Technologie- und
Innovationsmanagement

WORKING PAPER

How open is open source – Software and beyond

Kerstin Balka
Christina Raasch
Cornelius Herstatt

October 2009
Working Paper No. 58

Hamburg University of Technology
Schwarzenbergstr. 95, D-21073 Hamburg, Germany
Tel.: +49 (0)40 42878-3777; Fax: +49 (0)40 42878-2867

www.tu-harburg.de/tim

Abstract

Traditionally the protection of intellectual property is regarded as a precondition for value capture. The rise of open source (OS) software and OS tangible products, so-called open design, has challenged this understanding. Openness is often regarded as a dichotomous variable (open-source vs. closed-source) and it is assumed that online developer communities demand full opening of the product's source.

In this paper we explore openness as a gradual and multi-dimensional concept. We conduct an Internet survey ($N = 270$) among participants of 20 open design communities in the domain of IT hardware and consumer electronics. We find that open design projects pursue complex strategies short of complete openness and that communities value openness of software more highly than openness of hardware.

Our findings suggest that open design companies can successfully employ strategies of partial openness to safeguard value capture without alienating their developer community.

1 Introduction

Economic theory tells us that firms generate innovations in order to reap economic rents. It also tells us that inventions require intellectual property protection in order for imitation competition to be prevented and thus for innovative firms to capture the value they created (Arrow, 1962). Intellectual property rights carry this assurance and thereby serve to incentivise firms to perform their innovating function in the economy. This is the private investment model of innovation (Demsetz, 1967; von Krogh & von Hippel, 2003).

Over the last decade researchers have directed a spotlight on open source innovation as the polar opposite of this model, which accordingly has been termed closed-source innovation. Open source innovation is understood to be 'extremely open' (cf. Gassmann, 2006) since it requires information to be freely revealed to all. The innovator gives up the right to exclusive exploitation of her invention (Harhoff, Henkel, & Hippel, 2003) - a strategy that must seem injurious to any degree of value capture. Accordingly researchers have been puzzled by many inconsistencies of the open source model with the private investment model of innovation (Lerner & Tirole, 2001). Specifically, the motivations of supposedly rational individuals and companies to contribute to such projects and the seemingly self-contradictory notion of open-source business models have been a focus of research (Hecker, 1999; von Krogh et al., 2008).

In this paper we show that this dialectic structure is in fact an over-simplification of the concept of openness and its relation to value capture. Despite proliferating research on open source innovation, the entire construct of openness has received too little attention to date, both theoretically and empirically. West and O'Mahony (2008) and Henkel (2006) are among the very few who explore openness as a gradual and multi-dimensional concept and link intermediate levels of openness to value capture.

Within the scope of this paper we study forms of openness of software and hardware, their empirical prevalence and their relevance for members of open source communities. For this purpose we conducted a survey ($N = 270$) among participants of 20 open design communities. We find that open design projects pursue complex strategies short of complete openness and that communities value openness of software more than openness of hardware, although both are regarded as important. Our findings suggest that well-tailored strategies of partial openness can successfully be employed by open design companies to safeguard differentiation and value capture without alienating their developer community.

The paper is structured as follows: Section 2 presents the theoretical background to our research, specifically prior findings on the concept of openness, and goes on to derive research hypotheses. After the explanation of our research methodology and data collection in Section 3, we present empirical findings in Section 4. Section 5 discusses our results in relation to previous findings, especially the relationship between openness and value capture; and Section 6 concludes.

2 Theoretical background

The term "open source" (OS) originates from the software industry and denotes the free revelation of the source code. An actor grants "access [to his proprietary information] to all interested agents without imposition of any direct payment" (Harhoff et al., 2003, p. 1754).

Beyond pure software development the term "open design" (Vallance, Kiani, & Nayfeh, 2001) provides a framework for sharing design information stemming from hardware as well as other physical objects. This design has different implications for aesthetics, usability, manufacturing, quality, and so forth. Manufacturers often use modular designs to organize complex products. A modular design is composed of modules that are in turn made up of components. (cf. Singhal & Singhal, 2002).

