

# Control vs Freedom: How Companies Manage Knowledge Sharing with Open Source Software Communities

Juliane Wissel  
Technical University of Munich  
[juliane.wissel@tum.de](mailto:juliane.wissel@tum.de)

Michael Zaggl  
Aarhus University  
[zaggl@mgmt.au.dk](mailto:zaggl@mgmt.au.dk)

Aron Lindberg  
Stevens Institute of Technology  
[aron.lindberg@stevens.edu](mailto:aron.lindberg@stevens.edu)

## Abstract

*Knowledge sharing is essential for successful collaboration between companies and external communities. We lack knowledge regarding the micro-processes companies deliberately introduce to manage knowledge sharing with such outside parties. We research these processes in the context of collaboration between companies and open source software (OSS) communities by posing the question: How do companies design explicit mechanisms to manage knowledge exchange with OSS communities? We conduct an explorative case study at Siemens AG. Siemens introduced a formal template process which can be adapted by the organizational units according to their demands. Results show that the extent to which the process is implemented depends on the level of closeness to core intellectual property of the organizational unit and the intensity of the involvement in OSS communities. Developers use several methods to shortcut the process. Our study contributes to the literature on organizational knowledge sharing, company-involved OSS development, and open innovation of firms.*

## 1. Introduction

Knowledge sharing plays an increasingly important role for company-involved open source software (OSS) development. As companies become more and more involved in OSS communities [1, 3, 7, 15, 18, 28], they do not only passively use OSS, but they also need to actively contribute to OSS communities in order to implement specific functionalities and spread their standards. Therefore, the interaction of companies and OSS communities provides a suitable setting for analyzing knowledge flows between organizations as continuous knowledge sharing is essential for a successful collaboration [6, 16].

There are numerous barriers which hinder knowledge sharing, both within and between organizations. Regarding the latter, the existing literature mainly has focused on barriers which refer to the inter-organizational climate and relationship (e.g. lack of trust, conflicting cultures and values) [33]. These barriers are not intentionally set up by organizations and therefore can be described as naturally occurring barriers. While this perspective has proved to be insightful, it does not include mechanisms deliberately used by companies to manage the knowledge exchange with OSS communities.

A qualitative investigation across multiple companies, which we conducted prior to our main study, showed that many of them introduced formal micro-processes to manage their contributions to OSS communities. These processes on the side of the company are crucial to minimize the risk of knowledge spillovers, protect company reputation from being impacted by low-quality contributions, and avoid violation of intellectual property rights. However, if these processes are not well designed they can turn into an undesired knowledge sharing barrier. We lack knowledge about these micro-processes of knowledge contributions from companies to OSS communities and how different organizational units handle them [12].

In order to develop a fundamental understanding of these micro-processes and draw a more complete picture of knowledge exchanges between companies and OSS communities, we approach in this paper the following research question: *How do companies design explicit mechanisms to manage knowledge exchange with OSS communities?*

We approach this question in an explorative case study at Siemens AG. Siemens is an excellent research context because they introduced a centralized micro-process capturing OSS contributions. Organizational units can freely choose whether and how this process is adopted. OSS development plays a central role for Siemens. As a global business-to-business enterprise, the company uses many different technologies in their products and services involving numerous software

components. Further, Siemens is an extensive user of multiple OSS applications.

Based on semi-structured interviews with employees who work related to OSS software (e.g. software developers, managers involved in OSS-related decision-making) and internal documentation, we found that Siemens manages the knowledge flow to OSS communities using a formal process. In our analysis, we distinguish between a normative perspective by examining the intention of the process and a positive (descriptive) perspective by analyzing the actual process implementation.

From the normative perspective, the process is a template describing the sequence from the developer's intention to contribute the code to the final permission/rejection. The specific configuration varies between the different organizational units; that is, the process can be adapted by individual units to suit their specific needs. More concretely, the extent to which the template process is implemented depends on the following characteristics of the units: the level of closeness to the core intellectual property of the unit and the intensity of the involvement in OSS communities (i.e. number and type of contributions, number of OSS communities involved). Further, we identify two forms of company expertise involved in the approval of contributions: legal expertise and technical expertise. Both forms of expertise need to agree; each can independently of the other reject the contribution.

