



Private-Collective Innovation and Open Source Software: Longitudinal Insights from Linux Kernel Development

Dirk Homscheid, Jérôme Kunegis, Mario Schaarschmidt

► To cite this version:

Dirk Homscheid, Jérôme Kunegis, Mario Schaarschmidt. Private-Collective Innovation and Open Source Software: Longitudinal Insights from Linux Kernel Development. 14th Conference on e-Business, e-Services and e-Society (I3E), Oct 2015, Delft, Netherlands. pp.299-313, 10.1007/978-3-319-25013-7_24 . hal-01448048

HAL Id: hal-01448048

<https://inria.hal.science/hal-01448048>

Submitted on 27 Jan 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Private-Collective Innovation and Open Source Software: Longitudinal Insights from Linux Kernel Development

Dirk Homscheid^{1,2}, Jérôme Kunegis², Mario Schaarschmidt^{1,2}

¹ University of Koblenz-Landau, Institute for Management, Koblenz, Germany

² University of Koblenz-Landau, Institute for Web Science and Technologies, Koblenz,
Germany

{dhomscheid, kunegis, mario.schaarschmidt}@uni-koblenz.de

Abstract. While in early years, software technology companies such as IBM and Novell invested time and resources in open source software (OSS) development, today even user firms (e.g., Samsung) invest in OSS development. Thus, today's professional OSS projects receive contributions from hobbyists, universities, research centers, as well as software vendors and user firms. Theorists have referred to this kind of combined public and private investments in innovation creation as private-collective innovation. In particular, the private-collective innovation model seeks to explain why firms privately invest resources to create artifacts that share the characteristics of non-rivalry and non-excludability. The aim of this research is to investigate how different contributor groups associated with public and increasing private interests interact in an OSS development project. The results of the study show that the balance between private and collective contributors in the Linux kernel development seems to be changing to an open source project that is mostly developed jointly by private companies.

Keywords: Open Source, Open Source Community, Private-Collective Innovation, Linux Kernel.

1 Introduction

Open source software (OSS) has changed how researchers and practitioners look at software development and related business models [1–3]. OSS differs from traditional in-house software development, as its outcome, the software, is freely accessible by anyone. As a result, major companies such as Facebook, Google and Twitter use open source technologies. In 2012 one million open source projects were catalogued and this figure was projected to double within two years [4]. In addition, according to IDC research, the OSS market should be worth approx. \$ 8 billion in 2013 [5].

Due to this success, firms have also started to actively engage in OSS development [6, 7]. While in early years, software technology companies such as IBM and Novell invested time and resources in OSS development, today even user firms (e.g., Samsung) invest in OSS development [8]. Thus, today's successful OSS projects receive contributions from hobbyists, universities, research centers, as well as from software vendors and user firms [9, 10]. Theorists have referred to this kind of combined public and

private investments in innovation creation as private-collective innovation [11]. This concept asserts that a private investment model – where firms create and commercialize ideas themselves – and a collective invention model, where multiple economic actors create public goods innovations, may coexist under certain circumstances [12]. In particular, the private-collective innovation model seeks to explain why firms privately invest resources to create artifacts that share the characteristics of non-rivalry and non-excludability [13].

The private-collective model also implicitly assumes that private and public investments in innovations are approximately equal. However, successful OSS projects receive more than 75% of their code from contributors¹ who are paid by a company [8] and the majority of code is written between 9 am and 5 pm – again indicating that contributions are predominantly provided by firms [14]. These figures contrast with the picture of private-collective innovation as an invention mode where public and private interests manifest equally. The aim of this research therefore is to investigate how different contributor groups associated with public and increasing private interests interact in an OSS development project.

In order to study the interplay of both interest groups we not only need to consider demographic characteristics of the community but also the structural patterns of interactions in it. To achieve this goal, we analyze developers active in the Linux kernel (LK) development community from a social network point of view, as the interaction between the members of a software development community reflects the structure of their collaboration. In particular, we investigate degree distributions and the Gini coefficient in the contributor network with respect to the private and collective contributor groups. Network centrality measures are important indicators of influence in OSS development and are known to deviate according to having firm sponsorship or not [15].

We start with detailing what volunteer and firm-sponsored (i.e., employed) developers motivate to participate in OSS development. Then, we discuss the private-collective innovation model in more detail. Based on a dataset of mailing list communication of LK developers from 1996 to 2014 we calculate network measures for each type of developer (e.g., firm-sponsored, hobbyist, university-affiliated, etc.) and compare them for each year. We discuss implications for research and provide further avenues for research concerning private-collective innovation.

2 Theoretical Background

2.1 Open Source Software Contributors

OSS Communities

An OSS project relies on contributors who make up the core element of an OSS community. OSS is commonly understood as a type of software that can be used, changed, and shared by any person. The software itself is in most cases developed by a

¹ In this study the terms *contributor*, *developer*, *actor*, *participant* and *programmer* are used synonymously to denote people who are active in OSS projects and in Linux kernel development.

heterogeneous group of people and distributed under specific licenses, which guarantee the above-mentioned characteristics of OSS [16].

