Affiliated Participation in Open Source Communities

Adam Alami
Computer Science
IT University of Copenhagen

Andrzej Wąsowski
Computer Science
IT University of Copenhagen

Abstract*—*Background: **The adoption of Free/Libre and Open Source Software (FOSS) by institutions is significantly increasing, and so is the affiliated participation (the participation of industry engineers in open source communities as part of their jobs).** ***Aims:*** **This study is an investigation into affiliated participation in FOSS communities. So far, little is known about the affiliated participation and the forces that influence it, even though the FOSS innovation model is increasingly becoming a serious contender for the private investment model in many sectors.** ***Method:*** **We present a qualitative inquiry into affiliated participation in the Robot Operating System (ROS) and Linux Kernel communities, using twenty-one in-depth interviews and participatory observation data from twenty-nine community events.** ***Results:*** **Our results show that affiliated participation in these communities is constrained by several barriers: objections of senior management, protection of the company's image, protection of intellectual property, undefined processes and policies, the high cost of participation, and unfamiliarity with the FOSS system.** ***Conclusions:*** **These barriers should be addressed in any organization considering using FOSS as a significant acquisition, distribution, and development strategy.**

***Index Terms*—FOSS, Open Source Software Adoption, Open Source Software Participation, Affiliated Participation**

I. Introduction

Free/Libre Open Source Software (FOSS) was born as an informal and niché movement. By mid 90's, it became a recognized software development and distribution model. The emergence of the Internet removed the physical barriers in collaboration and accelerated FOSS growth. The traditional collocated development process ceased to be the only option—the collaboration of highly recognized experts facilitates faster progress and innovation [1]. In 2008, Hauge [2] reported that already half of the surveyed companies used open source components. The FOSS process is considered a phenomenon of the *collective action* innovation model. The increasing interest in adopting FOSS could mean that FOSS becomes a contender for the *private investment* innovation model. If this assumption is valid then it is timely to investigate commercial adoption of, and the participation in, the FOSS movement.

The value of the FOSS market is estimated by some to exceed 1.9 billion [3]. We can safely assume that FOSS code is used in many of our everyday technical gadgets, software and tools. Companies and government institutions not only use FOSS, but increasingly choose to open source the code of their products. In 2016, Walmart open sourced a version of its cloud management system. In 2011, ExxonMobil released an open source "Standards DevKit" (a developer toolkit). They wanted to foster collaboration amongst oil and gas companies. In 2016, several financial companies (Morgan, Wells Fargo, and the London Stock Exchange) launched "Hyperledger"—an open source project aiming to build blockchain-based capability to track the exchange of financial assets, including stocks, and bonds. The names of the companies involved mark an interesting shift in the attitude towards open source. This trend is observed even in government policies. Recently, in 2016, the US government released a federal code source policy. It institutes a pilot program requiring that government agencies release 20% of new custom-developed code as open source.[1]

The commercial interest mixed with the collective action work model and communitarian ideology raises gripping research questions. How does a company adopt community maintained source code? How is the engagement with the community shaped? What are the forms of participation? What makes the engagement successful and well functioning? We set out to study how individuals working for commercial companies participate in FOSS communities, the so called *affiliated participation* phenomenon:

RQ1: *What are the participation models used by companies and institutions to engage with the community?*
RQ2: *What are the barriers for employees of companies to actively contribute to FOSS as part of their main job?*

We investigate these questions using qualitative research methods, collecting data during semi-structured interviews with 21 participants and through participatory observation in 29 events and meetings. We work with two large FOSS communities: ROS and The Linux Kernel. The Robot Operating System (ROS) is a framework that is widely used in robotics. The Linux kernel is an open-source Unix-like computer operating system kernel. Both communities enjoy lively participation from many commercial actors, both contributing and benefiting from the development. We find that affiliated participation is constrained by a few barriers: senior management objection, company image, intellectual property protection concerns, undefined processes and policies, the high cost of participation, and unfamiliarity with the system. When these barriers are unmanaged and the company has a business model and strategies (i.e. product, branding) misaligned to the community processes and system of values, a passive behavior toward contributing is observed.

Ideally, when a company starts to use open source code, it should envision a community participation process and complement it with a set of participation policies. In addition, the com-

[1] https://sourcecode.cio.gov, seen September 2018

pany software acquisition strategy should reflect that decision. Otherwise, the participation becomes passive, and the company becomes a consumer of the community produced goods without contributing back to the community. A passive participation strains the community's sustainability, it leads the community into regression which hinders growth and ability to innovate.

The paper proceeds by discussing the prior research in Sect. II. Section III presents the studied FOSS communities and argues why they are a suitable choice for understanding affiliated participation. In Sect. IV, we define our mixed research method and discuss the rationale behind it. Section V presents the key findings, and Sect. VII interprets the findings as possible actions for companies interested in improving their participation in FOSS. We conclude in Sect. IX.

II. Related Work

FOSS adoption by commercial entities: Several works show the extent of the FOSS adoption in industry, the demography of the participating companies, and the participation behavior [4]. Already in 2006, companies contributed to 97 out of the 300 most active SourceForge projects [5]. Yet the market of FOSS is difficult to size [2]. The existing attempts [6]–[10] focus on few projects like the LAMP stack itself, Linux, or end-user applications (mail or office tools). Studies from Finland, UK, Australia, and US report low FOSS adoption in the public sector—Linux, used by more than 50% of respondents, is a notable exception [11], [12]. Together with the other elements of the LAMP stack, Linux was also frequently used in other sectors 15 years ago [9]. However, the adoption varies widely across countries, sectors, and company sizes, from as low as 17.7% in Sweden and as high as 43.7% in Germany [9]. A survey on Australia's top companies reports that 26% used a varied spectrum of FOSS products in 2005 [7]. With the exception of Linux, Apache HTTP Server, and perhaps a few others, most surveys report that less than 30% respondents adopted FOSS. Less is known about the extent of the internal adoption of FOSS in these companies and the participation behavior in the community. Some of our subjects claim that a pure passive adoption is a sub-optimal form of participation, where not all benefits are realized.

FOSS-related business models: Two main business models for FOSS are (i) the *support seller*, where a company sells services associated with a FOSS project, and (ii) the *loss-leader*, where a company uses FOSS to grow the user base of an industrial software product by promoting it towards a FOSS community, typically using a free license for a variant [13]. Fitzgerald [14] identified four adoption models: *value-adding service enabler* (similar to support seller), *market creation*, *leveraging community development*, and *leveraging the FOSS brand*. How do companies actually implement these business models varies, with significant heterogeneity, especially regarding the degree of openness to FOSS [15]. This heterogeneity is reflected degree of adoption, re-use, and integration [14], [16], [17]. None of these works investigates and explains how the companies actually engage with the community; what makes their business models successful, and how the engagement with the community fuels or mitigates the risks of adopting FOSS.

