

Entering an ecosystem: The hybrid OSS landscape from a developer perspective

Hanna Mäenpää¹, Fabian Fagerholm¹, Myriam Munezero¹, Terhi Kilamo², and Tommi Mikkonen¹

¹ Department of Computer Science, University of Helsinki, Helsinki, Finland
`hanna.maenpaa@helsinki.fi`

² Tampere University of Technology, Tampere, Finland

Abstract. Hybrid Open Source Software projects are virtual organizations that express characteristics of both static and dynamic behavior. They are choreographed through complex organizational structures that mix centralized governance with distributed community drivenness. While many communities use standard software tools to support their development processes, each community has its own ways of working and invisible power structures that influence how contributions are submitted, how they are verified and how decisions about the long-term direction of the software product are made. Navigating this environment is especially challenging for new developers who need to prove their abilities to gain rights to make contributions. This paper provides a viewpoint on the factors that influence a new developer's perception of the hybrid OSS developer community landscape. We apply an established developmental theory to build an initial model for the developer's context and discuss the model's validation, providing its practical and theoretical implications for building and managing on-line developer communities.

Key words: Open Source, community, management, on-boarding, developer experience

1 Introduction

Open Source Software (OSS) products are built collaboratively by developers that jointly form a developer community. These communities are geographically distributed, and they use virtual work environments, software tools and on-line collaboration platforms, along with mutually agreed policies and practices for developing their software product as a common good [Kil14]. Developer communities can comprise of independent individuals or representatives of companies, or in the case of "hybrid" communities: both [GBICMR13]. In hybrid communities, development decisions are influenced by versatile goals of individuals and often competing commercial interests [TRGB15]. Therefore, ensuring that stakeholders have equal opportunities to contribute is key for sustaining the reciprocity of the collaboration and the motivation of the community's members.

Even though individuals exist within the same context in developer communities, they may perceive their environment in different ways. This is especially

true for new developers, who are yet to learn about the culture, customs and realities of the community they want to join. So far, research has been performed on how developer communities should be organized to accommodate participation [DGO11, WO08], what factors help entry and on-boarding of its members [FGMB14, FSBM14], and which traits healthy and welcoming communities should display [Jan14, WJ07]. However, connections in between the community context and the developer's experience have largely been unexplored, due to e.g. the lack of a common vocabulary in describing the community phenomena [AdOJ17]. In this paper, we address this gap by linking current literature on developer community context and management to the viewpoint of an individual developer by using an established theory of environmental factors in human development [Bro77]. By studying the "Where, Who, and What" of the Hybrid Open Source developer community context, we aim to lay ground for new understanding on how hybrid developer communities can be built and managed to be welcoming for new developers.

This article is structured as follows. Section 2 first reviews how the hybrid OSS community context can be understood based on recent literature. Section 3 provides an overview of the Ecological Systems Model of Bronfenbrenner [Bro77], and describes its application to the hybrid OSS developer community context. Section 4 discusses and concludes the work.

2 The community context

One of the first models to describe online developer communities was the onion model of Nakakoji et al. [NYN⁺02]. The model focused on the developers with the most influence on the source code, establishing the terms "project leaders" and "core developers", who base their actions on implicit knowledge about the roles of the group and trust their peers in "just doing what needs to be done" [MFH02]. In Nakakoji's model, role of "co-developers" was differentiated to active developers and peripheral developers who delivered source code updates, yet were not in an active role in decision-making. Within this group, distributed communication of the developers increased requirements for work coordination, quality assurance and transparent, merit-based accreditation [MFH02]. The exterior layer of the model became formed by passive software users, community bystanders and bug reporters who had no direct contact with the source code [NYN⁺02]. While inviting new contributors and sustaining them contributes to the sustainability of a project [Jan14], it has long been preferable that new members enter the community through the outer layers and proceed towards the core based on their gained merit [YK03, NYN⁺02].

Today, OSS developer communities can be enveloped in an ecosystem of third party beneficiaries, who build their businesses on the OSS software produced by the community. These stakeholders can interact through shared socio-technical systems [FBACF17], taking approaches towards the developer community that can vary from symbiotic to opportunistic behaviors [DM05]. The commercial actors do not necessarily need to climb up the meritocracy ladder, yet they

can e.g. deploy developers to the project or be granted special privileges in its decision-making based on partnership models [SWvK15]. As the stakeholders can be fierce competitors on a shared market [TRGB15], they can be forced to form strategic alliances to advance their mutual goals such as stability or interoperability of the software [FBACF17]. These invisible power structures are hard to perceive, yet they can influence priorities of development and decrease a new developer's say in the project. Here, a well documented governance model, along with transparent and consistent management practices can help to increase equality among contributors [BJ12].