In general, a community arises when different people come together and share a common interest [17]. Thus, von Hippel and von Krogh [11] conceptualize OSS development communities as “*Internet-based communities of software developers who voluntarily collaborate in order to develop software that they or their organizations need*” [11, p. 209]. Besides the fact that OSS communities consist of hobbyists, who voluntarily provide their resources to the community, the definition also involves another important contributor group – organizations. Organizations differ from hobbyists in terms of their motivation to engage and are represented in the community by their employed developers. In turn, employed developers might be considered as proxies for firm interests in the community.

Motivation of Voluntary OSS Developers

The pertinent literature specifies intrinsic and extrinsic motivation as major drivers for hobbyists to engage in OSS projects (e.g., [18–21]). Intrinsic motivation is the execution of an activity due to the accompanying enthusiasm and not for the achievement of specific results [22]. A behavior is extrinsically motivated when an activity is performed for reward, recognition or because of an instruction from someone or an obligation [22]. Although researchers agree on different forms of intrinsic and extrinsic motivation, there is often disagreement about their relevance. The most relevant forms of both motivation types in the context of OSS developers are described briefly in the following.

In connection with OSS developers, researchers investigated a plethora of intrinsic motivators. Among these, *joy-based intrinsic motivation* is the strongest and most prevalent driver of OSS contributors [19]. Joy-based motivation is closely linked to the creativity of a person. Frequently, contributors to OSS projects have a strong interest in software development and related challenges [18].

Another fundamental aspect of intrinsic motivation is *altruism*, which is the desire to help others and to improve their welfare. In OSS communities, developers code programs, report bugs, etc., at their own expense, which includes the invested time and opportunity costs. They participate in the OSS community, without taking advantages of its outcome [18, 21, 23].

In addition, the *OSS ideology* plays a crucial role for many contributors and involves

- joint collaborative values, such as helping, sharing and collaboration,
- individual values, such as learning, technical knowledge and reputation,
- OSS process beliefs, such as code quality and bug fixing and
- beliefs regarding the importance of freedom in OSS, such as an open source code and its free availability and use for everyone [24].

Besides these distinguishing aspects of participants’ intrinsic motivation to engage in OSS projects, researchers have found that extrinsic stimuli can also have an impact on the activities of actors in communities (e.g., [18–21]).

An extrinsic stimulus is given through a *personal need* of a developer. Many OSS projects are launched because the initiators needed software with specific functions that

are not available to date, and they have the willingness and knowledge to develop these [21].

OSS communities offer the possibility for developers to *improve their programming skills* and their knowledge through participation in a project. Programmers are free to choose in which tasks they participate according to their interests and abilities. As a result, the self-learning participants are experiencing a continuous learning curve and build a repertoire of experiences, ways and means to solve specific software development tasks [21].

In addition, “signaling incentives” as described by Lerner and Tirole [25] can also be a reason for people to participate in OSS communities. The incentives cover, inter alia, the *recognition by other members* of the community and the *improvement of the professional status*.

Motivation of Firms Involved in OSS Development

In addition to hobbyists, companies are also active in OSS communities. While voluntary OSS contributors are driven by intrinsic and extrinsic values, economic and technological aspects motivate firms to participate in OSS projects. In recent times, companies open outwards to organize their innovation activities more effectively and efficiently. A means to complement their own resource base are innovation communities. In the case of software companies, OSS communities form a resource pool these firms can benefit from – depending on the strategy they pursue [6, 26]. Literature investigating motivational aspects of companies active in OSS projects reveals that economic theory is not sufficient to explain the relation between firms and their OSS engagement. Andersen-Gott et al. [27] have reviewed this issue and identified the following three categories of motivational factors that are relevant for companies active in OSS communities.

1. Innovative capabilities

If the involvement of a company in an OSS project is aligned with the business model it maintains, the interaction with the community can lead to better or new products which imply a competitive advantage. The inclusion of external contributors increases the firm’s innovative capacity.

2. Complementary services

The dominant way for firms to appropriate from OSS is by providing complementary services to customers (e.g., training, technical support, consultancy and certifications [28]) aligned with their business strategy [29]. Firms pursuing this concept deploy own employees that also contribute to the open source project and community work. Thus, the company (1) acquires external knowledge through their own employees active in OSS development and (2) has access to complementary resources in the community, which are difficult to replicate internally [30].

3. Cost reduction

Companies can publish the source code of their proprietary software under an OS license, try to attract external developers and build a community around the software. In this case, the company will get, for example, ideas for new features, bug reports, documentation, and extensions of the software from external contributors without having to pay for it [25, 31]. Further, in the long run, the code is maintained by the

community, such that the firm has lower costs than its competitors with proprietary software [32]. However, it should be noted that establishing an ecosystem and an active community around released source code is no easy task as rivals could pursue similar strategies [26, 33].

