



PDF Download
3769012.pdf
18 December 2025
Total Citations: 0
Total Downloads: 146

 Latest updates: <https://dl.acm.org/doi/10.1145/3769012>

RESEARCH-ARTICLE

Addressing OSS Community Managers' Challenges in Contributor Retention

ZIXUAN FENG, Oregon State University, Corvallis, OR, United States

KATIE KIMURA, Oregon State University, Corvallis, OR, United States

BIANCA TRINKENREICH, Colorado State University, Fort Collins, CO, United States

IGOR STEINMACHER, Northern Arizona University, Flagstaff, AZ, United States

MARCO A. GEROSA, Northern Arizona University, Flagstaff, AZ, United States

ANITA SARMA, Oregon State University, Corvallis, OR, United States

Open Access Support provided by:

Oregon State University

Colorado State University

Northern Arizona University

Accepted: 09 September 2025

Revised: 25 July 2025

Received: 20 February 2025

[Citation in BibTeX format](#)

Addressing OSS Community Managers' Challenges in Contributor Retention

ZIXUAN FENG, Oregon State University, USA

KATIE KIMURA, Oregon State University, USA

BIANCA TRINKENREICH, Colorado State University, USA

IGOR STEINMACHER, Northern Arizona University, USA

MARCO GEROSA, Northern Arizona University, USA

ANITA SARMA, Oregon State University, USA

Open-source software (OSS) community managers face significant challenges in retaining contributors, as they must monitor activity and engagement while navigating complex dynamics of collaboration. Current tools designed for managing contributor retention (e.g., dashboards) fall short by providing retrospective rather than predictive insights to identify potential disengagement early. Without understanding how to anticipate and prevent disengagement, new solutions risk burdening community managers rather than supporting retention management. Following the Design Science Research paradigm, we employed a mixed-methods approach for problem identification and solution design to address contributor retention. To identify the challenges hindering retention management in OSS, we conducted semi-structured interviews, a multi-vocal literature review, and community surveys. Then through an iterative build-evaluate cycle, we developed and refined strategies for diagnosing retention risks and informing engagement efforts. We operationalized these strategies into a web-based prototype, incorporating feedback from 100+ OSS practitioners, and conducted an in situ evaluation across two OSS communities. Our study offers (1) empirical insights into the challenges of contributor retention management in OSS, (2) actionable strategies that support OSS community managers' retention efforts, and (3) a practical framework for future research in developing or validating theories about OSS sustainability.

CCS Concepts: • **Human-centered computing** → *Empirical studies in collaborative and social computing*; Empirical studies in collaborative and social computing.

Additional Key Words and Phrases: Open Source Software, Contributor Retention, Community Management, Socio-technical Systems

1 INTRODUCTION

Managing contributor retention in Open Source Software (OSS) projects is inherently challenging due to their volunteer-driven, peer-based production model, which lacks formal organizational structures and stable commitments. These challenges are further amplified by the volatile relationship companies have with OSS, as corporate priorities shift and funding fluctuates [32, 100, 151]. OSS communities need active and sustained participation to remain healthy [5]. High turnover rates in OSS hinder project progress, threaten the sustainability of the

Authors' Contact Information: Zixuan Feng, fengzi@oregonstate.edu, Oregon State University, Corvallis, Oregon, USA; Katie Kimura, Oregon State University, Corvallis, Oregon, USA, kimuraka@oregonstate.edu; Bianca Trinkenreich, Colorado State University, Fort Collins, Colorado, USA, bianca.trinkenreich@colostate.edu; Igor Steinmacher, Northern Arizona University, Flagstaff, Arizona, USA, Igor.Steinmacher@nau.edu; Marco Gerosa, Northern Arizona University, Flagstaff, Arizona, USA, Marco.Gerosa@nau.edu; Anita Sarma, Oregon State University, Corvallis, USA, Anita.Sarma@oregonstate.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2025 Copyright held by the owner/author(s).

ACM 1557-7392/2025/9-ART

<https://doi.org/10.1145/3769012>

community, and risk project failure [136, 145–147, 151]. Without practical tools to anticipate and mitigate these risks, unexpected contribution breaks may disrupt the sustainability of projects [67]. When contributors leave due to burnout, lack of recognition, shifting personal interests and priorities, or difficulty integrating within the community, projects risk losing skill sets and institutional knowledge [130].

Even though project maintainers recognize the importance of retaining contributors, they acknowledge its difficulty, as one participant from our study noted, “*It’s hard to get more contributors engaged in [the project]*” [P1]¹, and another commenting, “*We’ve had some people who have been very committed to this project, [but] definitely have burnt out of contributing to it*” [P4]. Recognizing the significance of this challenge, existing research has proposed a range of interventions. For instance, Guizani et al. [64] introduced a prototype to support managers in attracting and retaining newcomers, while Qiu et al. [99] and Ramchandran et al. [102] developed dashboards that help identify community health trends and highlight activity downturn events. Industry tools like DevStats [26] and the Amazon OSS Dashboard [7] provide static snapshots of contributors’ activities. Although these visualizations offer valuable data, they are often limited to descriptive analytics, which help projects mitigate issues after the fact and are not proactive. Another line of research draws inspiration from health-related studies [65, 79], predicting contributor retention based on historical contribution patterns [13, 21, 35, 56]. However, these models have yet to be integrated into tools for day-to-day community management. Without fully understanding community managers’ challenges, continuously introducing new interventions may not effectively support community managers. Instead, identifying these challenges should be the first step in designing meaningful support mechanisms.

Thus, while several works aim to tackle contributor retention, community managers still struggle with where, when, and how to intervene to retain their contributors [60, 64]. In this work, we close this gap by providing a deeper understanding of both the challenges managers face and the strategies that could support their retention efforts, offering insights into where interventions are most needed, when community managers should act, and how different strategies can be implemented to sustain engagement. Without such an understanding, well-intentioned research-driven interventions risk adding more complexity rather than alleviating it. Our work is guided by the following two research questions:

RQ1: What are the challenges in proactively managing retention in OSS?

RQ2: What strategies can support OSS community managers in diagnosing and managing retention challenges?

To answer these questions, we followed the Design Science Research (DSR) paradigm [36, 108], which aims to address real-world software engineering challenges by creating and evaluating artifacts that generate actionable design knowledge. We conceptualized the problem of contributor retention in OSS as a recurring and socio-technical issue that lacks operationalizable, community-centered solutions. We first identified 10 core challenges in contributor retention management through a triangulated empirical analysis involving interviews with OSS community managers (🗣️), multi-vocal literature review (📖), and expert validation surveys (📝).

In response to these empirically grounded challenges, we then developed actionable strategies that aim to empower community managers with practical insights for supporting retention management. We compiled a list of nine strategies based on insights from the literature review (📖). These strategies were operationalized through a prototype using an iterative build–evaluate cycle and a user-centric design process co-developed with input from over 100 OSS practitioners (👥). The prototype served both as an intervention and a validation scaffold to ensure the strategies’ relevance and usability in practice.

¹Interview participant ID.

To validate our intervention and move toward technological rule generalization, we evaluated our prototype in two distinct OSS communities—Pyomo [126] and DeepSpeed [89] (👤)—each with different governance structures and community cultures. We loaded their project data into the prototype and invited their governance teams to evaluate the prototype. This approach ensured our strategies were both theoretically grounded and practically applicable to OSS projects.

Through this study, we aligned research advancements with the pragmatic needs of practitioners by exploring the realities of retention management. We also proposed integrated strategies to help OSS communities engage and retain contributors proactively. Our study provides (1) empirical insights into the challenges OSS community managers face in sustaining contributor engagement, along with a structured analysis of the strategies that support retention efforts; (2) a human-factors-driven framework that serves as a replicable blueprint for future researchers seeking to develop or validate theories about OSS sustainability and software engineering processes; and (3) practical guidance for OSS practitioners, offering diagnostic strategies to surface disengagement risks and inform retention-related decisions.

2 BACKGROUND AND RELATED WORK

Effective retention management is vital for OSS project sustainability, as it preserves critical expertise, prevents contributor attrition, and fosters long-term engagement and project growth [67, 100, 149, 151]. However, managing retention in OSS is challenging due to the many factors that influence why contributors stay or leave. This section reviews existing literature on how contributor retention is managed in OSS.

2.1 The Complexities of Managing Contributor Retention

Research on OSS contributor retention has examined both individual and project-level indicators. Individual metrics include commit frequency, comments, and pull request activity [13, 35, 100], while project factors encompass popularity and collaboration environment characteristics [13, 35, 67, 143, 150, 151].

Existing studies indicate that contributors' demographic attributes can influence their likelihood of disengaging, particularly when certain groups experience exclusion within project communities. For instance, Trinkenreich et al. [132] found that women contributors are underrepresented in the OSS community and face a heightened risk of disengagement. Zhang et al. [147] found that affiliated developers contribute more than volunteers in the OpenStack Foundation, but this can lead to a Pareto-like phenomenon affecting an OSS project's sustainability if dominant companies withdraw. Beyond gender, other demographic factors—such as geographic region and English proficiency—may also affect retention [40, 43]. While these insights highlight critical risks to retention, translating them into actionable strategies remains a persistent challenge for practitioners.

Moreover, contributor retention is shaped by a wide range of factors, requiring community managers to actively track and interpret various signals. However, they then face challenges in leveraging retention data due to its fragmented and complex nature [60, 64]. For example, one of the most popular OSS project health initiatives—the CHAOSS project [28]—defines over 50 metrics, organized into 10 categories such as Organization, Software, Contributions, Contributors, and Governance and Leadership, to comprehensively assess project health. In most cases, community managers lack an understanding of where to focus their efforts, what to prioritize, and how to act upon these metrics to manage retention challenges. How these complexities can collectively affect community managers' day-to-day efforts remains unclear. Today, little is known about the hurdles community managers encounter when interpreting and acting upon these different data sources, leaving a gap in our understanding of how to support them effectively.

2.2 Limitations of Existing Interventions

Previous interventions for managing contributor retention span academic and industry solutions. Academic approaches include prototypes for engagement that indirectly support retention by helping with newcomer attraction and recognition [64], community health monitoring dashboards that visualize project activities (e.g., pull requests and issue tracking) [99], and activity tracking systems for incubator projects [102]. Industry tools like DevStats [26] and the Amazon OSS Dashboard [7] offer similar monitoring capabilities. However, these solutions primarily provide retrospective insights, potentially delaying critical interventions. This reactive approach limits visibility into where proactive support is most needed for community managers.

In addition to analysis of historical data, OSS researchers have drawn inspiration from medicine and human resources disciplines, adopting prediction models such as the Cox regression to predict contributors at higher risk of disengagement based on their contribution history [13, 35, 56]. Similarly, Kaplan-Meier analysis has been used to evaluate the survival rate of demographic groups within OSS communities [40, 100]. While prediction models like Cox regression and Kaplan-Meier analysis have shown statistical promise, they remain isolated in research contexts and have not been integrated into tools or assessed for their effectiveness in helping community managers' decision making.

2.3 Our contributions

To bridge these gaps, our study employs a mixed-methods approach guided by the DSR paradigm to create a comprehensive understanding of the complexities of managing contributor retention in OSS. We identified community managers' challenges by triangulating findings through interviews, multi-vocal literature reviews, and surveys. Building on these insights, we curated a set of strategies designed to help OSS community managers identify and address disengagement risks, making informed decisions. These strategies were embedded in a web-based prototype, refined through a user-centric design process involving over 100 OSS practitioners. Finally, we conducted in situ evaluations with two OSS communities to assess the practicality of these strategies in supporting retention management in OSS.

3 METHOD OVERVIEW

This section provides an overview of our methodology (Figure 1), outlining how we answered each research question by following the DSR paradigm [36, 108] through a mixed-methods [34, 125], two-phase study that addresses the problem of contributor retention in OSS.

Phase 1 - Problem definition: We conceptualized the problem to identify the challenges in managing retention in OSS by triangulating insights from semi-structured interviews, a multi-vocal literature review [53], and an expert survey.

Phase 2 - Solution design and validation: We designed a solution by first synthesizing strategy candidates from prior work and community input, then applying participatory, user-centric design methods to iteratively refine a prototype through multiple focus group sessions with over 100 OSS contributors at major community events. We validated the strategies in context by conducting an in situ user evaluation across two OSS communities (Pyomo and DeepSpeed).

These two phases were guided by our two research questions as follows:

(RQ1) For problem identification, we focused on empirically surfacing and understanding the challenges related to retention management in OSS. To ensure methodological rigor, we triangulated multiple sources of evidence: We initiated our investigation by conducting *semi-structured interviews* (🗣️) with OSS community managers. These initial findings were then triangulated [48] through (1) a *multi-vocal literature review* (📖) (academic and industry/gray literature) [53] and (2) an *expert validation survey* (📝). The methodology details are explained in Section 4.

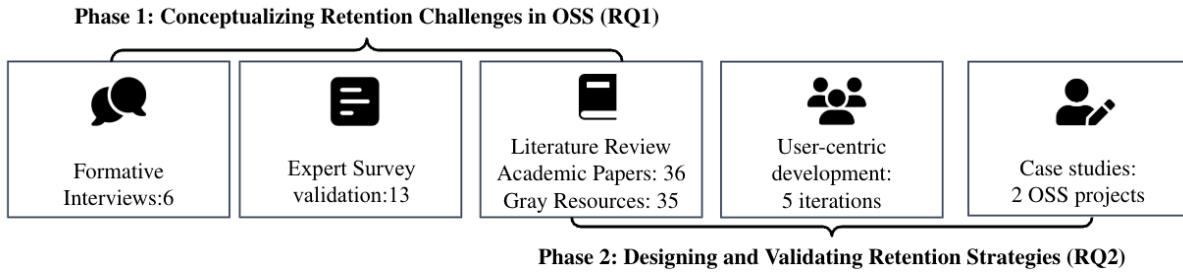


Fig. 1. Method Overview

(RQ 2) For solution design and validation, we continued our mixed-method approach to identify, develop, and refine strategies that support OSS community managers in diagnosing contributor engagement patterns and responding to retention challenges. First, we revisited the *literature* (📖) used for RQ1 to compile a list of potential strategies. Then, we developed a web-based prototype that operationalized these strategies. Prototyping enabled us to move beyond theoretical discussions by giving OSS community managers an interface to interact with and assess the strategies within the context of their project. This approach enabled us to gather richer, more actionable feedback, revealing how these strategies can be integrated into existing workflows and identify potential limitations [17]. See the list of strategies in Section 5.

To ensure a community-driven prototype design, and in line with DSR's emphasis on iterative development, we followed a *user-centric design* (👤) approach [46] (see Figure 2), holding multiple rounds of focus group discussions with over 100 OSS contributors throughout the design process, detailed below:

1. INITIAL FEEDBACK FROM FOCUS GROUP DISCUSSION I. We adhered to design principles and guidelines from Few [47] to create an early prototype version. We then gathered initial feedback from community managers at an industry event (FOSSY 2023), using their insights to refine our design.
2. INITIAL IMPLEMENTATION AND CONSULTATIONS WITH KUBERNETES AND FLUTTER. Based on input from FOSSY 2023 participants, we consulted with OSS managers from two large-scale projects, Kubernetes and Flutter, to enhance and tailor the prototype's features.
3. ITERATIVE REFINEMENT FROM FOCUS GROUP DISCUSSION II. We organized a community session at an industry event (Linux Plumbers 2023), engaging with community managers and maintainers. Feedback from this session helped us iterate on the prototype to better align with community needs.
4. FINAL FEEDBACK FROM FOCUS GROUP DISCUSSION III. In a community session at another event (OPENUK 2023), we received no further design feedback, indicating that our design was sufficiently robust to meet community expectations. The details of how we operationalize strategies in the prototype are discussed in Section 6.

To validate the strategies in context, as emphasized in the DSR paradigm, we conducted an *in situ user evaluation* (👤) by introducing the prototype to two OSS communities with different governance structures and cultures (Pyomo [126] and DeepSpeed [89]). We populated the prototype with their project data and invited their governance teams to evaluate whether the prototype influenced their decision-making around contributor retention. This step provided us with real-world feedback on the usefulness of the provided strategies and insights. The details of study design are explained in Section 7.

This study was approved by the Oregon State University Institutional Review Board (IRB) under the project titled *Sustainable OSS: A Framework for Proactive Identification and Reduction of Contributor Disengagement*. All research activities involving human participants were carried out in accordance with IRB requirements, including informed consent procedures.

