On the open design of tangible goods

Christina Raasch, Cornelius Herstatt, Kerstin Balka

Mar 26th, 2009

R&D Management, forthcoming

On the open design of tangible goods

Christina Raasch, Cornelius Herstatt, Kerstin Balka

Abstract

Open Source Software development has received considerable scholarly attention, much of which is based on the presumption that the ‘open source model’ holds some lessons of broader applicability. Nonetheless, our knowledge of its deployment outside the software industry is very limited. This paper focuses on the open source development of tangible objects, so-called open design. We propose a generalised definition of open source development. Drawing on 27 exploratory interviews and six comparative case studies selected from a pool of more than 75 projects, we analyse the workings of open design. The analysis reveals that open design is already being implemented in a substantial variety of projects with different organisational and institutional structures.

1. Introduction

Over the last decade, substantial scholarly attention has been devoted to open source software (OSS). While the software industry is obviously an important one in many respects, a principal reason motivating much of this research is the underlying assumption that the “open source model” of development (Osterloh/Rota 2007) holds some lessons of broader applicability (Nuvolari/Rullani 2007, p. 227). Many researchers view the ‘OS model’ as a system of communal production (Gläser 2007) for which software offers an advantageous, but by no means exclusive environment (von Hippel/von Krogh 2003; Shirky 2005; Chesbrough/Appleyard 2007). If, however, “open source beyond software” is not an oxymoron, we should strive to extend our knowledge of the factors influencing the transferability of this development model.

Previous research results suggest that the production of intangible products and particularly open content may be more accessible to the ‘OS method’ than the development of physical products (e.g. Maurer/Scotchmer 2006). Examples such as Wikipedia, but also to some extent academic research are therefore receiving increasing attention. However, practical examples also point to the potential for success of the ‘OS model’ in the development of *physical* products, a hitherto little-researched field (Hope 2005). In this paper, we take a closer look at the transferability of the open source model of development to other industries. Our examination focuses on the modes of OS development of tangible objects, particularly on the actors involved, the objects being developed, and the question how the ideas and solutions of a group of dislocated volunteers can be coordinated to create an innovative physical object.

Our paper is structured as follows: Section 2 describes the theoretical foundations and the empirical relevance of the OS model of innovation (OSI) in a non-software-specific context. We proceed to describe our research methodology and data sources in Section 3. The research design is exploratory and rests on a comparative case study preceded by expert interviews. We study six cases, which are introduced briefly in Section 4 and analysed in detail in Section 5. Section 6 concludes and discusses the theoretical and practical implications of our findings.

2. Theoretical and empirical relevance of 'open source beyond software'

2.1 The meaning of open source beyond software

In the scholarly literature, OSS is identified as an example of a "new innovation model" beyond markets, hierarchies and strategic alliances (Osterloh/Rota 2007) that has also been referred to as the "community-based model" (Shah 2005), the "open source method" (Osterloh/Rota 2007), or "opensourcing" (Agerfalk/Fitzgerald 2008). Open source development is an example of the private-collective model (von Hippel/von Krogh 2003) and one form of open technology (Nuvolari/Rullani 2007).

In view of the sweeping success of this new model in the software industry, an increasing number of researchers and practitioners believe that other industries may also avail themselves of open source development processes in the future (e.g. Lee/Cole 2003; Lerner/Tirole 2004; Maurer/Scotchmer 2006; Fleming/Waguespack 2007). As a confirmation they refer to a small number of existing projects, e.g. the OScar project or biotechnology projects (e.g. Hope 2005; Pearce/Ferguson 2006, Müller-Seitz/Reger 2008).

Yet, "what does it mean to apply the term 'open source' in fields outside software development, which do not use 'source code' as a term of art? […] It seems the time has come to devise a new, broader term" (Economist 2004). To generalise the 'OS model' to a non-industry-specific level, we propose the following model, which we call Open Source Innovation (OSI): OSI is characterised by free revealing of information on a new design with the intention of collaborative development of a single design or a limited number of related designs for market or non-market exploitation.

This definition contains four critical aspects: (1) OSI is characterised by a non-market, non-contractual transfer of knowledge among the actors involved in invention and between those actors and those involved in exploitation. Actors share some ideas, designs, and other relevant information with a non-definite set of other actors without any immediate recompense or expectation thereof. An open licence may establish rules of knowledge sharing and re-usage. The actors engaging in such activity are often called volunteers, in the sense that the activity is voluntary at the level of the individual and of the contributing corporation, but not necessarily at the level of each member of such a company. (2) Actors share their ideas with the clear purpose of contributing to the joint development of (3) a single, integrated design or a small number of interrelated designs. Put reversely, OSI does not refer to design information being revealed by actors without the understanding that this design is a part of a larger design task carried out in collaborative fashion by a group of actors. (4) The design thus developed is exploited in the sense that it is produced and sold on a market, integrated into other products that are marketed, or deployed during the development of such products. Exploitation can be either for-profit or not-for-profit or both.