2.2 Private-Collective Innovation

In organization science, two different modes of innovation are dominant, namely the private investment and the collective action model.

The *private investment model* is associated with a rather closed innovation behavior. Innovators tend to protect their internally developed proprietary knowledge as this is the source of their profits and competitive advantage [11]. Here innovation is clearly seen as a closed process driven by private investments in order to lead to private returns for the innovator [34].

The *collective action model* of innovation is connected to the provision of a public good. Innovators collaborate in order to develop a public good under conditions of market failure. The produced good is characterized by non-excludability and non-rivalry [35]. This model requires that innovators supply their collected knowledge about a project to a common knowledge base and thus make it a public good. This innovation method can unfortunately be exploited by free riders, who wait until other contributors have done the work and use the outcome for free [11, 35].

OSS communities are an example for a mixture of both mentioned innovation models. OSS contributors freely reveal their privately developed source code as a public good. The developers do not make commercial use of their property rights, although the source code is created as a result of private investments. This innovation behavior is termed private-collective innovation [11]. To get a deeper understanding of how OSS communities combine the best of both models, OSS innovation is first considered from the private investment and second from the collective action point of view.

From the *private investment model perspective*, OSS deviates in two major aspects from the conventional private investment model. First, software contributors are the actual innovators in OSS rather than commercial software developers, because they create software that is needed either by themselves or by the community. Second, OSS developers freely reveal the source code, which they have developed by private means; this manner stands in contrast to the classical innovation behavior. Due to the lack of a commercial market for the sale or licensing of OSS, it is made openly available as a public good [11]. Rewards for the developers are provided in forms other than money or commercialization of property rights. Contributors gain private profits such as reputation, experience or reciprocity [25, 36].

From the *collective action model view*, the community produces a public good with its attributes of non-excludability and non-rivalry [35]. Taking the above given description of the collective action model into account, the non-excludability would bring a dilemma with it because free riders benefit from the software but do not contribute to the good compared to the developers. This circumstance is not a problem, as in line with the OS ideology people voluntarily participate in OSS development and share the results without costs [11]. Moreover, contributors obtain benefits, for example problem

solving expertise, learning and enjoyment, from the participation on developing a public good, which the free rider cannot get [25, 37]. The benefits in form of selective incentives are connected to the development process of the good and thus only accessible for the participants. Therefore, OSS contributions cannot be seen as pure public goods as these have significant private elements that evolved out of the ideology, which support the community [11].

In sum, the private-collective model of innovation combines the advantages of both private investment and collective action model. Table 1 compares the most important aspects of the three innovation models from an economic perspective and in relation to OSS development.

Table 1. Comparison of different aspects for the private, collective and private-collective innovation model (Source: adapted from Demil and Lecocq [38] and Schaarschmidt et al. [39])

	<i>Private</i>	<i>Collective</i>	<i>Private-Collective</i>
<i>License</i>	Proprietary	Open	Open
<i>Copyright ownership</i>	Company	Collective	Collective
<i>Number of participating companies</i>	One	None	One or more
<i>Revenue stream</i>	Direct	None	Indirect
<i>Control intensity</i>	High	Low	Low
<i>Knowledge sharing intensity</i>	Low	High	High

3 Method

3.1 Research Context

To find a relevant OSS project for our research, we have taken different aspects into account (e.g., size of the project, activity and continuity, company involvement, availability of a large set of data). Finally, we chose the LK project as our research context, which has served as an example for OSS in many previous studies [e.g., 40].

The LK project was initialized by Linus Torvalds in 1991 and has been one of the most active OSS projects since its beginning. There are software releases every three month on average, which are possible because of the fast-moving development process and the broad foundation of contributors, ranging from hobbyists to companies. Thus, it involves more people than any other OSS project. The kernel itself makes up the core component of any Linux system and is used in operating systems for mobile devices right up to operating systems for supercomputers. Typically, a new release of the kernel comprises over 10,000 patches contributed by over 1,100 developers representing over 225 companies and is published under the GNU General Public Licence v2 [8].

Besides the fact that the LK is one of the largest cooperative software projects ever started, it has also an economic relevance, as many companies have business models that rely on the LK or on software working on top of the LK, respectively. Many of these companies do actively participate in the improvement of the kernel and thereby take effect on the orientation of the development. Very active companies in the kernel development, among others, are RedHat, Intel, IBM, Samsung, Google and Oracle [8].

3.2 Data Collection and Coding of Contributor Categories

To obtain the data needed for our research, we crawled the LK mailing list web archive². We use mailing list data as it is suitable to calculate network positions that represent developers' influence [15]. The "linux-kernel"³ mailing list has the purpose of discussing LK development topics as well as of reporting bugs. The observation period of the LK community ranges from 1996 (beginning of the web archive) to 2014.