Affiliated participation and community relations: The detailed qualitative aspects of commercial participation in FOSS projects has attracted relatively little attention of researchers. In a systematic literature review on commercial use of open source software [18], Höst and Oručević-Alagić list only minimal work regarding the ways companies get engaged in FOSS communities: contributions happen either through individual developers [13], [19], or by a substantial commitment.

Henkel [20] observes that for-profit organizations protect their contributions to the community selectively. They perceive active participation as overly open, unsuitable for a company. Many affiliated participants contribute out of personal interest, rather than as representatives of a company. Yet, half of the supervisors are aware that their engineers share code. Furthermore, only 22.8% of respondents describe their firm's policy towards contributing actively as encouraging participation in FOSS, and 16.8% think that it is restrictive. Even though, this study sheds some light on affiliated participation, it does not attempt to understand the participation behavior. Our research objective is to analyze this issue in depth.

Lundell et al. [21], as well as Dahlander and Magnusson [22] identified three types of relationship between companies and FOSS communities: parasitic (in which the commercial interest is indifferent to its effect on FOSS), symbiotic (mutually beneficial relationships, in which both the firm and the community gain advantage), and commensalistic (relationships between the two entities where one party, the firm, benefits from the other without affecting negatively the FOSS community). Lundell et al. [21] suggest that most relationships are symbiotic. Our study shows that the relationships differ a lot between the two studied communities—not all communities have been able to successfully develop a vibrant symbiotic environment.

Open innovation in software engineering: The FOSS movement has enabled a new kind of innovation in software intensive product development, the *open innovation*—a distributed innovation process based on managed knowledge flows across organizational boundaries. Under open innovation firms use both external and internal ideas with internal and external paths to market when working to advance their technology [23]. Activities are inbound or outbound and classified as *pecuniary* (related to competitive assets and producing rewards) vs. *non-pecuniary* (related to non-competitive assets without immediate rewards). Inbound activities use input from outside the organization and outbound activities exploit internally developed innovations [24].

According to Munir et al. [25] innovation occurs as an exchange of information about new technology, and it is one of the main drivers for collective inventions. Both cooperation and competition exist in open innovation, and this results in value creation, expanding benefits from the process, and value appropriation, as benefits are seized from the process. Value creation expands the market, and value appropriation determines the firm's share of the market [26].

However, FOSS is more than exchanging ideas or information. It carries a strong personal aspect (collaboration of individuals, as opposed to collaboration of companies) and

TABLE I
A CENSUS OF THE PARTICIPATING COMPANIES IN THE STUDY.

Company	Company sector	No. of employees	Community	Age[Y]	Model	Revenue or Budget
1	Industrial robotics start-up	12 (FY 2018)	ROS	10	passive	€0.3 million (FY 2017)
2	Industrial robotics	102 (FY 2018)	ROS	16	passive	$1.2 million (FY 2017)
3	OSRF Foundation	29 (FY 2018)	ROS	10	active	-
4	Academic institution	5,189 (FY 2017)	ROS	177	active	€21.3 million (FY 2017)
5	Industrial robotics vendor	36 (FY 2018)	ROS	7	passive	€1.6 million (FY 2017)
6	Industrial research institute	2,602 (FY 2018)	ROS	10	latent	$583 million (FY 2018)
7	Industrial research institute	25,000 (FY 2018)	ROS	72	latent	€2.3 billion (FY 2017)
8	Industrial research institute	2,761 (FY 2018)	ROS	10	passive	$322.3 million (FY 2017)
9	Industrial robotics vendor	50 (FY 2018)	Linux Kernel	26	passive	$2.9 million (FY 2018)
10	Linux distributor	12,600 (FY 2018)	Linux Kernel	26	active	$2.9 billion (FY 2017)
11	Telecommunication	4,796 (FY 2018)	Linux Kernel	37	passive	$23.5 million (FY 2017)
12	Telecommunication	4,796 (FY 2018)	Linux Kernel	37	passive	$1.87 billion (FY 2017)
13	Software vendor	21 (FY 2018)	Linux Kernel	24	latent	$1.25 billion (FY 2017)
14	Linux distributor	1,467 (FY 2018)	Linux Kernel	27	active	$365.5 million (FY 2017)

TABLE II
INTERVIEW SUBJECTS BY COMMUNITY, COMPANIES, JOB DESCRIPTION, AND WORK EXPERIENCE.

Participant #	Company	Subject role	Country	Exp. [Y]	Nature of involvement with FOSS
ROS community members					
Participant 1	Company 1	co-founder	France	5	An organization using the core of ROS to control tailored robotics systems combining various robot components.
Participant 2	Company 2	director	USA	12	A company using ROS components to build military robotics systems
Participant 3	Company 3	core developer	USA	10	The steward of the ROS community.
Participant 4	Company 4	technical lead	Netherlands	12	A large university offering consulting and research on industrial robotics.
Participant 5	Company 5	developer	Germany	6	An organization specialized in 3D sensors that enable perception and localization for robots. Uses ROS components to develop products.
Participant 6	Company 6	developer	Singapore	8	An organization leveraging ROS components to build robotics systems for customers.
Participant 7	Company 6	developer	Singapore	10	
Participant 8	Company 7	developer	Spain	10	An organization leveraging ROS components to build robotics systems for customers.
Participant 9	Company 5	developer	Germany	10	
Participant 10	Company 8	technical lead	South Korea	13	An organization leveraging ROS components to develop robotics systems for customers.
The Linux Kernel community members					
Participant 11	Company 10	kernel engineer	Denmark	18	A Linux distributor that provides consulting and support services
Participant 12	Company 10	kernel hacker	Denmark	10	
Participant 13	Company 10	principal engineer	Brazil	23	
Participant 14	Company 10	kernel engineer	USA	10	
Participant 15	Company 10	kernel engineer	USA	12	
Participant 16	Company 11	embedded Linux engineer	Spain	5	A company packaging Linux with in-house telecommunications & hardware products
Participant 17	Company 11	embedded Linux engineer	USA	7	
Participant 18	Company 12	kernel engineer	USA	30	A developer of complex software for oil and gas industry. Bundles Linux with its products.
Participant 19	Company 13	kernel engineer	USA	10	A developer of software for the telecommunication industry.
Participant 20	Company 13	kernel engineer	USA	8	
Participant 21	Company 14	project manager	USA	30	A Linux distributor that provides consulting and support services.

associates many risk and prejudices (for instance regarding IP protection). Our study sheds light on the internal work in this processes, and the obstacles contributors and companies face in daily engagement; many of which cannot be explained in terms of market interplay or innovation.

