 2: *How and in which roles do participants in collaboration systems for data engineering interact socially?*

Research Question 3: *What are challenges to collaboration in data engineering and why?*

We contribute a descriptive overview of the elements of collaboration systems, during data engineering by data users. Elements include participants, activities they attempt during the data engineering process, the tools participants use and artifacts they create. Additionally, we describe the social systems that participants work in, highlighting the different roles they fulfil and how they interact with others during collaborative data engineering.

Furthermore, we contribute a list of challenges that data engineering practitioners face during collaboration with a rich description and insights from interviews. Based on these insights, we discuss guidelines that contribute to successful open collaborative data engineering projects. For practitioners and researchers in an open data context, we provide a list of recommendations to increase the adoption of open collaborative data engineering in their communities. Together, these contributions provide a starting point for a theory of open collaborative data engineering that can be extended in future research. The insights gathered about social systems and challenges experienced by practitioners can be used to define better requirements for future software tools that plan to support collaborative data engineering.

This paper extends previous work [7] with a qualitative survey among data engineering practitioners, with a focus on social challenges experienced during collaborative data engineering. In addition, guidelines and recommendations based on these insights are discussed.

The following structure will be used to present the work: Initially, related work is discussed in section 2. A description of the research design that was used to answer the research questions follows in section 3. The results are presented in section 4, followed by a discussion of their implications and suggestion of guidelines and recommendations in section 5. Potential limitations and how we attempted to mitigate them are shown in section 6, after which we provide a summary and outlook in section 7.

2 RELATED WORK

Collaborative work, especially distributed collaboration, has extensively been studied in the domain of software engineering. O'Leary et al. [17] provide an overview of distributed collaboration types, based on a systematic

literature review. They point out that it is important to consider both technical as well as social contributing factors for successful distributed collaboration, and describe previously identified factors found in literature. We follow a similar approach, considering both technical and social elements of collaboration systems but with a much more narrow focus. The insights regarding open collaboration in data engineering presented in this article provide additional data on contributing factors in one specific domain.

In collaborative software engineering, researchers have explored approaches with increasingly larger numbers of participants and less imposed social structure, from distributed software development or global software development, over crowdsourcing approaches like hackathons to open-source software development.

Distributed software development or global software development is especially relevant in an enterprise context [10]. Challenges described include communication with remote colleagues and accessing expert knowledge [8]. In their structured review, Jiménez et al. [10] conclude that technological tools and processes must be adapted to the specific needs of an organization to reap the benefits of distributed software development. LaToza and van der Hoek [14] contribute a model to categorize crowdsourcing approaches like hackathons or open-source software development and come to a similar conclusion – the need to develop and adapt workflows for software development tasks. With its close relationship to crowdsourced software engineering approaches, open collaborative data engineering faces similar challenges and more insight into the underlying social dynamics and challenges is needed to develop more appropriate tools and workflows.

Open collaborative data engineering is most closely related to open-source software development. In both, organizational structure is not formally enforced but emerges from the community. Previous work analyzed raw data from public mailing lists [1] and software forges [29] to gather insights into community structures and roles. More recently, the structure of collaborative projects and resulting challenges to social interactions and software architecture have been studied empirically [3, 25]. In this article, we focus on a description of social interactions and challenges in open collaborative data engineering in the hope to enable similar work in the future. Nonetheless, we also contribute guidelines and recommendations based on insight from practitioners.

For open-source software development, project forges have proved essential to enable open collaboration, for example by providing standard tools and artifacts inside companies [20] or increasing awareness of community activity, as found on GitHub [5]. Due to its popularity, GitHub has a strong influence on collaboration workflows used in software engineering with its pull-based development flow becoming the de facto standard in open-source software development.

Data science has been a focus of academic activity recently, however, most publications that contribute insights about collaborative work focus on machine learning or data analysis, often in commercial settings. In open data contexts, publications that describe data engineering almost solely look at how data publishers can provide better quality. As far as we know, no reviews of how open data users collaborate during data engineering exists.

Workflows of data scientists and their impressions of automated AI are the focus of recent work by Wang et al. [28], but they also provide a review of academic literature about data science teams and tools they use. They conclude overly complex tools pose a barrier for subject-matter experts to participate in data science teams. Likewise for corporate settings, Terrizzano et al. [26] describe data engineering at IBM, including barriers to data usage. Zhang et al. [30] gather data on collaboration of data science workers in large companies by conducting an anonymous survey. Their results show data scientists collaborate actively, work in small teams, and use a variety of tools. Because their work is based only on survey responses from employees at IBM, they are unsure if their results generalize outside of environments at large corporations. Our work builds on these studies by providing more data on collaboration by data users in different contexts, like open data and hackathons.

In the process of developing a collaborative framework for feature engineering, Smith et al. [22] identify the four main challenges as task management, tool mismatch, evaluation of contributions and maintaining infrastructure from literature and user studies. Whereas their work takes the perspective of supporting machine

learning projects, the challenges identified in this article relate to collaboration during data engineering on open data without an ML focus.

Zuiderwijk et al. [31] review the literature about open data ecosystems, identifying important elements and their activities that contribute to successful open data publication and reuse. They describe scenarios of interactions across the ecosystem that include the release of data, search for data, processing, and use of data as well as providing feedback to publishers. While their work provides insights into the wider ecosystems that exist for open data, we focus only on the collaboration by data users themselves to take a more detailed viewpoint.

Collaboration during open data analysis is studied by Choi and Tausczik [4] using interviews and surveys. Participants work in small, interdisciplinary teams and create tools and reports based on data. The authors identify the need for further research about how platforms can best support open data analysis, as traditional hubs for software engineering like GitHub lack important features. By contributing challenges to collaborative data engineering, we support further research and the development of better tools towards that goal.

3 RESEARCH DESIGN

The need for a review was identified from an initial pilot study of the literature that revealed that collaboration during data engineering is seldom discussed. We approached the research questions with a two-step research design, first reviewing the existing literature and then using the acquired knowledge to interview practitioners about their experience.

Initially, we performed a systematic literature review (SLR) according to Kitchenham [12]. The pilot study revealed that the focus of academic literature is often on later stages of a data science project, like data analysis or machine learning. In the context of open data, research about collaboration practices between data users is rare, while data publishers are covered. We therefore concluded that a review of the existing literature on collaboration systems of data engineering by open data users would be needed to close this gap.

In a second step, we complemented the acquired insight from literature with a qualitative survey according to Jansen [9]. We aimed to create a description of social systems and the diversity of challenges to collaborative data engineering among practitioners. We gathered qualitative data by conducting semi-structured interviews, informed from the previous literature review. In addition to new insights, we asked interview participants if they had experienced challenges we had previously identified from literature to verify our results.

3.1 Systematic Literature Review

We performed a systematic literature review (SLR) according to Kitchenham [12] to answer RQ1 *Which elements of collaboration systems for data engineering by open data users are described in literature?* To do so, we defined a search strategy consisting of data sources, queries, inclusion, and exclusion criteria following a pilot study. From the pool of relevant literature, we extracted and synthesized elements of collaboration systems for data engineering by open data users.

3.1.1 Search Strategy. An initial pilot study was performed to create the search strategy. For the pilot study, we iteratively looked at a broad range of literature from different academic databases to get an overview of existing research directions on collaborative work, data engineering and open data. We included the most relevant articles that were part of the pilot study directly in a pool of potential articles. From the results, we also decided on selecting Google Scholar and Scopus as sources for our literature search to include a wide range of publications because all relevant articles were found in these databases. We decided on searching for articles that include *open data* and *workflow*, *process*, *practices* or *participants* and variations of those terms. Most publications on open data were published after 2008 [18], so we limited both searches to that time. An overview of the full search process is shown in Figure 1.

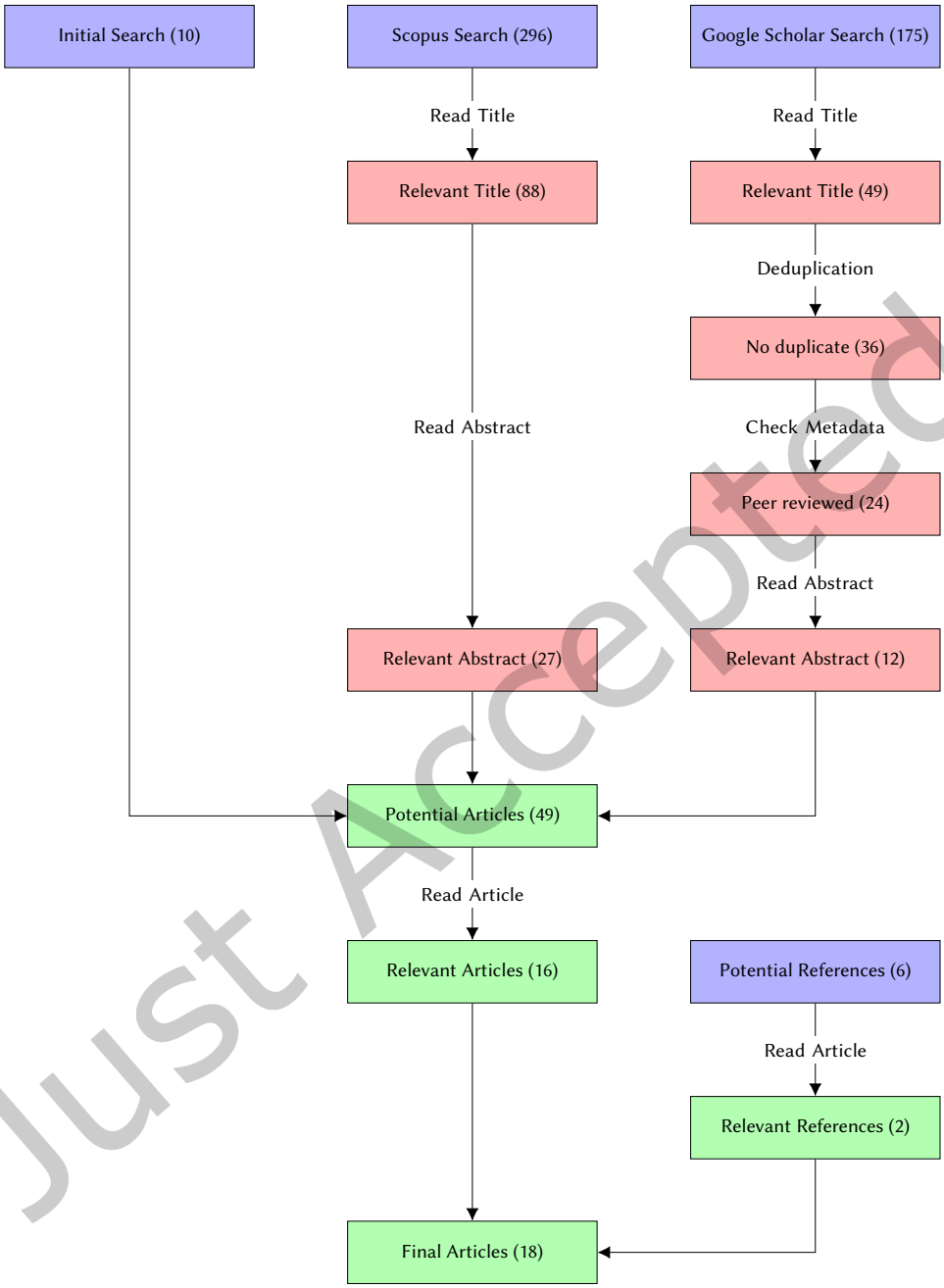

Fig. 1. Process of the systematic literature review

For Scopus, search results were restricted to journal articles or conference proceedings in English or German with a publication stage of final, leading to 296 results. We used the following search string in article titles and abstracts, making sure the keywords appear within five words of each other:

```
("open data" OR "open-data")
W/5 ("workflow" OR "workflows"
OR "process" OR "processes"
OR "practices" OR "participants")
```

