2007

To Open Source or not to Open Source: That's the Strategic Question. Results from a Survey Among Eight Leading Software Providers

M. Wijnen-Meijer
Wilhelmina van Pruisenweg, maarten.wijen-meijer@ictu.nl

Ronald Batenburg
Utrecht University, r.s.batenburg@cs.uu.nl

TO OPEN SOURCE OR NOT TO OPEN SOURCE: THAT'S THE STRATEGIC QUESTION

RESULTS FROM A SURVEY AMONG EIGHT LEADING SOFTWARE PROVIDERS

Wijnen-Meijer, Maarten, OSOSS, Wilhelmina van Pruisenweg 104, The Hague, The Netherlands, maarten.wijnen-meijer@ictu.nl

Batenburg, Ronald, Institute of Information and Computing Science, Utrecht University, Padualaan 14, Utrecht, The Netherlands, r.s.batenburg@cs.uu.nl

Abstract

The central goal of this paper is to explore the reasons software providers open source their own proprietary software. This was done by examining the perceived effects of this decision from the viewpoint of the software providers. A substantial number of perceived effects of open sourcing were derived, as well as possibly relevant product characteristics. In face to face interviews with high representatives of eight of the largest software vendors in the Netherlands (Apple, Microsoft, Oracle, Hewlett-Packard, IBM, SAP, Compuware and Computer Associates), the perceived effects and the actual open source decision of 21 different software products of these companies were queried. Five of these products were indeed open source software. From the interview data it appears that the main effects of open sourcing a software product are perceived to be an increase in product value, possibilities for peer review, user testing, fragmentation and product usage. The greater the perceived effect of open sourcing a product on the profitability and the mobilization of external R&D the more likely a product is to be open sourced. Furthermore, open sourcing seems to be more common for products with a cost leadership strategy than for products with a product leadership strategy.

Keywords: Open source, open sourcing, software product strategy ,licensing.

1 INTRODUCTION

It goes without saying that open source software has become a major trend within the IT industry; and beyond. A survey of Forrester Research in 2004 among 140 North American firms showed that 46% of these firms are using open source software today, and 14% plan to use it in the future (Giera, 2004). Half of these deployments is for mission critical applications. The open source applications used most are Linux, Apache, MySQL and Tomcat. A European survey among 955 governments in 2005 has shown that 79% of them use open source software. Linux is used by 47% of all European local governments, 34% use MySQL, 33% use Apache, 26% use Mozilla. 24% use PHP and 22% use OpenOffic.org (Ghosh et al., 2005). Software such as Apache, Bind, and Sendmail have a market share well over 50% (Spiller et al., 2002).

There are many reasons why software users deploy open source software. Ghosh et al. (2005) identified the current prime reasons: cost effectiveness, improved flexibility, expiration of maintenance, availability of support through software vendors, independence from software vendors, increased technical requirements, increased interoperability, security aspects and improved reliability (cf. Wichmann, 2002a). Governments value open source software for its conformance to open standards that help to ensure accessibility of governmental information (Varian et al., 2003).

Contrary to the previous, the reasons why software providers contribute (or do not contribute) to the open source market is hardly studied nor is it thoroughly understood. Some software providers seem to use open source software internally because of its perceived cost effectiveness (Grand et al., 2004). Other software providers offer third-party open source software to their customers in order to facilitate the usage of own their closed source software. Still others open source their own closed software products. A classic example is the open sourcing of the Netscape browser by Netscape Corporation in 1998. A fairly new but potentially important example is the Java programming language by Sun.

When software providers contemplate the option of open sourcing their proprietary software they must predict the effects this decision can have. There are many possible effects on many different characteristics in many different areas. For instance, open sourcing a product might increase the development pace and product quality but at the same time decrease profitability. The final decision of a software provider whether or not to open source a particular software product involves the trade-off of many benefits and drawbacks. In order to understand the motivations of software providers with regard to open source, it is vital to understand the perceptions software providers have of the effects that open sourcing a software product will have. This drives the research question of this paper:

What do software providers perceive as the effects of open sourcing of their proprietary software and how does this influence their decision to open source their proprietary software or not? Also, how do the product characteristics influence this decision?

In the following section, a conceptual model is developed based on existing literature to structure this decision making process. A number of hypotheses about the potential effects of factors and conditions for open sourcing is derived from this model. Then it is described how representatives of Apple, CA, Compuware, HP, IBM, Microsoft, Oracle, and SAP were interviewed to collect data with regard to both open source and closed source products. The results were used to test the hypotheses. We consider this an important contribution to the understanding of commercial involvement in open source software development. Although there is a lot of theory about the development of open source software, including the assumed commercial benefits, there is very limited empirical research that validates these claims (cf. Rossi & Bonaccorsi, 2005). The closing section reflects on our empirical results, i.e. the finding of what factors and conditions appear to determine a software vendor's decision to open source its proprietary software – or not.

2 LITERATURE STUDY AND THEORY

In this section we present an exploration of the literature to collect the effects software providers might perceive when contemplating the decision whether or not to open source a proprietary software product. We define these perceptions in terms of four key aspects: product innovation, independent customers, open companies and source of competition. In addition, we propose several product characteristics that might be relevant for the decision of open sourcing as well.