2.1 Openness as a gradual concept

When Harhoff et al. (2003, p. 1753) discuss free revealing of proprietary information, they mean “that *all* existing and potential intellectual property rights to that information are voluntarily given up [...] and all interested parties are given access to it.” Many researchers follow this strict definition and treat openness as a dichotomous variable - open source vs. closed source (e.g. Bitzer, 2004; Dahlander, 2005). Free revealing has been observed in fields as diverse as iron production (Allen & Meyer, 1990), pharmaceuticals (Hope, 2004), and sports equipment (Franke & Shah, 2003).

Practitioners, however, believe that “hard-line approaches, whether open source or proprietary, don’t work [that well in the world of today]” (Thomas, 2008). Bonaccorsi, Rossi, and Giannangeli (2006) and Henkel (2006) find that firms address this issue by revealing selectively, i.e. they carefully decide which parts to reveal and which to keep proprietary.

West (2003) moves the gradual concept of openness one step further by observing that many open source projects impose various limitations to openness. He proposes a distinction between ‘open parts’ and ‘partly open’. The ‘open parts’ strategy refers to the selective free revealing of some components of a modular object. A project developing an open source embedded device could accordingly reveal their software components or their hardware components or both, and within their list of software (and hardware) components they can decide which components to reveal and which to keep proprietary. The ‘partly open’ strategy refers to the release of a design under restrictive terms. The open source project can for example restrict the permitted usage to non-commercial use or limit the group of people who get access to their knowledge. Within the scope of this study we focus on ‘open parts’ strategies.

2.2 Three forms of openness

In order to analyze openness of software (SW) and hardware (HW) components in close detail, we extend a framework proposed by West and O’Mahony (2008) to account for settings beyond software. While the authors distinguish between transparency and accessibility, we add replicability as a third form of openness.

Transparency (T) refers to the quantity and quality of information which is freely revealed to developers. Information in that sense could for example be software source code or hardware schematics and design files.

Accessibility (A) denotes the possibility for community members to actively participate in product development. This participation may happen through open discussions only or contributions could be directly taken up into official product releases.

Replicability (R) denotes the availability of individual components and thus the possibility for the self-assembly of the product. Objects including closed components could be copied if those components are obtainable; conversely objects which are entirely open source might not easily be copied if some components are difficult to produce and not obtainable from external suppliers.

2.3 The community perspective

Initially the free software and open source movement described itself as a community of programmers, committed to software freedom, and working against established intellectual property owners (cf. Stallman, 2007). With the emergence of open source business models the interests of the community and the commercial companies involved needed to be balanced (e.g., Mahony & Naughton, 2004).

West and O'Mahony (2008) find that by restricting access to community processes, firms limited their community's ability to attract new members and grow. Also Raasch et al. (2009, p. 389) observe an awareness that, "by deciding to 'leave enough room to encourage private investment', the community can improve its probability of success". People in charge try to promote project success by carefully weighing community and commercial requirements. First findings suggest that this trade-off is accepted by community members, as long as the balance is perceived to be fair.

2.4 Research hypotheses

In the realm of open design, the tangibility of the product may affect the form and degree of openness. For products that require heavy production cost and relatively little development cost, the open source approach could be less suitable (cf. Lee & Cole, 2003).

Therefore we propose that open design products which include both hardware and software components select differing degrees of openness for tangible and non-tangible parts of their design. We suggest that across all three forms of openness software components are more frequently open source than hardware components. The following research hypotheses obtain:

- H1-T: In open design software components are more transparent than hardware components, i.e. software source code is more frequently and more easily available than hardware documentation.

- H1-A: Software is more accessible than hardware, i.e. the community can exert more influence on software development than on hardware development.
- H1-R: Software components are more replicable than hardware components, i.e. software parts are more often available for self-assembly of the product than hardware components.

