The Nuts and Bolts of Open Source: A Taxonomy of Glue Work in OSS Projects

August 15, 2025

Abstract

The success and sustainability of open source software (OSS) projects hinge not only on visible technical contributions but also on GLUE WORK—the often invisible and underappreciated efforts that hold projects and communities together. Despite their importance, GLUE WORK remains inconsistently defined, difficult to trace, and rarely acknowledged. To address this gap, we conducted a multiple case study of OSS projects, using interviews, focus groups, and surveys across diverse contexts. From this empirical foundation, we inductively developed a taxonomic theory of GLUE WORK, grounded in the theory of invisible labor and organized around three analytical lenses: sociocultural (what GLUE WORK is), sociospatial (where it can be traced), and sociolegal (how it can be acknowledged). We identify four overarching categories—Code & Technical Management, Mentoring & Support, Documentation, and Community Management—comprising 12 types of GLUE WORK. These types of work are traceable via various platforms such as version control and issue tracking systems (e.g., GitHub) and social media (e.g., LinkedIn). They can be acknowledged through community announcements, project documentation, and interpersonal communication. Our theory offers a systematic framework for understanding, tracing, and recognizing GLUE WORK—ultimately helping OSS communities build more inclusive and sustainable ecosystems.

1 Introduction

Would a software development project be successful if everyone on the team spent their workday working on feature development? No, it would not. Developing and maintaining a large-scale software system requires a gamut of efforts, some of which are visible, like feature development or code commits, others, like onboarding newcomers, responding to user queries, closing duplicate bugs, or helping peers, are less so, but equally important. We call this GLUE WORK (Reilly, 2019).

In open source software (OSS), contributors, including some volunteers and others employed, organically come together to build and maintain large, complex, intellectual products (Feng, 2023). Traditionally, feature development contributions have been valued as a means to showcase project progress (Zhou & Mockus, 2012). In contrast, other types of aforementioned activities that form the "glue" that holds a project together are often unappreciated (Young et al., 2021). Even code-related efforts such as writing maintenance code, refactoring code, and performing quality assurance frequently fail to receive equal recognition and acknowledgment as feature development (Geiger, Howard, & Irani, 2021).

The lack of visibility and acknowledgment of GLUE WORK aligns with broader discussions on invisible labor (Poster, Crain, & Cherry, 2016) in sociology, where contributions that lack formal status or quantifiable metrics often go unnoticed (Stohl, Stohl, & Leonardi, 2016), such that contributors are not rewarded in society. In OSS, this "fairness" problem manifests when those doing GLUE WORK are not promoted on par with "awesome coders" or given the opportunity to participate in technical discussions (Young et al., 2021).

How to reward GLUE WORK is a nontrivial problem. Some OSS community initiatives have begun efforts to track GLUE WORK, such as the *All-contributors bot* (*all-contributors*, 2024), which enables maintainers and contributors to recognize different types of contributions through self-nomination or maintainer acknowledgment. However, such an approach is limited to self-reporting. Young et al. (2021) found in their study that among the 14,191 GitHub repositories that include an *All-Contributors.src* file, only about 20% intentionally adopted the bot but had minimal usage, leaving the persistent use of the tool and its broader impact uncertain.

This raises an important question: why does GLUE WORK remain largely invisible and underacknowledged in OSS communities? Despite its importance, few studies have systematically examined GLUE WORK in OSS in a comprehensive and structured way. Without such understanding, this labor will continue to be undervalued, and contributors performing OSS GLUE WORK will be uncertain about engaging with projects.

To address this gap, we adopt the framework of invisible labor from Hatton (2017) as an analytical lens to systematically investigate GLUE WORK in OSS. This framework identifies three interrelated dimensions through which certain forms of labor become systematically devalued. *Socioculturally*, the hacker mentality often measures success through a unidimensional, code-centric lens, leaving many contributions that keep projects viable unnamed and invisible (Trinkenreich et al., 2021). *Sociospatially*, platform-centric workplaces highlight easily visible activities while obscuring many contributions buried in less visible tasks or external channels (Feng, Chatterjee, Sarma, & Ahmed, 2022). *Sociolegally*, communities lack shared norms and governance procedures for fairly acknowledging GLUE WORK, which raises issues of fairness due to inconsistent or absent acknowledgment practices and ultimately undermines the long-term sustainability of OSS projects (Trinkenreich, Guizani, Wiese, Sarma, & Steinmacher, 2020). Guided by this theory, we investigate the following three research questions:

RQ1. What types of GLUE WORK occur in OSS, and how can they be characterized?

RQ2. Where does GLUE WORK reside within OSS practices and infrastructures?

RQ3. How and where can GLUE WORK be acknowledged?

To answer these questions, we conducted a multiple case study of GLUE WORK across OSS projects operating under diverse governance models. Our cases span foundation-led communities, company-sponsored ecosystems, and scientific OSS projects. This variation allows us to ensure the generalizability of our findings while capturing the nuanced ways GLUE WORK manifests and is treated across contexts.

The contribution of this study is threefold. First, to our knowledge, this paper presents the first empirically grounded and theoretically informed taxonomy of GLUE WORK in OSS.

Second, we identify four main categories—Code & Technical Management, Mentoring & Support, Documentation, and Community Management—comprising 12 types of GLUE WORK. Third, we provide insights into how GLUE WORK can be tracked, acknowledged, and rewarded. Our theory offers a systematic framework for understanding, monitoring, and valuing GLUE WORK, thus helping create more inclusive and sustainable OSS communities. We have summarized our findings in a companion website (*https://gluework.netlify.app/*) that provides practical strategies for managing OSS projects.

2 Conceptual Background

Within social theory, labor is often understood as the tasks individuals perform as part of their work. More expansively, the labor process encompasses the broader context of work, including the sequencing of tasks within a production system and the function of a job within an organizational structure (Crain, Poster, & Cherry, 2016). While not all tasks within the labor process are invisible, some forms of work remain unrecognized or unacknowledged. Invisible labor, as first conceptualized by Daniels (1987), refers to work that is culturally and economically devalued, under-acknowledged, or unrecognized, despite its critical role in sustaining communities (Crain et al., 2016).

Such labor is often obscured by societal norms that prioritize visible, measurable outcomes (ILO, 2012). Hatton (2017) developed a framework for understanding how invisible labor becomes economically devalued through three interrelated dimensions: sociocultural, where cultural ideologies diminish the perceived value of certain types of labor; sociolegal, where labor is excluded from formal recognition or compensation structures due to legal and policy definitions; and sociospatial, where work is performed in locations (physical or digital) that are peripheral to or disconnected from recognized centers of productivity and visibility. Together, these three dimensions contribute to the phenomenon of GLUE WORK in OSS. Table 1 maps Hatton (2017) dimensions to our analysis of GLUE WORK in OSS.

Table 1: Dimensions of Invisible Labor in OSS and Their Consequences

Mechanisms	Analytical Focus	Rationale
Sociocultural	What it is: Clear scope and terminology	OSS feature-centric culture caused many contributions to remain systematically unseen.
Sociospatial	Where to track: Data sources and instrumentation	Many contributions occur off centric collaboration or buried in many platforms. As a result, the project cannot "see" this labour.
Sociolegal	How to acknowledge: Recognition logic and fairness rationale	Many contributors are excluded from project metrics and reward systems because their work falls outside project policies, governance structures, and community norms, leaving no agreed-upon mechanisms for acknowledgment.

2.1 Defining What Counts: Sociocultural Blind Spots in OSS Labor.

In OSS communities, cultural ideologies such as the hacker mentality, which prioritizes new feature development (Trinkenreich et al., 2020), shape what is valued and what is overlooked. For example, contributions not directly tied to new feature development are often marginalized, despite being essential to the project's long-term health. Mentoring in OSS, including both formal and informal forms, contributes far beyond technical advice, as it not only supports task completion but also helps individuals navigate the social and cultural momentum of the OSS community (Feng, Kimura, Trinkenreich, Sarma, & Steinmacher, 2024). At the extreme, contributors may be unsure what constitutes a contribution and whether it is worth mentioning (e.g., refactoring legacy code, updating documentation) (Zhou & Mockus, 2012). Likewise, coordination and communication roles improve developer satisfaction and project cohesion by fostering smoother interactions among contributors and across organizational boundaries (Guzzi, Bacchelli, Lanza, Pinzger, & Van Deursen, 2013). As a result, these contributions remain naturalized as everyday maintenance, but unnamed and unvalued.

2.2 Out of Sight: Sociospatial Barriers to Recognizing OSS Contributions.

Sociospatial mechanisms explain that many forms of labor are physically separated from centralized workplaces (Daniels, 1987). Many contributions in OSS do not take place within centralized collaboration platforms like GitHub. Many contributions, such as end-user support, advocacy, and mentoring, often occur outside these platforms or are obscured by technical jargon (Squire, 2015). Although these activities require significant time and effort, they are rarely tracked or acknowledged because they fall outside the boundaries of existing tooling and visibility mechanisms (Smirnova, Reitzig, & Alexy, 2022).

2.3 Who Gets Credited? Sociolegal Exclusion and the Invisibility of OSS Glue Work.

OSS operates as a volunteer-driven ecosystem, amplifying the under-recognition of GLUE WORK contributors compared to "heroic coders" (Young et al., 2021). Many contributions, such as advocacy, license management, and community engagement, are critical to project sustainability but often lack fair acknowledgment due to the absence of governance structures, inadequate recognition mechanisms, and cultural biases favoring code-centric contributions (Trinkenreich et al., 2020).

While tools like the All-Contributors bot (*all-contributors*, 2024) aim to address this by enabling nomination of varied roles (e.g., documentation, mentoring), adoption remains extremely low. In 2021, among more than 200 million repositories—including at least 52 million public repositories in 2020 (Rad, 2021)—only 14,191 GitHub repositories contained an *All-Contributors.src* file. Only 20% of those made minimal use of it, reflecting a lack of community consensus on what constitutes valuable GLUE WORK and how to recognize it. This is compounded by the absence of robust mechanisms to track and credit systematically GLUE WORK, leaving contributors uncertain about the legitimacy of their efforts or those of others, particularly when contributions deviate from culturally celebrated coding norms. Without clear governance or guidelines for equitably acknowledging GLUE WORK, these contributions

will remain invisible, undermining contributor motivation, reducing project inclusivity, and ultimately harming the project's sustainability.

3 Developing a Taxonomic Theory of GLUE WORK in OSS.

In this section, we provide an overview of the study design and introduce a high-level structure of our taxonomic theory of GLUE WORK in OSS.

3.1 Study Design Overview.

Methodological Approach. To answer our research questions, we conducted multiple case studies adopting an interpretive lens to understand how GLUE WORK is enacted and framed in practice.

The multiple case studies have been widely used in information systems (IS) Research (Maruping & Matook, 2020; Bandara, Gable, & Rosemann, 2005; Slaughter, Levine, Ramesh, Pries-Heje, & Baskerville, 2006) due to their strength in uncovering emergent dynamics across different organizational settings. It is particularly well-suited for answering "how" and "why" questions about contemporary phenomena situated in real-life contexts, especially when the phenomenon and its context are deeply intertwined and not easily separable (Gregor, 2006).

Case Selection. To guide our use of the multiple case study method, we adopt an interpretive perspective that views reality as socially constructed through situated interactions and collective sensemaking (Walsham, 1995). This perspective aligns with our research goal: to understand how contributors and stakeholders in OSS communities interpret, negotiate, and assign meaning to forms of GLUE WORK that may otherwise be invisible within formal contribution models. It also shapes how we approach the data itself—treating organizational discourse, visual and textual artifacts, and interviews not as neutral facts but as meaning-laden expressions shaped by both participant framing and our interpretive lens as researchers (Klein & Myers, 1999).