4 CHALLENGES IN MANAGING CONTRIBUTORS’ RETENTION IN OSS

4.1 Research Methodology

Following the DSR paradigm, we began by conceptualizing the problem space. To understand how to help community managers with contributor retention, we first investigated the challenges OSS community managers face. We employed a multiple-triangulation validation strategy [48], as shown in Figure 1, starting with interviews (🗣️) to directly understand challenges from community managers. We then triangulated our findings through a multi-vocal literature review (📖) [53], which included gray literature and academic research, and an expert validation survey with OSS project managers (N=13) with diverse community management experiences (👤).

4.1.1 Semi-structured Interviews 🗣️:

Interview Recruitment: Guided by principles from qualitative methodology [20, 61], we recruited participants with diverse OSS experience levels, community management tenure, and project sizes—including managers from small, medium, and large OSS communities. Table 1 summarizes the participants’ demographics. We first recruited two community managers from the authors’ contacts: one from Cloud Native Computing Foundation (CNCF) and the other from Community Health Analytics in Open Source Software (CHAOSS). We then employed a snowball sampling method by asking participants to help us recruit additional participants. Recruitment ended after six interviews, following the principle of thematic saturation [112], as no new categories of insight emerged. Starting with the fifth participant, interviews ceased to generate additional perspectives on the challenges of managing contributor retention, and the sixth interview served as a validation check.

Table 1. Demographic information of interview participants.

ID	Gender	OSS Experience	Management Experience	Current Project Size*
P1	Woman	More than 10 years	Between 3 and 5 years	Medium
P2	Woman	More than 10 years	Between 5 and 10 years	Medium
P3	Man	Between 3 and 5 years	Less than 1 year	Large
P4	Prefer not to say	Between 3 and 5 years	Between 1 and 3 years	Small
P5	Man	More than 10 years	More than 10 years	Small
P6	Man	More than 10 years	Between 5 and 10 years	Large

Project Size: Small projects (1-25 contributors), Medium projects (26 - 100 contributors), Large projects (>100 contributors)

Among the six interviews, four were conducted on-site, and two were conducted remotely through a video conference platform². The interviews lasted from 45 minutes to 1 hour, were recorded with the participants’ consent (following the university-approved IRB protocol), and were then transcribed. Participants were offered a \$50 gift card as a token of appreciation for their time.

Interviews Design: Each session began with a brief introduction to the study. After confirming the participants’ consent, we asked them to reflect on the composition and roles within their community (e.g., “How many newcomers are there in your project?”). We used this icebreaker question to help participants recall their community’s composition and contextualize the interview regarding their management experiences.

We then asked about each participant’s role and responsibilities in the project. Next, we invited them to share their thoughts on contributor retention and any observations they had about contributors’ disengagement. We followed up to determine whether they actively monitored or assessed retention using specific tools or metrics, taking into account their challenges. Throughout the interview, open-ended follow-up questions were actively used to encourage deeper insights into how they manage contributor retention.

Pilot Interviews: Before conducting interviews, we held three rounds of pilot interviews with graduate research assistants who were knowledgeable about OSS communities and had relevant experience in OSS

²Zoom

research. Feedback from these pilot sessions helped us refine the interview questions. For example, participants suggested adding ice-breaking questions to contextualize the interview and follow-up prompts for situations in which respondents provided short answers about their experiences in managing contributor retention. They also recommended asking participants to share specific examples of times they observed or anticipated a contributor leaving the project. (Refer to the supplementary material for detailed interview scripts [41]).

Interview Data Analysis: We did not separate the interviews and their analysis. Instead, after each interview, we performed an inductive thematic analysis to identify emerging themes [19, 57]. We made multiple passes over the transcribed data, avoiding presupposed codes and allowing codes to emerge naturally from the content. Throughout the analysis, three authors discussed new codes, developed a preliminary codebook, and continuously evaluated the rationale for assigning specific codes. We conducted the entire procedure through continuous comparison during coding sessions and used negotiated agreements [49, 54].

Our analysis resulted in the identification of codes related to (1) the challenges of monitoring contributor retention in OSS, (2) methods for monitoring contributor retention, and (3) experiences of managing retention in OSS. The three researchers agreed on 10 challenges in monitoring contributor retention (see Table 3).

4.1.2 Triangulation I: Literature Review 📖. We conducted a multi-vocal literature review to triangulate the challenges in managing contributor retention that we learned from the interviews and to investigate whether we overlooked any additional challenges.

Scope of Literature Review: Our literature review was designed to cover both academic and gray literature (e.g., OSS blogs), as studies have mentioned that disconnection and miscommunication between research and industry remain prevalent in software engineering communities [105]. OSS practitioners often share their experiences on blog platforms, which academic studies might overlook. By including both literature resources, we aimed to provide a more comprehensive understanding of the challenges in managing retention in OSS.

Keywords: We determined a list of search keywords to ensure the relevance of the captured studies about retention in OSS. We used three studies on retention management in OSS that we were already familiar with to decide on keywords [100, 110, 153]. Proposed search keywords included, “retention,” “disengagement,” “turnover,” “inactive,” “abandonment,” “survive,” and “survival”. We added “Open Source” or “OSS” to each string to focus on the OSS community.

Academic literature review: We required the search keywords to appear at least once in the title, abstract, or keywords. This approach is consistent with existing systematic literature reviews in OSS research [44]. The digital libraries (IEEE Xplore, ACM Digital Library) we used have been frequently utilized in other systematic literature reviews in software engineering [8, 44, 132]. Additionally, following the recommendations of Kitchenham and Brereton [72] for secondary studies in computer science and software engineering, we employed backward snowballing to identify any relevant studies we might have missed.

Pilot search queries in academic digital libraries: Before applying the search string across the digital libraries, we conducted a pilot search to validate the query. We used three control studies [100, 110, 153] to test the search query, verifying that these studies were retrieved.

Academic literature list: Our initial search after removing duplicates resulted in 256 publications. Then, the first and second authors read the titles and abstracts and only selected papers focused on retention in OSS; we only kept papers with more than five pages; those with fewer than five pages are usually not considered full research papers in top-tier venues (e.g., [2, 3, 44]). When we finished filtering out papers, 24 remained.


To collect additional studies, we performed a single iteration of backward snowballing [72, 141], looking for further studies published in journals and other conferences. This search resulted in us adding 11 papers to our literature list, resulting in a final total of 35 academic papers.

To assess the completeness of our list, we performed a completeness check [44] on the three most recent studies from our list: [21, 99, 131]. This step ensured we did not miss any relevant studies, as we only performed

a single iteration of backward snowballing, and keyword search results may have missed some related studies. The first two authors independently reviewed every reference in these three papers, listing those relevant to contributor retention. The two authors then reviewed each other’s lists and reached a negotiated agreement. In the end, we found 35 academic studies, with only one missing from our list [101]. Thus, our completeness ratio is 99%, with one additional paper being added to our original list, resulting in a final total of 36 academic papers (See the Supplementary Materials for the list of academic literature [41]).

Gray literature review: We searched the top 50 OSS blog platforms from the list of “Best Open Source Blogs and Websites” on Feedly.com [39]. These blog platforms included Open Source Initiative [94], Google Open Source Blog [59], Linux.com [87], the Software Freedom Conservancy blog [117], Fedora People [38], and other foundation-based platforms. We removed 18 platforms from the list as they were either project-specific, solely released project updates, or personal web pages (See the Supplementary Materials for the list of gray literature platforms [41]). Unlike larger OSS blog platforms, which typically undergo some review and reflect common or widely accepted practices within the community, personal web pages can present unverified opinions, making them less reliable. We used the exact search string that we used to find academic literature, which resulted in 35 blog articles (See Gray Literature References [G1] - [G35]).

The first two authors independently reviewed papers and blog content to find any challenges related to monitoring contributors’ retention. We held weekly meetings to present our findings and discuss the identified challenges until we reached an agreement. The challenges agreed upon in this open coding iteration were added to the interview results.

4.1.3 Triangulation II: Expert Validation survey . We then conducted an expert evaluation survey to triangulate the challenges of managing retention.

Survey design: Our survey contained 14 questions, including closed- and open-ended questions. The survey started with three demographic questions regarding gender, OSS experience, and experience as OSS managers. The following 10 questions aimed to understand the participants’ agreement with the challenges we identified in managing contributors’ retention. The final question was open-ended, asking participants to list any challenges not listed in the survey.

Pilot survey: We conducted five pilot studies with graduate students (n=3) and OSS contributors (n=2). After each pilot study, we collected feedback, which we used to refine the survey. For example, pilot study participants suggested adding explanations and short examples for each challenge. See the supplementary materials for the survey questions [41].

Participant recruitment: Once we finalized the survey, we began the recruitment process by contacting five community managers new to the study (i.e., didn’t participate in the earlier interviews) and had more than five years of experience managing OSS communities. These community managers helped us recruit 15 participants. After removing seven incomplete answers, we had 13 valid responses. The demographics of the survey participants are shown in Table 2. Nine out of 13 participants have experience managing large projects (more than 100 contributors), and five mentioned having managed small to large communities. The diverse OSS and management experiences provided a foundation for effectively validating the identified challenges.

Survey analysis: We quantitatively analyzed the survey’s closed-ended questions to understand participants’ level of agreement on the identified challenges in contributor retention monitoring. We did not receive any responses from the open-ended questions regarding new challenges.

In summary, our analysis of the interviews revealed 10 challenges. During the literature review, we found no new challenges; 10 out of 10 were validated from research studies, and nine (out of 10) challenges were validated from blog posts, as shown in Table 3. The survey responses helped us triangulate the 10 challenges, as shown in the last column of Table 3. Dark green indicates “strongly agree”, and light green indicates “agree”, with each challenge receiving support from more than half of the participants.

Table 2. Expert survey participant demographics

ID	Gender	OSS experience	Managing experience
SP1	Man	More than 10 years	SML
SP2	Woman	3 years but less than 5 years	S
SP3	Woman	3 years but less than 5 years	SM
SP4	Man	More than 10 years	L
SP5	Man	More than 10 years	SML
SP6	Man	5 years but less than 10 years	SML
SP7	Man	More than 10 years	SML
SP8	Man	More than 10 years	S
SP9	Man	More than 10 years	L
SP10	Woman	More than 10 years	SML
SP11	Man	1 year but less than 3 years	M
SP12	Prefer not to say	5 years but less than 10 years	L
SP13	Man	5 years but less than 10 years	L
Managing experience: Small projects (1-25 contributors), Medium projects (26 - 100 contributors), Large projects (>100 contributors)			

4.2 Results

Here, we unpack the challenges of managing contributor retention, which we group into four categories: (1) workload and time constraints, (2) fragmented contributor and contribution tracking, (3) contributor disengagement, and (4) data privacy. Table 3 presents these challenges. For each challenge, the table includes its definition, an example, and supporting evidence from our triangulation validation.

4.2.1 Workload and Time constraints. C1. Lack of a Retention Management Mechanism: Participants described the absence of a mechanism or infrastructure to support contributor retention efforts. Instead of relying on streamlined or automated tools to infer disengagement, community managers have to resort to manual, piecemeal methods: scanning spreadsheets [P2], reviewing mailing lists [P1, P6], or relying on personal familiarity with contributors [P2, P4]. As one participant noted, *"I derive a lot of the data manually"* [P3]. Relying on manual processes is impractical for large-scale communities with thousands of contributors [12, 82, 151]. As OSS communities grow, managers struggle to track and combine diverse data sources and metrics to gauge engagement and retention accurately [4, 119, 123].

C2. Overwhelming Responsibilities for OSS Community managers have been widely reported in OSS communities [G2, G3, G4, G5, G6, G7, G8, G9, G10, G11, G12, G13, G14, G31]. One Blog noted, *"Lack of time was cited as the leading reason 'not to contribute' and a motivation to leave a community"* [G6]. Similarly, one interview participant mentioned they were *"already probably putting in somewhere between a 50-70 hour work week"* [P3], while another noted that *"doing the interpersonal work communities need to grow is overwhelming"* [P2].

C3. Challenges of Managing Turnover: Turnover within OSS communities can undermine sustainability by causing knowledge loss and increasing onboarding costs for newcomers [69, 80, 82, 120, 129]. *"87% of hiring managers experience difficulties recruiting enough open source talent, similar to last year when 89% reported challenges in finding the right mix of experience and skills"* [G18]. Our interview participants similarly reported that managing turnover is challenging *"because even if someone isn't a 'maintainer', if they've been responsible for most of the code changes in our database backend driver, losing them abruptly and unexpectedly is still going to be a problem for the project"* [P3]. Consequently, it can be difficult to identify candidates with comparable expertise to replace contributors who leave the project. Such incidents may discourage community managers, which can in turn force them to leave the community [14, 145].

C4. Time-consuming to Engage and Grow the Contributor Base: P2 described the personal effort involved in keeping contributors engaged: *"Usually, if someone has been active and then goes inactive, I will reach out and say, 'Hey, what's up? No pressure, just wanted to reach out and see how you're doing.' And then, at that point, people will respond nine times out of 10, and someone will explain"* [P2]. However, such individualized interaction is not

Table 3. Challenges of managing contributor retention.

ID	Challenges in managing retention	Example (It is challenging to...)	Participant ID	Academic literature	Gray Literature	Survey (N=13)*
Workload and time Constraints						
C1	Lack of a Retention Management Mechanism	manage retention due to lack of tools for diagnosis and insight.	P1, P2, P3, P4, P5, P6	[4, 82, 122] [12, 80, 123] [82, 151]	[G1]	
C2	Overwhelming Responsibilities	allocate time effectively to balance responsibilities while monitoring retention.	P2, P3, P4, P5, P6	[80, 120, 129] [14, 69, 82] [12, 92, 110] [13, 82, 145]	[G2, G3, G4] [G5, G6, G7] [G8, G9, G10] [G11, G12, G13] [G14, G31, G33]	
C3	Challenges of Managing Turnover	identify candidates with equivalent skills to mitigate the impact of turnover.	P2, P3, P4, P5, P6	[122]	[G15, G16, G17] [G18, G19, G20] [G26, G30, G32]	
C4	Time-consuming to Engage and Grow the Contributor Base	dedicate sufficient time to interact with and support contributors effectively.	P1, P2, P3, P5, P6	[122]	[G35]	
Fragmented Contributor and Contribution Tracking						
C5	Challenges in Tracking Project Data	gain a clear view of overall retention trends within the project.	P3, P6	[13, 80, 144] [101]	[G22, G23, G28]	
C6	Challenges in Tracking Contributor Data	aggregate and analyze contributor activities and engagement to identify at-risk members.	P2, P3, P6	[13, 80, 144] [101]	[G1, G23, G27] [G29]	
C7	Challenges in Acknowledging Hard-to-Track Contributions	identify and recognize untracked contributions, such as marketing efforts.	P1, P2, P3, P4	[13, 101]	[G11, G24, G25]	
Managing Diversity and Contributor Retention Risks						
C8	Challenges in Fostering Inclusive Collaboration	evaluate the demographics of contributors at risk of disengagement.	P1, P2, P3, P4, P5, P6	[100, 129, 136]	[G10, G17, G21] [G24, G26, G27] [G32, G33, G34]	
C9	Challenges in Anticipating Contributor Attrition	track contributor activity to identify early signs of disengagement.	P2, P3, P6	[78, 82, 145] [100, 106, 153] [12, 13, 110] [14, 111, 144] [35]	[G15, G16, G30] [G32, G17, G18] [G19, G20]	
Data Privacy						
C10	Challenges in Ensuring Data Privacy While Tracking	ensure the confidentiality of contributor data during tracking processes	P1, P3	[6, 100, 152]	-	

The dark green section represents the percentage of participants who “Strongly Agree,” while the light green section shows those who “Agree.” The gray section corresponds to “Neutral,” the light red corresponds to “Disagree,” and the red section to “Strongly Disagree.”

feasible in day-to-day practice, especially for large communities; actively engaging with and cultivating a vibrant contributor base is time-consuming [122]. “*Make sure you’re prepared to spend the time doing community building and interacting with people to make it truly shine. We struggled with that early on, and it really hit us hard*” [P5].

4.2.2 Fragmented Retention Data. C5. Challenges in Tracking Project Data: Project retention indicators take many forms, such as the number of active contributors, the number of contributors who have left, the number of newcomers joining, and are often scattered across multiple channels, including pull requests and issues. Because of this, community managers find it challenging to assess the overall retention health of their projects “*Sometimes I need to see the overall trend across a project or several projects at once if they’re related to each other.*” [P6]. This process demands substantial effort because it involves collecting fragmented project data and aggregating them [13, 80, 101, 144].