2.1 Product innovation

Perhaps the most important aspect of open source software is the capability to innovate through "common based peer production". While traditional R&D is performed isolated and secretly inside companies, an increasing number of scholars as Chesbrough (2005) promote a model of open innovation. The idea of open innovation is well described in "Innovation happens elsewhere" by Goldman & Gabriel (2005). The gist of this book is that software providers can successfully work together in open source projects to create innovations upon which new (proprietary) products and services can be built. This argument can be recognized at Computer Associates (CA) which open sourced its Ingres database in 2004. CA (2004) indicated that open sourcing this database "stems from the desire to foster innovation. By encouraging the participation of outside contributors, CA allowed more ambitious projects within the Ingres community and new features to be released to the user base much faster." Maximizing intellectual property value does not always mean keeping it from others (Shapiro, 1999). Collaboration with the open source community is perceived by some as beneficial.

The benefits of open sourcing proprietary software in terms of product innovation might thus be perceived as: an increase in external R&D activities, more possibilities for peer review and user testing, an increase in the product quality, increased efficiency through heightened internal programming discipline as code is public, and an increased attractiveness for skilled programmers.

2.2 Independent customers

A second characteristic of open source software is the lack of vendor lock-in (Pavlicek, 2000; Raymond, 2001; Wichmann, 2002b; Goldman et al., 2005). Vendor lock-in is the result of high switching costs due to the usage of proprietary and non-standard software that is only available through one vendor. Open source software tends to be compliant with open standards and open interfaces (Varian et al., 2003) which eases migration to competing products. In principle, open source software can be adapted and developed by others. Users of open source software are generally not dependent on a single vendor to provide updates, implement features or provide compatibility with other products that the user might have. The more users, the more open source software can provide this value to users clearly benefiting from the network effect. For public organizations in particular, prevention of vendor lock-in is very important. The fact that software providers cannot lock-in their customers, does not mean they cannot generate customer loyalty. One way for software providers to generating customer loyalty and differentiate themselves from others is branding and reputation.

Based on literature the effects software providers might perceive from open sourcing a closed source software products in terms of independent customers can be: an increase in support and loyalty from software users, opportunities to easily redraw end-of-life products while enabling customers to support themselves, an increase in reputation, user base, and usage and hence greater value for customers.

2.3 New business models

Open sourcing a closed source software product can have effects that go beyond the product itself. One of the major indirect effects is an increased demand for complementary services and products once proprietary software is open sourced (Raymond, 2001; Feller et al., 2002; Wichmann, 2002b; Fink, 2003; Henkel, 2003; Von Hippel et al., 2003; Dahlander, 2004; Weber, 2004; Hang, 2005).

Open sourcing software creates spillovers from which other companies can benefit. This sometimes creates the impression that open sourcing creates a truly level playing field. But open sourcing also means that software providers can gain new sources of income through the sales of complementary products and services that can potentially make up for this loss in original product sales (Fink, 2003). The ability to create a new business model basically determines whether open sourcing a specific product is truly profitable and how it will impacts the bottom-line of the software provider (Henkel, 2003; Hang, 2005). We distinguish four types of business models:

- The first business model is based on open source services. Raymond (2001) has called this model “give away a recipe, open a restaurant”. Software providers open source their software and provide commercial support. The open sourcing of the Ingres database by Computer Associates (2004) illustrates business model: CA has been focusing on services, support and certification around their open source application. With this model a software provider takes more the role of an integrator and benefits through services, branding and reputation.
- Secondly, a business model is to open source a base products but sell proprietary additions. In this case the open source software is used as a ‘loss leader’. Fink (2003) points out that this model can only be successful as long as no open source equivalents for these proprietary additions are developed. The loss leader model is only effective if the loss leader fuels demand for the non-gratis and proprietary additions a software provider has to offer. This model can be illustrated with many of the open source integrated development environments, which enforce the position of underlying toolkits, framework and languages.
- A third potential business model is to create open and closed source siblings. In some cases open source products are not complemented by different products as with the previous model, but by nearly identical closed source products with clear added value. This model is known as dual licensing. Sun Microsystems has been actively exploring this business model. Sun has open sourced StarOffice into Openoffice.org and Solaris into OpenSolaris, but they still continue to sell the

original closed source products with extra features, indemnification and support (Goldman et al., 2005). Another well known example is the MySQL database.

- Fourth and final, hardware sales can be the angle of a business model. Open source software has effectively been used to increase hardware sales as well. With this model a company is not so much a software provider financing its open source software development by selling hardware, but rather a hardware provider trying to sell more hardware by offering open source software to reduce license costs. As argued above, successful open source business models utilize open source to complement the differentiating and profit making aspects of their business.

Here, we summarize the benefits software providers might perceive based on literature in terms of new business models: an increase in revenue from sales of related services, increased revenue from sales of related products, and hence an increase in profitability and reputation.

