Managing Commercial Conflicts of Interest in Open Source Foundations

Florian Weikert, Dirk Riehle, and Ann Barcomb(✉)

Friedrich-Alexander-Universität Erlangen-Nürnberg,
Martensstr. 3, 91058 Erlangen, Germany
florian@weikert.it, dirk@riehle.org, ann@barcomb.org
https://oss.cs.fau.de, https://dirkriehle.com/

Abstract. When companies opt to open source their software, they may choose to offer the project to an open source foundation. Donating the software to an open source foundation offers a number of advantages, such as access to the foundation's existing tools and project management. However, in donating the software, the company relinquishes control of the software and grants other foundation members—including competitors—the same rights to the software. Using a multiple-case study research approach, this paper examines how foundations manage conflicts of interest in the open sourcing donation scenario. We find that foundations primarily use a set of well-defined mechanisms to prevent such conflicts from arising, and that the use of these mechanisms can depend on the foundation type.

Keywords: Open source foundations · Sponsored open source · Commercial open source · Open source software

1 Introduction

Open source software (OSS) is ubiquitous in today's world. OSS is widely used within companies, not only for tooling and infrastructure, but also as a critical component of the supply chain [1,17,38]. Companies are not only using OSS, but are also contributing to OSS projects [2,3] and *open sourcing,* or releasing as OSS, millions of lines of source code [31]. Although many *single vendor open source* companies opt to retain intellectual property rights to their software, some companies donate their software to non-profit foundations [12,25,33,34,36]. Becoming a *donor* can help companies create open standards, lower their development costs, increase sales of complementary products or services, and take advantage of faster innovation [35,39], but it comes at a price.

By transferring their intellectual property rights to a non-profit foundation, these companies give up the control over their software [43] and have only the same privileges as other members of the foundation. Foundation members may even compete with each other, potentially introducing conflicts and tensions because of differing interests [15,20].

S. Hyrynsalmi et al. (Eds.): ICSOB 2019, LNBIP 370, pp. 130–144, 2019.
https://doi.org/10.1007/978-3-030-33742-1_11

In the context of donated projects, our research question is:

- **RQ**: How do foundations handle the conflicting interests of their members, when one member is a donor?

We investigate these questions through an exploratory multiple-case case study of four OSS projects which were created by companies, and subsequently donated to non-profit foundations.

The contributions of this paper are:

- A theory of conflict prevention in open source foundations.
- A discussion of the different types of foundations and the impact of foundation type on the use of conflict prevention mechanisms.

The rest of this paper is structured as follows: related work is reviewed in Sect. 2. Section 3 outlines the research process and Sect. 4 describes the results of our research. Limitations and suggestions for future work are discussed in Sect. 5. Finally, Sect. 6 concludes this paper.

2 Related Work

We identified three areas of research that were relevant to our topic: open source foundations, the evolution of company-created projects into open source, and conflicts in open source projects.

2.1 Open Source Foundations

Open source foundations are not all the same. For example, comparisons of the Apache Software Foundation (ASF) and the Eclipse Foundation have shown significant differences. Riehle [37] describes differences in legal status, mission, philosophy and governance structures. Furthermore, the power within the Eclipse Foundation is concentrated on the executive director, while the ASF gives most of its power to the board of directors [32]. Our work differs from previous work on conflict in open source foundations by explicitly considering the type of foundation involved.

The benefits of foundations are that they can handle donations [53]; provide communities with tools to handle corporate interests [23,27,28]; and manage and protect projects, intellectual property rights and communities [35]. Foundations enable shared development of software, thus reducing costs, helping to create a common standard, and increasing both reputation and visibility of members [35].

2.2 Project Evolution: From Company-Founded to Community-Managed

West and O'Mahony classify projects based on whether they are currently managed by a community (*autonomous*) or by a sponsoring company (*sponsored*) [52]. As projects may evolve from sponsored to autonomous, they also

introduce the categories *synthetic* (sponsor-created, i.e. started as a sponsored project) and *organic* (community-created, i.e. started as an autonomous project) to describe the original state of the project [30]. In this work, we consider the origin of the project as a possible source of conflict within foundations, and specifically examine projects which are synthetic and autonomous according to this framework.

