Information and Organization 25 (2015) 99–114

Contents lists available at ScienceDirect

Information and Organization

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

How do firms influence open source software communities? A framework and empirical analysis of different governance modes

Mario Schaarschmidt [a,*], Gianfranco Walsh [b], Harald F.O. von Kortzfleisch [a]

[a] Institute for Management, University of Koblenz–Landau, Universitätsstrasse, 156070 Koblenz, Germany
[b] Friedrich-Schiller-University of Jena, Carl-Zeiss-Strasse 3, 07743 Jena, Germany

ARTICLE INFO

Article history:
Received 17 February 2014
Received in revised form 29 March 2015
Accepted 31 March 2015
Available online xxxx

Keywords:
Open source software
Single-vendor projects
Multivendor projects
Communities
Governance modes
Control theory

ABSTRACT

This paper explores how software firms can apply different types of governance approaches to open source software development projects (OSSDPs) and draws on control theory to propose that firms may influence OSSDPs by employing either leadership or resource deployment control. A matrix differentiating four types of OSSDPs: firm- versus community-initiated projects and one participating firm (single-vendor projects) versus many firms (multivendor projects), and accompanying hypotheses regarding a firm's participation for each type are developed. Using data from 83 Eclipse projects to test the hypotheses, findings indicate that (1) firms more actively employ both leadership and resource deployment in firm-initiated projects than in community-initiated ones and (2) firms are more likely to use resource deployment control over leadership control in multivendor projects. Key theoretical and managerial implications are discussed.

1. Introduction

Decreasing license fees in the software market have given momentum to the open source software (OSS) development approach as a viable alternative to proprietary software development approaches (Augustin, 2008; Lerner & Schankerman, 2010; Teigland, Di Gangi, Flaten, Giovacchini, & Pastorino, 2014). Recent estimates exemplify OSS success and expect the OSS software market to be worth $46 billion in 2015 (Statista, 2015). Revenue models associated with OSS range from dual licensing, which involves offering a product under an OSS license and (at least) one proprietary license, to revenue streams generated entirely through

* Corresponding author. Tel.: +49 261 287 2856; fax: +49 261 287 2851.
E-mail addresses: mario.schaarschmidt@uni-koblenz.de (M. Schaarschmidt), walsh@uni-jena.de (G. Walsh), harald.von.kortzfleisch@uni-koblenz.de (H.F.O. von Kortzfleisch)

http://dx.doi.org/10.1016/j.infoandorg.2015.03.001
1471-7727/

the sale of complementary products or services (Bonaccorsi, Giannangeli, & Rossi, 2006; Fitzgerald, 2006; Olson, 2005). The OSS source code is written in a human readable programming language and open to anyone, so capable users, whether individuals or firms, can modify the code according to their own needs (Von Hippel & Von Krogh, 2003). If a modification contributes to the quality of the original software (e.g., fixing a bug, adding new functionality), users often grant that extension freely to the project, which provides them greater reputational gains (Roberts, Hann, & Slaughter, 2006; Xu, Jones, & Shao, 2009). Thus, an OSS project often involves a relatively heterogeneous community of developers, bug fixers, users, and firms, if the project raises commercial interest (Dahlander & Magnusson, 2008). As communication within this heterogeneous group is characterized by computer-mediated interactions, OSS constitutes a relationship between information technology and organization that is different from relationships known to exist in traditional organizational forms (Simon, 1973).

From the firm's point of view, structures as those in OSS allow for integrating external resources (Grand, von Krogh, Leonard, & Swap, 2004). If a software firm can encourage voluntary external work, its development costs decrease, because in these situations the firm does not have to pay for the external contributions. Many firms increasingly initiate their own OSS projects rather than building on existing ones (Bonaccorsi & Rossi, 2006; Dahlander, 2007; Dahlander & Magnusson, 2008; Riehle, 2012). Thus, there is robust support for the existence of different types of open source software development projects (OSSDPs), but a dearth of research examines the differences in control mechanisms that govern such projects. We propose a conceptual framework based on (1) the distinction between projects that feature one dominant firm (single-vendor projects; SVP) and those with more than one participating firm (multivendor projects; MVP), and (2) whether they were initiated by a firm or the community of developers, and use this framework to analyze firm influence in OSSDPs.

As firms increasingly deploy resources to OSSDPs (Ghapanchi, 2013; Ghapanchi, Wohlin, & Aurum, 2014), they need to gain a better understanding of available governance modes that might enable them to exert some influence or even control. The proposed framework guides us in analyzing how firms make use of different options available to influence development communities. Conventional governance mechanisms, such as behavior or output control (Ouchi, 1979), are inapplicable to systems based on volunteer work where no contractual relationship exists (Lattemann & Stieglitz, 2005). However, developers who earn salaries from their employer are simultaneously embedded in organizational settings (Dahlander & O'Mahony, 2011; Dahlander & Wallin, 2006; Henkel, 2009). By inserting their norms and beliefs, which partly reflect the employing firm's influence, into the OSSDP, these employees allow firms to (indirectly) influence the project's trajectory—depending on the number of programmers assigned and their role in the project. Building on organizational control theory and governance literature, which distinguishes formal from informal control mechanisms (Aghion & Tirole, 1997; Kirsch, 1997; Ouchi, 1979; Rustagi, King, & Kirsch, 2008), we pursue a better understanding of different governance mechanisms in different types of OSSDP.

Based on the proposed framework and different options of influence, we analyze firm activity in projects that are hosted on Eclipse, a firm-dominated repository for business-related OSSDPs. We find support for our framework and identify different governance modes that in turn stem from different business interests.

The remainder of the paper is organized as follows. First, we review and synthesize literature on OSSDPs and organizational control. We then develop the rationale for the use of alternative conceptualizations of control followed by the development of our hypotheses. Next, we describe our research methods and results. We conclude by discussing theoretical and managerial implications.

2. Theoretical background and hypothesis development

2.1. A framework for classifying open source software projects

Literature on innovation in online communities and OSS is diverse (Aksulu & Wade, 2010). One stream of literature centers on online collaboration in various contexts such as music (e.g., Dahlander & Frederiksen, 2012; Jarvenpaa & Lang, 2011), cars (e.g., Füller, Matzler, & Hoppe, 2008), and software (e.g., O'Mahony & Ferraro, 2007). Although this research was heavily influenced by early research on free and open source software projects, OSS usually functions as an example of innovation with or by users, but is not at the core of the research interest. Another stream of research is more concerned with OSS as such and investigates internal organization of OSSDPs (e.g., Dahlander & O'Mahony, 2011; Rullani & Haeflinger, 2013; Stewart & Gosain,

2006), or how OSS might complement professional firms' innovation activities (e.g., Dahlander, 2005; Enkel, Gassmann, & Chesbrough, 2009; Stam, 2009). For example, Crowston and Howison (2006) analyzed 122 large projects and found that not all projects are highly centralized as one might surmise. However, apart from few exceptions (e.g., West & Gallagher, 2006), only recently research has shifted from studying firms in their role as users to firms as developers of OSS (e.g., Andersen-Gott, Ghinea, & Bygstad, 2012; Teigland et al., 2014).

For this study, we classify OSSDPs as projects with one participating firm, or SVPs, and those in which more than one firm is active, or MVPs. The former are similar to a situation with proprietary software vendors, especially if they feature a dual licensing[1] approach (Augustin, 2008; Olson, 2005; Riehle, 2012). The latter tend instead to mimic R&D alliances or joint ventures (Schaarschmidt, Walsh, MacCormack, & Von Kortzfleisch, 2013; Westergren & Holmström, 2012) and might not entail a direct revenue stream. Instead, multiple firms combine their resources to build a platform and promote standards, which will enable them to sell on-top applications and complementary products or services. Moreover, MVPs usually aim to reduce product costs through cooperation in product development.

Ownership structures are the key characteristics for differentiating SVP and MVP. Whereas intellectual property (IP) related to MVPs is spread and owned by a community of stakeholders (including multiple individual programmers and/or firms), IP related to SVPs is owned (and therefore controlled) by one stakeholder with an explicitly commercial purpose (Riehle, 2012; Schaarschmidt et al., 2013). In addition, Dahlander (2007) distinguishes OSSDPs by their impetus, because a project initiated by a firm versus a community likely exhibits different norms and beliefs. We therefore propose a framework based on these two distinctions (Table 1).