Our model builds on and intersects with several of the models of collaborative development proposed in the literature (cf. Raasch/Herstatt et al. 2008). The theoretical foundation supporting OSI is laid by the private-collective model (von Hippel/von Krogh 2003; von Hippel/von Krogh 2006), whereby actors invest private resources towards the production of a public good. Within this model, OSI focuses on a collaborative development process involving several contributing actors and therefore requiring organisational mechanisms to co-ordinate the efforts expended by different actors.

OSI also needs to be related to user innovation networks or communities (von Hippel 2005; von Hippel 2007; Füller/Jawecki et al. 2007). OSI can be generated by commercial or private contributors or a mixture of both. If commercial companies participate in OSI, they may profit from using the product being developed, but may also have different objectives, e.g. an increase in sales of complementary products (Henkel 2006). The user-innovation model, by contrast, is based on “open, voluntary, and collaborative efforts of users” (Shah 2005, p. 1), meaning private or commercial, but still users. Additionally, OSI refers to collaborative development efforts on an integrated design (or a number of closely related designs or design variants), a condition that does not necessarily hold for user communities (Lüthje/Herstatt et al. 2005).

2.2 Empirical relevance of ‘open source beyond software’

A principal distinction in relation to OSI can be drawn between intangible and tangible objects of development. In the digital realm, so-called *open content* is currently proliferating and consequently attracting considerable attention. The entire family of Wikis (Peddibhotla/ Subramani 2007), cultural goods such as music or films being co-developed, co-funded and shared freely, open science (Dasgupta/David 1994; Hellström 2003), the development of educational materials (OECD 2007), bioinformatics databases (Thompson 2002), geographic maps of the world (the OpenStreetMap project), and other applications have demonstrated the potential inherent in open content. They suggest that, in the digital realm at least, the open source model is viable and can offer business opportunities to companies. At least one qualification seems noteworthy, however: Open content platforms neither necessarily aspire to nor always deliver any substantial degree of innovativeness, and therefore do not form a strict subset of OSI.

The second group, called *open design* (Vallance/Kiani et al. 2001), describes open hardware as well as other *physical* objects being developed in accordance with the OSI model. While much of the development work in this group can be accomplished virtually, the ultimate purpose is the design and production of a physical artefact. Not least due to the success of OSS, open design is enjoying an upsurge (Hope 2003). Mostly unheeded by scholarly research, a multitude of open design projects has constituted itself since approx. 2005, ranging from bicycles to microchips and from MP3 players to manufacturing equipment. While most of these projects are still in development, others have had marketed products for several years.

This paper focuses on open design rather than open content, since it not only seems the less researched of the two fields, but also lies further from the original realm of software. It is important to point out, however, that the distinction between open design and open content is not as stringent as it may appear, as "hardware is becoming much more like software” (von Hippel in Thompson 2008). “An increasing number of physical products are becoming so data-centric that the physical aspects are simply executional steps at the end of a chain of digital manipulation“ (Shirky 2007). This observation notwithstanding, tangible products do require actual physical production, an aspect often regarded as a significant challenge to open design (Maurer/Scotchmer 2006).

3. Approach and methodology of empirical research

The dearth of secure knowledge on open design necessitates an exploratory approach to this paper. We choose a multiple case study approach, believing this avenue to be the most appropriate and promising method at present to proceed. Case study research is generally chosen when "why" or "how" questions are being posed on current events over which the investigator has little control (Yin 2003). It is particularly suited to research into new topics with the purpose of advancing theory (Eisenhardt 1989; Gillham 2000). Multiple case analysis is generally regarded as being more robust than single case studies (Herriott/Firestone 1983; Yin 2003). Our empirical research proceeded in three major steps:

At the outset, we conducted 27 exploratory expert interviews, of which 10 were conducted with coordinators of OSI projects and 17 with industry experts from several industries. During these interviews, we collected information on the OSI landscape and began to gather case study candidates. Thus, two of these interviews referred to projects later chosen as case studies.

Next, we compiled a pool of more than 75 open design projects, based on a thorough research of secondary data and suggestions by our interview partners. Cases were systematised by means of 24 criteria.

We proceeded to determine criteria for case selection. At this early stage of research on the open design phenomenon an encompassing view seemed advantageous. As recommended by Pettigrew (1988), we therefore selected cases to "fill theoretical categories and provide examples of polar types" (Eisenhardt 1989, p. 537). Based on our interviews, two dimensions of categorization were particularly relevant: artefactual characteristics (complexity, modularity, industry) and characteristics of the developer community (individual developers only, joint community-company projects). For each category we strove to have at least two cases in order to allow findings to be replicated (Yin 2003). As an auxiliary condition we demanded that all projects be well established, attracting a reasonable number of contributors and demonstrating some development progress across time. This criterion was not tantamount to selecting only successful projects, but served to exclude very early-stage projects with yet unclear organizing structures and very little interaction.