Taking a positive (descriptive) perspective, we find that developers use several methods to shortcut the process. Experienced developers who have gained trust by their unit managers are allowed to engage with specific OSS projects without having to ask for approval for every single contribution. Put simply, they were allowed to abbreviate the process. Further, in isolated cases where contributions happen very rarely, developers made an agreement with their superiors to contribute under their personal identity to avoid the need to establish a formal process.

The study contributes to the literature on organizational knowledge sharing [22, 26, 27] and the governance of company-involved OSS development [24, 32]. We also contribute to the literature on open innovation of firms [5, 8, 9].

## **2. Theoretical background**

### **2.1. Company-involved OSS development**

The collaboration between companies and OSS communities is characterized by intensive knowledge exchange. Companies are increasingly engaging with

the development of OSS [10, 15, 17]. To sustain a relationship with OSS communities, mutual sharing of knowledge is essential [6, 16]. Contributions to OSS projects can be argued to constitute the economically most important non-monetary outbound form of innovation [9].

Companies make use of the knowledge from OSS communities by using OSS in their products. At the same time, they increasingly contribute back to the respective communities in different ways such as sponsorship, letting their developers take over managerial and administrative tasks inside the projects, or contributing code [4]. These activities can all be considered as knowledge sharing activities [29]. There are several reasons for companies to reveal information and knowledge [14]. By getting involved in OSS communities, companies expect to profit from several advantages, including a faster technological development due to the access to external knowledge, increased competitiveness, and attracting skilled developers.

### **2.2. Knowledge sharing activities and barriers**

Being a valuable source of competitive advantage, knowledge is one of the most important resources companies create and hold [13, 23, 30]. Therefore, effective knowledge sharing is essential to enhance market performance [27]. Knowledge sharing is strongly related to further processes like knowledge flow, transfer, learning, and distributed collaboration [12]. It takes place at different levels, including at the individual, team, organizational, and inter-organizational level [33]. The first three levels refer to knowledge sharing within one organization whereas the inter-organizational level is directed towards knowledge exchange across different organizations. Collaboration between different organizations (e.g. between companies and OSS communities) becomes increasingly important as they cannot develop all necessary competencies on their own [19].

At times knowledge sharing practices do not result in desired outcomes, often due to various knowledge sharing barriers. Researchers have identified several barriers dependent upon the level on which the exchange takes place [33]. When looking at knowledge sharing from one organization to another, critical barriers are, for example, the fear of losing competitive advantage, conflicting cultures and values, and a lack of trust between the organizations [33]. All these barriers are mainly rooted in the inter-organizational climate and relationship [20] and can, thus, be characterized as naturally occurring barriers.

The extant literature has focused on analyzing such barriers [27]. In contrast, knowledge about barriers

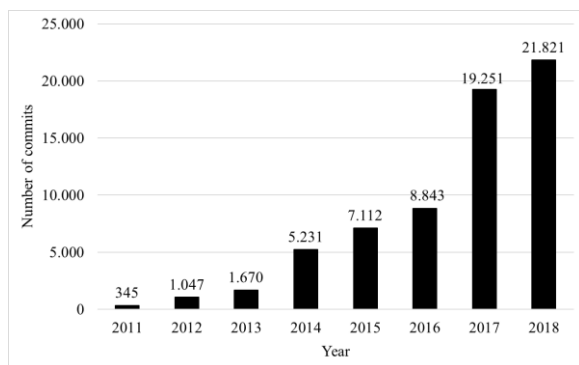
resulting from mechanisms intentionally designed by organizations is scarce. More specifically, the micro-processes companies introduce to manage knowledge contributions to OSS communities are not yet investigated. Furthermore, it is not yet clear how different organizational units deal with these processes. In this study, we aim at addressing this issue.

### 3. Methodology

#### 3.1. Research context

To address our research question, we adopt an in-depth, single case study approach. This research design is appropriate to engender deep understanding of rarely explored phenomena within a real-life setting [11]. We explore Siemens AG, a German multinational conglomerate company with headquarters in Munich and Berlin. Siemens is suitable for this study for two reasons.

First, OSS is a highly relevant topic in many Siemens units. The number of OSS components used in Siemens products is increasing steadily and the awareness of the need for an active engagement with OSS communities is rising among Siemens employees. Figure 1 shows the number of commits on GitHub by Siemens employees per year. The graph illustrates a strong increase in OSS community involvement over the last years.