As the hybrid environment with mixed goals and motivations grows in size, a commercial or foundation-based central organization ("community sponsor") can be established to ensure the community's sustainability by orchestrating its work. To support its operation, the central orchestrator can deploy paid contributors to the project [WO08] and sponsor and manage the community's development tools, allowing it to control the development process to a degree. The orchestrator can also acquire funding, recruit new developers and arrange marketing- and community-building activities, which play an indistinguishable role in growing the community and ensuring its health [MH13]. A community ecosystem needs to endure changes and sustain its productivity in changing settings [MH13]: health can be seen as a mix of the community's internal performance, its vigor and its resilience to external disruptions [WJ07]. To support these aims, it is vital to understand how new developers need to be supported to become productive members and increase their participation in the community's activities.

A key element in the hybrid OSS community phenomenon is its governance: the way an organization is managed, including its powers, responsibilities and decision making practises [DK09]. Governance principles include how rights to participate are managed and how responsibilities are allocated amongst the community's members [Jan14]. This includes the assignment of roles and decision-making rights, as well as the measures and policies that enable continuous assessment of the developer organization's performance and well-being. When the community operates in a transparent way, its members can feel they can understand the community's state and have an opportunity to influence development activities and processes. Restricting autonomy of the open developer community may endanger the developers' motivation. This is especially true in hybrid projects where the governance structure can affect the motivation of outside volunteers' participation [Sha06] and thus, also the uptake of the software.

2.1 A focus on the developer

A software developer's experience is tied closely to the project's development tools and infrastructure, the developers' feelings about their work and the sense of value that they see for their contributions [FSBM14, KM06]. In a healthy community, members should be able to achieve sufficient returns for their work, to satisfy their needs and be encouraged to make new contributions [WJ07]. Here, individual work tasks and their relevant knowledge become an essential

Discipline	Definition of context	Source
Management science	Circumstances, conditions, situation or environments that are external to a specific phenomenon and that enable or constrain it.	Welter [Wel11].
Organization behavior	Situational opportunities and constraints that affect the occurrence and meaning of organizational behavior as well as functional relationships between variables.	Johns [Joh06].
Context-aware computing	Location of use, the collection of nearby people and objects as well as the changes to those objects over time.	Schilit [SAW94].
Software development	Rich and complex network of elements across different dimensions such as personal, project, organization, domain, product, teams, etc.	Antunes et al. [ACG10], De Araujo et al. [DSB ⁺ 04].
Linguistics	Meaning surrounding intra-textual clues that may be based on or be outside of the text itself.	Chin [Chi94].
Hybrid OSS development communities	Combination of open community-driven development and company internal development processes.	Sharma [SSR02].
OSS communities	A thematic division of developers into core and periphery based on frequency of contribution and nature of their contributions.	Nakakoji et al. [NYN ⁺ 02], Mockus et al. 2002 [MFH02],

Table 1. Summary of selected definitions of context from different domains.

part of the context [KM06]. In the online developer community realm, this context expands to a wide variety of freely available information and online tools such as the versioning system, workflow coordination and quality management tools, which together with discussion forums, chats and mailing lists make distributed coordination and collaborative decision-making possible. Interactions amongst the community’s members take place in both formal meetings and informal, spontaneous online commentaries. In the hybrid environment, a part of the discussions and the project’s decision-making take place outside the reach of the open community. This can limit the sense of own possibility to influence developments and hamper meaningful social interactions, which are known to be major barriers for the entry of newcomers to OSS projects [SSGR15].

In OSS communities, newcomers are required to prove their ability and motivation to make meaningful contributions to the project before they can acquire rights to make changes to the software source code and participate in decision-making about the software product’s future [SSR02]. To achieve this, new developers need to grasp not only the touch points for participation, meaning of the software’s licensing and governance processes, yet they also need to understand the social dynamics and inherent conventions of the community. In-

herently, the organization of the development process, access to decision-making and tasks, along with transparency of knowledge and processes form an important leverage for managing contributions[WO08]. Successful entry to a community is greatly dependent on the community's ability to provide this information and mentor new contributors to achieve an active role, which in part relies on a smooth learning process for new developers and their swift integration to the socio-cultural values, work processes and prevailing practices of the community[FSBM14]. Hence, designing and managing the entry process of new developers requires understanding of the nature of the community context and its reflection on the new developer's experience.