Google Scholar does not offer the ability to enforce keywords to be close together. Searching the full text of articles returned many irrelevant articles, to narrow down the search and only include relevant results, we searched the title of publications with the following search string:

```
allintitle:workflow OR workflows OR
process OR processes OR practices
OR participants "open data"
```

The title-based Google Scholar search returned 175 articles.

After executing the initial search, we worked with 481 potential publications and ensured the results were relevant with several additional checks.

As a basis for decisions, we used the following inclusion and exclusion criteria:

- **Include** articles that describe data engineering workflows or processes with open data
- **Include** articles reporting on data engineering during a concrete project with open data
- **Exclude** articles that are not peer-reviewed journal or conference papers
- **Exclude** articles exclusively on data publishers
- **Exclude** articles that could not be retrieved in full

For both result sets, we excluded irrelevant papers based on their title and kept 88 results from Scopus and 49 from Google Scholar. Because Google Scholars results were less restricted, we also removed 13 not-peer reviewed articles and 12 duplicates from them.

Finally, we read the abstracts of every article and applied the inclusion and exclusion criteria to them to create a pool of 49 potentially relevant articles, 10 from the pilot search, 27 from Scopus and 12 from Google Scholar.

Next, we read all articles in full, noting down references that seemed especially relevant in the context of our research questions. After verifying that these were peer-reviewed and relevant, we included a further two articles [15, 16] from forward references.

Based on the inclusion and exclusion criteria applied to the full text of an article, we excluded a further 33 articles from the pool of potentially relevant articles, mainly because they focused on other phases of the data science lifecycle and did not include a description of data engineering.

This process led to a final pool of 18 relevant articles. We searched for articles published after 2008 but identified relevant articles between 2013 and 2021 (see Figure 2). The searches were executed during March and April 2022, meaning no article published in 2023 is included.

The search process, queries, and results are available in the published raw data [1].

3.1.2 Data Extraction & Synthesis. We extracted data according to the descriptive data synthesis described by Kitchenham [12], using data extraction sheets for mentions of any activity, participant, tool used, or artifact created during data engineering with open data.

For every article, we noted any mention into the corresponding data sheet, merging any that were substantially similar to previously identified elements. Because publications that describe projects including open data often do not focus on data engineering, we included elements of collaboration systems liberally, even if they were not the main focus of the text. In the case of data engineering activities, we grouped activities into larger categories

[1] Available on Zenodo at https://doi.org/10.5281/zenodo.6598447

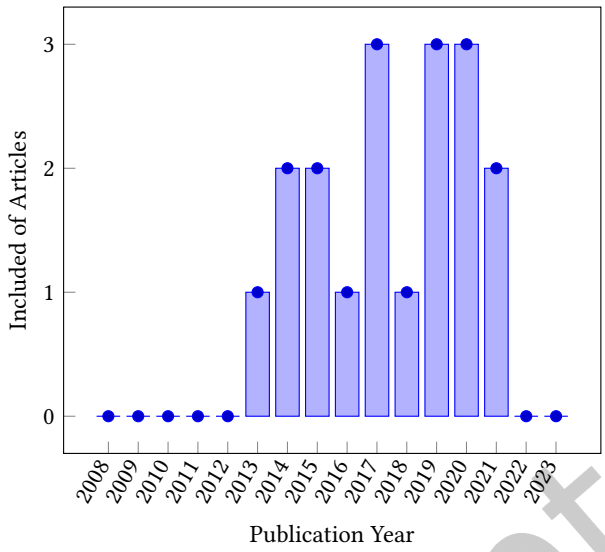

Fig. 2. Publication date of included articles

but report the individual activities separately as well. We created the categories of data engineering activities after data extraction, based on the list of activities, to include a detailed overview of the data engineering process.

We wrote descriptions and examples for all elements of collaboration systems that were identified during the data extraction and made the raw data available online[1].

Finally, we shared the results of the data extraction with a practitioner working on data engineering with open mobility data as a member check [6] (see Table 2). From their feedback, we added some activities and artifacts that were not described in literature. Overall, their feedback was positive, and they felt that the data was complete and aligned with their experiences.

3.1.3 Stopping Criteria. Theoretical saturation [2] was chosen as stopping criterion for the search because we wanted to identify the diversity of elements in collaboration systems for data engineering by open data users. Therefore, we tracked the number of new elements we added with every article (see Figure 3). We considered theoretical saturation reached when we did not gain any new insights after analyzing multiple articles and concluded the search.

3.2 Qualitative Survey

After gathering and analyzing the results from the systematic literature review, we extended the research design with an additional qualitative survey using semi-structured interviews according to Jansen [9]. Data was gathered from data engineering practitioners as a form of data source triangulation [27], allowing for insights outside academic literature. Jansen describes a typical empirical cycle of one-shot qualitative surveys as consisting of the definition of knowledge aims and sampling, data collection and finally analysis of the collected data.

3.2.1 Knowledge Aims & Sampling. Experiences of practitioners are needed to answer RQ2 *How and in which roles do participants in collaboration systems for data engineering interact socially?* and RQ3 *What are challenges to collaboration during data engineering and why?* We therefore aimed to create an inductive description of the

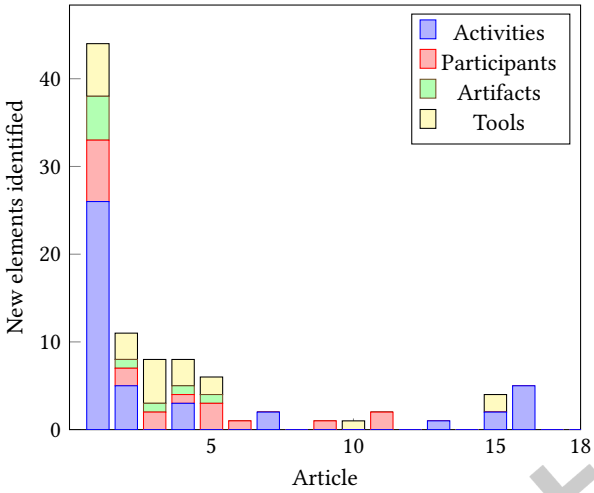

Fig. 3. New elements identified by article

diversity of social systems and challenges to collaboration in data engineering among people who have attempted collaborative data engineering before, informed by our previous knowledge from the structured literature review.

Pseudonym	Job Role	Employer	Project Types	Data Domains	Professional	Open Data
E1	Data acquisition	Academic publisher	Data curation	Material science	Yes	Both
E2	Software engineer	Software agency	Civic society	Transport	No	Yes
E6	Data engineer	Nonprofit organisation	Hackathons	Transport, Energy, Political	No	Yes
E8	Executive	Software agency	Hackathons	Geographical, Financial	Both	Yes
E9	Data engineer	Nonprofit organisation	Scientific	Medical, Biological	Yes	No

Table 1. Participants in semi-structured interviews

Because our goal was to describe the diversity and not to make statistical inferences, we decided to select a theoretically diverse sample. We reached out to a variety of contacts that worked on projects that included data engineering, and selected interview participants based on demographic data and attributes of the data engineering projects they typically attempt. An overview of participants and their attributes is shown in Table 1.

For a wide range of insights, we were able to sample perspectives from different job roles working directly with data, software, or management. Participants are employed in academia, industry or nonprofit organizations and work on diverse project types from data curation to hackathons. Typical data domains cover open data like transport or geographical, heavily regulated ones like medical or financial and complex scientific ones like material science.

Participants mostly work with open data in a non-professional context. However, we decided to also include projects that included collaborative data engineering on closed data or in professional contexts to gain additional insights.

3.2.2 Data Collection. We designed semi-structured interviews to collect qualitative data. We developed the interview guide according to the steps identified in Kallio et al. [11].

First, we identified the prerequisites for using semi-structured interviews based on the insights from literature during the SLR. This previous work helped us to conclude that data on roles and social interactions during

collaborations was missing from the literature, but could be answered by asking practitioners. We also had already identified some challenges to collaborative data engineering that we could use to create the themes of the interview. Additionally, the data from the SLR also completed the second phase, retrieving and using previous knowledge.

Based on this knowledge, we developed an initial interview guide that was presented and discussed in internal testing with other researchers during a peer debriefing session (see Table 2). While interviewing participants, we always ended the interview with a question about topics we did not cover or should be asking about. With the feedback from participants, we continuously revised the interview guide. The final document is included in Appendix A.2.

The interview guide includes the main themes that we wanted to ask about to answer the research questions. These were:

(1) Demographic data
(2) The concept of collaborative data engineering itself
(3) Social systems in collaborative data engineering (Roles, interactions, and tools)
(4) Challenges to collaborative data engineering (Social, cultural, technical and previously identified challenges)

Additionally, every theme included some further question prompts to remind the interviewer of questions to ask. However, if the interview naturally flowed from a topic, we allowed for deviation from these detailed questions as long as all major themes were covered.

The interviews themselves were conducted electronically, over Zoom. We provided an interview handout describing research context, the interview process and how we would manage the resulting data to every interview participant a few weeks before the interview itself (the handout document can be read in Appendix A.1). We restated the interview process and important definitions of the research context before conducting each interview.

After an interview completed, we transcribed the recordings and replaced any personally identifying information with pseudonyms. The resulting transcript was then shared with the interviewee, asking them to correct any mistakes and provide a final acknowledgment of their consent to the transcript being used. As a result of interviewee feedback, we fixed some errors in tool names but made no changes to the content of interviews.