From this interpretive stance, contextual variation is viewed as a necessary source of insight, enabling researchers to uncover how meanings are shaped by different socio-technical

Table 2: Overview of Case Sites

	OSSNA 2024	FOSSAsia 2024	Microsoft Open Source	CURIOSS
Background	A premier vendor neutral conference hosted by the Linux Foundation for open source developers and technologists to collaborate, share, and learn about the latest open source innovations.	A major open source event in Asia for developers, students, and startups to collaborate on open source projects and build businesses.	Microsoft's initiative, managed by its Open Source Programs Office (OSPO), to actively contribute to and support open source ecosystems through projects, sponsorships like the FOSS Fund, and memberships in foundations such as the Linux Foundation.	CURIOSS is a community for University and Research Institution Open Source Program Offices (OSPOs), aimed at facilitating networking, collaboration, and resource sharing among OSPO representatives worldwide to enhance open source initiatives in academia.
Composition	Attracted thousands of participants across 15 micro-conferences.	Hosted 5,055 participants from 50 countries.	Over 60,000 employees using GitHub, managing 14,000 repositories.	Contributors from 22 universities worldwide.
Major OSS Projects	Linux Kernel, Kubernetes	visdom, susi.ai	VS Code, TypeScript	Carnegie Mellon University, Johns Hopkins University
Participants	Focus group with OSS founders, developers, designers, and community managers. Survey distributed to all registrants.	Survey deployed to all registered participants.	Participants include engineers, writers, product managers, and community leads.	Survey shared by a CURIOSS member through their mailing list.

environments (Walsham, 1995). We therefore employed a theoretical sampling strategy to select four distinctly situated OSS organizations, as shown in Table 2. This approach allowed us to capture case variability across different dimensions, including domain (scientific vs. commercial), scale (community-led vs. corporate-sponsored), maturity, governance model, and regional scope (Walsham, 1995). The diversity of these cases not only enriches our dataset but also strengthens theoretical triangulation, enabling us to cross-validate and deepen our insights by comparing patterns across different contexts and levels of analysis (Venkatesh, Brown, & Sullivan, 2016).

Data Collection. The data for our multiple case study were collected empirically between 2024 and 2025 (Table 2):

- ***Establishing Community-Grounded Definitions of* GLUE WORK*:* Focused Group Discussion and Followup Interview.** We conducted a focus group discussion with 20+ OSS contributors at Open Source Summit North America 2024, followed by follow-up interviews (N=9) with those who did not attend, to explore community-grounded definitions and contextual characteristics of GLUE WORK, capturing diverse perspectives from a global conference setting.

- ***Triangulating* GLUE WORK *Characteristics:* Surveys I & II.** We collected responses

from Open Source Summit North America (N=107) and FOSSAsia (N=74) to triangulate and validate the characteristics and perceived types of GLUE WORK, leveraging the large, diverse participant pools of these major events to reflect regional and cultural variations in OSS practices.

- ***Deepening Understanding of Experiences with Recognition of* GLUE WORK*:*** **In-depth Interview.** We conducted eight in-depth interviews with practitioners from Microsoft Open Source Projects, drawing on earlier findings to gain nuanced insights into contributors' experiences and perceptions of GLUE WORK recognition, focusing on the corporate-sponsored OSS ecosystem and its unique governance structures. We selected Microsoft due to its leadership in corporate-sponsored OSS and its influence on shaping governance practices, offering a unique perspective on how GLUE WORK is managed within large-scale, industry-supported ecosystems.

- ***Triangulating Recognition of* GLUE WORK*:*** **Survey III.** We surveyed OSS practitioners (N=78) from the CURIOSS Community to triangulate where and how GLUE WORK can be recognized and acknowledged. Building on insights from global conferences and corporate ecosystems, this final step provides complementary quantitative and qualitative perceptual data from a scientific, university-driven OSS community, offering a distinct academic perspective to validate and extend our cross-contextual understanding of GLUE WORK.

Rigor and Credibility. This study was designed and executed in alignment with contemporary principles of qualitative IS research (Sarker, Xiao, & Beaulieu, 2013), with an emphasis on transparency, methodological rigor, and trustworthiness (Eisenhardt, 1989; Dubé & Paré, 2003). To enhance credibility, we employed established strategies such as using multiple investigators, overlapping data collection and analysis phases, and refining our semi-structured interview protocols with input from domain experts. We also provide supplementary materials, including our codebook, thematic structures, representative quotes, interview guide, and validation instruments, to support the trustworthiness of our findings and enable readers to assess the strength of our claims (Anonymous, 2025).

The research team acknowledged the interpretive nature of both data and analysis. As such, we remained reflexive about our preconceptions and the impact of our positionality on the inquiry process. Our aim was not to produce universal generalizations, but to build rich, context-sensitive theory grounded in the voices, practices, and challenges faced by real-world OSS contributors.

3.2 Overview of Taxonomic Theory of GLUE WORK.

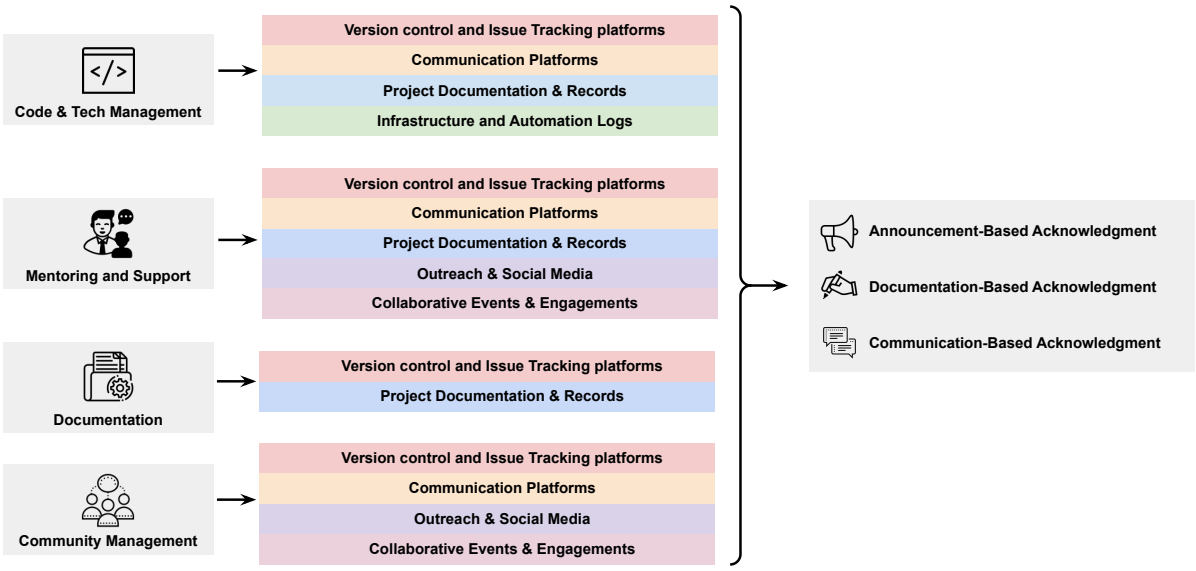

Figure 1: Theoretical Taxonomy of GLUE WORK in OSS.

We have developed a taxonomic theory of GLUE WORK in OSS that aligns with Gregor (2006)'s classification of analytic (Type I) theories. Our theory provides analytic and descriptive insights, systematically organizing and summarizing the salient attributes of GLUE WORK.

Figure 1 presents our high-level taxonomy of GLUE WORK in OSS, organized along three analytical dimensions: (1) What it is (RQ1)?, (2) Where to track it (RQ2)?, and (3) How to acknowledge it (RQ3)?. We identified 12 distinct GLUE WORK over four overarching categories of GLUE WORK: Code & Technical Management, Mentoring & Support, Documentation, and Community Management. Each category reflects a core functional domain and is observable via specific trackable channels (e.g., version control, communication platforms, documentation, automation logs, outreach, and events).

Furthermore, all types of GLUE WORK can be acknowledged and recognized through three modes—announcement-based, documentation-based, and communication-based (e.g., contributor listings, thank-you notes, community announcements). The following sections expand on how we constructed and explain the taxonomy through each analytical dimension.

4 Defining GLUE WORK in OSS (RQ1).

In this section, we begin by unpacking what GLUE WORK entails within OSS communities. We first describe how each case study was conducted, including methodological details and our approach to data analysis. We then present our first analytical lens on GLUE WORK, its characteristics, and the different forms it can take in practice.

4.1 Construct Identification and Conceptual Grounding.

To understand the landscape of GLUE WORK in OSS communities and lay the foundation for our taxonomy, we conducted a multi-vocal literature review (Garousi, Felderer, & Mäntylä, 2019) (combined review of academic and gray literature), followed by case studies involving focus group discussions, interviews, and surveys.

Literature Review. We reviewed both academic literature and industrial gray literature (e.g., blogs), as previous studies have highlighted persistent disconnections and miscommunications between academic research and industry practice (Rico, 2020).

Keyword: We first determined a list of search keywords about GLUE WORK in OSS, such as "glue work" AND "OSS" and "glue work" AND "open source" from our RQs. To capture discussions around contributions beyond coding, we included keywords like "Non code" AND "OSS", "non coder contributor" AND "OSS", "Non code" AND "Open Source", "non coder contributor" AND "Open Source", "Non technical" AND "OSS", and "Non technical" AND "open source". These keyword choices were informed by prior academic studies on relevant studies in OSS (Young et al., 2021; Trinkenreich et al., 2020) and the first blog discussing GLUE WORK (Reilly, 2019).

Academic Digital Libraries and Pilot Search: We began by searching six popular digital

libraries commonly used in prior literature reviews in IS (Webster & Watson, 2002; Leidner & Kayworth, 2006) and software engineering (SE) (Feng et al., 2024; Trinkenreich, Wiese, Sarma, Gerosa, & Steinmacher, 2022): four libraries relevant to IS (MIS Quarterly via JSTOR, Information Systems Research via JSTOR, AIS eLibrary, and the Journal of Management Information Systems) and two relevant to SE (ACM Digital Library and IEEE Xplore). We required the keywords to appear at least once in the title, abstract, or keywords. This approach is consistent with existing studies in OSS research (Feng et al., 2024).

To ensure the robustness of our keywords, we conducted a pilot search using three studies we were already familiar with in both IS and SE libraries. The keywords successfully retrieved two of the control studies (Young et al., 2021; Yeats, 2008).

Academic literature list: Our initial search after removing duplicates resulted in 86 publications. Then, the first and second authors read the titles and abstracts and only selected papers that were relevant to GLUE WORK. When we finished filtering papers, 22 remained (10 from SE libraries, and 12 from IS libraries).

To collect additional studies, we used backward snowballing to broaden the scope to other libraries (Kitchenham et al., 2009). This approach added 23 papers to our literature list, resulting in a total of 45 academic papers. Many of the studies found through snowballing did not appear in our first search pass as they focused on broader or indirectly related areas and did not explicitly mention GLUE WORK (See Supplementary for the list (Anonymous, 2025)).

Gray Literature Review was conducted on the top 50 OSS blog platforms from the list of "Best Open Source Blogs and Websites" on Feedly.com (Feedly, 2024). These blog platforms included Open Source Initiative (Open Source Initiative, 2024), Google Open Source Blog (Google Open Source, 2024), Linux.com (*Open Source Maintainers: What They Need and How to Support Them*, n.d.), the Software Freedom Conservancy blog (Software Freedom Conservancy, 2024), and other foundation-based platforms. We removed 18 platforms that were project-specific, focused on release updates, or were personal web pages without a review process, making them less reliable (*Open Source Maintainers: What They Need and How to Support Them*, n.d.; OpenSSF Community, 2024). We used the same keywords from the academic literature review, resulting in two blog articles (Blog, 2023; Opensource.com,

2020). Such scarcity of blogs on this topic highlights an urgent need to understand and recognize the GLUE WORK in OSS.