C6. Challenges in Aggregating Contributor Data: Delving into details about individual contributors, such as understanding their trajectory, identifying where they are stalling, and recognizing the types of work they excel at, is difficult, but important to find and support at-risk contributors. No existing tool provides such insight *“There are many other times when I want to see the overall trend for a particular person”* [P6]. Such inability to unify contributor-specific data impedes efforts to diagnose disengagement and proactively support at-risk members [G1, G23].

C7. Challenges in Acknowledging Hard-to-Track Contributions: Appropriately recognizing various forms of contribution in OSS is difficult, mainly when contribution tracking relies on repository visible metrics (e.g., commits, pull requests) [13, 101]. The concept of “glue work”, first introduced by Tanya Reilly [104], refers to the essential yet frequently overlooked contributions in software development that hold teams and projects together. This issue is especially prevalent in OSS communities, where the dynamic nature of volunteer-driven collaboration can obscure the importance of these crucial efforts (e.g., PR reviews or mentoring). These types of glue work often remain in the shadow of feature code development [G11, G24, G25].

4.2.3 *Contributor disengagement.* **C8. Challenges in Fostering Inclusive Collaboration:** While fostering a welcoming environment for contributors with diverse backgrounds is beneficial [50, 100, 129, 136], it places additional demands on community managers who are often already overwhelmed. Studies show that contributors may face barriers rooted in their different backgrounds, such as their English proficiency, cultural differences, and gender [100, 129, 136]. These concerns have drawn increasing attention across OSS communities [G2, G8, G15, G21, G24, G32]. *“I don’t know the demographics. It is highly international; there are people from all over”* [P6].

C9. Challenges in Anticipating Contributor Attrition: Anticipating contributor attrition involves more than observing overarching trends; it requires analyzing each contributor’s history of participation [12, 13, 78, 82, 100, 106, 145, 153]. Prior research suggests that examining a contributor’s historical activities can offer early indicators of potential departure [110, 153]. However, systematically monitoring this information is a challenging task for practitioners. *“You can analyze the mailing list and compare it with the commit logs and the bug tracker to see who has been active for a long time, but I’ve observed many of the same names remaining active over the years”* [P6]. In some large projects, managers only realize a contributor has left after noticing the project encountered a failure [80].

4.2.4 *Data Privacy.* **C10. Challenges in Ensuring Data Privacy While Tracking:** When community managers dedicate considerable effort to aggregating contributor data for retention tracking, data privacy becomes critical. Understanding contributor backgrounds is essential for fostering inclusivity and improving retention strategies. However, this process introduces privacy challenges, especially when data inaccuracies exacerbate existing issues [82, 100]. *“But the problem you encounter is that it delves into personal data about people, right? So, it becomes really hard when you start measuring this and considering who gets access to it and how you ensure people’s private information remains private. Therefore, in some cases, we don’t measure it because we’re too concerned about doing it”* [P1].

4.2.5 *Triangulation.* To validate the interview results, we carried out two forms of data triangulation: (1) a literature review of academic and gray literature and (2) a survey.

Triangulation with Literature Review: All 10 identified challenges were also discussed in academic literature, with the most frequently discussed challenges being C1 (lack of retention mechanism), C2 (overwhelming responsibilities), and C9 (anticipating contributor attrition). The gray literature supported 9 out of the 10 challenges; C10 (ensuring data privacy while tracking) was not supported in our gray literature review. The top-3 reported challenges in the gray literature were C2, C3, and C9. This shows that both academic and gray literature are well aware of the problem of maintainer burnout because of being overwhelmed with responsibilities (C2) as well as challenges in being able to predict contributor attrition (C9).

Triangulation with Survey Responses: For all 10 challenges, more than 50% of survey respondents agreed (strong agreement/agreement shown by dark/light green bars) that these challenges exist in their projects. The top three highest rates of agreement were for challenges C3 (managing turnover), C4 (engaging and growing the contributor base), and C8 (fostering inclusive collaboration). The list of the top three is different from what we found in the academic literature, but this is likely because the challenges are project-dependent.

5 STRATEGIES IN MANAGING CONTRIBUTORS RETENTION

This section introduces strategies to support managing contributor retention, derived from a review of 36 studies on OSS retention. These strategies highlight practices and approaches that community managers can adopt to improve contributor engagement. As part of our DSR process, we continued to follow a triangulated, iterative build–evaluate cycle [36, 108] to validate the practicality and helpfulness of the synthesized strategies. Figure 1 provides an overview of our process, which comprises: (1) a literature review (📖: this section), (2) user-centric design approach (👥: Section 6), and (3) case studies with future users in their OSS communities (👤: Section 7). In this section, we discuss how we identified strategies from the literature review.

5.1 Method-literature review (📖)

To identify potential strategies for the challenges we have identified, we revisited the set of 36 studies related to OSS retention from Section 4. We reused the search results from Section 4 since the search keywords used to identify these papers were not limited to only challenges but included all aspects of retention in OSS. We did not include gray literature at this stage because we plan to directly engage with OSS community managers to gather their experiences and opinions, thereby gaining broader, real-world insights into how these strategies can be practically scaled.

Each of these studies had focused on one or set of challenges in retention management and proposed recommendations for practitioners. For instance, some studies emphasize the importance of closely monitoring and engaging newcomers to enhance retention rates [12, 35, 35, 55, 78, 100, 110, 120, 122, 124, 124, 151–153]. Other studies provide information to community managers that aid in retention management [100, 122, 123, 134, 151]. Additionally, several studies have explored using statistical models to predict the likelihood of contributors leaving a project, offering community managers the necessary insights to take proactive measures [4, 12, 35, 82, 82, 88, 106, 134, 145, 153].

We conducted another round of open coding protocol on these 36 academic studies in our primary list to focus on potential strategies for managing retention. During the analysis, each emerging code was compared to the existing codes to determine whether it represented a separate category or was a subset of an existing code. The first three authors applied this procedure via continuous comparison throughout the coding sessions [49, 54]. Any disagreements during the sessions were resolved through a negotiated agreement to maintain reliability [54]. In total, we synthesized nine strategies (Table 4).

5.2 Result

To address the challenges identified in Section 4, we synthesized nine strategies that OSS community managers can use to diagnose, track, and respond to retention-related issues. Table 4 summarizes these strategies and maps them to the corresponding challenges. These strategies are grouped into two categories: (1) Community Engagement and Retention Analytics and (2) Project and Contributor Management. For each strategy, the table presents a description, its function (e.g., diagnose, reduce workload, and compliance). For example, S1 (Track Project Engagement) monitors contributor activity and retention trends to diagnose potential emerging disengagement.

5.2.1 Community Engagement and Retention Analytics. Strategies in this category focus on creating statistical analysis to help community managers take proactive action.

Table 4. Strategies for Mitigating Challenges in OSS Retention Management

ID	Strategy	Description	Function	Challenges Addressed	References
Community Engagement and Retention Analytics					
S1	Track Project Engagement	Tracks contributor engagement by analyzing activities, retention trends.	Diagnose Disengagement	C1, C2, C5, C6, C7	[82, 114, 145] [4, 13, 100] [35, 109, 151]
S2	Track Newcomers	Tracks new contributors, analyzing their participation patterns and engagement levels.	Diagnose Attrition	C1, C2, C4, C6	[78, 100, 120] [12, 124, 153] [35, 110, 151] [123, 152]
S3	Track Individuals Contributions	Tracks individual contributor activities to evaluate their level of engagement, contribution patterns, and impact within the project.	Diagnose Disengagement	C1, C2, C5, C6, C7	[35, 124, 151] [111, 145, 152] [100, 109, 110] [22, 80]
S4	Predict Contributor Attrition	Identifies contributors who are at risk of disengagement	Diagnose Attrition	C1, C2, C3, C5, C6, C9	[82, 106, 145] [4, 82, 153] [35, 88, 91] [97, 134, 144]
S5	Understand the Impact of Attrition	Analyzes the impact of contributor turnover on different project areas	Diagnose Project Health	C1, C2, C3	[16, 33, 144] [82, 144, 145] [148]
S6	Promote a Welcoming Environment	Understand contributors' backgrounds to foster a welcoming environment.	Diagnose Disengagement	C8	[82, 100, 129] [68, 134, 136] [40, 107] [93, 118]
Project and Contributor Management					
S7	Ensure Privacy Practices	Implements data privacy to protect aggregated contributor information and sensitive data.	Compliance	C10	[82, 100, 127] [9, 142]
S8	Automated Notifications	Uses automated notifications to provide updates on the project health.	Reduce Workload	C2	[69, 120, 122] [84]
S9	Utilize Tools for Contributor Engagement	Employs engagement tools to enhance contributor interactions	Reduce Workload	C2, C4	[22, 120, 122] [69, 123]

S1. Track Project Engagement: Understanding engagement is essential for community managers to manage retention, as it serves a diagnostic function by surfacing early signs of disengagement [82, 114, 145]. For example, Schilling [109] mentioned the importance of monitoring turnover rates, which helps assess whether the project's retention rates are healthy. Zhang et al. [145] recommended tracking the number of newcomers and members who left, while Lin et al. [82] emphasized measuring the average tenure days to understand retention. Similarly, Sharma et al. [114] and Qiu et al. [100] analyzed retention using contribution activities. Together, these metrics can provide a comprehensive mechanism to track project engagement and support community managers [4, 13, 35, 109]. However, extracting this data can be time-consuming, "...I mean a lot of it depends on me as the community manager keeping track of people which is hard..." [P2]. Additionally, different metrics need to be actively monitored, many of which are fragmented across multiple locations (e.g., issues, pull requests). Using this strategy therefore requires an automated monitoring system.

S2. Track Newcomers: A key way to manage retention is to onboard newcomers and track their progress to diagnose early signs of contributor attrition [35, 78, 110, 120, 151–153]. For example, Steinmacher et al. [120] found that mentors help newcomers integrate into the project's culture, which requires them to monitor newcomers' contributions and identify areas where support may be needed [12, 124]. Similarly, Feng et al. [44] emphasized that effective onboarding strategies rely on understanding newcomers' specific challenges. Lee [78] mentioned that this process depends on how well the organization can track not only who the newcomers are but also their activities. This is a challenge because of the fragmented nature of OSS tools and data sources (e.g., relevant data

can be located across issue trackers, code repositories, and discussion boards), which makes it difficult to piece together a cohesive picture of newcomers' activities and progress [44, 45].

S3. Track Individual Contributions: Tracking each contributor's activity, not only newcomers, serves as a diagnostic function by providing insights into individual engagement and productivity within the community [22, 35, 80, 110, 111, 145, 151, 152]. For example, Zhou and Mockus [151] and Qiu et al. [100] highlighted that individual performance and engagement histories significantly influence retention. OSS projects can build stronger relational ties by paying close attention to the activities and social dynamics of individual developers [22, 80, 81, 110, 152]. However, current platforms like GitHub and other OSS collaboration tools often disaggregate this information, making it challenging for managers to track contributors effectively. Most existing tools, such as DevStats [1], Amazon Dashboard [7], and Climate Coach [99], rarely provide aggregated data on individual contributors.

S4. Predict Contributor Attrition: Tracking contributors' attrition is important to help with retention [4, 35, 82, 82, 88, 106, 134, 145, 153]. For example, Phillips and Phillips [97] mentioned that tracking attrition of high-impact members who choose to leave can help identify areas where attraction/retention needs to be proactively managed. Research studies have proposed statistical modeling approaches to analyze the risk of contributors leaving projects [35, 100, 144, 153]. However, these approaches have yet to be integrated into practical tools for community managers.

S5. Understand the Impact of Attrition: Diagnosing project health by understanding the impact of attrition helps to ensure that the project remains competitive and retains its most needed contributors [82, 144, 144, 145]. For example, studies [16, 103] emphasize the need to identify which parts of a project are most affected by contributor disengagement, enabling managers to understand better and mitigate the effects of attrition. When contributors leave, community managers must leverage data on existing contributors' contributions to identify candidates to take over tasks, thereby minimizing the negative impact of turnover [33, 75].

S6. Promote a Welcoming Environment: Studies have found that a non-welcoming environment negatively impacts retention [93, 100, 107, 129, 134, 136]. One practical approach to creating a welcoming, collaborative environment is to understand contributors' backgrounds [100, 118], which supports early diagnosis of exclusion risks and helps create conditions for long-term engagement. For instance, Trinkenreich et al. [131] highlighted that when contributors' varied backgrounds and contributions are acknowledged, it can enhance collaboration and retention in OSS projects. Similarly, being aware of contributors who are non-native English speakers can help projects recognize the need for multi-language documentation [43]. However, collecting background information (e.g., where contributors come from and which affiliations they belong to) is challenging because it is not readily available and often requires community managers to manually gather and analyze data from multiple sources [82, 100, 136].

5.2.2 Project and Contributor Management. Strategies in this category emphasize automated management tools that enable streamlined processes for both contributors and community managers.

S7. Ensure Privacy Practices: While gathering contributor background information can support inclusion initiatives, privacy concerns are a challenge. Efforts to anonymize developer information can inadvertently lead to misclassifications, potentially causing distress among contributors [82, 100, 127]. Moreover, if such information is unavailable, using tools to aggregate data raises questions about how to protect individual privacy [83, 127]. For instance, Wong et al. [142] and Andrade and Borba [9] reported cases of contributors' email addresses and affiliations leaking, raising significant concerns for many organizations. Balancing the need for contributor information with privacy protections requires careful consideration [155].

S8. Automated Notifications to Monitor Community Health could help support community managers and reduce their workload [120]. Such automated notifications can provide timely updates on problems or metrics that require attention [69, 120, 122]. For example, Lin et al. [84] found that using automated notifications to highlight

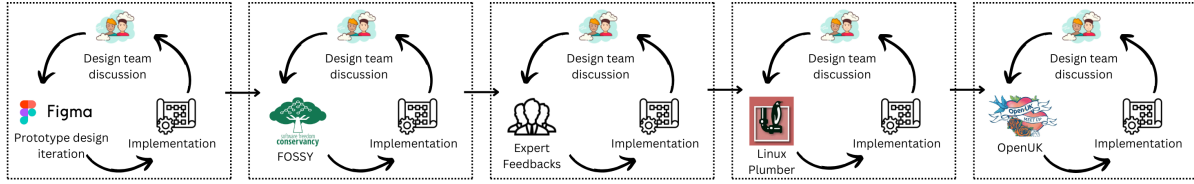


Fig. 2. User-centered design process illustrating iterative design and implementation cycles, incorporating multiple focus group discussions with over 100 OSS stakeholders.

updates helps users quickly focus on important information, reducing the need for constant manual monitoring. However, while dashboards like DevStats [1] offer insights on community activity, they can sometimes add to the cognitive load when community managers are already overwhelmed by routine tasks [73, 98].

S9. Utilizing tools for Contributor Engagement is important to help reduce community managers' workload. Automated processes, such as onboarding newcomers and offboarding contributors, help community managers save time while continuing to foster a collaborative environment. For instance, Steinmacher et al. [120, 122, 123] highlighted that a smooth first contribution is critical for increasing engagement and retention in OSS projects. Welcome emails often have the highest open rates—up to 60% [24].

Another mechanism is to deploy pulse-check surveys that gather data on contributors' well-being and engagement levels [43]. Among the 36 OSS academic studies reviewed, 13 used surveys to understand contributors' experiences in OSS. However, continuously deploying such surveys while addressing privacy concerns is challenging and time-consuming. Therefore, an automated survey distribution system is essential for community managers to maintain consistent communication, collect valuable feedback, and proactively identify potential reasons for attrition [138].

6 PROTOTYPE DESIGN (👥)

Following the DSR paradigm, we decided to evaluate the efficacy of the above strategies by operationalizing them in a web-based prototype, enabling an in situ evaluation. The prototype acts as a vehicle for intervention and knowledge generation, as DSR emphasizes validating technological rules through real-world instantiation [36, 108]. Without a functional prototype reflecting real-life data from their own projects, participants would be unable to assess the strategies' effectiveness. Survey or interview-based evaluation requires recall of past experiences, whereas prototypes anchor feedback in interactive experiences, enhancing the consistency and reliability of responses [51]. We followed a user-centric approach when implementing the strategies in our prototype by conducting iterative focus group discussions with OSS stakeholders (OSS community managers and maintainers) (Figure 2). These discussions helped: (1) bridge user expectations by clarifying their needs and identifying obstacles in integrating the strategies into real-world context [115]; (2) receive continuous feedback and make iterative modifications based on participants' interactions with the prototype [76, 154]; and (3) validate and prioritize features, ensuring that the prototype aligned with both technical feasibility and user expectations [31].

This iterative refinement process contributes to the rigor of our study, as recommended in the DSR paradigm, by gradually expanding the scope and realism of validation while managing the risks associated with deploying interventions in operational OSS contexts [36, 108]. Our iterative, context-sensitive approach, rooted in stakeholder engagement and practical feasibility, helps mitigate potential disruption to ongoing OSS activities while ensuring that our strategies reflect usable and reliable design knowledge prior to in situ evaluation.