We identified actors that occur multiple times on the list, for example with different email addresses, but identical sender names. We have mapped these to one person object related to the email address s/he has used when sending a message to the list for the first time.

The identified people interacting in the LK mailing list act partly on behalf of companies. To get a deeper understanding if the actors in the mailing list are affiliated with a firm we used the domain name of the email addresses to assign people to a contributor category. Developers sending messages from a domain indicating that the person is employed by the corresponding company are classified as employed contributors, whereas people using email addresses from public email providers such as yahoo.com were classified as hobbyists. Likewise, we identified developers with email addresses indicating universities and research institutions. Assigning LK actors to a contributor category was done in a semi-manual and semi-automated process in order to obtain a high accuracy of the attributions. Detailed information about the different contributor categories is provided in Table 2.

Table 2. Contributor categories

Contributor Category	Description	Examples
Companies	People with email addresses from companies	intel.com, redhat.com
Hobbyists	People with private email addresses	gmail.com, yahoo.com
Universities	People from universities	columbia.edu, duke.edu
Research Institutions	People from research institutions and public authorities (e.g., IEEE, government, military)	ieee.org, nasa.gov, army.mil

The cleaned dataset comprises 1,941,119 communication replies for the total time period with overall 86,509 contributors involved. The overall distribution of the contributor groups is made of 37.96% of company developers, hobbyists represent 51.22%, universities account for 9.65% and research institutions make up 1.17%. Descriptive information about the dataset is given in Figures 1 and 2. Figure 1 shows the quantity of identified contributors per contributor group and year from 1996 to 2014. Figure 2 states the amount of messages sent per contributor group and year for the investigated period.

² <http://marc.info/?l=linux-kernel>

³ <http://vger.kernel.org/vger-lists.html#linux-kernel>

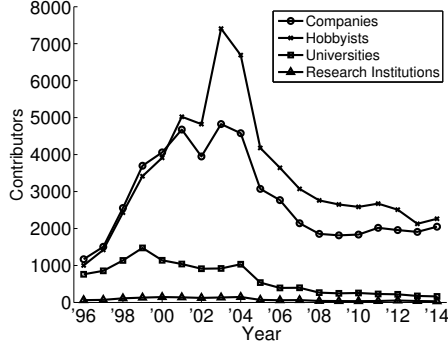


Fig. 1. Contributors per group and year

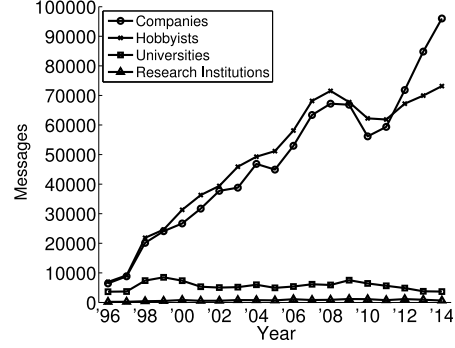


Fig. 2. Amount of messages sent per group and year

3.3 Social Network Analysis

A social network represents persons connected by edges. Social networks can represent friendship relationships, communication, interaction contacts or other types of social relationships. Social network datasets are widely used, not only in the area of social network analysis, but also in the areas of data mining, sociology, politics, economics and other fields [41].

In order to study the interactions of developers in the LK community, we perform an analysis of the LK mailing list's communication network. Communication within the Linux developer community can be modelled as a directed network, in which nodes are developers and directed edges (i.e., arcs) are a reply of one developer to another. In our dataset, we ignore all messages that are not replies to other developers. A relationship between the sender of a starter message and others does only emerge when one or more people reply to the starter message. We perform a structural analysis of this network to study the interplay of developers interacting.

The directed network of replies we consider is annotated with two additional metadata:

- For replies, the posting timestamp is known. This allows us to make a longitudinal analysis of the considered network statistics.
- For developers, we know their company, university or other affiliation, if any, allowing us to identify four categories of developers, as described in Section 3.2.

We perform social network analysis with Matlab and the KONECT Toolbox [42].

The contribution of one user in a directed social network can be used to measure both the activity and the importance of that user in the community. We achieve this by considering the following network-based measures, each of which is defined for individual nodes:

- The in-degree of a node equals the total number of replies received by a developer. The in-degree can thus be interpreted as a measure of importance of a developer.

- The out-degree of a node equals the total number of replies written by a developer, and can thus be interpreted as a measure of the activity of a developer.
- As a network-wide measure, we additionally define the Gini coefficient of the in-degree distribution [43], which denotes the inequality of the in-degrees. It is zero when all developers have equal in-degrees and one when a single developer received all replies. It can thus be interpreted as a measure of diversity of the community [44].