Expert Consultation. In case we overlooked any literature, we consulted researchers from Google's Open Source Programs Office (Google Open Source Programs Office, 2023) and conducted author snowballing from the control study of the pilot search (Young et al., 2021) to identify additional materials on GLUE WORK. The researchers provided five more OSS resources, including four OSS websites (*all-contributors*, 2024; Experience, 2024; *CHAOSS: Community Health Analytics Open Source Software*, n.d.; NISO CRediT, 2024), and a conference talk video (YouTube, 2023) (See Supplementary doc for the list (Anonymous, 2025)).

Data analysis: While a few gray literature artifacts explicitly used the term "glue work", most academic studies did not. We therefore analyzed whether the contributions discussed in each paper aligned with the three mechanisms of invisible labor—invisibility due to sociocultural, sociospatial, or sociolegal factors (Hatton, 2017). The first and third authors independently reviewed the selected academic papers and gray literature to identify contributions that were essential to OSS functioning yet potentially underacknowledged or culturally devalued, thereby falling into the three mechanisms of invisibility.

We employed an open-coding approach to inductively capture the types and forms of these contributions, followed by a negotiated agreement process (Raiffa, 2007). Weekly meetings were held to present findings, resolve discrepancies, and reach consensus through discussion and reflection on existing theories, adhering to established standards and accepted practices for qualitative data analysis (Srivastava & Chandra, 2018). The findings primarily highlighted the characteristics of GLUE WORK, such as the types and forms it may take (Refer to the supplementary document (Anonymous, 2025) for mapping of the literature).

Focus Group Discussions and Follow-up Interviews at Open Source Summit North America. After building our initial understanding of GLUE WORK, we organized a focus group discussion at Open Source Summit North America (Linux Foundation, 2024).

The focus group discussion session (Kontio, Bragge, & Lehtola, 2008) included guided and open-ended questions (See Supplementary (Anonymous, 2025)). The session lasted 40

minutes and was divided into three phases: We began by introducing what we have learned about GLUE WORK through literature reviews. We then structured the discussion session by proposing questions such as: *(1) What types of tasks do you consider as* GLUE WORK *in OSS? (2) What percentage of your OSS work is acknowledged by your peers or organization? (3) What are the biggest challenges you face or perceive in performing* GLUE WORK *for OSS projects? (4) What incentives would motivate you or others to contribute more actively to* GLUE WORK *in OSS projects?*

Finally, we opened the floor for discussion where participants were encouraged to freely share their GLUE WORK experiences, challenges, and perspectives of GLUE WORK, fostering an unstructured and dynamic exchange of ideas. More than 20 participants attended the session, representing a range of backgrounds, from non-technical contributors to maintainers, which was video-recorded and subsequently transcribed.

Follow-up Interviews: To address potential limitations of the focus group discussion, such as time constraints and the absence of some participants, we conducted short, follow-up interviews with OSS contributors the day after the focus group discussion, engaging with different groups of people. Each interview consisted of two key questions aimed at understanding (1) the types of GLUE WORK participants have experienced and (2) their perceptions of GLUE WORK. Before conducting these interviews, we conducted two pilot studies with OSS practitioners to ensure the interviews were concise and narrative-driven.

One author of this paper, a Cloud Native Computing Foundation Ambassador specializing in marketing and advocacy, facilitated recruitment by engaging with nine conference attendees during lunch and social breaks for interviews. We stopped at nine as we reached information saturation. Prior to conducting the interviews, we prepared a checklist based on literature reviews and focus group discussions to validate the existing types of GLUE WORK and identify any new ones. Each interview was audio-recorded and transcribed afterward.

While we found no new types of GLUE WORK during the interviews, we did observe similar responses regarding interviewees' experiences with GLUE WORK from the fifth participant onwards, such as challenges in recognition and acknowledgment. As these were short follow-up interviews, we conducted four additional interviews to confirm data

Table 3: Demographic of DI

ID	Gender	Senority	Roles	Organization
FI1	Man	3-5 years	Community Manager	Github
FI2	Woman	6-10 years	Software Engineer	GitLab
FI3	Woman	3-5 years	Community Manager	CNCF
FI4	Man	3-5 years	Software Engineer	GitBook
FI5	Man	>11 years	Software Engineer	TikTok
FI6	Woman	6-10 years	Outreach	Linux Kernel
FI7	Man	<2 years	Outreach	Redhat
FI8	Man	3-5 years	Software Engineer	Redhat
FI9	Man	<2 years	Software Engineer	Costco

saturation. Table 3 presents the demographic information of participants in the follow-up interview, who are from various organizations with diverse backgrounds.

Data analysis: The data analysis followed a continuous comparison process grounded in negotiated agreement. We began with a set of prior codes derived from our literature review. Two researchers independently coded the transcripts and held iterative meetings to compare emerging codes with existing ones, determining whether each new instance represented a distinct category or a sub-dimension of an existing construct. Table 4 outlines the steps in our coding process, which allowed us to identify multiple types and characteristics of GLUE WORK, guided by theoretical insights on invisible labor mechanisms (Hatton, 2017).

Survey Validation at Open Source Summit North America and FOSSAsia 2024. The identified types of GLUE WORK were then triangulated with a larger population of OSS contributors from different regions to enhance the generalizability and credibility of our findings (Venkatesh et al., 2016). Specifically, we surveyed OSS developers who attended FOSSAsia (FOSSASIA, 2024) (N = 74) and the Open Source Summit North America (Linux Foundation, 2024) (N = 107).

Survey design: Our research team consisted of researchers from Google, which sponsored both conferences, allowing us to integrate our survey into the conference's organizational processes.

At FOSSAsia, we were allowed to ask one question; at Open Source Summit North America, we were allowed to ask two questions. The first question, *"Which types of contributions do you think are* GLUE WORK*?"*, aimed to triangulate the types of GLUE WORK.

Table 4: Sample Coding Analysis

Sample	Code by Author 1	Code by Author 2	Consenusus first order	Second-order code	Final construct
"There is not recognition specifically and I also don't think there is groups of people trying to align on how to contribute or even expand those contributions" [FI7].	Unrecognized efforts	Lack of structural support	Lack of formal acknowledgment	Sociolegal invisibility	Characteristics of GLUE WORK
"They're also not speaking up, they're doing the website, they're doing the repository cleanups, they're doing all kinds of really awesome work that is not seen" [FP].	Invisible labor	Untraceable	Unseen contribution	Sociospatial invisibility	Characteristics of GLUE WORK
"Like maybe you came to a project to make a technical contribution, and you ended up making it so that each of the issues and bugs are documented. And that might not feel valuable to you, because no one values glue contributions. This kind of work just isn't culturally recognized. And then the other thing is just getting burnt out" [FI3].	Undervalued contribution	Burnout from invisible work	Culturally devalued maintenance	Sociocultural invisibility	Characteristics of GLUE WORK
"I've organized student hackathons. I have organized high school hackathons" [FI1].	Event organizing	Community coordination	Event organizing	Community building	Types of GLUE WORK
"I worked on a CICD pipeline issued to automate the licenses file for the open source project that we were working on" [FI2].	License management	License maintenance	License management	Code and technical management	Types of GLUE WORK
"So basically my whole team are non-coders because we all create content, "we translate content, we edit content, we publish content" [FGD].	Non-code contribution	Content contribution	Writing/editing documentation	Documentation	Types of GLUE WORK

The second question (asked only at Open Source Summit North America), *"Which aspects of* GLUE WORK *do you think positively impact OSS sustainability?"*, was motivated by our qualitative analysis and served as a starting point to understand the impact of GLUE WORK in OSS (See Supplementary (Anonymous, 2025)).

Survey responses: Responding to these questions was optional in both surveys. We received the survey responses from the conference organizing team (demographic data was not released). At FOSSAsia, we received 73 responses to Q1. At Open Source Summit North America, we received 107 responses to Q1 and 101 responses to Q2.

4.2 GLUE WORK in OSS (RQ1)

Characteristics of GLUE WORK. GLUE WORK refers to contributions that are **critical** to a project's health as it helps, "*bring the community together*" [FI6] [1]. These contributions include efforts such as marketing and advocacy [FI1, FI4, FI7], documentation [FI1, FP, FI9], event organization [FI1, FI3, FI7], and quality assurance [FGD] [2]. *"It's such a good feeling to*

[1] Follow-up Interview Participant [2] Focus Group Discussion Participant

have people come and recognize you. And it's not just me, it's the entire team" [FI4]. However, GLUE WORK receives *"**no recognition**"* [FI7] and remains **unacknowledged**. *"No one values* GLUE WORK *contributions"* [FI3]. *"I think they don't get enough credit at times"* [FI8].

Despite this lack of recognition, participants strongly emphasized that GLUE WORK is **essential** to project sustainability, team efficiency, and long-term momentum. *"*GLUE WORK *is necessary to keep the project alive, sustainable"* [FI1-FI9, FGD]. *"*GLUE WORK *helps spread awareness because otherwise you're not gonna hear so much about it"* [FI4]. *"A lot of projects wouldn't be successful if they didn't have the amount of* GLUE WORK *that is being done"* [FI3].

Category of GLUE WORK in OSS. Next, we unpack different types of GLUE WORK contributions in OSS, which we group into four categories: (1) code and technical management, (2) mentoring and support, (3) documentation, and (4) community management. Table 5 presents the taxonomy of GLUE WORK in response to RQ1, developed through the triangulation of multiple data sources.

Code and Technical Management. Writing Maintenance Code includes refining the existing codebase, addressing compatibility issues, and ensuring long-term maintainability (Trinkenreich et al., 2020). However, unlike feature development, maintenance work is often complicated, heavy (Christa, Madhusudhan, Suma, & Rao, 2017), invisible, and undervalued (Young et al., 2021). Due to limitations in collaboration mechanisms, such as GLUE WORK, it is often difficult to track, further complicating its acknowledgment. *"We have a handful of repos... and a lot of the work I was doing was around building those up... the repo would only have [his] name on it but there would be like four of us... helping build it up"* [FI5].

Reviewing Code helps maintain consistent project health and code base quality (Bosu, Carver, Bird, Orbeck, & Chockley, 2016). It not only ensures that contributions adhere to community standards and uphold product quality but also fosters collective learning and a shared understanding among contributors (Wessel, Serebrenik, Wiese, Steinmacher, & Gerosa, 2022). Although it is trackable in current collaboration platforms (e.g., GitHub), it is largely undervalued (Sarma & Chen, 2024). *"I think [reviewers] don't get enough credit at times.*

Table 5: Catalog of GLUE WORK (See Supplementary for the codebook (Anonymous, 2025))

	Type	Definition	Example	FGD*	FI*	FOSSAsia*, n=73 OSSNA*, n=107
Code and Technical Management	Writing maintenance code	Sustaining, refining, and stabilizing the existing codebase.	Fixing deprecated functions in a web framework to ensure browser compatibility.	✓	FI2, FI5	45% 65%
	Reviewing code	Providing feedback on code contributions.	Reviewing and providing feedback on a pull request.	✓	FI5, FI8	32% 56%
	Quality assurance and testing	Testing to maintain software functionality and quality.	Writing unit tests for a new function.	✓	FI2, FI8	40% 33%
	License management	Managing legal and compliance requirements.	Protects the project's legal integrity.	✓	FI2	30% 27%
	Managing security incidents	Managing vulnerabilities and ensuring software security.	Coordinating a security fix for an encryption module.		FI6	38% 26%
Mentoring and Support	Mentoring, supporting individual contributors	Guiding peers by offering suggestions, recommendations, and support.	Providing onboarding support to newcomers through mentoring programs.	✓	FI1, FI3, FI6, FI5, FI8	38% 60%
	End-user support	Helping users with issues or questions about the software.	Answering questions on Stack Overflow.	✓	FI3	37% 41%
	Bug/issue reporting	Reporting software bugs and issues.	Filing a bug report or submitting an issue.	✓	FI3	32% 36%
Documentation	Writing/editing documentation	Creating and updating guides, manuals, and project documentation.	Updating the README, creating documentation, and internationalizing documents.	✓	FI1, FI8	43% 62%
	Research on community	Investigating community dynamics to understand participation, engagement.	Conducting a survey to gather feedback on contributor experiences.	✓	FI6,	36% 34%
Project and Community Leadership	Operations leadership and governance	Overseeing the project's direction to guide the project's growth.	Organizing monthly committee meetings.	✓	FI3, FI4	43% 28%
	Community Engagement	Promoting the project externally and advocating for its adoption while also engaging and supporting contributors internally.	Moderating community forums, facilitating discussions, and organizing contributor meetups.	✓	FI1, FI3, FI4, FI7, FI8	34% 43%
FGD: Focus Group Discussion; FI: 10 min Follow-Up Interview; FossAsia: 100%; OSSNA: 100%						

That is something we definitely need to look at how we can attribute them more or like give more recognition" [FI9]. As automated bots become prevalent in code reviews, human reviews remain indispensable, ensuring that contributors who dedicate time to reviews receive meaningful engagement and acknowledgment (Feng et al., 2022).

