Schreiber, Roland Robert; Wieland, Thomas

Inter-organizational collaborations in open-source software ecosystems

Date of secondary publication: 15.01.2026

Version of Record (Published Version), Article

Persistent identifier: urn:nbn:de:bvb:473-irb-112591x

Primary publication

Schreiber, Roland Robert; Wieland, Thomas (2026): Inter-organizational collaborations in open-source software ecosystems, in: The journal of systems and software : JSS, Amsterdam [u.a.]: Elsevier, Vol. 235, Nr. 112765, pp. 1–18, doi: 10.1016/j.jss.2025.112765.

The Journal of Systems and Software 235 (2026) 112765

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

Inter-organizational collaborations in open-source software ecosystems

Roland Robert Schreiber [a,*], Thomas Wieland [b]

[a] *University of Bamberg: Otto-Friedrich-Universitat Bamberg, Bamberg, Germany*
[b] *University of Applied Sciences Coburg: Hochschule Coburg, Coburg, Germany*

ARTICLE INFO

Editor: Professor Tao Zhang

Keywords:
Social network analysis
Open-source
Organizational influence
Software development project
Open-source ecosystems
Coopetition
Evolution

ABSTRACT

Context: Open-source software (OSS) ecosystems are pivotal to modern software development by fostering innovation through inter-organizational collaboration. While becoming increasingly important, research on the patterns and dynamics of this collaboration in OSS remains limited.
Objective: This study investigates inter-organizational collaboration within different OSS ecosystems, focusing on identifying key contributing and influential organizations, analyzing collaboration patterns, and understanding their evolution over time.
Method: An exploratory case study with a mixed-methods approach was conducted, with data collected from five prominent OSS ecosystems on GitHub (React, Vue, TensorFlow, Bootstrap, and Flutter) involving 9947 developers and 339 organizations over ten years. Both qualitative and quantitative analyses were performed, including software repository mining and applying different levels (micro/macro) and dimensions (rational/structural/functional) of social network analysis (SNA) to examine collaboration patterns.
Results: Key organizations, such as Alphabet, TensorFlow, Meta, Nvidia, Intel, MobileIron, IBM, and AMD, emerged as significant contributors, acting as central hubs for collaboration within OSS ecosystems. The collaboration networks showed an initial phase of rapid growth followed by stabilization, indicating project maturation. Notably, competing firms were found to contribute to the same ecosystems, underpinning the inherently cooperative, reciprocal, and multiplex nature of OSS development.
Conclusion: The findings highlight the pivotal role of top-contributing organizations in fostering collaboration and driving innovation within OSS ecosystems. Understanding these dynamics provides valuable insights for developing strategies to enhance the effectiveness, scalability, and long-term sustainability of OSS ecosystems, benefiting both individual contributors and participating organizations.

1. Introduction

Modern enterprises gain a competitive advantage by developing unique software solutions that drive innovation and deliver long-term benefits, but still face continual pressure to improve quality and functionality (Crowston et al., 2012; El-Haddadeh, 2020; Nylén and Holmström, 2015). Open-source software (OSS) has emerged as a key enabler of increased productivity, cost reduction, and community-driven innovation (Harhoff et al., 2003; Nagle, 2019; Scholtes et al., 2016).

Many organizations now engage in collaborative OSS ecosystems, partnering with other companies, universities, and even rivals; this combination of cooperation and competition is a phenomenon known as “coopetition” (Bengtsson and Kock, 2000; Nguyen Duc et al., 2017). Alphabet and Meta, for example, collaborate on OSS projects such as TensorFlow despite that they are direct competitors in artificial intelligence (AI) and machine learning (ML) (Schreiber, 2023).

Inter-organizational collaboration offers significant advantages, such as shared expertise and resource pooling, but also presents challenges, including coordination across time zones and managing standards (Chen and Pan, 2014; Dorn et al., 2016; Nguyen Duc et al., 2017; Teixeira and Lin, 2014; Xia et al., 2016). As software complexity increases, collaborative development is crucial for successful project execution (Damasiotis et al., 2018; Mao et al., 2017). The failure of software development projects is often attributed to poor communication and interaction among customers, developers, users, and project managers (Franken et al., 2015; Serrador and Pinto, 2015). Since inter-organizational collaboration plays a central role in OSS ecosystems, it is essential to understand how these collaborations are structured and how they evolve over time. To enhance the effectiveness and

---

* Corresponding author.
E-mail address: roland-robert.schreiber@stud.uni-bamberg.de (R.R. Schreiber).

https://doi.org/10.1016/j.jss.2025.112765
Received 31 May 2025; Received in revised form 21 November 2025; Accepted 29 December 2025

efficiency of software development, it is crucial to inspect and manage social networks purposefully. Social networks, in this context, are structures of relationships (ties) between individuals or organizations (nodes) that collaborate in software development projects and often emerge organically rather than being explicitly designed or managed. Social network analysis (SNA) is a valuable instrument for uncovering hidden structural issues and collaboration patterns in software development projects, which is paramount (Fischbach et al., 2009; Schreiber and Zylka, 2020; Zheng et al., 2016).

In this research, we have examined inter-organizational collaborations in various OSS ecosystems, employing a mixed-methods and case study approach (Rainer and Wohlin, 2024; Yin, 2011) that integrates qualitative data from public documentation with quantitative data from software repository mining and different social network metrics. A ten-year time span was chosen to capture long-term trends, project maturation phases, and the evolution of collaboration patterns, providing a robust basis for understanding the dynamics of OSS communities. The SNA was applied at different levels (micro/macro) and dimensions (rational/structural/functional) to provide a comprehensive understanding of the collaboration networks. Through this blended methodological approach, we have acquired profound insights, leveraged synergy effects, and minimized biases associated with single-method approaches (Creswell and Creswell, 2017; Mertens, 2023).

The key contributions of this research include:

- A multi-ecosystem analysis of inter-organizational collaboration across five prominent OSS projects;
- A ten-year, longitudinal SNA study of collaboration dynamics and evolution;
- Empirical evidence on coopetition between major technology firms (e.g., Alphabet, Meta, Nvidia);
- A cross-case comparative analysis identifying structural patterns and differences across OSS ecosystems;
- A theoretical synthesis bridging social network theory and platform governance.

The paper is organized as follows. Section 2 presents the theoretical background and related work on OSS ecosystems and SNA. In Section 3, we formulate the research questions and delineate the research focus. Section 4 is a detailed description of the case study methodology, encompassing data collection and research methods. Section 5 elaborates on the results of the OSS project case study and our evaluation of these results. Section 6 is a discussion of our findings. Section 7 addresses the limitations of our study. In Section 8, we present significant conclusions.

2. Background and related work

2.1. Open-Source software ecosystems

This section provides the conceptual foundation by outlining key characteristics of OSS ecosystems. Open collaboration in software development, particularly in OSS ecosystems, has become a significant area of research due to its profound impact on innovation and digital infrastructure. Over the past two decades, OSS ecosystems have established themselves as a cornerstone of the software industry, providing essential components and infrastructures such as operating systems, libraries, component repositories, and complete platforms. Their significance continues to grow as they adapt to new technological and security demands, further accelerating innovation (Eckert et al., 2019; Mergel, 2015).

This study distinguishes clearly between open-source projects and open-source ecosystems. An OSS project refers to a specific software repository (e.g., facebook/react or tensorflow/tensorflow) that serves as the technical core of collaborative development. In contrast, an OSS ecosystem comprises the broader socio-technical network of related projects, organizations, and contributors interacting around this core. By defining these boundaries explicitly, we analyze project-level collaboration structures as integral components of their respective ecosystems.

Unlike traditional closed-source development, OSS projects are licensed in ways that encourage open collaboration, allowing anyone to study, change, and distribute the software (Baltes and Diehl, 2019; Laurent, 2004). Transparency, collaboration, and openness are foundational principles that enable developers to collectively improve software (Guerrero et al., 2019; Sedera et al., 2016). This collaborative ethos accelerates development, facilitates bug fixes, and ensures continuous improvement (Adomako and Nguyen, 2023; Couñago-Blanco et al., 2024; Le Pennec and Raufflet, 2018; Vivona et al., 2023). To achieve a competitive advantage, companies often seek heterogeneous external resources and establish valuable relationships through OSS participation (Guerrero et al., 2019). Software forges such as GitHub are useful in OSS communities (Cosentino et al., 2017), offering tools for version control, social interaction, documentation, and application programming interface (API) support (Guendouz et al., 2015; Rashid and Prakash, 2022; Squire, 2014).

The diversity of decision-making structures that govern projects is an important factor in OSS ecosystems. These structures range from centralized leadership, where a single actor or core team makes key decisions, to hybrid models that integrate centralized control with community input, and to fully decentralized models that rely on community consensus (de Laat, 2007; O'Mahony and Karp, 2022). The decision-making structure of a project significantly influences how contributors interact, form collaborations, and make contributions in OSS development (Newton and Fiore, 2024).

The variety of licensing models, which significantly influence usability and distribution, is another crucial aspect of OSS ecosystems (Almeida et al., 2019; Santos, 2017). These licenses range from more restrictive models such as the GNU General Public License (GPL), which requires derivative works to also be released under the GPL, to more permissive licenses such as the MIT, Apache, and Berkeley Software Distribution (BSD) licenses, which offer greater flexibility, especially for commercial applications (Shahrivar et al., 2018). The choice of license has substantial implications for the possibility of commercial use and for how companies interact with OSS projects (Lindman et al., 2010).

Inter-organizational collaboration is a key feature of OSS ecosystems. Developers from different organizations, including competitors, work seamlessly together, fostering innovation and knowledge exchange (Bonaccorsi and Rossi, 2006; Gonzalez-Barahona et al., 2013). Establishing relationships to gain a strategic edge over rivals is another important focus (Guerrero et al., 2019). These coopetitive relationships are complex, challenging to manage, and often lead to organizational conflicts (Bengtsson and Kock, 2014; Tidström, 2009). Understanding the dynamics of inter-organizational collaboration within OSS developments is essential for harnessing its potential (Runeson et al., 2021).

2.2. Introduction to social network analysis

To delve deeper into OSS collaboration, we turn to SNA, which provides a powerful lens through which we can examine the intricate web of relationships among developers, organizations, and ecosystems. By mapping interactions, identifying key players, and analyzing network structures, SNA sheds light on collaboration patterns and their implications (Borgatti and Foster, 2003; Wasserman and Faust, 1994). With different SNA methods, researchers gain valuable insights into the dynamics, reveal hidden structural issues, top influencers, and collaboration patterns of OSS landscapes, which can aid in ensuring success (Camacho et al., 2020; Fischbach et al., 2009; Šmite et al., 2017).

The SNA literature highlights the importance of understanding both the macro and micro levels of social networks (Claridge, 2020). The macro-structure level analyzes the entire network by partitioning it based on specific characteristics of vertices (Duxbury, 2024). Micro-structural (ego) network analysis examines the roles of individual actors within the overall network. This implies that micro-level interactions give rise to macro-level network structures and are essential for achieving a deeper understanding of SNA research (Bolíbar, 2016; Stadtfeld et al., 2020).

Furthermore, social networks at various levels have different dimensions characterized by rational, structural, and functional properties. The rational dimension focuses on the links between individuals and can be described in terms of, for example, intensity, reciprocity, and multiplexity. Structural properties, such as density and the size of a network, correspond to the morphology of relationships between actors. Functional dimensions refer to transactional content, such as how two developers might exchange knowledge or information. For our research, we identified the factors in Table 1 as relevant to the different dimensions of SNA at different levels of analysis (see Table 1).

Many social network theories are based on basic social network properties, dimensions, and levels (see Table 2). By integrating theoretical concepts with practical analysis, our study contributes to a deeper understanding of inter-organizational collaborations in OSS ecosystems by examining both macro and micro levels and their interplay with rational, structural, and functional dimensions.

2.3. Application of SNA in software engineering research

There is considerable research into the features of the communication networks of community members and the structural characteristics of developer collaboration networks in OSS ecosystems. The development of software in large OSS landscapes is knowledge-intensive, human-centered, globally distributed, as well as across time zones, and the interactions between the members are complex and self-organized (Behfar et al., 2018; Shah, 2006). Furthermore, OSS frequently lack publicly available definitions of project responsibilities and group structures, especially in smaller or less mature projects (Bock et al., 2023; Crowston and Howison, 2005; Tamburri et al., 2019).

