The emergence of openness:

How and why firms adopt selective revealing in open innovation

Published in Research Policy 43 (2014), 879–890

Joachim Henkel[1, 2, *], Simone Schöberl[1, 3], Oliver Alexy[1]

[1] Technische Universität München, TUM School of Management, Arcisstr. 21, 80333 Munich, Germany. T: +49 89 289-25741, F: +49 89 289-25742, E: henkel@wi.tum.de, o.alexy@tum.de

[2] Centre for Economic Policy Research (CEPR)

[3] McKinsey & Company, simone_schoberl@mckinsey.com

[*] corresponding author

Abstract

Open innovation is often facilitated by strong intellectual property rights (IPRs), but it may also function, and even be boosted, when firms deliberately waive some of their IPRs. Extant literature has pointed out the potential benefits of such behavior, but falls short of explaining what triggers firms to practice it in the first place and to maintain or extend it. Since the waiving of IPRs runs counter to common views on strategy and competition and to engrained practices, this is a non-trivial question. To address it, we conduct an empirical study in a segment of the computer component industry which traditionally has taken a rather proprietary stance. With the advent of the open source operating system Linux, firms increasingly waived their IPRs on software drivers. We trace and analyze this process using both qualitative and quantitative methods. Our results indicate that component makers went through a learning process, which led some to realize how selectively waiving IPRs may be beneficial for their business. We uncover customer demand pull as the initial trigger and observe how a positive feedback loop sets in subsequently, leading to a further increase in the use of selective revealing. Overall, we find that openness develops into a new dimension of competition. We discuss the implication of our findings for research on open innovation and highlight how they impact managers in practice.

Acknowledgements

The authors are grateful for feedback from guest editor Joel West, editor Ashish Arora, Linus Dahlander, Henning Piezunka as well as seminar participants at Harvard Business School, UNC Kenan-Flagler Business School, EPFL Lausanne, TUM School of Management, ESC Rennes, and participants at DRUID and the Open Innovation Conference, Imperial College, June 2012. Alexy acknowledges financial support from the Engineering and Physical Sciences Research Council's IMRC at Imperial College London.

1 Introduction

In essence, innovators have two means of maintaining exclusivity to innovation-related knowledge: secrecy and legal exclusion rights. When practicing outbound or the 'coupled' type of open innovation, the latter characterized by combining inflows and outflows of knowledge (Enkel et al., 2009), the innovator by definition gives up secrecy on the outbound knowledge. Yet, legal exclusion rights may remain and, it has been argued, these are often beneficial or even necessary for successful open innovation (Arora et al., 2001; Chesbrough, 2003& 2006a).

But open innovation may also function without exclusion rights, in situations where legal protection is either ineffective or voluntarily waived by the innovator. The latter case, termed "selective revealing" by Henkel (2006), is most interesting from a strategic perspective. Contrary to received wisdom, various studies have shown that firms may benefit from voluntarily waiving some of their intellectual property rights (IPRs) in an open innovation process (e.g., Chesbrough and Appleyard, 2007; Pisano, 2006; West, 2003). In doing so, they may, among other things, increase the efficiency and effectiveness of their R&D efforts by enticing other actors to join in co-creation of value or increasing their chances at standard setting.

Yet, for firms steeped in the paradigm of closed innovation, the transition to open innovation is challenging—and in particular the adoption and implementation of selective revealing (Alexy et al., 2013b). Waiving legal exclusion rights entails obvious risks, while the benefits are uncertain and need to be proven. Indeed, it has been observed that while we have tremendously increased our understanding of open innovation and selective revealing, we still do not fully comprehend how closed firms move to open innovation in general, and to selective revealing in particular (Alexy and Dahlander, 2013; Christensen et al., 2005). We thus ask *how and why do firms try out selective revealing in open innovation, and what determines if they subsequently* *maintain or even extend this engagement*? This question is important since, first, selective revealing has a large potential—and also considerable risks—for improving innovation efficiency and effectiveness, and second, most industries are still characterized by tight protection of IP and might be ripe for such a learning process.

Since our research question is best described as "intermediate theory research," we use a multi-method design linking qualitative with quantitative methods (Edmondson and McManus, 2007). Our empirical setting is the embedded component industry, which has historically taken a rather strict proprietary stance on IP and now seems to be opening up. Taking a process perspective, we analyze the emergent strategies of embedded component manufacturers (ECMs) with respect to the revealing of driver source code for the Linux operating system. As Linux drivers may be kept proprietary with their source code secret, they are perfectly suited for studying strategic decisions about openness.[1] Furthermore, a driver is strictly complementary to the corresponding component, such that the bundle of both has to be seen as one product. The case we study is thus markedly different from the logic of giving away a program as open source software (OSS) in order to increase demand for (non-strictly) complementary offerings (e.g., Raymond, 1999). Rather, revealing a driver amounts to disclosing the blueprint of a part of the focal product. We began by conducting 16 in-depth interviews and document analysis, followed by a survey addressing the entire population of ECMs in three sectors: processors, single-board computers, and data acquisition cards.

Our main findings provide a first picture of the emergence of openness in an industry. For

[1] ECMs distribute their (compiled) drivers as part of the components they sell. If the drivers were to be considered "derivative work" of Linux in the sense of the GPL, then such distribution would require making the source code available to the receiver of the compiled code. Drivers are commonly not considered derivative work, though (unless of course they contain code under the GPL). While there is some controversy, it is accepted practice to provide Linux drivers in compiled form only (for a discussion, see Henderson, 2006).

ECMs, the growing importance of Linux triggered a change in the revealed preferences of customers, who hitherto had taken closedness of drivers for granted. Because Linux is OSS, customers now expected the driver source code to be publicly available. It was this *demand pull* that led ECMs to rethink the established practice of keeping drivers closed. The interplay of increasingly articulated customer need for openness and ECMs' experiences with openness over time initiated a *positive feedback loop* increasing average levels of selective revealing. Eventually, this process facilitated the emergence of openness not only as a phenomenon, but as a *new dimension of competition*. In elaborating on these three mechanisms (demand pull, feedback loop, new competitive dimension), we extend our understanding of open innovation, and the emergence and implications of selective revealing in this context.

2 Background

A broad and growing literature addresses potential advantages of open innovation over closed innovation (Dahlander and Gann, 2010; West and Bogers, 2013). In particular, questions about organizations purposefully sharing knowledge with external actors have received increasing attention in the literature (e.g., Clarkson and Toh, 2010; Polidoro and Toh, 2011). Sharing knowledge across firm boundaries does not imply that the originator of this knowledge relinquishes ownership. Quite the contrary—it has been argued that strong IPRs are often beneficial and potentially even necessary for open innovation (Chesbrough, 2003& 2006a; Hagedoorn and Ridder, 2012; West, 2006). Examples include R&D alliances between biotech and pharmaceutical companies (Chesbrough, 2003), contributions to open standards such as UMTS in the ICT industries (Simcoe, 2006), commercial engagement in OSS (Fosfuri et al., 2008), or transactions on markets for technologies (Arora et al., 2001) such as those offered by open innovation intermediaries (Jeppesen and Lakhani, 2010).

In other instances, though, innovators may benefit by sharing their knowledge (and thus giving up secrecy) *and* waiving, or not establishing in the first place, legal exclusion rights to it. Harhoff et al. (2003) have termed this behavior "free revealing." However, this notion is sometimes understood to describe altruistic and intrinsically motivated information sharing. In order to emphasize that we study profit-oriented behavior we use the term "selective revealing" (Henkel, 2006), which implies that the focal actor does not reveal out of principle but rather as a result of weighing the commercial pros and cons.

The matrix in Figure 1 illustrates the innovator's choices. The vertical axis describes if innovators deny or grant third parties de-facto access to their innovation-related knowledge (in other words, if they maintain secrecy or not); the horizontal axis, if innovators establish and enforce legal exclusion rights. The figure thus captures the two most important dimensions of the appropriability regime (Teece, 1986), IPRs and secrecy, and innovators' choices regarding their use or waiving of these mechanisms.[2] In this context, we are concerned with situations in which IPR protection and secrecy are possible but deliberately waived—selective revealing.