If a project receives its initial resources and code from a company, it can also be described as a *spinout* [51]. Spinouts build on an established code base and are usually supported by their creators. However, this also means that new contributors face a steep learning curve because they must understand the existing code before they can contribute. As a result, the original sponsor may remain the largest contributor after spinning out the project. For example, even several years after it was donated, the majority of the source code in OpenStack was created by its donor, Rackspace [19].

2.3 Conflicts in Open Source Projects

One source of conflicts arises from the different interests of corporate and individual participants. For example, corporate sponsors have tried to steer the development via financial rewards or wanted to close parts of the code, thus violating the philosophy of open source software [13,42]. Other companies exploit OSS by taking more than they contribute [2]. As a result, tensions can be seen as a consequence of corporate behavior: that is, whether companies respect and give back to the community [7].

Another source of conflicts is within companies. Individuals who participate in OSS as employees of a company do not always promote the technical or business interests of their employers [3,29,40,47].

Finally, there can be conflicts between companies which are members of the same foundation. Sometimes, companies can collaborate on OSS and compete in the same market, without obvious conflicts [45]. This 'community of competitors' coordinates OSS development for mutual benefit by focusing on non-differentiating components [15]. However, inter-company rivalry can manifest in several ways. When companies make contributions without a complementary donation of intellectual property such as patents, it hinders innovation and limits the commercial benefit of OSS [50]. When multiple companies are contributing to a project, a company has to invest more resources to influence the project [16,41]. Companies can also be concerned about losing key developers to competitors [40]. Our emphasis is on the conflicts that can arise between foundation members, specifically in the case where one member is the project donor.

Van Wendel de Joode [46] argues that conflict management is mandatory for the success of software projects and identifies four mechanisms for managing conflict: third party intervention through mediators or arbitrators, code modularity to increase independence, parallel development lines to allow multiple solutions and the option to fork the project. Other techniques which have been proposed to resolve conflict are the promotion of shared beliefs and values, and discussions on persistent and public channels such as mailing lists [10,24].

3 Research Process

Our research is an exploratory investigation of conflict in open source foundations, and we wanted to consider different foundations in order to develop a broader understanding of how conflict between donors and other foundation members is handled. We chose an exploratory multiple-case case study research approach combined with grounded-theory-based analysis [6,54].

3.1 Case Study Design

A multiple-case case study allows researchers to employ replication logic to generalize case findings [54]. Our study consists of non-profit foundations as the primary unit of analysis. Our embedded, or sub-units of analysis, were made up of the different legal entities involved: the foundation itself, as well as individual companies.

As shown in Tables 1 and 2, we selected four cases based on their unique characteristics (age of the foundation, acceptance of corporate members, public versus member benefit, and whether they existed prior to the donation), thus using theoretical sampling [6] of polar types. Because of our grounded theory approach, we did not start with a preexisting theory [11]. Following Yin [54], we used a case study protocol and a case study database.

Table 1. Overview of cases

Case	Project	Donor	Creation date	Donation date
ACS	Apache CloudStack	Citrix Systems, Inc.	2008	2012[a]
CF	Cloud Foundry	Pivotal Software, Inc.	2011	2014
EC	Eclipse	IBM Corporation	2001	2004
OSt	OpenStack	Rackspace US, Inc.[b]	2010	2012

[a] CloudStack entered the Apache Incubator in 2012 and graduated to a top-level project in 2013.
[b] Parts of the project were donated by NASA.

Table 2. Overview of foundations

Case	Foundation	Creation date	Benefit[c]	Corporate members
ACS	Apache Software Foundation	1999	Public	No
CF	Cloud Foundry Foundation	2014	Member	Yes
EC	Eclipse Foundation	2004	Member	Yes
OSt	OpenStack Foundation	2012	Member	Yes

[c] Foundations for public benefit were established as "charitable organizations" based on Section 501(c)(3) of the United States Internal Revenue Code, while foundations for member benefit were incorporated as 501(c)(6) organizations ("trade associations").