We then drew on our pool of projects to select cases for in-depth analysis, pursuing a multi-stage filtering process to optimise for these criteria. Purposeful sampling of information-rich cases was chosen as the most appropriate approach. In particular, we sought to mirror the diversity of projects found in our case pool by selecting mutually dissimilar projects. In the end, a choice of six open design projects resulted.

During data collection for each case, we used multiple sources of evidence: First, documentations and archives written by the developer communities themselves or other authors were studied extensively. Second, we observed each community over a ten-week period, reading and triggering mailing list discussions, chat conversations, and forum entries. Third, 3-6 in-depth expert interviews per case were conducted (cf. Appendix) using a semi-structured interview guideline. Experts were mostly interviewed via telephone, but some preferred chat room conversations. Remaining questions were resolved in follow-up telephone calls, chats, or emails. Our interview partners were chiefly either the project coordinators or company CEOs respectively or members of the project core team.

For triangulation, we compiled and then compared the information gathered during different interviews as well as a large amount of secondary data (Yin 2003). Remaining questions or incongruities were addressed to our interview partners for clarification. Finally, the case study results were reviewed by some of our expert interviewees.

4. Overview of cases

In accordance with the selection criteria discussed in the previous section, we examine cases from very different industries, including automotive, consumer electronics, mobile communication, and beverages, for example. The cases vary in terms of the complexity of the object being developed, ranging from simple (beer) to very complex (cars). Most of the projects attract several hundred contributors. Two projects already resulted in fully functional products offered in the market (Free Beer and Neuros OSD). Two cases (Openmoko and RepRap) spawned test versions or components available for sale. While the OSGV (Open Source Green Vehicle) project plans to build a prototype in 2009, OScar is still in early development. Figure 2 provides an overview of the selected cases.

	Start	Industry	Object	Coordinator	Community size
OScar	1999	Automotive	‘World car’ with sustainable mobility concept	Markus Merz, Monocom, Germany	~ 3,000
RepRap	2004	Printing / Rapid prototyping	Self-replicating 3D-printer for home use	Dr. Adrian Bowyer, Bath University, UK	~ 100
Free Beer	2005	Beverages	Beer	Rasmus Nielsen, Superflex, Denmark	~ 15
OSGV	2005	Automotive	7-passenger sports utility vehicle with various fuel options	David W. Lee, SSM, USA	~ 250
Openmoko	2006	Mobile communication	Mobile telephone	Sean Moss-Pultz, Openmoko Inc., Taiwan	~ 2,000
Neuros OSD	2003	Home entertainment equipment	Linux-based, embedded media center	Joe Born, Neuros Technology Inc., USA	~30,000

Figure 2: Overview of selected cases

(1) OScar: In 1999, the OScar project was initiated by Markus Merz, CEO of a small consulting firm designing e-business strategies for the automotive industry. He planned to complete a car design by virtual collaboration and then offer it to OEMs and suppliers for production. Merz set out by writing a ‘manifesto’ and building a website to rally supporters to the project. Free access, public ownership of designs, and democratic decision processes with himself simply as an orchestrator were the principal tenets of his plan. The initial goal to complete a design within 36 months soon appeared too optimistic; two factors slowing the project down were the absence of cheap or free CAD programs meeting the project’s requirements and a lack of focus and direction that accompanied the entirely decentralised approach.

After a relaunch of the project in 2005 and a redesign of the website, OScar began to attract thousands of readers and co-developers. Merz provided some development guidelines for OScar 0.2, but otherwise the design evolves from discussions in the community forums. The car is meant to be simple and cheap to produce (a ‘world car’) as well as economical in terms of fuel consumption. To date, the project is still in early design phase; completion is assumed to take another five to ten years.

(2) RepRap: The RepRap project was initiated by Dr. Adrian Bowyer, a senior lecturer at the University of Bath, in 2004. In an article on the Internet and a press release he announced his goal to create a self-replicating machine, i.e. a 3D-printer that is able to produce most of the parts needed to assemble another such machine. A commercial 3D-printer currently costs approx. US $20,000, whereas the price target of RapRap lies below US$400.

The project relies on decentralised development and production accomplished by geographically dispersed project members. Initially new members simply volunteered and were welcomed to the team. As the community grew, a core team was formed to drive and coordinate development, while maintaining a decentralised, mostly non-hierarchical structure. The first fully functional, self-replicated RepRap v1.0 (called "Darwin") was completed in June 2008 and celebrated as a great success. Next steps are the reduction of included third-party components, the advancement of the printing technology, and the recruiting of new community members.

