



Networking beyond the software code? an explorative examination of the development of an open source car project

Gordon Müller-Seitz^{a,*}, Guido Reger^b

^a Department of Management, Freie Universität Berlin, Boltzmannstr. 20, 14195 Berlin, Germany

^b Centre for Entrepreneurship and Innovation at the University of Potsdam, University of Potsdam, August-Bebel-Straße 89, Haus 1, 14482 Potsdam, Germany

ARTICLE INFO

Keywords:

Open innovation
Open source software
Project networks
Modularity
Motivation
Automotive industry

ABSTRACT

At present, several initiatives have emerged that claim to be innovative while acting according to the mechanisms of open source software (OSS), a field frequently deemed to be a role model for open innovation. Against this background, this study focuses on a case study of the development of an automobile. Based on a commons-based peer production-informed perspective, we show that this project displays a variety of characteristics that are usually associated with OSS projects. In particular, parallels can be drawn between the intrinsic and extrinsic motivations, the ability to 'broadcast' ideas due to the virtual nature of the tasks, and the self-selection of tasks due to their modular nature. The drawing of such parallels, however, must be done cautiously because diverse factors, such as opportunity costs, regulations, and feasibility studies, limit the applicability of OSS principles to this non-software related network of dispersed voluntary contributors within a commons-based peer production framework. Herein, we attempt to clarify how OSS projects can and cannot work as role models for open innovation in the automotive as well as other product-oriented industries.

© 2010 Elsevier Ltd. All rights reserved.

It is the goal of the Oscar Project to develop a car according to Open Source principles (Oscar, 2008).

1. Introduction

Automotive studies that focus on product development predominantly concentrate on individual organisations, such as GM, or on collaborative efforts between companies, such as the alliance between Renault and Nissan. As such, these analyses have one aspect in common; they focus on for-profit corporations engaged in the development of novel products.

Given this observation, the present study aims at analysing these claims made by previous studies in a different realm, the development of a car by volunteers. These volunteers act according to the principles of open source software (OSS), where software is created and refined by volunteers on a non-profit basis. For this study, the Oscar project (www.theoscarproject.org), which was initiated to create a car using the OSS conception model, was investigated.

Against this background, the *objective of this study* is to address the following research questions:

Research question 1: What parallels exist between the field of OSS and the Oscar project?

Research question 2: What barriers exist regarding the implementation of OSS mechanisms to the development of a non-software related product?

Given these research questions, this study aims to illuminate the parallels and differences between the Oscar project and the OSS paradigm, which is frequently deemed to be a role model for innovation management. For this purpose, we revert to the commons-based peer production (CBPP) framework as introduced by Benkler (2002, 2006). He posited that OSS projects operate exceedingly efficiently due to specific motivational and coordinative mechanisms. We will show that, although there are similarities between the development of a tangible product and the development of OSS, the applicability of the OSS concept to the development of a tangible product, like a car, is limited.

The *structure of this article* is as follows. First, the study summarises previous findings from OSS-related research. Second, the study elaborates on the CBPP framework that guides our line of reasoning. Third, the research setting is depicted. Fourth, the study outlines the methodological approach. Fifth, the OSS-related nature of the Oscar project is discussed in light of the CBPP framework. Sixth, the managerial and theoretical

* Corresponding author. Tel.: +49 30 838 56 359; fax +49 30 838 56 808.

E-mail addresses: gordon.mueller-seitz@tu-berlin.de (G. Müller-Seitz), reger@uni-potsdam.de (G. Reger).

implications of the findings are discussed. The article concludes with a summary, an analysis, and suggestions for future research directions.

2. A survey of studies on open source software

Open source software represents a unique approach for developing computer software (Lakhani and von Hippel, 2003; Pykäläinen, 2007; Raymond, 2001). The most striking feature of OSS projects is the lack of traditional organisational mechanisms. This is best manifested in the absence of conventional hierarchies, rules, and internal organisational bodies (Crowston and Howison, 2006; von Krogh et al., 2003). Although differences exist between each OSS project, the term “open” indicates that anyone is permitted to study, change, improve, and distribute the unmodified or modified software. In this connection, “open source” implies that the source code of the software is freely available, in contrast to conventional proprietary software solutions. The overall goal of these projects is the joint development of software. Some of the most prominent examples of open source software are Apache, Samba, or Linux.

Bearing these characteristics in mind, two central aspects are important for the present study. First, traditional approaches in organisational theory are not able to fully explain the OSS phenomenon properly. Second, conventional motivational notions do not seem to apply to the OSS arena.