2.4 Source of competition

Open sourcing a closed source product can have significant impact on the competitive relations between software providers. Some software providers hope that open sourcing a products helps to make it a standard that will be supported by other software providers as well (Wichmann, 2002b; Fink, 2003; Harhoff et al., 2003; Von Hippel et al., 2003; Goldman et al., 2005; West & Gallagher; 2005). When IBM open sourced its RDC speech recognition development tools, the company's press release (2004) stated that “the initiative, supported by more than 20 key industry players from speech vendors to platform providers, is aimed at ending the battles over competing, proprietary specifications”.

Open sourcing a non-gratis closed source software product can be used to create pricing pressure on competitors, as open source software has no license fees (Wichmann, 2002b; Fink, 2003; Henkel, 2003). The pressure on competitors is augmented by the fact that open sourcing software might also lower the barriers of entry to a market (O’Reilly, 1999; Fink, 2003). New communities can build custom version of open sourced software and new companies might try to sell and support these. A good example of how open sourcing software can create diversification on a common code base is the Eclipse project. Eclipse is an integrated development environment which was open sourced by IBM in 2001. Today it is used by many software providers and open source communities as a basis for their own integrated development environments to which they add extra open source or proprietary modules to support new functionality.

To summarize: the benefits software providers perceive from open sourcing, in terms of sources of competition, can be defined as: an increase in opportunities to create a market standard, increased pricing pressure on competitors and an lowered barriers of entry.

2.5 Product characteristics

In addition to the above effects for open sourcing, we assume that software providers also take certain product characteristics into account when deciding about open sourcing. Literature on this subject is sparse. In The Business of Software (2004) Cusumano describes several product characteristics that may prove to be relevant for the decision of open sourcing:

- Cusumano discriminates between two fundamental different intended audiences: enterprises and home users. As enterprises are usually willing to pay for detailed product documentation and expert technical support, the open source business models (2.3) can be expected to be of most relevance for enterprises.
- Cusumano distinguishes niche and mass audiences, and software with a horizontal or a vertical functional scope. Horizontal applications perform generic tasks and can be deployed very broadly. Vertical applications on the other hand perform specific functions that support specific business processes. Software that is geared towards a mass market or that provides horizontal functionality will have a broader appeal and may thus be much more likely to attract outside developers.
- Another dimension is the market position of a software product. This can be either leading, complementary of following. Software is leading if it has the largest market share or if it is at the forefront of technological development. Complementary software is software that is meant to be used in conjunction with these leading products. Following software is software that is not that often used

and lacks advanced features. In the case of leading software it is clear that a software provider would probably loose much more than it would stand to gain by open sourcing the software. It is highly unlikely software providers would open source a product with a large market share that drives an ecosystem of complementary products or that has distinct technical capabilities that a select set of customers would most likely be glad to pay for.

With respect to software and information products in general, Shapiro & Varian (1999) recognize two fundamental competitive strategies: product differentiation or cost leadership. Product differentiation is often referred to as product leadership. These two strategies are considered to be incompatible and software providers are forced to either focus on added value by offering distinct and advanced products, or on a low license fee for a normal product by using economies of scale. It is most probable that products with a cost leadership strategy are more likely to be open sourced.

Therefore, it can be expected that software providers perceive benefits from open source if they consider software products with a horizontal functional scope for a mass market, consider complementary or following software or products which follow a cost leadership strategy.

3 DATA AND METHOD

In the previous sections, the perceived effects on open sourcing have been defined and related expectations have been formulated. In this section we will focus on the validation of these definitions and expectations by describing the research method, the construction of a survey and the selection of a group of software providers that have participated in this research. The results of the empirical research are then presented in the next section.

At this point it is useful to define ‘a software provider’. The literature that has been used in deriving our hypotheses does not contain a uniform definition for software provider. It is obvious however, that all software providers have in common that they sell one or more software products. The concept of software product has been defined by Xu et al. (2005) as “a packaged configuration of software components or a software-based service, with auxiliary materials, which is released for and traded in a specific market”. For the purpose of this study, organisations can be considered software providers if they are a commercial entity that license and distribute their own software to a broad audience as an independent product. The definition of software providers does not require software development and sales to be the most dominant line of business though.

As it is impossible to survey all software providers about all their products. In this research we have focused on large software providers. Their size makes them more interesting in terms of their credibility and opinion. Large software providers might be setting an example for the policy of smaller software providers. Also, since the software products of large software providers are more likely to be widely used, the impact on the software market and the user experience is much larger when these products are open sourced. Hence, the opinion of large software providers has a significant effect on the adoption and development of open source software in general. Focusing on large software providers opens an opportunity to generalize the results by only using a small number of respondents.

For practical reasons the scope of the empirical research has been limited to The Netherlands. Only large software providers active in this country are included in the survey. The Dutch Computable magazine (2004) creates a yearly list of large IT companies specifically for The Netherlands, called “The Computable 100”. This list ranks the top 100 IT companies based on their estimated revenue, net profit and number of employees in The Netherlands. From this list of software providers their main closed and open source software products are coded as closed or open sourced and categorized into software product categories. From this exercise (on an aggregated level) three software product categories have been found that allow for the comparison of more than one open sourced software product with a considerable number of closed source software products: (1) operating systems (OS), (2) databases (DB) and (3) integrated development environments (IDE). This makes it possible to compare the software products within the software categories and between software categories as well.