Effective coordination among software developers in OSS ecosystems is crucial for maintaining software quality and promoting sustainable evolution. Previous research has extensively examined the social project structure, highlighting the importance of core-periphery dynamics and cluster segment analysis (Cataldo and Herbsleb, 2012; Kalliamvakou et al., 2015; Kwan et al., 2011). For example, Cataldo and Herbsleb 2012 (Cataldo and Herbsleb, 2012) found that cohesive social ties among core developers significantly enhance productivity. While these studies provide valuable insights into internal team dynamics, our research addresses a gap by extending this understanding to inter-organizational collaborations. Specifically, we reveal how competing firms such as Alphabet and Meta cooperate within the same OSS projects, a phenomenon that has received limited attention in the current literature (Nguyen Duc et al., 2017). This broader focus not only confirms the importance of network centrality but also underscores the role of coopetition in driving innovation across organizational boundaries.

In addition, research has focused mainly on social project structure, coordination, and clique analysis, which includes the topics of core-periphery (core team and enhanced team) and cluster segment analysis (Concas et al., 2017; El Asri et al., 2017; Jergensen et al., 2011; Joblin et al., 2017b; Yu et al., 2016) in particular. Core developers are crucial for ecosystems because they possess key knowledge, hold central positions, contribute significantly, and shape the culture of the communities (Jergensen et al., 2011; Joblin et al., 2017a; Toral et al., 2010; Wang et al., 2020). Other findings support that cohesive social ties between team members in their social networks lead to greater productivity (Allaho et al., 2013; Bird et al., 2008; Singh et al., 2011). There are also publications focused on prediction to support various software engineering processes such as the bug triage (Jeong et al., 2009), defect prediction (Bird et al., 2009), software vulnerability coordination (Ruohonen et al., 2018; Schreiber, 2024), leadership change forecasting (Howison et al., 2006), automatic core-developer identification (Bock et al., 2023), and build failure prediction (Wolf et al., 2009).

Table 1
Factors to the different dimensions and levels of SNA.

Dimension Level	Rational dimension	Structural dimension	Functional dimension
Micro (Ego)	• Reciprocity measures the extent to which relationships in a social network are mutual, indicating the bidirectional nature of connections between nodes (Wasserman and Faust, 1994). • Multiplexity The number of different types of content or relationships that exist within a single tie between two nodes in a network (Monge and Contractor, 2003). • Trust (Cooperations) Trust in a network context refers to the confidence one node has in the reliability, integrity, and competence of another node, influencing the strength and quality of their relationship (Buskens, 2002). • Homogeneity/ Homophily The degree of similarity between nodes (McPherson et al., 2001).	• Bridge An individual node that provides the only link between multiple clusters (Tichy et al., 1979).	• Transactional content Transactional content at the micro level refers to the specific types of exchanges, such as information, coding, and services, that occur through the link between two individual nodes in a network (Monge and Contractor, 2003).
Macro	• Intensity The strength of the relationship between nodes in a network (Scott, 2013). • Stability in a network context refers to the persistence and resilience of the network over time (Stadtfeld et al., 2020).	• Density The number of ties divided by the total number of possible links (Wasserman and Faust, 1994). • Size The total number of nodes (Wasserman and Faust, 1994). • Degree centrality The number of direct connections a node has in a network (Freeman, 1978).	• Transactional content Transactional content refers to the specific types of exchanges at the macro level, such as developing together between different nodes in a network (Monge and Contractor, 2003).

There is, though, a lack of research on the dynamics of social networks within various OSS ecosystems, particularly concerning activities of developers from different organizations. Gaining a deeper understanding of these inter-organizational interactions is crucial for building and supporting OSS ecosystems. There is growing interest in research aimed at uncovering and understanding the informal structures within

Table 2
Relevant social network theories.

Theory concept	Definition
Clique analysis	Clique analysis identifies subsets of closely connected nodes within a network, revealing how larger social structures are composed of small, cohesive components (Gupta et al., 2022).
Structural holes	Structural holes refer to gaps between different parts of a network that can be bridged by a node, which provides it with brokerage opportunities and access to complementary sources of information (Burt, 2009).
Strength of weak ties	The strength of weak ties theory proposes that weak ties are crucial for accessing diverse information and opportunities across different social circles (Granovetter, 1977).
Power-law distribution	Power-law distribution in networks describes a phenomenon where a few nodes (hubs) have many connections, while most nodes have relatively few, and follow a scale-free distribution pattern (Barabási and Albert, 1999).
Small world	Small-world experiments examine the shortest average path length for social networks, meaning most nodes can be reached from any other node through a small number of steps (Singh, 2007).

these ecosystems (Herbold et al., 2021; Schreiber and Zylka, 2020). There is also a lack of work on the formation of subgroups, productivity, participation of different organizations in evolution, company strategy, and organizational behavior within software communities (Bock et al., 2023; Herbold et al., 2021; McClean et al., 2021; Osborne et al., 2024; Schreiber, 2024, 2023; Schreiber and Zylka, 2020).

In summary, while significant progress has been made in understanding the internal dynamics of OSS projects, there remains a critical gap in the literature concerning inter-organizational collaborations at both the macro and micro levels. Addressing these research gaps is essential for fostering sustainable OSS ecosystems that can leverage collective expertise.

3. Research questions

Our research questions (RQs) address inter-organizational collaborations in OSS ecosystems involving developers and organizations. We examine the forms of collaboration, cooperation, and competition within OSS development across ecosystems.

RQ1. Which top organizations contribute to the selected OSS ecosystems?

RQ2. What patterns of collaboration between organizations can be observed in OSS ecosystems?

RQ3. How does collaboration evolve in the selected OSS project ecosystems over time?

RQ4. Do competing institutions contribute to the same OSS ecosystems?

4. Methodology

This study follows a qualitative multiple-case study design, as outlined by R.K. Yin's method (Yin, 2011) and Runeson's method (Runeson, 2012), combining qualitative and quantitative analyses of mined software repositories through SNA and public data. This strategy is well-suited for exploratory research (Runeson, 2012), especially when investigating complex, real-world phenomena such as the dynamics of OSS landscapes. To conduct this analysis, we selected five prominent OSS ecosystem projects from the GitHub Universe, based on criteria such as popularity (measured by the number of stars), industrial relevance, and diversity of participation (involvement of multiple organizations). The ecosystems were chosen purposefully to serve as the sample for our multi-case study (Yin, 2017) (see Table 3).

The selection aimed to cover a broad spectrum of open-source software domains and technologies that are widely adopted in practice. All five projects are highly active and recognized as reference implementations within their respective fields (e.g., web frameworks, machine learning, UI toolkits). They vary in terms of project age, governance structures (e.g., community-driven vs. corporate-led), technical scope, and licensing models, which ensures a diverse and balanced sample for cross-case comparison. Furthermore, each project attracted substantial organizational participation, making them well-suited for investigating inter-organizational collaboration patterns over time. Following this logic of analytical generalizability (Yin, 2011), this purposeful variation allows us to explore inter-organizational collaboration across different technological and governance contexts rather than aiming for statistical representativeness. Fig. 1 outlines the research framework, illustrating the steps involved in data collection and analysis.

4.1. Data collection

For data collection, we focused on the primary project within each software ecosystem while acknowledging the presence of diverse subprojects within these projects. For instance, the *facebook/react* repository serves as the main entry point for the React ecosystem, while *tensorflow/tensorflow* represents the core of the TensorFlow ecosystem. This operationalization ensures consistency across cases and allows us to analyze collaboration patterns at the most relevant point of organizational engagement.

For the input data of SNA, we mined the logs of the selected software repositories to obtain deeper insights into the respective ecosystem communities. The Git system records all the commits of the OSS ecosystem, typically along with comments, and documents them in the changelog, and so we extracted the relevant information for our study from the changelog. We then identified developer affiliations based on code commits, established links between developers who worked on the same files to identify and quantify their collaboration, and constructed a social network that visualizes these collaboration patterns.

To ensure methodological consistency across ecosystems, this study focuses exclusively on collaborations manifested through shared source

Table 3
Descriptive statistics of selected projects (as of March 16, 2024).

Project Name	*Stars*	*Investigated Time Periods*	*Language*	*Project Domain*	*Project License*	*Description*
react	220,163	2014 – 2023	JavaScript	JavaScript Library	MIT License	A library for web and native user interfaces.
vue	206,592	2014 – 2023	TypeScript	JavaScript User Interface Framework	MIT License	Vue.js is a progressive, incrementally adoptable JavaScript framework for building user interfaces on the web.
tensorflow	181,316	2014 – 2023	C++, Python	Machine Learning Framework	Apache License	An OSS machine learning and neural network framework.
bootstrap	167,090	2014 – 2023	JavaScript	Web Development	MIT License	The most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first projects on the web.
flutter	160,702	2014 – 2023	Dart	User Interface Developer Kit	BSD-3-Clause License	Facilitates the rapid development of aesthetically pleasing mobile and cross-platform applications.

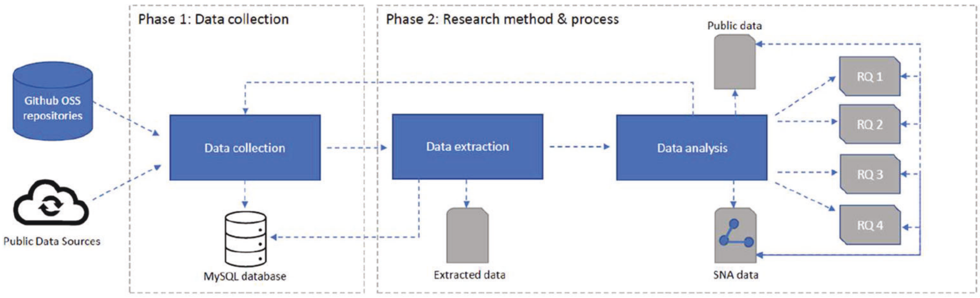


Fig. 1. Research framework.

code modifications recorded in Git commits. Other collaboration forms such as pull requests, issue tracking, or code reviews were intentionally excluded because their structure, accessibility, and availability vary substantially across projects and repositories. This deliberate scope allows for a consistent, reproducible, and quantifiable operationalization of collaboration across all selected ecosystems. Nevertheless, we acknowledge that these non-code-based interactions represent valuable coordination mechanisms and therefore constitute an important direction for future research.

To complement this technical data, our screening for this case study also incorporates publicly available data from various sources to verify affiliations and project context. We identified and selected the most relevant public announcements, white papers, blogs, and publicly available documentation.

4.2. *Research method*

The empirical data from the entire extraction process span from 2014 to the end of 2023, covering ten full calendar years. This extensive timeframe allows us to capture long-term trends and patterns in OSS ecosystems. We employed a multitude of techniques to obtain and validate developer affiliations at the time of the commit they made to the selected projects. To assign developers via email addresses, we used disambiguation heuristics (Oliva et al., 2012) and performed additional checks and manual corrections. Developers were mapped to organizations based on verified email domains and commit metadata. In cases where contributors used multiple domains or changed employers during the observation period, affiliations were assigned according to the majority of verified commits within each time slice. Contributors whose affiliation could not be reliably determined were temporarily classified as ‘volunteers’ until manual verification. About 8 % of all developer records required manual verification or correction after automated matching, which helped ensure consistent and reliable affiliation data across all ecosystems.

In order to analyze the various projects’ trace data together, independently of the different release cycles and in line with existing SNA guidelines (Howison et al., 2011), we divided the data we obtained into different time slices. This enabled us to conduct a deeper, comparative, and standardized analysis of OSS ecosystems and existing collaboration patterns. Each time slice covers one calendar year, aligning with prior SNA-based studies on OSS evolution (Howison et al., 2011; Teixeira et al., 2015). Annual segmentation offers a balanced level of temporal granularity, sufficiently fine-grained to capture evolving collaboration dynamics, yet broad enough to minimize noise from short-term fluctuations such as release cycles or temporary activity spikes. For visualization purposes, Fig. 8 and Table 6 aggregate data into two-year intervals to improve readability and highlight longer-term structural shifts. This aggregation affects only the graphical representation and does not alter the underlying yearly analytical results. Our methodology specifically captures collaborations only when developers worked on the same source code within the same observed time slice. This approach allowed us to construct collaboration networks for both developers and organizations specific to each time slice, facilitating a detailed and segmented analysis of collaboration patterns. To ensure the traceability and persistence of our data and methodology, we stored all collected data systematically in a MySQL database. For analyzing collaboration patterns within the complex whole network while minimizing the impact of outliers, we concentrated our research on developers affiliated with the top-eight contributing institutions to the selected projects over the entire observation period. This focus established a clear and well-defined scope for our study, allowing us to delve deeper into significant collaboration trends and patterns within the OSS ecosystems.