6.1 Prototype Walkthrough

Before detailing how we employed the user-centric approach to implement these strategies, we begin by presenting a walkthrough of the finalized prototype (Figure 3) to anchor and contextualize our subsequent discussion.

Let’s take the example of Tyler, a community manager of an OSS project using our prototype. Upon logging into the system, features in Section A provide automated support for engagement and trend reports; using this, Tyler sets up onboarding emails to welcome new contributors and schedules offboarding emails to stay engaged with departing contributors.

Tyler next explores features in Section B, which offers an overview of the project’s retention metrics. This section displays statistics such as the number of active contributors, the number of newcomers, and the project’s turnover rate. Tyler also checks the distribution of contributors across demographic groups—specifically comparing affiliated contributors to volunteer contributors. Noticing that the proportion of affiliated contributors (represented by the purple bar) is significantly wider than that of volunteers (green bar), Tyler suspects a potential decline in volunteer retention. To investigate further, Tyler compares the retention rates between these two groups.

Shifting the focus to Section C, Tyler checks the survival analysis by comparing affiliated and volunteer contributors, observing that the survival rate for volunteers (green line) is lower than that for affiliated contributors (purple line). The analysis reveals that less than 20% of volunteer contributors remain active after one year, prompting Tyler to identify individuals at the highest risk of leaving the project.

To continue this investigation, Tyler then moves to Section D to create a prediction model in investigating the high-risk contributors who are leaving, using contribution activities such as commits, pull requests, and issue tracking; they build a Random Survival Forest model [70] to predict contributors at risk of disengagement. Upon identifying a contributor with the highest risk score, Tyler clicks on their profile, triggering a detailed popup (D’) that outlines the contributor’s activity history. They notice this contributor has been actively involved with pull requests tagged under [TensorFlow API].

To assess the potential impact of this contributor leaving the project, Tyler proceeds to the PR Tag Management System (Section E’) and selects the [TensorFlow API] tag; they verify that this individual is not the most critical contributor for that area. Therefore, there is no high risk for the projects. Tyler repeats this evaluation for other high-risk contributors.

Lastly, to ensure that no contributors are overlooked, still within Section E, Tyler switches to the Newcomer Table tab and Inactive Contributor tab to track recent newcomers and individuals who have been inactive for over six months.

For additional prototype features and detailed technical specifications, please refer to the supplementary documents [41].

6.2 User-centric Design Iteration

In this section, we describe how we operationalized different strategies in our prototypes discussed in Section 5. We focus on identifying the information needed for each strategy, the data sources leveraged, and the methodological decisions of implementation.

6.2.1 Step 1: Prototyping Strategies: The first step in operationalizing the strategies was to prototype each strategy and identify what information was needed. Our research team held weekly meetings (throughout one month) to discuss how to prototype each strategy, focusing on identifying relevant data sources and determining the appropriate methods for implementation, which we describe below.

Retention Metrics to Understand Engagement: We decided to use *retention metrics* that have been discussed in existing OSS studies to operationalize S1. TRACK PROJECT ENGAGEMENT and S2. TRACK NEWCOMERS, including the number of active contributors [153], newcomers [120, 122–124, 144, 153], number of contributors who have left [12, 67, 109, 136, 145, 151, 153], total contributors [12, 90, 151, 153], turnover rate [12, 82, 100, 136, 144, 145, 153],

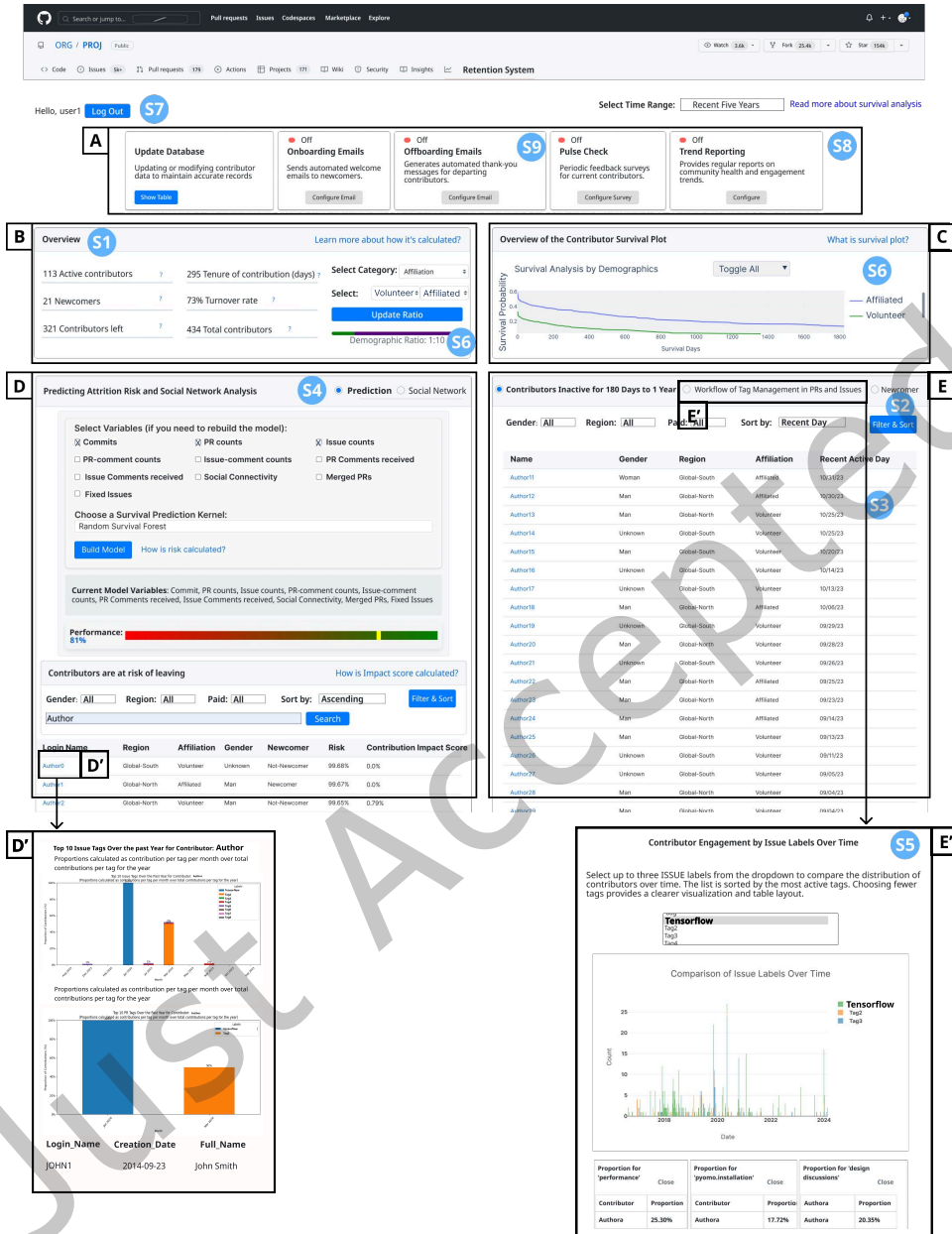


Fig. 3. S1–S9 represent strategy indices, as defined in Table 4. Sections A–D illustrate different components of the prototype, including automation setup (A), project retention overview (B), survival analysis (C), contributor attrition analysis (D) and inactive/newcomer/Tag management (E). Additional detailed views are indicated as D' and E', corresponding to contributor-specific data and tag management. See the supplementary materials for the walk-through video [41].

and the average tenure of contribution (in days) [100, 109, 110, 133, 145, 151, 153]. Furthermore, it is also important to provide community managers with a list of newcomers. By identifying newcomers early, community managers can proactively offer mentorship or resources that improve their experience, ultimately increasing retention and contributing to a sustainable contributor pipeline [42, 120, 121].

Contributor Engagement and Attrition Risks: Community managers need detailed information to S3. TRACK INDIVIDUAL CONTRIBUTORS, such as each contributor’s contribution history, specificity (the part of the project they have worked on), and social network. Access to this information helps community managers better understand contributors’ engagement and identify potential factors that may influence their participation [25].

In addition to static engagement metrics, proactive monitoring of contributors’ activity patterns can provide community managers with early indicators to S4. UNDERSTAND CONTRIBUTOR ATTRITION. Studies in OSS have utilized statistical models to predict the likelihood of individual contributors leaving a project [13, 35, 35, 56, 110].

Once community managers acquire proactive information about declining contributor activity and engagement, it’s important to S5. UNDERSTAND THE IMPACT OF ATTRITION on the project. This involves assessing the contributor’s past contributions, determining the project areas that might be affected by their departure, and identifying other team members who can assume those responsibilities. One approach is to leverage the existing tagging systems widely used in issue-tracking platforms [10]. For instance, issue tags can quickly indicate categories such as bug, feature, and optimization, helping contributors identify areas of interest or expertise [71]. Similarly, in some communities, pull request tags can indicate the progress or priority of changes [71]. Therefore, by analyzing the distribution of contributions across these tags and identifying the contributors associated with them, one can gain insights into who is working on specific areas [137]. If a contributor who has been heavily involved in a particular tag stops contributing, community managers can identify the gap to find another contributor to take on the responsibility.

To S6. PROMOTE A WELCOMING ENVIRONMENT, it is important for community managers to understand if certain demographic groups have a higher risk of attrition [12, 82, 100, 136]; for instance, monitoring retention rates between affiliated and non-affiliated contributors can reveal disengagement patterns [62]. Existing studies have used the Kaplan-Meier survival analysis estimate to compare demographic groups and identify potential retention imbalances within the community [100]. In our prototype, we chose to use the commonly studied demographic lenses in OSS research: newcomer status, gender, region, and affiliation [12, 43, 82, 100, 136].

Access Controls to Maintain Privacy Practice: For all of the information and aggregated data we have discussed above, access control mechanisms to S7. ENSURE PRIVACY PRACTICES, such as secure login systems, are essential in protecting sensitive data within OSS projects to reduce the risk of unauthorized exposure [96]. Features such as a login page that only allows community managers to sign up – with sign-up requests being sent to a designated administrator for approval – help ensure that sensitive information is kept private to the fullest extent possible.

Automated Approaches to Community Health Monitoring and Contributor Engagement: S8. AUTOMATED NOTIFICATION TO MONITOR COMMUNITY HEALTH is essential to help community managers with their workload. In our prototype, we incorporated a feature that allows community managers to schedule notifications for project reports that include data such as the number of newcomers who have joined or contributors who may be leaving. This approach ensures that managers receive information they consider relevant in a convenient manner.

Regularly engaging with contributors is vital for addressing disengagement early [32, 121]. However, managing high volumes of routine communication can be time-consuming and may lead to burnout. To mitigate this, we propose to S9. UTILIZE TOOLS FOR CONTRIBUTOR ENGAGEMENT. For example, automating the sending of welcoming emails to onboard newcomers, expressing gratitude to departing contributors, and conducting wellness surveys via email can foster a positive community environment. These automated interactions help maintain consistent communication and support within the community.

6.2.2 Step 2: Initial Prototype: We designed the initial prototype by following the design principles outlined in Few [47] to display the necessary information from the previous step. Three researchers summarized these design principles, ensuring that each design element chosen for implementation was grounded in these established guidelines. We began with low-fidelity prototyping and conducted two sandbox sessions to finalize the low-fidelity prototypes. We then used Figma³ to design the prototype with interactions. The detailed design guidance we followed can be found in supplementary documentation [41].

6.2.3 Step 3: Collecting feedback from stakeholders: The next step in our design approach was to gather stakeholder feedback, allowing stakeholders to freely express their ideas and provide insights on how users might interact with the tool [18]. As shown in Figure 2, we collected four rounds of feedback, including three focus group discussion sessions with OSS conference participants and 10 one-on-one consultation sessions with OSS community managers.

Stage I. Initial Feedback from Focus Group Discussion I: We had our first focus group discussion at the FOSSy Conference 2023 [116]. We used a Figma high-fidelity prototype for this session instead of an implemented version for flexibility in quick iterative improvements.

The focus group discussion session was about an hour long, and we spent the first 10 minutes showcasing the prototype's features and functionalities, while the remaining time was allocated for audience feedback. The first and last authors led the group discussion. There were 20 participants from whom we collected various suggestions, which ranged from topics such as UX design to backend work. One of the main pieces of feedback we received was about predicting the likelihood of a contributor leaving the project. Our audience suggested that the variables used in predictions should not be fixed, as different projects have different engagement patterns. They emphasized that the users should decide the metrics, as different communities can have unique characteristics and needs.

Discussion and Improvement: After the focus group discussion, the research team convened to discuss and iterate over these feature requests and implementation through weekly meetings spanning one month. To better incorporate the main suggestion from the audience, we also agreed that when predicting the likelihood of individuals leaving the project, we should not be limited to a particular statistical model (regression). Thus, we decided to implement all three standard survival models: Cox regression [27], neural network [37], and random forest [70]. This approach gives users more flexibility to fit different variables and select models based on their preferences and needs. Additionally, we provide an indicator for showing the performance of each model to help them select the most suitable model (for their metrics of interest). This ensures that users can make data-driven decisions when interpreting and applying their projects' results.

Another requested feature was to list inactive contributors [28], defined as those who have paused their contributions for more than six months but less than a year. This report allows community managers to identify and target these contributors with tailored re-engagement strategies.

Stage II. Initial Implementation and Consultations with Kubernetes and Flutter: In this stage, we implemented the first version of the prototype. The initial implementation closely followed the high-fidelity prototype. We chose to use the Flask framework of the Python language [11] for implementation, as it is a low-cost, efficient, and quick approach.

Instead of placeholders for data usage, we used actual OSS project data for the implementation, which also helped test the prototype's scalability and usability. We chose Flutter⁴ as a use case because (1) it has a large community with more than 500 contributors and over 10 years of history and (2) is very active, with more than 10 issues/pull requests updated every day. Additionally, we had access to talk to the Flutter community managers to collect feedback.

³<https://www.figma.com/prototyping/>

⁴Flutter GitHub Repository. Available at: <https://github.com/flutter/flutter>

All features were implemented using CSS, HTML, and JavaScript. REST APIs were leveraged to pull data from Flutter’s GitHub repository. For this proof of concept implementation, we chose the simplest inference tools for identifying contributor demographic information. We inferred contributor affiliations based on email domains (e.g., google.com vs. gmail.com). Non-corporate email domains were classified as unknown. To infer the gender and region of contributors, we used the Namsor API [113], a name inference tool that estimates the gender of an individual on a probability scale from -1 to +1, by incorporating the individual’s full name and geographic information. However, relying solely on geographic data from GitHub profiles reduces precision [135], mainly when contributors’ GitHub locations differ from the geographic origin of their names. To address this, we implemented a two-step process: first, we used the Namsor API to predict the geographic origin of contributors’ names; then, we used both the predicted origin and the full names to infer contributors’ gender [63]. To minimize errors, we applied a probability threshold, filtering out predictions with confidence levels below 90%, in line with best practices from prior research [23, 113]. For implementing the survival analysis for Flutter, we conducted preliminary tests to validate the reliability of our approach as detailed in Feng [40]. This prior work provides a foundation for incorporating survival analysis into our prototype.

Consultations with Expert (Flutter): After implementing the system, we had a two-hour-long meeting with the project manager of Flutter to walk him through the system and its features. He validated the need for these strategies when it came to community management and provided valuable feedback. For example, he suggested creating a contribution impact score to quantify and compare the impact of individual contributors’ efforts in an aggregated form.

Discussion and Improvement: We addressed this primary feedback through the implementation of a “contribution impact score”, which calculates a contributor’s score by considering their number of contributions in proportion to a baseline score configured as the highest number of contributions made by a contributor. For example, if the top contributor has 50 commits, another contributor with 40 commits would receive a score of 0.8. In our case, we treated all types of contributions as equal as a proof of concept. Future iterations can refine this strategy by assigning weights to different contribution types for a more nuanced assessment tailored to the specific needs of the OSS community.

Consultations with Experts (community leaders from CNCF): We collected the next round of feedback with the modified prototype. We attended KubeCon 2023 [85] and invited 10 community leaders from different communities to review our (implemented) strategies.

The sessions were informal, with each lasting 15-20 minutes. We used the first three minutes to introduce the strategies through a walkthrough of the prototype and then let the project leaders freely explore the system to provide feedback. We received comments on what data sources to use as well as the look-and-feel of the prototype itself. One community leader suggested that strategies for engagement need to cover the entire community, which includes more than just the main Kubernetes repository, as contributors in their community may contribute to different side repositories depending on their project goals. Another project leader had concerns about collecting demographic information, particularly because of privacy issues, and how to ethically ask people to identify themselves.