Eight large software providers were prepared to participate in the survey. In total 21 software products have been selected for which the decision of open sourcing has been investigated. Hence, these 21 observations are the level of the following analysis. Five of these products have indeed been open sourced in the period of 2000 to 2004. The following table gives an overview of the software providers and software products by product category and whether or not the software was open sourced.

	OS		Database		IDE	
Software provider	Closed	Open	Closed	Open	Closed	Open
Hewlett-Packard	HP-UX, Tru64 OpenVMS					
IBM	AIX		DB2, Informix	Cloudscape	Rational	Eclipse
SAP				Sap DB/Maxdb	Netweaver	
Microsoft	Windows CE		MSsql		Visual studio	
Oracle			Oracle		Jdeveloper	
Compuware					OptimalJ	
Apple	Mac OS X	Darwin			Xcode	
Computer Associates				Ingres		

Table 1. Selected closed and open sourced software by software provider and product category.

As can be noticed from media sources, some of the software providers are actively involved in open source software such as HP (Feller et al., 2002; Wichmann, 2002b), IBM (Feller et al., 2002; West, 2003; Weber, 2004), SAP (Wichmann, 2002b), Apple (West, 2003; Weber, 2004), and CA (Wichmann, 2002b). Extensive desk and media research was performed to collect the necessary context information for our field work. If we summarize the current policy position of the eight software companies with respect to open source and open sourcing, it becomes clear that open source is on the radar of many software providers. It is also clear however, that open source is not replacing closed source software in any way. Instead, software providers seem to be experimenting with ways in which they can benefit from open source software without endangering their other products. Hence, each of the selected companies provide an interesting context and case for our interviews and data collection.

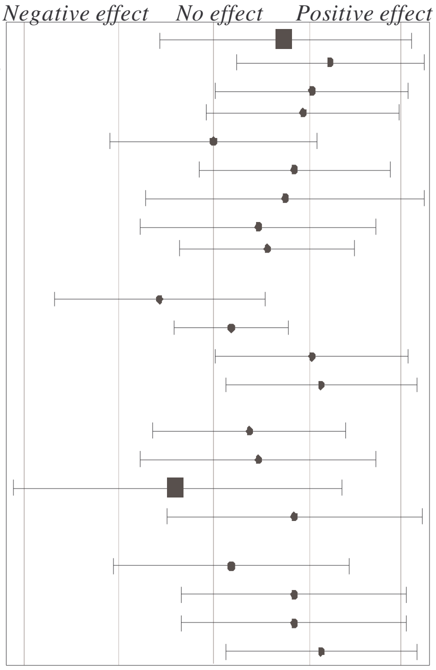
In our field work, high-ranked representatives of each of the eight software providers presented above were interviewed in mostly two-hours sessions during winter 2004 and spring 2005. The respondents completed a questionnaire for each software product that was identified for their company. So while the HP-respondent was interviewed for three software products, the CA-respondent was only interviewed for one product. With regard to the potential effects (section 2.1 – 2.4) of open sourcing questions were asked in the following format: “What effect does open sourcing your product have on the exploitation of external knowledge and R&D activities?”. Answer categories were pre-coded as “positive”, “negative” and “no effect“. In addition, a question was posed about the importance of this effect: “How important is this for the decision of open sourcing?”, with answer categories “not important”, “moderately important”, “important”. In case the software in question was open source the respondents were asked to take themselves back to the moment the decision of open sourcing was made. In the case of closed source software the respondents were asked to formulate company policy.

The product characteristics (section 2.5) for open sourcing were measured through direct questions about the intended type of users of the particular software product. An survey question example is: “What type of user is your product intended for – home users?”, with answer categories “yes” and “no”. Finally, questions about the company and open source licensing were posed, if applicable. During the meeting the respondents were explained how to interpret the questions and were asked to talk out loud while completing the survey while their remarks were being recorded on tape. At the end of the session the survey was collected. A few weeks later all respondents received a digital copy of the survey with their answers filled out and with a summary of their remarks. They were asked to verify that the answers and remarks were indeed correct and had been correctly recorded and summarized. In the case of Microsoft the session was repeated as the respondent felt uncomfortable with the initial answers. All respondents encountered no problems completing the survey.

4 RESULTS

4.1 The perceived effects of open sourcing

We start with presenting an overview of the perceived effects of open sourcing (section 2.1 – 2.4) as presented above. Base of the statistics are the judgements of the eight respondents for the 21 software products they collectively market. The answers to the particular questions, "negative effect", "no effect" and "positive effect", were coded as 1, 2 and 3 respectively.