Edges between developers or organizations were established whenever two contributors modified the same file within the same time slice. Each edge was weighted by the frequency of such co-edits. To avoid artificial inflation of connections, only pairs with at least one shared edit event were included (threshold ≥ 1). For network construction and validation, only active contributors with at least one verified co-edit event within each time slice were included in the collaboration network. Developers who contributed only once or without any shared edits were excluded to prevent distortion of network density and centrality metrics. Inactive nodes (degree $= 0$) were filtered out before network visualization and metric computation to preserve analytical validity. Network validation was conducted through cross-time consistency checks and random subsampling to ensure that the observed structural patterns (e.g., degree distributions and density trends) remained stable over time. All extracted data were stored in a normalized MySQL schema designed for reproducibility, including developer identifiers, commit metadata, and affiliation mappings.

5. Results

5.1. *RQ1 – which top organizations contribute to the selected OSS ecosystems?*

To address RQ1, we examined the relationships between the macro-level structural characteristics of the developer community and its associated organization across the selected ecosystems. To assess inter-organizational collaboration, we applied established network metrics that capture structural properties of developer interaction. The number of ties indicates the total number of unique collaboration links between developers, reflecting the overall connectivity of the network. The average degree measures how many connections each developer has on average, representing collaboration intensity. Network density is defined as the ratio between actual ties and all possible ties in the network, capturing how tightly connected the developer community is. These metrics allow us to track the evolution of collaboration activity over time. The metrics analyzed include the total number of developers, the number of ties, average degree, and network density. Table 4 summarizes the results.

The developer count increased sharply from 2014 to 2017, peaking at 1718 in 2017, followed by stabilization between 1100 - 1600 developers. Both the number of ties and average degree rose until 2020, reflecting increased collaboration intensity. The number of ties peaked

Table 4
Basic network metrics - Developer network analysis.

Year Description	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023
Total number of Developers	463	709	1279	1718	1624	1602	1644	1393	1227	1130
Number of ties	3009	5319	15,552	26,558	27,317	31,738	35,411	25,996	21,583	21,903
Average Degree	13.931	16.291	26.072	32.828	36.253	42.43	45.603	39.033	37.052	40.115
Density	0.032	0.025	0.022	0.02	0.024	0.028	0.029	0.029	0.032	0.037

at 35,411 in 2020, with the highest average degree at 45.603, suggesting strategic organizational investments during this period. Network density initially declined, which is expected with network growth. However, it began to rise again in 2018, reaching 0.037 in 2023, which indicates a tighter-knit network over time due to increased collaboration among developers.

We examined the contributions of volunteer developers compared to organizational developers over different periods to better understand their respective roles within the selected ecosystems.

Fig. 2 illustrates the trend in developer involvement over time. While volunteer developers have consistently represented the largest proportion of contributors, organizational involvement grew significantly, particularly after 2017. This shift suggests an increasing role for structured, institution-driven contributions within OSS ecosystems.

We next calculated the proportions of commits made by volunteers and organizations. Fig. 3 illustrates the proportion of commits made by institutional developers versus volunteer developers. Over the years, there was a noticeable shift, with organizational contributions increasing significantly, especially after 2017. Over time, the OSS projects transitioned from a volunteer-driven community to ones with substantial organizational involvement, with institutional contributions outpacing those of volunteers.

To further understand the contribution dynamics, we plotted a Lorenz curve to depict the distribution of contributions among different organizations (Fig. 4). Applying the Lorenz curve, an established method for illustrating economic inequality (Davies et al., 2022), provides valuable insights into the distribution dynamics within the OSS landscape.

The Lorenz curve in Fig. 4 shows the distribution of contributions among the 339 identified organizations in the OSS ecosystems over the complete observation period. It underscores the significant inequality in the distribution of contributions, revealing that a small number of companies accounted for the majority of contributions. The curve's deviation from the diagonal line (line of equality) clearly illustrates this substantial disparity. This is further quantified by the Gini coefficient of 0.984, which indicates a high level of inequality in the distribution of contributions (Blesch et al., 2022; Siddiq et al., 2023). Such a coefficient indicates that a few dominant organizations played a crucial role in OSS development, driving key innovations and shaping the direction of these ecosystems. This uneven contribution pattern may not solely result from strategic priorities but may also reflect adoption challenges faced by many companies, such as limited component choices, time pressure, and the absence of formal evaluation processes, which can hinder broader organizational engagement in OSS ecosystems (Butler et al., 2022).

Our findings reveal that eight institutions played significant roles in contributing to the source code of the OSS ecosystems (see Table 5).

To obtain deeper insights, we analyzed the functional dimension of the top contributing organizations in terms of commits, encompassing both source code and comments. Notably, Alphabet and TensorFlow led in programming-related commits, reflecting their extensive involvement in core development activities. Furthermore, the top contributing institutions dominated not only in the number of commits but also in critical categories such as bug fixes and continuous programming on the related OSS software projects.

To complement our quantitative findings, official statements from the top-contributing organizations offer valuable insights into their primary strategic motivations and the perceived benefits of participating in OSS ecosystems. We conducted a qualitative content analysis of official public sources, including company blogs, press releases, white papers, keynote speeches, and developer documentation. These sources were systematically screened for statements explaining why the respective organizations engage in open-source development.

The collected statements reveal the alignment between organization goals and their active involvement in open-source projects. Alphabet ("Google Open Source," 2024), TensorFlow ("Contribute to TensorFlow, " 2023), Meta ("Meta Open Source," 2024), Nvidia (Cohen, 2023), Intel ("Open Ecosystem," 2024), MobileIron ("Ivanti," 2024), IBM ("Story," 2020), and AMD ("AMD Research Open-Source Projects," 2024) have all publicly emphasized their commitment to OSS as a means to drive innovation, share technological advancements, and enhance their business operations across various domains. Fig. 5 is a detailed categorization of the main strategic motivations of top-contributing

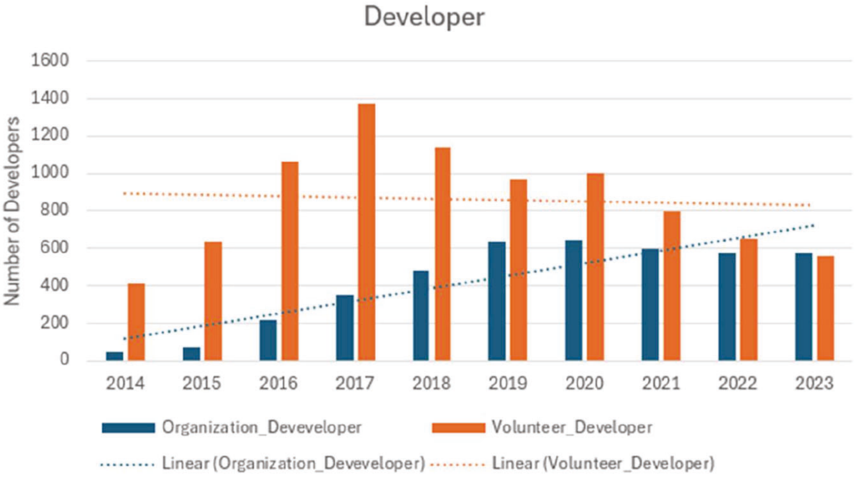


Fig. 2. Volunteer developer versus organizational developer.

Fig. 3. Volunteer developer commits versus organizational developer commits.

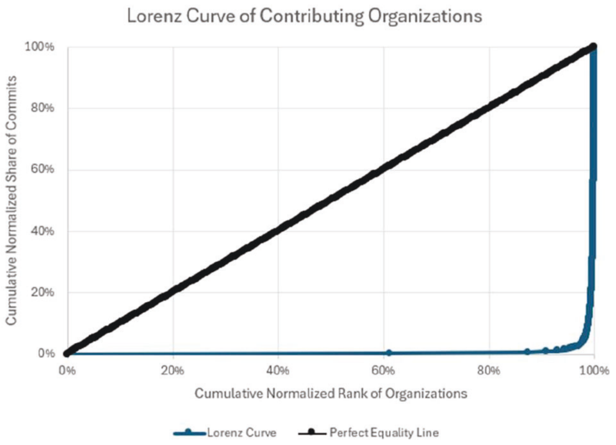


Fig. 4. Lorenz curve of contributing organizations.

Table 5
Top 8 organizations contributing to the OSS ecosystems.

Organization	Description
Alphabet	Alphabet (formerly Google) is a US-based multinational technology company that specializes in internet-related services and products, including a search engine, cloud computing, software, and hardware.
AMD	Advanced Micro Devices (AMD) is a US-based multinational semiconductor firm that develops computer processors, graphics cards, and related technologies for business and consumer markets.
IBM	International Business Machines (IBM) is a multinational technology, software, and consulting company that provides hardware, middleware, and software, as well as hosting and consulting services.
Intel	Intel is a US-based multinational technology corporation known primarily for its semiconductor chips, microprocessors, and related technology used in computers and data centers.
Meta	Meta (formerly Facebook) is a US-based multinational technology conglomerate focusing on social media platforms, virtual reality, and the development of the metaverse, as well as artificial intelligence.
MobileIron	MobileIron is an American software company specializing in enterprise mobility management, providing solutions for secure access, management, and protection of mobile devices and applications.
Nvidia	Nvidia is a US-based multinational technology company best known for its graphics processing units (GPUs), which are used extensively in gaming, artificial intelligence, and high-performance computing.
TensorFlow	TensorFlow is an open-source organization, originally founded by Google, that offers a comprehensive ecosystem for building and deploying machine learning models. It operates as a collaborative organization, driving innovation through its developer and active contributions.

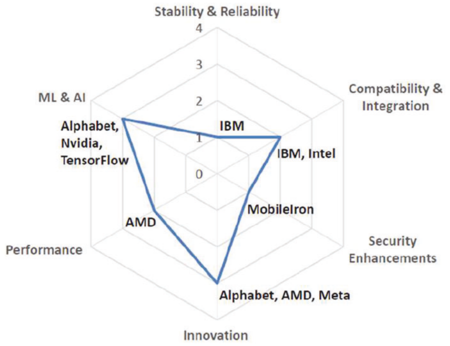


Fig. 5. Strategic motivations from top-contributing organizations.

institutions, highlighting key focus areas such as innovation, ML, and AI. The presence of a dimension in the figure indicates that it was explicitly emphasized in at least one official statement by the organization. The values indicate the main strength of motivation for each organization, with higher scores (further from the center) reflecting more frequent or explicit references in public statements and documentation. This approach allows us to highlight not only shared motivations across organizations, but also strategic positioning and focus differences among them.

5.2. RQ2 – what patterns of collaboration between organizations can be observed in OSS ecosystems?

The SNA analysis of inter-organizational collaborations in the OSS ecosystems was conducted at the macro level, focusing on the structural relationships between different institutions. The SNA revealed numerous nodes representing developers that are densely connected, indicating robust collaborative efforts within the OSS ecosystem. The nodes are clustered in specific regions, with each cluster representing a different OSS project. Specifically, the clusters correspond to the following projects: React (light green), Vue (green), TensorFlow (pink), Bootstrap (red), and Flutter (deep green).

The network visualization in Fig. 6 illustrates dense connections, which signify strong collaborations among developers. Key developers, who act as hubs within the network, connected and delivered contributions to different OSS projects. These central nodes played a crucial role in the ecosystem, supporting the theory of weak ties (Granovetter, 1977). Different colors represent various OSS projects, providing a visual representation of how developers were distributed. The density around specific nodes suggests that certain projects attracted more collaborative efforts.

Connections between different colored clusters reveal developers

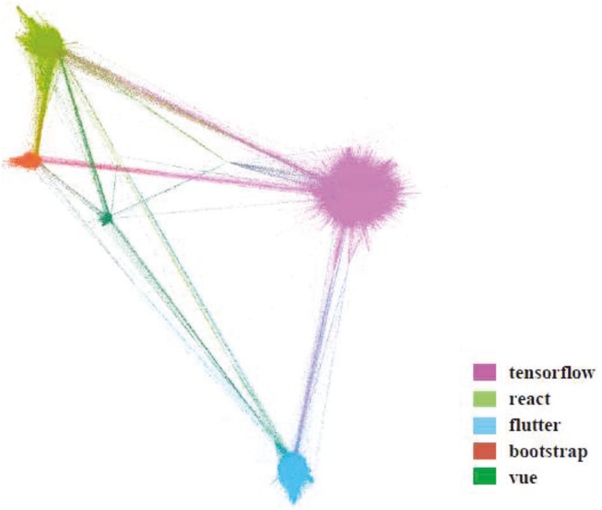


Fig. 6. Overview of the developer network OSS ecosystems.

who contributed to multiple projects. This inter-project collaboration indicates shared knowledge, tools, and goals between projects. These developers acted as bridges, facilitating the transfer of knowledge and innovations across projects, thus enhancing the overall ecosystem's robustness and supporting stability.