3.2 Data Sources

We utilized a broad array of data sources, including documents (e.g. foundation bylaws and rules, protocols of board meetings, mailing list discussions, blog posts and press releases), podcasts and conference videos. We also conducted interviews with selected foundation representatives. More specifically, we contacted potential interview partners who had been active in the project for several years and who were (former) employees of the respective donor. With the exception of Cloud Foundry, we had one semi-structured interview per foundation. The interviews lasted between 45 and 90 min and were conducted via Skype, recorded and then transcribed. We refined our questions after each interview [6]. The case study protocol, interview protocols, and the case study database are published in an appendix [48].

As displayed in Table 3, our research incorporated more than 280 data sources, which we used for triangulating the insights and the resulting theory.

Table 3. Data sources by case and step of analysis

	Preliminary	Grounded theory			
Case	Documents	Interviews	Documents	Podcasts	Videos
ACS	24	1	20	4	3
CF	69	-	17	1	-
EC	15	1	10	-	3
OSt	81	1	26	5	4

3.3 Data Analysis

As is typical for grounded theory research, data collection and analysis happened simultaneously [5]. The whole process was iterative because we re-visited previously collected data and findings after new insights had emerged.

First, we started with a preliminary analysis step by creating a chronology of the most important events in the histories of the foundations and by identifying governance structures as well as the most important entities within the foundation. Moreover, we created an overview of participating companies and tracked the affiliations of contributors and board members.

Next, we used a software for qualitative data analysis (MAXQDA) to analyze the documents, all interview transcripts and the partial transcripts of the videos and podcasts while following the grounded theory approach of Corbin and Strauss [6].

We labeled text fragments with codes that emerged from the data (open coding). For example, when interview partners cited structures and rules that were borrowed from existing foundations, we assigned the code *learning from existing foundations*. Codes and text fragments were constantly compared to

each other. Furthermore, we revised the codings after each interview in order to incorporate new insights.

The next step entailed combining codes into categories based on shared concepts (axial coding). For instance, the category *limited foundation power* included the codes *no authority over volunteers* and *prioritizing project health over vendor dominance*.

Finally, we started selective coding in order to reduce our model to a core category. Although *conflict resolution* appeared to be a suitable candidate, we discovered that the category *conflict prevention* was central to our findings. We focused on this category by further developing its subcategories such as *governance*, *strategies*, *culture*, *screening processes* as well as *values* and *common motivation*. Moreover, we identified the causal relation of *bad behavior* and the influencing factor *foundation type*.

We wrote analytic memos throughout analysis [5] and compared the emerging theory to existing literature. This paper uses a theory-building logic [54] where individual case reports are omitted in favor of a comprehensive theory.

4 Results

Our research question concerned the mechanisms foundations use to handle conflicting interests between members, when one member is a donor. Our analysis identified the main category of *conflict prevention*, as described in Sect. 4.2. We also observed that the concrete strategies were influenced by the type of the foundation as described in Sect. 4.3.

4.1 Sources of Conflict

Since the foundations attracted a diverse audience of individuals and sometimes companies, the existence of different interests and goals was hardly surprising. However, this led to conflicts when members tried to enforce their own interests at the expense of other foundation members (*bad behavior*). For example, some wanted to take over specific projects or committees. This was especially likely when corporate members were competing with each other, thus having conflicting interests. Competition was especially fierce when these members targeted the same users, the market potential was huge or the technology was disruptive. The interview partners were also aware that this could ultimately threaten the success of the foundation.

However, foundations and their members not only expected but even encouraged competing companies to join. For example, the donors welcomed some of their competitors if those would help them fight a single dominant competitor (see Table 4).

Table 4. Dominant competitors as common motivation

Case	Quote
ACS	"CloudStack is Citrix's effort to take on VMware and enlist the rest of the vendor community in doing so."
EC	"Having others joining was exactly what should have happened, right? I mean you had in a way 'One enemy' and that one enemy was Microsoft. Everybody else - whether they were a competitor or not for you - was not really it. It was not really an issue. You wanted to get unified against Microsoft."
OSt	"Rackspace knew that in order to compete with Amazon they needed to have software that was like Amazon's. And the only way to get software like Amazon's was to band together every competitor of Amazon and develop that software."

4.2 Conflict Prevention

Figure 1 depicts our theory of conflict prevention in open source foundations. The five major subcategories from the code system were (1) *screening processes*, (2) *governance rules*, (3) *prevention strategies*, (4) *common interests* and (5) *culture and values*. We now describe these major categories and their relationships.