Quality Assurance and Testing safeguard software reliability. Cuccuru et al. (2014) highlights the importance of QA practices, such as testing and issue validation (Contributor Roles Taxonomy, 2022). Yet, these behind-the-scenes efforts often go unrecognized and unacknowledged because their success is measured by the absence of failures rather than the creation of visible features (Sjøberg, Anda, & Mockus, 2012) *"For example, someone involved in their project wound up doing QA for us, which significantly improved quality [...], projects often leaving such critical contributions unacknowledged"* [FGD].

License Management ensures that projects meet legal and compliance requirements (August, Shin, & Tunca, 2018). *"We automated that update to their notice file, which kept them in compliance"* [FI2]. Such efforts prevent legal conflicts and foster safe, stable OSS environments (Link, 2010). Similarly, Managing Security Incidents is essential for maintaining project stability by identifying vulnerabilities, coordinating timely fixes, and ensuring ongoing project integrity (Wen, 2017). However, *"the data can't be easily collected or easily tracked"* [FGD].

Mentoring and Support. Mentoring Contributors includes supporting new and existing contributors, guiding contributors on community norms, and providing resources to lower barriers to entry and support ongoing participation (Feng et al., 2025). *"Because the youngsters, they don't understand how to begin with open source development. So I sort of try to bridge the gap [...] guide them towards what they should be working on"* [FI9]. However, such contributions are not recognized and acknowledged (Feng et al., 2022).

End-user Support includes answering questions on forums, assisting with installation or configuration, and clarifying usage scenarios (Sutanto, Kankanhalli, & Tan, 2014). Accessible support channels and helpful community members can improve project attractiveness and maintain an active, thriving ecosystem (Qiu, Li, Padala, Sarma, & Vasilescu, 2019). However, such contributions often go unrecognized and unacknowledged, as they are hard to track

across different platforms, such as forums, consortia, conferences, and social media (Squire, 2015). *"Open source doesn't have to be oh, this is a community of just developers, this [just for] support open source and use the products and get feedback for the products"*[FI6].

Bug and Issue Reporting involves identifying problems and prioritizing fixes, ultimately improving usability and enhancing software quality (Tan et al., 2014). It guides the continuous refinement of the project, leading to better decision making, more targeted improvements, and a stronger sense of community ownership, ultimately reinforcing the project's long-term success (Bigliardi, Lanza, Bacchelli, D'Ambros, & Mocci, 2014). However, *"issues and bugs might not feel valuable to you because no one values glue contributions"* [FI3].

Documentation and Communication. Documentation makes the software more approachable and inclusive (Bigliardi et al., 2014). Contributors create and update manuals, guides, and internationalized content, which help clarify project structures, standards, and workflows (Rehman, Wang, Kula, Ishio, & Matsumoto, 2020). *"My whole team are non-coders because we all create content, we translate content, we edit content, we publish content"* [FGD]. Even though these contributions are trackable through commits, they often remain underacknowledged. *"I think their [documentation contributors'] contributions are sometimes even way more work than just developers do"* [FI9].

Research on Community including conducting community surveys, health assessments, and OSS metrics analyses, provides insights into contributor experiences, diversity, and engagement patterns (*Metric: Types of Contributions*, 2024). Like many other OSS research studies, this project results from years of extensive data collection from multiple sources and the tireless efforts of many researchers. OSS communities may remain unaware of these efforts without visibility, such as giving talks at OSS conferences. *"because they don't speak up, no one has a chance to acknowledge them"* [FGD].

Community Management. Operational Leadership and Governance guide the strategic direction, resource allocation, and policy-making processes at the project, foundation, or ecosystem level (Feller, Finnegan, Fitzgerald, & Hayes, 2008; Medappa & Srivastava, 2019). *"I believe that sustaining the communities that we build is actually very, very important. But then the process of doing that is extremely hard"* [FGD]. However, such challenging efforts

often remain overshadowed and rarely receive kudos unless the foundation or contributors actively speak publicly to recognize these contributions. *"So I was one of the people that helped start [Foundation] way back in the day... It's about helping drive the foundation to make sure all these projects are funded..."* [FGD].

Similarly, Community Engagement creates and maintains a supportive, inclusive environment (Yeats, 2008). These contributions range from externally advocating for its adoption (Experience, 2024) to fostering collaborative social dynamics within the community (Singh & Phelps, 2013). *"I pretty much do...program management, community management"* [FGD]. *"Marketing our current open source products... going into communities... spreading the natural word,"* and *"posting on LinkedIn... going and trying to connect those communities and just talking with people"* [FI7]. However, these efforts are hard to track, leading to minimal acknowledgment, which can ultimately cause contributors to burn out. *"They're also not speaking up, they're doing the website, they're doing the repository cleanups, they're doing all kinds of really awesome work that is not seen"* [FGD].

Survey Responses. In addition to qualitative interviews, we triangulated these GLUE WORK types through surveys at FOSSAsia (n=73) and Open Source Summit North America (n=107). Results are shown in Table 5's last column. This empirical evidence corroborates our qualitative findings, demonstrating that all the types of GLUE WORK we have identified are integral parts of OSS contribution landscapes.

In our Open Source Summit North America survey, Question 2 asked participants which GLUE WORK has the most positive impact on project sustainability. *"Mentoring and supporting contributors"* received the highest number of responses (23%), followed by *"writing maintenance code"* (13%) and *"writing or editing documentation"* (10%). The remaining types of GLUE WORK received responses ranging from 3% to 9%. However, "license management" and "research on community" didn't receive any responses. *"I think the reality is that people pick a license once, and then they're kind of stuck with it for all time"* [FGD]. While we didn't observe a concentrated single type of GLUE WORK that dominated the responses, this occurrence highlights that GLUE WORK encompasses a diverse range of contributions critical to OSS sustainability. These responses further motivated us to explore

RQ2 by investigating the impact of GLUE WORK and designing additional studies.

5 Tracking & Acknowledging GLUE WORK (RQ2, RQ3)

In this section, we unpack GLUE WORK through two analytical lenses: where to track it (RQ2) and how to acknowledge it (RQ3).

5.1 Construct Identification and Conceptual Grounding

We conducted a multiple case study across two OSS communities—Microsoft Open Source and the CURIOSS community, using a sequential, qualitative-driven mixed-methods design. While our primary data came from these two case studies, we also drew on insights surfaced during the focus group discussion conducted for RQ1. Although tracking and acknowledgment were not the explicit focus of that discussion, several participants made brief or implicit references to how GLUE WORK became visible or valued in their projects. We incorporated these insights into our coding and conceptual development, particularly when they converged with findings from the interviews and survey.

Interview at Microsoft Open Source Projects: To understand where to track and how to acknowledge the GLUE WORK, we conducted a series of interviews within Microsoft Open Source projects. These projects blend business-driven models with OSS governance structures, involving both compensated employees and volunteers, and support commercially viable applications. This diversity makes Microsoft Open Source an ideal context for studying GLUE WORK, as contributors' motivations and project outcomes are tied to both community retention and commercial reputation.

The interviews were designed to be semi-structured, allowing for the emergence of unanticipated insights, as recommended by guidance from IS research on complex sociotechnical phenomena (Sarker et al., 2013). Each interview began with an overview of the study and questions related to demographics. Next, participants were asked to identify which contributions from the GLUE WORK taxonomy they considered GLUE WORK (or not) and explain their decision. We then asked about their experiences with these GLUE WORK

Table 6: Demographic of DI

ID	Gender	Senority	Roles	Organization
DP1	Woman	<5 years	Software Engineer	VScode
DP2	Man	>10 years	Founding Member	FluidFramework
DP3	Woman	<5 years	Software Engineer	react-native-windows
DP4	Man	6-10 years	Technical Writer	VScode
DP5	Man	6-10 years	Software Engineer	TypeScript
DP6	Man	6-10 years	Content Developer	Microsoft OSS projects
DP7	Man	6-10 years	Product Manager	Microsoft Open Technologies, Inc.
DP8	Man	6-10 years	Community Manager	Microsoft OSS projects

contributions (e.g., observations of the impact of GLUE WORK, availability of resources/support available for GLUE WORK). The interview then delved into how GLUE WORK contributions are recognized and acknowledged (See the Supplementary for details of the study design (Anonymous, 2025)).

Before conducting the interviews, we piloted the protocol with three OSS researchers. Their feedback helped refine the interview questions. For instance, they suggested adding questions about existing recognition practices in the interviewees' projects.

Participant Recruitment: We first recruited two participants from the authors' contacts. We then employed a snowball sampling strategy, leveraging the networks of these contributors to recruit additional participants. We stopped recruitment after eight interviews, as we had reached saturation after incorporating insights from both the focus group discussion and follow-up interviews. We determined that from the sixth participant onwards, no new insights emerged regarding the impact on sustainability and the challenges they face. The seventh and eighth interviews served as validations of our findings.

Based on participants' preferences, the interviews were conducted remotely or on-site. Each interview lasted between 40 and 60 minutes. Before the interview started, participants signed a consent form. Participation in the study was voluntary, and participants received no compensation for their time and effort. Participants held diverse OSS roles, including technical writers, engineers, community managers, and content developers (Table 6).

Data Analysis: We analyzed interviews iteratively after each session. Audio recordings were transcribed using Microsoft Teams, and we conducted thematic analysis, beginning with open coding to inductively generate initial codes, following guidance from Braun and Clarke

Table 7: Sample Coding Analysis (Tracing and Acknowledgement)

Sample	Code by Author 1	Code by Author 2	Consenusus first order	Second-order code	Final construct
"I think there would be a lot of bug requests from people saying that you are out of date with packages"[DI3].	Tracing Bug reporting through bug tracking system	Tracking issue resolution history	Bug tracking through GitHub Issues	Version control and issue tracking platforms	Tracking
"It depends on where the community lives and how you organize it. Sometimes it forms on the side, but usually you set a place like GitHub Issues or Discord for engagement" [DI1].	Tracing engement through the communication	Monitoring discussion in community channels	Engagement tracking via Discord/GitHub	Communication platforms	Tracking
"That kind security maintenance of contribution can be tracked through the pipeline logs"[DI3].	Tracing Secruity maintenance through pipeline	Logging maintenance activities	Tracking security maintenance via CI/CD	Infrastructure and automation logs	Tracking
"When there's a release of VS Code, the documentation includes the names of all contributors, who gets it knows who contributed" [DI1].	Named in release documentation	Contributor listing in changelogs	Contributor Listing	Documentation-based acknowledgment	Acknowledgment
"Even if it's as simple as 'thanks for taking the time,' engagement is really important"[DI2].	Verbal appreciation	Direct thanks during interactions	Acknowledgment through peer communication	Communication-based acknowledgment	Acknowledgment
"We actually have a rotating duty on our engineering team to follow up on social media channels, like X or LinkedIn, where users post feedback or questions. It's one way contributors get acknowledged" [DI4].	Recognizing help via social media	Social media engagement as acknowledgment	Community Announcement	Announcement-based acknowledgment	Acknowledgment

(2006). This inductive approach avoided presupposed codes, allowing patterns to emerge directly from the data. The first two authors met weekly to discuss emerging codes, develop and iteratively refine a preliminary codebook, and resolve any disagreements through a negotiated agreement process (Raiffa, 2007). The resulting code themes capture the perceived impact of GLUE WORK and where it is tracked and acknowledged. Table 7 shows an example of our coding process. (see Supplementary Materials for the complete codebook (Anonymous, 2025)).