In contrast to pure software, tangible products need to be physically produced prior to being marketed. Unless community members assemble the good themselves, this production may be closed and left in the hands of a manufacturer reserving certain rights and appropriating (some portion of) the created value. As discussed in Section 2.3 prior findings suggest that communities prefer openness over closeness. This shall be investigated in our second set of hypotheses:

- H2-T: Transparency is important to open design communities.
- H2-A: Accessibility is important to open design communities.
- H2-R: Replicability is important to open design communities.

Taking our two sets of hypotheses together, we assume that software in general is more open than hardware in open design projects and that communities value openness. Looking at the vast number of software projects compared to the relatively small number of hardware projects (cf. Balka, Raasch, & Herstatt, 2009), we further assume that openness of software components is more important to communities than openness of hardware components:

- H3-T: Transparency of software components is more important to open design communities than transparency of hardware components.
- H3-A: Accessibility of software components is more important to open design communities than accessibility of hardware components.
- H3-R: Replicability of software components is more important to open design communities than replicability of hardware components.

3 Empirical research approach

3.1 Methodology and data collection

A web-based questionnaire survey among active participants of 20 open design communities was conducted in order to systematically explore the relevance of openness.

The selection of communities was a critical task. Possibly due to the novelty of the phenomenon, there is no complete directory of open design projects. For our case selection we followed Balka et al. (2009) and used the directory of 'Open Innovation Projects'. We carefully chose communities along three criteria: We selected projects (i) with more than 10 active participants developing (ii) objects which include both software and hardware components. Additionally, (iii) the development must have reached a stage in which first prototypes are available. This approach ensures a sufficiently large number of potential respondents and increases the availability of secondary information about the projects. A list of the surveyed communities is shown in Table 4.

A pre-test with 37 respondents delivering 22 full answers was conducted in August 2009 to ensure the validity of the items and to see how respondents react to the questionnaire (e.g., Garson, 2002). Since overlong internet questionnaires are often not completed (e.g. Batinic & Bosnjak, 2000), we plan for a completion time of about 5-7 minutes.

Data collection started on 2 September and lasted till 5 October 2009. The survey was announced on project mailing-lists and posted in forums and blogs. Moreover, a web-page was installed on which the goals of our research were explained. In order to increase the acceptability of the study, we strove to adhere to open source values by announcing that aggregated results of the study would be published on the Internet directly after completion.

The questionnaire included both multiple-choice and free-text questions. All items relating to the degree of openness were to be answered on 5-point Likert scales from "strongly disagree" (1) to "strongly agree" (5). The option "No answer" was available for every question.

3.2 The sample

During data collection we counted 688 unique visitors on the entry page, whereof 457 started the survey. 270 answers are sufficiently complete to be considered for further analysis, i.e. respondents finished at least two out of five sections from the survey. This results in a response rate of 39% when taking the number of visitors as target population (cf. Batinic & Bosnjak, 2000). Compared to similar studies (e.g. Roberts et al., 2006; von Krogh et al., 2009) this share appears satisfying. However, the number of answers per question varies between 189 and 270. The number of observations (n) therefore fluctuates. This is particularly the case for statistical analyses requiring pairs of observations to be complete.

270 participants (2% females, 98% males; mean age = 32 years, range:

14 –70 years) were included in the analysis. On average the participants are involved in their projects for about 16 months (range: less than 1 month to 6 years). Their positions in the project are 3% project leaders, 9% core team members, 55% developers and 33% users. On average they spend 9 hours per week active in the project community (range: 0 –70 hours).

3.3 Data analysis

Data has been analyzed using the statistical software R (R Development Core Team, 2005). Descriptive statistics about questionnaire items and constructs are presented by their means (μ) and variances (σ^2).