What effect does open sourcing your product have on …:	*a*	*s*	*p*
(2.1) Product innovation	2.38	0.67	0.05
- the exploitation of external knowledge and R&D activities	2.62	0.50	0.14
- the possibilities for peer review (4.1.1)	2.52	0.51	0.05
- the possibilities for user testing (4.1.3)	2.48	0.51	0.25
- the product quality	2.00	0.55	0.81
- the product security	2.43	0.51	0.16
- the development costs	2.38	0.74	0.33
- the development pace	2.24	0.62	0.31
- your attractiveness for programmers	2.29	0.46	0.22
- your programming discipline			
(2.2) Independent customers	1.71	0.56	0.60
- vendor lock-in	2.10	0.30	0.97
- the possibilities for product support	2.52	0.51	0.90
- the value of your product (4.1.4)	2.57	0.51	0.09
- the usage of the software (4.1.2)			
(2.3) New business models	2.19	0.51	0.81
- your revenue of selling related services	2.24	0.62	0.70
- your revenue of selling related products	1.81	0.87	0.05
- your profitability	2.43	0.68	0.05
- your reputation			
(2.4) Source of competition	2.10	0.62	0.44
- making your product a market standard	2.43	0.60	0.74
- the pricing pressure for competitors	1.57	0.60	0.25
- the barriers of entry	2.57	0.51	0.71

Figure 1. Perceived effects of open sourcing software products as judged by eight representatives from eight different companies over 21 different software products.

Figure 1 lists the average value (a) and the standard deviation (s) for the perceived effects of open sourcing. The rangebars show the average value as a dot or square with one standard deviation on both sides. Figure 1 also lists the significance (p) of a Pearson's chi-square test with the answers and whether or not the software is open source. In case of a significant association ($p \leq 0.05$) the dot in the rangebar has been replaced by a square. In fact, the perceived effect of open sourcing on (1) the exploitation of external knowledge and R&D activities, and (2) the profitability of the software provider are positively associated with the software being open source. In other words, the more a software provider perceives a positive effect of open sourcing on the exploitation of external knowledge and R&D, or on his profitability, the more likely the software is to be open sourced.

As discussed in section 2.1 open source is a form of common-based peer production. Open sourcing is only useful if it results in external involvement. If there are no external contributions open source development resembles proprietary development conducted in a glass house (West et al., 2005). It would be very sensible for software providers to base the decision of open sourcing on the perceived amount of external R&D. In paragraph 2.2 it was discussed how open sourcing creates spill-overs. This is costly as the software is then freely available to customers and competitors. Software providers can nevertheless derive private benefits from open sourcing by using appropriate business models. Profitability is an aggregated metric in which these costs and benefits are balanced. As software providers are commercial enterprises that are motivated by profit it is very natural that the decision of open sourcing is dependent on the perceived effects of open sourcing on profitability.

Besides the two effects described, there are also five effects that are not significantly associated with the decision of open sourcing are nevertheless important for our analysis due to their size:

- *Possibilities for peer review.* In general software providers agree that open sourcing increases the possibilities for peer review. One respondent pointed out that this effect is always positive when the source was initially closed. Another respondent however indicated that the source code of his company's proprietary software applications was already available to customers under proprietary licenses. In those cases the respondent perceived no additional effects of open sourcing on peer review. While the possibilities for peer review increase it is doubtful to many respondents whether this will result in more contributions. A high learning curve is often cited as a detergent.
- *Fragmentation.* Most software providers perceive fragmentation as a drawback of open sourcing. One respondent claimed that open sourcing some of his company's products would not so much fragment the product itself, but rather fragment the market place because the software products would then be in direct competition with a leading open source software project. This was counter to the strategy of his company, as it was very much investing in this open source alternative. Another respondent claimed that although forking is possible, it is would be in nobody's interest to fork his company's open sourced software product as many similar applications exist and the application itself is already available for many different platforms. And even if fragmentation would occur, he perceived that there would be no confusion for the customers as his company owns the trademarks of the official version. A third respondent pointed out that fragmentation is not always possible to avoid. As open source software is integrated into other products, there will always be differences between versions.
- *Product usage.* Software providers perceive that open sourcing increases the usage of the open sourced product. One respondent indicated that open source applications can be ported to new platforms by the community. This could increase the amount of potential customers. The same respondent indicated that open sourcing an integrated development environment could not only have a beneficial effect on the usage of the IDE itself, but as the IDE is used to create applications for a specific platform, it could increase the usage of that platform as well. Another respondent indicated that the increased usage caused by open source will be the largest in the lower market segment since the product becomes basically gratis. He predicted that open sourced software stands a great chance of being fragmented into multiple incompatible versions however. While the overall usage might increase, the increased usage has to be divided over several fragmented versions, so usage of the original product might actually decrease. A better tactic to increase usage in the lower market segments in his opinion would be to create special gratis proprietary express versions of high end products.
- *Possibilities for user testing.* Software providers in general agree that open sourcing increases the possibilities for user testing. One respondent however indicated that users can normally test almost all proprietary software for free. For his company testing is free of license restrictions, even in the case of large enterprise applications. If users indicate they want to test an application, it is installed and configured for testing without costs. Another respondent pointed out that his company uses large pools of functional beta testers to test its proprietary products before they are released to the general public. He indicated that open sourcing would not aid user testing in those circumstances. A third respondent stated that the customers of his company's open source product seem to feel more responsibility for the product when it is open source and are therefore more eager to test the product out and give feedback and bug reports.
- *Product value.* Software providers consort that open sourcing a software product increases the value of the product. For two respondents this is mainly related to reputation, albeit in two different ways. One of them indicated that his company would stand to lose its reputation when it would engage in open source software development. As a result the value of his company's products would decrease. In the long run however he judged that the value of the products could increase due to support of the community and the creation of open source add-ons. For another respondent the short term effect of open sourcing is the opposite, namely positive press, which would increase his company's reputation. In the long run the value would increase even more because in his opinion customers value open source software as more secure.