3 Modeling the context

The hybrid OSS ecosystem is a complex context, which we chose to explore through the lens of the Bioecological model of Bronfenbrenner (1979) that explains personal learning as "a function of forces that originate from the versatile settings amongst which the individual is a part of" [Bro79]. The model organizes environmental factors as layers similar to that of the Onion Model, positing the individual human being with his or her personal characteristics in the epicenter. Bronfenbrenner's view emphasizes a person's learning as an experiential process to which a person's most proximate interactions have the most influence. The approach can be used to divide the hybrid OSS ecosystem context to four layers: its micro-, meso-, exo- and macrosystem.

The microsystem level illustrates phenomena of practical everyday life: a person's most immediate surroundings and human contacts. More distant phenomena such as infrequent contacts, local culture and its institutions are positioned to the *exosystem* level. The layer in between, *mesosystem*, represents interactions between the micro- and exosystems, emphasizing the synergistic nature of the phenomena. As the exterior layer of the model, Bronfenbrenner describes the concept of a *macrosystem*, which deals with the values, events and expectations of the surrounding culture and society at large.

Bronfenbrenner denotes that all the layers have symbiotic relationships with each other and as for the ever-evolving nature of reality, are subject to changes in time (*chronosystem*). We used this thinking to depict the hybrid OSS ecosystem context in Figure 1. The next chapter, along with Table 2 elaborates the elements in our proposal in detail.

3.1 The example application

We used examples from literature presented in Section 2 to deploy the different layers of the Bioecological model. In our application (Figure 1), the individual software developer is placed as the focal point: as both a perceiver of the hybrid ecosystem context – and the subject of action and related personal development. Each individual has her personality, competencies and preferences that reflect to

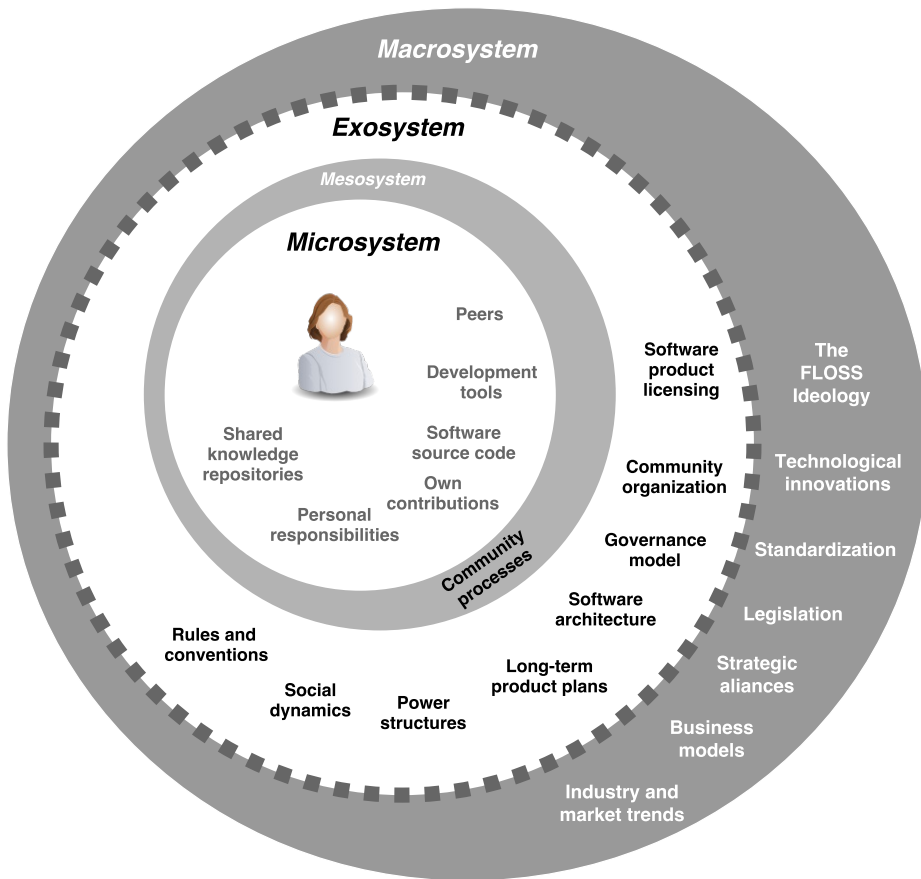


Fig. 1. Application of the Bioecological systems model [BM06] into the hybrid OSS ecosystem context.