(3) Free Beer: Free Beer, previously known as Vores Øl, is commonly regarded as the first open source beer. It was originally created by students in a class on copyright law at the IT-University of Copenhagen together with Superflex, a Copenhagen-based artist group. They set out to illustrate how concepts of the free software movement could be applied outside the digital world. In 2005, they brewed about 100 bottles of Vores Øl 1.0 in the school cafeteria.

The subsequent amount of media attention and feedback they received came as a surprise. From these discussions of the recipe and the experiment itself, the project emerged.

Superflex then took the idea further and has since released several new, revised versions based on feedback from testers. By now Free Beer has been brewed on a large number of different occasions ranging from student parties and conferences in Germany to exhibitions in Brazil. In addition, Project21, a Swiss student organization for sustainable development, created a spin-off under the Free Beer trademark in 2007. They developed a new recipe and first brewed 1,000 liters in a local brewery. Again they received unexpectedly strong feedback and hence decided to continue and find a sponsor. Within one year, they sold approx. 5,000 liters of Free Beer. Today, Free Beer is produced and marketed by two breweries in Copenhagen and Zurich.

(4) OSGV: The OSGV (Open Source Green Vehicle) project was initiated by David W. Lee, who is still project manager; it is run by the SSM (Society for Sustainable Mobility), a non-profit automotive engineering group set up in late 2005. The project goal is to employ open design to contribute to solving environmental issues, particularly to promote sustainable mobility (cf. Lee 2008).

Following market analysis and lengthy discussions on the project forums, Lee decided that the design should be a seven-passenger SUV. The design should allow easy swapping of the power generating unit and thus fuel type (gasoline, biodiesel, hydrogen fuel cell, ethanol, natural gas). A core developer team then defined open specifications on the system, sub-system and interface level. Based on this set of tightly controlled specifications, a community of about 250 members completed a preliminary design in early 2008. Further development and the building of a prototype are to be accomplished by 2009. To achieve this goal, SSM has licensed the design to a venture-capital backed start-up company which will finish development and take on production.

(5) OpenMoko: In early 2006, Sean Moss-Pultz assembled a group of four core developers to run an open source software project for a Windows-Smartphone on behalf of FIC (First International Computer, Taiwan), a producer of hardware. In answer to strong media and industry interest in the project, FIC put their Windows plans on hold and increased resources for developing a Linux-based Smartphone. In late 2006, they announced the development of an open mobile telephone, in which openness would include software and hardware. Today, two years later, thousands of community members participate in the development of both hardware and software, shaping the result to a large extent.

Openmoko, Inc., founded as a subsidiary of FIC in early 2008, employs approx. 70 geographically dispersed developers to improve the current design into a mobile phone suitable for the mass market. The first 5.000 prototypes were put on the market in July 2007 and sold out within one month; in July 2008, the second generation was launched with similar success.

(6) Neuros OSD: Neuros Technology, a Chicago-based spin-off of Digital Innovations (DI) with approx. 35 employees, sells audio and video devices, among them the Neuros OSD (Open Source Device). The Neuros OSD is an open source, Linux-based, embedded media center, i.e. a device to record digital content from various sources and to store and output this content in a standardised format used by most digital devices, from MP3 players to mobile telephones and televisions.

Faced with three problems, the need to innovate, the need to obtain very detailed customer feedback on application features, and the need for speed in the fast-moving video space, Neuros CEO Joe Born successively laid open its design specifications and development process to a growing community for feedback, bug tracking, documentation, and co-development. Some parts of the design, e.g. components delivered by third party vendors, as well as corporate decision processes, remain closed to the public.

5. How does open design work?

5.1 Who drives open design projects?

Mirroring findings from OSS, we identify community-driven (OScar, Free Beer) and company-driven (Neuros OSD, OpenMoko, OSGV) open design projects as well as projects led by members of research institutions (RepRap). In the Neuros OSD case, one company directs the development process and delivers most of the work, laying open its designs for a community to co-develop. OScar, by contrast, is entirely community-driven, similar to the RepRap project in which a community is guided by a university researcher combining project and academic work. In between lie OpenMoko, rather closer to, but not as extreme as the Neuros OSD case, Free Beer, which unites a community of users with commercial contributors and producers, and OSGV, which set out with a community of individual contributors and researchers led by a non-profit organisation, but has been largely taken over by a start-up company.

Some scholars of collaborative development have suggested that developers active in open design must possess strong and specialized skills, and that this necessity poses a substantial impediment to open design (Lerner/Tirole 2004). Our case studies mostly corroborate this proposition. In all cases we studied, apart from Free Beer, developers with special skills are required, e.g. engineering, informatics, or electronics. In addition to depth of expertise, open design projects also tend to require developers from a considerable diversity of backgrounds. RepRap developers believe their group to be more multi-disciplinary than observed in most software projects. Similarly, the completion of a functional car design necessitates mechanical engineers and automotive engineers as well specialist disciplines such as high-power electronics, fuel efficiency, etc.