The presence of inter-cluster connections suggests that certain projects may be integrated or have overlapping functionalities, necessitating collaborative efforts from developers across different projects. Understanding these integration points can help in mapping the interdependencies within the OSS environment and planning for collaborative development efforts.

Next, we performed an organizational SNA to highlight collaborations between different institutions. We focused on the top-eight contributing organizations and their interactions within the ecosystem. The execution of organizational social SNA (see Fig. 7) provides detailed insights into inter-organizational collaborations. The top-eight institutions – Alphabet, TensorFlow, Meta, Nvidia, Intel, MobileIron, IBM, and AMD – were identified through Lorenz curve analysis, underscoring their significant contributions and influence within the observed OSS projects. Each node represents a contributing organization, and the size of the node reflects the number of affiliated developers contributing to the selected projects, i.e., the organization's developer-based participation level. The edges represent collaboration links, defined by shared file contributions within the same time slice. All edges are undirected and weighted by the frequency of such co-contributions. The figure spotlights the network of contributions and collaborations among organizations, with specific colors assigned to the top participating corporations.

Visible collaborations show that these top organizations frequently interacted with each other and with a broad range of smaller entities. This suggests that the collaborative environment generated significant added value and that resources were shared to drive development. Additionally, the diverse range of contributions from these organizations, spanning from core programming to bug fixes and documentation, underscores their strategic involvement in various aspects of the OSS ecosystem.

The collaborations illustrated among top institutions indicate potential alliances and partnerships. The insights gained from the SNA underscore the critical role of both central and peripheral organizations in maintaining a vibrant, stable, and dynamic OSS landscape. Central institutions such as Alphabet and TensorFlow provided leadership and drove major initiatives, while peripheral organizations contributed to different areas and ensured a diverse range of projects. The collaborative clusters identified in the analysis underscore the importance of strong partnerships in fostering innovation and achieving shared goals. The strategic alliances and partnerships among top organizations shape the direction and set priorities of OSS development in projects. For example, partnerships such as the OpenXLA initiative involving Alphabet, AMD, Intel, and Nvidia demonstrate collaborative efforts to advance machine learning frameworks ("OpenXLA Dev Lab 2024," 2024). Similarly, the AI Alliance, spearheaded by IBM and Meta, highlights the shared commitment to fostering trustworthy and open AI technologies ("AI Alliance," 2024).

5.3. *RQ3 – how does collaboration evolve in the selected OSS project ecosystem over time?*

To investigate how collaboration in the selected OSS projects evolved over time, we performed an additional macro-structural analysis using SNA. This analysis focused on the temporal dynamics of

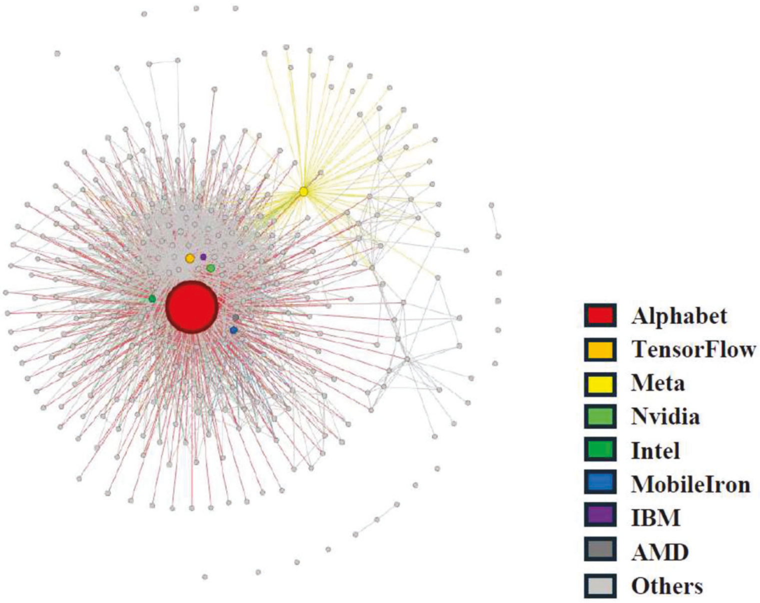


Fig. 7. Social network of organizations.

collaborations among organizations and key contributors within these ecosystems.

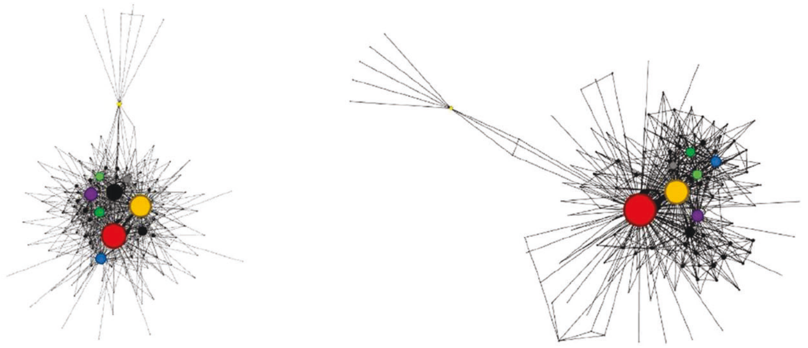
Furthermore, we highlighted the top-eight organizations – Alphabet, TensorFlow, Meta, Nvidia, Intel, MobileIron, IBM, and AMD – to show their relevance in the OSS ecosystems. By examining successive snapshots of the network taken every two years, we can identify trends, patterns, and shifts in collaborative behavior, effectively capturing long-term dynamics while filtering out minor and short-term fluctuations.

The analyzed network structures as a whole exhibit a strong core-periphery dynamic among developers and organizations. While our analysis focuses primarily on structural roles, the evolving participation of peripheral organizations across time slices suggests a continuous influx of new actors. These contributors are often involved in early testing, experimental branches, or smaller-scale contributions, which prior OSS research has linked to innovation and adaptability (Bock et al., 2023; Jergensen et al., 2011; Joblin et al., 2017a).

The core provides stability and productivity, whereas the periphery introduces fresh ideas, builds new connections, and enhances ecosystem flexibility, while also contributing valuable feedback on user needs and software quality (Crowston and Howison, 2005; El Asri et al., 2017;

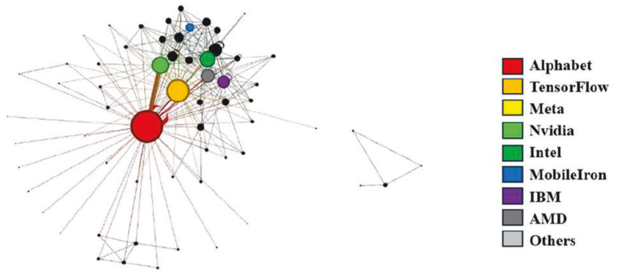
(a) 2014 - 2015

(b) 2016 - 2017



(c) 2018 - 2019

(d) 2020 - 2021



(e) 2022 - 2023

Fig. 8. Evolution of inter-organizational collaboration networks (visualized in two-year aggregated intervals, for readability; underlying analysis based on yearly data).

Joblin et al., 2017a).

The visual representation of the organization network in Fig. 8 depicting the analysis over different years provides a clear view of the changes in collaboration patterns. By comparing these snapshots, we can observe the highlights, development, and dissolution of collaborative clusters. We can also identify the emergence of new key contributors and shifts in organizational involvement over the periods.

- In 2014–2015 (see Fig. 8a), the network was relatively sparse, with a limited number of companies and connections. Few central nodes were present, indicating that key contributors or organizations had not yet emerged as prominent leaders in the ecosystem.
- By 2016–2017 (see Fig. 8b), there was a noticeable increase in the number of nodes and connections. Some nodes began to stand out as central figures, indicating the appearance of key contributors and firms. The network shows early signs of clustering, with groups of institutions beginning to form collaborative sub-networks.
- In 2018–2019 (see Fig. 8c), the network reached its peak in terms of the number of organizations and connections. Clusters were well-formed, indicating robust collaboration among groups of institutions. Prominent central nodes were evident, highlighting the leading roles of key contributors in driving collaboration and development.
- By 2020–2021 (see Fig. 8d), a slight decline in the number of nodes and connections was observed. Despite the reduction, the network remained well-clustered with strong collaborative sub-networks. Central nodes continued to play a pivotal role in maintaining the network's structure and facilitating interactions.
- In 2022–2023 (see Fig. 8e), the number of nodes decreased significantly, reflecting a consolidation phase in the ecosystem. The remaining organizations were highly interconnected, with dense connections indicating intensive collaboration. Central nodes became more pronounced, underscoring the influence of a few key groups in sustaining the ecosystem.

The analysis of network metrics over time provides quantitative insights into the evolution of collaboration in the OSS ecosystems. Table 6 presents basic network metrics for the organization network analysis from 2014 to 2023, including the total number of organizations, the number of ties, average degree, and density.

From 2014 to 2023, the number of institutions and ties in these OSS projects fluctuated, peaking in 2018 before declining. The average degree increased until 2018, followed by a slight decline. Network density varied, with the highest value in 2022. These temporal trends provide valuable insights into the evolution of collaboration within the OSS ecosystems, highlighting phases of growth, peak collaboration, and eventual consolidation. The analysis revealed a high concentration of connections among a few nodes, underscoring the critical role of key organizations in maintaining network robustness.

5.4. RQ4 – do competing institutions contribute to the same OSS ecosystem?

To address research question 4, we conducted a micro-structural analysis focusing on the contributions of competing institutions to the same OSS ecosystems. Specifically, we used ego network analysis to examine the evolution of collaborations involving the top-eight contributing organizations.

Table 6
Basic network metrics - Organization network analysis (aggregated in two-year intervals for readability; underlying analysis based on yearly data).

Description Year	Total number of Organizations	Number of ties	Average Degree	Density
2014–2015	30	33	2.20	0.076
2016–2017	115	293	5.09	0.045
2018–2019	155	623	8.03	0.052
2020–2021	139	537	7.73	0.056
2022–2023	73	245	6.71	0.093

Fig. 9 extends the ego network analysis for the top eight contributing organizations, offering a comprehensive view of how these entities engage within and across OSS ecosystems. Each node represents one organization from our case study (e.g., IBM, AMD, Alphabet, etc.). Edges represent observed collaboration between organizations, defined as joint contributions to the same OSS ecosystem during the same time period. Node size reflects the number of ecosystems an organization is involved in, capturing the breadth of its engagement. Edge thickness indicates the intensity of collaboration, based on the number of ecosystems where both connected organizations are active contributors.

Fig. 9a shows Alphabet's significant connections with competitors such as AMD and Nvidia, indicating robust collaborative efforts despite market competition. In Fig. 9b, TensorFlow's network reveals strong ties with both Alphabet and Nvidia, highlighting its central role in collaborative OSS projects. Fig. 9c shows Meta's extensive network, emphasizing its collaborative efforts with Alphabet, Tensorflow, and other key players, demonstrating a strategic approach to leverage shared resources. Fig. 9d shows that Nvidia's network included strong ties with Alphabet, Tensorflow, and Intel, indicating active participation in collaborative development. Fig. 9e showcases Intel's significant connections with Alphabet, Tensorflow, and MobileIron underscoring its strategic collaborations in the OSS ecosystem. Fig. 9f, which highlights MobileIron's network; although less extensive than the others, still shows meaningful collaborations with Alphabet and Intel. Fig. 9g features IBM's strong ties with Alphabet, Intel and Nvidia, indicating IBM's active role in the collaborative landscape. Finally, Fig. 9h shows AMD's network; despite being smaller than others, it includes key connections with Nvidia and Alphabet, reflecting AMD's involvement in collaborative efforts.

The collaborative efforts among competing companies within the OSS ecosystem enhance innovation by combining diverse expertise and resources. This collective approach leads to the development of more robust and versatile OSS projects. Organizations benefit mutually from these collaborations, gaining access to shared knowledge and advancements that would be challenging to achieve independently. This mutual benefit reinforces the value of contributing to OSS ecosystems.

All network metrics, figures, and supplementary materials underlying the analyses and results presented in this paper are provided in the replication package accompanying this submission. The package includes methodological documentation, longitudinal datasets, and visualizations supporting the cross-ecosystem analyses.

6. Discussion

This study provides valuable insights into the dynamics of inter-organizational collaboration in OSS ecosystems by examining both macro- and micro-level interactions through SNA across five major projects over a ten-year period. By integrating relevant factors across different dimensions and levels of SNA, along with various theoretical frameworks such as structural holes, weak ties, clique analysis, power-law distribution, and small-world networks, we offer a comprehensive understanding of how these ecosystems evolve and adapt. In the following, we discuss the results in relation to our research questions and place them in the context of existing literature and relevant theoretical frameworks.