**Figure 1. Number of GitHub commits by Siemens employees per year**

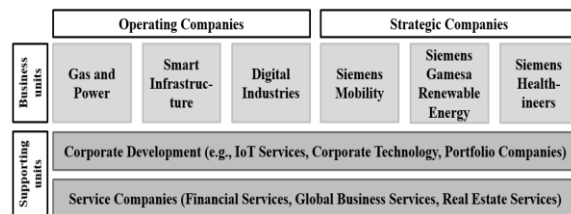
Second, Siemens is structured as multiple business units, each with a large degree of autonomy. The business units use OSS to varying extents and for different purposes. Thus, different demands on the process apply within the company.

Siemens has recently set up a template OSS contribution process. This process has been made available on the intranet so that it can be adopted as it is or adapted to the specific demands of the business

units. This gives us the opportunity to analyze how these business units manage their OSS contributions depending on their specific demands.

#### 3.2. Organizational overview

With about 379,000 employees worldwide and revenue of approximately €83.0 billion in 2018, Siemens is one of the largest producers of energy-efficient technologies. The company is a leading supplier of power generation and transmission systems, medical diagnosis, as well as infrastructure and industry solutions. This portfolio reflects a large diversity of business-to-business products, systems, and solutions. In almost all of the areas Siemens is active in software is gaining importance. Figure 2 shows the organizational structure of Siemens.



**Figure 2. Organizational structure of Siemens**

Below the Group level, there are three *Operating Companies* and three *Strategic Companies* reflecting the core businesses. Each of the *Operating* and *Strategic Companies* is divided into different business units. They are supported by the corporate units from *Corporate Development* and the *Service Companies* which all provide cross-divisional functions across Siemens. The authors are not affiliated with Siemens and, thus, can provide an impartial investigation.

#### 3.3. Data collection

First, we conducted a qualitative pre-study across several different companies to get a deeper understanding of the phenomenon and of the real-life practices. The 19 interviews with employees working on OSS-related issues revealed that many companies introduced formal micro-processes to manage their contributions to OSS communities underlining the currency of the topic.

In our following main study, we also collected qualitative data. For triangulation, we used data from various sources [34]: semi-structured interviews, internal documentation, and direct observations during OSS-related meetings and a company visit. Internal documentation included wiki entries, process

descriptions and visualizations, and checklists. 12 interviews were conducted with software developers and architects dealing with OSS, experts for third party software, and managers involved in OSS-related decision-making. The interviewees are related to two business units and one corporate unit. Table 1 summarizes the interviewee profiles.

**Table 1. Interviewee profiles**

(Related) Unit	Description	Interviewees
Corporate Unit 1 (CU1)	Central research and development unit which provides cross-divisional services along the entire value chain to the business units	Expert for third party software (ETPS1) Research scientist OSS (RS1) Open source expert (OSE1) Open source expert (OSE2) Software developer (SD1) Software developer (SD2)
Business Unit 1 (BU1)	Provides products, systems, and solutions for a reliable transmission and distribution of electrical energy	Expert for third party software (ETPS2) Software architect (SA1) Software developer (SD3)
Business Unit 2 (BU2)	Provides motion control systems and solutions for production and tooling machines	Expert for third party software (ETPS3) Software developer (SD4) Software developer (SD5)

The interviews were guided by a protocol, which was designed prior to data collection according to the research questions [2]. This protocol was adapted to the characteristics of the interviewees and evolved based on the insights from previous interviews. All interviews were audio-recorded with the permission of the respondents and transcribed verbatim. Each interview lasted between 30 and 70 minutes, resulting in about 10.5 hours of recording. The interviews were conducted in English and German. Quotes from interviews conducted in German were translated into English by the authors.

### 3.4. Data analysis

At the beginning of our analysis, we developed a deep understanding of the template process, mainly based on internal documentation and insights during a company visit. This is essential to be able to analyze subsequently the extent to which this process has been adopted in different organizational units and how their specific characteristics influence the process design. The main data source in this step were interviews. We analyzed the data applying an inductive approach [25] supported by the software tool MAXQDA. The coding followed techniques proposed for developing grounded theory, such as open, axial, and selective coding [31]. Throughout the process, we also triangulated the findings across data sources (i.e. interviews, internal documentation, and direct observations) to be able to modify emerging patterns [21].