4 Results

4.1 Comparison of In-degree and Out-degree

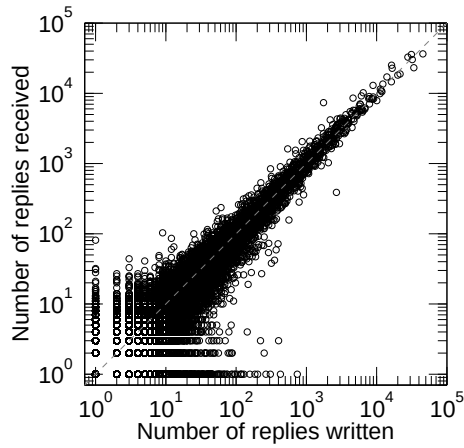


Fig. 3. Comparison of in-degree and out-degree

In a first analysis, we compare the in-degree and the out-degree of all developers, i.e., the number of replies given vs. the number of replies received. Figure 3 shows the results of this analysis. We can observe that both measures are highly correlated – developers who receive many replies also write many replies. Thus, for the LK community the activity and the importance of developers correlate highly.

4.2 Comparison of Degree per Group

In this analysis, we want to find out whether the developer-based measures of activity and importance vary from one group to another. We compute the distributions of out-degree and in-degree, for each group for the whole dataset aggregated over all years. The results are shown in Figures 4 and 5.

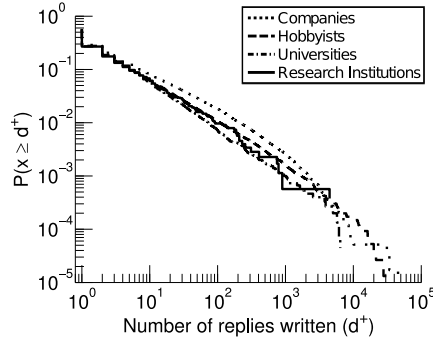


Fig. 4. Out-degree distribution

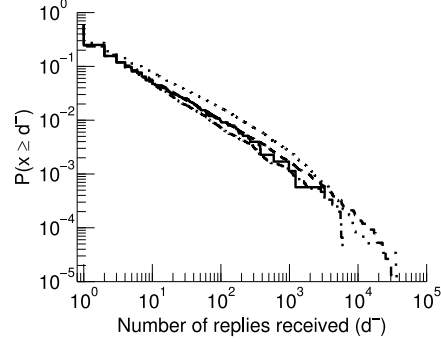


Fig. 5. In-degree distribution

The plots show that:

- The highest activity as measured by the out-degree is achieved by company developers, then hobbyists, and the lowest activity is given by developers from research institutions and universities.
- The measure of importance, the in-degree, correlates and shows the same pattern as for the activity: company developers have the most importance, then hobbyists, and finally developers from research institutions and universities.

These results are consistent with the observation that the measures of activity and importance correlate.

To verify the statistical significance of our results, we perform pairwise Mann–Whitney U tests, testing whether values of each statistic for one type of developer are statistically different from the values for another group. The group differences are statistically significant ($p < 0.05$; company developers vs. hobbyists: $p < 0.10$ for the out-degree), except for developers from companies vs. developers from research institutions and hobbyists vs. developers from research institutions for the in-degree and out-degree; developers from universities vs. developers from research institutions for the out-degree.

4.3 Longitudinal Analysis

In order to study the change of the community over time, we compute three group-wide measures of activity and importance for each individual year in the range 1996 to 2014.

- The average value of the out-degree and the in-degree of all developers in each group, restricted to all replies given and received, respectively during a given year.
- The Gini coefficient of the in-degree distribution of all developers of a given group, restricted to all replies received during a given year.

The results of the analysis are shown in Figures 6, 7 and 8.