Prior experience with OSS development is not typically deemed an important prerequisite for participation in open design, and indeed many developers do not possess any. Still, an OSS background instills a knowledge of a “totally different development culture and an appreciation of openness, of the fact that open or free standards are preferable to some proprietary solution for the design process” (OpenMoko). Developers with prior experience in OSS projects may thus be particularly likely to subscribe to the goals pursued in open design.

5.2 What is being developed?

In all our cases except one (Free Beer), the artefact being developed is modular. Perhaps even more importantly, these open design projects make a particular effort to develop the artefact in a modular way. Thus, the open design even of very complex objects is considered feasible, as long as they can be modularised. These findings are in accordance with studies on modularity in the field of OSS (MacCormack/Rusnak et al. 2006; Baldwin/Clark 2006).

In the Free Beer project, in which an integral product is developed, the low level of complexity of the object enables collaborative and dislocated task completion despite the lack of task granularity. This suggests that the feasibility of open design depends jointly on the degree of modularity and the degree of complexity of the artefact. High complexity necessitates a modular architecture, while simple objects may be amenable to open design despite modularity being low.

Our cases reveal that development tasks of considerable complexity are tackled within open design settings. One important strategy to manage this complexity is outsourcing. Allowing for outsourcing to third-party suppliers, the fact that a mobile phone, for instance, includes a large number of sophisticated technical components, which would probably be beyond the OpenMoko community to develop afresh, does not preclude success within an open design set-up. This is illustrated by the following quotation:

> "At a certain point in time someone has to make the boards […] and things of that nature. But the thing that we're working on, the open source, is the design. […] nobody has time to assemble the boards." (RepRap)

However, this strategy shifts complexity to a different level, i.e. specifications, interfaces, and documentation.

Summarising first findings on the characteristics of the artefacts developed in open design we find that, first, modular objects and simple objects seem particularly well suited to open design, second, feasibility seems to depend jointly on modularity and complexity, and third, more complex, but modular designs can be completed with the help of third-party suppliers.

5.3 How are open designs developed and produced?

In our case studies, we identify three different loci of production: with external manufacturers, with the community, or with the focal organisation coordinating the project. OSGV and OScar plan to take the first route, once they get to this stage; RepRap takes the second approach, and Neuros OSD and OpenMoko the third. Free Beer is brewed both by the community and by large commercial breweries.

Production by companies external to the project is typically chosen in open design projects, which do not have an OEM as their coordinator, such as OScar or OSGV. Besides the cost of testing and building physical objects and the potential of exploiting economies of scale in manufacturing, product safety and liability issues are important considerations.

The RepRap 3-D printing machine is assembled by each developer from standard parts commonly available for sale and parts printed by another RepRap according to the project design files. Assembly allows for some choices, depending on the developers' personal requirements and willingness to pay. Development tasks are performed by the community on their own machines, which are in various stages of completion. If an improvement is voted to be integrated into the official project design, they must either update their machines or stay with the older version.

> "If changes are made to the hardware, then everybody that's got hardware already built has to start again or modify it. There's some cost associated to this, which would not occur with software. They can just download it and they got it." (RepRap)

> "Because every [component] is necessarily seen and assembled by someone that builds a RepRap, there's a very complete peer review going on." (RepRap)

As these quotations illustrate, there is no clear-cut separation between design, prototyping, and production in the community production regime employed be RepRap, at least not at this stage. Once the design is finalised, the option of involving an external manufacturer for mass-production may be considered – although this does not seem a likely avenue at present.

An amalgamation of development and production stages can also be identified in our two cases which feature production by the company coordinating the project. Both OpenMoko and Neuros OSD rely on "release procedures [which] are deliberately very soft" (Neuros OSD), whereby different test versions are produced and released to the community, either for free or at a cost, for testing and improvement. The target group for this co-development and co-testing process is successively expanded; advanced test versions are offered for general sale. Thus one version is sold in the market, while the follow-on improved version is being developed based on community feedback. What seems noteworthy in this context is a considerable fault tolerance among the community: Their eagerness to buy ‘bug-ridden’ beta versions or ‘developer samples’ for co-development took supplying companies by surprise.

“This is our version of ‘release early, release often’. Developers decide whether they want to buy (or get for free) an earlier version full of bugs or a later version.” (Neuros OSD)

“Gamma is a ‘white box’ pre-production product stage especially geared for hackers and hard-core early adopters. Gamma is the next stage after Beta, it's the first pre-production run, and we make it available to the public. Central to Neuros' strategy is releasing products to the market quickly to get early feedback from hard core users as well as to give hackers a head start. In recognition of the fact that such products often change quickly and are not fully field tested, we offer an extended no questions asked return policy on such products. In addition, we also often offer compensation for participation in feedback surveys and focus groups, etc.” (Neuros Technology 2008)

5.4 Is open design really open?

Laying open design information was seen as the fundamental norm of open design by all our interviewees, and many took pride in advancing a design of a physical product in an open environment.