Discussion and Improvement: Similar to the previous steps, we revisited and discussed the feedback with our design team, refining the prototype accordingly. However, we were unable to find a suitable solution regarding the privacy concerns related to the contributor’s demographic information. As a result, we decided to hold another focus group discussion (see next stage) to gather additional feedback from a broader range of participants.

Stage III. Iterative Refinement from Focus Group Discussion II: We hosted a focus group discussion session at the Linux Plumbers Conference [86], where the majority of participants were OSS maintainers, allowing us to get their feedback. During the discussion, we mainly focused on how to help community managers understand the contributors’ background and privacy issues (one of the sticking points from the prior stage). We presented our approach to infer contributors’ identity by using research tools to infer their gender and region.

Although participants had concerns about approaches to infer contributors' identities, this feature was also highly voted by the OSS community, so we could not simply drop it. Some participants mentioned that if we use inference tools for gender identity and a contributor sees it, it could offend the contributor if their gender is misclassified. Additionally, some participants suggested we could ask contributors to self-report their identities, but others raised concerns about newcomers having to self-report their identity when joining a project, as it may be concerning for those who do not want to disclose this information or are unsure about their long-term commitment.

Discussion and Improvement: We updated our prototype design such that demographic information remains private behind login pages and administrators must approve access to this information. Additionally, maintainers felt that such a system was not meant for the general public (where there can be more negative repercussions of revealing demographics), but could be used by community managers to manage retention. We also gave project leaders the option to create wellness check surveys so that contributors can self-report their demographics if they feel comfortable. Additionally, we provided a mechanism to allow managers to review and correct the inference tool's outcomes to improve accuracy and give the community more flexibility.

We acknowledge that we implemented strategy (S6) by using automated inference of demographic information, which may be unsuitable for real-world applications. Future implementations of this strategy need to be designed for transparency and ethical data handling of sensitive data in alignment with varying community norms, data protection regulations, and cultural contexts.

Stage IV. Final Feedback from Focus Group Discussion III: With the updated version of the prototype, we conducted another focus group discussion at OPENUK 2024 [95], with over 30 remote participants for a 30-minute session. Ten minutes were devoted to introducing the strategies through a prototype walkthrough, followed by 20 minutes of open discussion primarily focused on clarifying the prototype and its features. We received no further comments or suggestions; several attendees expressed interest in using the prototype, and four participants later emailed the speaker to inquire about adopting it for their communities. This feedback indicated that the prototype was ready for in situ evaluation (See the supplementary documentation [41] for a prototype walk-through video).

7 IN SITU USER EVALUATION

To enhance the rigor of our study and evaluate the strategies in real-world settings, we conducted an in situ user evaluation, using the prototype features as a proxy to assess the effectiveness of the retention strategies. In situ evaluation refers to assessing an artifact within the actual environment in which it is intended to be used, offering ecological validity by capturing realistic workflows, stakeholder roles, and decision-making contexts [29]. As demonstrated in Lechelt et al. [77], in situ approaches allow participants to reflect on and interact with the system “in the moment,” providing authentic, temporally grounded feedback that might otherwise be lost in post-hoc reflections or surveys. We performed this evaluation by applying the prototype to two large OSS communities—Pyomo/pyomo [126] and Microsoft/DeepSpeed [89]. Both projects have more than 100 contributors.

We selected the projects to ensure they differed in their governance structures and application domains to understand the generalizability and practicality of the implemented strategies. Pyomo is a Python-based scientific software package widely used in academic settings, offering optimization tools for modeling and analysis. Over the past seven years, it has been developed and maintained by a community of researchers and practitioners in their spare time. DeepSpeed is a deep-learning optimization library for PyTorch, sponsored by Microsoft and commonly adopted in the software industry. It has nearly five years of history and a more corporate governance model.

Table 5. Demographic information of interview participants.

ID	Community	Gender	OSS Experiences	OSS Management Experience
PY1	Pyomo	Man	3 to 5 years	3 to 5 years
PY2	Pyomo	Woman	3 to 5 years	3 to 5 years
PY3	Pyomo	Woman	6 to 10 years	3 to 5 years
PY4	Pyomo	Woman	> 10 years	3 to 5 years
D1	DeepSpeed	Man	3 to 5 years	3 to 5 years
D2	DeepSpeed	Man	3 to 5 years	3 to 5 years
D3	DeepSpeed	Man	3 to 5 years	1 to 2 years
D4	DeepSpeed	Man	1 to 2 years	1 to 2 years

7.1 Method

Evaluation Approach. We adopted a cross-sectional evaluation of the project primarily to obtain immediate, actionable insights about the effectiveness of the implemented strategies. Cross-sectional evaluation designs have been widely validated for capturing current user behaviors and adoption potential [74, 139], making them well-suited for rapid assessment. By contrast, longitudinal approaches (e.g., diary studies) would be complicated by the inherently dynamic nature of OSS projects. For example, new package releases or onboarding programs could introduce major shifts in retention. To account for differences in perspectives and needs of individual managers, we invited all the governance members (project community managers) from both Pyomo and DeepSpeed to participate in the study.

Evaluation Questions. As we used a web-based prototype to implement the strategies, we leveraged the evaluation framework proposed by Lam et al. [76] for visualization tools. These include: (1) What features are useful? (2) What features are missing? (3) How can features be reworked to improve the supported work processes? (4) Are there limitations to the current prototype that would hinder its adoption? (5) Is the prototype understandable, and can it be learned?

We group these questions into two top-level evaluation questions:

EQ1. How does the prototype help project managers mitigate community management challenges? This covers sub-questions (1), (2), and (5), and focuses on the prototype’s usefulness, its ease of learning, and any missing features. It prompts participants to reflect on how the prototype supports their retention management needs.

EQ2. How can the prototype complement project managers’ current retention management approaches? This aligns with sub-questions (3) and (4), assessing how the prototype integrates into existing workflows and identifying potential barriers to adoption.

Participant recruitment. We first reached out to the project managers of Pyomo and DeepSpeed who had expressed interest in evaluating our prototype. These managers then helped us recruit the remaining project manager team members. We recruited eight participants in total: four from Pyomo and four from DeepSpeed. Table 5 shows participants’ demographics.

The study design was approved by the university’s IRB, which determined that the study involved minimal risk. We compensated our participants with a \$50 gift card for their time. The study was conducted in-person (n=3) or remotely (n=5) via Zoom, depending on the participants’ preferences. All interactions with the prototype were browser-based.

Evaluation protocol. The study began with an introduction to the project, which described the structure of the study, followed by three demographic questions: gender, OSS seniority, and project leadership status. Participants then viewed a six-minute tutorial video. The video was designed to ensure that information about the prototype was presented consistently to all participants. The participants were then free to explore the prototype while thinking aloud about how they would use it to manage their community, highlighting any features and information they found interesting or that needed improvement. To ensure that participants understood the think-aloud process, we prepared a practice session to walk them through the think-aloud protocol.

After the exploration, participants completed two sets of post-study questionnaires. The first set of questions aims to understand participants' agreement on *“how each implemented strategy within the prototype helped to mitigate retention challenges”*. The second set explored *“how the prototype could complement or enhance participants' existing community management strategies”*. Additionally, we included questions such as whether they would be willing to purchase it, recommend it to friends, and continue to use it. Lastly, we asked open-ended questions, focusing on which features they liked or disliked and any other recommendations they had. See the supplementary documentation [41] for details about evaluation design.

Before conducting the study with participants, we piloted it with two graduate research assistants experienced in OSS research. Based on their feedback, we revised the study design, adding a think-aloud practice to support participants unfamiliar with the method. To gather further input, we presented our study design to a graduate HCI design class at Oregon State University, involving 30 graduate students. Feedback from the students and the professor led to additional revisions, such as inviting a native English speaker to record the tutorial video to minimize language barriers. Finally, we conducted two more pilot studies with OSS practitioners, iterating until no further feedback or suggestions were provided. Refer to the supplementary materials [41] for details about the study, tutorial videos, and study scripts.

Data analysis. We began by transcribing the video and audio recordings. We used qualitative analysis to answer the evaluation questions by analyzing participants' user experiences, their think-aloud responses, and answers to post-study open-ended questions. Two authors independently coded the four interview transcripts to identify initial themes. We then met to compare and discuss codes using the constant comparison method, refining them into a preliminary codebook by merging overlapping codes and creating new ones where necessary. The same two authors independently coded the remaining interview transcripts using this codebook. After completing the coding, they met again to review the code, negotiate differences, ensure consistency, and resolve discrepancies. We also quantitatively analyzed the post-study questionnaire data to complement our qualitative findings.

7.2 Results

EQ1. Helpfulness in Mitigating Challenges: Column one in Table 6 presents participant responses on how our prototype could mitigate community management challenges. Among the eight participants, there was strong agreement across all questions, with no more than one disagreement for any challenge across the two projects.

Community managers found the prototype helpful in MANAGING OVERWHELMING RESPONSIBILITIES [PY3, PY4, D1–D4]⁵. *“I think the most helpful features are to understand contributions—specifically, the contributor activity section, which shows the top contributors for different areas. That is very important, along with the at-risk information, as both help me plan the project's features”* [D2].

Participants emphasized the prototype's helpfulness in MANAGING RETENTION. *“Seeing the overall retention trend. That's just interesting”* [PY1]. *“I know people leaving, how we can do with this limited bandwidth. I'm always feeling guilty to not give praise. Early sign. When I notice it is too late”* [D1]. Similarly, a participant noted the difficulty of tracking turnover manually, explaining, *“The turnover rate, I guess... Never thought of our repo that way... It's not easy to track when someone's gone... seeing it in a number is more straightforward”* [D2]. *“Our biggest issue with retention is just actually being able to get to and solve issues. If it helped us do that in a better way, especially with our top contributors, solving their issues, you know, giving that priority. That would be a big help”* [D3].

The prototype received unanimous agreement for its helpfulness in ENGAGING CONTRIBUTORS [PY1–PY4, D1–D4]. Participants emphasized the importance of understanding and interacting with contributors, particularly less visible ones. *“Interacting with contributors... we get asked a lot who our users are. We actually really don't*

⁵PY: Pyomo participant; D: DeepSpeed participant.

Table 6. The table presents agreement levels (N=8) for identified challenges, proposed strategies, and future adoption intentions.

Mitigate Challenge?	Agreement (N=8)	Complement Strategy?	Agreement (N=8)	Future Adoption?	Agreement (N=8)
Overwhelming Responsibilities		Track Project Engagement		Will you recommend?	
Challenges of Managing Turnover		Track Newcomers		Will you use?	
Time-consuming to Engage and Grow the Contributor Base		Track Individuals' Contributions		Will you purchase?	
Challenges in Tracking Project Data		Predict Contributor Attrition			
Challenges in Tracking Contributor Data		Understand the Impact of Attrition			
Challenges in Acknowledging Hard-to-Track Contributions		Promote a Welcoming Environment			
Challenges in Fostering Inclusive Collaboration		Ensure Privacy Practices			
Challenges in Anticipating Contributor Attrition		Automated Notifications			
Challenges in Ensuring Data Privacy While Tracking		Utilize Tools for Contributor Engagement			

The dark green section represents the percentage of participants who "Strongly Agree," while the light green section shows those who "Agree." The gray section corresponds to "Neutral," the light red corresponds to "Disagree," and the red section to "Strongly Disagree."

know... but the quiet people using Pyomo, we have no idea" [PY4]. D4 mentioned the potential to foster engagement: "You can send the email... help contributors feel welcome, especially people [that] are not affiliated, but you encourage them to contribute". Additionally, the idea of automated features for personalized welcome emails resonated with PY2: "Having an automated system to do kind of a personalized welcome, I think, is pretty neat... I'm not aware of an easy way to do that".

Another unanimous agreement was on the prototype's helpfulness in TRACKING CONTRIBUTORS' DATA [PY2–PY4, D1–D4]. "If you need to learn about somebody who has done something, you have to go to GitHub and check... There's no way... aggregated, and all the contribution data of one person" [D2]. Similarly, PY4 mentioned, "Aggregating the different types of contributions... I actually really like that. I like what you're pulling out about... whether certain sub-packages and those sorts of things are getting support".

Community managers found the prototype particularly helpful for RECOGNIZING HARD-TO-TRACK CONTRIBUTIONS that are not easily accessible through GitHub's interface, which helped them better understand the extent of each contributor's engagement [PY1–PY4, D1–D4]. "This is the most helpful for me because it gives me a chance to see people I don't recognize and remember why they contributed" [PY1]. "I think there are a lot of contributions, particularly in, you know, contributing ideas through issue comments or contributing PR reviews and things like that. So just contributions to keeping the repo healthy" [PY2].

Another aspect where the prototype demonstrated its helpfulness was in FOSTERING INCLUSIVE COLLABORATION [PY2, PY4, D1, D3, D4]. PY1 mentioned "I like that idea a lot... In a community like Pyomo or Spack, which is huge, you can toggle different demographics to compare with the overall. If a contributor is left out, it can help identify if they need more support". Similarly, PY2 mentioned "Assessing the demographics of those who may disengage... I haven't seen any other tools for gathering GitHub statistics like this" [PY2].

However, we observed some reasons for disagreement in certain areas. Regarding MAINTAINING PRIVACY, the responses were mostly neutral due to differing work environments. "Maintain privacy when aggregating

contributors' data... I'm going to put this as neutral because, you know, coming from a national lab viewpoint, we have some strict ideas and policies around privacy" [PY2].

EQ2. Complementing Community Managers' Current Management Approach: Column two in Table 6 shows participants' agreement on how the prototype complements their current retention management strategies. The majority of participants agreed that the prototype effectively complements their current strategies.

Participants agreed that the prototype could assist in tracking contributors' activities and engagement [PY1–PY3, D1–D4] and enhance interactions with contributors [PY2, PY3, D1–D4]. "I think these statistics are interesting... seeing changes in activity could be interesting and useful. I also think engaging with contributors is by far probably the most useful feature" [PY2]. One community manager from DeepSpeed was surprised by the data provided, "Whoa, whoa, [...] active contributors" [D1], and [D4] noted, "This could be super helpful for understanding which contributors may engage".

In addition to tracking overall engagement, participants unanimously agreed on the value of tracking newcomers [PY1–PY4, D1–D4]. "Tracking how many newcomers... must be useful" [D1]. "I really like that in particular. The idea of being able to see the newcomers... especially in a community like Pyomo, where a lot of us are paid to do it" [PY1].

However, there were a few aspects where we observed one instance of disagreement, particularly around data privacy and contribution attrition. The reasons for this stemmed from differing management and OSS philosophies. PY1 mentioned that "Maintaining privacy for contributors' data... we don't really care, so probably not just for us". "Privacy part... I mean, it's all just kind of public data, in a way, anyways" [D4]. Similarly, in discussing contribution attrition, "For us... that's actually probably not the most important thing. I think there are a lot of people we know we're going to lose anyway" [PY4].






Lastly, column three of Table 6 shows participants responses for future adoption; we asked participants three questions regarding their willingness to continue using the tool, recommend it to other managers, and purchase it. All participants indicated they would recommend our prototype to other managers. "I'm gonna say, I would definitely recommend it" [PY2]. Other participants highlighted how it could help balance workloads: "You know, we only have limited space to run PRs, so a lot of our stuff gets backed up" [D3] and "[it] could be helpful in balancing the workload across different projects" [PY2]. Regarding the willingness to purchase the tool, participants gave careful consideration, mentioning that any purchase decision would need to go through community management meetings: "So the reason I'll say 'maybe' on the willingness to purchase is because we don't have a lot of direct money" [PY1].

8 DISCUSSION

As we observed, managing contributor retention in OSS is a multifaceted challenge, requiring a balance between automated mechanisms, privacy concerns, and access to retention-related data. Our findings highlight several tensions that emerge in attempting to design effective retention strategies while ensuring inclusivity and fairness. **Privacy vs. Data Aggregation.** One tension we observed lies in the balance between protecting contributors' privacy and leveraging data aggregation for retention insights. While we can use automated data collection to track contributors' activities and predict attrition, it also raises ethical concerns regarding the surveillance of these contributors. Our findings indicate that while community managers benefit from tracking contributor activity to anticipate disengagement (C9), there are significant concerns regarding data privacy and potential misuse (C10). Some participants hesitated to implement predictive models due to ethical concerns surrounding monitoring contributor behavior. The aggregation of contributor activity may reveal individual disengagement patterns, potentially impacting personal autonomy and raising concerns about data ownership.