--- *Insert Figure 1 about here* ---

Several anticipated benefits will drive if and to what degree firms will engage in selective revealing. Here, marketing and technological benefits feature particularly prominently in the literature (for recent reviews, please see, e.g., Afuah and Tucci, 2012; Alexy et al., 2013a; Baldwin and von Hippel, 2011; Henkel, 2006; Henkel and Baldwin, 2011). With regards to

[2] Figure 1 simplifies in at least two respects. First, the choice at either axis is not a binary and not even a uni-dimensional one. Second, the first line ("access denied, secrecy") and the first column ("IPRs established and enforced") both need the qualifier "*to the extent possible*," which depends on the prevailing appropriability regime. Note that waiving either IPRs, or secrecy, or both means of protection is different from what Pisano (2006) refers to as an "endogenous appropriability regime."

marketing-related benefits, this literature highlights positive effects on reputation, word-of-mouth-advertising, and increased opportunities to sell the produce due to price reductions and increased customizability that are the results of selective revealing. Regarding technical aspects, benefits such as reduced production cost, increased reliability, the use of standard components, and access to new markets may allow firms to identify and successfully engage in new opportunities to create and capture value. This is particularly the case given that selective revealing mitigates transaction cost (since no costly bilateral contracting happens) and decreases hold-up (since revealing is a credible and usually irreversible commitment). At the same time, it is clear that selective revealing is not without risk. Beyond the obvious concern about imitation and loss of competitive advantage,[3] also issues of reduced compatibility, reliability, safety and security, and an increase in maintenance cost may arise.

Thus, firms need to determine on a case-by-case basis if the net benefit of selective revealing is positive. This is precisely the condition captured in “selective” revealing or in the notion of “opening parts” (West, 2003). However, even if selective revealing were to be an (objectively) positive alternative, the presence of strong inertial forces raises questions about when and why firms emphasizing closed innovation will choose to implement it. First, traditional strategizing still puts a strong emphasis on the generation of monopoly rents to be protected through formal and informal mechanism (Cassiman and Veugelers, 2002; Winter, 1987). Such strategies may also be reflected in the mission of the IP department (Alexy et al., 2009), as well as in the job and skill profiles of employees in R&D (Alexy et al., 2013b). These issues give rise to the question of why and how firms move from closed even IP-based open innovation, to open

---

[3] However, as noted by several authors (e.g., Clarkson and Toh, 2010; Polidoro and Toh, 2011), firms may also strategically benefit from others imitating selectively revealed knowledge.

innovation embedding selective revealing, and maintain or extend this practice.

3 Data and methods

3.1 Empirical context

To answer our research questions, we study the emergence of openness in the sense of selective revealing in subsectors of the computer components industry. We chose embedded processors, embeddable single board computers (SBC), and embeddable data acquisition (DAQ) cards. These components require software programs to run—so-called drivers—which are specific to the operating system used. Since the embeddable component industry is characterized by a high degree of diversity and a high importance of reliability, buyers (who are usually larger device manufacturer) particularly value driver characteristics such as customizability, reliability, the ability to fix problems directly, and adaptability to changing software platforms.

Traditionally, ECMs have taken a rather proprietary stance on IP. They kept drivers secret by distributing them in machine-readable format only and also used copyright for protection, thus positioning themselves in the top left box of Figure 1 (in some cases, they granted source code access under an NDA, which corresponds to the bottom left box). As "extreme cases" (Eisenhardt, 1989), ECMs are thus well suited for our study. Notably, with the advent of the Linux operating system and the development of variants particularly relevant to these industries (loosely called "Embedded Linux") selective revealing has become increasingly important to ECMs.

When developing drivers for Linux—a new market ECMs where quick to embrace—they face a choice of whether to keep their drivers secret as before, or whether to release the source code to their customers or even the general public under the General Public License (GPL), the open source license governing the use of Linux. This choice exists because, contrary to popular belief, revealing drivers for Linux as OSS is not mandatory. Indeed, openness of driver source code exhibits a strong variation between ECMs. It is also a strategically important decision, since it may be the driver that differentiates the product from competing offers. This revealing decision is at the center of our analysis. In more detail, we analyze three aspects of selective revealing: the way in which it is practiced, its level, and the change of this level over time.

3.2 Research design

In order to gather a complete understanding of the phenomenon in its context, and to ensure well-founded conclusions, we created a sequentially-ordered, mixed-methods research design (Edmondson and McManus, 2007; Jick, 1979).

First, we identified a population of 267 firms supposedly manufacturing, for Linux, one of the three types of embedded components mentioned above. To this end, we searched websites dedicated to Embedded Linux, trade journals on embedded components, and vendor catalogues. We tried to contact all of these companies by phone, email, or postal mail to identify a suitable respondent. Of the 247 that could be reached (others might, e.g., have ceased to exist), 35 firms indicated that they were not actively developing for the Linux operating system, reducing our sample population to 212 ECMs. Of these, 20 were not willing to take part in the survey, leaving us with 192 firms. With a subgroup of these, plus select customers, we conducted a total of 16 semi-structured interviews (see Section 4.1) inquiring, for example, why companies had started to engage in selective revealing and which advantages customers expected.[4]

---

[4] While Henkel (2006) also looks at Embedded Linux, we would like to emphasize that this study is substantially different. First and foremost, it is a separate data collection effort that is the basis of the second author's PhD dissertation (Käs, 2008). As to a potential overlap, we cannot exclude the possibility that a few firms are in both samples, but this would be a pure coincidence. Second, the revealing decision in the two papers is also substantially different. In our study, there is no legal obligation to license these drivers as OSS, and so the revealing decision is truly voluntary. In contrast, respondents in the study by Henkel (2006) mostly created "derivative work" of Linux and thus were obliged to disclose the source code. Their choice was one of timing only.

Second, we gathered archival data on our entire population of ECMs such as conference presentations, mailing lists, firm websites, industry journals, technical whitepapers, and product catalogues.

Third, based on the data gathered from these two sources and existing literature, we developed a large-scale survey (see Section 5.1), which was subsequently distributed to all 192 ECMs. Thus, qualitative and quantitative methods complemented each other, with interview and archival data used for substantiating a conceptual framework that could then be tested using quantitative methods (Eisenhardt and Graebner, 2007). At the same time, we used the qualitative findings to support the interpretation of, and clarify and illustrate the quantitative analysis based on the survey data (e.g., Jick, 1979). Given the resulting dependence of our survey items on our qualitative findings, we purposefully depart from the standard paper structure to present each study in turn below.

4 Qualitative study

4.1 Data collection and analysis

The qualitative study was designed to provide insights into core constructs and relationships (Edmondson and McManus, 2007) as a basis for a comprehensive conceptual frame. We conducted a total of 16 interviews with 16 different firms of different size and from three different continents by phone between November 2005 and November 2006, eight each with ECMs and device manufacturers, the core customers of most ECMs. Interviews were conducted with senior managers or heads of software development, or IP managers who knew about the first (potential) involvement in selective revealing. These interviews lasted between 16 and 55 minutes and were in most cases (13) tape-recorded and transcribed verbatim.

Next, we analyzed the interviews using qualitative content analysis (Mayring, 2004), at the core of which is the development of a coding scheme (or 'system of categories'). Data was coded using a combination of deductive and inductive approaches. We started by deductively creating, prior to fieldwork, a provisional list of codes. The interviews were analyzed and coded on an ongoing basis, to allow for refinement of the interview guide and refocusing (Miles and Huberman, 1994, p. 65). Consistent with this procedure, the codes were continuously revised during analysis. Finally, all interviews were again checked against the final coding scheme.[5]

4.2 Findings

The qualitative study uncovered a process that has led the industry from a proprietary stance on IP to high levels of selective revealing. We learned about newly articulated customer preferences creating a trigger and demand-pull for openness, cognitive inertia as an impediment, and benefits to the component makers as the result and, later, an additional driver of openness.

4.2.1 Newly articulated customer preferences

Before Linux made inroads into the embedded market, proprietary operating systems such as Windows CE or VxWorks and various manufacturer-specific types were dominant. Customers would not expect to get access to the driver source code, mainly because they accepted the ECMs' argument that those could not reveal the sources due to third-party IP restrictions. Thus, even though source code access had always been desirable for customers, these preferences were rarely articulated. Without pressing demand for source code, in turn, ECMs preferred to stay on the safe side with respect to their IP and kept their code secret (and still do for proprietary operating systems):

> *"Windows customers certainly don't expect to get the source code. So we will do what is appropriate for the given operating system." (US ECM [Interview 4])*

[5] The full coding scheme is available from the authors upon request.