## 4. Findings

### 4.1. The template OSS contribution process

In the second half of 2017, the demand for a Siemens-wide template for the OSS contribution process came up in the *Open Source Task Force*. This task force aims at connecting all Siemens units dealing with OSS to give them the opportunity to discuss OSS-related topics and exchange experiences: *“This was a topic which popped up after all other topics were handled slowly but surely. How is the clearing to be done, how is everything archived, how is the delivery to be done, etc.”* (ETPS2). In addition to representatives from the legal and the intellectual property department, strategic procurement, and internal IT, the *experts for third party software* of each unit are members of the task force. *Expert for third party software* is a role designated to one person in each unit. It includes the responsibility to make sure that an adequate product clearing is performed to guarantee that third party software components, including OSS, are used according to agreed-upon license terms.

The main reasons for setting up a Siemens-wide template OSS contribution process were (1) to protect employees as well as Siemens’ business interests and reputation, (2) to comply with legal and internal regulations, (3) to provide transparency to decision makers regarding the effect of the contribution on Siemens’ code and intellectual property, and (4) to adhere to the rules and customs of the OSS ecosystem.

The template OSS contribution process was derived from an already existing tool-supported approval process for publications (e.g. conference papers, journal publications). CU1, more specifically the team responsible for Siemens-wide OSS-related issues, took the leading role in the development of the template process as they had already designed a contribution process for their specific unit based on the publication approval process: *“[In our unit] we already have an OSS contribution process for a long time and we brought it into the discussion with the task force as it was already tool-supported. [...] We took the opportunity to say, okay, let’s sit together and design a process that can be used as template process”* (ETPS1). The tool support facilitates identification of persons responsible to be involved in the respective process and documentation of process outcomes. The already existing process for publication approval was adapted to the requirements of OSS contributions. Figure 3 shows the Siemens-internal visualization of the template OSS contribution process.

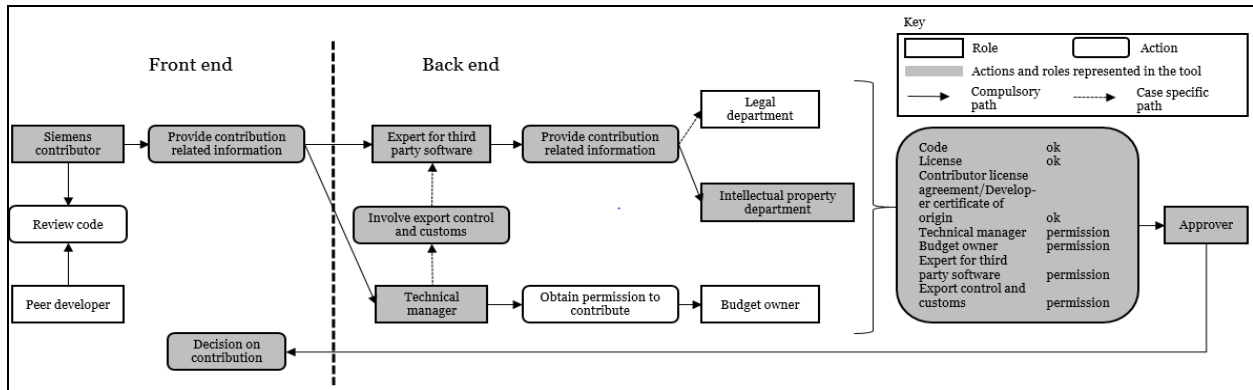


Figure 3. Template OSS contribution process at Siemens

The template process is split into two parts, a frontend and a backend. The frontend marks the part of the process with active engagement of the developer (i.e. a *Siemens contributor*). As a first step, the contributor needs to ensure that the source code is clean and ready for contribution. This step includes a review of the code conducted by an experienced peer developer. Subsequently, the contributor has to provide the following information via the publication approval tool: (1) Name and URL of the OSS project, (2) license of the project, (3) contribution policy of the project (e.g. possible contributor license agreement or developer certificate of origin), (4) context in which the code to contribute was developed, and (5) cleaned source code.

In the next step, the tool automatically informs the *expert for third party software* of the corresponding organizational unit and the *technical manager* (i.e. usually the line manager) that their action is required in the new workflow. The *technical manager* has to confirm that he obtained permission to contribute from the budget owner of the project in which the code has been developed. If the contribution aims at a crypto library, the *technical manager* also has to consult the department for export control and customs.