Automated Notifications and Engagement Fatigue. Another point that was evidenced during our study was the role of automated notifications and outreach in managing retention. While the literature showed that

Table 7. Triangulation of Challenges and Strategies in Managing Retention in OSS

	Interview 	Survey 	Literature Review 	User Centric Design 	In-situ User Evaluation 
Challenges	✓	✓	✓		
Strategies			✓	✓	✓

automated systems can help community managers reduce their workload (S8) [140], several participants (P2, P4, P6) reported concerns that frequent notifications might contribute to notification fatigue and disengagement (C2). Participants noted that contributors who receive excessive automated messages may feel overwhelmed, perceiving a lack of personal touch in interactions. Our results also suggest that the impact of automated outreach varies depending on project size and governance structures. That said, it becomes important to work on further understanding how personalization and contextual adaptation may be used in each specific case to avoid harming the project instead of helping the maintainers.

Cross-sectional vs. Longitudinal Retention Insights. One interesting point to discuss regards whether cross-sectional (snapshot-based) or longitudinal (historical) approaches provide more useful insights for retention tracking. Our study found that many OSS projects rely on cross-sectional retention analyses due to time and resource constraints (C5, C6). However, this approach often fails to capture long-term contributor trajectories, leading to misinterpretations of short-term fluctuations as retention failures. On the other hand, longitudinal tracking offers more profound insights into contributor retention trends, but it also introduces complexity and noise (C1, C2). OSS community managers indicated that while they value historical retention data, sustained monitoring is challenging due to competing responsibilities (P2, P3, P6). This underscores the need for hybrid approaches that integrate both short-term and long-term retention signals, allowing maintainers to identify both immediate risks and broader systemic trends.

Triangulation of Our Findings. We used triangulation to ensure the robustness and reliability of our findings: challenges in managing retention and strategies in supporting contributor retention. The summary of our triangulation is provided in Table 7.

We first investigated the challenges of managing retention in OSS through multiple triangulation approaches. We began by synthesizing challenges from interviews with experienced OSS managers whose extensive expertise laid the foundation for understanding these challenges. We then triangulated these findings through a multi-vocal literature review of academic and community-driven gray literature. Finally, we surveyed with OSS community managers with varying experience levels as a final validity check. Refer back to Table 3 for a summary of the complete triangulation process and its results.

As for strategies in supporting OSS community managers in diagnosing and managing retention challenges, to ensure their helpfulness and practicality, we also employed multiple triangulation validation strategies. We began by synthesizing strategies through literature reviews. We then operationalized them in a web-based prototype and conducted an in situ evaluation. The prototype was designed using a user-centric approach, engaging over 100 OSS community managers throughout the iterative development process to ensure the operationalized strategies in our prototype were aligned with real-world needs. We then conducted an in situ evaluation within two OSS communities. These evaluations confirmed their helpfulness and generalizability to real-world OSS management. The complete results are summarized in Table 4.

Impact of Our Work. Our work addresses challenges in managing OSS communities, particularly in managing contributor retention, which has become a distinct, complex task [60]. In larger communities, this responsibility often requires dedicated teams, reflecting the growing importance of contributor management [60, 153]. Our research offers a consolidated list of challenges, enabling future research and OSS governance committees to understand these issues better and address them effectively. Meanwhile, our work also highlights the importance

of responsibility in managing contributors' retention. Such emphasis helps OSS governance committees become aware of and acknowledge community managers' responsibilities and difficulties faced.

We provided a set of strategies to support OSS community managers in diagnosing and responding to contributor retention challenges. These strategies were integrated into a prototype that demonstrates how they can be operationalized in practice. Validated through multiple rounds of stakeholder discussions and in situ evaluations, the strategies provide a guidebook that OSS communities can adapt to their specific needs. For example, current tools and practices in OSS governance could integrate the strategies we synthesized to enhance their design and better address the challenges faced by community managers.

Furthermore, our research approach serves as a showcase for identifying and establishing the ground truth of challenges and strategies to OSS ecosystems' unique social and organizational dynamics. Our design approach is rooted in human-centered practices. By engaging stakeholders throughout the process, we ensured that the strategies we reported in this study aligned with their needs, goals, and challenges.

Future Design Opportunities: Incorporating Design Strategies with Large Language Models: Given the increasing prevalence of large language models (LLMs) and generative AI [66], integrating these technologies with our design strategies can help automate retention management tasks in OSS communities (S8 and S9). LLMs can enhance automated notifications by generating personalized project activity summaries and identifying signs of contributor disengagement. By analyzing contributor activity patterns, LLM-powered alerts can provide community managers with actionable insights, such as flagging contributors who may need additional support or recognizing emerging challenges that impact project health. Additionally, LLMs can generate contextualized messages that balance encouragement and information, reducing the likelihood of generic or overwhelming notifications. Moreover, LLMs can improve contributor engagement by personalizing engagement strategies. For example, community managers may not even need to draft a generic onboarding message. LLMs can tailor recommendations for tasks, learning resources, and mentorship opportunities based on contributors' prior activity and skill levels.

Not All Turnover Is Equal: Rethinking Retention Metrics. While our study focuses on surfacing and responding to retention challenges, it is essential to recognize that not all contributor departures are inherently negative. Departures may occur for various reasons, some of which reflect natural or beneficial lifecycle events. For instance, contributors may disengage after completing a specific feature they prioritized [15]; a pattern observed in our findings (C3, P5). Similarly, maintainers may step down after years of service to avoid burnout and enable new leadership, fostering community renewal [82, 91, 149]. Such transitions signal healthy community dynamics and should not be treated as retention failures.

By contrast, departures driven by exclusion, burnout, or unresolved interpersonal conflicts pose significant risks to community sustainability [91]. Future research could explore a typology of departures, such that they are categorized as concerning (e.g., exclusion-driven), natural (e.g., task completion), or positive (e.g., leadership transitions), to guide context-sensitive retention strategies. Such a framework could enhance diagnostic tools, enabling community managers to prioritize interventions to avoid negative disengagement while acknowledging natural or positive transitions. Longitudinal studies of contributor trajectories could validate this typology by identifying patterns of exit intent and informing more effective retention practices, as they could observe changes over time to trace natural contributor behavior patterns [34].

9 LIMITATIONS

Like any empirical study, our research contains limitations. Here, we discuss these limitations and the measures we employed to address them.

The initial list of challenges was derived from the six participants, which may lead some to question whether the challenges we identified were representative enough. To account for this, we analyzed for saturation, and

no new challenges emerged in the last two interviews. In qualitative analysis, the adequacy of sample size is determined by whether data saturation is reached, rather than a minimum number of interviews [52, 112]. Although the number of participants required for saturation is debated [128], prior work suggests that key themes typically emerge within six interviews [61], while Creswell and Poth [30] recommends a range of five to eight. To overcome any limitation from the number of interview participants, we triangulated the challenges by conducting multi-vocal literature reviews and expert evaluation surveys. Our participants represented diverse backgrounds, including differences in gender, experience, and seniority in managing OSS communities.

Following the same methodology, the identified strategies were also triangulated from a literature review, a user-centric design approach, and in situ user evaluation involving over 100 stakeholders from OSS communities. This approach allowed us to gather rich, contextual insights and iteratively refine our strategies in real-world settings. We operationalized our strategy into a prototype through user-centric design. This prototype represents one of countless possible ways to implement the identified strategies. While it aims to understand the helpfulness and practicality of these strategies, it is not intended as a fully scaled or universally applicable solution. Instead, the prototype serves as a proxy in our triangulation methodology to validate the practicality and helpfulness of the strategies in real-world OSS contexts.

During the prototype development phase, we adopted a user-centric design approach and iteratively collected feedback through three rounds of focus group discussions from different conferences across different communities and regions. Due to conference constraints, we did not collect detailed demographic data from focus group participants. However, to enhance generalizability, we conducted these sessions across multiple OSS venues (e.g., FOSSY, Linux Plumbers, OpenUK) and engaged participants from a range of communities and geographic regions.

To evaluate strategies and ground our findings in authentic usage settings, we conducted an in situ user evaluation with managers from two OSS projects to assess the operationalized strategies embedded within our web-based prototype. While such an evaluation does not capture long-term behavioral adoption or usage trajectories, it does allow managers to engage with strategy representations using their own project data and decision-making processes, yielding context-rich and temporally grounded insights [29, 77]. Moreover, we opted not to conduct a controlled study, as it was not deemed necessary for our research; the goal of the evaluation was to understand the perceived helpfulness of the strategies rather than to compare the tool itself against another tool [58].

Another concern could be the representativeness of the number of evaluation participants. In addition to meeting the qualitative analysis requirements mentioned in the first limitation, this user evaluation is just one step in our triangulation process, and the overall exploration of strategies has involved over 100 stakeholders. Moreover, our user evaluation was conducted across two communities with varying natures and scopes, as discussed in Section 7, allowing us to capture nuanced perspectives on how these strategies function in practice.

Finally, regarding the predictive models for forecasting contributor attrition, different models exhibit varying levels of accuracy. Therefore, we provided a set of three models along with the accuracy information of each model to help users select the model that is most appropriate for their needs.

10 CONCLUSION

In this paper, we report ten challenges and nine strategies for managing contributor retention in OSS. Our findings are grounded in multi-triangulation, with over 100 stakeholders participating in our project. Our results highlight both the complex responsibilities community managers face in supporting contributor retention, as well as a set of strategies to inform governance decisions and strengthen retention practices across OSS communities.

We believe that our study will be valuable to OSS communities by identifying challenges from multiple perspectives and equipping community managers with data-driven strategies to foster sustainable engagement.

By bridging the gap between academic research and real-world OSS practices, our work showcases a human-factors-oriented framework that serves as a foundation for future researchers developing theories on OSS sustainability. Looking ahead, our future work will expand on this foundation by scaling the strategies and exploring how AI can enhance retention efforts.

The research artifacts for this study are available publicly at the companion website [41].

ACKNOWLEDGMENT

We thank all participants in our study, including those who contributed through interviews, surveys, and design discussions, for their time and valuable insights. We are especially grateful to the managing teams of the Pyomo and DeepSpeed projects for their participation and support. We also thank Edward Gilmour for his helpful feedback during the preparation of this paper. This work was partially supported by NSF Grant Nos. 2303043, 2303042, and 2303612.

REFERENCES

- [1] 2024. CNCF DevStats. <https://devstats.cncf.io/>. Accessed: 2024-06-08.
- [2] 2025. Foundations of Software Engineering (FSE) 2025. <https://conf.researchr.org/home/fse-2025>. Accessed: 2024-06-08.
- [3] 2025. International Conference on Software Engineering (ICSE) 2025. <https://conf.researchr.org/home/icse-2025>. Accessed: 2024-06-08.
- [4] Adem Ait, Javier Luis Cánovas Izquierdo, and Jordi Cabot. 2022. An empirical study on the survival rate of GitHub projects. In *Proceedings of the 19th International Conference on Mining Software Repositories*. 365–375.
- [5] Reem Alfayez, Pooyan Behnamghader, Kamonphop Srisopha, and Barry Boehm. 2017. How does contributors involvement influence open source systems. In *2017 IEEE 28th Annual Software Technology Conference (STC)*. IEEE, 1–8.
- [6] Bradley J Alge, Gary A Ballinger, Subrahmaniam Tangirala, and James L Oakley. 2006. Information privacy in organizations: Empowering creative and extrarole performance. *Journal of applied psychology* 91, 1 (2006), 221.
- [7] Amazon. 2024. OSS Dashboard. <https://github.com/amzn/oss-dashboard>. Accessed: 2024-06-08.
- [8] Talat Ambreen, Naveed Ikram, Muhammad Usman, and Mahmood Niazi. 2018. Empirical research in requirements engineering: trends and opportunities. *Requirements Engineering* 23 (2018), 63–95.
- [9] Rodrigo Andrade and Paulo Borba. 2020. Privacy and security constraints for code contributions. *Software: Practice and Experience* 50, 10 (2020), 1905–1929.
- [10] Deeksha Arya, Wenting Wang, Jin LC Guo, and Jinghui Cheng. 2019. Analysis and detection of information types of open source software issue discussions. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*. IEEE, 454–464.
- [11] Fankar Armash Aslam, Hawa Nabel Mohammed, and Prashant S Lokhande. 2015. Efficient Way Of Web Development Using Python And Flask. *International Journal of Advanced Research in Computer Science* 6, 2 (2015).
- [12] Guilherme Avelino, Eleni Constantinou, Marco Tulio Valente, and Alexander Serebrenik. 2019. On the abandonment and survival of open source projects: An empirical investigation. In *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE, 1–12.
- [13] Lingfeng Bao, Xin Xia, David Lo, and Gail C Murphy. 2019. A large scale study of long-time contributor prediction for GitHub projects. *IEEE Transactions on Software Engineering* 47, 6 (2019), 1277–1298.
- [14] Ann Barcomb, Klaas-Jan Stol, Dirk Riehle, and Brian Fitzgerald. 2019. Why do episodic volunteers stay in FLOSS communities?. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*. IEEE, 948–959.
- [15] Ann Susan Barcomb. 2019. *Retaining and managing episodic contributors in free/libre/open source software communities*. Ph.D. Dissertation, University of Limerick.
- [16] Stefan Kambiz Behfar, Ekaterina Turkina, and Thierry Burger-Helmchen. 2018. Knowledge management in OSS communities: Relationship between dense and sparse network structures. *International Journal of Information Management* 38, 1 (2018), 167–174.
- [17] Lonni Besançon, Paul Issartel, Mehdi Ammi, and Tobias Isenberg. 2016. Hybrid tactile/tangible interaction for 3D data exploration. *IEEE transactions on visualization and computer graphics* 23, 1 (2016), 881–890.
- [18] Susanne Bødker, Christian Dindler, Ole S Iversen, and Rachel C Smith. 2022. What Is Participatory Design? In *Participatory Design*. Springer, 5–13.
- [19] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.
- [20] Virginia Braun and Victoria Clarke. 2021. *Thematic analysis: A practical guide*. (2021).
- [21] Fabio Calefato, Marco Aurelio Gerosa, Giuseppe Iaffaldano, Filippo Lanubile, and Igor Steinmacher. 2022. Will you come back to contribute? Investigating the inactivity of OSS core developers in GitHub. *Empirical Software Engineering* 27, 3 (2022), 76.