“The basic premise is: Everything that does not absolutely have to be kept internal, is public.” (OpenMoko)

To think of open design as entirely open, however, would be over-simplistic. In order to paint a more fine-grained picture, we draw on a distinction proposed by West (2003) between “open parts” and “partly open”. According to the ‘open parts’ strategy, the project coordinator “grant[s] all rights to a subset [of the design]”, whereas “partly open” refers to the release of a design “under restrictive terms” (West 2003, p. 1277).

Counter to the common understanding of ‘open’ design, five of our six case studies actually reveal some limitations to openness, according to either one or both elements of West’s conceptual dyad. They rely on trade secrecy, trademarks, and in some cases even patents (for OSS cf. Dahlander and Magnusson 2005). Limitations to the scope (‘open parts’) or the degree (‘partly open’) of openness are sought either by the suppliers of components or by the focal company or core team of the project. In several cases (e.g. OSGV, Openmoko) we identified limitations to openness in both degree and scope, the regime varying among the components. In other words, these designs included entirely open parts, partly open parts, and closed parts. Figure 3 provides some case-based examples of these two strategies.

	‘Open parts‘	‘Partly open‘
Rights retained by focal organisation	“In order to establish a suitable level ‘proprietary’ position to attract investments, the ownership of […] the intellectual properties should be maintained by a non-profit / for-profit corporation instead of allowing them to flow into the public domain (free-for-all).” (OSGV)	“For those interested in modifying the Neuros at the hardware level, we will do our best to release any and all relevant documentation regarding the technical specifications of the device as soon as we can. While market pressures may inhibit the amount of information we can share, we will share everything we can at the earliest possible time.” (Social Contract, Neuros OSD)
Rights retained by supplier	“Some parts of our system are not Open Source Software. We are forced to distribute them as binaries due to agreements with the vendor. It was deemed a reasonable tradeoff to bring the product to market at all. Some parts may be rewritten cleanly as OSS, but others are quite hard to get rid of. […] Here's a brief rundown of what belongs to the non-open category, and what may be opened” (Neuros OSD).	“Open specifications are crucial. […] Open circuit diagrams could be really useful for software developers, but board layouts are only needed by someone who wants to copy the product 1:1. For the sake of ‘peaceful co-existence‘, the developer community should not ask too much in terms of openness.“ (OpenMoko)

Figure 3: Examples of ‘open parts’ and ‘partly open’ strategies in open design projects

The common denominator behind all these examples appears to be the endeavor to balance the interests of the designer community and commercial companies involved, e.g., as suppliers or manufacturers. In several of our cases, there is an awareness that, by deciding to “leave enough room to encourage private investment” (OSGV), the community can improve its probability of success.

The major exception among our cases is the OScar project core team who consciously reject this avenue: According to founder Markus Merz, adherence to OS principles, particularly openness, takes precedence over bringing the car to market. OScar is primarily intended to prove the feasibility of open car design, which is why actual production is deemed secondary.

In all other cases decision makers try to promote project success by carefully weighing community and commercial requirements. Some findings, particularly from the OpenMoko and OSGV projects, suggest that this trade-off is actually accepted by community members, as long as the balance is perceived to be fair. One of the OpenMoko core team members describes a give-and-take negotiation process.

> “On the part of technical developers, there is a desire to have completely open circuit diagrams; but there is also an understanding for the magnitude of the concession this would be for the investor. The compromise is: Circuit diagrams are revealed under a special NDA [non-disclosure agreement] in order to facilitate the development of the software by programmers not working within the walls of the company.” (OpenMoko)

One could speculate that developer culture in open design projects tends to be more lenient compared to OSS with regard to limitations to openness for the purpose of earning private rents. It is well understood that many designs are hard to produce by users themselves and therefore need to be manufactured commercially – if they are to be available to the community at all. While conclusions based solely on our results would appear premature, a comparison to OSS environments might yield interesting differences in community attitudes and behaviour.

6. Summary and implications for research and industry practice

The potential of external volunteers for product (co-)development was demonstrated starkly by the success of OSS development in recent years. In view of this phenomenon, many experts from academia and practice suggest that the ‘open source model’ may be transferable to other industries. This assumption notwithstanding, little is known about the preconditions, opportunities and barriers of transferring the OS model of software development to other domains, especially to the open design of physical products.

Our analysis aims to stimulate this stream of research by pointing out the recent upsurge of practical cases and deriving first tentative findings on the workings of the open design model of new product development (NPD). The analysis sheds light on some patterns and options in project set-up and highlights the heterogeneity of open design projects. Comparisons to OSS as well as traditional NPD point to similarities as well as substantial differences.