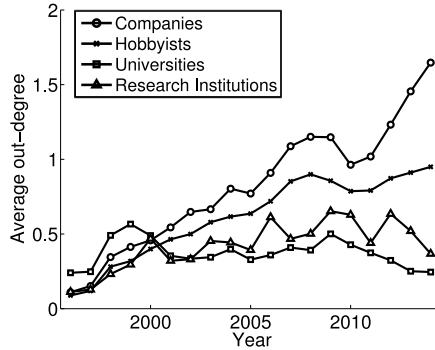


Fig. 6. Average out-degree

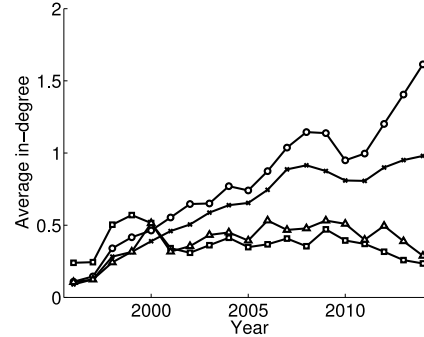


Fig. 7. Average in-degree

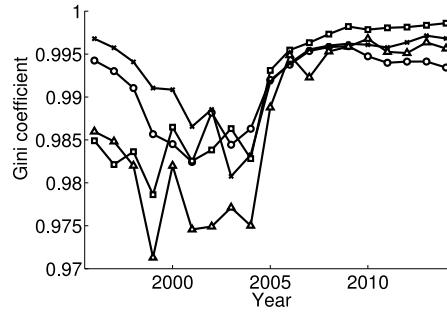


Fig. 8. Gini coefficient

The average out-degree and in-degree (Figures 6 and 7) show a consistent result with the degree distribution shown in Figures 4 and 5. The average out-degree standing for activity of the developers of the different groups increases for the developers from companies as well as hobbyists over time and does not change significantly for the developers of the other contributor categories. The measure of importance, the in-degree, shows a similar behavior. The values for the developers from companies and hobbyists increase and do not change significantly for the other types of developers. The network wide measure Gini coefficient (Figure 8) decreases slightly for developers from companies in the last years. The high fluctuation of the Gini coefficient for universities and research institutions is related to the small group sizes.

Across all developer groups and times, the Gini coefficient is very near to one ($> 98\%$, up to fluctuations). This value is higher than in the large majority of social networks [43], indicating that the importance in the community is concentrated in a very small number of actors when compared to other typical social networks.

5 Discussion and Conclusion

5.1 Discussion

This study provides results of an activity analysis of different contributor groups in the LK development from 1996 to 2014. The aim of this study was to investigate how different contributor groups associated with public and increasing private interests interact in an OSS development project. To achieve this goal, we analyze developers active in the LK development community from a social network perspective, as the interaction between members of a software development community reflects the structure of their collaboration.

The first result of our analysis shows that the out-degree, as a measure for activity, and the in-degree, as a measure of importance, correlate highly both in general and for each contributor group individually. Thus, developers who write many replies to the mailing list also receive many replies. This phenomenon in the LK mailing list differs from forum communication, where a variety of user roles with different communication patterns can be identified [45]. Both the highest activity and importance, is achieved by company developers followed by hobbyists. The lowest activity and importance can be attributed to developers from research institutions and universities. Connecting these results to the amount of contributors per contributor category (Figure 1) it can be seen that although the amount of developers from companies is less than the amount of hobbyists the impact made by employed developers on the LK community is larger. The mentioned impact is expressed by the amount of messages sent per contributor group and year (Figure 2) as well as stated by the measures of activity (Figure 6) and importance (Figure 7), clearly seen especially from 2010 on.

The second result of our analysis shows that the Gini coefficient, as a network wide measure, decreases in recent years for developers from companies and remains constant for the other groups. Although the decrease is small, this can be seen as a tendency that the importance in the community for the group of developers from companies is distributed to more actors. When considering the early years of the LK development (from 1996 until 2000) it can be seen that university members were the most active and important force in the project (Figure 6, 7). From 2001 the activity and importance of hobbyists and companies increases alike. Although the amount of company contributors remains relatively stable for the period starting from 2010 until 2014 (Figure 1) the activity (Figures 2, 6) and importance (Figure 7) in that time increases sharply.

5.2 Conclusion

The aforementioned observations help to answer the research question of how different contributor groups associated with public and increasing private interests interact in an OSS development project, here the LK development. In the beginning, the LK project was driven by intrinsic motivated enthusiasts who are hobbyists and university members (e.g., students) and are associated with pure collective interests. As software and services that were built on top of the LK got more and more influence, the participation of firms, for example, motivated by offering complementary services (see Section 2.1),

increased. As a consequence the private interests in this project increased, too, especially from 2010 on with the diffusion of mobile devices powered by Android [46], which uses the LK as foundation of the operating system.

Summarizing the results of this study, it can be concluded that the balance between private and collective contributors in the LK development seems to be changing to an open source project that is mostly developed jointly by private companies. These firms need the kernel for their products and services. Thus, the LK project is no longer just an open source alternative for hobbyist to develop open source software for the reason not be locked in by proprietary software. The advantages for the participating companies outweigh the drawbacks in terms of collective copyright ownership and less control in the project (see also Section 2.2, Table 1) as the LK community can be utilized to complement their own resource base for innovations.

5.3 Implications for Research

Our findings can be classified into the context of private-collective innovation. The LK development project is an outstanding OSS project. The engagement of companies has increased in the last years, as more and more firms have business models that rely on the kernel. Although companies cannot dictate what the community should do, they can in a way influence the trajectory of the project by assigning employed developers to the project [10, 47]. As the results of our study for the LK project show, employed developers can take key positions in a community due to the intensity of the commitment, expressed by activity and importance, for the community. With this in mind, future research should more thoroughly discuss the nature and structure of firm presence in OSS development. The majority of early research on OSS somehow neglected firm presence and centered on developer motivation while later research discussed emerging OSS business models [e.g., 1]. However, our study calls for more longitudinal studies on firm presence in OSS as (1) firm engagement varies over time and (2) former hobbyists might transform into employed developers (the latter was no focus of this study).