6.1. Top-contributing organizations in OSS ecosystems

The analysis of contribution data across the five OSS ecosystems revealed that a small group of organizations account for a substantial share of activity. In particular, companies such as Alphabet, Meta, Nvidia, Intel, IBM, AMD, MobileIron, and contributors affiliated with TensorFlow organization emerged as central players. These

(a) Alphabet (b) TensorFlow

(c) Meta (d) Nvidia

Fig. 9. Ego organization network analysis.

organizations consistently occupied core positions in the collaboration networks, as indicated by high degree centrality and betweenness scores across the entire ten-year period.

This result is in line with prior research on the dominance of large firms in open-source development (Gonzalez-Barahona et al., 2013; Schreiber, 2023), and confirms that corporate contributions are not only extensive but also strategically positioned to influence the direction of OSS projects.

From a network-theoretical perspective, these actors serve as brokers bridging structural holes (Burt, 2009), connecting otherwise disconnected segments of the network and enabling the flow of knowledge, tools, and practices across organizational boundaries. Their bridging roles enhance the overall resilience and integration of the ecosystem, particularly in complex, modular codebases (Crowston et al., 2012).

One notable structural feature is the power-law distribution of contributions, reflected in a Gini coefficient of 0.984. This indicates a high concentration of contributions among a few organizations, a phenomenon well known in OSS and other networked systems (Barabási and Albert, 1999). While such concentration can promote efficiency and stability, it also raises concerns about dependency and influence asymmetries.

Interestingly, the dominance of these central organizations does not preclude the participation of smaller entities and independent contributors. However, their peripheral positions in the network suggest limited access to strategic decision-making or architectural influence. This finding underlines the importance of governance mechanisms and inclusive practices to maintain openness even as projects scale and centralize.

In summary, the presence of top-contributing organizations is essential to the functioning and growth of OSS ecosystems. These actors drive technical innovation and provide coordination capacity, but their influence must be balanced with openness to ensure long-term sustainability.

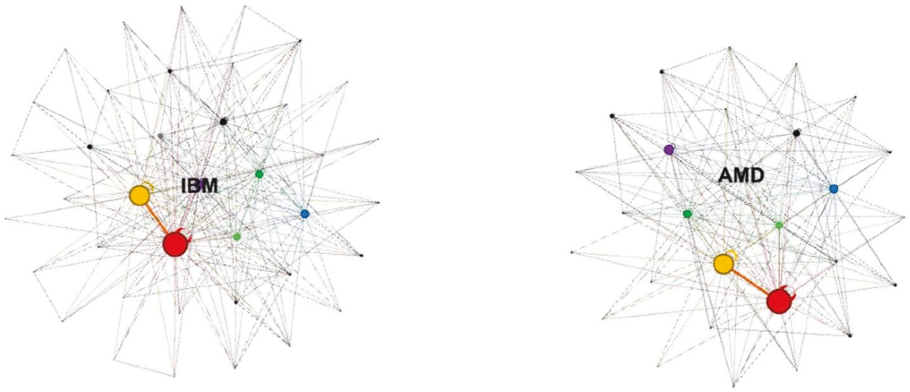
6.2. *Inter-organizational collaboration patterns*

In response to RQ2, our analysis revealed several characteristic patterns of collaboration between organizations within OSS ecosystems. These patterns vary across ecosystems but share certain recurring structural and relational features.

Across the five ecosystems studied, the networks exhibit a core-periphery structure, in which a few dominant organizations such as Alphabet, Meta, and Nvidia maintain dense, multidirectional collaborations, while many peripheral organizations are only loosely connected or linked through a single core actor. This confirms previous observations in OSS research that collaboration tends to centralize over time around organizational hubs (Crowston and Howison, 2005; Joblin et al., 2017b, 2017a).

(e) Intel

(f) MobileIron



(g) IBM

(h) AMD

Fig. 9. (*continued*).

Another notable feature is the presence of structural holes between clusters of organizations, particularly across different technological or regional domains. Central organizations often acted as brokers, maintaining ties to multiple otherwise unconnected actors (Long et al., 2013). This reflects the theory that brokers enhance network efficiency by controlling access to information and opportunities, which in OSS contexts translates to access to architectural decisions and knowledge diffusion (Burt, 2009; Serafino et al., 2021).

Interestingly, we found low levels of homophily in organizational collaboration. Firms from different regions, industries, and sizes frequently engaged in code-level cooperation, suggesting that OSS ecosystems support cross-boundary openness. This is consistent with the collaborative ethos of open-source development and contrasts with findings in intra-organizational software engineering, where team structures tend to be more homogeneous (McPherson et al., 2001).

However, the data indicate limited reciprocity among organizations. Although some dyadic ties between firms are reciprocal, many others exhibit asymmetric collaboration, where one organization (often a smaller player) contributes to shared resources without a balanced exchange. This could be explained by differences in strategy, available resources, or the role of the organization within the ecosystem (e.g., user vs. platform provider).

Finally, multiplexity, defined as the presence of multiple layers of interaction between the same actors was observed primarily among core contributors. These organizations not only co-develop source code but also engage in testing, documentation, community support, and standardization efforts. This richness of interaction types may serve to stabilize relationships and reinforce long-term collaboration, as suggested in prior work on network resilience in software ecosystems (Šmite et al., 2017).

Taken together, these collaboration patterns highlight the dual nature of OSS ecosystems: they are at once open and inclusive, yet structured and strategically coordinated. While dominant actors drive coordination and integration, peripheral contributors broaden the ecosystem's diversity and reach.

6.3. *Temporal dynamics of collaboration in OSS ecosystems*

To answer RQ3, we examined the evolution of inter-organizational collaboration over a ten-year period across the selected OSS ecosystems. The analysis revealed clear temporal patterns that suggest a shift from loosely connected collaboration structures to increasingly dense and strategically coordinated networks.

Across all ecosystems, we observed a consistent increase in network density and clustering coefficients over time, indicating stronger and more interconnected relationships among participating organizations. This development reflects a broader trend towards ecosystem consolidation, in which initial openness and experimentation give way to stable alliances and formalized roles (Crowston et al., 2012). While prior studies have investigated the technical evolution of microservice-based OSS projects (Assunção et al., 2023), our work emphasizes how organizational structures and collaboration networks co-evolve with these systems, shaping the intensity, structure, and long-term sustainability of inter-organizational collaboration.

For instance, in early project phases (e.g., 2014–2016), collaboration was characterized by sporadic co-contributions and many disconnected organizational nodes. In contrast, during the later stages (e.g., post-2019), we found the emergence of dense subgraphs and cohesive clusters, particularly in ecosystems like TensorFlow and Flutter. These clusters often centered around key technology providers and industry leaders.

The growing centrality of core organizations – notably Alphabet, Meta, and Intel – suggests a process of institutional dominance, where these actors not only maintain high activity levels but also increasingly coordinate and integrate contributions from others. This is consistent with research on the centralizing forces in maturing OSS ecosystems (Runeson et al., 2021).

Moreover, the networks exhibit a persistent small-world structure over time, characterized by short path lengths and high clustering. This topology facilitates efficient information diffusion and collaboration while maintaining structural robustness, a finding that aligns with the scale-free network model (Barabási and Albert, 1999).

These phases suggest that the maturity and popularity of an OSS ecosystem influence not only the number of contributors but also the nature of their interactions. Strategic alliances, long-term coordination, and division of labor become more pronounced as projects grow. In this context, projects that meet key quality and technical criteria are more likely to attract sustained organizational investments in OSS ecosystems (Li et al., 2022).

The nature of collaboration can be generalized from our study across different OSS ecosystems, and this evolution can be represented in distinct different phases, as Fig. 10 shows. Fig. 10 illustrates the methodological phases of the analysis, rather than aggregated temporal intervals, and summarizes how collaboration structures evolved across distinct evolutionary stages. For instance, the TensorFlow project exhibited a high degree of collaboration among developers, indicating a mature and stable project, whereas newer projects such as Flutter showed less dense collaboration networks.

The general phases include an Early Phase, characterized by initial collaborations. This is followed by a Growth Phase, marked by increased alliances with key contributors. The Expansion Phase sees dense clustering and the inclusion of specialized actors, while the Consolidation Phase focuses on stabilizing and maintaining the quality of collaborations. This suggests that the maturity, size and popularity of an OSS project may influence the extent and nature of collaboration (Barabási and Albert, 1999; Crowston et al., 2012). As projects evolve and reach higher levels of maturity, they often spawn subprojects or extensions in which innovative additions are no longer directly integrated into the central branch but develop independently (Jullien et al., 2025). These subprojects allow for specialized innovation while maintaining the stability and focus of the main project. Newer projects should focus on building strong community engagement, while mature projects should aim to maintain their collaborative environment and encourage new contributors to take on leadership roles.

In summary, collaboration in OSS ecosystems evolves from open, experimental beginnings toward more structured and coordinated relationships, driven by both technical complexity and the strategic behavior of key players.

6.4. *Coopetition: collaboration between competing institutions*

In response to RQ4, our findings confirm that competing firms frequently collaborate within the same OSS ecosystems. This phenomenon of coopetition is particularly evident in projects such as TensorFlow and Flutter, where Alphabet, Meta, Nvidia, AMD, and other major players contribute to shared codebases while maintaining direct competition in areas like AI, cloud computing, and mobile platforms.

To better understand the nature of these relationships, we analyzed the strength of inter-organizational ties, measured by the frequency of shared file edits and co-commit events. These high-frequency interactions indicate not only mutual technical alignment but also intentional coordination, despite the underlying market rivalry.

Comparative analysis revealed that coopetitive ties differ from typical collaborations in several ways:

- They exhibit higher multiplexity, often extending beyond code to documentation, issue tracking, and community engagement.
- They tend to form more persistent and stable connections over time.
- They show low reciprocity asymmetry, meaning that contributions are more balanced compared to peripheral-to-core collaborations.

Temporal analysis further shows that these strategic relationships often intensify during periods of technological change or standardization, such as the rollout of major machine learning APIs or cross-platform frameworks. For instance, increased collaboration density coincided with the introduction of major technological initiatives such as OpenXLA (Alphabet, AMD, Intel, Nvidia) and the AI Alliance (IBM, Meta). These cases illustrate how external events and strategic alliances triggered new patterns of inter-organizational engagement and reshaped ecosystem governance structures.

These patterns suggest that coopetition in OSS is not incidental, but rather a strategic response to shared challenges and innovation goals. While our results reveal collaboration among firms that also compete in related markets, we acknowledge that such patterns indicate, rather than conclusively prove, the presence of strategic coopetition. In this study, coopetition is inferred from a combination of structural evidence,

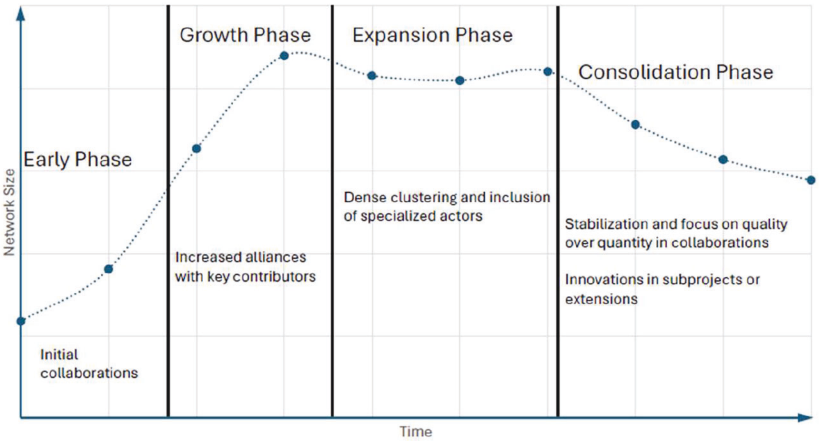


Fig. 10. General phases of OSS ecosystem evolution.

as manifested in shared code contributions and sustained cross-firm interactions, and contextual indicators derived from publicly available corporate communications (e.g., strategic statements on innovation sharing and open standardization efforts). This approach ensures that our interpretation remains theoretically grounded while recognizing the methodological limits of inferring strategic intent from network-based data and official statements. Firms appear to leverage open-source projects as neutral platforms where pre-competitive collaboration can take place, lowering development costs, shaping standards, and accessing shared talent pools.

This aligns with earlier studies on OSS as a vehicle for cross-organizational alignment (Teixeira and Lin, 2014; West and Gallagher, 2006) and complements prior SNA-based analyses of industrial OSS ecosystems that revealed similar event-driven collaboration patterns (Teixeira et al., 2015). This study extends the literature by offering a quantitative, longitudinal perspective on how coopetition manifests over time and across ecosystems.

In conclusion, coopetition in OSS ecosystems is a stable and structured phenomenon, characterized by strong, balanced, and persistent ties between otherwise competing organizations. This underscores the role of OSS as a coordination mechanism beyond traditional firm boundaries, enabling innovation without central control.