In case of unclear license terms of the OSS project or the requirement of an unknown contributor license agreement or developer certificate of origin, the *expert for third party software* involves the legal department. The intellectual property department is consulted by default to ensure that no intellectual property is affected by the contribution.

In general, two forms of company expertise are involved in the process: legal expertise and technical expertise. If all parties involved give their permission, the *approver* (i.e. a person with the power to sign in the name of Siemens) gives the final permission to contribute and signs the contributor license agreement, if necessary. If one of the required permissions is not given, the contribution request is rejected.

#### 4.2. Adoption of the template OSS contribution process

According to ETPS1, “dictating a process is always difficult”. Introducing the process as mandatory for the whole organization disregards the fact that each unit has specific demands when collaborating with OSS communities and, hence, strongly restricts the interaction. Therefore, Siemens decided that each unit could choose to adopt the whole process or a modified version of the process or to stick to the already existing procedures. This decision-making scope grants a certain flexibility to the organizational units in their interaction with OSS communities and secures a better compatibility of company and community interests by reflecting the OSS mentality. At the same time, the process ensures that the organizational units comply with external regulations as well as community norms, both of which are essential for maintaining a positive perception of Siemens as a whole.

In our analysis, we identify different process adoption approaches across the investigated corporate and business units. Decisions with regards to what extent the template process is implemented depends on the following specific characteristics of the units: the level of closeness to core intellectual property of the organizational unit and the intensity of the involvement in OSS communities (i.e. number and type of contributions, number of OSS communities involved). Table 2 shows a comparison of the different process adoption approaches in relation to specific characteristics of the organizational units.

In the case of a strong closeness of the unit to core intellectual property and a low intensity of involvement in OSS communities, the template process was fully adopted. One example is BU1. Before the process implementation, BU1 was not contributing back to OSS communities. It was only when two developers with the intention to contribute actively approached the respective *expert for third party software* that the need

for a process arose: *“I pushed [the development of the template process] actively as we had two colleagues who desperately wanted to [contribute code]”* (ETPS2).

**Table 2. Comparison of process adoption approaches**

		Intensity of involvement in OSS communities	
		High	Low
Level of closeness to core IP of the organizational unit	High		Full adoption of the template process (BU1)
	Low	Adoption of the template process with modifications (CU1) No adoption of the template process (rely on existing process) (BU2)	No adoption of the template process (workaround established) (BU2)

BU1 deals with critical infrastructure for energy supply and thus is close to core intellectual property. Developments in this area have to be protected and hence are not intended to be made open source: *“When it comes to functionalities, you always have to discuss. Intellectual property is always an issue. [...] We always have to consider what is core know-how and has to be protected”* (SA1). BU1 fully adopted the template process. Since the process implementation, only two contributions in the form of bug fixes to two different OSS projects have undergone the process indicating a low intensity of involvement in OSS communities. Due to the manageable engagement in OSS communities, the current rule is that every planned contribution has to undergo the process: *“In principle, every change needs to go through [the process]. [...] Should something appear again, same developer, same component, then we might think about shortening it a bit”* (ETPS2).

For units characterized by a low level of closeness to core intellectual property and a high intensity of involvement in OSS communities, we find two possible outcomes. First, the template process was adopted with certain modifications. One example is CU1. In this unit, a large number of developers is actively involved in several different OSS projects. Contributions comprise different types, including feature enhancements. These facts underline a high intensity of involvement in OSS communities.

CU1 decided to adopt the template OSS contribution process. However, certain facilitations are granted which range up to general approvals for developers on OSS project. This means that the developers do not have to undergo the process for each contribution, but only once when asking for approval to engage actively in a specific OSS project under certain conditions (e.g. under a specific license). Thus, the effort not only for the developers but also for all

other people involved in the contribution process is reduced: *“If you are seriously dealing with OSS, [...] you have to find a way which is legally and practically feasible. This means enabling the daily work without leaving the legal framework”* (SD1).

Further, more flexibility for the developers in their interaction with OSS communities is achieved. These general approvals require a certain amount of trust in the developers that they do not leave the set scope of action: *“[The process] comes along with the trust that you as a developer stay in this framework”* (SD1). Thus, it is only granted to senior developers who have already demonstrated both their technical skills and their ability to interact with the target OSS communities according to their rules and practices.