One- and two-sample t-tests have been conducted to confirm or reject our research hypotheses. We are aware that t-tests require normally distributed, independent samples, a requirement which is not fully met by our data set. Nevertheless, it can be assumed that t-tests provide reasonable results due to the large sample size. To support our findings we have additionally conducted one- and two-sample Wilcoxon tests (the latter is also known as 'Mann-Whitney' test) to prove our results by non-parametric statistical methods.

In order to assess the stability of our findings we repeated all t-tests and Wilcoxon tests on different sub-samples. The respondents of one community, respectively, were excluded and replaced by random answers from the remaining sample to check whether our results depend on the answers from single projects. All stability tests confirmed our findings showing p-values below 1%; only the p-value for H3-T rises to up to 10% when excluding the community "Openmoko" from the sample.

A potential non-response bias has been investigated by comparing the demographics of the respondents of our survey to similar studies (e.g., von Krogh et al., 2009). This did not yield significant differences.

4 Empirical findings

4.1 Software is more open than hardware

In our questionnaire 9 questions were included to analyze the degree of openness of software and hardware components respectively. The participants were asked to answer each question separately for software and hardware, therefore every questions has been included twice. 4 questions measured transparency, 3 evaluated accessibility, and 2 related to the replicability of the product or the components of the product.

Table 1 shows means and variances for these 9 questions. Question T - 3 has been put reversely; accordingly we expected its mean for software to be lower than its mean for hardware. For the other questions we expected the opposite. One-sided t-tests show significant differences at 1%-significance levels across all questions; one-sided wilcoxon tests confirm this finding.

	Software		Hardware		Difference		
Question	μ	σ^2	μ	σ^2	T-test	Wilcoxon	n
T - 1a	4.4	0.8	4.2	1.2	$p < 0.001$	$p < 0.001$	252
T - 1b	3.5	1.3	3.2	1.3	$p < 0.001$	$p < 0.001$	221
T - 2	4.1	1.0	3.9	1.3	$p < 0.001$	$p < 0.001$	249
T - 3*	3.0	1.1	3.0	1.1	$p < 0.01$	$p < 0.01$	200
A - 1	4.5	0.6	4.0	1.1	$p < 0.001$	$p < 0.001$	220
A - 2	4.4	0.7	3.9	1.1	$p < 0.001$	$p < 0.001$	228
A - 3	4.3	0.7	3.9	1.1	$p < 0.001$	$p < 0.001$	198
R - 1	4.4	0.8	3.3	2.0	$p < 0.001$	$p < 0.001$	222
R - 2	3.8	0.7	3.4	1.8	$p < 0.001$	$p < 0.001$	176

* reverse question

Table 1: Summary of degrees of openness for software and hardware components

For our hypotheses H1-T, H1-A, and H1-R we are interested in differences between software and hardware components concerning the entire constructs transparency, accessibility and replicability. Those constructs must be designed as linear combinations of the respective single questionnaire items, e.g., via factor analysis. As we can see from Table 1 significant differences in means between software and hardware can be observed for every single questionnaire item. This ensures strong support for our three hypotheses without calculating the constructs.

We hence conclude that software components in open design products are indeed more transparent, more accessible, and more replicable than hardware components.

4.2 Openness is important to open design communities

To analyze the importance of openness we included three questions in our survey, i.e. one per construct, again measuring this variable for software and hardware respectively. As we do not distinguish between software and hardware components in H2-T, H2-A, and H2-R, the respective results are merged through using their means for the analysis of this set of hypotheses.

