Available online at www.sciencedirect.com

ScienceDirect

Research Policy 35 (2006) 953–969

research policy

www.elsevier.com/locate/respol

Selective revealing in open innovation processes: The case of embedded Linux

Joachim Henkel*

Dr. Theo Schöller Chair in Technology and Innovation Management, Technische Universität München, Arcisstr. 21, D-80333 Munich, Germany

Received 26 August 2005; received in revised form 27 March 2006; accepted 18 April 2006
Available online 5 June 2006

Abstract

This paper provides a quantitative study ($N = 268$) of patterns of free revealing of firm-developed innovations within embedded Linux, a type of open source software (OSS). I find that firms, without being obliged to do so, contribute many of their own developments back to public embedded Linux code, eliciting and indeed receiving informal development support from other firms. That is, they perform a part of their product development open to the public—an unthinkable idea for traditionally minded managers. Such openness obviously entails the challenge of protecting one's intellectual property. I find that firms address this issue by revealing selectively. They reveal, on average, about half of the code they have developed, while protecting the other half by various means. Revealing is strongly heterogeneous among firms. Multivariate analysis can partly explain this heterogeneity by firm characteristics and the firm's purpose behind revealing. An analysis of reasons for revealing and of the type of revealed code shows that different types of firms have different rationales for openness. Implications for management are that the conflict between downsides and benefits of openness appears manageable. Provided selective revealing is practiced deliberately, the opportunities of open development dominate.

JEL classification: L86; M11; O31

Keywords: Open source software; Embedded Linux; Free revealing; Appropriation; IP protection

1. Introduction

With some simplification, one can describe the traditional view of innovation as taking place entirely within one firm. In contrast to such closed innovation, open innovation processes are characterized as spanning firm boundaries (Chesbrough, 2003). This may mean that technology is treated as a tradable good to be bought and sold on the market (Arora et al., 2001). However, openness in innovation processes can reach far beyond such market-mediated exchange. Under suitable circumstances, firms may make their technology available to the public in order to elicit development collaboration, but without any contractual guarantees of obtaining it.

Open innovation in this sense is the subject of the present paper. I explore the commercial development of open source software (OSS) for embedded systems such as machine controls or VCRs (e.g., VDC, 2004). One of the benefits that firms can derive from using OSS is informal development collaboration (Feller and Fitzgerald, 2002). Realizing this advantage requires that a firm reveals its code to the public—an obvious pre-

* Tel.: +49 89 28925741; fax: +49 89 28925742.
E-mail address: henkel@wi.tum.de.

0048-7333/$ – see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.respol.2006.04.010

requisite for open innovation. Some OSS licenses work in the same direction by restricting means to keep code proprietary. However, these factors pushing for openness conflict with a firm's need to protect its intellectual property. So, how can open innovation be reconciled with intellectual property protection?

Using a quantitative empirical study of embedded Linux ($N = 268$), I explore how firms manage this conflict. First, I clarify that firms, despite the fact that Linux is OSS, indeed have a choice between openness and protection. I then analyze what share of their developments firms make public, what type of code they reveal, and what motivates them to do so. Using multivariate analysis I explore which firm characteristics determine revealing behavior. In particular, I investigate if and under what conditions openness leads to informal development collaboration, i.e., open innovation.

For several reasons, embedded Linux is ideally suited for studying the above questions. First, it is nearly exclusively developed by commercial firms, while hobbyists play only a minor role. Second, the fact that it comes under an OSS license makes firms consider revealing their own developments, which they would probably not even think of doing in the case of proprietary software. Still, they have considerable latitude in either sharing or protecting their code. As a result, openness is a conscious decision. Third, as one of the most widely used operating systems in this field (Webb, 2002; VDC, 2004), Linux is of highest relevance for manufacturers of embedded devices. Such devices, in turn, account for the vast majority of all processors—around six billion in 2002 (Ganssle and Barr, 2003). Hence, studying the innovation process of embedded Linux is not just instructive for understanding open innovation, but has implications for the large (and growing) embedded systems industry.

Central results are the following. Firms are aware of and routinely use various means of protecting their code. However, despite the possibility of protection they reveal on average about half of the code they develop for embedded Linux. The degree of openness turns out to be strongly heterogeneous among firms in my sample. Exploring this heterogeneity using multivariate analysis, I find that the share of its code a firm reveals is far from random. Instead, the analysis indicates rational cost/benefit considerations. In particular, the more important obtaining external development support is as a motive for free revealing, the more code the respective firm reveals. Furthermore, small firms ceteris paribus reveal significantly more, likely because, due to resource scarcity, they expect to benefit more from external development support.

Thus, open and collaborative innovation processes indeed take place. The private-collective model of innovation (von Hippel and von Krogh, 2003) is found to work also in a commercial environment. However, firms practice "selective revealing" so as to minimize competitive losses—and are able to do so while abiding by the applicable OSS license. The patterns of free revealing I find are consistent with profit-maximizing behavior. It thus seems conceivable that OSS, even OSS under the GPL, becomes a standard part of industrial firms' innovative activity. The key is to understand what to reveal and what to protect—i.e., to repartition innovative activities into an open and a protected part in a manner consistent with private profits.

The remainder of the paper is organized as follows. In Section 2, background information is given on firms' benefits and downsides of developing OSS and on embedded Linux. Section 3 presents research design and data. In Section 4, analysis and results are presented. Section 5 concludes with a summary and a discussion.

2. Literature review

The present paper links to four strands of literature: information trading, revealing of user innovations, collective invention, and commercial OSS development. In order of increasingly close relation to this paper's subject, I will briefly review the relevant literature in the following, and will point out in what respect the present paper differs.

2.1. Information trading

Open information exchange between firms often occurs within a dyad of individuals. This phenomenon of "information trading" has been analyzed, among others, by von Hippel (1987) and Schrader (1991) and more recently by Dahl and Pedersen (2004). This literature finds that, despite the lack of formal contractual agreements, the information provider expects her counterpart to reciprocate when, in the future, she in turn requests information. A parallel to revealing OSS code is that in both cases the individual developer holds a gatekeeper position. The important difference, however, is that in a trading situation information is given to one particular recipient only. The provider thus knows if the conditions of acquaintance and mutual trust are fulfilled, which Bouty (2000) identified as preconditions for an interpersonal exchange of strategic resources in her study of R&D scientists. Furthermore, within a dyad a lack of reciprocation can clearly be attributed to one individual, and retaliation (in particular, ceas-