The second outcome for this configuration is that the respective unit sticks to an already existing process instead of adopting the template process. One example is BU2. In this unit, an established OSS contribution process exists, yet not tool-supported. This process is embedded in the product lifecycle management process of BU2. If developers want to make a contribution, they have to fill out the publication request for OSS. This document comprises information about the development context, the OSS itself, and a checklist with the main aspects developers have to consider when planning a contribution. The completed form has to be signed by the *expert for third party software* and a person with the power to sign in the name of Siemens to get the permission to contribute. The permission can also be granted on project level, similar to the facilitation introduced by CU1.

Apart from the missing tool support, the process shows many similarities with the template process. However, it seems to be less complex due to the smaller number of persons involved. The reduced effort and resulting flexibility are highly appreciated by those teams of the unit who make several contributions per day during critical development phases, ranging from bug fixes to feature enhancements: *“[The general approval] was very important for me. If I do several patches a day in a critical development phase, I don’t want to pass multiple hierarchy levels each time to get a permission from someone who most likely cannot evaluate technically what is going on”* (SD4).

The fact that the above-mentioned form was only filled out three times since its creation in 2012 reflects a generally low willingness to contribute in BU2. However, it cannot be completely ruled out that contributions are made without adhering to the process. In a team with a low intensity of involvement in OSS communities a workaround was implemented. An agreement was made between the superiors and the developers which allows them to contribute bug fixes under the personal identity and not on behalf of

Siemens. This procedure was established about 12 years ago when there was no experience with OSS contributions yet to avoid the need to establish a formal process: *“At that time, there were definitely reservations [about OSS], we didn’t know how we would do [contributions]. We agreed that if [the contribution] really does important things, I can do it under my private name instead of contributing it officially in the name of Siemens. In those days, this was the easiest resort without having to set up formal processes”* (SD5).

The effort to create an OSS contribution process was considered as too high compared to the benefit of the contributions. This agreement is still valid today and there are no endeavors to change the procedure so that a small number of contributions stays under the radar.

## 5. Discussion

As company-involvement in OSS development requires intensive knowledge sharing, the collaboration of companies and OSS communities offers a perfect environment to examine knowledge flows between organizations. Companies increasingly engage in OSS communities. To reach the greatest benefit from this collaboration, companies need to continuously exchange knowledge with the communities. However, at the same time, companies need to minimize the risk of knowledge spillovers, protect company reputation that suffers from low-quality contributions, and avoid violations of intellectual property licenses. To balance these contradictory goals, many companies introduce explicit micro-processes. These processes manage their knowledge contributions to OSS communities. If these processes are not well designed, they can also become an unnecessary barrier which hinders knowledge sharing with OSS communities.

In our study, we examined the OSS contribution processes in different organizational units at Siemens. The case of Siemens provides an excellent research context as they recently introduced a centralized micro-process capturing OSS contributions and organizational units could freely decide whether and how to adopt this process.

By introducing this process, the company manages to maintain the balance between controlling their employees and providing them with a certain flexibility and freedom in shaping their knowledge contributions to OSS communities. This flexibility reduces the risk of the process becoming a knowledge sharing barrier due to a possible incompatibility with the specific characteristics of the organizational units. Furthermore, we found that in absence of a contribution process

employees develop own processes adjusted to their own needs to enable knowledge sharing with OSS communities. This occurs in cases with a low level of closeness to core intellectual property and a low intensity of involvement in OSS communities.

The study contributes to several literature streams and to practice. First, we contribute to the literature on organizational knowledge sharing [22, 26, 27] and on the governance of company-involved OSS development [24, 32]. We move the focus from naturally occurring knowledge sharing barriers related to inter-organizational climate and relationship towards artificially set up barriers in the form of micro-processes. Companies need to ensure that all units comply with legal and OSS community regulations. At the same time, each organizational unit has specific demands and characteristics with regard to the interaction with OSS communities. We provide insights into the micro-processes companies introduce to keep the balance between controlling their employees and providing them with a certain flexibility in managing their knowledge contributions to OSS communities. We additionally contribute to the literature on open innovation of firms [5, 8], more specifically on outbound innovations [9].