3.2.3 Data Analysis. As in the SLR, we used descriptive data synthesis according to Kitchenham [12] to analyze the interview transcripts. To do so, we set up data extraction categories:

(1) Social Systems
  (a) Roles
  (b) Interactions
(2) Challenges to collaborative data engineering
  (a) Social / Cultural Challenges
  (b) Technical Challenges
  (c) Previously Identified Challenges

With these categories in mind, we read through every transcript, highlighting and classifying sections from it according to the categories. We then combined similar segments into a topic with a brief description. This way, we arrived at a list of extracted topics and quotes from interview participants to support them.

In a final step, we further grouped the identified roles into three categories of project group, auxiliary roles and data community as we gained more understanding about the differences from interviews.

3.3 Quality Assurance

We employed peer debriefing sessions [24] to increase the credibility of the results. In these sessions, we discussed aspects of the research design and results with other researchers that had experience with the research methods used but were not involved in the topic or execution of the research itself.

	Method	Participants	Topic
#1	Peer Debriefing	2 Researchers	Search Strategy & Results
#2	Member Checking	1 Open data expert	SLR results
#3	Peer Debriefing	2 Researchers	Identified challenges
#4	Peer Debriefing	2 Researchers	Research design, interview study
#5	Peer Debriefing	2 Researchers	Interview process

Table 2. Feedback methods used

Participants and topics of the feedback sessions are shown in Table 2.

In the first peer debriefing, we presented the systematic literature review with a focus on the search strategy and initial results. From the feedback, we adapted our presentation to include more details about how we arrived at the final set of articles. In a second peer debriefing, we then discussed the challenges we had identified from the literature.

The follow-up interview study was also discussed in peer debriefings. We first presented our planned research design, including how it was informed by the previous literature review. In an additional meeting, we gathered feedback on the interview process itself from experienced interviewers to ensure we were conducting the interviews appropriately.

We presented the results of the literature review to an open data expert for their feedback about completeness as a member check [6]. For this, we created a handout document including the research context and asked if we had identified elements they thought were wrong or missed any important elements. Their feedback included some new elements, but there were no incorrectly identified elements. After adding the new elements, the expert confirmed they had no additional comments.

4 RESULTS

4.1 Elements of collaboration systems for data engineering described in literature

We extracted participants, activities, created artifacts, and tools used during data engineering by open data users from literature to answer RQ1, *which elements of collaboration systems for data engineering by open data users are described in literature?*

4.1.1 Participants. Working with data is a complex activity involving multiple skill sets. Therefore, participants in collaboration systems for data engineering come from various backgrounds, shown in Table 3, each contributing their expertise. Unsurprisingly, data scientists are often part of projects involving data engineering. When working with open data, open data experts can contribute knowledge about data sources or, together with legal advisors, help navigate the legal framework for data use. Lastly, making data usable is also an engineering challenge, which means software developers are an essential part of collaboration systems for data engineering.

Subject-matter experts play an important role in data projects, especially in the more complex problems that can be found in open science. Often, researchers are part of collaborative data engineering projects in the role of subject-matter experts. They help collaborators understand the meaning of data and assess data quality.

Participants	
Businesses	Mediators
Citizen Scientists	NGOs
Civil Servants	Open Data Experts
Data Scientists	Organizations
Subject-matter Experts	Private Citizens
Government Agencies	Researchers
Hackathon Participants	Software Developers
Infomediaries	Startups/Entrepreneurs
Journalists	Students
Legal Advisors	

Table 3. Participants in data engineering

Commercial entities also participate, from large businesses that use open data to improve their existing products to startups that innovate with new applications using only open data. Depending on the company, participation in collaborative data engineering varies from active contribution to passive consumption of the final result.

A special position inside of open data ecosystems is taken by intermediate entities called infomediaries [31] that are located between open data producers and consumers and add value to data by processing it. These participants take in raw data and improve it for multiple downstream projects, a central part of open collaborative data engineering.

Besides commercial use, open data is primarily used in the context of open governments. Actors from public administration, like civil servants and government agencies, not only publish open data but also reuse data for their projects. Interested citizens interact with open data as journalists, as members of NGOs, or as students during Hackathons.

Common to the use cases of open data by students, citizen scientists or hackathon participants is a low amount of organization and direction, an environment that open collaboration could be productive in.

Acquire	Assess	Communicate	Extend	Improve	Maintain	Understand
Build Infrastructure	Ensure Anonymity	Ask Publisher	Add Metadata	Aggregate	Archive	Analyze
Discover	Evaluate	Discuss	Create Features	Clean	Document	Ask Experts
Extract	Preview	Find Community	Label	Combine	Refresh	Experiment
Read Documentation	Measure Availability	Find Skilled Users	Rate	Curate		Learn subject-matter knowledge
Search	Verify License	Give Feedback	Translate	Enrich		Learn Structure
Select	Visualize / Plot Data	Request Data		Link		
Store		Share Data (Publisher)		Normalize		
Validate		Share Data (Stakeholders)		Reformat		
		Share Information		Repair		
				Structure		

Table 4. Activities performed during data engineering by open data users

4.1.2 Activities. We could identify a large list of activities that are attempted as part of data engineering by open data users. All activities, as well as larger themes, are shown in Table 4. The overview includes all activities that were described as part of collaborative data engineering in the literature, but most projects only include a subset of activities.

At the start of any data-driven project, users must first source a usable data set. To do so, they perform an iterative cycle of activities related to *acquiring*, *understanding* and *assessing* data.

Activities related to the acquisition of data begin with data discovery, either organically or from directed search. Users have to extract data and store it in a system that is fit for their use case. Because data sources are not standardized or have download limits, extracting data often requires reading provided documentation, building custom tools to interact with APIs and finding storage space.

Once data has been acquired, its usability has to be assessed. Dealing with licensing issues, making sure data is correctly anonymized, and verifying the data source has sufficient availability are common problems at this stage. Aside from technical and legal issues, data content and structure has to be understood to assess it usefulness for a project. Users engage in exploratory data analysis, using tools to preview data content (e.g., by plotting it) or data structure. Subject-matter knowledge is a requirement for working with more complex data sets and has to be either learned by the data users themselves or by finding and collaborating with subject-matter experts.

After acquiring, understanding and assessing data, data users process it, either by *improving* it or by *extending* it. During these activities, technical knowledge is required as users change data formats and structure, normalize values and fix errors. An activity that is challenging but adds a large amount of value to a data set is linking it with other sources.

Extending data most often takes the form of providing additional metadata, for example by writing usage reports or rating data sets on open data platforms. From expert feedback, we also included translating data as an activity, this can take the form of translating structural aspects like column names or content like the names of cities for open mobility data. If a data set is supposed to be the basis for machine learning projects, labeling data and creating features are common activities as well [19].

At the end of a collaborative data engineering process, activities related to the *maintenance* of the results, like archiving the resulting data, are required. An important but often neglected activity is the documentation of a project, learnings about data content and structure and the reasoning behind data engineering decisions. Some data domains like mobility deal with regularly updating data (e.g., public transport schedules that are released every few months) so data users must build infrastructure to refresh source data.

Underlying all these activities is *communication* with other participants. To date, concrete communication about a data set mostly flows from data users to data publishers in the form of questions, feedback or requests for more data. This interaction between data publishers and consumers is expected and supported by many open data portals. Direct communication among a larger data community is rarer and is mainly related to searching for other participants with a missing skill set or subject-matter knowledge.

4.1.3 Tools and Artifacts. We could not identify a standard tool used in collaborative data engineering among open data users. A summary of tools described in the literature is shown in Table 5. Noteworthy is Open Refine, which was mentioned multiple times.

Tools used by participants in collaborative data engineering range from self-developed using general-purpose programming languages to existing applications like sheet software or Wikis, depending on the technical skill level of project members. We included custom-made *Software Applications* from expert feedback as it was pointed out that collaborators not only develop open-source software but also share closed-source software applications like validation tools with the community.

As mentioned for the activities, understanding and assessing data often requires data exploration. Visualization tools provide a fast way to check data quality and content. More permanent solutions like automated data pipelines are developed using general-purpose programming languages or Jupyter notebooks with the help of classical software engineering infrastructure like git and GitHub.

Unlike project forges like GitHub for software engineering, open data portals play nearly no role in fostering collaboration among a community. They are mentioned often in the literature, but nearly always only as a data

Tools used	
Auth Providers	Kaggle
Big Data Processing Tools	Notebooks
Blogs / Websites	Official Discussion Board
Command Line Tools	Open Data Repositories
Data Science Libraries	Open Refine
Databases	Sheet Software
Domain-Specific Languages	Statistical Computing Languages
Domain-Specific Software	Translation Software
General Purpose Languages	Travis
git	Visualization Tools
GitHub	Wikis

Table 5. Tools used during data engineering by open data users

source and not as a platform to connect data projects. Whereas GitHub is the de facto standard to find software engineering projects that are open to collaboration, too many unrelated open data portals exist for any one of them to play a similar role.

Documentation of data projects, experience reports with data sets or expert advice is therefore scattered, and data users have to write blog posts, participate in discussion boards and read publisher websites.

Created Artifacts	
CI Definitions	Notebooks
Comments on Data	Processed Data
Data Quality Ratings	Raw Data
Documentation	Software Applications
Feedback-/Experience Reports	Source Code
Metadata	

Table 6. Created artifacts by open data users during data engineering

Similar to tools, no standard artifact exists that open data users collaborate on. Table 6 shows a summary of artifacts from the literature, most of which are metadata like comments or software artifacts to handle data. The processed and improved data itself was created as part of the data engineering process, but seldom shared with the community. Open data portals often do not enable users to contribute any improvements to a data set back to the publisher, and tools like GitHub that are made to share source code are not well suited for sharing most data set formats.

4.2 Social systems in collaborative data engineering

We have identified roles and interactions from interviews with practitioners that work on data-driven projects that include collaborative data engineering to answer RQ2 *How and in which roles do participants in collaboration systems for data engineering interact socially?* Here, we initially present an overview of the roles and highlight essential interactions or those unique to collaborative data engineering in detail. As we were interested in a

description of the diversity of roles and interactions, not every collaborative data engineering effort necessarily includes all roles and interactions mentioned here.