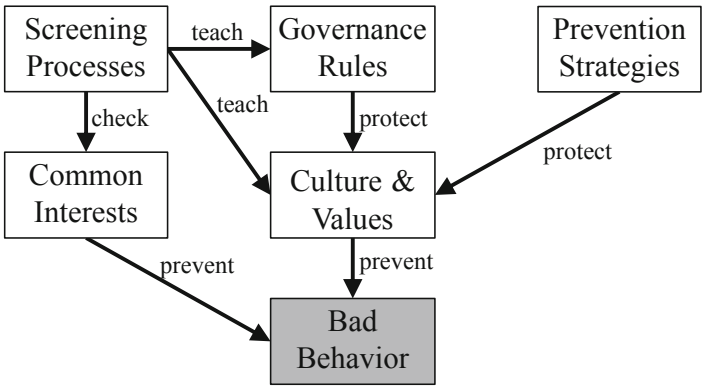

Fig. 1. Mechanisms for conflict prevention and their relations.

Screening Processes. Potential new members and projects had to pass specific screening processes before being accepted in a foundation (see Table 5). These processes had two goals.

First, they tried to identify *common interests* and to assess the motivation of potential members. Moreover, the technical, cultural and strategic fit of new

projects was determined. If a new project was proposed for donation, the foundation also wanted to make sure that the donor was interested in its long-term support while tolerating other participants.

Second, foundations—in particular the ASF—used an incubation process to teach both their *governance rules* and *culture and values* to new projects.

Additionally, these processes were described as *two way vetting processes* (ACS) since they did not only allowed the foundation to assess new members, but also allowed the potential members to see whether the foundation suited their needs.

Table 5. Different types of screening processes

Subject	Example
Committers	Committed enough for the task and matched the human attitudes required to work well with others (ACS)
Companies	They need to acknowledge that they have in mind the fact that the success of the foundation is the success of their own business (OSt)
Projects	The community has learned and demonstrated that it understands the principles and processes laid by the Apache Software Foundation and that it can now operate more autonomously. (ACS)

Governance Structures and Rules. As shown in Table 6, foundations had established formal governance structures and rules that were codified in their bylaws. Corporate members required such clearly-defined structures.

For example, *transparent affiliations* meant that individual contributors had to disclose their affiliations when joining the foundation. Moreover, any changes in status had to be communicated immediately. *Distributed decision-making* through clearly defined voting processes enabled a large base of members to voice their opinions. If the foundation was organized as a *meritocracy*, privileges such as the right to vote or write-access to source code repositories had to be earned through contributions. Consequently, even employees of smaller companies could reach high ranks in the foundation because of their individual contributions. Although some foundations accepted corporate members, they made sure that the amount of control through sponsorship was limited (*decoupling funding from control*). The foundations in this paper also established a *separation of powers* by transferring the technical authority to specific committees. As a result, the board of directors could only make legal and management decisions. In order to address the resource inequality of participating companies and individuals, some foundations offered different *tiers of membership*, depending on the size and financial possibilities of their members. As a result, these tiers could send their own representatives to the board. However, some foundations had established *representation limits* which limited the number of employees a single company could have on the board. Finally, the foundations and their committees saw themselves as *independent entities*. They made sure that their

success did not depend on just a few companies by recruiting independent staff and by monitoring the behavior of corporate members.

Table 6. Examples of governance structures and rules

Rule	Example
Transparent Affiliations	I will promptly update any change in my Affiliate status as defined in the Bylaws. (OSt)
Distributed Decision-Making	The decisions are made by the vote, as required in our by-laws. (ACS)
Meritocracy	It is [a] meritocracy and he did a good job that is why he is in the position he is. (ACS)
Decoupling Funding From Control	You cannot just, you know, shower the foundation in money and then you get all of the power. It is not going to happen. (EC)
Separation of Powers	There is also very strong separation between the technical decisions and the other things like the management in general, the general management of the foundation. (OSt)
Tiered Membership	Tiered structure is exactly to give representation to big companies and to smaller companies and to individuals who are part of a larger free software and open source community who want to care about this project. (OSt)
Representation Limits	No more than two directors shall be Affiliated (the 'Director Diversity Requirement'). (OSt)
Independent Entities	The Executive Director may not be an employee, officer, director or consultant of any Member of the Eclipse Foundation. (EC)

Explicit Strategies. In addition to creating governance structures, the foundations also employed a set of explicit prevention strategies to protect their *culture and values* (Table 7).