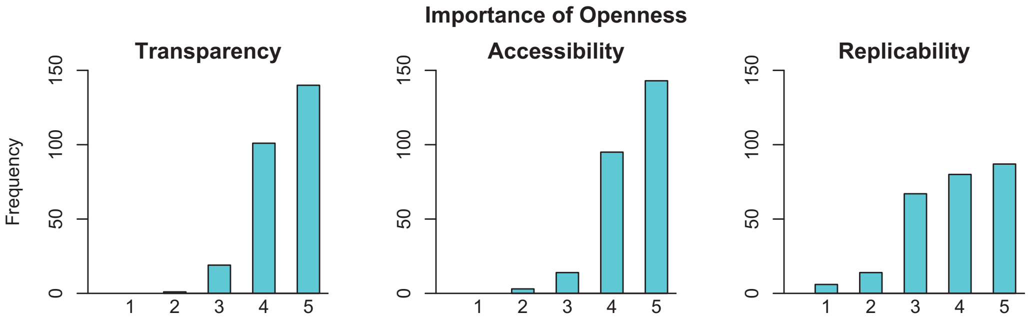

Figure 1: Histograms showing the frequency of answers from "Strongly Disagree" (1) to "Strongly Agree" (5) concerning the importance of openness across the three aspects

			$\mu > 3$		$\mu > 4$		
Constr.	μ	σ^2	T-test	Wilcoxon	T-test	Wilcoxon	n
T	4.5	0.4	$p < 0.001$	$p < 0.001$	$p < 0.001$	$p < 0.001$	261
A	4.5	0.4	$p < 0.001$	$p < 0.001$	$p < 0.001$	$p < 0.001$	255
R	3.9	1.0	$p < 0.001$	$p < 0.001$	n.s.	n.s.	254

Table 2: Summary of importance of openness

The histograms in Figure 1 show a strongly right-skewed distribution of answers. For every question a clear majority of participants states that openness is indeed important to them. To evaluate our hypotheses statistically, we calculate one-sided t-tests and Wilcoxon tests, testing for item means to be significantly higher than "Neutral" (3). As summarized in Table 2 we find all three hypotheses to be strongly supported at significance levels below 0.1%.

To better understand respondent's view of the importance of openness we repeat this procedure testing for item means to be significantly higher than "Agree" (4). For the constructs transparency and accessibility we still observe significant results, only the construct replicability does not show a mean significantly higher than 4.

Therefore we conclude that the availability of information and the opportunity to actively participate is indeed very important to open design communities. The possibility for self-assembly is also deemed important.

4.3 Openness of software components is more important than openness of hardware components

The third set of research hypotheses - H3-T, H3-A, and H3-R - is analyzed by handling the questionnaire items relating to the importance of openness separately for software and hardware.

As summarized in Table 3 one-sided t- and Mann-Whitney-tests indicate significantly higher means for software compared to hardware at significance levels below 1%. For a graphical presentation of the shift in answers between software and hardware, Figure 2 shows 100%-bar charts of the answers.

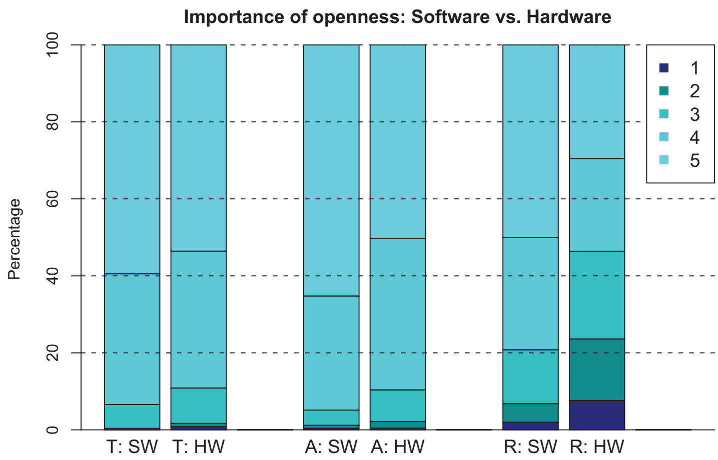

Figure 2: Graphical representation of differences in answers concerning the importance of openness for software and hardware

	Software		Hardware		Difference		
Construct	μ	σ^2	μ	σ^2	T-test	Wilcoxon	n
Transparency	4.5	0.4	4.4	0.6	$p < 0.01$	$p < 0.01$	237
Accessibility	4.6	0.4	4.4	0.6	$p < 0.001$	$p < 0.001$	229
Replicability	4.2	1.0	3.5	1.6	$p < 0.001$	$p < 0.001$	233

Table 3: Summary of differences in importance of openness between software and hardware

Accordingly our findings support all three hypotheses and we hence conclude that transparency, accessibility, and replicability of software components are indeed more important to community members than the same forms of openness of hardware components .