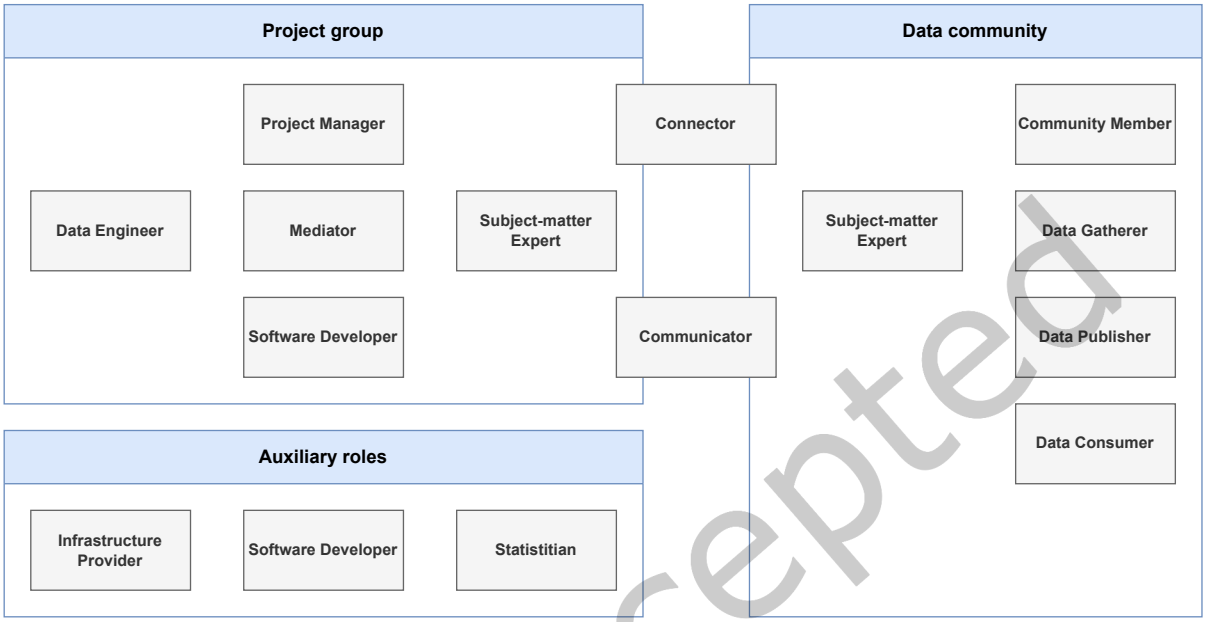

Fig. 4. Roles in collaborative data engineering as identified in interviews

When summarizing the roles that were described by interview participants, it became clear that collaborative data engineering interactions happen between three larger groups, as shown in Figure 4.

At the core, each data-driven project was attempted by members of a project group. This group is made up of participants that organize around a particular goal and data source, processing and publishing data towards archiving that goal.

Because of the interdisciplinary nature of data science, data processing inside the project group is driven by the three roles of data engineer, software developer and subject-matter expert. These roles provide their respective insights for working with the data, with the data engineer contributing knowledge about data formats and algorithms, the software developer providing infrastructure or technical requirements and the subject-matter expert explaining data meaning. Often, especially in smaller teams or in open data contexts, the role of data engineer and software developer are combined in a single contributor with a technical background.

Between these roles, a mediator must translate from the technical viewpoint to a subject-matter viewpoint and vice-versa. Mediation is a core interaction in collaborative data engineering projects, shown in Figure 5. A contributor assumes the role of a mediator and provides subject-matter experts with help in case of technical problems. On the other hand, they must explain the subject-matter to the other contributors, like software developers and data engineers. Often, the mediator role is filled by technical members of the team that, over time, learn enough from subject-matter experts to teach other technical contributors. In some projects, mediators are the subject-matter experts that get technical feedback from software engineers. As one example, E8 assumed the role of a mediator because they had a software development background and worked with subject-matter experts that had no technical experience: “It was the analog equivalents of them. So geographers, financial experts, but

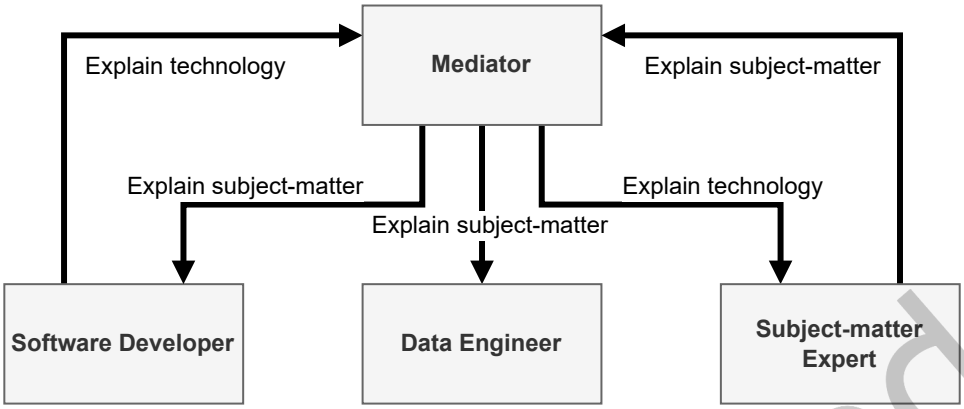

Fig. 5. Mediate during collaborative data engineering

they don't have a lot of digital experience or can only scratch the thing at the surface. So there was a big gap between basically working from a really technical side to a really non-technical side." On the other hand, E1 saw part of their role in educating the data engineers on their team about the subject-matter due to their scientific background: "I just put people in touch with each other's teams and see if I can translate some of the science to the data people."

Finally, project teams include a project manager role. In commercial projects, this role does traditional project management work like defining a list of priorities and planning tasks. In comparison, open collaborative workflows rely on the self organization of participants. Therefore, the main contribution of a project manager role to open projects like hackathons is suggesting an idea and making sure participants align with it. E6, an experienced host of open data hackathons, calls the role 'idea owner' and describes it as: "[...] usually the person who pitches the challenge or proposes that use of that data set at the beginning of the event, but then sticks around and makes sure that the idea gets worked on."

Due to their collaborative nature, all data projects we have interviewed participants from are embedded in a larger data community that is not directly involved in the project but interested in the same data set or subject. Subject-matter experts can be part of the data community as well, and only periodically contribute to a collaborative data engineering effort without being part of the project group, either by finding the project by themselves or by someone from the project group reaching out to them.

Naturally, data publishers and data consumers of a specific dataset are members of this community. In some projects, data gatherers are also involved if a dataset is created from individually collected data points. Because all of these roles directly work with the data, they have a shared interest in collaboratively developing a consistent data schema, as shown in Figure 6. During this activity, project managers have to, on the one hand, stay in constant contact with data publishers and data consumers in the community (using connectors) and subject-matter experts and software developers on the other hand. Data publishers and software developers describe technical, availability or legal limitations to what and how data can be used. Inside those parameters, subject-matter experts can help formulate requirements for data schemas to capture the data domain adequately, while data consumers describe their projects and what information they are missing in the available data.

Additionally, in a broader sense, a data community is made up of individual community members that share a common understanding of data semantics and expectations towards data projects in their space. E6 expresses the concept of a data culture as follows: "I think there is something like the notion of a data culture. [...] Individual

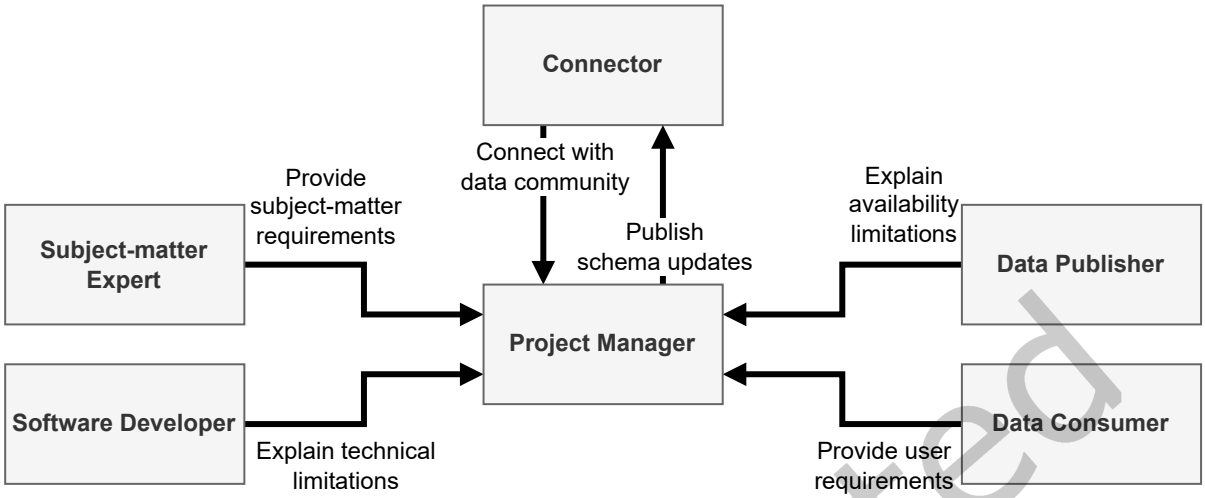

Fig. 6. Develop data schemas during collaborative data engineering

people grow up with more or less expectation around things like data privacy or just maybe rigor." Different data cultures can form around the same data set. Examples in the domain of open government data include the very rigorous, careful approaches of working with data by federal statistical offices, and the more playful, result-driven projects attempted by political activists during hackathons.

Bridging the gap between project groups and the larger data community are connectors and communicators. Whereas communicators describe and share the results of the work of project groups with the larger community, connectors actively bring together different roles in the social system.

The role of communicator can be performed by members of the project group itself, especially during hackathons, as E6 describes: "People who love visualizations and infographics are typically people who just communicate well, they would describe the the problem, the solution and the steps to get me to reach it using various media. Social media posts or illustrations [...]". Also performing this role are journalists who report about data projects. Because a report based on data is a potential downstream project for a data set, journalists also help to provide requirements and prioritization of what to work on. Occasionally, they can even contribute subject-matter expertise as E8 experienced: "Sometimes the journalists can help interpret the data, because they can help clarify the keys or what to look at for in the data."

In contrast to only sharing results with the data community, connectors work to bring together different entities in the ecosystem. In open collaborative projects, participants need to find each other to work together. Here, connectors are part of the ecosystem as networking organizations such as the Open Knowledge Foundation or data hubs like the German GovData.