5.4 Limitations and Suggestions for Future Research

Our research has some limitations that have to be considered when utilizing our study's outcomes. The categorization of LK mailing list actors by the hostname of their email addresses into four contributor groups is an approximate but sufficient classification. It is known that there are developers that do not use their company email address while contributing and actors may do personal work out of the office [8]. We mapped developers that occur more than once on the LK mailing list to one person object related to the email address s/he has used when sending a message to the list for the first time. Multiple occurrences happen if the message sender uses different email addresses over time, but the same sender name (e.g., because of company changes). We have not considered these dynamics in our analysis. Furthermore, it has to be considered that the LK project is a unique OSS project, so that our conclusions cannot directly be transferred to other OSS projects where firm-sponsored developers are involved. Further research can consider to investigate the different types and content of the interaction as well as

the aforementioned dynamics of developers or compare the multi-vendor project LK to single-vendor OSS projects.

References

1. Bonaccorsi, A., Giannangeli, S., Rossi, C.: Entry Strategies Under Competing Standards: Hybrid Business Models in the Open Source Software Industry. *Management Science* 52, 1085–1098 (2006)
2. Capra, E., Francalanci, C., Merlo, F., Rossi-Lamastra, C.: Firms' Involvement in Open Source Projects: A Trade-Off between Software Structural Quality and Popularity. *Journal of Systems and Software* 84, 144–161 (2011)
3. Riehle, D.: The Single-Vendor Commercial Open Source Business Model. *Information Systems and e-Business Management* 10, 5–17 (2012)
4. Guignard, C.: Ten Facts and Figures About Open Source Software, <http://blog.nxcgroup.com/2013/ten-facts-and-figures-about-open-source-software/>
5. IDC: Worldwide Open Source Software 2009–2013 Forecast (2009)
6. Grand, S., Von Krogh, G., Leonard, D., Swap, W.: Resource Allocation Beyond Firm Boundaries: A Multi-Level Model for Open Source Innovation. *Long Range Planning* 37, 591–610 (2004)
7. Schaarschmidt, M., Von Kortzfleisch, H.: Divide et Impera! The Role of Firms in Large Open Source Software Consortia. *Proceedings of the 15th Americas Conference on Information Systems (AMCIS)*. San Francisco, USA (2009)
8. Corbet, J., Kroah-Hartman, G., McPherson, A.: Linux Kernel Development. How Fast It is Going, Who is Doing It, What They are Doing, and Who is Sponsoring It. *Linux Foundation*, San Francisco, California (2013)
9. Teigland, R., Di Gangi, P.M., Flåten, B.-T., Giovacchini, E., Pastorino, N.: Balancing on a Tightrope: Managing the Boundaries of a Firm-Sponsored OSS Community and its Impact on Innovation and Absorptive Capacity. *Information and Organization* 24, 25–47 (2014)
10. Schaarschmidt, M., Von Kortzfleisch, H.: Firms' Resource Deployment and Project Leadership in Open Source Software Development. *International Journal of Innovation and Technology Management* 12, 1–19 (2015)
11. Von Hippel, E., Von Krogh, G.: Open Source Software and the “Private-Collective” Innovation Model: Issues for Organization Science. *Organization Science* 14, 209–223 (2003)
12. Stürmer, M., Späth, S., Von Krogh, G.: Extending Private-Collective Innovation: A Case Study. *R&D Management* 39, 170–191 (2009)
13. Alexy, O., Reitzig, M.: Private–Collective Innovation, Competition, and Firms' Counterintuitive Appropriation Strategies. *Research Policy* 42, 895–913 (2013)
14. Riehle, D., Riemer, P., Kolassa, C., Schmidt, M.: Paid vs. Volunteer Work in Open Source. *Proceedings of the 47th Hawaii International Conference on System Science (HICSS)*. Hilton Waikoloa, Big Island (2014)
15. Dahlander, L., Wallin, M.W.: A Man on the Inside: Unlocking Communities as Complementary Assets. *Research Policy* 35, 1243–1259 (2006)
16. Open Source Initiative: The Open Source Definition, <http://opensource.org/osd>