4.2 Relationships between perceived effects

Apart from their means and standard deviations, do we see clusters of perceived effects along other lines than the initial four classes we defined? Are these effects single aspects of the respondent's open source perception or can new types of considerations be discovered from our data? To explore this, the following picture is presented.

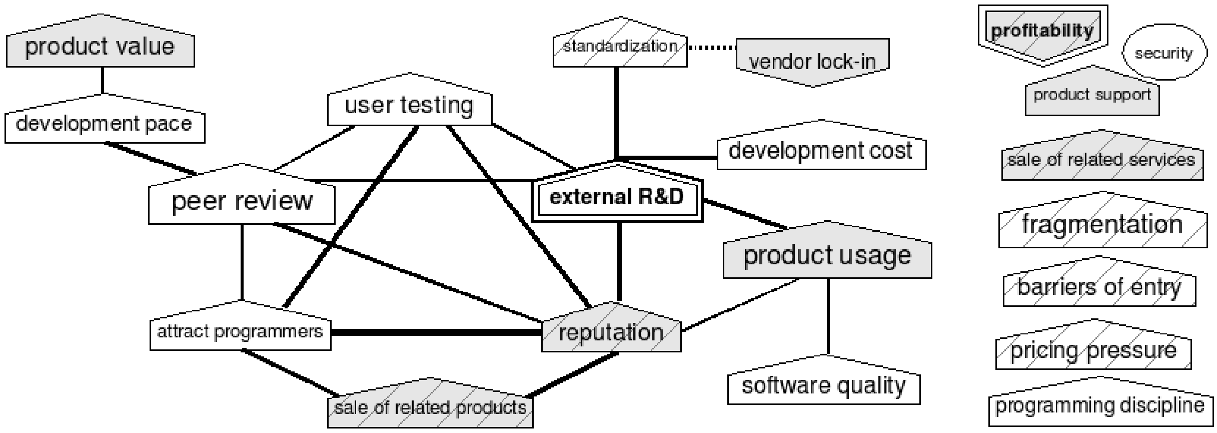


Figure 2. Interrelations between the perceived effects of open sourcing software products.

The perceived effects that have a double border are significantly associated with whether or not the software is open source sourced. The different background colours and shading identify different categories (section 2.1-2.4). If the effect is positive, the shape of the effect points upwards. If the effect is negative, the shape points downwards. If there is no effect the shape is oval. The size of the shape and the font are proportional to the magnitude of the effect. The larger the shape and the font, the larger the effect, both positive or negative. The effects are interconnected with a line in cases where the Pearson's chi-square efficient of association between the effects is significant ($p \leq 0.01$). The width of this line is proportional to the size of the Pearson chi-square. The wider the line, the more the two perceived effects are associated. The line is solid in case the association is positive, the line is dotted in case the association is negative. From this plot two prominent clusters can be recognized:

- *Product innovation.* The perceived effects related to product innovation (section 2.1; the ones with the white background pattern) form the heart of this cluster in the sense that they are very much interrelated with each other and many of the perceived effects from other categories (section 2.2 – 2.4). These perceived effects also tend to be somewhat larger on average (a = 2.44) than the other effects that inside the cluster (a = 2.20). Very much central to the cluster are peer review and external R&D. These two are connected by user testing. All three characteristics represent external involvement which touted as the most important aspect of open source development, albeit on different levels. Together these accelerate the development pace, reduce development costs, increase the dissemination of the software, foster standardization and attract new programmers.
- *Reputation*. Reputation is very central to the cluster as well. Although reputation may not be a motivation in itself to engage in open source software development, reputation is very instrumental since it is a visible tag that can be used by others to coordinate their involvement. For instance, developers might use this tag to decide where they want to work for a software provider (attractiveness) whether they want to contribute to an open source project (peer review), other software providers might want to contribute their expertise to a project based on the reputation of a software provider (external R&D) and users might use this tag while selecting the software they want to use (product usage and user testing) or buy (related products).

Another interesting association is that vendor lock-in is negatively associated with standardization. This can be explained, as standardization means that software providers compete on the quality of the implementation of a standard, rather that compete on standards. Users can therefore more easily switch between products supporting the same standards or products having the same source code base.

In addition we see that product value and development pace are positively correlated. This association could indicate that software providers believe that their customers value the faster development cycle that is associated with open sourcing. The fact that product value is only associated with development pace would then suggest that software providers do not believe that the other perceived effects of open sourcing influence the product value, for instance product quality, or product support.