In professional projects, connectors include data marketers and customer support employees that gather data use cases from customers. Prioritizing what data to work on and how a final data product should look like is an important interaction for these collaborative data engineering efforts. Figure 7 shows the different roles that interact during it. E1, who is working on a commercial data product, describes the need for insight from customers: "[...] everybody has a different need for that data set or the use case for a particular data set is so diverse. One person just wants it as a reference. [...] And then there's the whole AI/ML guys who want to use it to train models. [...] that's another thing we have to explore.". In their data project, these user requirements

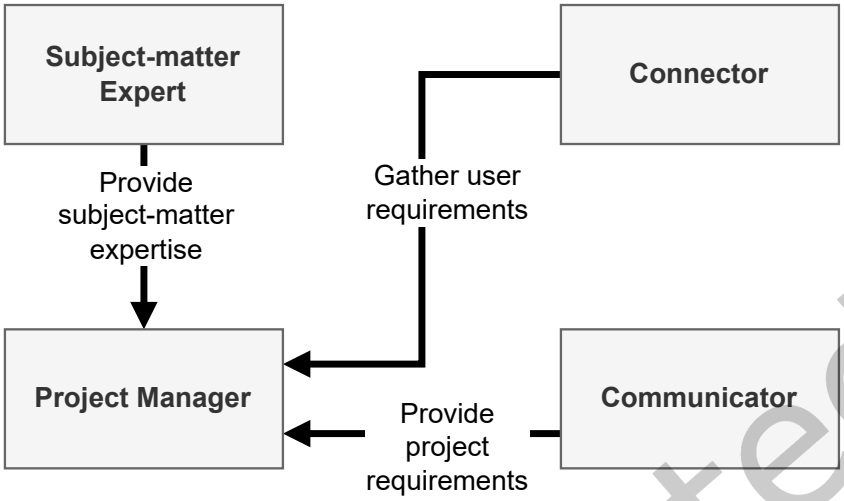

Fig. 7. Prioritize during collaborative data engineering

are gathered by data marketers. For hackathons with open data, requirements often arise from how the data is planned to be used by participants like journalists that fill a communicator role. In any case, subject-matter experts have to share their insight into what data is of high enough quality to be used and how important it is in the context of the planned use.

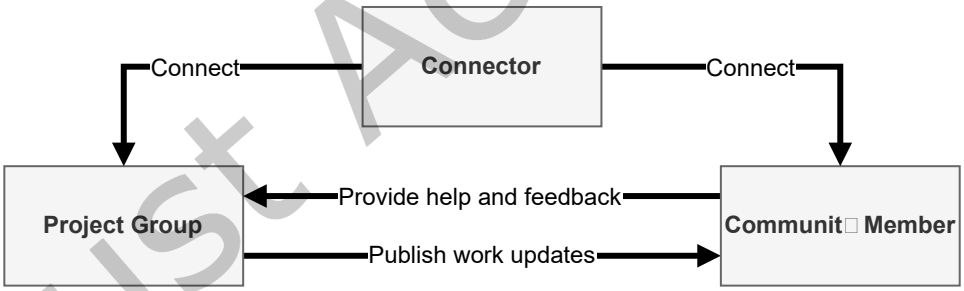

Fig. 8. Public work during collaborative data engineering

Increasingly common in open data projects is the concept of public work (see Figure 8). During public work, participants share updates about their progress and problems on social media or even livestream their work on services like TwitchTV. E6 mentions public work from their experience with open data hackathons: "People engaging in activities which they immediately communicate through some social media channel. Things like the way some people use LinkedIn or Twitter to announce very regularly their activities, their projects, their progress reports, all the way to things like streaming." This form of working is made possible by connectors that bring together interested members of the data community and members of a project group that are open to feedback or want to increase the visibility of a data product. In the case of livestreaming, this form of working

leads to a new way of gathering fast feedback and help from other contributors. E6 goes on to say: "So you have a channel where people post commentary. They drop in your channel to say, why are you doing this and not that?"

Finally, the project group is sometimes supported by auxiliary roles that provide services related to data engineering but do not participate directly in a data project. In open data contexts, these auxiliary roles are mostly related to providing infrastructure (like open data portals) or software developers that contribute open-source tools. In commercial settings, we have also encountered other, more highly specialized roles. As an example, due to the required rigor in the medical and biological data space that E9 works in, they are supported by statisticians: "[...] we have a statistician team. So, we are the data team and we send the data to the statistician team [...] they do the statistics.".

4.3 Challenges to collaborative data engineering

In this chapter, we highlight challenges that have either been described in the literature or been mentioned by interview participants to answer RQ 3, *What are challenges to collaboration in data engineering and why?* We report on challenges in three categories, previously identified challenges from literature, technical challenges from interviews and social challenges from interviews. As with roles and interactions (see subsection 4.2), the list of challenges highlighted here is not exhaustive but focuses on challenges that are either essential or unique to collaborative data engineering.

ID	Type	Title	Mentioned by
C1	SLR	Need for specialized skills but high barriers to participation	E1, E2, E8, E9
C2	SLR	Finding and connecting with community members	E1, E2, E6, E8
C3	SLR	No well-understood collaboration practices	E1, E2, E6, E8, E9
C4	SLR	No standard tools or artifacts	E1, E2, E6, E8
C5	Technical	Data representation	E1, E2, E6
C6	Technical	Inadequate tools	E2, E8, E9
C7	Technical	Infrastructure for data projects	E8
C8	Technical	Bad data sources	E1, E2, E8, E9
C9	Social	Conflicts with data publishers	E1, E2, E8
C10	Social	Unclear data use cases	E1, E2
C11	Social	Data semantics	E1, E2, E6
C12	Social	Missing incentives	E1, E6, E8
C13	Social	Missing knowledge	E1, E2, E6, E8, E9

Table 7. Challenges identified from literature and interviewee experiences

From the structured literature review, we identified four challenges [7], summarized in Table 7 with the type SLR. In all but one of the semi-structured interviews with practitioners, we explicitly asked for their experiences with the challenges and if they had also experienced them. For one interview, E9, we did not specifically ask for the challenges due to time constrains and a larger language barrier. Instead, we consider their confirmation of a challenge when they described a similar challenge they experienced themselves. Overall, the practitioners overwhelmingly experienced the previously identified challenges themselves.

C1, the need for specialized skills but high barriers to entry, could be confirmed by nearly every interview. Data engineering requires both subject-matter knowledge to understand data and technical and data skills to work with it. This effect also explains the roles of data engineer, software developer and subject-matter expert that we described in subsection 4.2. It is a challenge to find participants to fulfil all those roles, especially in small teams

or open data projects. In a survey among researchers, Kjærgaard et al. [13] found that only 7% of respondents were comfortable using RDF-files, meanwhile RDF is a standard data format for semantic web applications. One notable exception to the confirmation from interviews was E6, who disagreed with the need for specialized skills described in C1 because they host inclusive hackathons in which participants can try out unfamiliar roles and still contribute.

The challenge of finding and connecting with community members, C2, was also described in literature [21]. In this regard, open data projects lag other open collaboration ecosystems like open-source development or wiki content authoring. Contributing to this challenge is the fact that data portals are less frequented for data, as E8 describes: "For example, you found a software project, on GitHub, then it is really easy to collaborate. You can just open issues, fork it, contact the original maintainers of the project. But when you're on data portals for example, it is much harder because they are less frequented. You don't know who posted the data, maybe it was some kind of government agency or anything like that. So there is no one who feels responsible." Additionally, data portals are focussed on hosting data sets and are missing most social features of other project forges like GitHub or Wikipedia, making the discovery of other users hard. E8 goes on to elaborate: "And the users of the data sets are not visible anywhere. So that's a problem on the data portals, at least I know. It is actually really hard to get into contact with people that are working on it."

Well-understood collaboration practices specific to data engineering are missing, as stated in C3. With hard to discover data communities, this challenge leads to a split between data publishers and data consumers, with few data consumers working together to improve the overall data for every project. Contributing to this challenge are missing tools, prescribing a collaborative workflow that a community can follow. In their interview, E2 notes: "I have the feeling that all the kind of GitHub contribution model came with git and GitHub together, and Wikipedia came with a wiki and maybe we need the tool. [...] It will be very hard to define a workflow if you don't have some tool that kind of works." Existing project forges for software are used, but are missing features [4] and do not enforce a fitting workflow.

Overall, standard tools and artifacts to collaborate on are missing for collaborative data engineering (C4). In open-source software development, participants collaborate on clearly defined artifacts like libraries and frameworks, shared as source code. Currently, most artifacts that are created during collaborative data engineering are either metadata or the resulting data itself (see Table 6), which leads to a mismatch between data artifacts and tools that were made for software as E2 describes: "[SQLite] won't work with git, which works well with text files but not very well with binary files." Collaborating on data sets is also not a viable strategy when data needs to be regularly refreshed to stay up-to-date [26] or for domains in which the data is regularly updated. From their work with open transport data, E2 points out the potential of a standard workflow language that can be used with traditional software tools: "Maybe there will be at some point some workflow process language that will be accepted and will be universal enough, and that can be versioned on GitHub, where you can then have your own version and changing one bit of the process or fork it and rerun it. [...] You need code, you need data and you need workflows. And you want to be able to treat each one of the three of them."

From the interviews, we identified additional technical challenges, summarized in Table 7 with the type technical. Data representation, C5, is a challenge for any data-driven project, but more so for collaborative projects. Obvious issues are different syntaxes to represent the same underlying value, like the use of a comma or a dot to separate decimal numbers and the use of different units such as Celsius or Fahrenheit for temperature. Typing of values is a related problem, with many domain-specific value types, like postal codes, being unclear or lost in data transfer. This is especially problematic for open data because it is often shared in the form of CSV-files that cannot express custom value types. In addition, interview participants also experienced problems with missing standard taxonomies to create a shared understanding of data, for example in new scientific fields that have many research groups working in them.

Practitioners also highlighted that the tools they are using to work with data are inadequate (C6). Simple data wrangling tools work with small data sets, but cannot deal with larger amounts of data. Programming languages are complex and hard to use, especially for subject-matter experts. Likewise, tools to set up data pipelines are frustrating as E2 explains "So much configuration, so much code, and so much things I copy pasted and I don't how it works [...] And this gets very frustrating. If it's in tools that you do configure by a kind of programming. And if it's a graphical one, you feel frustrated because you spend your time clicking and you don't know where." Then, once a data pipeline is set up, there is little tool support to keep the data pipeline, its executions and the resulting data in sync. For their projects, E8 set up custom code to keep track of all pipeline runs and where the resulting data was saved by updating a database. With the need to regularly update data, this led to a large overhead of custom tool development.

Related to those problems, E8 also pointed out how much harder it is to host infrastructure for data projects (C7) than it is for fields with more existing standards, like mobile application development. This includes infrastructure providers for data (e.g., data lakes) but also for backend code like cloud providers. Even for theoretically simple tasks like hosting code that regularly fetches data from a source and loads it into a data sink, no common patterns exist that are used across a data community. Therefore, data engineers write custom solutions that others have to understand before they can collaborate.