the ways in which the person sees the community’s opportunities for interaction. The experience can be divided into cognitive, conative and affective aspects which guide her reactions to her surroundings [FGMB14]. Around the individual exists a *microsystem* of direct influences such as the developer’s contributions to the project, personal responsibilities, along with her interactions with peers and shared knowledge repositories such as the source code, online documentation and discussions.

Interactions with the microsystem phenomena take place in face to face settings, or as typical in online communities: through synchronous or asynchronous personal interactions with the help of socio-technical tools and platforms. The microsystem’s factors enable, allow and invite engagement, as well as inhibit participation and complicate proximate processes of the individual with her immediate environment [BM06]. The *exosystem* layer gathers phenomena that are

Table 2. The viewpoint of a new developer: examples of system properties

Individual		
Personal	Own competencies, goals, motivations and feelings.	[FGMB14]
Microsystem		
Personal	Own knowledge, tasks, contributions, responsibilities, own status/rank as a community member.	[KM06] [SSR02] [Jan14]
Stakeholders	Requirements engineers, code authors, committers, reviewers and maintainers. Own team members, mentors	[NYN ⁺ 02] [FSBM14]
Resources	Software source code. Personal development tools and common platforms: requirements management systems, code review/testing and continuous integration tools, shared knowledge repositories.	[Kil14]
Mesosystem: (the connecting layer)		
Organization of production	Code submission-, review-, testing-, integration- and release processes. Collaborative relationships amongst community members.	[WO08] [NYN ⁺ 02]
Exosystem		
General	Architecture of the software product. Rules and conventions. Licensing and IPR strategy. Organization and governance of the community and its orchestrator.	[SSR02] [WO08] [DK09]
Stakeholders	Different project teams and sub-communities. Project lead and high-level decision-making organizations, software users. Goals of partner organizations and keystone players.	[NYN ⁺ 02] [FBACF17]
Macrosystem		
General	Developments of related and common technologies, industry state of the practice, technology standardization, business and the marketplace. Strategic alliances.	
Ideology	FLOSS principles and their project-specific adaptation. Appreciation of self-fulfillment, diversity, equality and consensus-driven decision-making.	[LW03]

related to the community's internal culture and governance model, which indirectly influence the individual's environment and the way in which she engages with her microsystem. These two are connected by the *mesosystem*: the community's processes, including the different manual and automated stages of the software development process, along with its decision-making- and accreditation mechanisms. These define the community in terms of its structures, processes and events that occur outside the developing individual's immediate surroundings. The *macrosystem* comprises of the external environment that surrounds the OSS project and exists independently from it. However, it influences the exosystem and subsequently: the individual's behavior. While the Free and Open Source

Software (FLOSS) ideology is emblematic to the environment, other prevailing phenomena include the technological landscape, prevailing business models, stakeholder alliances, industry- and market trends.

4 Discussion and Summary

The Open Source development context is traditionally viewed through the notions of the Onion model and the core-periphery distribution of community members. We extend this definition by proposing a new application of an established theory on external factors in human development by Bronfenbrenner [Bro79] into the hybrid OSS ecosystem environment. We do this to build initial understanding on how the software ecosystem context can be perceived by a new developer that wishes to enter the community. Several factors may limit validity of our proposal. Firstly, Bronfenbrenner's Bioecological model of human development was originally created for analyzing the early stages of human development, including childhood and adolescence. Therefore, generalizing it for global, distributed software development can be questioned. However, it can be assumed that as a software developer enters an OSS project, she encounters a profound cultural adaptation process which requires learning both the shared language of the community, its ways of working, and its prevailing socio-cultural value systems.