III. SUBJECT COMMUNITIES

We have chosen to work with two communities (ROS and Linux Kernel) that enjoy a strong industrial participation and a significant adoption in the respective industries. Both communities are accustomed to commercial participation and

committed to building relations with commercial members.

The Robot Operating System (ROS) is a robotics middle-ware supporting a wide a variety of platforms that it slowly becomes a *de facto* standard in robotics. The project develops tools, libraries, component drivers, conventions, standard communication and coordination features, and implementations of essential robotics-specific functionality, for example localization or planning. A ROS-based application is composed of several ROS components complemented with application specific code.

ROS originated at the Stanford Artificial Intelligence Laboratory (SAIL). In 2007, the code was transferred to a start-up, Willow Garage, and released under an open source license. Since 2013, the Open Source Robotics Foundation (OSRF) stewards the work of the ROS community.

ROS Industrial is a branch of ROS, and a corresponding association, with focus on industrial applications. Since 2012, ROS Industrial has secured the collaboration of key players in the industry (e.g. ABB, Yaskawa, Siemens, John Deere, BMW, Bosch, etc.). For this reason, ROS is a relevant and interesting community to study the interplay and the mix of proprietary, closed source, open source, and free software development.

The Linux Kernel project develops an open-source Unix-like operating system kernel that is used across extremely many hardware platforms. Since its creation by Linus Torvalds in 1991, the project has successfully developed a sustainable community. According to the Linux Foundation, which today is the main body, stewarding the development of the kernel, the project attracted nearly 12,000 developers from more than 1,200 companies, who contributed code since tracking began in 2005. The adoption of the kernel by Android is a testimony for its commercial viability, sustainability and investment value in long term. This commercial success and the rich social environment of the kernel community leans it well to study of commercial involvement in the FOSS movement.

IV. Methods

Participation of affiliated members in open source projects is a multifaceted complex process. We approach it with interpretative deductive reasoning characteristic of qualitative methods, collecting data using in-depth interviews and participatory observations.

A. Interviews

Semi-structured interviews provide us with a reasoned interpretation of the participation process by the subjects.

Subject selection: We interviewed 21 members of the ROS and Linux communities. ROS subjects were recruited at community events in 2017 and 2018 (ROSCon, ROS-Industrial Conference, Danish ROS MeetUp). Linux subjects were approached via LinkedIn. We searched for contributors on LinkedIn, using community name and terms 'contributor'/'developer'. We contacted random entries from the search results. We asked the first four participants of the Linux community to facilitate further contacts (snowball sampling). We stopped gathering data when we reached saturation. Table I is a census of the participating companies. Linux and ROS are fundamentally important for all the involved companies. The open source code is part of their main products and services. Table II summarizes the demographics of the interviewed population. With the exception of one female (Linux) all other subjects were male. The interviewer, who also selected the subjects, had no prior relationships to the participants.

Design: Prior to conducting the interviews, we compiled an interview guide with main questions and a set of possible probing questions. Table III summarizes this structure. We commenced every interview with introduction questions, before diving into the core questions of the interview. Probing questions were evoked as needed to encourage the participant to expand a particular anecdote or add more details to the answer. We encouraged the interviewees to be unreserved and fluidly accommodated the changes in the course of discussion.

Data collection: All interviews were conducted using Google Hangouts. Face to face interviews were infeasible due to the geographical distribution of subjects (Tbl. II). Each interview lasted 40–60min and generated on average fourteen pages of verbatim. The transcriptions were approved by the subjects regarding narrative accuracy and interpretive validity.

B. Participatory Observations

The observations are part of a three years action research project in which we actively take part in improving the quality of ROS and quality assurance in ROS. The observing

TABLE III
Key parts of the interview framework

intro	Can you talk to me about your community?
	What motivated you to participate in this community in the first place?
core	Can you discuss your company engagement in the community? How do you engage with the community?
	Can you discuss your company contributions and contribution process? Are there any constraints from the company or from the community to contribute?
probing	Do you have a process in place for contributing to the community?
	What is the management's attitude toward contributing to FOSS?
	What type of contributions you allowed to contribute?
	Can you share with us an example of your company contributing to the community? And how you went about it?

TABLE IV
Participatory observation venues

Year	Community event	Occurrences	Size
2017	Danish ROS MeetUp	1	30 persons
2017	ROSCon 2017 conference	1	large event ($\approx$ 500)
2017	ROS Industrial Conference	1	large event ($\approx$ 200)
2018	ROS Industrial Developers Meeting	7	10 persons
2018	ROS Quality Assurance Working Group	12	23 persons
2018	Danish ROS MeetUp	1	15 persons
2018	ROSCon 2018 conference	1	large event ($\approx$ 500)
2018	ROS Industrial Conference	1	large event ($\approx$ 200)
2019	ROS Quality Assurance Working Group	4	16 persons

researcher becomes explicitly part of the process being examined [27]. Observation helps him to understand what is going on in the daily development of a particular social group. Sofaer argues [27] that it is impossible to get sufficient exposure to a group without becoming a participant—it is through interaction with the participants that a researcher can come to sense the feelings, attitudes, and perceptions of the subjects. Thus, in contrast to the interviews, which present a reasoned perspective of the subjects, observations expose direct attitudes, complementing the interview data.

In the field: We embedded ourselves in the ROS community by attending community events and meetings—in total 29 sessions. We helped to establish and joined the monthly meetings of the ROS quality assurance working group. The group consist of 26 members, other than the exception of one student, all members are affiliated to robotics companies, or research institutions. We also established report with the core team. The inclusive nature of the ROS community made us feel part of it fairly quickly. We took the stance of moderate participants, which allowed us to balance between being insiders and outsiders. Table IV summarizes the participatory activities.

Data collection: We observed the community while participating. The data were collected through three techniques: (1) informal conversations, (2) direct observations and (3) participation in community events and activities. Notes were taken on-the-fly and fields notes were compiled afterwards; in total 30 field notes, each 1.5 page long on average.

C. Data Analysis

We used thematic coding. We analyzed the collected empirical material following the guidelines of Robson and McCartan [28] and of Miles et al. [29]. We examined the data line-by-line using the following questions as a lens to identify codes (open coding): 1) What is this saying? What does it represent? 2) What is happening here? 3) What is at issue here? 4) What is he trying to convey? 5) What process is being described? When answering these questions, we assigned labels to the verbatim. Table V summarizes the selected themes and how they were inferred from the data. One author conducted the coding and the other author confirmed the emerging theory and categories from the collected data. Six debriefing sessions were organized where the examination of the codes and the coding process has taken place. The outcome of the data analysis was presented to the participants for validation (i.e. member checking). We shared the whole findings with all the participants and asked for feedback. All participants, who responded to our emails, confirmed our interpretation and supported the findings.