The results of our study also lead to valuable recommendations for practitioners. First, companies need to provide certain flexibility to the organizational units in adopting the OSS contribution process to accommodate the specifics of the organizational units and to minimize the risk of the process becoming a knowledge sharing barrier. Second, in order to particularly prevent experienced developers from deviating from the process, companies should grant certain facilitations based on technical skills and the experience in the interaction with OSS communities.

## 6. Limitations and future research

We have specifically chosen Siemens for our case study as it provides a suitable research setting as OSS plays a significant role for the company. The organizational units use OSS to varying extents and for different purposes. Further, Siemens employees increasingly engage actively with OSS communities. However, the single case study approach challenges the validity of the findings. Therefore, we plan to investigate the topic in further international companies, concentrating mainly on those complying with certain standards (e.g. CMMI certification).

The study provides us with a solid basis to further research the deviations in daily practice from the normative process. Current literature demands further research on how organizational mechanisms can

trigger desired knowledge sharing behavior and how individuals react to these mechanisms [12]. By further extending the study, we want to address this demand. We plan to describe the divide between the normative and the descriptive process in more detail. Some indications of this divide have already been identified; however, we plan to investigate further organizational units at Siemens to get a holistic view on the process adoption approaches. We especially aim at units with a high level of closeness to core intellectual property and a high intensity of involvement in OSS communities.

Building on the divide, we want to get richer insights into further potential workarounds that teams have established instead of adopting the template OSS contribution process. Finally, we want to analyze archival data (e.g. GitHub data, mailing lists, blog posts) from specific OSS communities Siemens is involved in to gain deeper knowledge about the actual community involvement of Siemens.

## 7. References

- [1] P. J. Ågerfalk and B. Fitzgerald, "Outsourcing to an Unknown Workforce: Exploring Opensourcing as a Global Sourcing Strategy", *MIS Quarterly* (32:2), 2008, pp. 385-409.
- [2] M. Alvesson, "Beyond Neopositivists, Romantics, and Localists: A Reflexive Approach to Interviews in Organizational Research", *Academy of Management Review* (28:1), 2003, pp. 13-33.
- [3] A. Bonaccorsi, S. Giannangeli, and C. Rossi, "Entry Strategies Under Competing Standards: Hybrid Business Models in the Open Source Software Industry", *Management Science* (52:7), 2006, pp. 1085-1098.
- [4] E. Capra, C. Francalanci, F. Merlo, and C. Rossi Lamastra, "A Survey on Firms' Participation in Open Source Community Projects", *Open Source Ecosystems: Diverse Communities Interacting*, C. Boldyreff, K. Crowston, B. Lundell, and A. I. Wasserman (eds.), Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 225-236.
- [5] H. W. Chesbrough, *Open Innovation: The New Imperative for Creating and Profiting From Technology*, Harvard Business School Press, Boston Mass., 2003.
- [6] C.-M. Chiu, M.-H. Hsu, and E. T. G. Wang, "Understanding Knowledge Sharing in Virtual Communities: An Integration of Social Capital and Social Cognitive Theories", *Decision Support Systems* (42:3), 2006, pp. 1872-1888.
- [7] M. G. Colombo, E. Piva, and C. Rossi-Lamastra, "Open Innovation and Within-Industry Diversification in Small and Medium Enterprises: The Case of Open Source Software Firms", *Research Policy* (43:5), 2014, pp. 891-902.
- [8] T. Cui, H. Ye, H. H. Teo, and J. Li, "Information Technology and Open Innovation: A Strategic Alignment Perspective", *Information & Management* (52:3), 2015, pp. 348-358.
- [9] L. Dahlander and D. M. Gann, "How Open is Innovation?", *Research Policy* (39:6), 2010, pp. 699-709.
- [10] L. Dahlander and M. Magnusson, "How do Firms Make Use of Open Source Communities?", *Long Range Planning* (41:6), 2008, pp. 629-649.
- [11] K. M. Eisenhardt, "Building Theories from Case Study Research", *Academy of Management Review* (14:4), 1989, pp. 532-550.
- [12] N. J. Foss, K. Husted, and S. Michailova, "Governing Knowledge Sharing in Organizations: Levels of Analysis, Governance Mechanisms, and Research Directions", *Journal of Management Studies* (47:3), 2010, pp. 455-482.
- [13] R. M. Grant, "Toward a Knowledge-Based Theory of the Firm", *Strategic Management Journal* (17:S2), 1996, pp. 109-122.
- [14] J. Henkel, "Selective Revealing in Open Innovation Processes: The Case of Embedded Linux", *Research Policy* (35:7), 2006, pp. 953-969.
- [15] S. Y. Ho and A. Rai, "Continued Voluntary Participation Intention in Firm-Participating Open Source Software Projects", *Information Systems Research* (28:3), 2017, pp. 603-625.
- [16] J. Kotlarsky and I. Oshri, "Social Ties, Knowledge Sharing and Successful Collaboration in Globally Distributed System Development Projects", *European Journal of Information Systems* (14:1), 2005, pp. 37-48.
- [17] G. von Krogh, S. von Haefliger, S. Spaeth, and M. W. Wallin, "Carrots and Rainbows: Motivation and Social Practice in Open Source Software Development", *MIS Quarterly* (36:2), 2012, pp. 649-676.
- [18] R. Macredie and K. Mijinyawa, "A Theory-Grounded Framework of Open Source Software Adoption in SMEs", *European Journal of Information Systems* (20:2), 2011, pp. 237-250.
- [19] A. Majchrzak, S. L. Jarvenpaa, and M. Bagherzadeh, "A Review of Interorganizational Collaboration Dynamics", *Journal of Management* (41:5), 2015, pp. 1338-1360.
- [20] R. McDermott and C. O'Dell, "Overcoming Cultural Barriers to Sharing Knowledge", *Journal of Knowledge Management* (5:1), 2001, pp. 76-85.
- [21] M. B. Miles and A. M. Huberman, *Qualitative Data Analysis: An Expanded Sourcebook*, Sage Publ., Thousand Oaks, 2009.