In SVPs initiated by a firm, the community does not exist a priori, nor will a firm-initiated SVP generally accept external contributions to its code base unless the contributor transfers the copyright to the vendor (Olson, 2005). In contrast, community-initiated SVPs begin with the community, which has no commercial interests a priori. As the project matures and develops commercial potential, an external investor (e.g., software vendor) or community founder creates a company with the help of venture capital (Schaarschmidt & Von Kortzfleisch, 2014). To generate revenues from dual licensing, a community-initiated SVP also needs to hold the entire copyright for the software product, which might be a threat if not all programmers active in the community migrate to the created company.

Firms engaging in a MVP do not intend to receive direct revenue from selling the software product. Instead, they contribute to a project together with other parties (typically other firms or voluntary programmers) that also deploy their resources (West & O'Mahony, 2008). Thus, the copyright for the developed product is widely distributed. Without ownership of the entire copyright, no party can benefit directly in a commercial sense, such as by pursuing a dual licensing approach, as the resulting artifact shares characteristics of a public good. As SVPs, MVPs might be initiated by either a firm (e.g., Google's Android) or the community (e.g., Linux kernel).

2.2. Governance and control in organizations

Most governance literature stems from two streams of research; political science and corporate management (Ruhanen, Scott, Ritchie, & Tkaczynski, 2010). The research sparked by corporate management ideas tends to investigate how organizations can exert control over strategically important activities and processes on a micro level (Choudhury & Sabherwal, 2003; Kirsch, 1997; Ouchi, 1979) and how to interact with external parties on a macro level (Eisenhardt, 1985; Ouchi, 1977; Williamson, 1985). In line with transaction cost economics (Williamson, 1985), the governance modes on a macro level represent forms of cooperation, ranked along a continuum from integrated (or vertical integration) to market transactions (Van de Vrande, Vanhaverbeke, & Duysters, 2009).

[1] Firms that use dual licensing provide a single software product under two licenses, one proprietary and one OSS license, but only charge for the proprietary license, which usually offers enhanced functionalities (Olson, 2005). At the core of this strategy are the different rights associated with software licenses. Software licenses regulate what users can or cannot do with the software. Under the General Public License (GPL), extensions to the code must be open source again and may not be distributed as proprietary software (see Santos, Kuk, Kon, & Pearson, 2013). Other forms of software licenses such as Mozilla Public License (MPL) are more business friendly as they allow greater interaction with proprietary software.

Table 1
Typology of commercialization approaches.

	SVP	MVP
Project initiated by firm	Approach I JBoss, MySQL	Approach II Android
Project initiated by community	Approach III SugarCRM, Sleepycat	Approach IV Linux

In organization theory, governance on the micro level combines different mechanisms to encourage people to do things that align with the organization's preferences (Choudhury & Sabherwal, 2003), so it can be defined as "any process by which managers direct attention, motivate, and encourage members to act in desired ways to meet the firm's objectives" (Cardinal, 2001, p. 22). Because control entails a portfolio of mechanisms designed to influence employees (Cardinal, 2001; Jaworski, 1988; Kirsch, 1997), governance and control are often and confusingly used interchangeably.

We specify a behavioral view of control, defined as "attempting to ensure individuals […] act in a manner that is consistent with achieving desired objectives" (Choudhury & Sabherwal, 2003, p. 292). We draw on the broader framework developed by Ouchi (1977, 1979), which is based on antecedent conditions of control such as the level of task programmability (i.e., knowledge about the transformation process available to a controller) and output measurability. Depending on whether the task is definable and the output of the task is measurable, *behavior*, *outcome*, and *clan control* are distinguished. For example, output control is suitable only if it is possible to measure employee activities, such as counting output produced by factory workers (Cardinal, Sitkin, & Long, 2004; Eisenhardt, 1985). Behavior control instead seeks to influence employees by defining rules and procedures to be followed (Kirsch, 1997; Ouchi & Maguire, 1975). Output control requires relatively little management direction, but behavior control depends completely on monitoring systems, which then require hierarchical management layers.

However, in the absence of clear task specification and output measurability, clan control is more likely within organizations; it can be achieved by minimizing divergent preferences among organizational members through socialization (Eisenhardt, 1985). By internalizing its goals, every employee shares the company's vision and contributes to the organizational culture. Control is particularly important in cases of divergence, and common goals, norms, and beliefs help decrease the level of required control, in that the group pursues ceremonies and acts like a clan (Kirsch, Ko, & Haney, 2010; Ouchi, 1979). Such clan controls, based on shared norms and beliefs, are particularly observable in start-ups where the applicability of other forms of control is limited (Stevenson & Jarillo, 1990). Whereas behavior and outcome controls constitute formal modes embedded in a bureaucratic organization design, clan control is more informal (Aghion & Tirole, 1997; Blau & Scott, 1962; Ouchi, 1977).

Yet, an often neglected influence mechanism over employee behavior for various modes of control is leadership, which is scarcely discussed in prior research (Abernethy, Bouwens, & van Lent, 2010). Similar to control, leadership comprises formal leadership, achieved through positions defined by a job description, and informal leadership, which likely results from the leader's charisma and has little to do with his or her position in the organization (Conger & Kanungo, 1987). Leadership in a bureaucratic sense implies decision-making competence, which corresponds to the highest form of influence (e.g., Arya, Glover, & Sivaramakrishnan, 1997). Thus, leadership offers strong access to influence and control in organizational settings (e.g., Choudhury & Sabherwal, 2003; Druskat & Wheeler, 2003).

In summary, control theory offers a framework to illustrate how environmental factors and control mechanisms depend on different antecedents and influence organizational outcomes variously. Leadership, both formal and informal, can constitute a strong path to influence in each control mode. However, control mechanisms such as hierarchies within an (bureaucratic) organization rely on threatening consequences, so it becomes difficult to implement them in systems based (at least partly) on volunteer work (Lattemann & Stieglitz, 2005); these settings instead require trust and intrinsic motivation as more adequate governance mechanisms (Shah, 2006; Stewart & Gosain, 2006). We therefore need to discuss how firms might seek to obtain influence or even control in OSSDPs.

2.3. Firm influence and control in open source software development

Firms pursue strategic interests when interacting with an OSS development community (Grand et al., 2004). As outlined previously, among these interests are reducing their development costs, seeking external knowledge, and extending distribution channels—interests that manifest in different forms of OSSDPs and business models. It is self-evident that in order to benefit from these strategies, community work has somehow to be aligned with what happens inside the boundaries of the firm. For example, a software vendor that relies on OSS (e.g., RedHat, Suse) might find himself in the following situation: A business customer requests a functionality that is in contrast to what the community sees as relevant for the project. The vendor then might frustrate or even lose the customer by not fulfilling the request, maintain a duplicate version of the software explicitly for a single customer (a very costly approach), or try to convince the community to change the development trajectory.

Thus, as community and firm interests do not entirely overlap (O'Mahony & Bechky, 2008), firms seek options to influence community work to avoid such situations. Firms may choose from a variety of options to control. For example, Gallivan (2001) identified strong explicit (i.e., rules and norms provided in the documentation and agreements), and implicit (i.e., emphasis on individual reputation) forms of control in open source development practices. In addition, participating in discussions via newsgroups and mailing lists, fixing bugs and suggesting new functionalities can be valid forms of influence. However, among the most powerful approaches to influence a project's trajectory is committing source code. Being a committer involves having the right to contribute or delete code artifacts, thus being able to directly influence a project's development trajectory.