Survey at CURIOSS Community: To ensure the robustness, validity, and generalizability of our findings in OSS communities for RQ2 (where to track GLUE WORK) and RQ3 (why/how to acknowledge GLUE WORK), we designed a survey following established IS survey design guidelines (Straub, Boudreau, & Gefen, 2004). The study aimed to validate the channels for tracking and acknowledging GLUE WORK, triangulating qualitative insights from interviews (Refer to the Supplementary Materials (Anonymous, 2025) for details).

To distribute the survey, we engaged the CURIOSS community, an OSS community known for its decentralized governance and emphasis on volunteer-driven contributions. CURIOSS's distinct structure, in contrast to Microsoft's hybrid model, enabled us to assess the transferability of identified tracking and acknowledgment mechanisms across varied OSS contexts, thereby strengthening the external validity of our conclusions.

Survey items were developed iteratively, drawing on qualitative findings from prior phases. The survey started with an overview, followed by a consent form. Participants were then asked two mapping questions: one asked participants to map identified channels to the type of GLUE WORK for tracking, and the other asked participants to acknowledge. After each mapping question, participants were allowed to suggest additional channels in the open-text field. We then asked participants demographic questions about gender, OSS experience, and region of origin. The final open-ended question asked participants if they had anything to add about GLUE WORK.

To enhance validity, as recommended by Churchill Jr (1979), we consulted with OSS contributors during three rounds of pilot testing, refining item wording and adding reference information to clarify terms such as GLUE WORK, "tracking," and "acknowledging".

Recruitment: We asked the Open Source Program Office at University X to help publicize the study through the CURIOSS community. We also encouraged participants to share the survey with others in their networks (i.e., snowball sampling) to increase sample diversity and enhance the generalizability of our findings. After removing invalid responses, we were left with 78 valid responses.

In our survey, 44 respondents (56%) identified as men, and 16 (21%) identified as women/non-binary. More than 50 participants (64%) reported having over five years of OSS experience. 31 respondents (40%) were from North America and 25 (32%) from Europe; the remainder were from Asia, South America, and Oceania.

Survey Data Analysis: We used descriptive quantitative analysis of the closed-ended questions to validate the channels, confirming their prevalence and applicability across OSS communities through frequency distributions and cross-tabulations (Phang, Kankanhalli, & Tan, 2015). This approach, supported by qualitative triangulation, aligns with Churchill Jr (1979)'s guidance for applied research, where content validity and descriptive validation suffice for cross-sectional studies.

As for the open-ended questions, we received 23 responses to the tracking channels question and 14 responses to the acknowledgment channels question. For tracking, 11 participants provided examples of existing channels, including mailing lists and forums

(communication channels), release channels (deployment logs), and conferences or events (community websites). Four mentioned individual emails, but this can be hard to track and breaches contributors' privacy. One participant mentioned performance review forms, while the remaining seven explained why they chose specific channels. On the acknowledgment side, one participant suggested adding acknowledgments directly into the User Interface (UI) to highlight UI teams, another mentioned using a NEWS.md file, and another raised performance reviews as a way to give credit. The rest either offered examples of the same categories or clarified selections.

Finally, 26 participants shared thoughts in the survey's concluding open-ended question, largely emphasizing the importance of recognizing GLUE WORK. *"Glue work can be fun, there's usually nobody opposing your contributions, and it's an easy way to get recognition for your skills"* [S4][3], while another remarked, *"It's important to acknowledge the contributors"* [S54].

5.2 Tracking and Acknowledging GLUE WORK

Table 8 presents channels for tracking and acknowledging different types of GLUE WORK. To ensure generalizability, this paper reports only channels validated by at least two data sources. Gray rows indicate channels triangulated using three data sources (focus group discussions, in-depth interviews, and surveys), while white rows represent channels validated by at least two sources. For additional details on channels identified by a single data source, see Supplementary (Anonymous, 2025).

Code and Technical Management, such as Writing Maintenance Code, Reviewing Code, and Quality Assurance can be tracked through channels such as pull requests and contribution histories, with version control systems receiving strong validation. For example, over 90% of participants reported using version control systems to track code maintenance.

"Reviewing code builds community and helps people understand that they're valued... if [contributors] have a positive experience... they'll do it over and over and over again" [DI7][4]. One of the simplest yet most effective ways to acknowledge these contributions is by

[3] CURIOSS Survey participant number [4] Microsoft Open Source Projects Interview Participant

Table 8: Category for Tracing and Acknowledging GLUE WORK (See Supplementary for the codebook (Anonymous, 2025))

	Tracking				Acknowledging			
GLUE WORK	Channel	DI	FGD	Survey (N=69)	Channel	DI	FGD	Survey (N=69)
Code and Technical Management								
Writing maintenance code	Version control (e.g., Git) Issue Tracking (e.g., PR) Issue Tracking (e.g., Issues)	DI-D8 DI-D8 DI-D8	Yes Yes Yes	91% 81% 67%	Documentation-Based (e.g., CONTRIBUTORS.md) Communication-Based (e.g., PR/Issue Comments) Announcement-Based (e.g., Newsletters) Communication-Based (e.g., Slack)	DI-D8 DI-D8 DI-D8 DI-D8	Yes Yes Yes Yes	77% 68% 46% 46%
	Documentation (e.g., Wiki) Communication Platforms (e.g., Slack) Automation Logs (e.g., CI/CD logs)	DI-D8 - DI1-DI3, DI7	- Yes -	50% 33% 29%	Documentation-Based (e.g., Release doc)	DI-D8	-	46%
Reviewing code	Issue Tracking (e.g., PR) Version control (e.g., Git)	DI-D8 DI-D8	Yes Yes	79% 51%	Communication-Based (e.g., PR/Issue Comments) Announcement-Based (e.g., Newsletters)	DI-D8 DI1, DI8	Yes Yes	69% 36%
	Issue Tracking (e.g., Issues)	DI1, DI2, DI7	-	51%	Documentation-Based (e.g., CONTRIBUTORS.md) Communication-Based (e.g., Slack) Documentation-Based (e.g., Release doc)	DI4, DI8 DI1, DI8 DI3, DI4, DI7, DI8	- - -	58% 46% 37%
Quality Assurance	Issue Tracking (e.g., Issues) Issue Tracking (e.g., PR)	DI1, DI2, DI4-DI8 DI1, DI2, DI4-DI8	Yes Yes	56% 54%	Communication-Based (e.g., PR/Issue Comments) Documentation-Based (e.g., Release doc) Announcement-Based (e.g., Newsletters)	DI1- DI8 DI1-DI8 DI1	Yes Yes Yes	58% 44% 32%
	Automation Logs (e.g., CI/CD logs) Version control (e.g., Git) Documentation (e.g., Wiki)	DI1, DI7 DI1, DI3, DI5, DI6, DI8 DI1	- - -	55% 51% 46%	Documentation-Based (e.g., CONTRIBUTORS.md) Communication-Based (e.g., Slack)	DI4, DI8 DI1-DI8	- -	56% 50%
License Management	-				Documentation-Based (e.g., Release doc) Communication-Based (e.g., Slack) Announcement-Based (e.g., Newsletters)	DI1-DI8 DI1-DI8 DI1-DI8	Yes Yes Yes	44% 36% 31%
	Documentation (e.g., Wiki) Version control (e.g., Git)	DI1 DI1	- -	64% 49%	Documentation-Based (e.g., CONTRIBUTORS.md)	DI1-DI8	-	37%
Managing Security	Issue Tracking (e.g., Issues)	DI2,DI4, DI5-DI7	Yes	71%	Communication-Based (e.g., PR/Issue Comments) Announcement-Based (e.g., Newsletters)	DI1, DI4, DI7 DI3-DI5, DI7, DI8	Yes Yes	64% 50%
	Communication Platforms (e.g., Slack) Version control (e.g., Git) Version control (e.g., PR) Automation Logs (e.g., CI/CD logs)	- DI3,DI1, DI4, DI7 DI6, DI7	Yes - - -	69% 62% 51% 35%	Contributor Listing (e.g., CONTRIBUTORS.md) Communication-Based (e.g., Slack) Documentation-Based (e.g., Release doc)	DI4, DI6, DI7 - DI2-DI8	- Yes -	53% 47% 44%
Mentoring and Support								
Mentoring, Supporting	Communication Platforms (e.g., Slack) Communication Platforms (e.g., office hours) Communication Platforms (e.g., X, Linkedin)	DI1-DI8 DI1, DI2, DI4-DI7 DI4, DI5	Yes Yes Yes	69% 63% 40%	Communication-Based (e.g., Slack) Announcement-Based (e.g., Newsletters) Announcement-Based (e.g., X, Linkedin)	DI1-DI8 DI2-DI7 DI4, DI5, DI7, DI8	Yes Yes Yes	59% 56% 36%
	Documentation (e.g., Event Platform) Issue Tracking (e.g., PR) Issue Tracking (e.g., Issues)	DI1, DI8 DI1 DI1	- - -	53% 46% 41%	Documentation-Based (e.g., Release doc) Communication-Based (e.g., PR/Issue Comments)	DI2-DI7 DI1, DI7	- -	44% 42%
End-user Support	Communication Platforms (e.g., Slack)	DI1, DI5, DI6, DI7	Yes	73%	Communication-Based (e.g., Slack) Announcement-Based (e.g., X, Linkedin) Announcement-Based (e.g., Newsletters)	DI3, DI5, DI7 DI4, DI5, DI7 DI2, DI4	- - -	59% 46% 44%
	Communication Platforms (e.g., Stack Overflow) Documentation (e.g., Wiki) Communication Platforms (e.g., X, Linkedin)	DI1 DI1, DI8 DI1, DI4, DI5, DI6	- - -	65% 63% 51%	(e.g., Release doc) (e.g., Release doc)	DI2	-	41%
Bug Issue Reporting	Issue Tracking (e.g., Issues)	DI1-DI8	Yes	90%	Communication-Based (e.g., PR/Issue Comments) Documentation-Based (e.g., CONTRIBUTORS.md)	DI4, DI5 DI1-DI8	Yes Yes	71% 33%
	Communication Platforms (e.g., Slack)	DI1-DI3, DI7	-	51%	Communication-Based (e.g., Slack) Documentation-Based (e.g., Release doc) Announcement-Based (e.g., Newsletters)	DI1, DI3, DI4-DI8 DI2 DI8	- - -	46% 35% 27%
Documentation								
Writing Editing Documentation	Documentation (e.g., Wiki) Issue Tracking (e.g., PR)	DI1-DI8 DI1-DI3, DI6	Yes Yes	85% 59%	Documentation-Based (e.g., Release doc) Documentation-Based (e.g., CONTRIBUTORS.md) Communication-Based (e.g., PR/Issue Comments)	DI1-DI5, DI7, DI8 DI2, DI4 DI1-DI5, DI7, DI8	Yes Yes Yes	60% 58% 56%
	Version control (e.g., Git) Bug Ticketing System (e.g., Issues)	DI1-DI8 DI5	- -	68% 32%	Communication-Based (e.g., Slack) Announcement-Based (e.g., Newsletters)	DI8 DI1, DI4, DI6, DI7	- -	45% 40%
Research on	Communication Platforms (e.g., X) Documentation (e.g., Academic, Blog)	DI1-DI8 DI1-DI8	Yes Yes	67% 31%	Announcement-Based (e.g., Newsletters) Announcement-Based (e.g., X, Linkedin)	DI1-DI8 DI1 - DI3, DI7, DI8	Yes Yes	56% 45%
	-				Communication-Based (e.g., Slack)	DI8	-	60%
Community Management								
Leadership Governance	Communication Channels (e.g., Slack)	DI1, DI6	Yes	76%	Communication-Based (e.g., Slack) Announcement-Based (e.g., Newsletters) Announcement-Based (e.g., X, Linkedin)	DI2, DI4 DI1-DI8 DI1, DI2, DI4, DI6- DI8	Yes Yes Yes	28% 27% 19%
	Documentation (e.g., Event Platform) Documentation (e.g., Wiki)	DI1, DI2, DI4, DI6- DI8 DI1, DI2, DI4	- -	78% 49%	Documentation-Based (e.g., Release doc) Documentation-Based (e.g., CONTRIBUTORS.md)	DI1-DI8 DI1, DI2, DI4, DI6- DI8	- -	26% 21%
Community Engagement	Communication Channels (e.g., Slack) Documentation (e.g., Project Website) Interactive Engagement (e.g., office hours)	DI1 DI1- DI3, DI4, DI7, DI8 DI1-DI5, DI8	Yes Yes Yes	67% 62% 59%	Communication-Based (e.g., Slack) Announcement-Based (e.g., X, Linkedin) Documentation-Based (e.g., CONTRIBUTORS.md)	DI3, DI4, DI8 DI2-DI4 DI1 - DI3, DI4, DI7, DI8	Yes Yes Yes	65% 53% 36%
	Communication Platforms (e.g., X, Linkedin)	DI1, DI4, DI5, DI7	-	50%	Announcement-Based (e.g., Newsletters)	DI1-DI8	-	58%