V. Findings

A. Models of Commercial Engagement in FOSS

In response to RQ1, we identify three participation models among our subjects: passive, active, and latent.

Passive participation: We have observed several cases of *passive participation*, where *an organization leverages the community products without contributing back*. For instance, Participant 1 admits that their passive attitude was conscious and strategic: *"Our strategy was from the beginning not to contribute (...) Soon, [we] will start contributing to bug fixes... There is a sentiment among our engineers to contribute back."*

Finding the right balance between contributing and reaping benefits is difficult for FOSS adopters, who struggle to protect themselves against competition while meeting the needs of customers [25]. Clearly, some subjects opted for non-pecuniary inbound engagement. (The inbound open innovation is the exploitation of externally available knowledge [30], [31].) Huizingh reports that companies engage in inbound open innovation deliberately, mostly due to concerns with sharing knowledge [31], [32].

It is known that the perspective of the active FOSS community is different: passivity adds little value to the growth and sustainability of the community; "free riders" [33], [34], who do not contribute to the development of the community but "reap the benefits," are a concern. This probably concerns some of the engineers, who are the part of the company directly interacting with FOSS contributors in online discussions.

> ***Observation 1.*** *Some of the studied companies consciously decided to benefit from FOSS in an inbound-only manner, without contributing back. Interestingly, the engineers, who interact directly with the community, contest this decision.*

Active participation: The passive participation seems to be raised more often as an issue in the ROS community than in Linux. However, also in the ROS community some of the organizations begin to realize that passive participation is not fully productive: *"We learned our lessons! Not up-streaming is a losing strategy."*[2] To fully exploit FOSS, organizations need to find ways to benefit also from giving, for example to share the cost of maintenance, to receive community-developed fixes, compatible extensions, and new features. Companies also realize that if the profitability depends on the success of the community, the long-term health of the FOSS project is also of importance for them. Some of the companies that we interviewed in the Linux Kernel community, have successfully built an *active participation* model that depends on the community: *all their developments are up-streamed and their engineers are an integral (even core) part of the community.*

Prior research confirms that it is possible and beneficial to combine non-pecuniary and pecuniary involvement with external knowledge, sharing cost and bearing cost of innovation [35]. Open innovation provides opportunities to reduce development costs, to shorten development time, and to enrich internal innovation processes [25]. Our subjects indicate that understanding of this tends to grow in organizations over time.

> ***Observation 2.*** *Some subjects argue that over time it is possible, and even beneficial, to develop an active FOSS participation strategy that combines pecuniary and non-pecuniary contributions, both inbound and outbound.*

[2] A quote from a ROS community event; up-streaming is contributing code under an open source license, which lets the community to take over maintenance.

TABLE V
THEMES: EXAMPLES, DEFINITIONS, AND WHY THEY WERE CHOSEN

Theme	The theme indicators in our data	Example of verbatim
Objection of senior management	Reports of decisions made by upper management against contributing, for example: Participant 10 links the company's decision not to contribute to the management's limited understanding of FOSS. Hence the lack of understanding and company policies are the reasons behind the objection. We also see, that the senior management owns the objection, not the subject.	*"We usually don't contribute that much to the community... It is part of this company policies but it is hard to contribute outside the company. One side I guess is a cultural thing so let's say that our bosses they don't understand well this open source and this community ideas, they don't understand that very well."* (Participant 10) *"We had a lot of push backs from management. We've done a lot of convincing."* (Participant 7)
Company's image	Direct associations between the quality of deliverable originating in the company and the company's image, for example: Participant 7 linked the company image to the quality of contribution.	*"I guess it is also an image thing. So every time you are contributing to something that is public and you are using your company name to contribute, they [management] want to be sure that the quality or the contribution is really valuable."* (Participant 7) *"Our management is concerned about our image. There is a lot of scrutiny over contributions."* (Participant 11)
Intellectual property	IP is repeatedly discussed by subjects, often linked to management's believe that contributing reduces the competitive advantage; a side-effect of (mis)understanding the FOSS cost/benefit model.	*"The main obstacle to upstreaming our code is management concerns of loosing the competitive edge."* (Participant 17)
Undefined processes and policies	Several subjects made a direct connection to the lack of clear policies and processes being problematic (an indecision).	*"It is confusing to most people. Our policies and processes are not clear! It create confusion when you can and when cannot contribute."* (Participant 6)
High cost of participation	The cost of participation appears in both interviews and observations. The subjects are aware of the additional burden introduced by contributing to FOSS.	*"The cost of upstreaming is high. You not only have to produce good code, but good Linux Kernel code that is accepted by the community"* (Participant 18)
Unfamiliarity with the "system"	Direct and indirect suggestions of unfamiliarity with the "system": the community rules, conventions, and processes. Idiosyncratic to the Linux community.	*"It's not that simple! a successful engagement requires familiarity with the system in place. Most companies are not."* (Participant 18)

Latent participation: Some organizations exercise a compromise *latent participation* model, where the *release of internally developed features is delayed until an economic gain has been guaranteed*: *"We need to recover our internal investment first before we can open source anything."* (a ROS developer asked if his company is willing to open source newly developed features). This selectively revealing strategy relies on keeping some parts internal while releasing less profit-making assets, exploiting dual licensing and restrictive licensing.

The latent model benefits both the company and the FOSS community. Contributing parts of the development, allows to embed developers in the community, and influence its direction. Simultaneously, it allows the open source community to push the organization toward sharing more [25], as we also observe in our data. Some authors recommend selective revealing [20], contributing parts considered as a commodity while keeping the differentiating components closed. Van der Linden et al. [36] emphasize that the timing of the contribution versus the release of the feature is key—the functionality will become commodity eventually due to a constant progress of technology. Also, this strategy does create a synchronization issue between the community version of the software and the in-house instance.