- [22] M. E. Nissen, "An Extended Model of Knowledge-Flow Dynamics", *Communications of the Association for Information Systems* (8:1), 2002, pp. 251-266.
- [23] M. E. Nissen, R. J. Orr, and R. E. Levitt, "Streams of Shared Knowledge: Computational Expansion of Knowledge-Flow Theory", *Knowledge Management Research & Practice* (6:2), 2008, pp. 124-140.
- [24] S. O'Mahony and B. A. Bechky, "Boundary Organizations: Enabling Collaboration Among Unexpected Allies", *Administrative Science Quarterly* (53:3), 2008, pp. 422-459.
- [25] M. Q. Patton, *Qualitative Evaluation and Research Methods*, Sage Publ., Thousand Oaks, 2015.
- [26] L. Razmerita, K. Kirchner, and P. Nielsen, "What Factors Influence Knowledge Sharing in Organizations? A Social Dilemma Perspective of Social Media Communication", *Journal of Knowledge Management* (20:6), pp. 1225-1246.
- [27] A. Riege, "Three-Dozen Knowledge-Sharing Barriers Managers Must Consider", *Journal of Knowledge Management* (9:3), 2005, pp. 18-35.
- [28] B. Rolandsson, M. Bergquist, and J. Ljungberg, "Open Source in the Firm: Opening up Professional Practices of Software Development", *Research Policy* (40:4), 2011, pp. 576-587.
- [29] S. K. Sowe, I. Stamelos, and L. Angelis, "Understanding Knowledge Sharing Activities in Free/Open Source Software Projects: An Empirical Study", *Journal of Systems and Software* (81:3), 2008, pp. 431-446.
- [30] J. C. Spender, "Making Knowledge the Basis of a Dynamic Theory of the Firm", *Strategic Management Journal* (17:S2), 1996, pp. 45-62.
- [31] A. L. Strauss and J. M. Corbin, *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*, Sage Publ., Thousand Oaks, 2003.
- [32] R. Teigland, P. M. Di Gangi, B.-T. Flåten, E. Giovacchini, and N. Pastorino, "Balancing on a Tightrope: Managing the Boundaries of a Firm-Sponsored OSS Community and its Impact on Innovation and Absorptive Capacity", *Information and Organization* (24:1), 2014, pp. 25-47.
- [33] P. Yih-Tong Sun and J. L. Scott, "An Investigation of Barriers to Knowledge Transfer", *Journal of Knowledge Management* (9:2), 2005, pp. 75-90.
- [34] R. K. Yin, *Case Study Research: Design and Methods*, Sage Publ., Los Angeles, 2009.