Our proposal is based on literature introduced in Section 1, along with personal expertise of the authors. As three practitioners have given generous feedback at different stages of its development, we claim that our application provides a sufficiently reliable representation of the software ecosystem context. However, we acknowledge several places for improvement based on this feedback. More attention to the microsystem level could be placed by observing or interviewing new developers who still need help in comprehending their environment. Also, further inquiries of active community members who mentor new contributors to the project could shed light on the different ecosystem phenomena. Finally, the mesosystem layer could be amended by studying the role of corporate processes as a factor of the ecosystem, which, however is applicable only to the special case of commercially orchestrated communities.

With- or without these improvements, our application can be used for further studies - as shown by the rich explorations of the Bronfenbrenner's model in social sciences (documented in e.g. [BM06]). Learnings from this line of research can provide confidence for building new experimental research designs that address interrelationships, change and continuity in the hybrid OSS ecosystem context. While most activities in OSS development leave a digital trace, researchers should be able to validate their results through rigorous practice. Future studies could ask bold questions to find e.g. connections in between management and practice: "Do strict and explicit conventions lead to less testing of newcomers' code contributions?", "Does a hierarchical governance model decrease the need for coordination?" or "Does transparency of long-term planning lead to prolonged commitment to the project?" As a more distant goal,

the verified model could help discovering metrics which operationalize health of communities in terms of their intake and retainment of contributors. This could provide new practical understanding on which factors should be placed in the limelight of community management while designing entry paths for new developers and facilitating their personal validation as new, productive members of a community.

References

- [ACG10] Bruno Antunes, Francisco Correia, and Paulo Gomes. Towards a Software Developer Context Model. In Jörg Cassens, Anders Kofod-Petersen, Marielba Silva Zacarias, and Rebekah K Wegener, editors, *Proceedings of the Sixth International Workshop on Modeling and Reasoning in Context*, CEUR Workshop Proceedings, pages 1–12, Lisbon, Portugal, aug 2010. CEUR-WS.org.
- [AdOJ17] Carina Alves, Joyce Aline Pereira de Oliveira, and Slinger Jansen. Software Ecosystems Governance - {A} Systematic Literature Review and Research Agenda. In *{ICEIS} 2017 - Proceedings of the 19th International Conference on Enterprise Information Systems, Volume 3, Porto, Portugal, April 26-29, 2017*, pages 215–226, 2017.
- [BJ12] Alfred Baars and Slinger Jansen. A framework for software ecosystem governance. In *Lecture Notes in Business Information Processing*, volume 114 LNBIP, pages 168–180, 2012.
- [BM06] Urie Bronfenbrenner and Pamela A Morris. The bioecological model of human development. *Handbook of child psychology*, 2006.
- [Bro77] Urie Bronfenbrenner. Toward an experimental ecology of human development. *American psychologist*, 32(7):513, 1977.
- [Bro79] Urie Bronfenbrenner. The ecology of human development: Experiments by nature and design. *American Psychologist*, 32:513–531, 1979.
- [Chi94] Elaine Chin. Redefining context in research on writing. *Written Communication*, 11(4):445–482, 1994.
- [DGO11] Ivan De Noni, Andrea Ganzaroli, and Luigi Orsi. The Governance of Open Source Software Communities: An Exploratory Analysis. *Journal of Business Systems, Governance and Ethics*, 6(1):1–18, 2011.
- [DK09] Yael Dubinsky and Philippe Kruchten. 2Nd Workshop on Software Development Governance (SDG). In *Proceedings of the 2009 31st International Conference on Software Engineering: Companion Volume*, ICSE '09 COMPANION, pages 455–456, Washington, DC, USA, 2009. IEEE Computer Society.
- [DM05] Linus Dahlander and Mats G Magnusson. Relationships between open source software companies and communities: Observations from nordic firms. *Research policy*, 34(4):481–493, 2005.
- [DSB⁺04] Renata Mendes De Araujo, Flávia Maria Santoro, Patrick Brézillon, Marcos Roberto da Silva Borges, and Márcio Gonçalves Pereira da Rosa. Context Models for managing collaborative software development knowledge. In *Workshop on Modeling and Retrieval of Context (MRC)*, pages 61–72, 2004.