The final technical challenge to collaboration are the bad data sources (C8). These issues include problems with the data itself, like no documentation or missing units for values that need to be verified with subject-matter experts, or semi-structured data like text that is hard to use. Challenges with data sources also extend to how they are made available, for example because of flaky infrastructure or old technology. E8 describes their experience working with data from a news publisher: "You basically get access to a FTP servers via really unsecure connections. It's like in the Middle Ages [...]". For data that is regularly updated, like open transport data, an additional challenge lies in the fact that collaboratively fixing errors in a data set and sharing the corrected data is of limited use because the next time an update is released, the same errors will be present if the fixes are not shared and accepted by the data publisher.

Collaborating with data publishers, for example by contributing back fixes for errors, is often not easy. This challenge is part of the social challenges we have extracted from interviews, shown as C9 in Table 7. For the open data practitioners we have interviewed, it is a problem to get into contact with data publishers and if they can, often no one feels responsible. For the larger data community, this is a challenge to collaboration because important participants from the publisher side do not contribute, either with knowledge about the data structure or by accepting feedback from the community. One of the reasons for data publishers not participating in collaboration is a territorial feeling about the data, a sentiment expressed by E2 as follows: "I published the data so I know how it should be and the community is wrong." These feelings are reinforced by fears from data publishers. Depending on the context, data publishers might fear contributors entering bad data (in the case of open data collaborations like OpenStreetMap) or fear users misusing data to harm others, for example by misrepresenting statistical data. Lastly, data publishers that are forced by law to publish open data, often government agencies, fear the loss of potential business value that is then captured by commercial entities or startups.

These unclear use cases for data also create a different challenge, captured in C10, in which data publishers and potential collaborative date engineering projects are unsure how downstream consumers will use data and what requirements they will have. Because of this, collaborative data engineering efforts have no clear way to evaluate if a data processing step is essential to increase data quality. This leads to costly overhead to reach out to potential consumers or to collaboratively define data schemas as a community, as described in subsection 4.2. The increasing popularity of machine learning poses a special challenge in this regard because their data requirements are unique. As an example given in interviews from the mobility domain, machine learning models can cope

with errors in training data reasonably well, but if a data set with accessibility information about wheelchair access includes only a few errors it makes for low-quality data for a mobile app that supports disabled users.

Similar to the technical challenge with data representation described as C5, different impressions of data semantics are a social challenge identified as C11. This challenge manifests itself as many viewpoints on data meaning from different data communities. For open science data, it could mean research communities using lacking standard naming for concepts. Naming issues are also described in more mundane examples, like long discussions about what constitutes a shelter at a bus stop. In international data context, simple naming issues like what to call a subway/metro/underground-train can confuse users. Aside from naming issues, political differences can make collaboration a challenge as well, often concerning geographical data like borders. Even outside conflicts, E2 describes a situation where the border between Germany, Switzerland, and Austria is not clearly defined at the Bodensee, leading to complications with the strict requirement for borders to exist in the data schema of OpenStreetMap.

For collaborative open data projects, missing or misaligned incentives pose a challenge. Especially because data engineering can be, as multiple interviewees point out, "boring work". Publishers of open data are often forced by law to make their data available, which leads to conflicts, as described in C9. Other domains face similar problems. For scientific data projects, researchers are rewarded less for curating and maintaining data and have to focus on publishing instead. For open collaborations by a data community like hackathons, different forms of incentives like certificates of participation or cryptocurrency are mentioned by interview participants. If a hackathon is run by activists, it is hard to provide similar rewards.

The feeling of ownership of an artifact like a software application or a wiki article can be an incentive to collaborate without a monetary reward. But as E6 points out, with data the ownership typically lies with a data publisher and not the community: "In many collaborative contexts, it can be difficult to really have a sense of ownership over a data set that is produced by a government or some kind of external company. It is quite rare that people produce their own data. So it's rare that people can invite each other to work on data that they really feel a sense of ownership for."

Lastly, collaborative data engineering projects suffer greatly from missing knowledge (C13). Data projects are challenging because of the need for knowledge about data engineering, software engineering and subject-matter, as identified by three roles in a project group (see subsection 4.2) and the fourth mediator role. Every interview included a discussion of this challenge. Specific examples include a lack of knowledge about data pipeline tools by subject-matter experts or unfamiliarity with data formats. Some projects, like the heavily regulated medical data E9 dealt with, also have challenges with the missing legal knowledge of contributors. Because the skill sets needed to work with data are so distinct, it is often challenging for contributors to correctly assess the level of knowledge of others. From a software developer viewpoint, E8 explains: "Regular people, don't how any data format works. You even have to start explaining what a key value store is. [...] With non programmers you have to first start on this level, which is difficult."

Missing knowledge means that participants must rely on learning information either from subject-matter experts or technical members of a community. If these roles are assumed by few people, the danger of overburdening them with too many questions exists.

5 DISCUSSION

Our results show, that collaborative data engineering projects are part of a fragmented ecosystem. The number of stakeholders involved, subject-matter differences and data cultures with their unique viewpoints and standards make it hard to successfully attempt open collaborative projects with data. It appears as if the open data ecosystem is developing similarly to open-source software but lagging years behind.

Increasing open collaborative data engineering projects would have wider implications on data ecosystems. The ability for data users to share and reuse improved data could lower individual costs. In turn, this would raise the quality and availability of data for the whole ecosystem. With easier to use data, individual projects like hackathons would lose less time to data engineering and could invest more into innovative applications of data.

The availability of high quality, open data sets is especially important with the recent increase in machine learning and artificial intelligence projects. Machine learning models need large amounts of machine-readable data for training and evaluation, meaning they are hard to develop outside commercial contexts. Democratizing access to the underlying data can enable more participants to develop their own models and evaluate existing ones critically.

Building on the insights described in section 4, we contribute a set of guidelines for successful collaborative data engineering projects, summarized in Table 8. In accordance with our research questions, we focus on the social systems and challenges that arise from collaborative work and less on the inherent technical challenges of individual data engineering. These guidelines provide a preliminary framework for practitioners to increase the adoption of collaborative data engineering in their projects. For researchers, these guidelines provide a starting point to extend them into a more complete theory of collaborative data engineering.

Following these guidelines, we make concrete recommendations to increase the adoption and success of open collaborative work during data engineering in the context of open data. An overview of these recommendations is shown in Table 9. Similar to the guidelines, these recommendations should be understood as preliminary. Policymakers and open-data enthusiasts that want to increase collaboration in data engineering should take these recommendations into account, but be open to changing their approach with additional insights.

5.1 Guidelines for Open Collaborative Data Engineering Projects

ID	Guideline	Based on
G1	Plan with data problems like distributed sources, updates, low-quality and limited access to publishers	C5, C8, C9, C12
G2	Make projects accessible to data engineers, software developers and subject-matter experts	Social Systems, C1, C13
G3	Enable collaboration by agreeing on standards, improving project visibility and curating data	C2, C3, C4, C5, C7, C8, C12
G4	Support projects with tools, built specifically for collaborative data engineering	C1, C4, C6

Table 8. Guidelines To Enable Open Collaborative Data Engineering

For all attempts at collaborative data engineering, it is essential to take the realities of data issues into consideration and not plan with an idealized view. In most contexts, the data will be distributed over many locations, regularly updated, of varying but often low quality, hard to access and missing metadata (C5, C8 as described in Table 7). Additionally, especially for open data, data publishers are usually hard to reach and have misaligned incentives that make them unlikely to contribute or resolve issues with their data (C9, C12). These challenges have to be considered in the planning of collaborative data engineering projects, for example by focusing on supporting data users to help themselves instead of trying to improve the source data directly.

Based on the roles in collaborative data engineering, summarized in Figure 4, the focus of collaborative data engineering projects should initially be on the project group as its members are part of every social interaction we have identified. While the project manager and mediator role can be flexible, different viewpoints arise from the triangle of data engineer, software developer and subject-matter expert. Any collaborative data engineering

project must make sure to be accessible and support members with these backgrounds. If a project group is lacking one of these essential roles, adding a member that can fill it should become the highest priority (C1).

With a stable project group, open collaboration with a larger data community can be established, allowing participants to share and re-use artifacts and lowering individual costs. To do so, it is essential to agree on how to share intermediate work results and collaborative workflows (C3 and C4). Finding and connecting with other community members will be a challenge (C2) that must be considered. Potential solutions include improving project and user visibility, providing a robust search, or supporting community members with connector or communicator roles.

Finally, projects should be supported with tools that are specifically built for collaborative data engineering (C4, C6). While collaborative data engineering shares many similarities with open-source software development, the reuse of software engineering workflows and tools for collaborative data engineering work might, in fact, be a detriment to experimentation because they are 'good enough' but not ideal. As described earlier, these tools must be accessible not only to software developers but also data engineers and subject-matter experts. Because these tools are mainly created by software developers, care must be taken to include the other viewpoints in their evaluation as well.

5.2 Recommendations to Increase Open Collaborative Data Engineering in Open Data

ID	Recommendation	Based on
R1	Define a standard artifact to collaboratively develop data pipelines	G1, G2, G3
R2	Adapt proven open collaborative workflows for data engineering	G2, G3
R3	Provide a project forge to drive adoption of standards	R1, R2, G4
R4	Develop tools that make running data pipelines and hosting data projects easier	G4
R5	Support the creation of data communities	G3

Table 9. Recommendations to Increase Open Collaborative Data Engineering in Open Data

Shared standards in a community are important for collaborative work. To that end, a standard artifact to collaboratively develop data pipelines should be defined. This artifact can not be the improved data itself because in many domains (for example schedules in open transport data) data sets are regularly updated and re-released, while contacting the data provider is challenging as described in C9 and C12 (see Table 7). If the cleaned data is shared as collaborative artifact without being able to fix errors at the source, every time new data is released, additional effort will be required to make the same changes again. Instead, the artifact should describe data pipelines that can be re-executed once the data source changes and apply the previously defined transformations and improvements again. Various options to model data pipelines exist. However, they are often commercial, leading to vendor lock-in and slow innovation. Others are GUI tools, like Apache Hop, that make collaboration complicated. An ideal tool to model data pipelines should be text-based and open to initially reuse the mature software engineering tooling, like version control systems and editors, but allow for rapid evolution of an ecosystem of own tools. As discussed for G2, this artifact must also be accessible for all members of a typical project team. Existing data pipeline solutions, based on general-purpose programming languages and frameworks like Apache Flink, can be used by expert software developers but lead to high barriers to participation, especially from subject-matter experts (C1). A potential collaboration artifact would be a domain-specific language to model data pipelines that can reuse existing software engineering tooling and is intuitive for software developers, but can still be understood by data engineers and subject-matter experts due to its reduced scope and domain-specific concepts.