- [22] Kevin Daniel André Carillo, Sid Huff, and Brenda Chawner. 2014. It’s not only about writing code: an investigation of the notion of citizenship behaviors in the context of Free/Libre/Open source software communities. In *2014 47th Hawaii International Conference on System Sciences*. IEEE, 3276–3285.
- [23] Elian Carsenat. 2019. Inferring gender from names in any region, language, or alphabet. doi:10.13140/RG.2.2.11516.90247
- [24] Chameleon. 2025. SaaS Welcome Emails: How to Make a Great First Impression. <https://www.chameleon.io/blog/saas-welcome-emails>. Accessed: 2025-01-27.
- [25] Jinghui Cheng and Jin LC Guo. 2019. Activity-based analysis of open source software contributors: Roles and dynamics. In *2019 IEEE/ACM 12th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*. IEEE, 11–18.
- [26] Cloud Native Computing Foundation. Accessed: 2024-06-01. CNCF DevStats. <https://devstats.cncf.io/>.
- [27] Bernard F Cole, Richard D Gelber, and Aron Goldhirsch. 1993. Cox regression models for quality adjusted survival analysis. *Statistics in Medicine* 12, 10 (1993), 975–987.
- [28] CHAOSS community. [n. d.]. CHAOSS. <https://github.com/chaoss>. [Online; accessed 2022-09-17].
- [29] Sunny Consolvo, Beverly Harrison, Ian Smith, Mike Y Chen, Katherine Everitt, Jon Froehlich, and James A Landay. 2007. Conducting in situ evaluations for and with ubiquitous computing technologies. *International Journal of Human-Computer Interaction* 22, 1-2 (2007), 103–118.
- [30] John W Creswell and Cheryl N Poth. 2016. *Qualitative inquiry and research design: Choosing among five approaches*. Sage publications.
- [31] Tiago Silva Da Silva, Angela Martin, Frank Maurer, and Milene Silveira. 2011. User-centered design and agile methods: a systematic review. In *2011 AGILE conference*. IEEE, 77–86.
- [32] 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*. 1277–1286.
- [33] Suman Kumar Dawn and Suparna Biswas. 2013. Talent Management: Designing Strategies. *ASBM Journal of Management* 6, 2 (2013).
- [34] Steve Easterbrook, Janice Singer, Margaret-Anne Storey, and Daniela Damian. 2008. Selecting empirical methods for software engineering research. *Guide to advanced empirical software engineering* (2008), 285–311.
- [35] Vijaya Kumar Eluri, Thomas A Mazzuchi, and Shahram Sarkani. 2021. Predicting long-time contributors for GitHub projects using machine learning. *Information and Software Technology* 138 (2021), 106616.
- [36] Emelie Engström, Margaret-Anne Storey, Per Runeson, Martin Höst, and Maria Teresa Baldassarre. 2020. How software engineering research aligns with design science: a review. *Empirical Software Engineering* 25 (2020), 2630–2660.
- [37] David Faraggi and Richard Simon. 1995. A neural network model for survival data. *Statistics in medicine* 14, 1 (1995), 73–82.
- [38] Fedora People. 2024. Fedora People. Website. <https://fedorapeople.org> Accessed: 2024-06-11.
- [39] Feedly. 2024. Feedly. Website. <https://feedly.com> Accessed: 2024-06-11.
- [40] Zixuan Feng. 2023. The State of Survival in OSS: The Impact of Diversity. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 2213–2215.
- [41] Zixuan Feng. 2025. *Addressing OSS Community Managers’ Challenges in Contributor Retention – Appendix*. doi:10.5281/zenodo.14902992
- [42] Zixuan Feng, Amreeta Chatterjee, Anita Sarma, and Iftekhar Ahmed. 2022. A case study of implicit mentoring, its prevalence, and impact in apache. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 797–809.
- [43] Zixuan Feng, Mariam Guizani, Marco A Gerosa, and Anita Sarma. 2023. The state of diversity and inclusion in apache: A pulse check. In *2023 IEEE/ACM 16th International Conference on Cooperative and Human Aspects of Software Engineering (CHASE)*. IEEE, 150–160.
- [44] Zixuan Feng, Katie Kimura, Bianca Trinkenreich, Anita Sarma, and Igor Steinmacher. 2024. Guiding the way: A systematic literature review on mentoring practices in open source software projects. *Information and Software Technology* (2024), 107470.
- [45] Zixuan Feng, Igor Steinmacher, Marco Gerosa, Tyler Menezes, Alexander Serebrenik, Reed Milewicz, and Anita Sarma. 2025. The Multifaceted Nature of Mentoring in OSS: Strategies, Qualities, and Ideal Outcomes. *arXiv preprint arXiv:2501.05600* (2025).
- [46] Maria Angela Ferrario, Will Simm, Peter Newman, Stephen Forshaw, and Jon Whittle. 2014. Software engineering for ‘social good’: integrating action research, participatory design, and agile development. In *Companion Proceedings of the 36th ICSE*. 520–523.
- [47] Stephen Few. 2006. *Information dashboard design: The effective visual communication of data*. O’Reilly Media, Inc.
- [48] Uwe Flick. 2004. Triangulation in qualitative research. *A companion to qualitative research* 3 (2004), 178–183.
- [49] Jane H. Forman and Laura J. Damschroder. 2007. *Qualitative Content Analysis*.
- [50] Armstrong Foundjem, Ellis Eghan, and Bram Adams. 2021. Onboarding vs. diversity, productivity and quality—empirical study of the openstack ecosystem. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. IEEE, 1033–1045.
- [51] Adrian Furnham. 1986. Response bias, social desirability and dissimulation. *Personality and individual differences* 7, 3 (1986), 385–400.
- [52] Grigori Fursin, Anton Lokhmotov, and Ed Plowman. 2016. Collective knowledge: Towards R&D sustainability. In *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 864–869.
- [53] Vahid Garousi, Michael Felderer, and Mika V Mäntylä. 2019. Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. *Information and software technology* 106 (2019), 101–121.

- [54] D Randy Garrison, Martha Cleveland-Innes, Marguerite Koole, and James Kappelman. 2006. Revisiting methodological issues in transcript analysis: Negotiated coding and reliability. *The internet and higher education* 9, 1 (2006), 1–8.
- [55] Marco Gerosa, Igor Wiese, Bianca Trinkenreich, Georg Link, Gregorio Robles, Christoph Treude, Igor Steinmacher, and Anita Sarma. 2021. The shifting sands of motivation: Revisiting what drives contributors in open source. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. IEEE, 1046–1058.
- [56] Luiz Giovanini, Daniela Oliveira, Huascar Sanchez, and Deborah Shands. 2021. Leveraging Team Dynamics to Predict Open-source Software Projects' Susceptibility to Social Engineering Attacks. *arXiv preprint arXiv:2106.16067* (2021).
- [57] Barney G Glaser. 2016. Open coding descriptions. *Grounded theory review* 15, 2 (2016), 108–110.
- [58] Daniel Gooch, Laura Benton, Rilla Khaled, Dominik Lukeš, and Asimina Vasalou. 2016. Creating bridges: the role of exploratory design research in an intelligent tutoring system project. *Interacting with Computers* 28, 3 (2016), 372–386.
- [59] Google Open Source. 2024. Google Open Source Blog. Blog. <https://opensource.googleblog.com> Accessed: 2024-06-11.
- [60] Philip Gray. 2022. To disengage or not to disengage: a look at contributor disengagement in open source software. In *Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Companion Proceedings*. 328–330.
- [61] Greg Guest, Arwen Bunce, and Laura Johnson. 2006. How many interviews are enough? An experiment with data saturation and variability. *Field methods* 18, 1 (2006), 59–82.
- [62] Mariam Guizani, Aileen Abril Castro-Guzman, Anita Sarma, and Igor Steinmacher. 2023. Rules of engagement: Why and how companies participate in OSS. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*. IEEE, 2617–2629.
- [63] Mariam Guizani, Zixuan Feng, Emily Arteaga, Katie Kimura, Diane Mueller, Luis Cañas Díaz, Alexander Serebrenik, and Anita Sarma. 2025. Community Tapestry: An actionable tool to track turnover and diversity in OSS. *Information and Software Technology* (2025), 107871.
- [64] Mariam Guizani, Thomas Zimmermann, Anita Sarma, and Denae Ford. 2022. Attracting and retaining oss contributors with a maintainer dashboard. In *Proceedings of the 2022 ACM/IEEE 44th International Conference on Software Engineering: Software Engineering in Society*. 36–40.
- [65] Shenyang Guo. 2010. *Survival analysis*. Oxford University Press.
- [66] Philipp Hacker, Andreas Engel, and Marco Mauer. 2023. Regulating ChatGPT and other large generative AI models. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*. 1112–1123.
- [67] Giuseppe Iaffaldano, Igor Steinmacher, Fabio Calefato, Marco Gerosa, and Filippo Lanubile. 2019. Why do developers take breaks from contributing to OSS projects? A preliminary analysis. *arXiv preprint arXiv:1903.09528* (2019).
- [68] Magid Igbaria and Jack Baroudi. 1995. The impact of job performance evaluation on career advancement prospects: An examination of gender differences in the is workplace. *Management Information Systems Quarterly* 19, 1 (1995), 5.
- [69] Aftab Iqbal. 2014. Understanding contributor to developer turnover patterns in oss projects: a case study of apache projects. *International Scholarly Research Notices* 2014, 1 (2014), 535724.
- [70] Hemant Ishwaran, Udaya B Kogalur, Eugene H Blackstone, and Michael S Lauer. 2008. Random survival forests. (2008).
- [71] Eirini Kalliamvakou, Christian Bird, Thomas Zimmermann, Andrew Begel, Robert DeLine, and Daniel M German. 2017. What makes a great manager of software engineers? *IEEE Transactions on Software Engineering* 45, 1 (2017), 87–106.
- [72] Barbara Kitchenham and Pearl Brereton. 2013. A systematic review of systematic review process research in software engineering. *Information and software technology* 55, 12 (2013), 2049–2075.
- [73] Robert E Kraut and Paul Resnick. 2012. *Building successful online communities: Evidence-based social design*. Mit Press.
- [74] Sari Kujala, Talya Miron-Shatz, and Jussi J Jokinen. 2019. The cross-sequential approach: a short-term method for studying long-term user experience. *Journal of Usability Studies* 14, 2 (2019), 105–116.
- [75] Sourabh Kumar. 2022. The impact of talent management practices on employee turnover and retention intentions. *Global Business and Organizational Excellence* 41, 2 (2022), 21–34.
- [76] Heidi Lam, Enrico Bertini, Petra Isenberg, Catherine Plaisant, and Sheelagh Carpendale. 2011. Empirical studies in information visualization: Seven scenarios. *IEEE transactions on visualization and computer graphics* 18, 9 (2011), 1520–1536.
- [77] Susan Lechelt, Frederik Brudy, Nicolai Marquardt, and Yvonne Rogers. 2021. Evalme: Exploring the value of new technologies for in situ evaluation of learning experiences. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–14.
- [78] Amanda Lee. 2018. One-time contributors to FLOSS: surveys and data analysis. *ACM SIGSOFT Software Engineering Notes* 43, 1 (2018), 1–6.
- [79] Elisa T Lee and Oscar T Go. 1997. Survival analysis in public health research. *Annual review of public health* 18, 1 (1997), 105–134.
- [80] Zhixing Li, Yue Yu, Tao Wang, Gang Yin, Shanshan Li, and Huaimin Wang. 2021. Are you still working on this? An empirical study on pull request abandonment. *IEEE Transactions on Software Engineering* 48, 6 (2021), 2173–2188.
- [81] Jenny T Liang, Thomas Zimmermann, and Denae Ford. 2022. Understanding skills for oss communities on github. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 170–182.
- [82] Bin Lin, Gregorio Robles, and Alexander Serebrenik. 2017. Developer turnover in global, industrial open source projects: Insights from applying survival analysis. In *2017 IEEE 12th International Conference on Global Software Engineering (ICGSE)*. IEEE, 66–75.

- [83] Bin Lin and Alexander Serebrenik. 2016. Recognizing gender of stack overflow users. In *Proceedings of the 13th international conference on mining software repositories*. 425–429.
- [84] Yong-Han Lin, Li-Ting Su, Uei-Dar Chen, Peng-Jui Wang, Yi-Chi Lee, and Yung-Ju Chang. 2023. Automatic, Manual, or Hybrid? A Preliminary Investigation of Users’ Perception of Features for Supporting Notification Management. In *Adjunct Proceedings of the 2023 ACM International Joint Conference on Pervasive and Ubiquitous Computing & the 2023 ACM International Symposium on Wearable Computing*. 98–102.
- [85] Linux Foundation. 2024. KubeCon + CloudNativeCon North America 2024. <https://events.linuxfoundation.org/kubecon-cloudnativecon-north-america/> Accessed: 2025-02-20.
- [86] Linux Plumbers Conference. 2025. Linux Plumbers Conference. <https://lpc.events/> Accessed: 2025-02-20.
- [87] Linux.com. 2024. Linux.com. Website. <https://www.linux.com> Accessed: 2024-06-11.
- [88] Jwen Fai Low, Tennom Yathog, and Davor Svetinovic. 2015. Software analytics study of Open-Source system survivability through social contagion. In *2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. IEEE, 1213–1217.
- [89] Microsoft. 2024. DeepSpeed. <https://github.com/microsoft/DeepSpeed>. Accessed: 2024-09-24.
- [90] Vishal Midha and Prashant Palvia. 2007. Retention and quality in open source software projects. (2007).
- [91] Courtney Miller, David Gray Widder, Christian Kästner, and Bogdan Vasilescu. 2019. Why do people give up flossing? a study of contributor disengagement in open source. In *Open Source Systems: 15th IFIP WG 2.13 International Conference, OSS 2019, Montreal, QC, Canada, May 26–27, 2019, Proceedings 15*. Springer, 116–129.
- [92] Mathieu Nassif and Martin P Robillard. 2017. Revisiting turnover-induced knowledge loss in software projects. In *2017 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 261–272.
- [93] Lynn R Offermann and Tessa E Basford. 2013. Inclusive human resource management. *Diversity at work: The practice of inclusion* (2013), 229–259.
- [94] Open Source Initiative. 2024. Open Source Initiative Blog. Blog. <https://opensource.org/blog> Accessed: 2024-06-11.
- [95] OpenUK. 2025. OpenUK - The UK Organisation for Open Technology. <https://openuk.uk/> Accessed: 2025-02-20.
- [96] Christian Payne. 2002. On the security of open source software. *Information systems journal* 12, 1 (2002), 61–78.
- [97] Jack Phillips and Patricia Phillips. 2015. *High-impact human capital strategy: Addressing the 12 major challenges today’s organizations face*. Amacom.
- [98] Jennifer Preece and Ben Shneiderman. 2009. The reader-to-leader framework: Motivating technology-mediated social participation. *AIS transactions on human-computer interaction* 1, 1 (2009), 13–32.
- [99] Huilian Sophie Qiu, Anna Lieb, Jennifer Chou, Megan Carneal, Jasmine Mok, Emily Amspoker, Bogdan Vasilescu, and Laura Dabbish. 2023. Climate Coach: A Dashboard for Open-Source Maintainers to Overview Community Dynamics. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–18.
- [100] Huilian Sophie Qiu, Alexander Nolte, Anita Brown, Alexander Serebrenik, and Bogdan Vasilescu. 2019. Going farther together: The impact of social capital on sustained participation in open source. In *2019 IEEE/ACM 41st international conference on software engineering (icse)*. IEEE, 688–699.
- [101] Naveen Raman, Minxuan Cao, Yulia Tsvetkov, Christian Kästner, and Bogdan Vasilescu. 2020. Stress and burnout in open source: Toward finding, understanding, and mitigating unhealthy interactions. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: New Ideas and Emerging Results*. 57–60.
- [102] Anirudh Ramchandran, Likang Yin, and Vladimir Filkov. 2022. Exploring Apache incubator project trajectories with APEX. In *Proceedings of the 19th international conference on mining software repositories*. 333–337.
- [103] Mehvish Rashid, Paul M Clarke, and Rory V O’Connor. 2017. Exploring knowledge loss in open source software (OSS) projects. In *Software Process Improvement and Capability Determination: 17th International Conference, SPICE 2017, Palma de Mallorca, Spain, October 4–5, 2017, Proceedings*. Springer, 481–495.
- [104] Tanya Reilly. n.d.. Being Glue. <https://www.noidea.dog/glue>. Accessed: 2024-06-01.
- [105] Sergio Rico. 2020. Exploring and improving industry-academia communication in software engineering. In *Proceedings of the 24th International Conference on Evaluation and Assessment in Software Engineering*. 379–382.
- [106] Derek Robinson, Keanelek Enns, Neha Koulekar, and Manish Sihag. 2022. Two approaches to survival analysis of open source Python projects. In *Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension*. 660–669.
- [107] Gema Rodríguez-Pérez, Reza Nadri, and Meiyappan Nagappan. 2021. Perceived diversity in software engineering: a systematic literature review. *Empirical Software Engineering* 26 (2021), 1–38.
- [108] Per Runeson, Emelie Engström, and Margaret-Anne Storey. 2020. The design science paradigm as a frame for empirical software engineering. In *Contemporary empirical methods in software engineering*. Springer, 127–147.
- [109] Andreas Schilling. 2014. What do we know about floss developers’ attraction, retention, and commitment? a literature review. In *2014 47th Hawaii International Conference on System Sciences*. IEEE, 4003–4012.
- [110] Andreas Schilling, Sven Laumer, and Tim Weitzel. 2012. Who will remain? an evaluation of actual person-job and person-team fit to predict developer retention in floss projects. In *2012 45th Hawaii international conference on system sciences*. IEEE, 3446–3455.