- [FBACF17] Oscar Franco-Bedoya, David Ameller, Dolors Costal, and Xavier Franch. Open source software ecosystems: A systematic mapping. *Information and Software Technology*, 91:160–185, 2017.
- [FGMB14] Fabian Fagerholm, Alejandro S Guinea, Jürgen Münch, and Jay Borenstein. The Role of Mentoring and Project Characteristics for Onboarding in Open Source Software Projects. In *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ESEM '14, pages 55:1—55:10, New York, NY, USA, 2014. ACM.
- [FSBM14] Fabian Fagerholm, Alejandro Sanchez Guinea, Jay Borenstein, and Jürgen Münch. Onboarding in Open Source Projects. *IEEE Software*, 31(6):54–61, 2014.
- [GBICMR13] J. M. Gonzalez-Barahona, D. Izquierdo-Cortazar, S. Maffulli, and G. Robles. Understanding how companies interact with free software communities. *IEEE Software*, 30(5):38–45, Sept 2013.
- [Jan14] Slinger Jansen. Measuring the health of open source software ecosystems: Beyond the scope of project health. *Information and Software Technology*, 56(11):1508–1519, 2014.
- [Joh06] Gary Johns. The essential impact of context on organizational behavior. *Academy of management review*, 31(2):386–408, 2006.
- [Kil14] Terhi Kilamo. Essential Properties of Open Development Communities. *Doctoral dissertation, Tampere University of Technology*, 2014.
- [KM06] Mik Kersten and Gail C Murphy. Using task context to improve programmer productivity. In *Proceedings of the 14th ACM SIGSOFT international symposium on Foundations of software engineering*, pages 1–11. ACM, 2006.
- [LW03] Karim R Lakhani and Robert G Wolf. Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. *Social Science Research Network*, 49:1–27, 2003.
- [MFH02] Audris Mockus, Roy T Fielding, and James D Herbsleb. Two case studies of open source software development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 11(3):309–346, 2002.
- [MH13] Konstantinos Manikas and Klaus Marius Hansen. Reviewing the health of software ecosystems—a conceptual framework proposal. In *Proceedings of the 5th International Workshop on Software Ecosystems (IWSECO)*, pages 33–44, 2013.
- [NYN⁺02] Kumiyo Nakakoji, Yasuhiro Yamamoto, Yoshiyuki Nishinaka, Kouichi Kishida, and Yunwen Ye. Evolution patterns of open-source software systems and communities. In *Proceedings of the international workshop on Principles of software evolution*, pages 76–85. ACM, 2002.
- [SAW94] Bill Schilit, Norman Adams, and Roy Want. Context-aware computing applications. In *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*, pages 85–90. IEEE, 1994.
- [Sha06] Sonali K. Shah. Motivation, Governance, and the Viability of Hybrid Forms in Open Source Software Development. *Management Science*, 52(7):1000–1014, 2006.
- [SSGR15] Igor Steinmacher, Marco Aurelio Graciotto Silva, Marco Aurelio Gerosa, and David F Redmiles. A systematic literature review on the barriers faced by newcomers to open source software projects. *Information and Software Technology*, 59:67–85, 2015.

- [SSR02] Srinarayan Sharma, Vijayan Sugumaran, and Balaji Rajagopalan. A framework for creating hybrid-open source software communities. *Information Systems Journal*, 12(1):7–25, 2002.
- [SWvK15] Mario Schaarschmidt, Gianfranco Walsh, and Harald F O von Kortzfleisch. How do firms influence open source software communities? A framework and empirical analysis of different governance modes. *Information and Organization*, 25(2):99–114, 2015.
- [TRGB15] Jose Teixeira, Gregorio Robles, and Jesús M. González-Barahona. Lessons learned from applying social network analysis on an industrial Free/Libre/Open Source Software ecosystem. *Journal of Internet Services and Applications*, 6(1):14, 2015.
- [Wel11] Friederike Welter. Contextualizing entrepreneurship conceptual challenges and ways forward. *Entrepreneurship Theory and Practice*, 35(1):165–184, 2011.
- [WJ07] Donald Wynn Jr. Assessing the health of an open source ecosystem. In *Emerging Free and Open Source Software Practices*, pages 238–258. IGI Global, 2007.
- [WO08] Joel West and Siobhán O’mahony. The Role of Participation Architecture in Growing Sponsored Open Source Communities. *Industry & Innovation*, 15(2):145–168, 2008.
- [YK03] Yunwen Ye and Kouichi Kishida. Toward an understanding of the motivation of open source software developers. In *Software Engineering, 2003. Proceedings. 25th International Conference on*, pages 419–429. IEEE, 2003.