2.1. Basic features of OSS projects

One of the main characteristics of OSS development is that it relies on information technology. Due to their development on the Internet, these projects are entirely virtual (Markus et al., 2000; cf. Aramand, 2008). Moreover, a legal agreement that is grossly different from conventional legal approaches to the development of software is in place. In this connection, the term “open source” denotes the freedom of the software developer to alter the source code and redistribute it, an aspect that led Stallman to coin the term “copyleft” (Stallman, 1998) as an antithesis to copyright (de Laat, 2005; van Wendel de Joode et al., 2003). The only obligation that ensues from the copyleft is to likewise distribute the results under the copyleft. Apart from that, the only rule in place take the form of style guides (Apache, 2007). Hence, most of the rules are informal in nature and are conveyed in the course of working with fellow developers or by means of official notices on the websites that convey the rules in a companionable manner. Prior research also conjectured that the comparatively low level of restrictions results in the possibility of engaging only with those tasks that the respective contributor feels comfortable with. This self-selection mechanism represents a contrast to traditional work settings where tasks are usually distributed on a more hierarchical or directive basis.

Apart from their technological and legal infrastructure, OSS projects are also characterised by a set of shared norms, like communities (O'Mahony, 2003; Hertel et al., 2003). As a result, these projects are not compared with formal organisations but are viewed as communities of practice (Kogut and Metiu, 2001; Lave and Wenger, 1991). Members of OSS projects share the assumption that the behaviour of for-profit corporations is often unethical. As a result, they intend to show some form of resistance against the for-profit corporations, such as Microsoft, by developing software on a non-profit base.

2.2. Motivational stimuli

As the contributors to OSS projects are not formal members of an organisation and there exists no remuneration for the highly

skilled developers (Raymond, 2001), the motivational mechanisms that explain the voluntary nature of these participants ought to be different (Benabou and Tirole, 2003; Lakhani and von Hippel, 2003). Referring to the conception of Deci (Deci, 1971, 1975), a dualistic differentiation between intrinsic (i.e., activities and behaviours that participants in OSS projects naturally engage in for their own sake) and extrinsic (i.e., where a direct compensation for the respective activity is anticipated) motivation is proposed.

With regard to intrinsic motivation, contributors to OSS projects often indicate that they just enjoy improving the source code (Lakhani and Wolf, 2005). Therefore, innate aspiration seems to be of chief importance, as it may lead to a greater possibility of attaining this goal (Sheldon and Elliot, 1998). In a similar vein, it has also been conjectured that these developers will invest a considerable amount of time and effort in refining the software code (Lakhani and von Hippel, 2003). Additionally, altruism and pro-social behaviour may stimulate further contributions among developers (von Krogh et al., 2003; Zeitlyn, 2003). Closely connected to this theory is the observation that most software developers are highly idealistic, which is often manifested in an anarchic code of conduct (Himanen, 2001; Pykäläinen, 2007). The main motivation stems from antagonising capitalism. Developers deem capitalism to be epitomised by huge software corporations such as Microsoft, Dell, or Intel. Consequently, developers are inspired to contribute their time and effort insofar as they perceive that belonging to a virtual community is beneficial to them (Lakhani and Wolf, 2005).

Additionally, there are personal and future rewards that can be subsumed under the category of extrinsic motivation. For instance, developers might benefit from improvements of the software code that they refined (Lerner and Tirole, 2002). In these cases, their expertise comes to the fore in the respective community, whereas the genealogy of the software code is still accessible within the threads. A thread is the documentation, in the form of a list, that begins with either information or a question that is added by a developer that is subsequently discussed in the form of comments that are directly attached to the information given by the first developer. Thereby, information can be grouped according to the time that the information was added or with regard to the topic at hand. For example, a person might have a question regarding a particular aspect of programming the software. Subsequently, other programmers will respond to this question by commenting directly to his or her mail in an evolving list attached to the initial question.

Apart from personal benefits, social recognition among peers is another extrinsic motivation (Lerner and Tirole, 2001). By frequently updating the source code, a participant can enhance his reputation in the respective community (Lerner and Tirole, 2002), which can be labelled as a novel form of “self-marketing” or “status signalling” (Jeppesen and Frederiksen, 2006; Lee and Cole, 2003; Lakhani and von Hippel, 2003).

3. Commons-based peer production as a conceptual frame of reference

The efficiency of OSS projects cannot be described by conventional models of hierarchies, markets, or networks (Demil and Lecocq, 2006). Hence, alternative approaches to traditional frameworks, such as transaction cost economic analysis, (Williamson, 1991) are needed to incorporate the peculiarities of OSS projects as described above. As indicated, OSS contributors are neither paid to participate nor are they employed by the respective project.

Download English Version:

<https://daneshyari.com/en/article/1021943>

Download Persian Version:

<https://daneshyari.com/article/1021943>

[Daneshyari.com](https://daneshyari.com)