Observation 3. *The latent participants neutralize risks of disclosing the differentiating IP, while still benefiting from a better embedding into community than the passive participants (a non-pecuniary inbound innovation combined with deferred non-pecuniary and pecuniary outbound collaboration).*

B. Barriers to Commercially Affiliated Participation in FOSS

Since affiliated participation occurs under the umbrella of an institution, it is performed under some constraints. While affiliated participants have to follow rules, structures, and guidelines of their employer, independent participants are free of rules and have no organizational authority to report to. In response to RQ2, we identified six barriers to affiliated participation described in the paragraphs below and summarized in Tbl. VI. Each dot on the table indicates the presence of the behavior on the corresponding community or participation model. The last three columns represent where the behavior originates from. Each dot in those columns is an indication that the institution, community or individual is the originator of the behavior. There are slight differences between what barriers are experienced by subjects in the two communities, and by subjects adhering to various participation models. Also, the barriers seem to have various sources. We return to these issues in discussions below.

TABLE VI
BARRIERS VS COMMUNITIES, PARTICIPATION MODELS, AND ORIGINS

Participation barrier	ROS	Linux	Passive	Latent	Active	Institution	Community	Individual
Senior management objection	•	•	•			•		
Company's image	•		•			•		
Intellectual property protection	•	•	•	•		•		
Undefined processes & policies	•	•	•	•		•		
High cost of participation		•		•	•		•	•
Unfamiliarity with the system		•		•	•		•	•

Objection of senior management: Some subjects indicate that senior management shows little understanding of open source community environment, social structure and processes. They are willing to consume the community goods, but resist contributing actively. *"Our bosses don't understand well this open source and this community idea"* (Participant 8, asked why the company does not contribute to the community). Unfamiliarity with FOSS is just one of the reasons. This objection is based on various grounds. *"Our management does not support contributing back to the community. They [management] have several reasons"* (Participant 2). Active participants also admit that it takes commitment of the company, not only of the engineers, to succeed: *"My company is fully committed to the community. We upstream everything we produce internally"* (Participant 13).

The passively participating companies, are typically used to produce proprietary software and engage contractually with other parties, where risks are managed, and relationships are under control. The risks of contributing actively are unknown. It is unclear how to mitigate them and how to calculate the benefits. A passive participation is safer.

Synchronizing the product strategy and the participation model helps to realize the full benefits of FOSS participation [25], [37], [38]. Little is known about how companies need to design their business models to match different open innovation strategies. For this reason, companies may mistakenly think of open innovation as yet another "off the shelf" management practice that can be implemented almost as an add-on to existing practices and organizational arrangements in the company.

Observation 4. *Subjects in actively participating companies enjoy support of the management. Subjects in passively participating companies often indicate lack of management's support as a constraining factor; apparently caused by lack of experience and little proven business practice.*

Protection of the company's image: The FOSS contributions represent the company publicly, or at least to the respective community. The company's image is easily associated with their quality. Thus, we experience concerns that negative judgments of contributions may affect this image: *"It is also an image thing. So, every time you are contributing to something that is public and you are using the company name to contribute, they[management] want to be sure that the quality or the contribution is really valuable... there are a lot of thresholds to do that"* (Participant 8).

Businesses realize that they need to create a desirable and positive corporate image, not only through marketing resources, but also by creating positive and avoid negative situations. A passive engagement is a risk mitigation strategy that can help to shelter the company image. However, several active contributors have turned this situation around, by exploiting FOSS in branding and in attracting high quality employees. In particular, we see that once the FOSS project brand is strong (e.g. Linux), companies are more likely to try to exploit it. ROS users are much more reserved about this.

Observation 5. *Some of subjects see the FOSS community as a channel in establishing, maintaining or improving image of their brand, while others do not know how to do that.*

Protection of intellectual property (IP): We registered concerns that FOSS participation implies disclosing competitive IP. The idea of sharing source developed in-house is foreign; anything produced by an employee should remain protected in the company. Businesses are reluctant to expose the differentiating technologies and to risk loosing the competitive advantage. *"The main concern is leaking our proprietary code and any architectural design that's in the code. We use ROS but we have our own architecture on how to use ROS"* (Participant 2). *"My boss object up-streaming our work. He thinks that will reveal how we do things to our competitors"* (Participant 17).

This protective attitude (regardless if justified!) is at odds with core FOSS values: sharing and openness. In the communitarian philosophy of FOSS, withholding contributions to protect IP slows down the collective innovation process and favors a single entity. Openness is a manifestation of two cultural traits of open source communities: transparency and truth [39]. Pavlicek [39] believes that truth is a fundamental community asset [39]. He explains truth and transparency empowers the community to produce free software. This conflict of positions (community values and the protection of IP) is a ground reason for passive engagement with the community.

Henkel [20] claims that management is overly concerned about openness, concluding that a more positive attitude increases benefits of open innovation. Yet, numerous other authors advise companies to contribute with commodity features and keep differentiating factors in-house [20], [37], [40]. Bosch [41] and Van Linden et al. [36] explain that the release of commodity functionality has its advantages; companies can benefit from the reduction of the cost of maintenance and focus on the differential capabilities.

Observation 6. *Passive and latent participants object to active participation for IP loss concerns. Active participants have developed a business model that is less IP-sensitive.*

Undefined participation process and policies: Some of the studied organizations suffer from a lack of formal participation policies and governance. They think it is not important, or even not necessary, to adjust their internal processes to the community engagement. Participant 7, was asked if there is a process in place to manage contributions, replied: *"I'm talking to my management to set up a process where if we develop a driver for example we can contribute back to the community. So, I'm working on the process."*

The lack of polices and processes confuses the participating engineers. Some of them have roots in FOSS. They have contributed since they were students. Some were even hired based on the FOSS record. They still display strong hacker mentality, but are uncertain how being paid affects the engagement with the community. Munir et al. [24] postulate that software organizations that want to benefit from open innovation via FOSS engagement need to adapt their internal processes.

Observation 7. *Subjects, who have successfully implemented an active participation model, have aligned their internal policies to reflect the community engagement. Those with passive and latent participation have not done so.*

High cost of participation: Subjects admit that active engagement is expensive: *"The cost of getting something through this process [upstreaming] is high." "[Passive participation] is the easy way to engage with the community. It takes a lot of effort to produce code that is up-streamable"* (participants 17 and 11 resp.). Both financial and psychological costs of community engagement are high. This is most visible in the Linux case, which is known for a very high barrier of entry. The typical costs include preparing the code contributions at the expected quality, meeting coding styles and conventions, accepting rejections, and dealing with multiple review cycles, preparing documentation, and tests. Often lengthy negotiations with other community members are required.

The economic formula for the participation is not well understood. If the community engagement is not seen as a long term investment but rather as seeking "freebies," we are probably dealing with a rather short term uninformed vision. Munir et al. [24] explains that FOSS participation can be costly. Open innovation is costly and it is not always easy to start it. It should be determined by the strategic, organizational, and managerial contexts of the firm, and the benefits and costs must be evaluated. In such case it would be able to not only generate cost, but also the appropriate profit [42].