Because of transparent processes, foundations could *monitor the behavior* of their members and act accordingly. They also tried to *allow community participation* by including the larger community in as many decision-making processes as possible. For example, proposed governance changes were made subject to community review. This required the foundations to *enforce public communication* by announcing and discussing decisions on public mailing lists. Finally, they could use *project-specific strategies* if a project was dominated by a single corporate member. This could mean sending in independent contributors, terminating the project or creating a competing project.

Common Interests. Even if members were competing with each other, they shared some common interests. For example, their contributors were described

as *engineers by heart* (EC) who valued merit and the technical value of solutions more than corporate agendas and employment relations.

Common interests were also observed on the business side. Companies participated for pragmatic business reasons as they had commercial dependencies on the foundations' projects. Some of them saw foundations as a possibility to create a common platform against a dominant competitor (see Table 4).

Consequently, these companies were interested in growing their former projects by attracting new allies who had the same enemy. However, this could only work if they behaved in a collaborative way. Moreover, *bad behavior* was limited by the costs it would incur: *the money you spend and the outcome you get out of this is the big equalizer in this game.* (EC)

Culture and Values. Interview partners cited the existence of a good culture and shared values as essential for project success. While they helped to prevent *bad behavior*, the absence of such values could even destroy a project.

Openness describes not only open access to the source code, but also to decision processes, committees and documents. Interview partners pointed out that too much openness slowed down decision processes and could scare away potential commercial members. *Transparency* was as important as openness. Consequently, foundations tried to restrict the use of non-public communication like private mailing lists. If a foundation allowed corporate members, *equality* of opportunity was important to attract smaller companies and individual members: *The main role of the foundation is to make sure there is a level playing field where everybody feels safe* (OSt). The foundations also valued *merit* by acknowledging the amount of work contributors spent on the projects. *Neutrality* was named *the single most important thing of all* (EC). Consequently, foundations should not prefer single members. Instead, a truly vendor-neutral foundation would create a safe place where even competitors could collaborate. Similar to equality, a lack of neutrality would scare away contributors. Having competing members inside a foundation was seen as a sign of its *independence*. Moreover, foundations did not want to depend on specific members. Finally, they valued *diversity* of their members to benefit from different experiences and backgrounds.

4.3 Different Types of Foundations

We noticed that the ASF differed from the other foundations in several characteristics. For example, it tried to minimize corporate influence by allowing only individual members and by discouraging the display of member affiliations (*non-affiliation*). Our interview partners explicitly described the ASF as a *user-led* foundation, while the others were *vendor-led*. As a result, the ASF could not employ rules such as *transparent affiliations* and *representation limits*.

Instead, it emphasized culture and values more than the other foundations did. For example, its cultural principles—the Apache Way—and meritocracy were explicitly mentioned in its bylaws. This could be explained with the origin of the ASF: unlike vendor-led foundations, it was founded by a group of individuals

Table 7. Examples of prevention strategies

Strategy	Example
Monitor Behavior	The mission of the foundation is to make sure that all the companies and all the groups that are involved into development of the project actually behave. (OSt)
Allow Community Participation	Bylaws and legal documents for community review. (OSt)
Enforce Public Communication	The Apache mantra is, if it doesn't happen on the list it didn't happen kind of thing. (ACS)
Project-Specific Strategies	So at this very moment we sent in a few committers to also have the other implementation in that project. (EC)

(Apache Group), a "'grass roots' community of user-developers" [51, p. 1]. While companies are motivated by formal rules and structures [20,23,49], communities reflect the cultural beliefs and values [9] of traditional open source.

5 Limitations and Future Work

5.1 Limitations of the Study

Guba [21] proposed that the quality of qualitative work should be evaluated by its *credibility*, *transferability*, *dependability*, and *confirmability*, in place of measures which are appropriate for quantitative studies. For instance, a qualitative case study cannot claim statistical generalizability to a population, but this does not mean that it cannot offer theoretical generalizability, for instance through careful selection of polar cases [4,6,54].