With this text-based artifact, standard collaboration workflows should be adapted for data engineering. Especially proven approaches from open-source development can be a starting point due to the similarity of collaboration artifacts. However, similarly to tool development, the danger of being stuck with good-enough practices like GitHub's pull request model might stifle innovation that is more appropriate for data engineering. Therefore, existing practices should be evaluated and adapted individually, while ensuring they can be followed by all major roles involved in collaborative data engineering. With inspiration from open-source development, it will be especially important to ensure subject-matter experts are able to participate equally, even if they lack experience in software development.

However, the definition of such artifacts and workflows is not only an academic challenge because they can not enable open collaboration without community adoption. Suggesting new standards must therefore be accompanied by well-crafted tools that provide real value to practitioners to have a realistic chance of being used. In contrast to open-source development, data engineering lacks a centralized and highly frequented project forge that would enable collaborative projects to advertise themselves and be discovered. A project forge for collaborative data engineering projects would be useful in three major ways. First, it can standardize the tools and workflows used in the community by providing them in an easily accessible way, together with project hosting [20]. Additionally, a project forge can be a hub for the curation of high-quality data projects, unifying sources spread of many existing data portals. Finally, a project forge can allow community building by improving the visibility of both data projects and users, reducing challenges like C1 and C2. Initially, this can be done by implementing a search feature, but other possibilities like matching users to projects based on skills or interest using AI algorithms could be explored.

In addition to a project forge, a related software implementation should make running data pipelines and hosting data-driven projects easier. An essential requirement for these tools is to keep pipeline models, executions and the resulting data in sync. Without this functionality, collaborative data engineering projects have to invest a large amount of work in building their own solutions, as described by E8 for challenge C6. Making it easier to execute data pipelines, for example in a standardized cloud environment, enables more contributors to participate in their collaborative development because they need less technical equipment and expertise. This is especially relevant as the skill set needed to build and maintain data pipeline infrastructure at scale is distinct from other software development skills required during data engineering. As such, even project team members that can fulfil the software developer role would benefit from the reduced scope of their responsibilities.

Finally, the creation of data communities should be actively supported. Fundamentally, this means creating aligned incentives (in contrast to the missing incentives described in C12) for all participants in open collaborative data engineering ecosystems. As an example, with easier hosting for data projects and a centralized project forge based on the recommended tools, it will be possible to highlight projects that are only possible because of the underlying data. This in turn will reflect positively on public data providers and create incentives for them to improve the data they publish and to engage in the community. Making work visible to create incentives is also possible by supporting connector and communicator roles, as well as increasing the amount of public work (see 8).

6 LIMITATIONS

The search for our systematic literature review was scoped to Google Scholar and Scopus and only included peer-reviewed articles in English or German. The depth of the literature pool could be improved by extending the search to additional languages or including gray literature. However, we supplemented the data from literature with interviews and explicitly confirmed information extracted from articles with practitioners.

The sampling of participants in our qualitative interview study imposes some limitations on the resulting qualitative data. We sampled for theoretical diversity, actively approaching participants that would provide us

with new insights instead of building a statistically representative sample of the data community. While we consider this choice appropriate for our goals to describe the diversity of social systems and challenges during open collaborative data engineering, we cannot make statistical inferences from the data.

Data extraction was performed by descriptive data synthesis, both for the results of the literature review and for the qualitative interview study. Ideally, we would use additional qualitative data analysis methods to deepen our understanding of the data and describe the relationships between participants in data projects and challenges in more depth.

Bias is a possible threat to validity because open data is often published by government agencies. For this reason, much of the academic literature and practitioners concern themselves with open government data. As our goal was not to attempt quantitative data synthesis, the thread is less relevant, but the danger of missing information from other data domains remains. To mitigate this, we tried to sample practitioner interviews from other data domains and asked for practitioner feedback (see Table 2) for the results of our academic literature review.

The presented guidelines and recommendations in section 5 are based on insights gained from the literature and interviews. However, they were not independently evaluated. In combination with the non-representative sample of interview participants discussed earlier, these contributions should be understood as preliminary. We took care to point this distinction out and separated their presentation from the results of the systematic literature review and interview study itself.

7 CONCLUSION

In summary, we aimed to answer three research questions related to open collaboration during data engineering by open data users: *Which elements of collaboration systems for data engineering by open data users are described in literature?*, *How and in which roles do participants in collaboration systems for data engineering interact socially?* and *What are challenges to collaboration in data engineering and why?*

We provided an overview of elements in collaboration systems for data engineering by data users by performing a systematic literature review. We find data users from heterogeneous backgrounds that use a variety of tools to process data. The collaborative data engineering processes described in the literature include a wide range of activities, from technical, like writing scripts to fix errors, to social like finding and asking subject-matter experts for advice. While we could identify and categorize individual activities, we could not find a standard collaborative process that is followed across all data engineering projects.

To describe the social interactions in data projects in more detail, we performed a qualitative interview survey with participants. From the interviews, we extracted descriptive data about roles that contributors fulfil, as well as unique social interactions. The results indicate that the five roles of data engineer, software developer, subject-matter expert, mediator, and project manager make up the core of a collaborative data engineering project (for an overview, see Figure 4). We identified additional roles in the larger data community, as well as supporting roles that contribute special skills when needed. We describe important interactions between the roles that must happen to mediate between the different viewpoints, collaboratively define data schemas or prioritize data. Additionally, we found that the recent trend of public work also applies to collaborative data engineering projects, especially hackathons.

Based on qualitative data both from literature and interviews, we identified challenges to collaborative data engineering. Due to the complex, collaborative work required, these challenges are both technical and social. Technical challenges relate mostly to bad quality data sources, tools that are no good fit for data engineering and the complexities of hosting and maintaining data projects. Social challenges stem from the interactions in the larger data community, especially conflicts and little shared understanding between data publishers, project members and data consumers. For individual collaborative data engineering projects, incentives are

often misaligned or missing. Because many specialized skills are needed to work with data, missing technical, subject-matter or legal knowledge is a central challenge that has to be resolved. A summary of the challenges is shown in Table 7.

Building on our insights from the results, we described guidelines to follow to make collaborative data engineering projects successful, such as planning with data problems from the start and ensuring that projects are accessible to the main project group roles of data engineer, software developer and subject-matter expert. We point out the importance of agreeing on standards, making projects discoverable and curating data, as well as providing purpose-built tools instead of just relying on infrastructure built for collaborative software engineering. To increase adoption of open collaborative data engineering in open data, we made concrete recommendations. First, to create a standard collaboration artifact as well as workflows, then building adequate tooling in the form of a project forge and a cloud environment to execute data pipelines to drive their acceptance. Finally, the creation of data communities must be actively supported by creating incentives and enabling connecting social roles. The guidelines and recommendations can be found in Table 8 and Table 9 respectively.

Our contributions are relevant for a variety of audiences. Practitioners that want to introduce or increase collaboration during data engineering should keep the social systems described in Figure 4 in mind, be aware of the challenges described in Table 7 and consider following the guidelines presented in Table 8 to make their project successful. For policymakers and enthusiasts working with open data, our recommendations (see Table 9) to improve the open data ecosystem are relevant as well. Lastly, researchers can make use of the overview of activities, artifacts, tools, and participants (Table 3 - Table 6) and social systems from Figure 4 to understand the domain of open collaborative data engineering better.

More work is needed to deepen our knowledge about the identified challenges and potential solutions. While we focused on diversity, additional research is needed to highlight the differences between data projects, for example between open and closed data, and compare the social systems they create. The guidelines and recommendations we suggested based on the results should be evaluated and extended by applying them to real open collaborative data engineering projects and observing their impact.

Finally, we would like to extend our work with a larger, quantitative survey among data engineering practitioners to find additional challenges and statistically representative insights.

ACKNOWLEDGMENTS

This research has been partially funded by the German Federal Ministry of Education and Research (BMBF) through grant 01IS17045 Software Campus 2.0 (Friedrich-Alexander-Universität Erlangen-Nürnberg) as part of the Software Campus project 'JValue-OCDE-Case1'. Responsibility for the content of this publication lies with the authors.

The authors would like to thank the anonymous interview participants for their time and insight, as well as Georg-Daniel Schwarz for his extensive and constructive feedback. Additionally, the detailed feedback from the anonymous reviewers was helpful to improve the scope and clarity of the manuscript.

REFERENCES

[1] Christian Bird, David Pattison, Raissa D'Souza, Vladimir Filkov, and Premkumar Devanbu. 2008. Latent social structure in open source projects. In *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering* (Atlanta, Georgia) *(SIGSOFT '08/FSE-16)*. Association for Computing Machinery, New York, NY, USA, 24–35. https://doi.org/10.1145/1453101.1453107

[2] Glenn A Bowen. 2008. Naturalistic inquiry and the saturation concept: a research note. *Qualitative research* 8, 1 (2008), 137–152.

[3] Gemma Catolino, Fabio Palomba, Damian A Tamburri, Alexander Serebrenik, and Filomena Ferrucci. 2020. Refactoring community smells in the wild: the practitioner's field manual. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering in Society* (Seoul, South Korea) *(ICSE-SEIS '20)*. Association for Computing Machinery, New York, NY, USA, 25–34. https://doi.org/10.1145/3377815.3381380

[4] Joohee Choi and Yla Tausczik. 2017. Characteristics of collaboration in the emerging practice of open data analysis. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing* (Portland Oregon USA). ACM, New York, NY, USA. https://doi.org/10.1145/2998181.2998265

[5] Laura Dabbish, Colleen Stuart, Jason Tsay, and Jim Herbsleb. 2012. Social coding in GitHub: transparency and collaboration in an open software repository. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work* (Seattle, Washington, USA) *(CSCW '12)*. Association for Computing Machinery, New York, NY, USA, 1277–1286. https://doi.org/10.1145/2145204.2145396

[6] Egon G Guba. 1981. Criteria for assessing the trustworthiness of naturalistic inquiries. *Ectj* 29, 2 (June 1981), 75. https://doi.org/10.1007/bf02766777

[7] Philip Heltweg and Dirk Riehle. 2023. Challenges to Open Collaborative Data Engineering. In *Proceedings of the 56th Hawaii International Conference on System Sciences* (Hyatt Regency Maui), Tung X Bui (Ed.). 679–688. https://doi.org/10125/102714

[8] J D Herbsleb, A Mockus, T A Finholt, and R E Grinter. 2001. An empirical study of global software development: distance and speed. In *Proceedings of the 23rd International Conference on Software Engineering. ICSE 2001.* 81–90. https://doi.org/10.1109/ICSE.2001.919083

[9] Harrie Jansen. 2010. The Logic of Qualitative Survey Research and its Position in the Field of Social Research Methods. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research* 11, 2 (2010). https://doi.org/10.17169/fqs-11.2.1450

[10] Miguel Jiménez, Mario Piattini, and Aurora Vizcaíno. 2009. Challenges and Improvements in Distributed Software Development: A Systematic Review. *Advances in engineering software* 2009 (June 2009). https://doi.org/10.1155/2009/710971

[11] Hanna Kallio, Anna-Maija Pietilä, Martin Johnson, and Mari Kangasniemi. 2016. Systematic methodological review: developing a framework for a qualitative semi-structured interview guide. *Journal of advanced nursing* 72, 12 (Dec. 2016), 2954–2965. https://doi.org/10.1111/jan.13031

[12] Barbara Kitchenham. 2004. Procedures for performing systematic reviews. *Keele, UK, Keele University* 33, 2004 (2004), 1–26.