We build on and extend these considerations by introducing two forms of resource-based control firms might pursue that build on commits: “Control by Leadership” and “Resource Deployment Control”. Generally a firm's main function should not be to control employees. Control is costly, because it requires monitoring activities (Anderson & Oliver, 1987), may cause negative effects when perceived by employees (Brockner et al., 2004), and hinders innovation (Cardinal, 2001). Considering the absence of contract-based regulations and intellectual property protection in OSSDP with multiple firms, perceived divergence might be considered high, because firms' own workers, employees of other firms, and voluntary contributors all have an interest in influencing the project trajectory (Bonaccorsi & Rossi, 2006). Governance then should vary for paid human resources versus freely available ones. A paid employee, generally speaking, accepts certain responsibilities in exchange for remuneration, such as following a supervisor's instructions. In contrast, voluntary contributors are not contractually linked to the firm and may choose to retire from the project at any time (Shah, 2006). Relying on contributions from volunteers thus may be risky if the firm cannot bind these key contributors.

In line with the control modes proposed by control theory (e.g., Choudhury & Sabherwal, 2003; Kirsch, 1997; McMahon & Ivancevich, 1976), we argue that firms that want to influence a project's trajectory have two resource-based options: (1) control by leadership (CBL) which may help shaping explicit control in Gallivan's (2001) sense (e.g., through decision-making competence) and (2) resource deployment control (RDC), a more implicit mode. The first control mode depends on the firm's ability to install its own employees as project leaders, perhaps by giving those employees sufficient time to climb the meritocracy ladder in the community (Dahlander & O'Mahony, 2011; Giuri, Rullani, & Torrisi, 2008) or by hiring existing project leaders. Relatedly, leadership positions in an OSSDP may correlate with sustained contributions (Dahlander & O'Mahony, 2011; Markus, 2007; Scozzi, Crowston, Eseryel, & Li, 2008). When employees appear in leadership positions and possess decision-making responsibilities (Giuri et al., 2008; Henkel, 2009; Stewart, Ammeter, & Maruping, 2006), the firm can influence and control the OSSDP as it desires, because the firm controls the project leaders outside the OSSDP in a contractual manner. To avoid bearing ill consequences in cases of bad performance or misbehavior, project leaders likely act in accordance with the employing firm's interests.

Firms might apply what we suggest as a second option of influence, namely RDC. As discussed in the section on control modes, in the absence of prerequisites for formal control, clan control that reflects shared goals and ideology is appropriate (Barker, 1993; Stewart & Gosain, 2006). In relation to OSSDP, we distinguish two clans—one within the community and one within the focal company. As an employee of the firm and a member of the OSSDP community, a developer becomes embedded in two different systems of norms and beliefs (Dahlander & Wallin, 2006; Henkel, 2009; Teigland et al., 2014) and might build a dual identification (George & Chattopadhyay, 2005). By assigning developers who have internalized firm norms to an OSSDP, the firm

might leverage their own resources to obtain influence over the project—without formally applying for leadership positions. This resource deployment influence differs from classical clan control, where shared goals and aims evolve entirely within the community. However, injecting firm-socialized resources into a community increases the likelihood that the community norms and values evolve in a way congruent with those of the firm. The effect is even greater, when many firm-sponsored developers earn committer status, a status that typically involves working at the source files and electing project leaders. We therefore focus on committers in this study.

Thus, CBL may be viewed as an OSSDP-specific reflection of behavior and output control while RDC might be considered an OSSDP-specific reflection of clan control which takes into account developers' and committers' dual identification. In Table 2 (see also Kirsch, 1997, p. 219 for a comparison), we provide an overview of different control modes that firms can apply to achieve influence or even control over an OSSDP.

2.4. Hypothesis development

Our hypotheses build upon the two-dimensional framework developed in 2.1. The core of this framework is the distinction between type of project (SVP vs. MVP) and whether a project was initiated by a firm or the community. In all approaches firms may apply CBL or RDC to influence a project's trajectory—although these strategies make more sense in MVPs, where multiple firms aim to increase their level of influence. Both strategies require that the firm assigns additional resources to the OSSDP. Thus, we argue that the firm's ability to extend its level of influence is a function of the resources the firm provides. For example, firms can deploy multiple project leaders in large software projects (e.g., Mockus, Fielding, & Herbsleb, 2002). If they can capture more leading positions, they also gain more decision-making competence and CBL. In a similar vein, assigning more firm-sponsored committers to a project represents RDC, which further increases the firm's reputation within the OSSDP (Henkel, 2009). Intel and RedHat both deploy numerous maintainers, who are allowed to work on the source code, to the Linux kernel for example.

Previous research concurs that a considerable number of programmers earn pay for their work on OSSDPs (e.g., Lakhani & Wolf, 2005; Roberts et al., 2006; Xu et al., 2009). These developers are sponsored by different firms that have different interests in the project. While there is reason to believe that firms generally seek influence in any kind of OSSDP, there is also support for arguing that the level of influence is higher for projects that firms initiated. Firm-initiated projects reflect a firm's core interests and therefore perceive high importance compared to projects initiated by a community. Furthermore, firm-initiated OSSDPs might signal lower risks for late entrants (West & O'Mahony, 2008). As a consequence of multiple interests embedded in these projects, more firms start to employ RDC (i.e., assigning more firm-sponsored developers) and, in the long-run, CBL (i.e., more leadership positions are generated). We therefore posit for both modes of OSSDP-specific control that:

H1a. Firms make more use of control by leadership in firm-initiated projects than in community-initiated ones.

H1b. Firms make more use of resource deployment control in firm-initiated projects than in community-initiated ones.

Table 2
Mechanisms of control for firms participating in OSSDP.

	Behavior control	Output control	Clan control
Antecedent condition	Knowledge of appropriate behaviors Behavior observability	Output measurability	Appropriate behaviors unknown, outcomes not measurable
Mechanism in firm	Job description Hierarchy	Defined target, expected level of performance	Socialization Rituals and ceremonies
Mechanism in community	Leadership, project milestones	Leadership, Project milestones	Socialization Rituals and ceremonies
Mechanism firm in community	Acquisition of project leaders Control by leadership	Acquisition of project leaders Control by leadership	Assigning developers socialized within the firm Resource deployment control

Previous studies have argued that different modes of collaboration can be assigned to a continuum of integration (Villalonga & McGahan, 2005). We adopt the idea proposed by Van de Vrande et al. (2009) and rank the approaches in our framework (firm-initiated SVP, community-initiated SVP, firm-initiated MVP, community-initiated MVP) along a continuum from the integration of external parties to market transactions, from the firm's perspective.

Firm-initiated SVPs exhibit some similarities to proprietary software vendors (Riehle, 2012). The absence of an active developer community means that SVPs initiated by a firm consist of hierarchical structures. They accept contributions from outside (e.g., from volunteers or other firms) only if they can obtain the copyright for those contributions, which implies a more integrated collaboration mode. Although community-initiated SVPs share the same restrictions with regard to copyright ownership, the community of external developers might be more active and enforce a less integrated mode (O'Mahony, 2007). When MVPs are initiated by a firm, they often seem unattractive to external developers, who do not want to provide their resources for free to serve firm profit goals (Hahn, Moon, & Zhang, 2008). Unlike SVPs though, other firms participate to ensure the project's continuance, creating more interested stakeholders (De Laat, 2007). In MVPs initiated by a community, the firm must manage interactions with other firms and coordinate a community of external volunteers, who have engaged in the project long before the firm joined. The composition of these different external parties implies a loose coupling of stakeholders, such that community-initiated MVP is the least integrated form of collaboration. These governance modes in turn require different control mechanisms.

In least integrated governance modes, that is, in community-initiated MVPs, motivations for engaging in the OSSDP are arguably as diverse as the volunteers working on the project, and different firms want to align community work with their own interests. As the project was initiated by the community, there is reason to believe that community representatives are comparatively active in these kinds of projects. On the other side of the continuum, that is, firm-initiated SVPs, a single firm's influence is dominant such that voluntary committers might not be attracted to the same degree as they are to community-initiated MVPs. Thus, we surmise that integrated governance modes such as firm-initiated SVP display a more firm-driven approach than less integrated modes. As SVPs and MVPs are likely to differ in size, we suggest that the ratio of firm-sponsored to all committers is higher in more integrated modes.

H2. With more integrated governance modes, the ratio of firm-sponsored to all committers increases.