Credibility can be established through triangulation. Case studies are a form of research which naturally incorporate data triangulation. We examined four cases, with three of them being backed by our interviews. However, we compensated for this fact by analyzing more than 200 other documents. While we cannot claim to have reached theoretical saturation, several researchers have noted that four cases might be reasonable due to pragmatic reasons [8].

Transferability concerns claims of theoretical generalizability [4], as described above. Our four cases were selected to vary by age of the foundation, acceptance of corporate members, public versus member benefit, and whether the foundation existed before the software donation. The differences between the ASF and the other foundations studied suggests that the dimension of membership was especially relevant.

Confirmability describes the extent to which researcher bias is mitigated. Any grounded theory analysis might be subject to coding errors or misinterpretations, and could have been influenced by previous knowledge of the researchers [11]. However, one way of reducing bias is through venting, which entails sharing the

results with professional colleagues to ensure that the findings are consistent with their experiences [18]. Moreover, our consideration of extant literature enhanced the objectivity of our study [8].

Dependability is increased through maintaining a record of the research process. We maintained both a case study protocol and a case study database to improve reliability [54]. Furthermore, we applied constant comparison and memoing during grounded theory analysis to increase theoretical sensitivity [22].

5.2 Future Work

In our research, we identified two particularly interesting topics which could benefit from further study.

Role of Trust. Since individual members were volunteers, the foundations did not have formal authority over them, thus having to trust them and their motivations. Trust was also cited by one foundation member as important for conflict resolution. However, existing literature makes opposing claims. For example, Gallivan [14] regards control as far more important while other researchers stress the importance of trust in open source projects [26,44].

Effectiveness of Non-Affiliation. It is not clear whether non-affiliation solves the problems of commercial interests instead of merely hiding them, as the underlying economic motivations still existed. Additionally—unlike foundations—companies do have formal authority over their contributors. However, there have been multiple reported cases where employees prioritized community needs over those of their employers [29,47], so this question needs to be studied with a nuanced understanding of affiliation.

6 Conclusions

In this paper, we examined four non-profit open source foundations that managed projects originally created by companies. More specifically, we investigated how these foundations handled potential conflicts of interests of their corporate members.

By conducting a multiple-case study combined with grounded theory analysis, we established a theory of conflict prevention. We identified a combination of screening processes, governance rules, prevention strategies, culture, values and common interests to discourage bad behavior of foundation members. This is of practical value to new foundations, as well as to companies which are considering donating to a foundation. For researchers, our work contributes to the understanding of cooperation between competitors in open source foundations by explaining how conflict is prevented.

Finally, we highlighted potential future work when we discussed the role of trust and non-affiliation. The limitations of our process and a theory-testing approach might also warrant future research on this topic.

Acknowledgments. We would like to thank our interview partners for sharing their time and expertise with us: Mark Hinkle, Stefano Maffulli and the anonymous member of the Eclipse Foundation.