While these patterns illustrate the strategic nature of coopetition within individual ecosystems, the following section synthesizes these findings across all five cases to identify common structural logics and contextual variations.

6.5. Cross-case comparative analysis

Synthesizing the results across the five OSS ecosystems reveals several recurring collaboration patterns that transcend individual project contexts. While each ecosystem exhibits distinct characteristics shaped by its governance, technical scope, network analysis, and participant community, the comparative perspective highlights both converging structural logics and context-specific deviations.

Across all ecosystems, a stable core-periphery configuration emerged, where a small set of central organizations drive the majority of contributions, supported by a large peripheral base of independent developers. This structure reflects a balance between coordination and openness, allowing firms to maintain architectural and governance control while leveraging the distributed innovation capacity of the wider community. Such a pattern is consistent with prior research on network governance in open-source ecosystems, emphasizing hybrid coordination mechanisms that combine hierarchical and collaborative elements.

Governance structures appear to influence the degree of cohesion and persistence in inter-organizational collaboration. Ecosystems with centralized governance, such as TensorFlow and Flutter, exhibit higher network cohesion, denser inter-firm connections, and more persistent collaborative ties. This stability results from clearer organizational ownership and strategic alignment among leading firms. In contrast, community-driven ecosystems, such as Vue and Bootstrap, show more decentralized and dynamic collaboration patterns. These networks attract broader individual participation but experience higher turnover and less enduring inter-organizational coordination, reflecting the looser coupling typical of volunteer-driven development. The governance type was inferred from publicly documented project governance.

Organizational diversity also influences overall network density and knowledge diffusion. Ecosystems involving a heterogeneous mix of corporate and independent contributors (e.g., React and TensorFlow) display broader cross-firm connectivity, indicating a higher potential for multiplex interaction (conceptually understood as overlapping collaboration roles rather than measured across multiple interaction channels). Diverse participation enhances mutual learning and technological interoperability, aligning with open innovation theory, which posits that heterogeneous knowledge sources strengthen collective innovation capacity. Conversely, ecosystems with fewer organizational stakeholders tend to develop more cohesive but narrower collaboration clusters.

Domain-specific factors further explain variation in collaboration intensity. Projects situated in rapidly evolving technological domains, such as machine learning or cross-platform frameworks, tend to show stronger inter-firm coordination to manage architectural complexity and evolving standards. This explains the higher persistence of collaboration ties in TensorFlow and Flutter compared to web framework projects such as Vue or Bootstrap, where modular architectures enable greater independence among contributors.

Taken together, these cross-case insights suggest that inter-organizational collaboration in OSS ecosystems follows consistent macro-structural logics while varying in degree and persistence according to governance type, organizational diversity, and technological domain. The comparative analysis illustrates that collaboration structures in open-source ecosystems are not idiosyncratic to single projects but reflect systematic mechanisms of coordination, competition, and shared innovation. By integrating these patterns across ecosystems, this study contributes a comprehensive understanding of how governance and diversity jointly shape enduring collaboration architectures in open innovation contexts.

These findings align with the open-source ecosystem perspective on inter-organizational collaboration (Teixeira et al., 2015; Teixeira and Lin, 2014) and coopetition research (Bengtsson and Kock, 2014), both of which emphasize hybrid coordination mechanisms and mutual interdependence among participating organizations, thereby underscoring the defining characteristics of OSS ecosystems.

6.6. Implications

OSS ecosystems serve as neutral platforms where developers collaborate toward shared goals, such as technological advancement, support, and the integration of modern technologies. This environment benefits not only the developers and companies involved but also the broader user community. Developers appear to embrace the concept of coopetition, collaborating across corporate and organizational boundaries in ways that transcend traditional competitive dynamics.

The theoretical implications of our study extend the application of concepts such as structural holes, weak ties, and clique analysis to OSS ecosystems. These network theories provide a robust framework for understanding the evolution and resilience of collaboration networks. By examining both macro- and micro-level factors, we contribute to the broader discourse on network stability and adaptability, highlighting the mechanisms that drive both growth and resilience in OSS projects.

The practical implications of our study suggest that fostering both strong organizational hubs and diverse individual contributions is crucial for sustaining innovation in OSS ecosystems. Organizations can leverage their centrality to provide strategic direction (Krogh et al., 2012) while encouraging reciprocal and diverse interactions at the individual level to maintain adaptability. By bridging structural holes, forming cliques, leveraging weak ties, fostering reciprocity, and benefiting from the small-world nature of the network, institutions and individual contributors alike enhance the adaptability and innovation potential of OSS ecosystems (Lakhani and Hippel, 2003; Scacchi et al., 2006).

7. Limitations

This study analyzes the importance of top-influencing organizations within OSS ecosystems by examining the developers' and organizations' social relationships. Based on a case study approach (Runeson, 2012; Runeson and Höst, 2009), we investigated how developers and institutions collaborated in selected complex OSS ecosystems interact. The validity of our research results may be threatened in various directions. We used the scheme of validity and threats, which distinguishes between construct validity, internal validity, external validity, and conclusion validity (Howison et al., 2011; Runeson, 2012; Wohlin, 2012).

Construct validity concerns how well our operational measures represented the research objectives. Our analysis focused exclusively on collaborations manifested through source code modifications, extracted from Git repositories. This approach captures explicit technical contributions and co-editing behavior but excludes other relevant collaboration activities such as testing, code reviews, issue tracking, pull requests, and project discussions. To ensure transparency, we explicitly acknowledge that our operationalization captures only code-based interactions. While this provides a consistent and reproducible measure across ecosystems, it does not include non-code-based coordination (e.g., pull requests, issue tracking, code reviews), which often reflect communication and decision-making processes that go beyond what is captured in code alone. Future research should therefore integrate multiple collaboration signals to offer a more comprehensive picture of inter-organizational coordination and dynamics in OSS ecosystems. In addition, the qualitative component of this study is based on secondary data sources, including official documentation, press releases, and white papers. Although this design ensures methodological transparency and cross-ecosystem comparability, it does not capture first-hand practitioner perspectives. Future research should therefore include interviews or surveys to explore organizational motivations, governance mechanisms, and collaboration challenges in greater depth. Furthermore, we identified developers based on their names and organization email addresses, which could lead to incomplete institution affiliations. To mitigate these issues, we used disambiguation heuristics for empirical studies (Oliva et al., 2012; Wiese et al., 2016) and visualized results to support our findings.

With respect to the internal validity of our study, the results could be inaccurate if we misinterpreted the SNA visualizations and other metrics. To minimize this limitation, the process was carried out with various SNA metrics and different methods. We also made the entire research as traceable and straightforward as possible.

The extent to which the findings of a study can be applied to other contexts is referred to as external validity (Yin, 2014). This research is a case study of five important OSS projects in the GitHub universe. Generalizing the results to other OSS software development ecosystems is only partially possible. The insights and findings from the research of the different OSS ecosystems can be considered as a broad foundation. To minimize this issue, we investigated five different popular and important OSS ecosystems with different sizes, project domains, programming languages, and various aims. Although our dataset captures the longitudinal evolution of collaboration, it does not explicitly model external events such as license changes or corporate restructuring. Future research should incorporate event-based indicators to establish causal relationships between external stimuli and network evolution. Additionally, we analyzed historical archival data spanning about 10 years to minimize temporary deviations in this research. Furthermore, while our analysis identifies collaboration patterns consistent with strategic coopetition, these interpretations are derived from observable structural and contextual indicators rather than direct access to organizational decision-making processes. To strengthen contextual validity, we complemented network data with qualitative evidence from publicly available corporate statements and documentation, in which firms explicitly acknowledge collaborative or strategic partnerships within OSS ecosystems. Future research should expand on this approach by incorporating interviews or internal case studies to further validate organizations' strategic motivations, intentions, and governance mechanisms underlying inter-organizational collaboration in open-source contexts.

Conclusion validity concerns drawing the correct conclusions about relationships in data. In our study, we were concerned with the ability to replicate our findings. Our combination of SNA visualizations with the help of the research software Gephi (Bastian et al., 2009) and other measures aimed to provide a more accurate view of our research results. Each step of the case study was validated, and the researchers carried out periodic reviews.

8. Conclusion

This study provides a comprehensive analysis of inter-organizational collaborations within OSS ecosystems, focusing on the contributions and interactions among developers and organizations. Through SNA and a mixed-methods approach, we examined collaboration patterns within five OSS projects over a decade, across various levels and dimensions. Our findings underscore the pivotal role of leading institutions and their strategic motivations in driving collaboration and innovation, shaping both the codebase and the collaborative culture.

Leading organizations such as Alphabet, TensorFlow, Meta, Nvidia, Intel, MobileIron, IBM, and AMD act as hubs, establishing collaborative norms that foster diverse developer participation and innovation. Initiatives such as community events, mentorship programs, and resource contributions have been instrumental in encouraging broader participation and deeper engagement, illustrating the critical role these organizations play in driving OSS collaboration and sustainability. Collaboration patterns in OSS ecosystems evolve dynamically, often characterized by rapid initial growth followed by stabilization as projects mature.

Competing companies frequently adopt a coopetition approach within OSS ecosystems, leveraging varied expertise and resources to foster innovation, stability, and ecosystem health. Contributions span activities from programming and bug fixing to documentation and testing, demonstrating strategic organizational involvement across multiple aspects of development.

Our analysis answered the four research questions by identifying central organizations, revealing evolving patterns of coopetition, and outlining collaboration dynamics across time and ecosystems.

Based on our findings, this study contributes to the literature in five central ways:

1. **It provides a cross-ecosystem perspective on inter-organizational collaboration.** By analyzing five major OSS projects, we reveal consistent structural patterns and evolution dynamics across ecosystems, going beyond the scope of single-case analyses.
2. **It offers a longitudinal view of organizational-level dynamics.** Our ten-year SNA uncovers how collaboration intensity, centralization, and coopetitive structures develop over time, highlighting phases of growth, consolidation, and stabilization.
3. **It empirically demonstrates coopetition as a stable and strategic phenomenon.** Competing firms such as Alphabet, Meta, and Nvidia maintain persistent collaborative ties, indicating intentional, long-term partnerships within OSS ecosystems.
4. **It delivers a cross-case comparative analysis that identifies structural patterns and governance differences across OSS ecosystems.** This comparative perspective highlights how differences in ecosystem governance, organizational diversity, and technical domain shape collaboration intensity and network cohesion.
5. **It synthesizes social network theory with platform ecosystem research.** By integrating constructs like structural holes, multiplexity, and weak ties with platform governance theory, we provide a novel explanation for coordination and innovation in OSS.

For organizations aiming to influence OSS ecosystems, fostering collaboration and acting as connectors between disparate developer groups are essential strategies. These insights provide practical implications for decision makers and OSS community managers to design interventions that encourage inclusive collaboration, supporting OSS project sustainability and effectiveness.

In conclusion, understanding inter-organizational collaboration in OSS ecosystems is crucial for fostering innovation and sustaining ecosystem growth. By highlighting key organizational roles and collaboration dynamics, this study offers valuable insights to strengthen OSS ecosystem resilience. Emphasizing cross-organizational collaboration drives collective progress and technological advancement within the OSS community.

Future research should examine additional OSS ecosystems and platforms, exploring longitudinal effects of collaboration. Investigating how various contribution types impact project sustainability could further clarify collaboration dynamics. Future studies could also integrate additional collaboration signals, such as pull requests, testing activities, issue or feature discussions, to capture broader organizational interaction patterns. In addition, understanding how strategic organizational approaches influence collaboration and innovation may provide actionable insights for effective, sustainable OSS development practices.

CRediT authorship contribution statement

Roland Robert Schreiber: Writing – original draft. **Thomas Wieland:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The replication package for this study is publicly available on Zenodo at https://doi.org/10.5281/zenodo.18086905.

References

Adomako, S., Nguyen, N.P., 2023. Digitalization, inter-organizational collaboration, and technology transfer. J. Technol. Transf. https://doi.org/10.1007/s10961-023-10031-z.

AI Alliance [WWW Document], 2024. . IBM Newsroom. URL https://newsroom.ibm.com/AI-Alliance-Launches-as-an-International-Community-of-Leading-Technology-Developers,-Researchers,-and-Adopters-Collaborating-Together-to-Advance-Open,-Safe,-Responsible-AI (accessed 12.1.24).

Analyzing the social ties and structure of contributors in open source software community. In: Allaho, M.Y., Lee, W.-C., Özyer, T. (Eds.), 2013. Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). ACM IEEE, Canada, pp. 56–60.