In extension to H2, we take a closer look at MVPs, as for SVPs the number of stakeholders is lower which results in fewer diverse interests. As argued above, the multiple firms in MVPs may have diverse views of a project's trajectory, which manifest in different interests (Almirall & Casadesus-Masanell, 2010; West & O'Mahony, 2008). In turn, these different interests are pursued through different forms of taking influence and thus the use of CBL and RDC should increase. However, although it is possible to increase the number of project leads (i.e., applying CBL), the number of leadership positions in a project is limited by natural factors. In particular, as project leaders have to coordinate themselves, with each new leader the need for communication between leaders increases quadratically. In computer-mediated collaboration (of which OSSDPs are one form), there must be a maximum of leadership positions to reduce coordination effort between project leaders. Thus, if many firms want to pursue CBL they compete with one another for (one out of few) leadership positions. Overall, although the number of leaders may rise with more firms active in an MVP, this raise is limited. We therefore surmise that:

H3a. In MVPs, the increase of control by leadership is not related to the number of firms participating.

Leaders in OSSDPs are typically elected by other committers based on their reputation and prior contributions to the project (Dahlander & O'Mahony, 2011). One way to gain reputation is to develop stable software code and to engage in community work. However, this activity is complemented by the fact that employed committers might be encouraged to vote for their firm-fellows. Thus, firms may follow the strategy of assigning committers to projects who promote one firm representative in elections. This strategy is congruent with what we named RDC, namely inserting company norms and beliefs into an OSSDP by practicing clan control outside the boundaries of the firm. Thus, there are two reasons why the number of firms should increase the number of firm-sponsored committers: First, because these committers have direct access to the source code and second because many firm-representatives increase the likelihood of electing a firm-sponsored project leader. In line with our extended control theory we posit that:

H3b. In MVPs, the greater the number of firms participating, the greater is the use of resource deployment control.

3. Method

3.1. Research setting

To find data applicable for identifying different governance approaches for SVPs and MVPs, we turned to the Eclipse foundation as a suitable case study. Eclipse is a hybrid of a repository for projects and a foundation. The Eclipse foundation has achieved success, as exemplified by the membership of more than 100 firms. Eclipse as both a software product and a foundation also has a long-lasting history. It originated as a development environment within IBM. Its major competitors included Microsoft's Visual Studio and Sun's NetBeans. To gain momentum, IBM open sourced its development, which meant sharing a $40 million investment with competitors (Wagstrom, 2009). However, other vendors could build products on top of Eclipse, rather than using proprietary software from competitors.

Characteristically for Eclipse, many governance mechanisms are publicly available, including the process for becoming a committer or member responsibilities. Its governance rules ignore the size of a firm. For example, each firm can only assign one representative to the strategic board and thus the firm has only one vote—regardless of its size. This holds also true for IBM as Eclipse's initiator. The foundation's website also provides comprehensive information about Eclipse projects, including the name of every committer, affiliation, project status, and contributions to a project. In general, any contributor is welcome, though potential committers must undergo a process to prove their programming qualifications. Voluntary contributors and participating firms also must agree to certain process rules and a project charter. Finally, every project is based on the principle of meritocracy: The more you contribute and the higher the quality of your contribution, the more you are allowed to do.

In this study, we are interested in committers and project leaders as both reflect RDC and CBL. Committers are developers who have earned the right to work at a project's source files based on prior contributions such as bug identification and fixing, newsgroup activity, and other community work. Formally, new committers are elected by existing project committers. Once elected, they obtain themselves the right to vote. Committers also elect respective project leaders. Within the Eclipse world, two types of leadership exist: A project management committee for top-level projects and project leads for the majority of projects. Top level projects consist of multiple projects but decisions on a daily basis are performed on project leads level, not by the management committee, which is why we focus on project leads in this study. Each Eclipse project is managed by at least one project lead but the number of project leads can be increased when the majority of committers decide to do so. According to the Eclipse website, project leads are "responsible for ensuring that their project's committers are following the Eclipse Development Process, and that the project is engaging in the right sorts of activities to develop vibrant communities of users, adopters, and contributors."[2]

3.2. Research approach and variables

To test our hypotheses, we took data from the Eclipse website using a set of Java programs that automatically query and retrieve data and a php frontend. With the php frontend, student assistants were able to add missing information manually. We obtained a data set pertaining to 109 different projects, including information about the number of committers, number of firms, number of project leaders, project age, and committers' affiliations. A committer's affiliation was identified by (1) his/her email address and (2) by his/her affiliation that were explicitly mentioned on Eclipse websites. Email-Postfix and description of affiliation matched in each case. We also could isolate information about each committer's role within a project (committers can be active in multiple projects). We note that information concerning non-committers is not comprehensively recorded at Eclipse. We classify voluntary committers, number of voluntary project leaders, number of sponsored committers and number of sponsored project leaders in each project by checking their affiliation. Each firm representative (Actuate, IBM, Oracle, SAP, etc.) was coded as a firm-sponsored

[2] See URL: https://eclipse.org/projects/dev_process/development_proces, last access December 2014.

committer while others such as hobbyists or those with University affiliation were coded as voluntary. We identified firms active in a project by checking whether they were listed as supporting a project or not.

Concerning the hypotheses, we had to identify whether a project was initiated by a firm or a community and then determine the number of firms per project to classify these projects according to our framework. We coded projects with more than one participating firm as MVPs and those with exactly one participating firm as SVPs. We did not find a single project without firm participation, which confirms our perception of Eclipse as a firm-driven foundation. To separate community- from firm-initiated projects, we looked at who submitted the first commit in each project's history. Although each contribution is logged, a recorded commit does not indicate the size of the actual contribution (Arafat & Riehle, 2009), so we cannot identify directly if a contributor changed a whole function or just a few lines. Despite this limitation, many commits should signal firm or community activity. Of our 109 projects, only 83 received a commit in our study timeframe though—this means that 18 projects were simply in a nascent stage and 8 projects did not have commits' records at all. In a few cases, the first commits came from both developers affiliated with a firm and voluntary developers. We therefore used the number of commits at time 1 as a distinguishing factor; if a project was founded in, for example, August 2004 and received 500 commits by firms and 20 by volunteers in the first month, we coded it as a firm-initiated project.

We interpreted the number of committers as an indicator of RDC and the number of project leaders as a means to pursue CBL. Similar to prior work on OSSDP, we include project age, project size, and project stage as control variables (Hahn et al., 2008; Stewart & Gosain, 2006). For project age, we used the natural log of the month since the first commit; project size was coded as the natural log of the number of commits a project received, to compensate for skewness (Hahn et al., 2008). Project stage is a continuous variable defined by the Eclipse foundation with three values ranging from 1 = incubation, over 2 = incubation (pending) to 3 = mature. Incubation involves equipping an already existing OSSDP with the community-defined Eclipse-centric open source process. In stage 2, the project waits for Eclipse's approval and in stage 3 the project is a full Eclipse project.

3.3. Research results

The data collection process revealed 912 identified committers (277 of which are not sponsored) working for 110 different firms in 109 different projects. About 35% of committers are affiliated with IBM, followed by Oracle (4.6%) and Actuate (4.4%). 18% of committers have no firm-affiliation. Firm-sponsored committers work, on average, in 1.87 projects while not-sponsored committers work in, on average, 1.39. After checking for projects with little or no activity in terms of commits, we narrowed the list to 83 projects, of which 8 were community-initiated SVP, 9 firm-initiated SVP, 24 community-initiated MVP and 42 firm-initiated MVP. Projects in this subsample have 12.67 firm-sponsored committers and 1.90 voluntary committers on average. Of these projects, 46 had the status, in the order of project development, “incubation”, 5 had the status “incubation (pending)”, and 32 had the status “mature”. Projects received 27,777 commits on average (SD = 48,427) of which 23,714 came from firm-sponsored committers and 5639 from the group of volunteers. A project's average age was 34.92 months (SD = 20.97).[3] Within MVPs, the minimum number of firms is two, the maximum 10. Mean number of firms for community initiated MVPs is 3.58, for firm-initiated MVPs is 3.95. The number of project leaders ranges from 1 to 7 (mean = 1.6).