References

1. Ayala, C.P., Cruzes, D.S., Hauge, O., Conradi, R.: Five facts on the adoption of open source software. Softw. IEEE **28**(2), 95–99 (2011)
2. Bonaccorsi, A., Rossi, C.: Contributing to the common pool resources in open source software. a comparison between individuals and firms. Technical reports, Working Paper, Sant'Anna School of Advanced Studies Institute for Informatics and Telematics (August 2003)
3. Butler, S., et al.: On company contributions to community open source software projects. IEEE Transactions on Software Engineering (2019)
4. Cavaye, A.L.: Case study research: a multi-faceted research approach for IS. Inf. Syst. J. **6**(3), 227–242 (1996)
5. Charmaz, K.: Grounded theory as an emergent method. In: Leavy, P., Nagy Hasse-Biber, S. (eds.) Handbook of Emergent Methods, chap. 7, pp. 81–110. Gilford Press, New York (2008)
6. Corbin, J.M., Strauss, A.L.: Grounded theory research: procedures, canons, and evaluative criteria. Qual. Sociol. **13**(1), 3–21 (1990)
7. Dahlander, L., Magnusson, M.G.: Relationships between open source software companies and communities: observations from Nordic firms. Res. Policy **34**(4), 481–493 (2005)
8. Eisenhardt, K.M.: Building theories from case study research. Acad. Manag. Rev. **14**(4), 532–550 (1989)
9. Elliott, M.S., Scacchi, W.: Communicating and mitigating conflict in Open Source software development projects. Projects & Profits (2002)
10. Elliott, M.S., Scacchi, W.: Free software development: cooperation and conflict in a virtual organizational culture. In: Koch, S. (ed.) Free/Open Source Software Development, pp. 152–173. Idea Group Publishing (2004)
11. Fernández, W.D.: The grounded theory method and case study data in IS research: issues and design. In: Information Systems Foundations: Constructing and Criticising Workshop at The Australian National University, pp. 43–59 (July 2004)
12. Fitzgerald, B.: The transformation of open source software. MIS Q. **30**(3), 587–598 (2006)
13. Freeman, S., Siltala, J.: Freedom and profit: how suits and hackers are working it out on the desktop (2004), working paper presented in 4/EASST Joint Meeting, Paris 26/08/04
14. Gallivan, M.J.: Striking a balance between trust and control in a virtual organization: a content analysis of open source software case studies. Inf. Syst. J. **11**(4), 277–304 (2001)
15. Germonprez, M., Allen, J.P., Warner, B., Hill, J., McClements, G.: Open source communities of competitors. Interactions **20**(6), 54–59 (2013)
16. Germonprez, M., Kendall, J.E., Kendall, K.E., Mathiassen, L., Young, B., Warner, B.: A theory of responsive design: a field study of corporate engagement with open source communities. Inf. Syst. Res. **28**(1), 64–83 (2016)
17. Germonprez, M., Link, G.J., Lumbard, K., Goggins, S.: Eight observations and 24 research questions about open source projects: Illuminating new realities. In: Proceedings of the ACM on Human-Computer Interaction 2(CSCW), 57 (2018)

18. Goetz, J., LeCompte, D.: Ethnography and Qualitative Design in Educational Research. Academic Press, Cambridge (1984)
19. González-Barahona, J.M., Izquierdo-Cortazar, D., Maffulli, S., Robles, G.: Understanding how companies interact with free software communities. IEEE Softw. **30**(5), 38–45 (2013)
20. González-Barahona, J.M., Robles, G.: Trends in Free, Libre, Open Source Software Communities: From Volunteers to Companies/Aktuelle Trends in Free-, Libre-, und Open-Source-Software-. Information Technology (2013)
21. Guba, E.G.: Criteria for assessing the trustworthiness of naturalistic inquiries. Educ. Technol. Res. Dev. **29**(2), 75–91 (1981)
22. Hallberg, L.: Some thoughts about the literature review in grounded theory studies. Int. J. Qual. Stud. Health Well-being **5**, PMC2915820 (2010)
23. Hunter, P., Walli, S.: The rise and evolution of the open source software foundation. Int. Free Open Source Softw. Law Rev. **5**(1), 31–42 (2013)
24. Jensen, C., Scacchi, W.: Collaboration, leadership, control, and conflict negotiation and the Netbeans.org open source software development community. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences (2005)
25. Krishnamurthy, S.: An analysis of open source business models. In: Feller, J., Fitzgerald, B., Hissam, S.A., Lakhani, K.R. (eds.) Making Sense of the Bazaar: Perspectives on Open Source and Free Software, vol. 54, pp. 267–278. The MIT Press, Cambridge (2003)
26. Lattemann, C., Stieglitz, S.: Framework for governance in open source communities. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences (2005)
27. O'Mahony, S.: Guarding the commons: how community managed software projects protect their work. Res. Policy **32**(7), 1179–1198 (2003)
28. O'Mahony, S.: Nonprofit foundations and their role in community-firm software collaboration. In: Feller, J., Fitzgerald, B., Hissam, S.A., Lakhani, K.R. (eds.) Perspectives on Free and Open Source Software, pp. 393–413. The MIT Press, Cambridge (2005)
29. O'Mahony, S., Bechky, B.A.: Boundary organizations: enabling collaboration among unexpected allies. Adm. Sci. Q. **53**(3), 422–459 (2008)
30. O'Mahony, S., West, J.: What makes a project open source? migrating from organic to synthetic communities. Academy of Management conference, Technology and Innovation Management division, Honolulu, August 2005, p. 39 (2005)
31. Pearce, J.: 9.9 million lines of code and still moving fast - Facebook open source in 2014 (2014). https://code.facebook.com/posts/292625127566143/9-9-million-lines-of-code-and-still-moving-fast-facebook-open-source-in-2014/
32. Prattico, L.: Governance of Open Source Software Foundations: Who Holds the Power? Technology Innovation Management Review (2012)
33. Riehle, D.: The economic motivation of open source software: stakeholder perspectives. Computer **40**(4), 25–32 (2007)
34. Riehle, D.: The commercial open source business model. In: Proceedings of the Fifteenth Americas Conference on Information Systems. vol. AMCIS 2009, pp. 1–10 (2009)
35. Riehle, D.: The economic case for open source foundations. Computer **43**(1), 86–90 (2010)
36. Riehle, D.: The single-vendor commercial open course business model. Inf. Syst. E-Bus. Manag. **10**(1), 5–17 (2012)