In coming years, if and when practical applications of open design and thus sources of data proliferate, quantitative empirical study will be facilitated. More profoundly, however, the viability of several of our cases and in fact the entire model of open design is as yet unclear. Over the next decade, open design may take different routes outlined by the following three quotations:

> “Assessment of innovations at an early stage in their history is a hazardous activity. […] History offers numerous examples of innovations thought to be of great magnitude whose principles of operation proved to be either flawed or fraudulent. In the case of open source […] this possibility cannot be entirely ignored.” (Garcia/Steinmueller 2003, p. 4)

> “This is just a matter of time. Electronics will follow the way PCs took 25 years ago over the next couple of years. […] Hardware specialists who cannot afford huge investment in software will build standard-compliant hardware.” (Neuros OSD)

> “Open project evolution is somewhat Darwinian. Many fall by the wayside and are abandoned, some fork into newer and better projects leaving the parent behind, some just keep on going successfully.” (RepRap)

We expect that strategies to tackle the challenges presented by the translation of the OS model of software development to the physical world will be subject to experimentation in coming years. During this process, the increasing number of individuals, companies, and research institutions considering the initiation of open design projects or already participating in existing projects would profit from a more profound understanding of the issues advanced in this paper.

Acknowledgements

Our sincere thanks go to our interview partners from the six case studies for their time and support of our research. We would particularly like to thank David W. Lee and Roeland Hogt (OSGV), Joe Born and Martin Springer (Neuros OSD), Harald Welte and Michael Laurer (Openmoko), Markus Merz and Lukas Neckermann (OScar), and Dr. Adrian Bowyer and Zach Smith (RepRap). We are likewise indebted to two unknown reviewers for their valuable feedback during the review process. Finally, we gratefully acknowledge the generous support of our research by the German Federal Ministry for Education and Research (BMBF) (project code 16I1573). The authors carry sole responsibility for the content of this paper.

Appendix: List of interviews

	Openmoko	Type of interview	Length	Project founder	Project leader	Core team	Active developer
1	(1)	Chat	2h 30min				X
2	(2)	Chat	1h 30min				X
3	(3)	Chat	1h 10min			X	
4	(4)	Chat	40min				X
5	(5)	Phone	45min	X		X	
6	(6)	Phone	40min	X		X	
	RepRap						
7	(1)	Forum				X	
8	(2)	Chat	45min				X
9	(3)	Email					X
10	(4)	Phone	20min			X	
11	(5)	Phone	20min			X	
12	(6)	Phone	30min	X	X		
	Neuros						
13	(1)	Chat	35min			X	
14	(2)	Email					X
15	(3)	Phone	50min	X	X		
16	(4)	Phone	1h 20min				X
17	(5)	Forum		X	X		
	OSGV						
18	(1)	Phone	1h	X	X		
19	(2)	Phone	1h 10min	X	X		
20	(3)	Phone	45min			X	
21	(4)	Email					X
	OScar						
22	(1)	Phone	35min	X	X		
23	(2)	Phone	1h 10min				X
24	(3)	Phone	35min			X	
	Free Beer						
25	(1)	Phone	20min	X		X	
26	(2)	Phone	30min	X	X		
27	(3)	Phone	20min				X

Note: Case studies only, without exploratory expert interviews

References

Agerfalk, P. J., Fitzgerald, B. (2008) Opensourcing to an unknown workforce: Exploring opensourcing as a global sourcing strategy. *MIS Quarterly*, **32(2)**, 385-410.

Baldwin, C. Y., Clark, K. B. (2006) The architecture of participation: Does code architecture mitigate free riding in the open source development model?. *Management Science*, **52(7)**, 1116-1127.

Chesbrough, H., Appleyard, M. M. (2007) Open innovation and strategy. *California Management Review*, **50(1)**, 57-76.

Dahlander, L., Magnusson, M. G. (2005) Relationships between open source software companies and communities: Observations from Nordic firms. *Research Policy* **34(4)**, 481-493.

Economist. (2004) An open source shot in the arm? *The Economist*, **June 10th**.

Eisenhardt, K. M. (1989) Building theories from case study research. *Academy of Management Review*, **14(4)**, 532-550.

Fleming, L., Waguespack, D. M. (2007) Brokerage, boundary spanning, and leadership in open innovation communities. *Organization Science*, **18(2)**, 165-180.

Füller, J., Jawecki, G., Mühlbacher, H. (2007) Innovation creation by online basketball communities. *Journal of Business Research*, **60(1)**, 60-71.

Garcia, J. M., Steinmueller, Edward W. (2003) The Open Source way of working: A new paradigm for the division of labour in software development? SPRU Electronic Working Paper Series.

Gillham, B. (2000) *Case study research methods*. London: Continuum.

Gläser, J. (2007) The social order of open source software production. In: St.Amant, Kirk and Still, Brian (eds.), *Handbook of research on open source software: Technological, economic, and social perspectives*. Hershey, New York: Information Science Reference.

Hellström, T. (2003) Governing the virtual academic commons. *Research Policy*, **32(2)**, 391-401.