Almeida, D.A., Murphy, G.C., Wilson, G., Hoye, M., 2019. Investigating whether and how software developers understand open source software licensing. Empir. Softw. Eng 24, 211–239. https://doi.org/10.1007/s10664-018-9614-9.

AMD Research Open-Source Projects [WWW Document], 2024. . AMD. URL https://www.amd.com/en/corporate/research/open-source.html (accessed 7.23.24).

Assunção, W.K.G., Krüger, J., Mosser, S., Selaoui, S., 2023. How do microservices evolve? An empirical analysis of changes in open-source microservice repositories. J. Syst. Softw 204, 111788. https://doi.org/10.1016/j.jss.2023.111788.

Baltes, S., Diehl, S., 2019. Usage and attribution of Stack Overflow code snippets in GitHub projects. Empir. Softw. Eng 24, 1259–1295. https://doi.org/10.1007/s10664-018-9650-5.

Barabási, A.-L., Albert, R., 1999. Emergence of scaling in random networks. science 286, 509–512.

Bastian, M., Heymann, S., Jacomy, M., 2009. Gephi: an Open Source Software for Exploring and Manipulating Networks. In: Proc. Int. AAAI Conf. Web Soc. Media, 3.

Behfar, S.K., Turkina, E., Burger-Helmchen, T., 2018. Knowledge management in OSS communities: relationship between dense and sparse network structures. Int. J. Inf. Manag 38, 167–174.

Bengtsson, M., Kock, S., 2014. Coopetition—Quo vadis? Past accomplishments and future challenges. Ind. Mark. Manag.

Bengtsson, M., Kock, S., 2000. "Coopetition" in Business Networks—To Cooperate and Compete Simultaneously. Ind. Mark. Manag.

Bird, C., Pattison, D., D`Souza, R., Filkov, V., Devanbu, P. (Eds.), 2008. Latent Social Structure in Open Source Projects. SIGSOFT '08/FSE-16.

Bird, C., Rigby, P.C., Barr, E.T., Hamilton, D.J., German, D.M., Devanbu, P., 2009. The promises and perils of mining git. In: 2009 6th IEEE Int. Work. Conf. Min. Softw. Repos..

Blesch, K., Hauser, O.P., Jachimowicz, J.M., 2022. Measuring inequality beyond the Gini coefficient may clarify conflicting findings. Nat. Hum. Behav 6, 1525–1536. https://doi.org/10.1038/s41562-022-01430-7.

Bock, T., Alznauer, N., Joblin, M., Apel, S., 2023. Automatic Core-developer Identification on GitHub: a Validation Study. ACM. Trans. Softw. Eng. Methodol.

Bolíbar, M., 2016. Macro, meso, micro: broadening the 'social' of social network analysis with a mixed methods approach. Qual. Quant 50, 2217–2236. https://doi.org/10.1007/s11135-015-0259-0.

Bonaccorsi, A., Rossi, C., 2006. Comparing motivations of individual programmers and firms to take part in the open source movement: from community to business. Knowl. Technol. Policy 18, 40–64. https://doi.org/10.1007/s12130-006-1003-9.

Borgatti, S.P., Foster, P.C., 2003. The Network Paradigm in Organizational Research: a Review and Typology. J. Manag 29, 991–1013.

Burt, R.S., 2009. Structural holes: The social Structure of Competition. Harvard university press.

Buskens, V., 2002. Social Networks and Trust. Springer Science & Business Media.

Butler, S., Gamalielsson, J., Lundell, B., Brax, C., Mattsson, A., Gustavsson, T., Feist, J., Kvarnström, B., Lönroth, E., 2022. Considerations and challenges for the adoption of open source components in software-intensive businesses. J. Syst. Softw 186, 111152. https://doi.org/10.1016/j.jss.2021.111152.

Camacho, D., Panizo-LLedot, Á., Bello-Orgaz, G., Gonzalez-Pardo, A., Cambria, E., 2020. The four dimensions of social network analysis: an overview of research methods, applications, and software tools. Inf. Fusion 63, 88–120. https://doi.org/10.1016/j.inffus.2020.05.009.

Cataldo, M., Herbsleb, J.D., 2012. Coordination breakdowns and their impact on development productivity and software failures. IEEE. Trans. Softw. Eng 39, 343–360.

Chen, X., Pan, Y., 2014. The Study of Open Source Software Collaborative User Model based on Social Network and Tag Similarity. J. Electr. Commer. Res 15.

Claridge, T., 2020. Social capital at different levels and dimensions: a typology of social capital. https://doi.org/10.5281/zenodo.8016096.

Cohen, J., 2023. Right On Track: NVIDIA Open-Source Software Helps Developers Add Guardrails to AI Chatbots [WWW Document]. NVIDIA Blog. URL. https://blogs.nvidia.com/blog/ai-chatbot-guardrails-nemo/. accessed 7.23.24.

Concas, G., Marchesi, M., Monni, C., Orrù, M., Tonelli, R., 2017. Software quality and community structure in java software networks. Int. J. Softw. Eng. Knowl. Eng 27, 1063–1096.

Contribute to TensorFlow [WWW Document], 2023. . TensorFlow. URL https://www.tensorflow.org/community/contribute/sigs (accessed 7.23.24).

Cosentino, V., Izquierdo, J.L.C., Cabot, J., 2017. A systematic mapping study of software development with GitHub. IEEE. Access 5, 7173–7192.

Couñago-Blanco, E., Depino-Besada, N.I., Ferrer-Serrano, M., López-Manuel, L., 2024. Collaboration as an Enabler for Digital Transformation: the Helix Paradigm. In: Machado, C., Davim, J.P. (Eds.), Management For Digital Transformation. Springer International Publishing, Cham, pp. 161–182. https://doi.org/10.1007/978-3-031-42060-3_8.

Creswell, J.W., Creswell, J.D., 2017. Research design: Qualitative, quantitative, and Mixed Methods Approaches. Sage publications.

Crowston, K., Howison, J., 2005. The Social Structure of Free and Open Source Software Development (originally published in Volume 10, Number 2, February 2005). First Monday.

Crowston, K., Wei, K., Howison, J., Wiggins, A., 2012. Free/Libre open-source software development. ACM. Comput. Surv.

Damasiotis, V., Fitsilis, P., O'Kane, J.F., 2018. Modeling software development process complexity. Int. J. Inf. Technol. Proj. Manag. IJITPM 9, 17–40. https://doi.org/10.4018/IJITPM.2018100102.

Davies, J., Hoy, M., Zhao, L., 2022. Revisiting comparisons of income inequality when Lorenz curves intersect. Soc. Choice. Welf 58, 101–109. https://doi.org/10.1007/s00355-021-01343-w.

de Laat, P.B., 2007. Governance of open source software: state of the art. J. Manag. Gov 11, 165–177. https://doi.org/10.1007/s10997-007-9022-9.

Dorn, S., Schweiger, B., Albers, S., 2016. Levels, phases and themes of coopetition: a systematic literature review and research agenda. Eur. Manag. J 34, 484–500. https://doi.org/10.1016/j.emj.2016.02.009.

Duxbury, S.W., 2024. Micro Effects on Macro Structure in Social Networks. Sociol. Methodol 54, 1–26. https://doi.org/10.1177/00811750231209040.

Eckert, R., Stuermer, M., Myrach, T., 2019. Alone or Together? Inter-organizational affiliations of open source communities. J. Syst. Softw 149, 250–262. https://doi.org/10.1016/j.jss.2018.12.007.

El Asri, I., Kerzazi, N., Benhiba, L., Janati, M., 2017. From Periphery to Core: a Temporal Analysis of GitHub Contributors' Collaboration Network. In: Camarinha-Matos, L.M., Afsarmanesh, H. (Eds.), Collaboration in a Data-Rich World, IFIP Advances in Information and Communication Technology. Springer, Cham, pp. 217–229.

El-Haddadeh, R., 2020. Digital Innovation Dynamics Influence on Organisational Adoption: the Case of Cloud Computing Services. Inf. Syst. Front 22, 985–999. https://doi.org/10.1007/s10796-019-09912-2.

Fischbach, K., Gloor, P.A., Schoder, D., 2009. Analysis of informal communication networks–a case study. Bus. Inf. Syst. Eng 1, 140–149.

Franken, S., Kolvenbach, S., Prinz, W., Alvertis, I., Koussouris, S., 2015. CloudTeams: bridging the Gap Between Developers and Customers During Software Development Processes. In: Procedia Comput. Sci., 1st International Conference on Cloud Forward: From Distributed to Complete Computing, 68, pp. 188–195. https://doi.org/10.1016/j.procs.2015.09.234.

Freeman, L.C., 1978. Centrality in social networks conceptual clarification. Soc. Netw 1, 215–239.

Gonzalez-Barahona, J.M., Izquierdo-Cortazar, D., Maffulli, S., Robles, G., 2013. Understanding How Companies Interact with Free Software Communities. IEEE. Softw 30, 38–45. https://doi.org/10.1109/MS.2013.95.

Google Open Source [WWW Document], 2024. . Google Open Source. URL https://opensource.google/(accessed 7.23.24).

Granovetter, M.S., 1977. The strength of weak ties. Social Networks. Elsevier, pp. 347–367.

Guendouz, M., Amine, A., Hamou, R.M., 2015. Recommending Relevant Open Source Projects on GitHub using a Collaborative-Filtering Technique. Int. J. Open. Source. Softw. Process.

Guerrero, M., Urbano, D., Herrera, F., 2019. Innovation practices in emerging economies: do university partnerships matter? J. Technol. Transf 44, 615–646. https://doi.org/10.1007/s10961-017-9578-8.

Gupta, S.K., Singh, D.P., Choudhary, J., 2022. A review of clique-based overlapping community detection algorithms. Knowl. Inf. Syst 64, 2023–2058. https://doi.org/10.1007/s10115-022-01704-6.

Harhoff, D., Henkel, J., Von Hippel, E., 2003. Profiting from voluntary information spillovers: how users benefit by freely revealing their innovations. Res. Policy 32, 1753–1769.

Herbold, S., Amirfallah, A., Trautsch, F., Grabowski, J., 2021. A systematic mapping study of developer social network research. J. Syst. Softw.

Howison, J., Inoue, K., Crowston, K., 2006. Social dynamics of free and open source team communications. Open Source Systems. Springer, pp. 319–330.

Howison, J., Wiggins, A., Crowston, K., 2011. Validity issues in the use of social network analysis with digital trace data. J. Assoc. Inf. Syst 12, 2.

Ivanti [WWW Document], 2024. URL https://www.ivanti.com/blog (accessed 7.23.24).

Jeong, G., Kim, S., Zimmermann, T., 2009. Improving bug triage with bug tossing graphs. In: Proceedings of the 7th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on The Foundations of Software Engineering, pp. 111–120.

Jergensen, C., Sarma, A., Wagstrom, P., 2011. The onion patch: migration in open source ecosystems. In: Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering, ESEC/FSE '11. Association for Computing Machinery, New York, NY, USA, pp. 70–80. https://doi.org/10.1145/2025113.2025127.

Joblin, M., Apel, S., Hunsen, C., Mauerer, W., 2017a. Classifying Developers into Core and Peripheral: an Empirical Study on Count and Network Metrics. In: ICSE 2017.

Joblin, M., Apel, S., Mauerer, W., 2017b. Evolutionary trends of developer coordination: a network approach. Empir. Softw. Eng.

Jullien, N., Viseur, R., Zimmermann, J.-B., 2025. A theory of FLOSS projects and Open Source business models dynamics. J. Syst. Softw 224, 112383. https://doi.org/10.1016/j.jss.2025.112383.

Kalliamvakou, E., Damian, D., Blincoe, K., Singer, L., German, D.M., 2015. Open source-style collaborative development practices in commercial projects using github. In: 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering. IEEE, pp. 574–585.

von Krogh, G., Haefliger, S., Spaeth, S., Wallin, M., 2012. Carrots and rainbows : motivation and social practice in open source software development: management information systems. MIS. Quart 36, 649–676.

Kwan, I., Schroter, A., Damian, D., 2011. Does socio-technical congruence have an effect on software build success? a study of coordination in a software project. IEEE. Trans. Softw. Eng 37, 307–324.

Lakhani, K.R., von Hippel, E., 2003. How open source software works: "free" user-to-user assistance. Res. Policy 32, 923–943. https://doi.org/10.1016/S0048-7333(02)00095-1.

Laurent, A.M.S., 2004. Understanding Open Source and Free Software Licensing, Online-Ausg. EBL-Schweitzer. O'Reilly Media, Sebastopol.

Le Pennec, M., Raufflet, E., 2018. Value Creation in Inter-Organizational Collaboration: an Empirical Study. J. Bus. Ethics 148, 817–834. https://doi.org/10.1007/s10551-015-3012-7.