Partly, however, this seems to have been a convenient excuse for ECMs for not exploring the possibilities for openness. At least in some cases, providing source code access is also possible for proprietary operating systems:

> *"Sometimes, e.g. when we base the development of a board support package on the sources of the provider of a proprietary operating system, we need to check back with providers if they agree with this [providing the source code to customers], and to reassure that we do not infringe on third-party rights."(European ECM [Interview 8])*

With the advent of Linux, things changed (1) because the argument of third-party IP being infringed often no longer applied and since (2) with Linux, source code availability was *expected* (though not mandatory). Thus, many customers approached ECMs with a newly articulated preference for open driver source code, creating a *demand-pull for selective revealing*:

> *"If you are looking back five years prior to now, when we did not have all of our drivers open source, no doubt our customers were doing exactly that [call and require OSS drivers] (...). They were probably ahead of us at this time at understanding OSS and how they wanted to deal with it." (US ECM [Interview 4])*

These requests forced ECMs to weigh the pros and cons of openness explicitly and take a conscious decision for or against it. Linux, therefore, served as a trigger for rethinking the established practice of closed source drivers. Specifically, given explicit customer demand for openness, many ECMs needed to adjust their estimation of the advantages and disadvantages to factor in openness as a newly emerging purchasing criterion voiced by customers. In turn, ECMs voiced how with increasing customer demand for source code,[6] openness through selective revealing attained more and more competitive importance:

> *"If we wouldn't do it [provide driver source code] they [our customers] would be on the*

[6] The reasons for why customers demanded more openness are consistent with our earlier statements. Our interviewees most often voiced issues such as customizability, the ability to fix bugs, and vendor independence.

phone in five minutes." (US ECM [Interview 4])

Interviewer: "What would you say is the most important reason for your firm to have an open source approach, particularly regarding drivers?"
Interviewee: "The business reason is that customers are demanding it." (US ECM [Interview 10])

4.2.2 Inertia and learning

The increasing demand for driver source code led component makers to increasingly appreciate the merits of openness, but even with a positive evaluation further barriers had to be overcome. Our interviews showed that existing cultures and corresponding organizational processes slowed down the change toward openness. Similar to earlier findings (Alexy et al., 2013b) on inertial forces against the adoption of open source software development, such cultural issues were often the results of preceding antagonistic socialization and learning processes.

However, once "pulled into" selective revealing by external pressure, ECMs often went through a positive feedback loop. Many ECMs reported to reveal more today than some years ago because after initial exposure, they had successfully gone through a learning process, and gradually adapted their culture and processes toward openness. This led them to re-evaluate the potential technical merits and become less worried about potential downsides:

"As a member of the community we have learned how the community works, and got a lot better educated than we were five years ago. And we have also come to realize that doing them in open source [i.e., develop drivers collaboratively with the community in an OSS approach] generally works a lot better. You get more people contributing to the code; you have a much easier time of getting the driver into the kernel [i.e., the core of Linux] and into the distributions, which is really important. It cuts down on our costs. So there is definitely an economic argument to that as well. (…) Overall it is just a learning process that a company goes through" (US ECM [Interview 4])

It is important to see how this feedback loop is simultaneously fed by companies' learning as well as external pressures, as shown by the statement below. Notably, given continuously growing consumer demand for openness, ECMs were also pressured into selective revealing so as to not lose out against the competition:

> *Informant: "(...) Linux-specifically we actually open source a lot, and the tendency is to releasing more and more."*
> *Interviewer: "And what would you say is the main driver behind that?"*
> *Informant: "I think one thing is basically the company culture. It takes time for people who come from that closed way of working, keeping everything proprietary, to a more open way of working and learn about the benefits there. So, I think that is something where [we], but also other companies, have gone through some kind of learning curve. And it is, of course, the general trend in the industry to do this. So if your competition is also doing it, if your customers are asking for it—that helps in the transition." (European ECM [Interview 5])*

4.2.3 Experiences driving the feedback loop

Responding to customer demand for driver source code increased sales and customer satisfaction, as was to be expected. But beyond that, ECMs also reported marketing benefits, technical support, and legal certainty as positive experiences with selective revealing.

Consistent with the literature, marketing benefits related to reputation and visibility. Firms were highly cognizant of the advantages of being considered a good OSS player. Yet again, companies were rather led by observation they made competing for customers, rather than an internal assessment, in coming to believe for example in word-of-mouth effects of open source:

> *"I heard quite often that people saw other examples where companies had a real bad reputation for not open sourcing and [we as a company] then said: 'We really don't want to get that kind of bad publicity,' and that was a reason for open sourcing."(European ECM [Interview 5])*

Also in line with earlier findings, technical benefits for the vendor were said to originate from external development support, testing, bug fixing, adaptation to changing underlying platforms, and maintenance. Consistent with West and O'Mahony (2008), we learned that whether these benefits were indeed attained crucially depended on how selective revealing was practiced. Some

ECMs only offered source code for download on their website, with no intended interaction with customers—a behavior commonly labeled 'code without community' by the OSS community (first by Asay, 2007). Other ECMs used selective revealing to engage with customers, either by participating in existing OSS projects or by starting public OSS projects themselves. These latter formats were described as providing a novel setting for active co-development with external agents:

> *"A big part of dealing with open source drivers and code is more just interacting with all the other people that are participating in the movement, and not just dumping out source code. [...] I suppose it could be said that the software development process also ought to be opened. You are continually interacting with the community, and they can see what you are developing and you can see where the kernel is going because it is shifting." (US ECM [Interview 10])*

Extending this point, we found that customers' increased independence of the ECM with respect to code changes seemed to go along with a transformed and in some respect closer relationship. We found that selective revealing became a precursor of richer, more intense interaction between both ECMs and customers:

> *"Therefore we had positive experiences from providing the source code. It is fairer for both parties and, therefore, is well received. It is more of a true partnership—and this is what has lately become more important." (European ECM [Interview 9])*

However, at the same time, reports from customers made clear how these newly emerging collaboration opportunities often presented significant challenges to ECMs:

> *"This is where [the ECM] needed to learn, for sure (...). It was a new world for them. In the beginning, they really did not feel very comfortable (...) There are coding standards and some minimum requirements on how to interact with the community: whom can I contact, whom not; which is the right tone for communication with those people. There really is a lot to learn." (Device manufacturer [Interview 2])*

Finally, legal aspects played a role for some ECMs. Open source licenses and in particular the

GPL made firms reveal driver source code in order to circumvent any potential conflicts with the license, or to avoid any effort associated with avoiding such conflicts. Yet, the sense of a legal obligation was more often than not due to misinterpretations of OSS licenses.

Some firms also reported negative experiences with and perceived downsides of openness. Most prominently, firms noted how revealing drivers that contained differentiating IP would have implied a loss of competitive advantage. Firms that had not had experience with OSS sometimes merely decided against it for the fear of revealing too much and losing critical IP, or, as put by one interviewee: *"Management tends to be overcautious"* [Interview 11]. Yet, firms understood that selective revealing, as implied by the name, was not an all-or-nothing strategy, and adapted their behavior given competitive circumstances. For example, when asked what his company does if a driver contains competitive IP, one informant answered:

> *"In such a case we will provide the driver only as binary module. It needs to be loaded during runtime and thus does not fall under GPL. The source code to the driver can be revealed to customers if they sign a Non-Disclosure-Agreement." (European ECM [Interview 8])*

Other examples given to us were in line with Henkel and Baldwin's (2011) concept of IP modularization, that is, structuring the software artifact in a way that IP concerns are minimized.

4.3 Summary of the qualitative findings

Our interview results highlight an evolving, industry-wide trend of ECMs converting toward selective revealing that is driven by two factors, demand pull and positive feedback, resulting in openness moving from a dormant opportunity to becoming a novel competitive dimension.

Before the advent of Linux as an embedded operating system, it was industry practice for ECMs to provide only the binary code of their drivers, the main justification being that the code contained proprietary IP of the operating system vendor. Revealing driver source code mostly did not happen, and customers contented themselves with the compiled driver code. With the industry locked in to a proprietary stance on IP (in Figure 1, the top left box), opportunities for open innovation lay dormant—apart from isolated instances of source code disclosure under NDAs (IPR-based open innovation, bottom left box in Figure 1).