17. Preece, J.: *Online Communities. Designing Usability and Supporting Sociability*. Wiley, New York (2000)
18. Hars, A., Ou, S.: Working for Free? Motivations for Participating in Open-Source Projects. *International Journal of Electronic Commerce* 6, 25–39 (2002)
19. Lakhani, K.R., Wolf, R.: Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. In: Feller, J., Fitzgerald, B., Hissam, S.A., Lakhani, K.R. (eds.) *Perspectives on Free and Open Source Software*, pp. 3–22. MIT Press, Cambridge, MA (2005)
20. Roberts, J.A., Hann, I.-H., Slaughter, S.A.: Understanding the Motivations, Participation, and Performance of Open Source Software Developers: A Longitudinal Study of the Apache Projects. *Management Science* 52, 984–999 (2006)
21. Wu, C.-G., Gerlach, J.H., Young, C.E.: An Empirical Analysis of Open Source Software Developers? Motivations and Continuance Intentions. *Information & Management* 44, 253–262 (2007)
22. Ryan, R.M., Deci, E.L.: Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions. *Contemporary Educational Psychology* 25, 54–67 (2000)
23. Baytiyeh, H., Pfaffman, J.: Open Source Software: A Community of Altruists. *Computers in Human Behavior* 26, 1345–1354 (2010)
24. Stewart, K., Gosain, S.: The Impact of Ideology on Effectiveness in Open Source Software Development Teams. *MIS Quarterly* 30, 291–314 (2006)
25. Lerner, J., Tirole, J.: Some Simple Economics of Open Source. *Journal of Industrial Economics* 50, 197–234 (2002)
26. Dahlander, L., Magnusson, M.: How do Firms Make Use of Open Source Communities? *Long Range Planning* 41, 629–649 (2008)
27. Andersen-Gott, M., Ghinea, G., Bygstad, B.: Why do Commercial Companies contribute to Open Source Software? *International Journal of Information Management* 32, 106–117 (2012)
28. Fitzgerald, B.: The Transformation of Open Source Software. *MIS Quarterly* 30, 587–598 (2006)
29. Dahlander, L.: Appropriation and Appropriability in Open Source Software. *International Journal of Innovation Management* 9, 259–285 (2005)
30. Riehle, D.: The Economic Motivation of Open Source Software: Stakeholder Perspectives. *Computer* 40, 25–32 (2007)
31. Henkel, J.: Open Source Software form Commercial Firms: Tools, Complements, and Collective Invention. *Zeitschrift für Betriebswirtschaft* 4, 1–23 (2004)
32. Hawkins, R.E.: The Economics of Open Source Software for a Competitive Firm. *NETNOMICS: Economic Research and Electronic Networking* 6, 103–117 (2004)
33. Ågerfalk, P.J., Fitzgerald, B.: Outsourcing to an Unknown Workforce: Exploring Open-sourcing as a Global Sourcing Strategy. *MIS Quarterly* 32, 385–409 (2008)
34. Liebeskind, J.P.: Knowledge, Strategy, and the Theory of the Firm. *Strategic Management Journal* 17, 93–107 (1996)
35. Olsen, M.: *The Logic of Collective Action*. Harvard University Press, Cambridge, MA (1967)
36. Von Krogh, G.: The Communal Resource and Information Systems. *The Journal of Strategic Information Systems* 11, 85–107 (2002)

37. Raymond, E.S.: *The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly Media, Inc, Sebastopol, California (2001)
38. Demil, B., Lecocq, X.: Neither Market nor Hierarchy nor Network: The Emergence of Bazaar Governance. *Organization Studies* 27, 1447–1466 (2006)
39. Schaarschmidt, M., Walsh, G., MacCormack, A., Von Kortzfleisch, H.: A Problem-Solving Perspective on Governance and Product Design in Open Source Software Projects: Conceptual Issues and Exploratory Evidence. *Proceedings of the International Conference on Information Systems (ICIS), Research-in-Progress*. Milan, Italy (2013)
40. Lee, G.K., Cole, R.E.: From a Firm-Based to a Community-Based Model of Knowledge Creation: The Case of the Linux Kernel Development. *Organization Science* 14, 633–649 (2003)
41. Wasserman, S., Faust, K.: *Social Network Analysis. Methods and Applications*. Cambridge University Press, Cambridge (1994)
42. Kunegis, J.: KONECT - The Koblenz Network Collection. *Proceedings of the 22nd International Conference on World Wide Web Companion*. Rio de Janeiro, Brazil (2013)
43. Kunegis, J., Preusse, J.: Fairness on the Web: Alternatives to the Power Law. *Proceedings of the Web Science Conference 2012*. Evanston, IL, USA (2012)
44. Kunegis, J., Sizov, S., Schwagereit, F., Fay, D.: Diversity Dynamics in Online Networks. *Proceedings of the 23rd Conference on Hypertext and Social Media*. Milwaukee, WI, USA (2012)
45. Chan, J., Hayes, C., Daly, E.: Decomposing Discussion Forums using Common User Roles. *Proceedings of WebSci10: Extending the Frontiers of Society On-Line*. Raleigh, North Carolina (2010)
46. Google: Android @ I/O: the playground is open. Official Google Blog, <http://google-blog.blogspot.de/2012/06/android-io-playground-is-open.html>
47. Schaarschmidt, M., Walsh, G., Von Kortzfleisch, H.: How Do Firms Influence Open Source Software Communities? A Framework and Empirical Analysis of Different Governance Modes. *Information and Organization* 25, 99–114 (2015)