Henkel, J. (2006) Selective revealing in open innovation processes: The case of embedded Linux. *Research Policy*, **35(7)**, 953-969.

Herriott, R. E., Firestone, W. A. (1983), Multisite qualitative policy research: Optimizing description and generalizability, *Educational Researcher*, **12(2)**, 14-19.

Hope, J. (2003) Open source biotechnology?, http://rsss.anu.edu.au/~janeth/OSBiotech.html.

Hope, J. (2005) Open source biotechnology. Research School of Social Sciences and Faculty of Law. Canberra: Australian National University.

Lee, G. K., Cole, R. E. (2003) From a firm-based to a community-based model of knowledge creation: The case of the Linux kernel development. *Organization Science*, **14(6)**, 633-649.

Lee, D. W. (2008) *Good car-ma, bad car-ma: What you and your conscience should know before buying the next car*. Scotts Valley, CA: Create Space.

Lerner, J., Tirole, J. (2004) The economics of technology sharing: Open Source and beyond. Harvard NOM Working Paper, http://papers.ssrn.com/sol3/papers.cfm?abstract_id=620904.

Lüthje, C., Herstatt, C., von Hippel, E. (2005) User-innovators and "local" information: The case of mountain biking. *Research Policy*, **34**, 951-965.

MacCormack, A., Rusnak, J., Baldwin, C. Y. (2006) Exploring the structure of complex software designs: An empirical study of open source and proprietary code. *Management Science*, **52(7)**, 1015-1030.

Maurer, S. M., Scotchmer, S. (2006) Open Source Software: The new intellectual property paradigm. *NBER Working Paper Series*.

Müller-Seitz, G., Reger, G. (2008) Open beyond the software code? - An explorative examination of the development of an open source car. *Proceedings of the EIASM IPDM Conference 2008*, Hamburg.

Nuvolari, A., Rullani, F. (2007) Curious exceptions? Open source software and "open" technology. In: St.Amant, Kirk and Still, Brian (eds.), *Handbook of research on open source software: Technological, economic, and social perspectives*. Hershey, New York: Information Science Reference.

OECD (2007) Giving knowledge for free – The emergence of open educational resources. Centre for Educational Research and Innovation.

Osterloh, M., Rota, S. (2007) Open Source Software development - Just another case of collective invention? *Research Policy*, **36(2)**, 157-171.

Pearce, M., Ferguson, V. (2006) A change of scenery - Open Source licensing and its application to the biotech industry. *Patent World*, **178**.

Peddibhotla, N. B., Sumbramani, M. R. (2007) Contributing to public document repositories: A critical mass theory perspective. *Organization Studies*, **28(3)**, 327-346.

Pettigrew, A. (1988) Longitudinal field research on change: Theory and practice. *National Science Foundation Conference on Longitudinal Research Methods in Organizations*. Austin, TX.

Raasch, C., Herstatt, C., Blecker, T., Abdelkafi, N. (2008) Open Source Innovation - Out of software? *Proceedings of the EIASM IPDM Conference 2008*, Hamburg.

Shah, S. K. (2005) Open beyond software. In: Cooper, Danese, DiBona, Chris and Stone, Mark (eds.), *Open Sources 2*. Sebastopol, CA: O'Reilly Media.

Shirky, C. (2005) Epilogue: Open Source outside the domain of software. In: Feller, Joseph, Fitzgerald, Brian, Hissam, Scott A. and Lakhani, Karim R. (eds.), *Perspectives on Free and Open Source Software*. Cambridge, MA et al.: MIT Press.

Shirky, C. (2007) Generalizing peer production into the physical world. http://finance.groups.yahoo.com/group/decentralization/message/6967.

Thompson, C. (2008) Build it. Share it. Profit. Can Open Source hardware work? *Wired Magazine*, **16.11**.

Thompson, N. (2002) May the source be with you. *Washington Monthly,* **July/August**.

Vallance, R., Kiani, S., Nayfeh, S. (2001) Open design of manufacturing equipment. *Proceedings of the CHIRP 1st International Conference on Agile, Reconfigurable Manufacturing*. http://www.opendesign.org/CHIRP_Open_Design_Mfg_Equipment.pdf.

von Hippel, E. (2007) Horizontal innovation networks - By and for users. *Industrial and Corporate Change*, **16(2)**, 293-315.

von Hippel, E. (2005) *Democratizing innovation*. Cambridge, MA: MIT Press.

von Hippel, E., von Krogh, G. (2003) Open source software and the "private-collective" innovation model: Issues for organization science. *Organization Science*, **14(2)**, 208-223.

von Hippel, E., von Krogh, G. (2006) Free revealing and the private-collective model for innovation incentives. *R&D Management*, **36(3)**, 295-306.

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

Yin, R. K. (2003) *Case study research: Design and methods*. Thousand Oaks, CA et al.: Sage.