Importantly, unlike the other barriers discussed above, the cost of participation is controlled also by the FOSS community. While the large part of this cost may be inherent to the process, the community has some influence on how high the barrier of entry is, and how expensive it is to adjust to the collaboration.

Observation 8. *The cost of participation may be high, so companies need to integrate efficiently, weighing the cost against the prospective benefits. FOSS communities should be careful not to incur undue cost, especially on newcomers.*

Unfamiliarity with the "system": We find references to the FOSS "system" in our data, and statements that the "system" is a further constraint to commercial participation. The "system" refers to the social order, rituals, norms and practices of the community. Participant 18 states: *"[to] understand the process how this works. That's a big thing."* Participant 16 concurs: *"Understanding the system is something that's take time and management doesn't see the value on that."*

The high cost of participation and the unfamiliarity with the system may be addressed by hiring engineers with prior successful engagement in the community. *"When we hire new people, we always look for a cultural fit. We look for past experience in the community. Most people stay for long time. But, there are people who do not fit culturally."* (Participant 11) The contributors should share the community values and passion for the project. *"In order to be successful, you need to have passion for the project. Whoever working for the project needs to have that passion"* (Participant 12).

Observation 9. *Engineers inexperienced with the community culture and processes struggle to fit in and are inefficient. Hiring community members counteracts this and tightens the bond between the company and the FOSS project.*

VI. Trustworthiness

An important aspect of any qualitative endeavor is establishing trustworthiness [43]. Qualitative researchers establish that the findings of the study are credible, transferable, dependable, and confirmable. Trustworthiness is assured by the establishment of these four traits [43]. We will briefly discuss how we established these traits (see tbl. VII).

Credibility: The techniques we employed to address credibility are, namely, prolonged engagement, persistent observation, and methodological triangulation [44]. Peer debriefing has been used during the research process, one author conducted the analysis and the second author validated the emerging theory against the raw data. Six debriefing sessions were organized. We also assured credibility by member checking with the participants to test the findings and interpretations. We sent the interviews transcripts and description of the findings to the participants for validation.

Transferability: Transferability is the degree to which the results can be transferred to other contexts, sites or settings. [43]. In qualitative research, this quality of transferability refers to case-to-case transfer [45]. We provided thick descriptions of the research methods so that others can judge transferability [43].

Dependability: To ensure dependability we provided information that is s logical, traceable, and clearly documented [45]. When the research process is described completely, readers are better able to judge the dependability of the research [43]. If the process of the research can be audited, then it can ensure dependability [43].

Confirmability: Confirmability is the characteristic of the match between the researcher's interpretations and findings and the data, which requires the researcher to demonstrate how the conclusions and interpretations were made [45]. According to Guba and Lincoln [43], confirmability is established when credibility, transferability, and dependability are all achieved. In addition, we compiled an audit trail throughout the study. An audit trail is a documentation that provides readers with evidence

TABLE VII
Strategies and techniques employed in the study to meet trustworthiness requirements

Strategies used to establish trustworthiness	Credibility	Transferability	Dependability	Confirmability
Prolonged engagement	•			
Peer debriefs	•		•	
Observations	•			
Triangulation	•		•	
Participants checks	•			
Comprehensive and transparent research method		•	•	
Audit trail		•	•	•

of the decisions and choices we made, including theoretical and methodological issues in the study and a clear rationale for all decisions. The audit is useful for other researchers to follow the decision trail and reach the same conclusions.

VII. Discussion: How to Overcome the Barriers?

We now enter a more speculative mode of reasoning, and consider what actions and solutions emerge from our data that help companies and communities to overcome the barriers.

Table VI shows how the participation barriers map to models in our data. Clearly, subjects following different models were focused on different barriers. The active participants have likely been able to overcome the first four barriers that relate to company's management. A promising picture emerges that organizations might be able to evolve from passive participants, through latent contributors, to full active community members, who (as per reports of our subjects) find the participation beneficial. Barriers originate not only in the institutions, but also in the communities and individuals (Tbl VI). We stipulate that barriers need to be addressed gradually and on all sides.

In Table VIII we contrast the perspective of active and passive participants on each of the barriers. We provide examples of statements from both sides in the two middle columns and add a commentary in the rightmost column. In the following text, we summarize what actions suggest themselves.

Objection of senior management: We find that active participants have overcome senior management objections. It appears that engineers seeking active participation in FOSS projects in their work choose employers where management is committed. Software teams seeking active participation should prioritize good communication with senior management, and work towards commitment.

Company's image: While active participants leverage the community success to support their brands, passive and latent participants see the FOSS community as a liability to their image. This in itself means that the FOSS projects have an image value to be exploited—a certain quality and maturity stamp. At the same time, it is clear, that businesses interested in beneficial symbiosis with FOSS projects may want to evaluate the reputation of said projects first. A low reputation project incurs an image cost, while a high reputation project can be used in branding more easily.

Protection of intellectual property: Licensing fees are far from the only way to profit in the software industry. Hardware sales, support, consulting are known reliable streams of income. Furthermore, in fast moving software sectors, speed of innovation may be more important than any temporary technological advantage. Thus, some of the active participants foresee releasing their features' IP as a strategic trade-off.

Based on this study, we can recommend to identify a suitable business model, consistent with the FOSS participation model. If indeed stringent IP protection is key to profitability, we cannot recommend active participation. However intermediate forms, where features are released with delay, or non-critical features are contributed, can already enable benefits of FOSS participation, such as lower development cost, higher quality, and using the community for growing the market share.

Undefined participation process and policies: Participation processes and policies should be documented and communicated to the engineers, regardless of the participation model. Lack of defined participation processes and policies confuses the engineers, who need transparency regarding what can and what should be done when representing a company in the FOSS community. Moreover, lack of policy increases unnecessary risks, like premature release in the latent or passive participation models. In the active model, engineers need clear feedback that up-streaming and release engineering are indeed seen by the company as legitimate use of time—otherwise the benefits might not be fully unlocked. Finally, clear policies help the participating engineers to distinguish goals of the open source community and of the company—these do not have to be the same, and they do influence engineering decisions. This awareness should be used to affect the direction of the community.

High cost of participation: The cost of participation is primarily generated by the FOSS community itself, thus we recommend that communities consider whether some costs are not undue. For instance emotional costs (e.g. when communicating a patch rejection) can be reduced by using constructive language and avoiding hostilities. The ROS community is discussing associating mentors to newcomers, who can help them in the integration process. Some communities offer training materials, or training courses. Stewarding foundations accept donations to pay experienced members for some work, instead of using the in-house engineers who may incur a higher overhead if inexperienced.