- [111] Ezequiel Scott, Khaled Nimr Charkie, and Dietmar Pfahl. 2020. Productivity, turnover, and team stability of agile teams in open-source software projects. In *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, 124–131.
- [112] Favouate Y Sebele-Mpofu. 2020. Saturation controversy in qualitative research: Complexities and underlying assumptions. A literature review. *Cogent Social Sciences* 6, 1 (2020), 1838706.
- [113] Paul Sebo. 2021. Performance of gender detection tools: a comparative study of name-to-gender inference services. *Journal of the Medical Library Association: JMLA* 109, 3 (2021), 414.
- [114] Pratyush N Sharma, John Hulland, and Sherae Daniel. 2012. Examining turnover in open source software projects using logistic hierarchical linear modeling approach. In *Open Source Systems: Long-Term Sustainability: 8th IFIP WG 2.13 International Conference, OSS 2012, Hammamet, Tunisia, September 10-13, 2012. Proceedings* 8. Springer, 331–337.
- [115] Lucy Simko, Britnie Chin, Sungmin Na, Harkiran Kaur Saluja, Tian Qi Zhu, Tadayoshi Kohno, Alexis Hiniker, Jason Yip, and Camille Cobb. 2021. Would you rather: A focus group method for eliciting and discussing formative design insights with children. In *Proceedings of the 20th Annual ACM Interaction Design and Children Conference*. 131–146.
- [116] Software Freedom Conservancy. 2024. Free and Open Source Software Yearly (FOSSY). <https://sfconservancy.org/fossy/> Accessed: 2025-02-20.
- [117] Software Freedom Conservancy. 2024. Software Freedom Conservancy Blog. [Blog. https://sfconservancy.org/blog/](https://sfconservancy.org/blog/) Accessed: 2024-06-11.
- [118] Torrence E Sparkman. 2019. Exploring the boundaries of diversity and inclusion in human resource development. *Human Resource Development Review* 18, 2 (2019), 173–195.
- [119] Igor Steinmacher, Tayana Conte, Marco Aurélio Gerosa, and David Redmiles. 2015. Social barriers faced by newcomers placing their first contribution in open source software projects. In *Proceedings of the 18th ACM conference on Computer supported cooperative work & social computing*. 1379–1392.
- [120] Igor Steinmacher, Tayana Uchoa Conte, Christoph Treude, and Marco Aurélio Gerosa. 2016. Overcoming open source project entry barriers with a portal for newcomers. In *Proceedings of the 38th International Conference on Software Engineering*. 273–284.
- [121] Igor Steinmacher, Marco Aurélio Gerosa, and David Redmiles. 2014. Attracting, onboarding, and retaining newcomer developers in open source software projects. In *Workshop on Global Software Development in a CSCW Perspective*, Vol. 16. 20.
- [122] Igor Steinmacher, Christoph Treude, and Marco Aurelio Gerosa. 2018. Let me in: Guidelines for the successful onboarding of newcomers to open source projects. *IEEE Software* 36, 4 (2018), 41–49.
- [123] Igor Steinmacher, Igor Wiese, Ana Paula Chaves, and Marco Aurélio Gerosa. 2013. Why do newcomers abandon open source software projects?. In *2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*. IEEE, 25–32.
- [124] Igor Steinmacher, Igor Wiese, Tayana Uchoa Conte, and Marco Aurelio Gerosa. 2015. Increasing the self-efficacy of newcomers to Open Source Software projects. In *2015 29th Brazilian Symposium on Software Engineering*. IEEE, 160–169.
- [125] Margaret-Anne Storey, Rashina Hoda, Alessandra Maciel Paz Milani, and Maria Teresa Baldassarre. 2024. Guiding Principles for Using Mixed Methods Research in Software Engineering. *arXiv preprint arXiv:2404.06011* (2024).
- [126] Pyomo Development Team. 2024. Pyomo: Python Optimization Modeling Objects. <https://github.com/Pyomo/pyomo>. Accessed: 2024-09-24.
- [127] Mike Thelwall. 2011. Privacy and gender in the social web. In *Privacy online: Perspectives on privacy and self-disclosure in the social web*. Springer, 251–265.
- [128] Keith Townsend. 2013. Saturation and run off: How many interviews are required in qualitative research. In *ANZAM Conference 2013*. Australian and New Zealand Academy of Management, 1–17.
- [129] Bianca Trinkenreich. 2021. Please Don't Go—A Comprehensive Approach to Increase Women's Participation in Open Source Software. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*. IEEE, 293–298.
- [130] Bianca Trinkenreich, Mariam Guizani, Igor Wiese, Anita Sarma, and Igor Steinmacher. 2020. Hidden figures: Roles and pathways of successful oss contributors. *Proceedings of the ACM on human-computer interaction* 4, CSCW2 (2020), 1–22.
- [131] Bianca Trinkenreich, Klaas-Jan Stol, Anita Sarma, Daniel M German, Marco A Gerosa, and Igor Steinmacher. 2023. Do i belong? modeling sense of virtual community among linux kernel contributors. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*. IEEE, 319–331.
- [132] Bianca Trinkenreich, Igor Wiese, Anita Sarma, Marco Gerosa, and Igor Steinmacher. 2022. Women's participation in open source software: A survey of the literature. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 31, 4 (2022), 1–37.
- [133] Jason Tsay, Laura Dabbish, and James Herbsleb. 2014. Influence of social and technical factors for evaluating contribution in GitHub. In *Proceedings of the 36th international conference on Software engineering*. 356–366.
- [134] Marat Valiev, Bogdan Vasilescu, and James Herbsleb. 2018. Ecosystem-level determinants of sustained activity in open-source projects: A case study of the PyPI ecosystem. In *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 644–655.
- [135] Bogdan Vasilescu, Andrea Capiluppi, and Alexander Serebrenik. 2014. Gender, representation and online participation: A quantitative study. *Interacting with Computers* 26, 5 (2014), 488–511.

- [136] Bogdan Vasilescu, Daryl Posnett, Baishakhi Ray, Mark GJ van den Brand, Alexander Serebrenik, Premkumar Devanbu, and Vladimir Filkov. 2015. Gender and tenure diversity in GitHub teams. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. 3789–3798.
- [137] Bogdan Vasilescu, Yue Yu, Huaimin Wang, Premkumar Devanbu, and Vladimir Filkov. 2015. Quality and productivity outcomes relating to continuous integration in GitHub. In *Proceedings of the 2015 10th joint meeting on foundations of software engineering*. 805–816.
- [138] Lev Velykoivanenko, Kavous Salehzadeh Niksirat, Stefan Teofanovic, Bertil Chapuis, Michelle L Mazurek, and Kévin Huguenin. 2024. Designing a Data-Driven Survey System: Leveraging Participants’ Online Data to Personalize Surveys. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–22.
- [139] Xiaofeng Wang and Zhenshun Cheng. 2020. Cross-sectional studies: strengths, weaknesses, and recommendations. *Chest* 158, 1 (2020), S65–S71.
- [140] Mairieli Wessel, Igor Wiese, Igor Steinmacher, and Marco Aurelio Gerosa. 2021. Don’t disturb me: Challenges of interacting with software bots on open source software projects. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (2021), 1–21.
- [141] Claes Wohlin. 2014. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering*. 1–10.
- [142] Wai Peng Wong, Hwee Chin Tan, Kim Hua Tan, and Ming-Lang Tseng. 2019. Human factors in information leakage: mitigation strategies for information sharing integrity. *Industrial Management & Data Systems* 119, 6 (2019), 1242–1267.
- [143] Kazuhiro Yamashita, Yasutaka Kamei, Shane McIntosh, Ahmed E Hassan, and Naoyasu Ubayashi. 2016. Magnet or sticky? Measuring project characteristics from the perspective of developer attraction and retention. *Journal of Information Processing* 24, 2 (2016), 339–348.
- [144] Yiqing Yu, Alexander Benlian, and Thomas Hess. 2012. An empirical study of volunteer members’ perceived turnover in open source software projects. In *2012 45th Hawaii International Conference on System Sciences*. IEEE, 3396–3405.
- [145] Yuxia Zhang, Hui Liu, Xin Tan, Minghui Zhou, Zhi Jin, and Jiaxin Zhu. 2022. Turnover of companies in OpenStack: Prevalence and rationale. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 31, 4 (2022), 1–24.
- [146] Yuxia Zhang, Klaas-Jan Stol, Hui Liu, and Minghui Zhou. 2022. Corporate dominance in open source ecosystems: a case study of OpenStack. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 1048–1060.
- [147] Yuxia Zhang, Minghui Zhou, Audris Mockus, and Zhi Jin. 2019. Companies’ Participation in OSS development—An empirical study of OpenStack. *IEEE Transactions on Software Engineering* 47, 10 (2019), 2242–2259.
- [148] Lu Zhen, George Q. Huang, and Zuhua Jiang. 2009. Recommender system based on workflow. *Decision Support Systems* 48, 1 (2009), 237–245. doi:10.1016/j.dss.2009.08.002 Information product markets.
- [149] Minghui Zhou and Audris Mockus. 2010. Developer fluency: achieving true mastery in software projects. In *Fast Software Encryption Workshop*. <https://api.semanticscholar.org/CorpusID:16825213>
- [150] Minghui Zhou and Audris Mockus. 2011. Does the initial environment impact the future of developers?. In *Proceedings of the 33rd International Conference on Software Engineering*. 271–280.
- [151] Minghui Zhou and Audris Mockus. 2012. What make long term contributors: Willingness and opportunity in OSS community. In *2012 34th International Conference on Software Engineering (ICSE)*. IEEE, 518–528.
- [152] Minghui Zhou and Audris Mockus. 2014. Who will stay in the floss community? modeling participant’s initial behavior. *IEEE Transactions on Software Engineering* 41, 1 (2014), 82–99.
- [153] Minghui Zhou, Audris Mockus, Xiujuan Ma, Lu Zhang, and Hong Mei. 2016. Inflow and retention in oss communities with commercial involvement: A case study of three hybrid projects. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 25, 2 (2016), 1–29.
- [154] Mengdie Zhuang, David Concannon, and Ed Manley. 2022. A framework for evaluating dashboards in healthcare. *IEEE Transactions on Visualization and Computer Graphics* 28, 4 (2022), 1715–1731.
- [155] Michael Zimmer. 2020. “But the data is already public”: on the ethics of research in Facebook. In *The ethics of information technologies*. Routledge, 229–241.

GRAY LITERATURE REFERENCES

- [G1] A. Graner, Making ubuntu more personal: Identify contributions to engage more personally, <https://ubuntu-news.org/2011/01/22/making-ubuntu-more-personal-identify-contributions-to-engage-more-personally/>, accessed: 2024-03-01 (2011).
- [G2] Linux Foundation, Linux foundation research reveals new open source diversity, equity, and inclusion trends, <https://www.linux.com/news/linux-foundation-research-reveals-new-open-source-diversity-equity-and-inclusion-trends/>, accessed: 2024-03-01 (2023).
- [G3] ZDNet, What linux kernel maintainers do and why they need your help, <https://www.zdnet.com/article/what-linux-kernel-maintainers-do-and-why-they-need-your-help/>, accessed: 2024-03-01 (2023).
- [G4] Google Open Source, Showing our work: Study in understanding open source contributors, <https://opensource.googleblog.com/2023/09/showing-our-work-study-in-understanding-open-source-contributors.html>, accessed: 2024-03-01 (2023).
- [G5] Open Source For U, Github launches initiative to help oss contributors and maintainers, <https://www.Opensourceforu.com/2019/01/github-launches-initiative-to-help-oss-contributors-and-maintainers/>, accessed: 2024-03-01 (2019).
- [G6] Open Source For U, Linux maintainer greg k-h talks what keeps him busy, <https://www.Opensourceforu.com/2013/12/linux-maintainer-greg-k-h-talks-keeps-busy/>, accessed: 2024-03-01 (2013).
- [G7] InfoWorld, What does an open source maintainer do after burnout?, <https://www.infoworld.com/article/3563326/what-does-an-open-source-maintainer-do-after-burnout.html>, accessed: 2024-03-01 (2020).
- [G8] Linux Foundation, Open source maintainers: What they need and how to support them, <https://www.linuxfoundation.org/blog/open-source-maintainers-what-they-need-and-how-to-support-them>, accessed: 2024-03-01 (2023).
- [G9] Reddit, Cost of maintaining open source projects, https://www.reddit.com/r/freesoftware/comments/15zzmtx/cost_of_maintaining_open_source_projects/, accessed: 2024-03-01 (2023).
- [G10] Reddit, Notes from a tired maintainer, https://www.reddit.com/r/opensource/comments/1amlpyc/notes_from_a_tired_maintainer/, accessed: 2024-03-01 (2023).
- [G11] Reddit, What it feels like to be an open-source maintainer, https://www.reddit.com/r/opensource/comments/10gyqot/what_it_feels_like_to_be_an_open_source_maintainer/, accessed: 2024-03-01 (2023).
- [G12] FOSS Force, Six years in, maintainer darrick wong says goodbye to xfs, <https://fossforce.com/2023/08/six-years-in-maintainer-darrick-wong-says-goodbye-to-xfs/>, accessed: 2024-03-01 (2023).
- [G13] Linux Journal, Linux kernel reduction: Longterm support, <https://www.linuxjournal.com/content/linux-kernel-reduction-longterm-support>, accessed: 2024-03-01 (2023).
- [G14] The Register, Linux kernel report, https://www.theregister.com/2023/09/26/linux_kernel_report_2, accessed: 2024-03-01 (2023).
- [G15] Open Source Blog, Anyone and everyone: Leslie hawthorn reflects on 20 years of open source, <https://blog.Opensource.org/anyone-and-everyone-leslie-hawthorn-reflects-on-20-years-of-open-source/>, accessed: 2024-03-01 (2023).
- [G16] Open Source For U, The benefits and drawbacks of oss, <https://www.Opensourceforu.com/2023/12/the-benefits-and-drawbacks-of-oss/>, accessed: 2024-03-01 (2023).
- [G17] Linux Careers, How to recruit and retain top linux and open source talent, <https://www.linuxcareers.com/resources/blog/2023/01/how-to-recruit-and-retain-top-linux-and-open-source-talent/>, accessed: 2024-03-01 (2023).
- [G18] Linux.com, Linux professionals hard to find, say hiring managers, <https://www.linux.com/topic/open-source/linux-professionals-hard-find-say-hiring-managers/>, accessed: 2024-03-01 (2023).
- [G19] Linux.com, Interview with shuah khan, kernel maintainer linux fellow, <https://www.linux.com/news/interview-with-shuah-khan-kernel-maintainer-linux-fellow/>, accessed: 2024-03-01 (2023).
- [G20] Linux.com, Maintainer confidential: Opportunities and challenges of the ubiquitous but under-resourced yocto project, <https://www.linux.com/audience/maintainer-confidential-opportunities-and-challenges-of-the-ubiquitous-but-under-resourced-yocto-project/>, accessed: 2024-03-01 (2023).
- [G21] Free Software Foundation, Nominee discussion forum guidelines for communicating with kindness, <https://www.fsf.org/about/nominee-discussion-forum-guidelines-for-communicating-with-kindness>, accessed: 2024-03-01 (2023).
- [G22] ZDNet, Microsoft told we're not happy by github contributors to open source .net core wpf, <https://www.zdnet.com/article/microsoft-told-were-not-happy-by-github-contributors-to-open-source-net-core-wpf/>, accessed: 2024-03-01 (2023).
- [G23] Reddit, Call for a new public facing validation metric, https://www.reddit.com/r/opensource/comments/14t3vxc/call_for_a_new_public_facing_validation_metric/, accessed: 2024-03-01 (2023).
- [G24] OpenSource.com, How to build an inclusive community with recognition, <https://opensource.com/article/22/5/inclusive-community-recognition>, accessed: 2024-03-01 (2022).
- [G25] OpenSource.com, Octohat: Github non-code contribution tracker, <https://opensource.com/life/15/10/octohat-github-non-code-contribution-tracker>, accessed: 2024-03-01 (2015).
- [G26] Linux foundation research reveals new open source diversity, equity, and inclusion trends, <https://www.linux.com/news/linux-foundation-research-reveals-new-open-source-diversity-equity-and-inclusion-trends/>, accessed: 2024-03-01.
- [G27] Dei open source metrics, <https://opensource.net/dei-open-source-metrics/>, accessed: 2024-03-01.

- [G28] Measure open source project health, <https://opensource.net/measure-open-source-project-health/>, accessed: 2024-03-01.
- [G29] Making ubuntu more personal: Identify contributions to engage more personally, <https://ubuntu-news.org/2011/01/22/making-ubuntu-more-personal-identify-contributions-to-engage-more-personally/>, accessed: 2024-03-01.
- [G30] How to replace an 'irreplaceable' employee, <https://www.infoworld.com/article/2625687/how-to-replace-an-irreplaceable-employee.html>, accessed: 2024-03-01.
- [G31] Will the solo open source developer survive the pandemic?, <https://www.infoworld.com/article/3543796/will-the-solo-open-source-developer-survive-the-pandemic.html>, accessed: 2024-03-01.
- [G32] Dei open source metrics, <https://opensource.net/dei-open-source-metrics/>, accessed: 2024-03-01.
- [G33] Linux kernel reduction: Longterm support, <https://www.linuxjournal.com/content/linux-kernel-reduction-longterm-support>, accessed: 2024-03-01.
- [G34] Open source maintainers: What they need and how to support them, <https://www.linuxfoundation.org/blog/open-source-maintainers-what-they-need-and-how-to-support-them>, accessed: 2024-03-01.
- [G35] How to maintain a big free open source project, https://www.reddit.com/r/opensource/comments/16q15w5/how_to_maintain_a_big_free_open_source_project/, accessed: 2024-03-01.