We employed a multimethod data analysis approach; a single method could not capture all dimensions of our hypotheses. To test our hypotheses H1a and H1b, we first conducted a multivariate analysis of variance (MANOVA) to compare the mean values for each of the four categories for RDC and CBL. We ran a MANOVA instead of several independent ANOVAs to reduce the type-I error that emerges from multiple tests with the same data set (Field, 2005; Podsakoff, MacKenzie, Lee, & Podsakoff, 2003). A 2 × 2 MANOVA with RDC (number of voluntary committers versus number of firm-sponsored committers) and CBL (number of voluntary project leaders versus number of firm-sponsored project leaders) as dependent variables and

[3] The mean values for firm-sponsored commits and commits by volunteers do not sum up to 27,777. The reason is that only 64 projects received commits by volunteers. We report the mean value for these 64 projects here and do not take the 19 projects into account which did not receive commits by volunteers.

initiation (firm-initiated versus community-initiated) and vendor type (SVP versus MVP) as independent variables reveals significant main effects (initiation Wilks' Lamda = 0.866, $F_{(6, 73)} = 1.88$, $p < 0.1$, $\eta^2 = 0.134$; vendor type Wilks' Lamda = 0.845, $F_{(6, 73)} = 2.22$, $p < 0.05$, $\eta^2 = 0.155$). However, there is no significant effect of the multivariate interaction (initiation × vendor type Wilks' Lamda = 0.899, $F_{(6, 73)} = 1.36$, sig. = 0.245, $\eta^2 = 0.101$), as we show in Table 3.

Consistent with H1a and H1b, firm-initiated projects have significantly more project leaders and committers than their community-initiated counterparts. In addition, MVPs consist of more firm-sponsored project leaders and committers than do SVPs. However, the multivariate interaction between initiation and vendor type is not significant. That is, differences exist between whether or not a project was initiated by a firm regardless of the type of project (SVP vs MVP).

As the basis for H2, we considered the possibility of ranking different governance modes along a continuum from firm-initiated SVPs to community-initiated MVPs. We initially presumed that an ordered logistic regression would be an appropriate method for controlling for the influence of an ordered dependent variable; however, a Wald test (Brant, 1990) indicates the violation of some critical prerequisites. Therefore, we tested our hypotheses with a multinominal logit model, in line with previous research (Van de Vrande et al., 2009). Although in principle many of these results could be achieved with an ANOVA as well, multinominal logistic regression can include control variables. We estimated two models, one with control variables and the number of firm-sponsored and voluntary committers (Model 1) and one with the ratio of firm-sponsored to all committers (which can be interpreted as percentage) included, as suggested by H2 (Model 2). We provide the results in Table 4.

To present the results, we use community-initiated MVPs as the default category. In H2 we argued that the ratio of firm-sponsored to all committers increases with more integrated governance modes. From Table 4, we find partial support for our hypothesis. That is, in Model 1, we estimate governance modes without including the variables of interest, ratio of firm-sponsored to all committers. We see that compared to community-initiated MVPs, firm-initiated MVPs consist of more firm-sponsored ($b = 0.091$, $p < 0.05$) and fewer voluntary committers ($b = -0.282$, $p < 0.05$). However, we also see that the number of firm-sponsored committers is lower in community-initiated SVP ($b = -0.276$, $p < 0.1$), a fact that indicates that MVPs are larger than SVPs in terms of committers.

In Model 2, as in Model 1, firm-initiated MVPs differ significantly from the default category of community-initiated MVPs in terms of the number of voluntary (fewer) and firm-sponsored (more) committers. The other independent and the control variables show no significant effects. Regarding hypothesis H2, we find that firm-initiated MVPs have a significantly larger firm-sponsored to all committers ratio than community-initiated MVPs ($b = 0.077$, $p < 0.05$). However, no such effect is observable for SVP governance modes. Perhaps the significantly fewer number of cases of SVP (17), compared with MVP (66), in our data set produces this result. Furthermore, for Model 2, the values of the Nagelkerke and Cox/Snell R-square indicate an acceptable model fit (Field, 2005). In sum, we find partial support for H2.

Table 3
Interactions: means and standard deviations of vendor types and initiation.

			Vendor type			
			SVP		MVP	
	Variable	Initiated by	M	SD	M	SD
Resource deployment control	Number of voluntary committers	Firm	2.33 [a]	3.57	1.44 [b]	1.88
		Community	1.87 [b]	3.83	2.42 [c]	3.54
	Number of firm-sponsored committers	Firm	5.00 [a]	2.24	17.76 [b]	18.86
		Community	5.38 [a]	3.66	9.83 [c]	6.72
Control by leadership	Number of voluntary project leaders	Firm	.33 [a]	.50	.12 [b]	.51
		Community	.38 [a]	.74	.13 [b]	.34
	Number of firm-sponsored project leaders	Firm	.88 [a]	.64	1.54 [b]	1.21
		Community	.89 [a]	.60	1.61 [b]	1.14

Notes: For each consumer group, means sharing a common subscript within each row and column do not differ; means with different subscripts are different from one another ($p < .05$).

Table 4
Multinominal logistic regression.

	Model 1			Model 2		
	SVP firm	SVP com	MVP firm	SVP Firm	SVP com	MVP firm
Constant	6.733	−3.485	3.899	10.857*	.252	10.922**
	(4.255)	(5.885)	(3.097)	(6.122)	(7.215)	(4.908)
Project age	−1.271*	.040	−.806	−1.455*	−.092	−.935
	(.755)	(.931)	(.595)	(.799)	(.971)	(.626)
Project size	−.167	.437	−.096	−.144	.446	−.052
	(.361)	(.406)	(.232)	(.365)	(.412)	(.243)
Project stage	−.176	.183	−.218	−.279	.094	−.305
	(.640)	(.682)	(.406)	(.650)	(.688)	(.427)
Number of firm-sponsored committers	−.230	−.276*	.091**	−.209	−.252	.159**
	(.155)	(.148)	(.045)	(.192)	(.171)	(.063)
Number of voluntary committers	−.206	−.133	−.282**	−.465	−.374	−.818**
	(.172)	(.153)	(.134)	(−.393)	(.384)	(.324)
Ratio number of firm-sponsored committers to total number of committers				−.039	−.035	.077**
				(.049)	(.048)	(.039)
Chi^2	40.248			45.257		
−2 Log likelihood	152.574			147.564		
Cox/Snell Pseudo R^2	.388			.424		
Nagelkerke Pseudo R^2	.429			.469		

Notes: Community-initiated MVP is the comparison group. Unstandardized regression coefficients. Robust standard errors are in parentheses. Firm = firm-initiated; Com = community-initiated.
$N = 83$.
** $p < .05$.
* $p < .1$.

Finally, we used ordinary least square (OLS) regression to test H3a and H3b (see Table 5).[4] We regressed the number of firm-sponsored developers (Model 1) and the number of firm-sponsored project leaders (Model 2) on the controls, that is, project, age, size and stage, and the independent variable, number of firms. All of our independent variables are single-item measures, which is consistent with Bergkvist and Rossiter's (2007) notion that such measures are appropriate for constructs consisting of one object. As our hypotheses H3a and H3b pertained to MVPs, we use a subsample of N = 66 MVPs to test them. Our estimation shows that the number of firms in a project does not affect the number of firm-sponsored project leaders, as predicted in H3a. Model 2 in Table 5, which uses the number of firm-sponsored committers as the dependent variable, shows a significant effect of the number of firms on the use of RDC in support of H3b. The beta coefficient for the number of firms on number of firm-sponsored committers is 0.47 and significant ($p < 0.001$); when the number of firms changes by one standard deviation, the estimated outcome variable of firm-sponsored committers changes by 0.47 standard deviations, on average. Model 2 in Table 5 exhibits an explained variance of $R^2 = 0.28$, in robust support of H3b. We included another model for reasons of comparison (Model 3). Here, we regressed the number of sponsored project leaders (CBL) on the number of sponsored committers (RDC) including the control variables. We see that the use of RDC has no effect on CBL. Table 6 summarizes the results.