Unfamiliarity with the "system": Similar measures should be taken as for the cost of participation. Companies in both studied communities, and at all participation models, recommend hiring experienced FOSS engineers, preferably directly from the community. This entirely elevates this barrier, and it has a side effect that it tends to bring technical excellence to the company. In addition experienced FOSS contributors could mentor inexperienced employees, who have little experience in dealing with FOSS communities.

The participation of a commercial and non-commercial institutions in a FOSS community occurs through an employee or a group of employees. Sometimes, the affiliation remains anonymous: *"You sign up with your name not with the name of the company. People know my name not the company."* (Participant 12) However, in some instances, the member is known in the community to represent a company. Participation mechanisms in the FOSS communities have been originally created for individuals not for organizations. The increase in commercial participation calls for a change to the community participation model. For example, the sign-up form for the ROS community online forum is designed to capture individual demographic data only, but neglects the fact that the member signing up may be working for a company.

TABLE VIII
CONTRAST ANALYSIS: PARTICIPATION BARRIERS MANIFESTATION AND HOW THEY ARE RESOLVED IN SUBJECT ORGANIZATIONS.

Barrier	Manifestation (Passive Perspective)	Resolution (Active Perspective)	Researcher's Commentary
Objection of senior management	*"Our bosses they don't understand well this open source and this community ideas."* (Participant 10)	*"Our management is committed to the Linux community [...] very supportive of the community... We communicate all the time with management"* (Participant 11)	Poor understanding (passive) may be due to a communication failure. Managers in the active organization established a good communication channel with engineers.
Company's Image	*"Every time we want to contribute, our management object and each time the excuse is the quality many not be good enough for our company's image."* (Participant 12)	*"We believe contributing is good for our brand. It's a strong and successful project"* (Participant 15)	The employer of Participant 6 (passive) uses the risk optics to assess the cost, while the employer of Participant 15 (active) consciously exploits the reputation of Linux to strengthen own brand.
Protection of IP	*"The main concern is to reveal our architecture to our competitors."* (Participant 16)	*"To strike an effective balance between open source and proprietary code, the key is to think strategically... We engage with a strong community to help us with the process, identify bugs, and maintain a steady pace of new feature releases. Not developing these capabilities exclusively in-house frees up our engineers to focus on projects that really drive the business."* (Participant 15)	The passive participant clearly values the protection of IP more than the cost savings and the open innovation potential. The active participant has made a strategic decision to adopt FOSS both inbound and outbound, lowering cost and accelerating innovation.
Undefined processes & policies	*"We do not have an internal process or policies in place to tell us how and when to contribute."* (Participant 10)	*"Our policy is to upstream everything."* (Participant 11)	The passive participant suffers from lack of clear policy. The employer of Participant 11 (active) has a clear policy, that leaves no doubt to the engineers.
High cost of participation	*"It's not easy to contribute and when we do we get told it's not relevant or its quality not good enough. This process is very costly for us"* (Participant 18)	*"We are an integral part of the community."* (Participant 11)	Participant 11 has integrated into the community, learned its processes and accepted costs, including the initial hurdles. See also the comment under IP above.
Unfamiliarity with the "system"	*"When we hire cultural fit is very important... It takes time to get familiar with the system"* (Participant 14)	*"We look for past community engagement and participation. It is important to us the community exposure."* (ROS QA working group member)	A cultural fit and familiarity with the system is important to both inbound and outbound FOSS adoption.

VIII. LIMITATIONS

We briefly discuss the limitations of this study. First, the findings may not translate to other communities. Although, ROS and Linux exemplify commercial participation in FOSS, the behavior described in this paper may take different form in other communities. Still note that we observed a saturation of material with last interviews and events. Second, our participatory data only covers ROS. This make the data skewed toward ROS. But was valuable for triangulation. Third, robotics and operating systems are two distinctly different domains with two recognizably different populations of participants, consumers and vendors. These products target different markets with distinct dynamics. Interview and observation data reveal participants perspective, but do not capture the market dynamics and its influence on participation. In addition, with the exception of three participants (co-founder, director and project manager), most of our participants are developers. Finally, ROS and Linux are different in their development journey. While the Linux community is 28 years old, the ROS community is hardly 12.

IX. CONCLUSION

We have identified six participation barriers and discussed how they affect the three participation models, discussing them against known research. According to the body of knowledge on open innovation, the four interlocking elements of the business model are customer value proposition, profit formula, key resources, and key processes [46]. Our data confirms that these elements need to be understood and arranged in order for the company to reach the maximum level of beneficial involvement in FOSS. The high cost of participation prevents success if the company's value proposition is not sufficiently linked to an open source project. IP protection rules out open active participation for companies whose market advantage and profit is based on technology confidence. Management needs to devote resources and regulate participation to fully exploit the collaboration with FOSS—not just download free source code, but also to share the maintenance burden, receive bug fixes and new features to the up-streamed code. Depending on whether these conditions can be met, companies settle on passive, active, and latent participation models.

While there is lots of speculation regarding FOSS-based business models in open source advocacy writing, relatively little solid and documented patterns can be found in research works. This study has identified that active and latent participants are a good source of subjects to systematically collect and document such successful business models, to the benefit of software companies considering FOSS involvement.

Linux and ROS are clearly projects in related but very different domains. The operating systems market is 50 years old, large and mature, with established few key players. The robotics market has been extremely lively the last ten years, when the technology begun to meet the wider commercial applicability threshold. Most companies are small start-ups, complemented by few established machine and automation technology giants. Given the expected growth of commercial robotics, it is very interesting to investigate the business models for robotics companies adopting FOSS.

Acknowledgments: Work supported by the EU's H2020 research and innovation programme, grant No 732287 ROSIN. We thank the interviewees for making this research possible.

REFERENCES

[1] G. Anthes, "Open source software no longer optional," *Communications of the ACM*, vol. 59, no. 8, 2016.

[2] Ø. Hauge, C.-F. Sørensen, and R. Conradi, "Adoption of open source in the software industry," in *IFIP International Conference on Open Source Systems*. Springer, 2008.

[3] M. Michlmayr, F. Hunt, and D. Probert, "Quality practices and problems in free software projects," in *Open Source Systems*, 2005.

[4] J. West and S. O'Mahony, "Contrasting community building in sponsored and community founded open source projects," in *HICSS'05.*, 2005.

[5] A. Bonaccorsi and C. Rossi, "Why open source software can succeed," *Research policy*, vol. 32, no. 7, 2003.