directly expressing gratitude through pull requests, issue threads, and other communication channels, *"We need to say thank you"* [DI2]. This is especially important for newcomers, as it fosters a sense of belonging and can enhance retention (Steinmacher, Conte, Gerosa, & Redmiles, 2015).

Additionally, community announcements and documentation-based acknowledgments, such as listing all contributors, can also be used for acknowledgment, *"So what we do is we recognize contributors in our monthly release notes. We have a full listing, and thank you for all of the people that contribute, and that could be pull requests, that could be issues being logged, that could be issue triaging that is happening as well"* [DI4], highlighting the importance of structured recognition as noted by Vasilescu et al. (2015) on the impact of visibility on contributor motivation.

License Management is one of the most under-acknowledged areas, with contributions often remaining invisible and typically confined to internal teams (Trinkenreich et al., 2020). Tracking poses significant challenges. *"License management, that is not possible [to track], unfortunately, because our licenses, I believe, are managed by lawyers. So it's done by only one or two people in the team"* [DI1]. Channels such as documentation and version control systems can be used for tracking; in fact, 64% of survey participants identified project documentation as suitable for tracking. Acknowledgment through project documentation (44% of survey participants; DI1–DI8) and community announcements (34%; DI1–DI8) can help bring visibility to this work. Notably, the Cloud Native Computing Foundation has pioneered approaches to formally recognize a wide range of contributions by listing them in dedicated community repositories (Cloud Native Computing Foundation (CNCF), 2024).

Managing Security Incidents *"may be one of the most important things because in [Repo], when there are security incidents we take them very seriously"*, which can be tracked through issue logs and version control metadata, as validated by 71% of survey participants and confirmed by interview participants (DI2, DI4, DI5-DI7). In some projects, in addition to communication platforms and version control systems, infrastructure logs—such as those from continuous integration and continuous delivery/deployment, also help surface this work *"That kind [security maintenance] of contribution can be tracked through the pipeline logs"*

[DI3]. Acknowledgment in this area often occurs through communication-based channels, such as pull request comments, issue threads, expressions of gratitude, or project announcements. More than 50% of participants agreed with this, and it was also mentioned by interview participants and in focus group discussions. However, the form of acknowledgment varies depending on the visibility and sensitivity of the security work involved.

Mentoring and Support are often informal, relational, and challenging to capture through traditional software contribution records. *"Mentoring isn't something most of our engineers do... it's not in their job description"* [DI5].

Mentoring and Supporting Contributors can be tracked across various community platforms. Slack discussions and office hours were the most commonly validated mechanisms, with 69% and 63% of survey respondents, respectively, indicating they could be used for tracking such activities. Social media platforms (e.g., X, LinkedIn) were also mentioned as potential sources. Prior work by Feng et al. (2022) explored how machine learning techniques could help identify mentoring behaviors embedded in communication threads.

"Let's say some user logs an issue for our project, and then people from the community contribute by providing additional details or steps to reproduce the issue. So they contribute to the issue, but they're not the author of the issue, so they might not appear that way. Or they contribute to pull requests, not by committing code changes, but by providing useful information" [DI5]. To ensure such efforts are acknowledged, projects can adopt both informal and formal mechanisms. Informally, communication-based acknowledgments, such as expressions of gratitude in chat channels or issue threads, can validate contributors' support roles. Formally, documentation, contributor profiles, or project announcements can highlight mentors and make their efforts more visible to the broader community.

End-user Support is another often-overlooked form of GLUE WORK, especially considering that such contributions often live across different platforms, involving contributors who answer user questions, troubleshoot issues, and provide help through platforms like Slack or Stack Overflow (Von Krogh, Spaeth, & Lakhani, 2003). *"Somebody who was involved in their project who wound up doing QA for us, and that was incredibly beneficial, and that significantly improved our quality and was stuff that I didn't have to think about, right?"*

[FGD]. Communication platforms like Slack were the most frequently cited tracking mechanism, validated by 73% of survey participants and interviewees (DI1, DI5, DI6, DI7). Other channels, such as Stack Overflow (65%), project documentation (63%), and social media platforms like X and LinkedIn (51%) were also mentioned as trackable venues for end-user support contributions.

These contributions are typically informal, yet they are essential for sustaining usability and improving the overall user experience. Therefore, acknowledging them is essential. *"End-user support is extremely helpful; engineers benefit greatly from people telling them things they don't know"* [DI2].

Bug and Issue Reporting is one of the most structurally trackable forms of GLUE WORK. Contributors often report bugs by opening issues or providing feedback, forming a critical part of quality assurance and maintenance cycles. Issue tracking systems (e.g., GitHub Issues) were validated by 90% of survey respondents and all interviewees (DI1–DI8) as the primary tracking channel, aligning with Dabbish, Stuart, Tsay, and Herbsleb (2012), who identified issue trackers as central to OSS collaboration. Communication platforms like Slack were also recognized as secondary spaces where bug reporting is initiated or discussed (51.5%). *"Obviously, bug issue reporting is, you know, extremely appropriate. If they think they found a bug, that's great. They should put in good information"* [DI7].

"If you contributed a bunch of process improvements through discussion and sharing of expertise but didn't end up contributing a code change, there's probably no system for that recognition" [DI2]. Recognizing these efforts not only boosts contributor morale but also fosters a broader culture of appreciation and inclusivity. Communication-based acknowledgment (e.g., thank-you messages), documentation-based listings (e.g., release notes), and public announcements all serve as viable pathways for making this otherwise invisible labor more visible.

Documentation. Despite being highly visible in GitHub repositories, documentation work remains undervalued and underacknowledged in many OSS projects. *"Code is more valuable than documentation. The documentation would not need to exist without the code"* [DI6]. This perception contributes to the consistent under-recognition of documentation

contributions, even though they are essential for usability, onboarding, and long-term maintenance (Trinkenreich et al., 2020).

Documentation Contributions can be tracked through pull requests and commits in GitHub repositories. *"All our documentation is stored in GitHub repositories, so contributions are visible in commits and pull requests. Anyone who submits edits or additions to the docs is automatically recorded there"* [DI5]. However, these contributions often go unacknowledged in project artifacts such as release notes or contributor listings. Only 60% of survey respondents reported seeing documentation-based acknowledgment (e.g., release docs), even though 85% agreed that documentation work could be effectively tracked (DI1–DI8). Communication-based acknowledgment can also help foster an appreciative and inclusive project culture, for example, through pull request comments such as: *"Maintainers often add a quick thank-you message when merging PRs, like 'Thanks for improving the docs!"'* [DI7].

Research oriented contributions, such as writing academic blog posts, conducting collaborative investigations, or documenting design decisions, are an emerging but often overlooked form of GLUE WORK. These contributions are usually trackable through communication platforms (67%) and documentation artifacts such as research blogs or meeting notes (31%). As one participant explained, *"Collaborative research is logged in GitHub discussions or API Council meeting notes"* [DI7].

Acknowledging research work typically happens through social and outreach-based channels, such as community announcements and social media. For instance, 56% of survey respondents noted newsletters as a viable form of acknowledgment, while 45% pointed to platforms like X or LinkedIn. However, few systems exist to consistently surface and reward these research-related efforts within OSS workflows.

Community Management is an especially important form of GLUE WORK. Participants emphasized that it enhances community engagement, supports a large user base (Nakakoji, Yamada, & Giaccardi, 2005; Lyulina & Jahanshahi, 2021), and fosters sustained collaboration [DI1–DI5, DI7, DI8]. As one participant noted, *"I believe that sustaining the communities that we build is actually very, very important* [FGD].

Leadership and Governance-related Contributions such as organizing events, facilitating

community-wide decisions, or maintaining policies, can be tracked through multiple platforms. Communication channels (e.g., Slack) were identified by 76% of survey respondents and interviewees (DI1, DI6) as helpful in tracing such work. Governance-related documents like event platforms and wikis were also supported by 78% and 49% of survey respondents, respectively, with interview validation from DI1, DI2, DI4, DI6–DI8. Despite the availability of tracking channels, acknowledgment remains limited. Only 28% of survey participants noted communication-based acknowledgment (DI2, DI4), 27% reported newsletter-based recognition (DI1–DI8), and 19% reported social media mentions (DI1, DI2, DI4, DI6–DI8).

Community Engagement was also trackable through communication channels (e.g., Slack), which were validated by 67% of survey participants and interviewees (DI1), while documentation on community sites was validated by 62% of survey participants and interviewees (DI1–DI3, DI4, DI7, DI8). Interactive engagement formats, such as live sessions or Q&A hours, were supported by 59% of survey respondents and interviewees (DI1–DI5, DI8). Social media engagement (e.g., X, LinkedIn) received 50% support from survey participants (DI1, DI4, DI5, DI7).

As noted by several participants, the lack of acknowledgment for non-code GLUE WORK remains a challenge. Release notes tend to prioritize code contributions, *"through discussion and sharing of expertise, but [that] didn't end up contributing a code change"* [DI5]. However, *"Communities should feel engaged in the project and feel invested"* [DI1], which enhances the sense of welcome within OSS communities. *"I can point to this release and say, 'Yeah, I did this.' That's a sense of ownership"* [DI2].

6 Conclusion and Discussion

To our knowledge, this study is the first to provide an empirically grounded, theoretically informed taxonomy of GLUE WORK in OSS, extending the concept of invisible labor to a socio-technical, decentralized, digital collaboration platform. In the following section, we unpack the implications for both research and practice in OSS.

6.1 Theoretical Contribution

First, our study introduces a comprehensive and empirically grounded taxonomic theory of GLUE WORK that redefines how OSS contributions are conceptualized. While prior work acknowledges the existence of "invisible labor" in OSS, our theory systematizes this labor into 12 types grouped under four categories: Code & Technical Management, Mentoring & Support, Documentation, and Community Management. We not only identify what constitutes GLUE WORK but also explain its systematic devaluation and invisibility in the context of OSS culture. This re-conceptualization moves beyond ad hoc examples and provides a structured vocabulary and analytical foundation to study OSS labor inclusively.