4. Discussion

Understanding how to influence and control the development of an OSSDP is vitally important to firms that provide resources to a project. Drawing on a behavioral view of control, we argue that firms may choose between RDC, which authorizes developers socialized within firm boundaries to work for an OSSDP, and CBL, where employed developers capture leading positions in a project. In addition, we developed a framework that distinguishes OSSDPs initiated by a firm from those initiated by a community, as well as those that consist

[4] Project leaders are also committers. We therefore ran our analysis with and without counting leaders as committers. As there were no statistical differences, we ran our final analysis with the option that committers are allowed to also be project leaders.

Table 5
OLS regression.

	Model 1 Number of firm-sponsored project leaders	Model 2 Number of firm-sponsored committers	Model 3 Number of firm-sponsored project leaders
Number of firms involved	.11 ($p = .395$)	.47 ($p = .000$)***	.04 ($p = .772$)
Project age	.02 ($p = .905$)	−.09 ($p = .538$)	.03 ($p = .846$)
Project size	.09 ($p = .498$)	.07 ($p = .566$)	.08 ($p = .545$)
Project stage	.18 ($p = .249$)	.24 ($p = .072$)	.14 ($p = .364$)
Number of firm-sponsored committers			.14 ($p = .349$)
R^2	.07	.28	.08
R^2 adjusted	.01	.24	.01
N	66	66	66

Note: Standardized estimates.
*** $p < .001$.

of one participating firm (SVP) from those with multiple firms (MVP). The difference between SVP and MVP is important and often neglected in studies of OSSDPs, yet these approaches reflect different business models (e.g., Dahlander, 2007; Riehle, 2012; West & O'Mahony, 2008). Among the most salient findings of this research is that the number of firms in a project affects the number of sponsored committers, but not the number of sponsored project leaders. We discuss the implications of our research for theory and management in the following.

4.1. Theoretical implications

In light of our results, important theoretical considerations might add to our understanding of the role of firms in OSSDPs. First, we extended previous conceptualizations of OSSDPs (e.g., Dahlander, 2007; Riehle, 2012) by introducing a 2 × 2 matrix considering the dimensions SVP vs MVP and project initiation. This distinction is reflected well in the data we analyzed. However, we note that the difference between SVP and MVP might be more prevalent in terms of governance modes and overall number of committers than the difference between firm and community initiation.

Second, in our investigation we drew upon transaction cost economics and argue that different governance modes (firm-initiated SVP, community-initiated SVP, firm-initiated MVP, and community-initiated MVP) represent a continuum that reflects the number of transactions. Among MVPs, community-initiated ones differ from firm-initiated ones. As predicted, the number of firm-sponsored committers is significantly higher in firm-initiated MVPs, but the number of voluntary committers is significantly lower. We find no such effect for the group of SVPs. We had a limited number of SVPs in our sample, and the size of a project also might influence these results. In addition, we find differences in terms of firm-sponsored to all committers ratio only if we compared firm-initiated with community-initiated MVP. From a theoretical point of view, especially SVPs maintain business models that are less dependent on external contributions than MVPs. Thus, the ratio of firm-sponsored to all committers should be higher in SVPs. However, SVPs do

Table 6
Results of hypotheses testing.

	Hypothesis	Supported
H1a	Firms make more use of control by leadership in firm-initiated projects than in community-initiated ones.	Yes
H1b	Firms make more use of resource deployment control in firm-initiated projects than in community-initiated ones.	Yes
H2	With more integrated governance modes, the ratio of firm-sponsored to all committers increases.	Partly
H3a	In MVPs, the increase of control by leadership is not related to the number of firms participating.	Yes
H3b	In MVPs, the greater the number of firms participating, the greater is the use of resource deployment control.	Yes

not differ from MVPs in terms of firm-sponsored to all committers ratio. In sum, these results call for future conceptual work that aligns transaction cost perspectives with different forms of OSSDPs (Demil & Lecocq, 2006).

Third, this study's results extend knowledge pertaining to OSSDP coordination. We found that the number of project leads is not fixed and varies from project to project. However, the relation between number of people to be coordinated (i.e., committers) and the number of people who coordinate (i.e., project leader) is not linear. While this might not come as a surprise, it has clear theoretical and managerial implications. As a project grows in terms of committers, more work has to be aligned, which makes more coordination necessary. In turn, these coordinators have to coordinate themselves, which increases complexity and results in an optimum number of coordinators in the long run (Hoegl, Weinkauf, & Gemuenden, 2004). Thus, in early stages of a project it might be possible to extend the level of project leaders (and therefore extend the level of CBL), while this becomes more difficult in later stages.

Fourth, our findings have implications concerning literature on division of labor in the context of OSS. For example, Dahlander and Magnusson (2008) argue that creative work tends to be performed in the community, whereas routine tasks take place within firm boundaries. This strategy seems beneficial in terms of saving costs. However, if firms assign a large number of employees to an OSSDP, this recommendation might be questioned. Then, the relative benefit of using the community as a complement resource will be dampened. Thus, our research provides valuable avenues for considering division of labor in future research on firm-dominated OSSDPs.

Finally, our findings might stimulate conceptual work on leadership and influence beyond the boundaries of the firm. While much research has been devoted to either leadership in traditional organizations or OSSDPs, the current discussion would benefit from considerations of how leadership is pursued if many firm representatives compete for these positions.

4.2. Managerial implications

Our study has also several implications for management. A first implication stems from the finding that the difference between SVP and MVP is more explicit than the distinction based on who initiated the project. Because SVPs represent proprietary software vendors and their use of a dual licensing approach (Watson, Boudreau, York, Greiner, & Wynn, 2008), whereas MVPs are a means to collaborate efficiently in a consortium, the choice of a business model clearly determines the choice of control modes.

Second, firms that plan to engage in existing OSSDPs or that are willing to initiate their own projects should define their business model before determining which governance modes to use. Our results suggest that in cases when only few leadership positions are available, de novo entrants in existing projects prefer seeking RDC over CBL. In particular, when any new firm enters an OSSDP, the number of stakeholders grows. In theory, the de novo entrant deploys equal resources to the OSSDP, so that the entrant is able to influence a project's trajectory. We have argued that in those cases, the number of employed committers increases the ability to perform RDC. With our data, we cannot determine if this form of control is superior to CBL in terms of outcomes. Therefore, we are cautious about implying that RDC is a “superior” form of control in MVPs.

Instead, we speculate about two effects that lead to greater RDC. First, when firms enter a project late, it is very difficult for them to capture leadership positions, because leadership often derives from advance contributions to a project (Dahlander & O'Mahony, 2011; Scozzi et al., 2008). Therefore, de novo entrants either must wait until their own developers reach leadership positions or hire current project leaders. The latter tactic would not increase the number of project leaders. Second, more project leaders may simply lead to greater coordination costs. If increased coordination costs outweigh the benefits of the division of labor (Aiken & Hage, 1968), CBL would not be applicable, even if it seems appealing from a single firm's perspective. Rather, they have no choice other than to assign their developers to the OSSDP to exercise control, which we define as RDC. In sum we can argue that instead of expending resources or granting single developers the time to climb the meritocracy ladder within the community (Henkel, 2009), firms should assign many of their own developers to the OSSDP to encourage the transfer of their own norms into the project. These developers may act like a clan within the OSSDP, moving their (and the firms) preferred topics to the top of the agenda, thus, pursuing RDC.

4.3. Limitations and further research

As does almost any research, our study involves several limitations. First, regarding the data set, Eclipse is a firm-driven environment, thus, the fact that no project without firm involvement exists in the sample. Although this is intended, studies of projects from a random sample might reveal different results. Second, our data set does not allow us to distinguish the different firms in MVPs (with their different business models), so we only considered the aggregation of employed individuals, regardless of which firm pay their salary. As different business models (e.g., user firm, software vendor, infrastructure provider) are likely to affect the amount of RDC and CBL, further research should take into account the influence of different firms in OSSDPs. In addition, approximately 50% of the sponsored developers are affiliated with IBM. This is a consequence of Eclipse's history and does not affect our results regarding the difference between firm-initiated and community-initiated projects. Future research, however, could investigate the coexistence of dominant and less dominant firms (in terms of their relative developer deployment) in firm-driven OSSDPs. Third, we did not control for differences in developers' affiliations. Future research therefore could investigate how representatives of universities behave in OSSDPs. Fourth, from a theoretical point of view, control includes both setting directions ex ante and ex post monitoring in a recursive way. Employees receive advice, to which they respond by completing their task, which a supervisor then evaluates. We could not differentiate between ex ante and ex post control mechanisms, so we call for research that considers when each control mode is used. Fifth, with the data at hand it was not possible to control for a project's importance. However, competition in receiving leadership positions might be dependent on a project's importance in terms of expected commercial value or relevance for firms' strategies.