5 Discussion

5.1 Managerial implications

Our findings suggest that open parts strategies in open design are crafted at the component level, rather than the level of the entire design. Some parts of a design can be entirely closed, whereas others are opened up. In particular the degree of openness differs significantly between software and hardware components, in the sense that software is more transparent, accessible and replicable than hardware.

We also observe that openness indeed matters to community members. For all three forms (transparency, accessibility, and replicability), respondents declare that the degree of openness is important to them, albeit to different degrees. Again our results show significant differences between software and hardware components in this regard. Our analysis discloses that openness of software is significantly more important to community members than openness of hardware.

This suggests that companies working in open source settings can pursue differentiated strategies short of complete openness without alienating their developer communities. Every product whose design requires software and hardware development seems particularly suitable to this approach. Companies may accordingly involve communities into their software and parts of their hardware development and profit from the advantages of an open source approach, ranging from contributions from outside to increased publicity (cf. Bonaccorsi & Rossi, 2004). At the same time they can safeguard their position as manufacturers selling their product both to the community and the market. Firms in industries such as consumer electronics, telecommunication and IT hardware in particular may face opportunities for value capture from incorporating open source business models. Potentially they possess even more opportunities than firms in IT software as one might suspect. The different forms and degrees of openness yield a complex strategy space in which companies can position themselves. Thus they retain means of value capture and differentiation from competitors.

5.2 Limitations and future research

Before closing, we want to mention two important limitations of the present study. First, we focused on openness in development and production processes. It certainly would be desirable to replicate the presented study focusing on openness in organization and governance of open design projects. Second, we consciously limited our approach on analyzing open parts strategies. A close investigation of partly open strategies appears to be an interesting domain for future research. Beyond that, we experienced a lack of theoretical research on openness, which leaves opportunities for future research.

In conclusion, the present study has contributed to our understanding of the relevance of openness and has revealed important findings on different perceptions of openness between software and hardware. We hope that it helps to disclose business opportunities in the realm of open design and to stimulate further research on openness in this context.

References

Allen, N. J., & Meyer, J. P. (1990). The measurement and antecedents of affective, continuance, and normative commitment to the organization. *Journal of Occupational Psychology*, *63*, 1-18.

Arrow, K. (1962). Economic welfare and the allocation of resources for inventions. In R. Nelson (Ed.), *The rate and direction of economic activity.* Princeton, NJ: Princeton University Press.

Balka, K., Raasch, C., & Herstatt, C. (2009). Open source beyond software: An empirical investigation of the open design phenomenon. *In Proceedings of the R&D Management Conference*.

Batinic, B., & Bosnjak, M. (2000). Fragebogenuntersuchungen im Internet. In *Internet for psychologists* (p. 287-317). Hogrefe.

Bitzer, J. (2004). Commercial versus open source software: the role of product heterogeneity in competition. *Economic Systems*, *28*(4), 369-381.

Bonaccorsi, A., & Rossi, C. (2004). Altruistic individuals, selfish firms? the structure of motivation in open source software. *First Monday*, *9*(1).

Bonaccorsi, A., Rossi, C., & Giannangeli, S. (2006). Adaptive entry strategies under dominant standards: Hybrid business models in the open source software industry. *Management Science*, *52*, 1085 - 1098.

Dahlander, L. (2005). Appropriation and appropriability in open source software. *International Journal of Innovation Management*.