When Linux gained popularity in embedded systems, established ECMs initially treated it as just another operating system. Most of them were aware that providing binary drivers only was accepted practice, despite the GPL, and they used this discretion to keep their source code secret. Their customers, however, expected Linux drivers to be licensed as OSS, and requested the driver source code. The more ECMs complied with this request, the more confidently customers demanded the sources. Linux had created a demand pull for openness, and this customer demand served as a trigger for ECMs to rethink their industry practice of strong IP protection. Put differently, when customer demand had overruled earlier considerations of risks and costs of opening up, many companies actively started exploring for ways to benefit from this largely unavoidable, externally-enforced change.

Then, a positive feedback loop set in, as illustrated in Figure 2: firms made mostly positive experiences when they revealed driver source code; felt pressure from competitors practicing openness; and learned about how to best implement selective revealing through both experience and imitation. As a result, they further increased their degree of openness to a level that a few years earlier would have seemed impossible. Yet, some firms do not reveal any of their drivers, which is perfectly rational if the component is a commodity and the bundle's differentiation resides in the driver.

--- *Insert Figure 2 about here* ---

Finally, beyond price, quality and marketing, openness emerged as a *new dimension of* *competition* in this market. Akin to the evolution of disruptive innovations (Christensen, 1997), customers of ECMs changed their conception of what were the relevant dimensions of product performance—for these customers, openness became a relevant purchasing criterion. Some ECMs then strategically exploited this shift in consumer demand and promoted openness as a new performance criterion. Notably, performing well on the new openness dimension was attractive not only, but in particular to those firms whose technology was not at the leading edge of the market, since they faced a lower risk of losing technological differentiation. Illustrating this logic, Shankland (2006) describes how Intel tried to establish a competitive edge in graphics over its fierce rivals ATI and NVIDIA not by technological superiority but by driver source code availability.

In sum, customer demand fundamentally changed the evaluation of openness of many ECMs, who in turn changed their ways of doing business. The positive experiences they made set in motion a positive feedback loop for even more openness, leading to openness becoming a new dimension of competition at the level of the industry.

5 Quantitative Study

5.1 Survey design and variables

5.1.1 Sample

To corroborate and deepen the insights garnered through our qualitative study, we developed a survey instrument based on extant literature as well as our qualitative findings, and sent it to the sample of 192 ECMs described above between November 2006 and March 2007. Thus, the 74 answers we received are equivalent to a 39% response rate (or 35% if related to the larger sample of 212 ECMs). We tested for differences between our respondents and the remaining population regarding size, geographical origin, or subsector, but found them to be insignificant. Because of incomplete responses, the actual number of observation underlying our regression analysis is 67. Of these, 38 supply mainly single board computers (SBCs), 14 embedded processors (or "chips"), and 12 data acquisition cards (DAQs). Three have no clear focus; they are merged with the SBC firms to form the reference category.

5.1.2 Dependent variables

Our core questions relate to (1) whether or not ECMs selectively revealed at all and (2) if so, what share of their software they disclosed. To solicit our dependent variables, we thus asked ECMs about the share of driver source code that they made publicly available as OSS. Following the insights we had garnered through our interviews, we asked for both the share of source code revealed in general and the share revealed by contributing it to (existing or newly started) public OSS projects. From this information, for measuring whether companies engage at all in either type of revealing, we constructed dummy variables that take the value of 1 if the company reports a degree greater than 0. To measure the actual degrees, we take the percentage information reported by companies. The fact that these variables are single item measure should not be of concern, since we are trying to capture a concrete and measurable phenomenon that is known specifically to the survey respondent (Bergkvist and Rossiter, 2007; Rossiter, 2002).

Moreover, we asked ECMs to specify the extent of revealing for different years (2000, 2003, 2006), to make inferences about the existence of a potential trend. However, the number of firms (35) that provided data for 2000 and 2003 was insufficient to exploit intertemporal data for our analysis. Accordingly, our dependent variables relate to the year 2006. Still, t-tests about the degree of revealing over time by these 35 firms suggest a clear upward trend. Survey responses by these firms as to why they increased the share of revealing provide further insights.

5.1.3 Independent variables

Our qualitative study and existing literature suggest six main independent variables. Three of these capture the importance of *reasons* to reveal (which, for firms that did not reveal at all, are potential reasons): customer pressure, and marketing-related and technical benefits. The other three capture firm-internal determinants of openness: experience with embedded Linux, developing Linux drivers since the firm's foundation, and experience with selective revealing of driver source code.

Regarding the reasons for selective revealing, we asked respondents to rate its perceived advantages on a five point Likert scale ("strongly agree" to "strongly disagree"). Conducting Harman's one factor test suggests that common method bias should not be a concern (Podsakoff et al., 2003; Section 5.1.4. also addresses this issue). To measure *customer pressure*, we included a single item measure asking whether a reason for revealing was that "our customers expect to get access to the source code." *Marketing-related benefits* were calculated from the factor score of four items, after factor analysis confirmed unidimensionality and reliability was assessed positively. The measure of *technical benefits*, consisting of six items, was constructed in the same way. Details on both variables and their underlying items are provided in Table 1.

The first variable to capture firm-internal determinants of selective revealing is *experience with embedded Linux*. Here, firms self-reported the year in which they began selling products for the Linux market, which we validated through archival checks whenever possible. Second, to understand whether a company was *founded in order to possibly leverage the potential of selective revealing*, we constructed a dummy variable taking the value of one if the firm reported to have engaged in embedded Linux within a year of its founding, which is the case for 12 out of 67. We validated it through archival cross-checks, which we could successfully perform for 86% of the responding firms. Our idea here was that firms newly founded in the context of Embedded

Linux should have no form of lock-in into past ways of working, and thus face less inertial pressures against selective revealing by default. Finally, we used information on past engagement in selective revealing (see 5.1.2) to construct our measure of *revealing experience*. Firms that reported to have engaged in OSS only since 2006 or not at all received a value of 0 for this variable, between 2003 and 2006 a value of 1, between 2000 and 2003 a value of 2, and before 2000 a value of 3.

--- *Insert Table 1 and Table 2 about here* ---

We attempt to reduce the risk of spurious effects by including several further variables. In particular, we control for whether companies (erroneously) felt *legally obliged* to release their source code. We measure this factor by asking for agreement, on a 5-point Likert scale, whether compliance with the General Public License, the OSS license governing the use of Linux, was a reason why they considered engagement in selective revealing. Moreover, we control for *firm size* using dummy variables representative of different size category related to the number of employees. Finally, we use dummies to control for the respondent's *embedded components subindustry* in order to filter out potential baseline differences between these. We use SBCs as reference category. Descriptive statistics and correlations for all variables are given in Table 2.

5.1.4 Common method bias and robustness checks

A shortcoming of our survey instrument is that dependent and independent variables originate from the same questionnaire. In order to minimize potential biases (Podsakoff et al., 2003), we took three precautions. First, for the design, we relied on established items whenever possible to increase reliability. Questions were further informed by the qualitative interviews to guarantee applicability and realism. In addition, we pretested the survey extensively with the help of ten industry experts, participants, and academics familiar with the topic. The pretests confirmed the suitability of the survey and led to minor rewordings.

Second, regarding administration, our procedure of identifying the right agent inside the firm to answer the survey further decreases the risk of common method bias, since they should be capable of answering most questions without reverting to guessing or giving socially desirable answers. In addition, we emphasized anonymity of results to minimize apprehension bias.

Third, regarding our results and the conclusion we draw from them, we triangulate the survey results with two other sources of data: the interviews we conducted, and product catalogues of selected firms contained in our sample. While only limited documentation on drivers is available, we could gather for 150 of the 267 firms in the frame population a qualitative assessment of whether the firm does reveal drivers or not. Sixty-four of these 150 firms were among our 74 respondents, and for these the qualitative assessment was consistent with their survey response.

5.2 Descriptive results

In short, our survey results are fully in line with those of the qualitative study. From 2000 to 2006 (see Table 3), the number of firms in our sample that provided drivers for Linux increased from 35 to 67, reflecting the growing popularity of Linux. At the same time, the share of firms revealing at least some drivers (on their website or to an OSS project) in the respective year increased from 60% to 69% (row c in Table 3), and the average share of drivers revealed (including those firms that revealed none) from 52% to 57% (row d). The increase becomes more pronounced when focusing on cohorts: among the firms active in 2000 (row e) the share went from 52% to 61%, and among those active in 2003 but not in 2000 from 42% to 56% (row f). For the share of code revealed to OSS projects, we see a considerable increase (from 22% to 29% and from 1% to 21%, rows i and j) when looking at cohorts. The overall average, in contrast, does not increase since each group of newcomers to Linux drivers has, in the respective year, a negative effect on the average.