[13] Mikkel B Kjærgaard, Omid Ardakanian, Salvatore Carlucci, Bing Dong, Steven K Firth, Nan Gao, Gesche Margarethe Huebner, Ardeshir Mahdavi, Mohammad Saiedur Rahaman, Flora D Salim, Fisayo Caleb Sangogboye, Jens Hjort Schwee, Dawid Wolosiuk, and Yimin Zhu. 2020. Current practices and infrastructure for open data based research on occupant-centric design and operation of buildings. *Building and environment* 177 (June 2020). https://doi.org/10.1016/j.buildenv.2020.106848

[14] Thomas D LaToza and André van der Hoek. 2016. Crowdsourcing in Software Engineering: Models, Motivations, and Challenges. *IEEE Software* 33, 1 (Jan. 2016), 74–80. https://doi.org/10.1109/MS.2016.12

[15] Martin Lnenicka and Jitka Komarkova. 2019. Big and open linked data analytics ecosystem: Theoretical background and essential elements. *Government information quarterly* 36, 1 (Jan. 2019), 129–144. https://doi.org/10.1016/j.giq.2018.11.004

[16] Gustavo Magalhaes, Catarina Roseira, and Sharon Strover. 2013. Open government data intermediaries: a terminology framework. In *Proceedings of the 7th International Conference on Theory and Practice of Electronic Governance* (Seoul, Republic of Korea) *(Icegov '13)*. Association for Computing Machinery, New York, NY, USA, 330–333. https://doi.org/10.1145/2591888.2591947

[17] Kevin O'Leary, Rob Gleasure, Philip O'Reilly, and Joseph Feller. 2020. Reviewing the Contributing Factors and Benefits of Distributed Collaboration. *Communications of the Association for Information Systems* 47, 1 (2020), 24. https://doi.org/10.17705/1CAIS.04722

[18] Arie Purwanto, Anneke Zuiderwijk, and Marijn Janssen. 2020. Citizen engagement with open government data. *International journal of electronic government research* 16, 3 (July 2020), 1–25. https://doi.org/10.4018/ijegr.2020070101

[19] Vijay Janapa Reddi, Greg Diamos, Pete Warden, Peter Mattson, and David Kanter. 2021. Data Engineering for Everyone. *CoRR* abs/2102.11447 (2021). arXiv:2102.11447 https://arxiv.org/abs/2102.11447

[20] Dirk Riehle, John Ellenberger, Tamir Menahem, Boris Mikhailovski, Yuri Natchetoi, Barak Naveh, and Thomas Odenwald. 2009. Open Collaboration within Corporations Using Software Forges. *IEEE Software* 26, 2 (March 2009), 52–58. https://doi.org/10.1109/ms.2009.44

[21] Erna Ruijer and Albert Meijer. 2020. Open government data as an innovation process: Lessons from a living lab experiment. *Public performance & management review* 43, 3 (May 2020), 613–635. https://doi.org/10.1080/15309576.2019.1568884

[22] Micah J Smith, Jürgen Cito, Kelvin Lu, and Kalyan Veeramachaneni. 2021. Enabling Collaborative Data Science Development with the Ballet Framework. *Proc. ACM Hum.-Comput. Interact.* 5, Cscw2 (Oct. 2021), 1–39. https://doi.org/10.1145/3479575

[23] Micah J Smith, Roy Wedge, and Kalyan Veeramachaneni. 2017. FeatureHub: Towards Collaborative Data Science. In *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. 590–600. https://doi.org/10.1109/dsaa.2017.66

[24] Sharon Spall. 1998. Peer Debriefing in Qualitative Research: Emerging Operational Models. *Qualitative inquiry: QI* 4, 2 (June 1998), 280–292. https://doi.org/10.1177/107780049800400208

[25] Damian A Tamburri, Rick Kazman, and Hamed Fahimi. 2023. On the Relationship Between Organizational Structure Patterns and Architecture in Agile Teams. *IEEE Transactions on Software Engineering* 49, 1 (Jan. 2023), 325–347. https://doi.org/10.1109/TSE.2022.3150415

[26] Ignacio G Terrizzano, Peter M Schwarz, Mary Roth, and John E Colino. 2015. Data Wrangling: The Challenging Yourney from the Wild to the Lake.. In *CIDR*. Asilomar.

[27] V A Thurmond. 2001. The point of triangulation. *Journal of nursing scholarship: an official publication of Sigma Theta Tau International Honor Society of Nursing / Sigma Theta Tau* 33, 3 (2001), 253–258. https://doi.org/10.1111/j.1547-5069.2001.00253.x

[28] Dakuo Wang, Justin D Weisz, Michael Muller, Parikshit Ram, Werner Geyer, Casey Dugan, Yla Tausczik, Horst Samulowitz, and Alexander Gray. 2019. Human-AI Collaboration in Data Science: Exploring Data Scientists' Perceptions of Automated AI. *Proc. ACM Hum.-Comput. Interact.* 3, Cscw (Nov. 2019), 1–24. https://doi.org/10.1145/3359313

[29] Jin Xu, Yongqin Gao, S Christley, and G Madey. 2005. A topological analysis of the open souce software development community. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences* (Big Island, HI, USA). IEEE, 198a–198a. https://doi.org/10.1109/hicss.2005.57

[30] Amy X Zhang, Michael Muller, and Dakuo Wang. 2020. How do Data Science Workers Collaborate? Roles, Workflows, and Tools. *Proc. ACM Hum.-Comput. Interact.* 4, Cscw1 (May 2020), 1–23. https://doi.org/10.1145/3392826

[31] Anneke Zuiderwijk, Marijn Janssen, and Chris Davis. 2014. Innovation with open data: Essential elements of open data ecosystems. *Information polity* 19, 1,2 (June 2014), 17–33. https://doi.org/10.3233/ip-140329

A INTERVIEW DOCUMENTS

A.1 Interview Handout

Challenges to Open Collaborative Data Engineering - Interview Handout

Research Context

This interview is part of a research project about collaboration during data engineering. When collaborating, multiple people work together to achieve a common goal. During data engineering, raw data is made available for further use. Examples are adding structure, fixing errors or writing documentation. Our goal is to help data engineers collaborate, ultimately providing access to higher quality data for all participants.

We are interested in interviewing people that have attempted or contributed to a data-driven project (e.g., a written report, software application, or data collection) and collaborated with others to process data for use. The interview will include questions about the people and tools you worked with, your experience with data engineering and challenges you might have encountered.

If you have a question or concern, please contact Philip Heltweg (philip.heltweg@fau.de) at the Professorship for Open-Source Software, Friedrich-Alexander-Universität Erlangen-Nürnberg.

Interview Process

These steps are part of the interview process and follow up:

1. The researcher provides this interview handout to set expectations and context for the interview
2. We decide on a date, time and software for the interview
3. During the interview
    a. We start with an informal conversation that is not recorded
    b. After clearly stating that recording starts, we start the official interview
    c. After clearly stating that recording stops, we finish the conversation
4. The researcher transcribes and pseudonymises the recording
5. The researcher shares the pseudonymised transcription with you
6. After your agreement, the data can be used in upcoming publications related to the research project

Data Use

During the interview, we collect raw audio- and videodata. From that data, we create a pseudonymized transcript, using automatic speech recognition services and manual verification. Information about the pseudonyms is stored in a separate location and never combined with the transcripts.

After your agreement, we will use the pseudonymized interview transcript for our research project. To do so, the transcript might be part of qualitative data analysis and be partially quoted or published in full as part of an academic publication. We will not share information related to the pseudonyms used or use your data outside of the context of the research project.

A.2 Interview Guide

Interview Guide

Before the interview

- Remind yourself to be mindful of the participants time, mention end of interview time
- Give a short introduction about the context of the research
    - Define data engineering, collaboration, open collaboration
        - During data engineering, raw data is made available for further use. Examples are adding structure, fixing errors or writing documentation.
        - When collaborating, multiple people work together to achieve a common goal.
        - People are openly collaborating, if outside people can find and join the project, if decisions are made based on agreement rather than dictated by hierarchy, and if people can choose their own processes and work tasks in agreement with others. Examples of open collaboration are open-source software development or content authoring in wikis.
- State the themes / structure of the upcoming interview

During the interview

- Theme: Demographic data
    - Company, job title, role
    - For past data engineering projects, typical: Data domain, role (hobbyist vs. professional), Country, open or closed data
- Theme: Collaborative data engineering
    - Do you think it is possible to make data engineering its own activity, separate from data analytics or machine learning?
    - Do you think data engineering can be split into a generic part that is useful to many projects and a project-specific part?
    - Do you think this generic part of data engineering can be done in collaboration with others?
- Theme: Collaboration systems in data engineering
    - Who have you collaborated with before during data engineering? What are their roles?
    - How and when did you interact with them?
    - What collaboration and data-engineering tools have you used before?
- Theme: Challenges to open collaborative data engineering
    - Which social or cultural challenges have you faced during collaborative data engineering?
        - Workflows (Is there a standard? Can they be improved for data engineering?)
        - Domain specific challenges?
    - Which technical challenges have you faced during collaborative data engineering?
        - Programming languages (Is there a standard? Can they be improved for data engineering, e.g. to work with domain experts?)
        - Collaboration tools like GitHub (Is there a standard? Can they be improved for data engineering, e.g. to work with domain experts?)
    - Have you encountered these previously identified challenges?
        - Need for specialized skills but high barriers to participation
        - Finding and connecting with other community members
        - No standard tools or artifacts
        - No well-understood collaboration practices
- Have we missed any question about collaboration during data engineering that we should have asked?

After the interview

- Disable Zoom recording
- Follow up with transcript