Second, we extend the theory of invisible labor by contextualizing its sociocultural, sociospatial, and sociolegal mechanisms within distributed digital ecosystems like OSS. While prior work identifies these mechanisms as sources of invisibility (Hatton, 2017), there has been limited theorization on how they manifest in socio-technical systems. Our contribution lies in analytically mapping these mechanisms to OSS practices, including sociocultural norms that devalue non-code work, the sociospatial dispersion of non-code work across different platforms, and sociolegal gaps in governance and metric systems. By embedding these theoretical dimensions into our taxonomy, we deepen the understanding of why GLUE WORK remains systematically unseen—and how such contributions might be made visible and equitably acknowledged.

Third, we offer a hybrid framework that links contribution types to trackable channels and recognition mechanisms, enabling both researchers and practitioners to connect theory to action. Unlike static typologies, our taxonomic theory presents a dynamic model that accounts for the distributed and multiplex nature of OSS labor. We show how the same contribution can be simultaneously acknowledged, tracked, or made visible depending on the infrastructure through which it flows. By articulating six trackable channels and three acknowledgment mechanisms, we provide a theory-informed framework that guides the design of OSS tools, metrics, and policies to promote fairness, inclusion, and sustainability. This hybrid theorization bridges the descriptive and prescriptive gap in digital work research and contributes to broader discussions in IS and SE about equitable recognition in decentralized

environments.

Finally, our taxonomic theory contributes to broader theoretical conversations beyond OSS by offering analytical tools to study invisible labor in distributed socio-technical systems. Socio-technical systems research (Emery & Trist, 1960) has long emphasized the interaction between social and technical subsystems in coordinating complex work. Yet, many sustaining forms of labor, such as coordination, support, and infrastructural maintenance, remain under-theorized. For example, in traditional systems like enterprise resource planning deployments (Shehab, Sharp, Supramaniam, & Spedding, 2004), technical work is formally recognized, whereas invisible tasks, such as user training or troubleshooting, often go unacknowledged (Boudreau & Robey, 2005). Our taxonomy provides a lens to systematically surface undervalued labor, offering a structured vocabulary and analytical framing that extends beyond OSS into enterprise and other digital ecosystems.

In addition, our study provides a new analytic extension to social network theory (Bapna, Gupta, Rice, & Sundararajan, 2017; Kane, Alavi, Labianca, & Borgatti, 2014). Classic network models typically foreground visible interactions between nodes—such as commits, issue comments, or code reviews—while overlooking less tangible forms of contribution. This creates a visibility bias that limits our understanding of how OSS projects function and endure. By expanding the unit of analysis to include invisible coordination and relational work, our taxonomy enables researchers to incorporate underrepresented labor into social network models, thereby enhancing the understanding of social networks and power structures in organizational networks. This approach allows for more comprehensive mappings of influence, collaboration, and resilience in digital environments, and it supports more inclusive theoretical models of virtual work.

6.2 Implications for Practice

Recognizing and supporting GLUE WORK in OSS requires concerted effort across all roles in the ecosystem, not just from individual contributors, but also from project leaders and tool builders. Based on our findings, we offer tailored recommendations for how each group can promote, record, and reward GLUE WORK. These calls to action aim to foster a more inclusive

and sustainable OSS environment where diverse forms of contribution can be equally valued in OSS.

For **community leaders**, we encourage them to systematically adopt the different channels to record and acknowledge GLUE WORK, build them into project policies, and foster a culture where GLUE WORK efforts receive the same respect and visibility as code-based work. *"A lot of projects wouldn't be successful if they didn't have the amount of glue work that is being done"* [FI3]. Teams can encourage the practice of peer recognition of GLUE WORK, which not only gives credit and builds team strength but also helps in rewarding GLUE WORK. Finally, teams could include GLUE WORK contributions as part of their team retrospectives, where they talk about "what went right/wrong" regarding technical development and about GLUE WORK. This would help highlight the importance of GLUE WORK for the project and the effort it takes to perform them.

We also urge community leaders to actively incorporate GLUE WORK into project priorities and recognition frameworks to support the "deeply caring" individuals who bolster community well-being. Considering GLUE WORK when making strategic decisions will help your project flourish and retain dedicated contributors who strengthen the collaborative spirit. Consider providing career development opportunities tied to GLUE WORK like offering management training or cross-team collaboration roles for those excelling in coordination and mentorship. Finally, showcase the importance of GLUE WORK to your project by establishing guidelines related to GLUE WORK that mandate well-maintained documentation, formalize mentorship programs, or require effective onboarding processes.

For **OSS tool builders**, we encourage them to prioritize the development of mechanisms that track and reward these contributions. For example, consolidating diverse contribution channels into cohesive dashboards can spotlight overlooked community roles. *"I think the biggest challenge we've had is just trying to come up with a system that tracks their contributions"* [FGD]. For instance, leveraging advanced AI mechanisms, such as Large Language Models (LLMs), can enable automated analysis of GLUE WORK across platforms like GitHub, Discord, and mailing lists to identify and quantify activity, including mentoring, issue triaging, and community moderation. By fine-tuning LLMs to detect patterns of GLUE

WORK and integrating these insights into transparent metrics or reward systems, tool builders can create more inclusive frameworks that enhance the visibility and recognition of GLUE WORK, reducing manual effort and practitioner burden while preserving nuance.

For **general OSS contributors**, we have systematically listed the type of GLUE WORK that contributors can participate in OSS communities and presented the impact of GLUE WORK on communities, projects, and individual growth. For many, especially those without technical backgrounds, GLUE WORK offers an accessible entry point to contribute meaningfully to large OSS projects. As some of our participants noted, *"It's been critical to my career path, a productive space to be working and contributing in"* [DI6]. By participating in these activities, newcomers can develop skills, gain visibility, and establish a robust career profile, paving the way for professional growth in the tech ecosystem. *"For me, it's not glue work. For me, it's how I build a career out of it"* [FI1]. We hope this work broadens access to OSS, outlines new opportunities for diverse contributors, and advocates for equal visibility and recognition of GLUE WORK alongside more traditional technical contributions. *"I think it is the most valuable work that you can do for open source"* [FI3] and *"[*GLUE WORK*] was one of [the contributions] which made me want to lean in even further and contribute"* [FGD].

6.3 Limitations and Future Research

Our paper has several limitations that provide promising opportunities for future research. First, the GLUE WORK taxonomy may miss certain types of contributions. To mitigate this, we triangulated our taxonomy through multiple methods (e.g., focus group discussions, interviews, and surveys), which enhanced its reliability. Similarly, in the analysis for tracing and acknowledging in GLUE WORK, we recognize that our result may be limited to certain project cultures and dynamics. We mitigate this issue by only including items that were cross-validated with at least two independent data sources. Second, the subjective nature of qualitative research can introduce variability in data interpretation, as researchers' perspectives may influence the analysis and interpretation of focus group discussions, follow-up interviews, and in-depth interviews. To mitigate such bias, this study incorporated multiple case studies from diverse organizations and governance structures, with each research question

triangulated through surveys to enhance the robustness of the findings. Third, qualitative methods do not provide quantifiable metrics, such as the frequency of channels based on the number of GLUE WORK they can track, limiting their ability to present precise numerical data. Instead, we detail these numbers in the Supplementary doc (Anonymous, 2025). Overall, their efforts aim to construct a theoretical taxonomy that offers more generalizable insights, though the inherent subjectivity of qualitative analysis remains a limitation.

Future research on GLUE WORK can investigate the directives, performance, and impact of GLUE WORK on project sustainability. For example, empirical investigations are needed to quantify the broader effects of GLUE WORK. *"Making builds faster may not seem flashy, but it enables the team to ship faster"* [DI5]. Existing research tracking implicit mentoring in pull request comments empirically demonstrated that mentoring benefits contributors and projects (Feng et al., 2022). However, many other forms of GLUE WORK remain unexplored, especially in investigating the impact on the long-term sustainability of communities. For example, investigating whether a project's proportion of GLUE WORK correlates with its overall health could provide insights for community management.

Finally, we have synthesized our findings into a companion website [5] that provides actionable strategies for tracking, acknowledging, and celebrating GLUE WORK. Practitioners can use this resource to broaden participation and increase awareness of GLUE WORK within their software communities. In accordance with our anonymization policy, we will open-source the GLUE WORK companion website upon acceptance, inviting broader OSS contributions to expand the scope of GLUE WORK. The research artifacts for this study are available publicly (Anonymous, 2025).

References

all-contributors. (2024). Retrieved from `https://allcontributors.org/` (Accessed: 2024-10-02)

Anonymous. (2025, March). *The nuts and bolts of open source: A taxonomy of glue work in oss projects.* Zenodo. Retrieved from

[5] https://gluework.netlify.app/

https://doi.org/10.5281/zenodo.15003981 doi: 10.5281/zenodo.15003981

August, T., Shin, H., & Tunca, T. I. (2018). Generating value through open source: Software service market regulation and licensing policy. *Information Systems Research*, *29*(1), 186–205.

Bandara, W., Gable, G. G., & Rosemann, M. (2005). Factors and measures of business process modelling: model building through a multiple case study. *European Journal of Information Systems*, *14*(4), 347–360.

Bapna, R., Gupta, A., Rice, S., & Sundararajan, A. (2017). Trust and the strength of ties in online social networks. *MIS quarterly*, *41*(1), 115–130.

Bigliardi, L., Lanza, M., Bacchelli, A., D'Ambros, M., & Mocci, A. (2014). Quantitatively exploring non-code software artifacts. In *Qsic* (pp. 286–295).

Blog, G. O. S. (2023). *More voices, more bazel.* Retrieved from https://opensource.googleblog.com/2023/01/more-voices-more-bazel.html (Accessed: 2024-12-20)

Bosu, A., Carver, J. C., Bird, C., Orbeck, J., & Chockley, C. (2016). Process aspects and social dynamics of contemporary code review: Insights from open source development and industrial practice at microsoft. *IEEE Transactions on Software Engineering*, *43*(1), 56–75.

Boudreau, M.-C., & Robey, D. (2005). Enacting integrated information technology: A human agency perspective. *Organization science*, *16*(1), 3–18.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative research in psychology*, *3*(2), 77–101.

Chaoss: Community health analytics open source software. (n.d.). https://chaoss.community/. (Accessed: 2025-01-02)

Christa, S., Madhusudhan, V., Suma, V., & Rao, J. J. (2017). Software maintenance: From the perspective of effort and cost requirement. In *Icdect* (pp. 759–768).

Churchill Jr, G. A. (1979). A paradigm for developing better measures of marketing constructs. *Journal of marketing research*, *16*(1), 64–73.

Cloud Native Computing Foundation (CNCF). (2024). *Cncf people repository.* Retrieved from https://github.com/cncf/people (Accessed: 2024-12-20)

Contributor Roles Taxonomy. (2022). *CRediT - Contributor Roles Taxonomy.* Retrieved from https://credit.niso.org/ (Accessed: 2024-10-02)

Crain, M., Poster, W., & Cherry, M. (2016). *Invisible labor: Hidden work in the contemporary world.* Univ of California Press.

Cuccuru, G., Leo, S., Lianas, L., Muggiri, M., Pinna, A., Pireddu, L., ... Zanetti, G. (2014). An automated infrastructure to support high-throughput bioinformatics. In *Hpcs* (pp. 600–607).

Dabbish, L., Stuart, C., Tsay, J., & Herbsleb, J. (2012). Social coding in github: transparency and collaboration in an open software repository. In *Cscw* (pp. 1277–1286).

Daniels, A. K. (1987). Invisible work. *Social problems*, *34*(5), 403–415.

Dubé, L., & Paré, G. (2003). Rigor in information systems positivist case research: current practices, trends, and recommendations. *MIS quarterly*, 597–636.

Eisenhardt, K. M. (1989). Building theories from case study research. *Academy of management review*, *14*(4), 532–550.

Emery, F. E., & Trist, E. L. (1960). Socio-technical systems. *Management science, models and techniques*, *2*, 83–97.