--- Insert Table 3, Figure 3 about here ---

Analyzing the firms' revealing behavior in 2006 more closely, we find that the level of openness takes on all values between 0% and 100% (see Figure 3). Roughly one third of the firms (21) reveal no source code at all, another third (22) reveal an amount between 0% and 100%, and the rest (24) reveal all their drivers. This bimodal distribution is the natural result of censoring rather than an indication of two distinct groups.[7]

Asked about the reasons for revealing driver source code, respondents confirmed the important role of customer demand identified before. We asked firms that revealed more in 2006 than in 2000 for their reasons to do so (on Likert scales), and "increased interest by our customers" and "positive experiences w.r.t. customer satisfaction" received the highest shares of agreement (57% and 59%, resp., "agree" or "strongly agree") after "this has been a learning process [for] my company" (68%). (The average levels of agreement to all reasons for revealing related to marketing and technology are provided in Table 1.) Even 80% of all respondents agreed or strongly agreed that "availability of source code is a valuable marketing argument."

5.3 Regression analysis

We conducted two analyses to account for different types of selective revealing in line with the findings of our qualitative analysis. The first set of regressions relates to selective revealing in the form of *both* putting source code up for download or contributing it to public OSS projects;

---

[7] A linear prediction (using "predict" in STATA with the option "xb") of the degree of revealing after running a Tobit regression (see next subsection) yields a variable that is roughly normally distributed with a mean of 60 and a standard deviation of 89. The extreme points are -196 and 236 and there is no indication of a bimodal structure. Thus, the bimodal distribution of the observed degree of revealing is purely due to the censoring at 0% and at 100%, and no indication of specific antecedents driving the openness choices.

the second, to contributions to public OSS projects only.[8] For both types of revealing, we first analyze the decision to reveal at all using Probit models, and then the degree of revealing using a Tobit regression and a Probit regression (distinguishing between less than 100% and 100% of revealing).[9] Stepwise inclusion of variables suggests that multicollinearity should not be a concern (Maddala and Lahiri, 2009: Ch. 7).

We also use stepwise deletion to ensure that our results are not affected by the relatively small numbers of observations, which is particularly important for our second set of estimations. We find that our core results remain unchanged when eliminating variables from the regression.

5.3.1 Revealing in general

Results of multivariate analysis of the level of openness are shown in Table 4. They pertain to revealing in general, i.e., on the firm's own *website or to public OSS projects*.

As specification (1) shows, customer demand for source code has a significant (5%), positive effect on the decision to reveal at all. Given that selective revealing takes place (specifications 2, 3), customer pressure has no significant effect on its extent. Experience with embedded Linux has a negative and revealing experience a positive effect (both at the 1% level in (2), and 10% and 5%, respectively, in (3)) on the share of code that is revealed, throughout all specifications.[10]

---

[8] An analysis of "revealing to the firm's website" only would make little sense since the reference group would be highly heterogeneous (code not revealed to a website could either be not revealed at all or revealed to an OSS project). Furthermore, code revealed to an OSS project will often simultaneously be revealed on the website.

[9] This two-stage decision process suggests using Heckman regressions, since the decision to reveal at all might partly be driven by other factors than the decision of how much to reveal. We thus ran Heckman regressions with perceived legal obligation to reveal (due to the GPL) and customer pressure as potential identifying variables in the first stage. However, the former was insignificant in both stages, while for the latter it is theoretically unclear why it should have no effect on the degree of revealing. Also, in no case could we reject the Null hypothesis that the error terms in the two equations are uncorrelated (i.e., $\rho = 0$). We thus report separate regressions for the first and the second stage.

[10] The result that "experience with embedded Linux" is negatively related to the extent of revealing may appear counterintuitive. However, note that a firm's experience with embedded Linux is at least as long as its experience with revealing, and the latter has a large and significant positive coefficient. The more firm experience with

The likelihood that a firm reveals all of its drivers (3) is positively related to the importance of technical benefits (1%) and, interestingly, negatively to the importance of marketing benefits (5%). Somewhat surprisingly, a foundation imprinting effect (i.e., if the firm did embedded Linux since its foundation) can be found neither for revealing at all, nor for its extent.

--- *Insert Table 4 and Table 5 about here* ---

In short, the above results corroborate the insights gained in our qualitative study. First, we find evidence of customer demand as a trigger to reveal at all. Second, the degree of revealing appears to be driven largely by perceived technical benefits and experience with revealing, reaffirming our notion of a positive feedback loop.

5.3.2 Revealing to public open source projects

Analyzing the revealing to public OSS projects (specifications 4 to 6 in Table 4) yields partly similar, partly different results compared to revealing in general. For both types of revealing, perceived technical advantages and revealing experience are positively, and marketing concerns, negatively related to the extent of revealing (specifications 2, 3, 5, 6).

Of particular interest are the differences we observe. In contrast to regressions (1) to (3) analyzing revealing in general, customer pressure is not related to the decision to reveal to OSS projects (4), while experience with embedded Linux is, and "founded in context" is significant (5%) in specification (5) estimating the degree of revealing to OSS projects.[11] All three findings suggest that external factors are less, and firm characteristics more important for this type of selective revealing. Bi-directional openness seems to require a much stronger cultural heritage or

embedded Linux exceeds experience with revealing, the longer the firm has been using embedded Linux in a proprietary fashion, and the more engrained the practice of not revealing source code. This is reflected in the negative coefficient of "experience with embedded Linux."

[11] We had to drop the variable "founded in context" in model (5) to achieve convergence.

change than the simple disclosure of source code on the web. Accordingly, we would conjecture that these firms were set up with the explicit goal of exploiting the technical opportunities of participating in open source projects.

6 Discussion

6.1 Implications for theory

Our analysis paint a fairly consistent picture of the emergence of selective revealing. It enters the industry via customer demand, proliferates through positive feedback loops, and eventually becomes a new dimension of competition. These insights allow us to make three contributions to the ongoing discussion on open innovation and selective revealing.

First, our findings clearly suggest how considerable external pressure may be required so that companies initiate such radical re-evaluations of how they create and capture value. Put differently, it seems that the interaction with the outside world that constitutes open innovation is also instigated by the outside world. Specifically we present downstream customer demand for openness as a trigger of companies' initial opening up. We argue that external pressure in favor of selective revealing shifted the cost-benefit analysis preceding the decision to selectively reveal, literally overwriting concerns about risk and the power of inertia. Underlying this pressure are the coupled dynamics of technological opportunity and market pull (Kahl, 2006; Tripsas, 2008): The advent of Linux and the collaborative OSS development model caused changes in revealed customer preferences (up to the point that some customers made source code availability a purchasing criterion). Together, they triggered a discontinuous technological and organizational change (Tripsas, 2008) fruitfully enabling open innovation. Importantly, the type of open innovation that emerges is based not on IPRs but on selective revealing (Figure 1), providing considerable advantages with respect to transaction cost. Notably, what we observe may reflect a general change in the role of customers, who are increasingly demanding a more active role and strive to become co-creators of value. The rising importance of user driven innovation and crowdsourcing (e.g., Afuah and Tucci, 2012; Baldwin and von Hippel, 2011) bears witness to this change, suggesting that other industries may be prone to undergo the processes we have described herein.

Second, the distinction between selective revealing by putting code on a website versus joining or launching a public OSS project that first emerged in our qualitative analysis allows us to gain further theoretical traction. Specifically, selective revealing has been argued to be a potential first step toward more intensive collaborations with externals (starting with Raymond, 1999). Our results suggest that indeed, contributing code to a public OSS project or launching such a project facilitates open innovation in the sense of informal collaboration between the seller and the buyers, often resulting in considerable development support for the seller. Put differently, at first glance revealing code to public OSS projects and receiving code improvements back may appear like a combination of outbound and inbound open innovation. However, our analysis shows that what results is open collaborative innovation (Baldwin and von Hippel, 2011)—informal collaboration between two parties in an open innovation format, made possibly by the selective revealing of IP—to the benefit of all corporate actors involved. The intensified interactions between customers and suppliers give rise to more meaningful relationships that resembled strategic alliances or other forms of "true partnerships."