Finally, we did not investigate any effect of the use of the control modes developed for this study. There is a rich body of literature pertaining to OSS success (e.g., Ghapanchi & Aurum, 2012), which provides avenues for combining our control-related findings with a performance-enhancing perspective.

References

Abernethy, M.A., Bouwens, J., & van Lent, L. (2010). Leadership and control system design. *Management Acounting Review*, *21*, 2–16.

Aghion, P., & Tirole, J. (1997). Formal and real authority in organizations. *Journal of Political Economy*, *105*(1), 1–29.

Aiken, M., & Hage, J. (1968). Organizational interdependence and intra-organizational structure. *American Sociological Review*, *33*(6), 631–652.

Aksulu, A., & Wade, M.R. (2010). A comprehensive review and synthesis of open source research. *Journal of the Association for Information Systems*, *11*(11), 576–656.

Almirall, E., & Casadesus-Masanell, R. (2010). Open versus closed innovation: A model of discovery and divergence. *Academy of Management Review*, *35*(1), 27–47.

Andersen-Gott, M., Ghinea, G., & Bygstad, B. (2012). Why do commercial companies contribute to open source software? *International Journal of Information Management*, *32*(2), 106–117.

Anderson, E., & Oliver, R. (1987). Perspectives on behavior-based versus output-based salesforce control systems. *Journal of Marketing*, *51*(4), 76–88.

Arafat, O., & Riehle, D. (2009). The commit size distribution of open source software. *Proceedings of the 42nd Hawaiian International Conference on System Science, Hawaii, U.S.*

Arya, A., Glover, J., & Sivaramakrishnan, K. (1997). The interaction between decision control problems and the value of information. *Accounting Review*, *72*(4), 561–574.

Augustin, L. (2008). A new breed of P&L: The open source business financial model. *Conference presentation: Open source meets business, January 23, Nürnberg, Germany*.

Barker, J.R. (1993). Tightening the iron cage: Concertive control in self-managing teams. *Administrative Science Quarterly*, *38*(3), 321–341.

Bergkvist, L., & Rossiter, J.R. (2007). The predictive validity of multiple-item versus single-item measures of the same constructs. *Journal of Marketing Research*, *44*, 175–184.

Blau, P.M., & Scott, W.R. (1962). *Formal organizations*. San Francisco: Chandler Publishing.

Bonaccorsi, A., Giannangeli, S., & Rossi, C. (2006). Entry strategies under competing standards: Hybrid business models in the open source software industry. *Management Science*, *52*(7), 1085–1098.

Bonaccorsi, A., & Rossi, C. (2006). Comparing motivations of individual programmers and firms to take part in the open source movement: From community to business, knowledge. *Technology and Policy*, *18*(4), 40–64.

Brant, R. (1990). Assessing proportionality in the proportional odds model for ordinal logistic regression. *Biometrics*, *46*(4), 1171–1178.

Brockner, J., Spreitzer, G., Mishra, A., Hochwarter, W., Pepper, L., & Weinberg, J. (2004). Perceived control as an antidote to the negative effects of layoffs on survivor's organizational commitment and job performance. *Administrative Science Quarterly*, *49*(1), 76–100.

Cardinal, L. (2001). Technological innovation in the pharmaceutical industry: The use of organizational control in managing research and development. *Organization Science*, *12*(1), 19–36.

Cardinal, L., Sitkin, S., & Long, C. (2004). Balancing and rebalancing in the creation and evolution in organizational control. *Organization Science*, *15*(4), 411–431.

Choudhury, V., & Sabherwal, R. (2003). Portfolios of control in outsourced software development projects. *Information Systems Research*, *14*(3), 291–314.

Conger, J.A., & Kanungo, R.N. (1987). Toward a theory of charismatic leadership in organizational settings. *Academy of Management Review*, *12*(4), 637–647.
Crowston, K., & Howison, J. (2006). Hierarchy and centralization in free and open source software team communications. *Knowledge, Technology & Policy*, *18*(4), 65–85.
Dahlander, L. (2005). Appropriation and appropriability in open source software. *International Journal of Innovation Management*, *9*(3), 259–285.
Dahlander, L. (2007). Penguin in a new suit: A tale of how de novo entrants emerged to harness free and open source software communities. *Industrial and Corporate Change*, *16*(5), 913–943.
Dahlander, L., & Frederiksen, L. (2012). The core and cosmopolitans: A relational view of innovation in user communities. *Organization Science*, *23*(4), 988–1007.
Dahlander, L., & Magnusson, M.G. (2008). How do make firms make use of open source communities? *Long Range Planning*, *41*(6), 629–649.
Dahlander, L., & O'Mahony, S. (2011). Progressing to the center: Coordinating project work. *Organization Science*, *22*(4), 961–979.
Dahlander, L., & Wallin, M. (2006). A man on the inside: Unlocking communities as complementary assets. *Research Policy*, *35*(8), 1243–1259.
De Laat, P.B. (2007). Governance of open source software: State of the art. *Journal of Management and Governance*, *11*(2), 165–177.
Demil, B., & Lecocq, X. (2006). Neither market nor hierarchy nor network: The emergence of bazaar governance. *Organization studies*, *27*(10), 1447–1466.
Druskat, V.U., & Wheeler, J.V. (2003). Managing from the boundary: The effective leadership of self-managing work teams. *Academy of Management Journal*, *46*(4), 435–457.
Eisenhardt, K. (1985). Control: Organizational and economics approaches. *Management Science*, *31*(2), 134–149.
Enkel, E., Gassmann, O., & Chesbrough, H. (2009). Open R&D and open innovation: Exploring the phenomenon. *R&D Management*, *39*(4), 311–316.
Field, A. (2005). *Discovering statistics using SPSS* (2nd ed.). London, UK: Sage Publication.
Fitzgerald, B. (2006). The transformation of open source software. *MIS Quarterly*, *30*(3), 587–598.
Füller, J., Matzler, K., & Hoppe, M. (2008). Brand community members as a source of innovation. *Journal of Product Innovation Management*, *25*(6), 608–619.
Gallivan, M.J. (2001). Striking a balance between trust and control in a virtual organization: A content analysis of open source software case studies. *Information Systems Journal*, *11*(4), 277–304.
George, E., & Chattopadhyay, P. (2005). One foot in each camp: The dual identification of contract workers. *Administrative Science Quarterly*, *50*, 68–99.
Ghapanchi, A.H. (2013). Rallying competencies in virtual communities: A study of core processes and user interest in open source software projects. *Information and Organization*, *23*(2), 129–148.
Ghapanchi, A.H., & Aurum, A. (2012). Competency rallying in electronic markets: Implications for open source project success. *Electronic Markets*, *22*(2), 117–127.
Ghapanchi, A.H., Wohlin, C., & Aurum, A. (2014). Resources contributing to gaining competitive advantage for open source software projects: An application of resource-based theory. *International Journal of Project Management*, *32*(1), 139–152.
Giuri, P., Rullani, F., & Torrisi, S. (2008). Explaining leadership in open source software projects. *Information Economics and Policy*, *20*(4), 305–315.
Grand, S., von Krogh, G., Leonard, D., & Swap, W. (2004). Resource allocation beyond firm boundaries: A multi-level model for open source innovation. *Long Range Planning*, *37*(6), 591–610.
Hahn, J., Moon, J.Y., & Zhang, C. (2008). Emergence of new project teams from open source software developer networks: Impact of prior collaboration ties. *Information Systems Research*, *19*(3), 269–391.
Henkel, J. (2009). Champions of revealing — The role of open source developers in commercial firms. *Industrial and Corporate Change*, *18*(3), 435–471.
Hoegl, M., Weinkauf, K., & Gemuenden, H.G. (2004). Interteam coordination, project commitment, and teamwork in multiteam R&D projects: A longitudinal study. *Organization Science*, *15*(1), 38–55.
Jarvenpaa, S.L., & Lang, K.R. (2011). Boundary management in online communities: Case studies of the nine inch nails and ccmixter music remix sites. *Long Range Planning*, *44*(5–6), 440–457.
Jaworski, B. (1988). Toward a theory of marketing control: Environmental context, control types, and consequences. *Journal of Marketing*, *52*(3), 23–39.
Kirsch, L.J. (1997). Portfolios of control modes and IS project management. *Information Systems Research*, *8*(3), 215–239.
Kirsch, L., Ko, D. -G., & Haney, M. (2010). Investing the antecedents of team-based clan control: Adding social capital as a predictor. *Organization Science*, *21*(2), 469–489.
Lakhani, K., & Wolf, R.G. (2005). Why hackers do what they do: Understanding motivation and effort in free/open source software projects. In J. Feller, B. Fitzgerald, S.A. Hissam, & K.R. Lakhani (Eds.), *Perspectives on free and open source software* (pp. 3–22). Cambridge, MA: MIT Press.
Lattemann, C., & Stieglitz, S. (2005). Framework for governance in open source communities. *Proceedings of the 38th Hawaii International Conference on System Sciences, Hawaii, U.S.*
Lerner, J., & Schankerman, M. (2010). *The Comingled Code. Open source and economic development.* Cambridge, MA: MIT Press.
Markus, M. (2007). The governance of free/open source software projects: Monolithic, multidimensional, or configurational? *Journal of Management and Governance*, *11*(2), 151–163.
McMahon, J.T., & Ivancevich, J.M. (1976). A study of control in manufacturing organization: Managers and nonmanagers. *Administrative Science Quarterly*, *21*(1), 66–83.
Mockus, A., Fielding, R., & Herbsleb, J.D. (2002). Two case studies of open source software development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology*, *11*(3), 309–346.
O'Mahony, S. (2007). The governance of open source initiatives: What does it mean to be community managed? *Journal of Management and Governance*, *11*(2), 139–150.
O'Mahony, S., & Bechky, B. (2008). Boundary organizations: Enabling collaboration between unexpected allies. *Administrative Science Quarterly*, *53*(3), 422–459.