[6] R. A. Ghosh, "Economic impact of open source software on innovation and the competitiveness of the Information and Communication Technologies (ICT) sector in the EU," 2007.

[7] S. Goode, "Something for nothing: management rejection of open source software in Australia's top firms," *Information & Management*, vol. 42, no. 5, 2005.

[8] U. Nikula and S. Jantunen, "Quantifying the interest in open source system: case south-east Finland," in *1st International Conference on Open Source Systems (Scotto, M. and Succi, G. Eds.)*, 2005.

[9] R. A. Ghosh, R. Glott, B. Krieger, and G. Robles, "Free/libre and open source software: Survey and study," 2002.

[10] D. A. Wheeler, "Why open source software/free software (OSS/FS, FLOSS, or FOSS)? Look at the numbers," 2007.

[11] T. Waring and P. Maddocks, "Open Source Software implementation in the UK public sector: Evidence from the field and implications for the future," *Intl. Journal of Information Management*, vol. 25, no. 5, 2005.

[12] M. Välimäki, V. Oksanen, and J. Laine, "An empirical look at the problems of open source adoption in Finnish municipalities," in *7th International Conference on Electronic Commerce*. ACM, 2005.

[13] F. Hecker, "Setting up shop: The business of open-source software," *IEEE software*, vol. 16, no. 1, 1999.

[14] B. Fitzgerald and T. Kenny, "Developing an information systems infrastructure with open source software," *IEEE Software*, vol. 21, no. 1, 2004.

[15] A. Bonaccorsi, D. Lorenzi, M. Merito, and C. Rossi, "Business Firms' Engagement in Community Projects. Empirical Evidence and Further Developments of the Research," in *Emerging Trends in FLOSS Research and Development, 2007. FLOSS'07.* IEEE, 2007.

[16] Z. Obrenovic and D. Gasevic, "Open source software: All you do is put it together," *IEEE software*, vol. 24, no. 5, 2007.

[17] Ø. Hauge, "Adoption of Open Source Software in Software-Intensive Industry," Ph.D. dissertation, PhD thesis, 2010.

[18] M. Höst and A. Oručević-Alagić, "A systematic review of research on open source software in commercial software product development," *Information and Software Technology*, vol. 53, no. 6, 2011.

[19] K. R. Lakhani, R. G. Wolf, and Others, "Why hackers do what they do: Understanding motivation and effort in free/open source software projects," *Perspectives on free and open source software*, vol. 1, 2005.

[20] J. Henkel, "Selective revealing in open innovation processes: The case of embedded linux," *Research policy*, vol. 35, no. 7, 2006.

[21] B. Lundell, B. Lings, and E. Lindqvist, "Perceptions and uptake of open source in swedish organisations," in *IFIP International Conference on Open Source Systems*. Springer, 2006.

[22] L. Dahlander and M. G. Magnusson, "Relationships between open source software companies and communities: Observations from Nordic firms," *Research policy*, vol. 34, no. 4, 2005.

[23] H. Chesbrough *et al.*, "Open innovation," 2003.

[24] H. Munir, J. Linåker, K. Wnuk, P. Runeson, and B. Regnell, "Open innovation using open source tools: a case study at sony mobile," *Empirical Software Engineering*, vol. 23, no. 1, 2018.

[25] H. Munir, K. Wnuk, and P. Runeson, "Open innovation in software engineering: a systematic mapping study," *Empirical Software Engineering*, vol. 21, no. 2, 2016.

[26] E. van Burg, C. Giannopapa, and I. M. Reymen, "Open innovation in the european space sector: Existing practices, constraints and opportunities," *Acta Astronautica*, vol. 141, 2017.

[27] S. Sofaer, "Qualitative methods: what are they and why use them?" *Health services research*, vol. 34, no. 5 Pt 2, 1999.

[28] C. Robson and K. McCartan, *Real world research*. John Wiley & Sons, 2016.

[29] M. B. Miles, A. M. Huberman, and J. Saldana, *Qualitative data analysis*. Sage, 2013.

[30] U. Lichtenthaler and E. Lichtenthaler, "A capability-based framework for open innovation: Complementing absorptive capacity," *Journal of management studies*, vol. 46, no. 8, 2009.

[31] E. K. Huizingh, "Open innovation: State of the art and future perspectives," *Technovation*, vol. 31, no. 1, 2011.

[32] D. Kline, "Sharing the corporate crown jewels," *MIT Sloan management review*, vol. 44, no. 3, 2003.

[33] W. Scacchi, "Computer game mods, modders, modding, and the mod scene," *First Monday*, vol. 15, no. 5, 2010.

[34] M. L. Markus, "The governance of free/open source software projects: monolithic, multidimensional, or configurational?" *Journal of Management & Governance*, vol. 11, no. 2, 2007.

[35] S. W. van Rooij, "Open Source software in US higher education: Reality or illusion?" *Education and Information Technologies*, vol. 12, no. 4, 2007.

[36] F. Van der Linden, B. Lundell, and P. Marttiin, "Commodification of industrial software: A case for open source," *IEEE software*, vol. 26, no. 4, 2009.

[37] K. Wnuk, D. Pfahl, D. Callele, and E.-A. Karlsson, "How can open source software development help requirements management gain the potential of open innovation: an exploratory study," in *ESEM'12*, 2012.

[38] W. Stam, "When does community participation enhance the performance of open source software companies?" *Research Policy*, vol. 38, no. 8, 2009.

[39] R. Pavlicek and R. Foreword By-Miller, *Embracing insanity: Open source software development*. Sams, 2000.

[40] J. West and S. Gallagher, "Challenges of open innovation: the paradox of firm investment in open-source software," *R&d Management*, vol. 36, no. 3, 2006.

[41] J. Bosch, "Achieving simplicity with the three-layer product model," *Computer*, vol. 46, no. 11, 2013.

[42] F. Michelino, M. Caputo, A. Cammarano, and E. Lamberti, "Inbound and outbound open innovation: organization and performances," *Journal of Technology Management & Innovation*, vol. 9, no. 3, 2014.

[43] E. G. Guba and Y. S. Lincoln, "Naturalistic inquiry (Vol. 75)," *Beverly Hills, CA: Sage*, 1985.

[44] N. K. Denzin, *Sociological methods: A sourcebook*. Routledge, 2017.

[45] S. J. Tracy, "Qualitative quality: Eight "big-tent" criteria for excellent qualitative research," *Qualitative inquiry*, vol. 16, no. 10, pp. 837–851, 2010.

[46] T. Saebi and N. J. Foss, "Business models for open innovation: Matching heterogeneous open innovation strategies with business model dimensions," *European Management Journal*, vol. 33, no. 3, 2015.