Demsetz, H. (1967). Toward a theory of property rights. *American Economic Review*, *57*(2), 347-359.
Franke, N., & Shah, S. (2003). How communities support innovative activities: an exploration of assistance and sharing among end-users. *Research policy*(1), 157-178.
Garson, D. G. (2002). *Guide to writing empirical papers, thesis, and dissertations.* New York: Marcel Dekker, Inc.
Gassmann, O. (2006). Opening up the innovation process: Towards an agenda. *R&D Management*, *36*(3), 223-228.
Harhoff, D., Henkel, J., & Hippel, E. von. (2003). Profiting from voluntary information spillovers: How users benefit by freely revealing their innovations. *Research Policy*, *32*, 1753-1769.
Hecker, F. (1999). Setting up shop: the business of open-source software. *IEEE Software*, *16*(1), 45-51.
Henkel, J. (2006). Selective revealing in open innovation processes: The case of embedded linux. *Research Policy*, *35*(7), 953-969.
Hope, J. (2004). *Open source biotechnology (phd thesis).* Canberra: The 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.
Lerner, J., & Tirole, J. (2001). The open source movement: Key research questions. *European Economic Review*, *45*(4-6), 819-826.
Mahony, I. G., & Naughton, E. J. (2004). Open source software monetized: out of the bazaar and into big business. *Computer and Internet Lawyer*, *21*(10), 1-17.
R Development Core Team. (2005). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved 6 February 2009, from http://www.R-project.org
Raasch, C., Herstatt, C., & Balka, K. (2009). On the open design of tangible goods. *R&D Management*, *39*(4), 382 - 393.
Roberts, J. A., Hann, I.-H., & Slaughter, S. A. (2006). Understanding the motivations, participation, and performance of open source software developers: A longitudinal study of apache projects. *Management Science*, *52*(7), 984-999.
Singhal, J., & Singhal, K. (2002). Supply chains and compatibility among components in product design. *Journal of Operations Management*, *20*, 289 - 302.
Stallman, R. (2007). Why open source misses the point of free software. Retrieved 5 October 2009, from http://www.gnu.org/philosophy/open-source-misses-the-point.html

Thomas, G. (2008). *Microsoft and mixed source software.* Retrieved 5 October 2009, from http://www.mylemonadestand.net/weblog/archives/12
Vallance, R., Kiani, S., & Nayfeh, S. (2001). *Open design of manufacturing equipment.*
von Krogh, G., Spaeth, S., Haefliger, S., & Wallin, M. (2008). Open source software: What we know (and do not know) about motives to contribute. *Dynamics of Institutions and Markets in Europe (DIME) working papers on intellectual property rights*. Retrieved 5 October 2009, from http://www.dime-eu.org/files/active/0/WP38_vonKroghSpaethHaefligerWallin_IPROSS.pdf
von Krogh, G., Spaeth, S., Stuermer, M., & Hertel, G. (2009). *Results of maemo and openmoko community survey.* Retrieved 6 October 2009, from http://public.smi.ethz.ch/files/MaemoOpenmoko/PublicDescriptiveStatistics.html
von Krogh, G., & von Hippel, E. (2003). Open source software and the "private-collective" innovation model: Issues for organization science. *Organization Science*, *14* (2), 208-223.
West, J. (2003). How open is open enough? melding proprietary and open source platform strategies. *Research Policy*, *32*, 1259-1285.
West, J., & O'Mahony, S. (2008). The role of participation architecture in growing sponsored open source communities. *Industry and Innovation*, *15* (2), 145-168.

Appendix

Openmoko	Always Innovating Touch Book	Fab@home	Gp2x
OpenEEG	One laptop per child	Chumby	RepRap
Bug Labs	Neuros OSD & Link	OpenServo	Balloon
Gumstix	Beagle Board	Open WRT	SquidBee
Mikrokopter	BitsFromBytes	MakerBot	Arduino

Table 4: List of surveyed communities