O'Mahony, S., & Ferraro, F. (2007). The emergence of governance in an open source community. *Academy of Management Journal*, *50*(5), 1079–1106.
Olson, M. (2005). Dual licensing. In D. Cooper, C. DiBona, & M. Stone (Eds.), *Open sources 2.0: The continuing evolution*. Sebastopol, CA: O'Reilly.
Ouchi, W.G. (1977). The relationship between organizational structure and organizational control. *Administrative Science Quarterly*, *22*(1), 95–113.
Ouchi, W.G. (1979). A conceptual framework for the design of organizational control mechanisms. *Management Science*, *25*(9), 833–848.
Ouchi, W.G., & Maguire, M.A. (1975). Organizational control: Two functions. *Administrative Science Quarterly*, *20*(4), 559–569.
Podsakoff, P., MacKenzie, S., Lee, J. -Y., & Podsakoff, N. (2003). Common method biases in behavioral research: A critical review of the literature and recommended remedies. *Journal of Applied Psychology*, *88*(5), 879–903.
Riehle, D. (2012). The single vendor commercial open source business model. *Information Systems and e-Business Management*, *10*(1), 5–17.
Roberts, J.A., Hann, I. -H., & Slaughter, S.A. (2006). Understanding the motives, participation, and performance of open source software developers: A longitudinal study of the Apache projects. *Management Science*, *52*(7), 984–999.
Ruhanen, L., Scott, N., Ritchie, B., & Tkaczynski, A. (2010). Governance: A review and synthesis of the literature. *Tourism Review*, *65*(4), 4–16.
Rullani, F., & Haeflinger, S. (2013). The periphery on stage: The intra-organizational dynamics in online communities of creation. *Research Policy*, *42*(4), 941–953.
Rustagi, S., King, W., & Kirsch, L. (2008). Predictors of formal control usage in IT outsourcing partnerships. *Information Systems Research*, *19*(2), 126–143.
Santos, C., Kuk, G., Kon, F., & Pearson, J. (2013). The attraction of contributors in free and open source software projects. *Journal of Strategic Information Systems*, *22*(1), 26–45.
Schaarschmidt, M., & Von Kortzfleisch, H. (2014). Examining investment strategies of venture capitalists in open source software. *International Journal of Innovation and Technology Management*, *11*(4) (19 pages).
Schaarschmidt, M., Walsh, G., MacCormack, A., & Von Kortzfleisch, H. (2013). 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, December, Milan, Italy*.
Scozzi, B., Crowston, K., Eseryel, Y., & Li, Q. (2008). Shared mental models among open source software developers. *Proceedings of the 41st Hawaii International Conference on System Science, Hawaii, U.S.*
Shah, S. (2006). Motivation, governance, and the viability of hybrid forms in open source development. *Management Science*, *52*(7), 1000–1014.
Simon, H.A. (1973). Applying information technology to organization design. *Public Administration Review*, *33*(3), 268–278.
Stam, W. (2009). When does community participation enhance performance of open source companies? *Research Policy*, *38*(3), 1288–1299.
Statista (2015). Projected revenue of open source software from 2008 to 2020. URL: http://www.statista.com/statistics/270805/projected-revenue-of-open-source-software-since-2008/
Stevenson, H.H., & Jarillo, J.C. (1990). A paradigm of entrepreneurship: Entrepreneurial management. *Strategic Management Journal*, *11*(special issue), 17–27.
Stewart, K.J., Ammeter, A.P., & Maruping, L.M. (2006). Impacts of license choice and organizational sponsorship on user interest and development activity in open source software projects. *Information Systems Research*, *17*(2), 126–144.
Stewart, K.J., & Gosain, S. (2006). The impact of ideology on effectiveness in open source software teams. *MIS Quarterly*, *30*(2), 291–314.
Teigland, R., Di Gangi, P., Flaten, B. -T., Giovacchini, E., & Pastorino, N. (2014). Balancing the Tightrope: Managing the boundaries of a firm-sponsored OSS community and its impact on innovation and absorptive capacity. *Information and Organization*, *24*(1), 25–47.
Van de Vrande, V., Vanhaverbeke, W., & Duysters, G. (2009). External technology sourcing: The effect of uncertainty on governance mode choice. *Journal of Business Venturing*, *24*(1), 62–80.
Villalonga, B., & McGahan, A.M. (2005). The choice among acquisitions, alliances and divestures. *Strategic Management Journal*, *26*(13), 1183–1208.
Von Hippel, E., & Von Krogh, G. (2003). Open source software and the private-collective innovation model: Issues for organization science. *Organization Science*, *14*(2), 209–225.
Wagstrom, P.A. (2009). *Vertical interaction in open source software engineering communities.* Pittsburgh, PA: Carnegie Mellon University (PhD Thesis).
Watson, R.T., Boudreau, M. -C., York, P.T., Greiner, M., & Wynn, D. (2008). The business of open source. *Communications of the ACM*, *51*(4), 41–46.
West, J., & Gallagher, S. (2006). Challenges of open innovation: The paradox of firm investment in open source software. *R&D Management*, *36*(3), 319–331.
West, J., & O'Mahony, S. (2008). The role of participation architecture in growing sponsored open source communities. *Industry and Innovation*, *15*(2), 145–168.
Westergren, U.H., & Holmström, J. (2012). Exploring preconditions for open innovation: Value networks in industrial firms. *Information and Organization*, *22*(4), 209–226.
Williamson, O.E. (1985). *The economic institutions of capitalism: Films, markets, relational contracting.* New York: Free Press.
Xu, B., Jones, D., & Shao, B. (2009). Volunteers' involvement in online community based software development. *Information & Management*, *46*(3), 151–158.