Finally it appears that there is a significant association between selling related products and attracting programmers. This association is conspicuous and difficult to explain. One could suggest that this association is mediated by the perceived effect on reputation. An increase in reputation would have an effect on both customers, as symbolized by the sales of related products, as well as programmer in the sense that they are more willing to work for open source software providers.

4.3 The importance of product characteristics

Now that we know what perceptions software providers have with respect to open sourcing, the next step is to focus on the actual open sourcing decision of these companies. Although the previous section showed that many perceptions are relevant (and coincide in clusters as well) the plain fact is that 'only' five out of the 21 software products we studied were actually open sourced. In other words: there are many products for which respondents perceive a positive effect of open sourcing it, but that are not open sourced. In this section we investigate which product characteristics (section 2.5) influence this decision.

Question	Answers	chi	p
What type of user is your product intended for?			
- home users	(1) Yes, (2) No	0.0	.967
- corporate end-users	(1) Yes, (2) No	0.8	.365
- developers	(1) Yes, (2) No	0.1	.710
- administrators	(1) Yes, (2) No	0.0	.903
What is the market for your product?	(1) Mass, (2) Niche	0.0	.856
What is the functional scope of your product?	(1) Horizontal, (2) Vertical	0.8	.365
How would you characterize your product with respect to other products in the market?	(1) Leader, (2) Follower, (3) Complementor	3.2	.204
What is the strategic objective of your software product?	(1) Product leadership, (2) Cost leadership	4.0	.046†

Table 2. Product characteristics and their relation with the open/closed source decision.

The table above presents the association between product characteristics and the whether the software is open of closed. It is clear that only one product characteristic has a clear relationship with the decision of open sourcing: the strategic objective of the product. Additional analysis of the underlying contingency table shows that products that follow a cost leadership strategy are far more likely to be open sourced than products with a product leadership strategy. Four out of the five open source products that have been included in this survey follow a cost leadership strategy. This result blends with the previous results that an increase in external R&D investments is a central perceived effect of open sourcing. Software vendors that follow a cost leadership strategy with regard to a certain software product will focus on a low license fee. Hence, they can utilize external R&D to improve these products. In contrast, expensive and exclusive product features are pivotal to a product leadership strategy. Software providers that follow a product leadership strategy with regard to a certain software product, have consequently less to gain from external R&D, as they are already (internally) differentiated and developed with respect to features.

5 CONCLUSION AND FURTHER RESEARCH

5.1 Summary and conclusions

We draw back on the central goals of this paper, i.e. (1) to explore the perceived effects of open sourcing software products by providers, and (2) to analyse the relevance of these effects on the open source decision by software providers, and (3) to examine the influence of product characteristics. Based on the current academic literature and practitioner's publications, a substantial number of perceived effects of open sourcing were derived, including a number of product characteristic that are potentially relevant for the decision of open sourcing. Based on face to face interviews with high representatives of eight of the largest software vendors in the Netherlands (Apple, Microsoft, Oracle, Hewlett Packard, IBM, SAP, Compuware and Computer Associates), the perceived effects and the actual open source decision of 21 different software products of these companies were queried (including five open source software products).

From the interview data it appears that the main effects of open sourcing a software product are perceived to be an increase in product value, possibilities for peer review, user testing, fragmentation and product usage. These perceived effects are partly interrelated and corroborate with the mobilization of (more) external R&D and (more) reputation as well. The greater the perceived effect of open sourcing a product on the profitability and the mobilization of external R&D the more likely a product is to be open sourced. Furthermore, open sourcing seems to be more common for products with a cost leadership strategy than for products with a product leadership strategy.

In reflection on these results, a first critical remark concerns our sample size. As we cover the top players in the Dutch software industry, our empirical base is too small (and biased) to be generalized over a larger set of software providers. The results apply to only three categories of software products. Due to the sample size the differences between product categories cannot be determined. On the other hand, our study can be considered as one of the first attempts to empirically investigate the numerous statements, observations and explicit hypotheses about the drivers and barriers of open sourcing.

5.2 Further research

This paper focuses on the business rationale of open sourcing only. We believe there are many opportunities for future research in this area. The exploratory overview of the issues that are relevant to the decision of open sourcing can be a starting point for further case based research in which the decision of open sourcing is studied in more detail. This research could yield further qualitative and descriptive data, not only about perception and motivation, but also about the decision making process itself. Another interesting question is to what extend the success of company initiated open source projects is indeed determined by the effects as provided in this paper.

The details of open source licensing are very important when open sourcing software (Lerner et al., 2002). Our survey included questions on why an open source license was chosen, but the results were not conclusive due to the sample size. This topic could be address in further research, Also, it is clear that open source is just one particular regime for sharing source code. Many more can be imagined and indeed several software providers have tried to create their own regimes, which are sometimes called "gated communities" as they are only open to some extent (West, 2003; Laurent, 2004; Goldman et al., 2005). In this paper, open source and proprietary software are treated as a dichotomy. Further research might explore the motivations for software providers to engage in other property regimes as well.