This finding further suggests that different types of open innovation may not only feature operational, but also qualitative differences. In particular, our findings raise the question under which circumstances open collaborative innovation, given it can be implemented, would be superior to a simple outbound process, and call upon future research to inquire this matter. In this context, we also highlight how companies founded in an environment supportive of openness, or even specifically founded to leverage openness do not need to overcome the strong inertial forces against this particularly extreme form of open innovation. In our context, ECMs that developed Linux drivers since their foundation should embody a much more positive view on the possibilities inherent in selective revealing to OSS projects. Indeed, we find that the former firms revealed on average 46% of their driver source code to OSS projects as compared to 17% by firms that took up embedded Linux later.

Finally, our findings clearly highlight effects of selective revealing on competitive dynamics, with selective revealing as a means to differentiate one's offering from those of competitors or, in turn, revealing by competitors as a reason for the focal firm to practice it. Beyond price, quality and marketing, we found that openness emerged as a new dimension of competition in this market. Yet, if that is the case, we can draw on theories of competitive dynamics to ponder about potential implications of such a development. Specifically, we may elaborate on a potential linkage to Christensen's (1997) concept of disruptive innovation (discussed earlier, e.g., by Alexy et al., 2013a; Käs, 2008). Openness, so we find, may represent an important case of a new performance dimension valued in the nascent market of embedded Linux.

This finding may appear contrary to the insights of West (2003), who showed how openness usually weakened the firm's competitive position in the focal market, for which it had to compensate in another market. These viewpoints may be reconciled by recognizing that, in our case, there is just one market. Drivers by themselves are useless as products; they are strictly complementary to the corresponding component. Both together must be seen as one product, and revealing the driver amounts to disclosing the blueprint of a part of that product. Accordingly, those ECMs whose source of differentiation lies with the component rather than the driver or whose product overall is not leading-edge will not lose much technology-based competitive advantage through selective revealing. At the same time, given increasing customer demand for openness, selective revealing has a positive effect on these firms' competitive position vis-à-vis competitors that did not disclose their code. Intel's move to make its graphics card drivers OSS to better compete with the (higher performance) offers of ATI and NVidia (Shankland, 2006) illustrates this logic.

In this vein, we strongly feel that future research should look into what characteristics render a market particularly prone to the developments we observe. In addition, if increasing openness is found to regularly be the result of more openly articulated customer preferences, future research may look into how to identify the right customers, a question of general interest to research on disruption.

6.2 Boundary conditions, limitations and suggestions for future research

Our study faces a number of limitations. To start with, our quantitative analysis rests on a cross-sectional analysis of 67 observations. In addition, the industry in which our study is set may feature idiosyncrasies limiting the generalizability of our findings. In particular, we did not encounter a firm that entirely abandoned selective revealing after having adopted it, yet it seems plausible that such cases should occur. While we have tried to be explicit about these issues and to address them to our ability, some concerns may remain, which may at the same time offer significant opportunities for future research.

At the same time, our study should have external validity for industries bearing the following characteristics: (a) stout IP protection, since it implies a potential for increased openness; (b) fragmented markets, which imply high transaction cost of individual NDAs (Henkel and Baldwin, 2011); and (c) strong needs for downstream customization and thus (latent) customer preferences for openness. While these may seem restrictive conditions at first sight, earlier work on selective revealing has shown how these conditions may be present in industries as diverse as synthetic biology, furniture, the built environment, or agriculture (Füller, 2010; e.g., Henkel and Maurer, 2009; Kloppenburg, 2010). Also, this literature makes clear how the arguments we have put forward on data from 2006 and earlier are still fully relevant to today's business environment in which many industries remain largely "closed," and which, accordingly, are likely to undergo the same processes we have documented at some point in the future. We suggest that future research specifically target these as opportunities to enrich our understanding of how and why industries and firms within these become more open over time, and which factors moderate this process.

6.3 Implications for management

For practicing managers, our results are further evidence of the fact that open models of innovation need to be put on the radar screens of top-level management. In particular our proposed link to the concept of disruptive innovation challenges managers to be on the lookout for customers demanding and competitors implementing open models of innovation. It also raises questions about how to respond to this issue through organization design, and insights from work on disruptive innovation may suggest that the right response would be to install a separate organizational unit tasked with this challenge. Finally, our findings highlight a new competitive weapon for managers in companies that are not technology or market leaders. By betting on being open, some of these companies may be able to significantly increase their competitiveness.

Tables and Figures

Figure 1: Combinations of secrecy and legal exclusion rights

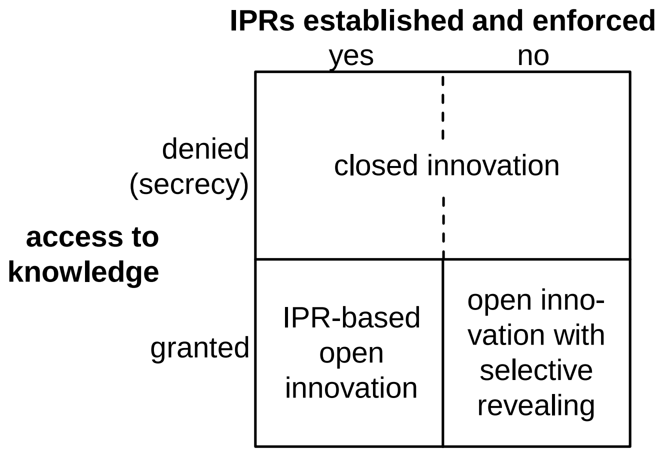


Figure 2: Histogram of share of revealed driver source code

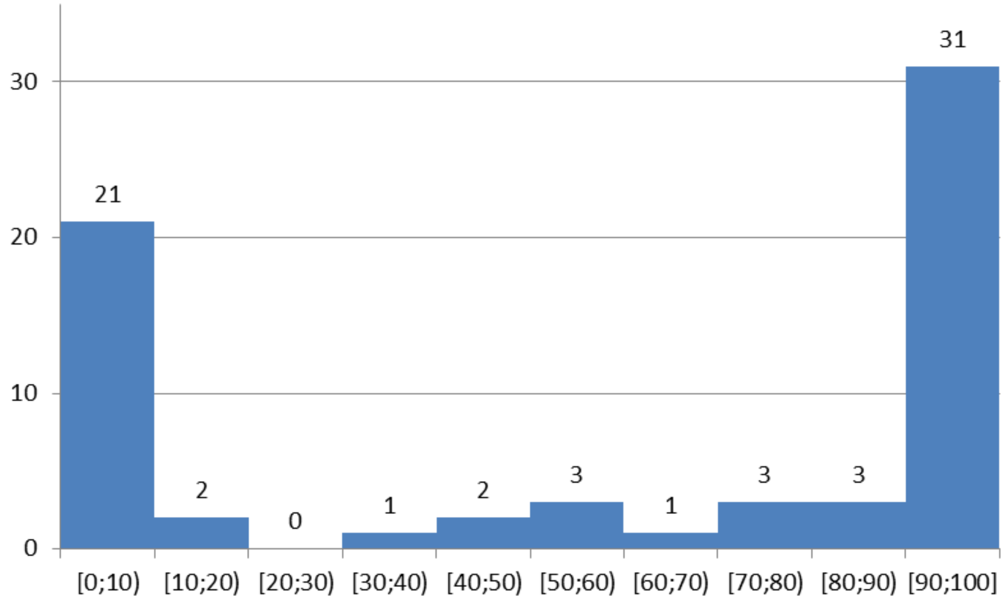
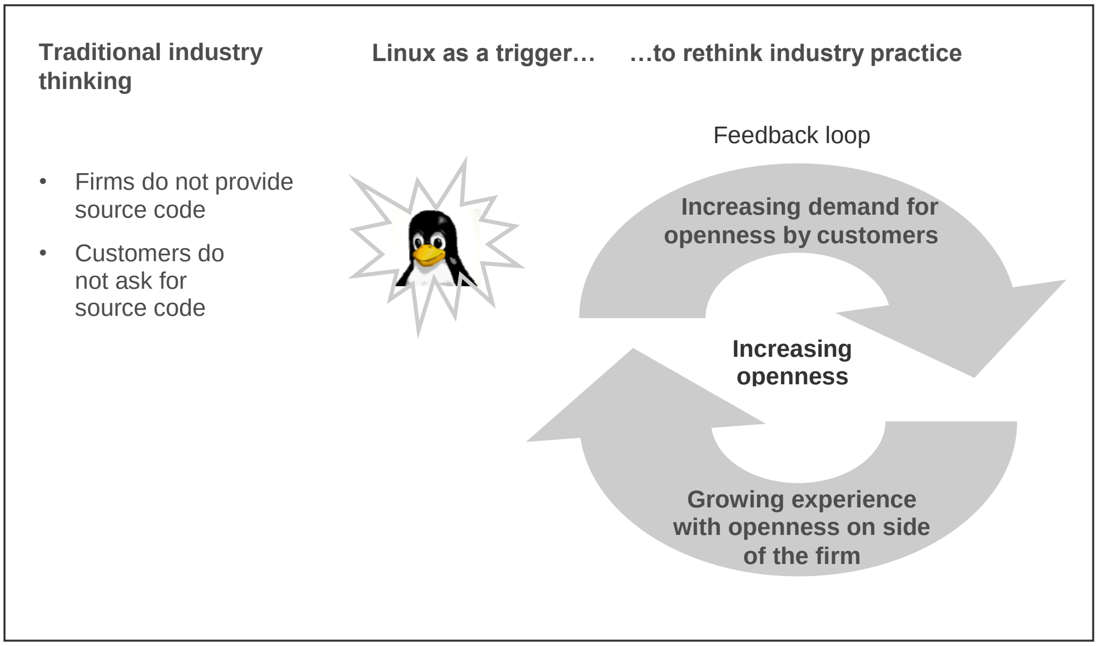