Li, X., Moreschini, S., Zhang, Z., Taibi, D., 2022. Exploring factors and metrics to select open source software components for integration: an empirical study. J. Syst. Softw 188, 111255. https://doi.org/10.1016/j.jss.2022.111255.

Lindman, J., Paajanen, A., Rossi, M., 2010. Choosing an Open Source Software License in Commercial Context: a Managerial Perspective. In: 2010 36th EUROMICRO Conference on Software Engineering and Advanced Applications. Presented at the 2010 36th EUROMICRO Conference on Software Engineering and Advanced Applications, pp. 237–244. https://doi.org/10.1109/SEAA.2010.26.

Long, J.C., Cunningham, F.C., Braithwaite, J., 2013. Bridges, Brokers and Boundary Spanners in Collaborative networks: a Systematic Review. BMC Health Services Research.

Mao, K., Capra, L., Harman, M., Jia, Y., 2017. A survey of the use of crowdsourcing in software engineering. J. Syst. Softw 126, 57–84. https://doi.org/10.1016/j.jss.2016.09.015.

McClean, K., Greer, D., Jurek-Loughrey, A., 2021. Social network analysis of open source software: a review and categorisation. Inf. Softw. Technol 130, 106442. https://doi.org/10.1016/j.infsof.2020.106442.

McPherson, M., Smith-Lovin, L., Cook, J.M., 2001. Birds of a Feather: homophily in Social Networks. Annu. Rev. Sociol 27, 415–444.

Mergel, I., 2015. Open collaboration in the public sector: the case of social coding on GitHub. Gov. Inf. Q 32, 464–472.

Mertens, D.M., 2023. Research and Evaluation in Education and psychology: Integrating diversity With quantitative, qualitative, and Mixed Methods. Sage publications.

Meta Open Source [WWW Document], 2024. URL https://opensource.fb.com/(accessed 7.23.24).

Monge, P.R., Contractor, N.S., 2003. Theories of Communication Networks. Oxford University Press, USA.

Nagle, F., 2019. Open Source Software and Firm Productivity. Manag. Sci 65, 1191–1215. https://doi.org/10.1287/mnsc.2017.2977.

Newton, O.B., Fiore, S.M., 2024. Understanding participation and corporatization in service of diversity in free/libre and open source software development projects. J. Syst. Softw 217, 112163. https://doi.org/10.1016/j.jss.2024.112163.

Nguyen Duc, A., Cruzes, D.S., Hanssen, G.K., Snarby, T., Abrahamsson, P., 2017. Coopetition of Software Firms in Open Source Software Ecosystems. In: Ojala, A., Olsson, H.H. (Eds.), Software Business, Lecture Notes in Business Information Processing. Springer, Cham, pp. 146–160.

Nylén, D., Holmström, J., 2015. Digital innovation strategy: a framework for diagnosing and improving digital product and service innovation. Bus. Horiz 58, 57–67. https://doi.org/10.1016/j.bushor.2014.09.001.

Oliva, G.A., Santana, F.W., de Oliveira, K.C.M., de Souza, C.R.B., Gerosa, M.A., 2012. Characterizing Key Developers: a Case Study with Apache Ant. In: Herskovic, V., Hoppe, H.U., Jansen, M., Ziegler, J. (Eds.), Collaboration and Technology, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 97–112. https://doi.org/10.1007/978-3-642-33284-5_8.

O'Mahony, S., Karp, R., 2022. From proprietary to collective governance: how do platform participation strategies evolve? Strateg. Manag. J 43, 530–562. https://doi.org/10.1002/smj.3150.

Open Ecosystem [WWW Document], 2024. . Intel. URL https://www.intel.com/content/www/us/en/developer/topic-technology/open/overview.html (accessed 7.23.24).

OpenXLA Dev Lab 2024, 2024. Google Open Source Blog. URL. https://opensource.googleblog.com/2024/05/openxla-dev-lab-2024-building-grouundbreaking-systems-together.html. accessed 12.1.24.

Osborne, C., Daneshyan, F., He, R., Ye, H., Zhang, Y., Zhou, M., 2024. Characterising Open Source Co-opetition in Company-hosted Open Source Software Projects: the Cases of PyTorch, TensorFlow, and Transformers. https://doi.org/10.48550/arXiv.2410.18241.

Rainer, A., Wohlin, C., 2024. Reporting case studies in systematic literature studies—an evidential problem. Inf. Softw. Technol 174, 107501. https://doi.org/10.1016/j.infsof.2024.107501.

Rashid, E., Prakash, M., 2022. An empirical analysis of inferences from commit, fork, and branch rates of top github projects. Int. J. Open. Source. Softw. Process.

Runeson, P., 2012. Case Study Research in Software engineering: Guidelines and Examples.

Runeson, P., Höst, M., 2009. Guidelines for conducting and reporting case study research in software engineering. Empir. Softw. Eng.

Runeson, P., Olsson, T., Linåker, J., 2021. Open Data Ecosystems — An empirical investigation into an emerging industry collaboration concept. J. Syst. Softw 182, 111088. https://doi.org/10.1016/j.jss.2021.111088.

Ruohonen, J., Rauti, S., Hyrynsalmi, S., Leppänen, V., 2018. A case study on software vulnerability coordination. Inf. Softw. Technol 103, 239–257. https://doi.org/10.1016/j.infsof.2018.06.005.

Santos, C.D.dos, 2017. Changes in free and open source software licenses: managerial interventions and variations on project attractiveness. J. Internet. v. Appl 8, 11. https://doi.org/10.1186/s13174-017-0062-3.

Scacchi, W., Feller, J., Fitzgerald, B., Hissam, S., Lakhani, K., 2006. Understanding free/open source software development processes. Softw. Process:. Improv. Pract. 11, 95–105.

Scholtes, I., Mavrodiev, P., Schweitzer, F., 2016. From Aristotle to Ringelmann: a large-scale analysis of team productivity and coordination in Open Source Software projects. Empir. Softw. Eng 21, 642–683. https://doi.org/10.1007/s10664-015-9406-4.

Schreiber, R.R., 2024. Organizational Influence on Security Development in Open-Source Software Projects. Int. J. Syst. Softw. Secur. Prot. IJSSSP 15, 1–20. https://doi.org/10.4018/IJSSSP.356659.

Schreiber, R.R., 2023. Organizational Influencers in Open-Source Software Projects. Int. J. Open. Source. Softw. Process. IJOSSP 14, 1–20. https://doi.org/10.4018/IJOSSP.318400.

Schreiber, R.R., Zylka, M.P., 2020. Social Network Analysis in Software Development Projects: a Systematic Literature Review. Int. J. Softw. Eng. Knowl. Eng.

Scott, J., 2013. Social Network Analysis Third edition. SAGE, London, UK.

Sedera, D., Lokuge, S., Grover, V., Sarker, Suprateek, Sarker, Saonee, 2016. Innovating with enterprise systems and digital platforms: a contingent resource-based theory view. Inf. Manage. Inf. Technol. Innov:. Driv. Chall. Impacts 53, 366–379. https://doi.org/10.1016/j.im.2016.01.001.

Serafino, M., Cimini, G., Maritan, A., Rinaldo, A., Suweis, S., Banavar, J.R., Caldarelli, G., 2021. True scale-free networks hidden by finite size effects. Proc. Natl. Acad. Sci 118, e2013825118. https://doi.org/10.1073/pnas.2013825118.

Serrador, P., Pinto, J.K., 2015. Does Agile work? — A quantitative analysis of agile project success. Int. J. Proj. Manag 33, 1040–1051. https://doi.org/10.1016/j.ijproman.2015.01.006.

Shah, S.K., 2006. Motivation, governance, and the viability of hybrid forms in open source software development. Manag. Sci 52, 1000–1014.

Shahrivar, S., Elahi, S., Hassanzadeh, A., Montazer, G., 2018. A business model for commercial open source software: a systematic literature review. Inf. Softw. Technol 103, 202–214. https://doi.org/10.1016/j.infsof.2018.06.018.

Siddiq, F.K., Klymentieva, H., Lee, T.J.C., 2023. Applying the Lorenz Curve and Gini Coefficient to Measure the Population Distribution. Int. Adv. Econ. Res 29, 177–192. https://doi.org/10.1007/s11294-023-09874-x.

Singh, P., 2007. Open source software development and the small world phenomenon: an empirical investigations of macro level colaboration network properties on project success. In: ICIS 2007 Proceedings, 18.

Singh, P.V., Tan, Y., Mookerjee, V., 2011. Network effects: the influence of structural capital on open source project success. Mis. Q 813–829.

Šmite, D., Moe, N.B., Šāblis, A., Wohlin, C., 2017. Software teams and their knowledge networks in large-scale software development. Inf. Softw. Technol.

Squire, M., 2014. Forge++: The changing Landscape of FLOSS Development.

Stadtfeld, C., Takács, K., Vörös, A., 2020. The Emergence and Stability of Groups in Social Networks. Soc. Netw. Soc. Netw. Res. Negat. Ties. Signed. Graphs 60, 129–145. https://doi.org/10.1016/j.socnet.2019.10.008.

Story [WWW Document], 2020. . IBM Dev. URL https://www.ibm.com/opensource/story/(accessed 7.23.24).

Tamburri, D.A., Palomba, F., Serebrenik, A., Zaidman, A., 2019. Discovering community patterns in open-source: a systematic approach and its evaluation. Empir. Softw. Eng 24, 1369–1417. https://doi.org/10.1007/s10664-018-9659-9.

Teixeira, J., Lin, T., 2014. Collaboration in the open-source arena: the webkit case. In: Proc. 52nd ACM Conf. Comput. People Res..

Teixeira, J., Robles, G., González-Barahona, J.M., 2015. Lessons learned from applying social network analysis on an industrial Free/Libre/Open Source Software ecosystem. J. Internet. v. Appl 6, 14. https://doi.org/10.1186/s13174-015-0028-2.

Tichy, N.M., Tushman, M.L., Fombrun, C., 1979. Social network analysis for organizations. Acad. Manage. Rev 4, 507–519.

Tidström, A., 2009. Causes of conflict in intercompetitor cooperation. J. Bus. Ind. Mark 24, 506–518.

Toral, S.L., Martínez-Torres, M.R., Barrero, F., 2010. Analysis of virtual communities supporting OSS projects using social network analysis: information and Software Technology.

Vivona, R., Demircioglu, M.A., Audretsch, D.B., 2023. The costs of collaborative innovation. J. Technol. Transf 48, 873–899. https://doi.org/10.1007/s10961-022-09933-1.

Wang, Z., Feng, Y., Wang, Y., Jones, J.A., Redmiles, D., 2020. Unveiling elite developers' activities in open source projects. ACM. Trans. Softw. Eng. Methodol. TOSEM 29, 1–35.

Wasserman, S., Faust, K., 1994. Social Network analysis: Methods and Applications. Cambridge university press.

West, J., Gallagher, S., 2006. Challenges of open innovation: the paradox of firm investment in open-source software. RD. Manag 36, 319–331. https://doi.org/10.1111/j.1467-9310.2006.00436.x.

Wiese, I.S., Da Silva, J.T., Steinmacher, I., Treude, C., Gerosa, M.A., 2016. Who is Who in the Mailing List? Comparing Six Disambiguation Heuristics to Identify Multiple Addresses of a Participant. In: 2016 IEEE International Conference on Software Maintenance and Evolution (ICSME). Presented at the 2016 IEEE International Conference on Software Maintenance and Evolution (ICSME), pp. 345–355. https://doi.org/10.1109/ICSME.2016.13.

Wohlin, C., 2012. Experimentation in Software Engineering.

Wolf, T., Schroter, A., Damian, D., Panjer, L.D., Nguyen, T.H.D., 2009. Mining Task-Based Social Networks to Explore Collaboration in Software Teams: IEEE Software.

Xia, H., Dawande, M., Mookerjee, V., 2016. Optimal Coordination in Distributed Software Development. Prod. Oper. Manag.

Yin, D.R.K., 2017. Case Study Research and Applications: Design and Methods. SAGE Publications, Inc, Los Angeles.

Yin, R.K., 2014. Case Study research: Design and Methods, 5. edition. Sage, Los Angeles, London, New Delhi, Singapore, Washington DC.

Yin, R.K., 2011. Applications of Case Study Research. SAGE.

Yu, Y., Wang, H., Yin, G., Wang, T., 2016. Reviewer recommendation for pull-requests in GitHub: what can we learn from code review and bug assignment? Inf. Softw. Technol.

Zheng, X., Le, Y., Chan, A.P.C., Hu, Y., Li, Y., 2016. Review of the application of social network analysis (SNA) in construction project management research. Int. J. Proj. Manag 34, 1214–1225. https://doi.org/10.1016/j.ijproman.2016.06.005.