Experience, C. (2024). *Introduction to contributor experience.* Retrieved from https://contributor-experience.org/docs/guide/cx/introduction.html (Accessed: 2024-12-20)

Feedly. (2024). *Feedly.* Website. Retrieved from https://feedly.com (Accessed: 2024-06-11)

Feller, J., Finnegan, P., Fitzgerald, B., & Hayes, J. (2008). From peer production to productization: A study of socially enabled business exchanges in open source service networks. *Information systems research*, *19*(4), 475–493.

Feng, Z. (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* (pp. 2213–2215).

Feng, Z., Chatterjee, A., Sarma, A., & Ahmed, I. (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* (pp. 797–809).

Feng, Z., Kimura, K., Trinkenreich, B., Sarma, A., & Steinmacher, I. (2024). Guiding the way: A systematic literature review on mentoring practices in open source software projects. *IST*, 107470.

Feng, Z., Steinmacher, I., Gerosa, M., Menezes, T., Serebrenik, A., Milewicz, R., & Sarma, A. (2025). The multifaceted nature of mentoring in oss: Strategies, qualities, and ideal outcomes. *arXiv preprint arXiv:2501.05600*.

FOSSASIA. (2024). *FOSSASIA: Open Technology for Social Change*. `https://fossasia.org/`. ([Accessed: 2024-12-20])

Garousi, V., Felderer, M., & Mäntylä, M. V. (2019). Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. *Information and software technology*, *106*, 101–121.

Geiger, R. S., Howard, D., & Irani, L. (2021). The labor of maintaining and scaling free and open-source software projects. *Proceedings of the ACM on human-computer interaction*, *5*(CSCW1), 1–28.

Google Open Source. (2024). *Google open source blog*. Blog. Retrieved from `https://opensource.googleblog.com` (Accessed: 2024-06-11)

Google Open Source Programs Office. (2023). *The business impact of open source*. Retrieved from `https://opensource.org/blog/google-open-source-programs-office-the-bu` (Accessed: 2024-12-12)

Gregor, S. (2006). The nature of theory in information systems. *MIS quarterly*, 611–642.

Guzzi, A., Bacchelli, A., Lanza, M., Pinzger, M., & Van Deursen, A. (2013). Communication in open source software development mailing lists. In *Msr* (pp. 277–286).

Hatton, E. (2017). Mechanisms of invisibility: rethinking the concept of invisible work. *Work, employment and society*, *31*(2), 336–351.

ILO, D. (2012). Statistical update on employment in the informal economy. *Imbens, GW (2014). Instrumental variables: An econometricians perspective*.

Kane, G. C., Alavi, M., Labianca, G., & Borgatti, S. P. (2014). What's different about social media networks? a framework and research agenda. *MIS quarterly*, *38*(1), 275–304.

Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering–a systematic literature review. *IST*, *51*(1), 7–15.

Klein, H. K., & Myers, M. D. (1999). A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS quarterly*, 67–93.

Kontio, J., Bragge, J., & Lehtola, L. (2008). The focus group method as an empirical tool in software engineering. In *Guide to advanced empirical software engineering* (pp. 93–116). Springer.

Leidner, D. E., & Kayworth, T. (2006). A review of culture in information systems research: Toward a theory of information technology culture conflict. *MIS quarterly*, 357–399.

Link, C. (2010). Patterns for the commercial use of open source: legal and licensing aspects. In *Europlop* (pp. 1–10).

Linux Foundation. (2024). *Open Source Summit North America.* `https://events.linuxfoundation.org/open-source-summit-north-america/`. ([Accessed: 2024-12-20])

Lyulina, E., & Jahanshahi, M. (2021). Building the collaboration graph of open-source software ecosystem. In *Msr* (pp. 618–620).

Maruping, L. M., & Matook, S. (2020). The multiplex nature of the customer representative role in agile information systems development. *MIS Quarterly*, *44*(3).

Medappa, P. K., & Srivastava, S. C. (2019). Does superposition influence the success of floss projects? an examination of open-source software development by organizations and individuals. *Information Systems Research*, *30*(3), 764–786.

Metric: Types of contributions. (2024). CHAOSS Community Knowledge Base. Retrieved from `https://chaoss.community/kb/metric-types-of-contributions/`

(Accessed: 2024-10-02)

Nakakoji, K., Yamada, K., & Giaccardi, E. (2005). Understanding the nature of collaboration in open-source software development. In *Apsec* (pp. 8–pp).

NISO CRediT. (2024). *Credit - contributor roles taxonomy.* Retrieved from https://credit.niso.org/ (Accessed: 2024-12-20)

Open Source Initiative. (2024). *Open source initiative blog.* Blog. Retrieved from https://opensource.org/blog (Accessed: 2024-06-11)

Opensource.com. (2020). *What is a glue team and why does your organization need one?* Retrieved from https://opensource.com/open-organization/20/6/glue-team (Accessed: 2024-12-20)

Open source maintainers: What they need and how to support them. (n.d.). https://www.linuxfoundation.org/blog/open-source-maintainers-what-the (Accessed: 2024-03-01)

OpenSSF Community. (2024). *Openssf blog guidelines.* Retrieved from https://openssf.org/community/blog-guidelines/ (Accessed: 2024-12-20)

Phang, C. W., Kankanhalli, A., & Tan, B. C. (2015). What motivates contributors vs. lurkers? an investigation of online feedback forums. *Information Systems Research*, *26*(4), 773–792.

Poster, W. R., Crain, M., & Cherry, M. A. (2016). Introduction: Conceptualizing invisible labor. *Invisible labor: Hidden work in the contemporary world*, 3–27.

Qiu, H. S., Li, Y. L., Padala, S., Sarma, A., & Vasilescu, B. (2019). The signals that potential contributors look for when choosing open-source projects. *PACMHCI*, *3*(CSCW), 1–29.

Rad, R. (2021, June). *Top github repo trends in 2021.* Retrieved from https://rajko-rad.medium.com/top-github-repo-trends-in-2021-e4fa0c724 (Medium)

Raiffa, H. (2007). *Negotiation analysis: The science and art of collaborative decision*

making. Harvard University Press.

Rehman, I., Wang, D., Kula, R. G., Ishio, T., & Matsumoto, K. (2020). Newcomer candidate: Characterizing contributions of a novice developer to github. In *Icsme* (pp. 855–855).

Reilly, T. (2019). *Being glue.* `https://www.noidea.dog/glue`. (Accessed: 2025-01-01)

Rico, S. (2020). Exploring and improving industry-academia communication in software engineering. In *Ease* (pp. 379–382).

Sarker, S., Xiao, X., & Beaulieu, T. (2013). Guest editorial: Qualitative studies in information systems: A critical review and some guiding principles. *MIS quarterly*, *37*(4), iii–xviii.

Sarma, A., & Chen, N. (2024). Effective teaching through code reviews: Patterns and anti-patterns. , *1*(FSE), 1262–1283.

Shehab, E., Sharp, M., Supramaniam, L., & Spedding, T. A. (2004). Enterprise resource planning: An integrative review. *Business process management journal*, *10*(4), 359–386.

Singh, P. V., & Phelps, C. (2013). Networks, social influence, and the choice among competing innovations: Insights from open source software licenses. *Information Systems Research*, *24*(3), 539–560.

Sjøberg, D. I., Anda, B., & Mockus, A. (2012). Questioning software maintenance metrics: a comparative case study. In *Esem* (pp. 107–110).

Slaughter, S. A., Levine, L., Ramesh, B., Pries-Heje, J., & Baskerville, R. (2006). Aligning software processes with strategy. *Mis Quarterly*, 891–918.

Smirnova, I., Reitzig, M., & Alexy, O. (2022). What makes the right oss contributor tick? treatments to motivate high-skilled developers. *Research Policy*, *51*(1), 104368.

Software Freedom Conservancy. (2024). *Software freedom conservancy blog.* Blog. Retrieved from `https://sfconservancy.org/blog/` (Accessed: 2024-06-11)

Squire, M. (2015). "should we move to stack overflow?" measuring the utility of social media for developer support. In *Icse* (Vol. 2, pp. 219–228).

Srivastava, S. C., & Chandra, S. (2018). Social presence in virtual world collaboration. *MIS quarterly*, *42*(3), 779–A16.

Steinmacher, I., Conte, T., Gerosa, M. A., & Redmiles, D. (2015). Social barriers faced by newcomers placing their first contribution in open source software projects. In *Cscw* (pp. 1379–1392).

Stohl, C., Stohl, M., & Leonardi, P. M. (2016). Digital age| managing opacity: Information visibility and the paradox of transparency in the digital age. *International Journal of Communication*, *10*, 15.

Straub, D., Boudreau, M.-C., & Gefen, D. (2004). Validation guidelines for is positivist research. *Communications of the Association for Information systems*, *13*(1), 24.

Sutanto, J., Kankanhalli, A., & Tan, B. C. (2014). Uncovering the relationship between oss user support networks and oss popularity. *Decision Support Systems*, *64*, 142–151.

Tan, L., Liu, C., Li, Z., Wang, X., Zhou, Y., & Zhai, C. (2014). Bug characteristics in open source software. *EMSE*, *19*, 1665–1705.

Trinkenreich, B., Guizani, M., Wiese, I., Conte, T., Gerosa, M., Sarma, A., & Steinmacher, I. (2021). Pots of gold at the end of the rainbow: what is success for open source contributors? *IEEE Transactions on Software Engineering*, *48*(10), 3940–3953.

Trinkenreich, B., Guizani, M., Wiese, I., Sarma, A., & Steinmacher, I. (2020). Hidden figures: Roles and pathways of successful oss contributors. *Proceedings of the ACM on human-computer interaction*, *4*(CSCW2), 1–22.

Trinkenreich, B., Wiese, I., Sarma, A., Gerosa, M., & Steinmacher, I. (2022). Women's participation in open source software: A survey of the literature. *TOSEM*, *31*(4), 1–37.

Vasilescu, B., Posnett, D., Ray, B., van den Brand, M. G., Serebrenik, A., Devanbu, P., & Filkov, V. (2015). Gender and tenure diversity in github teams. In *Proceedings of the 33rd annual acm conference on human factors in computing systems* (pp. 3789–3798).

Venkatesh, V., Brown, S. A., & Sullivan, Y. W. (2016). Guidelines for conducting mixed-methods research: An extension and illustration. *Journal of the Association for Information Systems*, *17*(7), 2.

Von Krogh, G., Spaeth, S., & Lakhani, K. R. (2003). Community, joining, and specialization in open source software innovation: a case study. *Research policy*, *32*(7), 1217–1241.

Walsham, G. (1995). Interpretive case studies in is research: nature and method. *European*

Journal of information systems, *4*(2), 74–81.

Webster, J., & Watson, R. T. (2002). Analyzing the past to prepare for the future: Writing a literature review. *MIS quarterly*, xiii–xxiii.

Wen, S.-F. (2017). Software security in open source development: A systematic literature review. In *2017 21st conference of open innovations association (fruct)* (pp. 364–373).

Wessel, M., Serebrenik, A., Wiese, I., Steinmacher, I., & Gerosa, M. A. (2022). Quality gatekeepers: investigating the effects of code review bots on pull request activities. *EMSE*, *27*(5), 108.

Yeats, D. (2008). The role for technical communicators in open-source software development. *Technical Communication*, *55*(1), 38–48.

Young, J.-G., Casari, A., McLaughlin, K., Trujillo, M. Z., Hébert-Dufresne, L., & Bagrow, J. P. (2021). Which contributions count? analysis of attribution in open source. In *Msr* (pp. 242–253).

YouTube. (2023). *Mendonça, pawson, tamir - contributor experience: Why it matters | scipy 2023*. Retrieved from `https://www.youtube.com/watch?v=uytmM0ulG6E` (Accessed: 2024-12-20)

Zhou, M., & Mockus, A. (2012). What make long term contributors: Willingness and opportunity in oss community. In *Icse* (pp. 518–528).