Figure 3: Linux as a trigger to rethink industry practice



Marketing reasons (Cronbach's alpha: 0.71)	*factor loading*	*mean*
It opens more doors for our hardware because it allows for more custom solutions.	0.76	3.70
Cooperation with the open source community is good marketing.	0.68	3.59
We want to appear as a good player in the open source community.	0.65	3.83
Revealing good code is a signal of quality for our products.	0.81	3.59

Technical reasons (Cronbach's alpha: 0.78)	*factor loading*	*mean*
We get better and faster testing and debugging.	0.80	3.38
It allows third parties to develop complementary software and hardware more easily.	0.71	3.46
It reduces our maintenance effort.	0.70	3.17
This way, our products stay compatible to other products.	0.70	3.14
Others develop the code further and reveal their developments in return.	0.74	3.36
We often do not have sufficient resources to make all developments on our own.	0.49	2.92

Table 1: Items underlying market-related and technical benefits questions

	1) Degree of revealing	2) Customer pressure	3) Marketing benefits	4) Technical benefits	5) Experience with embedded Linux	6) Founded in context	7) Revealing Experience	8) Perceived legal obligation	9) Large company	10) Medium-sized company	11) mainly chips	12) mainly DAQ cards
1)	1											
2)	0.193	1										
3)	0.292	0.264	1									
4)	0.238	0.279	0.616	1								
5)	-0.025	0.106	0.199	-0.019	1							
6)	0.116	-0.054	0.225	0.147	-0.076	1						
7)	0.679	0.324	0.421	0.19	0.416	0.08	1					
8)	-0.009	0.088	0.185	0.035	0.054	-0.418	0.004	1				
9)	0.045	-0.003	-0.073	-0.016	0.307	-0.327	0.157	0.191	1			
10)	-0.047	-0.1	-0.023	-0.123	-0.158	-0.099	-0.13	0.037	-0.731	1		
11)	0.21	0.044	0.178	0.069	0.231	0.047	0.257	0.156	0.422	-0.244	1	
12)	0.006	-0.137	-0.147	-0.103	0.197	-0.218	-0.054	-0.047	0.005	0.057	-0.24	1
Mean	57.39	3.78	0	0	6.42	0.18	1.85	3.71	0.33	0.52	0.21	0.18
S.D.	44.32	0.95	1	1	2.59	0.39	0.87	0.98	0.47	0.5	0.41	0.39
Min	0	1	-3.94	-2.85	1	0	1	1	0	0	0	0
Max	100	5	1.92	2.19	16	1	3	5	1	1	1	1

Table 2: Correlations and descriptive statistics for all variables (N=67)

	2000	*2003*	*2006*
(a) Number of ECMs providing Linux drivers	35	56	67
(b) Number of ECMs revealing at least some of their drivers	21	36	46
(c) Share of ECMs revealing at least some of their drivers	60,0%	64,3%	68,7%
Average share of drivers revealed *on website or to OSS projects* among ECMs that provided Linux drivers…			
(d) … in the respective year	52,0%	54,6%	57,4%
(e) … since 2000 or earlier	52,0%	60,1%	61,3%
(f) … in 2003 but not in 2000		45,2%	56,0%
(g) … in 2006 but not in 2003			47,7%
Average share of drivers revealed *to OSS projects* among ECMs that provided Linux drivers…			
(h) … in the respective year	22,0%	17,0%	22,0%
(i) … since 2000 or earlier	22,0%	26,7%	28,7%
(j) … in 2003 but not in 2000		1,0%	21,4%
(k) … in 2006 but not in 2003			1,8%
Number of ECMs selectively revealing drivers *on website or to OSS projects* among firms that provided Linux drivers of…			
(l) … all firms (N=67)	21	36	46
(m) … those releasing code since 2000 or earlier (N=35)	21	24	25
(n) … those releasing code since 2003 (N=21)		12	15
(o) … those releasing code since 2006 (N=11)			6
Number of ECMs selectively revealing drivers *to OSS projects* among firms that provided Linux drivers of…			
(p) … all firms (N=67)	13	17	25
(q) … those releasing code since 2000 or earlier (N=35)	13	16	17
(r) … those releasing code since 2003 (N=21)		1	7
(s) … those releasing code since 2006 (N=11)			1

Table 3: Extent of revealing of driver source code

Independent variables	**Revealing in general**			**Revealing to OSS project**		
	(1)Probit: Revealing	*(2) Tobit: Degree*	*(3) Probit (<100, = 100)*	*(4) Probit: Revealing*	*(5) Tobit: Degree*	*(6) Probit (<100, = 100)*
Customer pressure	**0.372*** (0.188)	-0.827 (7.116)	0.232 (0.308)	0.097 (0.208)	-6.834 (8.716)	-1.027 (0.899)
Marketing benefits	0.368 (0.263)	-17.348 (10.355)	**-0.994*** (0.471)	0.313 (0.264)	**-44.126*** (17.310)	**-2.567†** (1.505)
Technical benefits	0.003 (0.222)	**18.992*** (8.369)	**1.176**** (0.435)	0.301 (0.230)	**42.123**** (9.407)	**2.816†** (1.652)
Experience with emb. Linux	-0.061 (0.076)	**-10.688**** (3.413)	**-0.425†** (0.221)	**0.119†** (0.070)	-4.148 (3.198)	-0.336 (0.355)
Founded in context	0.524 (0.617)	19.184 (17.477)	1.234 (1.072)	0.581 (0.551)	**56.748*** (21.580)	
Revealing Experience		**37.590**** (10.506)	**1.321*** (0.579)		**32.614*** (12.533)	1.739 (1.188)
Perceived legal obligation	0.055 (0.209)			0.033 (0.209)		
Large company		40.637 (24.426)	2.458 (1.532)		**58.729*** (23.324)	
Medium-sized company		**38.154*** (18.768)	**2.838*** (1.297)		**49.800*** (21.293)	
mainly chips	0.765 (0.556)	13.245 (17.468)	**1.273†** (0.731)			
mainly DAQ cards	0.215 (0.476)	**38.762†** (19.892)	0.212 (0.754)	-0.208 (0.490)	0.596 (18.809)	
Constant	-0.928 (1.195)	45.056 (36.497)	**-3.833*** (1.812)	-1.716 (1.241)	-10.200 (41.980)	0.713 (2.184)
Observations	67	46	46	67	25	25
Wald/LR, chi^2	13.9	28.0	28.6	17.0	28.0	13.9
p value	0.0400	0.002	0.003	0.018	0.001	0.016
Pseudo R-squared	0.1584	0.10	0.42	0.19	0.13	0.51

Standard errors in parentheses. ** $p<0.01$, * $p<0.05$, † $p<0.1$. All tests are two-sided.