Finally, open sourcing somewhat resembles outsourcing. Although outsourcing contracts are bilateral and very detailed about the activities that have to be performed, the parallels are striking and imply interesting trajectories for further research. In both cases there is a decision to be made whether certain activities need to be performed internally or that they can safely be externalized. It is worthwhile to study the similarities and differences between these concepts in relation to these decisions.

References

Chesbrough, H. (2003). Open Innovation. The New Imperative for Creating and Profiting from Technology. Harvard Business School Press. ISBN 1-57851-837-7.

Computable (2004). Top 100 Krachtigste ict-bedrijven. [http://www.computable.nl/computable100/]

Computer Associates. (2004). Ingres and Open Source: Securing a Long Term Future. [http://opensource.ca.com/projects/Ingres/documents/whitepapers/opensourcefuture]

Cusumano, M.A. (2004). The Business of Software. Free Press. ISBN 0-7432-1580-X.

Dahlander, L. (2004). Appropriating the Commons: Firms in Open Source Software. MIT Working Paper [http://opensource.mit.edu/papers/dahlander2.pdf]

Feller, J. & Fitzgerald B. (2002). Understanding open source software development

Fink, M. (2003). The Business and Economics of Linux and Open Source. isbn://0130476773

Ghosh, R. & Glott, R. (2005). FLOSSPOLS Deliverable D3: Results and policy paper from survey of government authorities [http://flosspols.org/deliverables/]

Goldman, R. & Gabriel R. (2005). Innovation happens elsewhere. Open source as business strategy

Grand, S., Krogh, G. v., Leonard, D. & Swap, W. (2004). Resource Allocation Beyond Firm Boundaries: A multilevel model for Open Source Innovation. Long Range Planning 37 (2004) 591–610.

Hang, J., Hohensohn, H., Mayr, K. & Wieland, T. (2005). Benefits and Pitfalls of Open Source in Commercial Contexts. In: Stefan Koch (ed.), Free/Open Source Software Development.$

Henkel, J. (2003). Open Source Software from Commercial Firms—Tools, Complements, and Collective Invention. Zeitschrift für Betriebswirtschaft: Ergänzungsheft 4/04 – Strategische Anreizsetzung im Unternehmen.

Hippel, E. v. & Krogh G. v. (2003). Open Source Software and the "Private–Collective" Innovation Model: Issues for Organization Science. Organization Science, Vol. 14, No. 2, March–April 2003.

IBM (2004). IBM to Contribute Speech Software to Apache Software and Eclipse Foundations. [www-03.ibm.com/press/us/en/pressrelease/7293.wss]

Laurent, A.M. St. (2004). Open Source & Free Software Licensing. O'Reilly. ISBN 0-596-00581-4.

Lerner, J. & Tirol, J. (2002). The Scope of Open Source Licensing. [http://www.people.hbs.edu/jlerner/OSLicense.pdf]

Pavlicek, R.C. (2000). Embracing Insanity. Open Source Software Development. Sams. ISBN 0-672-31989-6.

Raymond, E.S. (2001). The Cathedral and the Bazaar: Musings on Linux and Open Source By an Accidental Revolutionary. O'Reilly & Associates. ISBN 0-596-00108-8.

Rossi C. & Bonaccorsi, A. (2005). Intrinsic vs. extrinsic incentives in profit–oriented firms supplying Open Source products and service. First Monday, volume 10, number 5 (May 2005)

Shapiro, C. & Varian, H.R. (1999). Information rules: A Strategic Guide to the Network Economy. Harvard Business School Press. ISBN 0-875-84863-X.

Spiller, D. & Wichmann, T. (2002). Basics of Open Source Software Markets and Business Models. Berlecom Research GmbH [http://www.berlecon.de/studien/downloads/200207FLOSS_Basics.pdf]

Varian, H.R. & Shapiro, C. (2003). Linux Adoption in the Public Sector: An Economic Analysis. [http://www.sims.berkeley.edu/~hal/Papers/2004/Linuxadoptioninthepublicsector.pdf]

Weber, S. (2004). The success of Open Source. Harvard University Press. ISBN 0-674-01292-5.

West, J. (2003). How open is open enough? Melding proprietary and open source platform strategies. Research Policy 32 (7) 1259-1285.

West J. & Gallagher, S. (2005). Open Innovation: The Paradox of Firm Investment in Open Source Software. Western Academy of Management, Las Vegas, March 2005. http://www.cob.sjsu.edu/OpenSource/Research/WestGallagher2004.pdf

Wichmann T. (2002a) Use of Open Source Software in Firms and Public Institutions. Evidence from Germany, Sweden and UK. Berlecom Research GmbH [http://www.berlecon.de/studien/downloads/200207FLOSS_Use.pdf]

Wichmann T. (2002b) Firms' Open Source activities: motivations and policy implications. Berlecom Research GmbH [http://www.berlecon.de/studien/downloads/200207FLOSS_Activities.pdf]

Xu, L. & Brinkkemper, S. (2005). Concepts of Product Software: Paving the Road for Urgently Needed Research. In: Proceedings of 1st International Workshop on Philosophical Foundations of Information Systems Engineering (PHISE'05). Notes in Computer Science, Springer Verlag.