37. Riehle, D., Berschneider, S.: A model of open source developer foundations. In: The 8th International Conference on Open Source Systems (OSS 2012), pp. 7–28. Springer (2012)
38. Riehle, D., Harutyunyan, N.: License Clearance in Software Product Governance, chap. 5, pp. 83–96. NII Shonan, Tokyo, Japan (2017)
39. Rossi, C., Bonaccorsi, A.: Why profit-oriented companies enter the OS field?: intrinsic vs. extrinsic incentives. In: ACM SIGSOFT Software Engineering Notes. vol. 30, pp. 1–5. ACM (2005)
40. Schaarschmidt, M., Stol, K.J.: Company soldiers and gone-natives: role conflict and career ambition among firm-employed open source developers. In: Thirty ninth International Conference on Information Systems. Association for Information Systems (AIS) (2018)
41. Schaarschmidt, M., Walsh, G., von Kortzfleisch, H.F.: How do firms influence open source software communities? a framework and empirical analysis of different governance modes. Inf. Organ. **25**(2), 99–114 (2015)
42. Siltala, J., Freeman, S., Miettinen, R.: Exploring the tensions between volunteers and firms in hybrid projects (2007), center for Activity Theory and Developmental Work Research. (Working Paper 36)
43. Skerrett, I.: Best practices in multi-vendor open source communities. Open Source Business Resource (2011)
44. Stewart, K.J., Gosain, S.: The impact of ideology on effectiveness in open source software development teams. MIS Q. **30**(2), 291–314 (2006)
45. Teixeira, J.: Understanding coopetition in the open-source arena: the cases of WebKit and OpenStack. In: Proceedings of The International Symposium on Open Collaboration, p. 39. ACM (2014)
46. Van Wendel de Joode, R.: Managing conflicts in open source communities. Electron. Markets **14**, 104–113 (2004)
47. Wagstrom, P., Herbsleb, J., Kraut, R., Mockus, A.: The impact of commercial organizations on volunteer participation in an online community. In: 2010 Academy of Management Meeting. Montreal, Canada (2010)
48. Weikert, F.: How Open Source Foundations Handle Conflicting Interests in Company-Started Projects. Master's thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg (2014)
49. Weiss, M.: Control and diversity in company-led open source projects. Open Source Business Resource (2011)
50. Wen, W., Ceccagnoli, M., Forman, C.: Opening up intellectual property strategy: implications for open source software entry by start-up firms. Manag. Sci. **62**(9), 2668–2691 (2015)
51. West, J., O'Mahony, S.: Contrasting community building in sponsored and community founded open source projects. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences (2005)
52. West, J., O'Mahony, S.: The role of participation architecture in growing sponsored open source communities. Ind. & Innov. **15**(2), 145–168 (2008)
53. Xie, Z.: Open Source Software Foundations. Open Source Business Resource (2008)
54. Yin, R.K.: Case Study Research: Design and Methods, 5th edn. SAGE Publications, Thousand Oaks (2013)