Table 4: Revealing of source code: Regression results

References

Afuah, A.N., Tucci, C.L., 2012. Crowdsourcing as a solution to distant search. Academy of Management Review 37.

Alexy, O., Criscuolo, P., Salter, A., 2009. Does IP strategy have to cripple open innovation? Sloan Management Review 51, 71-77.

Alexy, O., Dahlander, L., 2013. Managing open innovation, in: Dodgson, M., Gann, D., Phillips, N. (Eds.), Handbook of innovation management. Oxford University Press, Oxford, UK, p. forthcoming.

Alexy, O., George, G., Salter, A., 2013a. Cui bono? The selective revealing of knowledge and its implications for innovative activity. Academy of Management Review 38, 270–291.

Alexy, O., Henkel, J., Wallin, M.W., 2013b. From closed to open: Job role changes, individual predispositions, and the adoption of commercial open source software development. Research Policy 42, 1325-1340.

Arora, A., Fosfuri, A., Gambardella, A., 2001. Markets for technology: The economics of innovation and corporate strategy. MIT Press, Cambridge, MA.

Asay, M., 2007. Open-source code, but is it an open-source business?, (downloaded on August 26, 2013 from http://news.cnet.com/8301-13505_3-9804666-16.html).

Baldwin, C.Y., von Hippel, E., 2011. Modeling a paradigm shift: From producer innovation to user and open collaborative innovation. Organization Science 22, 1399-1417.

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.

Cassiman, B., Veugelers, R., 2002. R&D cooperation and spillovers: Some empirical evidence from Belgium. American Economic Review 92, 1169-1184.

Chesbrough, H.W., 2003. Open innovation: The new imperative for creating and profiting from technology. Harvard Business School Press, Boston.

Chesbrough, H.W., 2006a. Open business models: How to thrive in the new innovation landscape. Harvard Business School Press, Cambridge, MA.

Chesbrough, H.W., 2006b. Open innovation: A new paradigm for understanding, in: Chesbrough, H.W., Vanhaverbeke, W., West, J. (Eds.), Open innovation: Researching a new paradigm. Oxford University Press, Oxford, UK, pp. 1-12.

Chesbrough, H.W., Appleyard, M.M., 2007. Open innovation and strategy. California Management Review 50, 57-74.

Christensen, C.M., 1997. The innovator's dilemma: When new technologies cause great firms to fail. Harvard Business School Press, Cambridge, MA.

Christensen, J.F., Olesen, M.H., Kjær, J.S., 2005. The industrial dynamics of open innovation--evidence from the transformation of consumer electronics. Research Policy 34, 1533-1549.

Clarkson, G., Toh, P.K., 2010. ‘Keep out’ signs: The role of deterrence in the competition for resources. Strategic Management Journal 31, 1202-1225.

Dahlander, L., Gann, D.M., 2010. How open is innovation? Research Policy 39, 699-709.

Edmondson, A.C., McManus, S.E., 2007. Methodological fit in management field research. Academy of Management Review 32, 1155-1179.

Eisenhardt, K.M., 1989. Building theories from case study research. Academy of Management Review 14, 532-550.

Eisenhardt, K.M., Graebner, M.E., 2007. Theory building from cases: Opportunities and challenges. Academy of Management Journal 50, 25-32.

Enkel, E., Gassmann, O., Chesbrough, H., 2009. Open R&D and open innovation: Exploring the phenomenon. R&D Management 39, 311-316.

Fosfuri, A., Giarratana, M.S., Luzzi, A., 2008. The penguin has entered the building: The commercialization of open source products. Organization Science 19, 292-305.

Füller, J., 2010. Refining virtual co-creation from a consumer perspective. California Management Review 52, 98-122.

Hagedoorn, J., Ridder, A.-K., 2012. Open innovation, contracts, and intellectual property rights: An exploratory empirical study, Research Policy Special Issue Conference "Open Innovation: New Insights and Evidence", London, UK.

Harhoff, D., Henkel, J., von Hippel, E., 2003. Profiting from voluntary information spillovers: How users benefit by freely revealing their innovations. Research Policy 32, 1753-1769.

Henderson, B., 2006. Linux loadable kernel module howto: Copyright considerations with lkms, (downloaded on March 21 2013 from http://www.tldp.org/HOWTO/Module-HOWTO/copyright.html.)

Henkel, J., 2006. Selective revealing in open innovation processes: The case of Embedded Linux. Research Policy 35, 953-969.

Henkel, J., Baldwin, C.Y., 2011. Modularity for value appropriation: Drawing the boundaries of intellectual property Harvard Business School Finance Working Paper No. 11-054

Henkel, J., Maurer, S.M., 2009. Parts, property and sharing. Nature Biotechnology 27, 1095-1098.

Jeppesen, L.B., Lakhani, K.R., 2010. Marginality and problem-solving effectiveness in broadcast search. Organization Science 21, 1016-1033.

Jick, T.D., 1979. Mixing qualitative and quantitative methods: Triangulation in action. Administrative Science Quarterly 24, 602-611.

Kahl, S., 2006. Considering the customer: Determinants and impact of using technology, (downloaded on February 7 2008 from http://www.chicagobooth.edu/research/workshops/orgs-markets/docs/kahl-customer.pdf)

Käs, S., 2008. Rethinking industry practice: The emergence of openness in the embedded component industry, PhD Dissertation, TUM School of Management. Technische Universität München.

Kloppenburg, J., 2010. Impeding dispossession, enabling repossession: Biological open source and the recovery of seed sovereignty. Journal of Agrarian Change 10, 367-388.

Maddala, G.S., Lahiri, K., 2009. Introduction to econometrics, 4th ed. John Wiley & Sons, New York.

Mayring, P., 2004. Qualitative content analysis, in: Flick, U., von Kardoff, E., Steinke, I. (Eds.), A companion to qualitative research. Sage, London, pp. 266-269.

Miles, M.B., Huberman, A.M., 1994. Qualitative data analysis. SAGE Publications, Thousand Oaks, CA.

Pisano, G., 2006. Profiting from innovation and the intellectual property revolution. Research Policy 35, 1122-1130.

Podsakoff, P.M., MacKenzie, S.B., Lee, J.-Y., Podsakoff, N.P., 2003. Common method biases in behavioral research: A critical review of the literature and recommended remedies. Journal of Applied Psychology 88, 879-903.

Polidoro, F., Toh, P.K., 2011. Letting rivals come close or warding them off? The effects of substitution threat on imitation deterrence. Academy of Management Journal 54, 369-392.

Raymond, E.S., 1999. The cathedral and the bazaar: Musings on Linux and open source by an accidental revolutionary. O'Reilly, Sebastopol, CA.

Rossiter, J.R., 2002. The c-oar-se procedure for scale development in marketing. International Journal of Research in Marketing 19, 305-335.

Shankland, S., 2006. Intel aims for open-source graphics advantage, (downloaded on March 21 2013 from http://news.cnet.com/Intel-aims-for%20-open-source-graphics-advantage/2100-7344_3-6103941.html)

Simcoe, T., 2006. Open standards and intellectual property rights, in: Chesbrough, H., Vanhaverbeke, W., West, J. (Eds.), Open Innovation: Researching a New Paradigm. Oxford University Press, Oxford, UK, pp. 161-183.

Teece, D.J., 1986. Profiting from technological innovation: Implications for integration, collaboration, licensing and public policy. Research Policy 15, 285-305.

Tripsas, M., 2008. Customer preference discontinuities: A trigger for radical technological change. Managerial and Decision Economics 29, 79-97.

West, J., 2003. How open is open enough? Melding proprietary and open source platform strategies. Research Policy 32, 1259–1285.

West, J., 2006. Does appropriability enable or retard innovation?, in: Chesbrough, H.W., Vanhaverbeke, W., West, J. (Eds.), Open innovation: Researching a new paradigm. Oxford University Press, Oxford, UK, pp. 109-133.

West, J., Bogers, M., 2013. Profiting from external innovation: A review of research on open innovation. Journal of Product Innovation Management forthcoming.

West, J., O'Mahony, S., 2008. The role of participation architecture in growing sponsored open source communities. Industry and Innovation 15, 145-168.

Winter, S.G., 1987. Knowledge and competence as strategic assets, in: Teece, D.J. (Ed.), The competitive challenge: Strategies for industrial innovation and renewal. Ballinger Publishing Company, Cambridge, MA, pp. 159-184.