

**OPEN-SOURCING: ON THE ROAD TO THE ULTIMATE
GLOBAL SOURCE?**

Maha Shaikh

Information Systems and Innovation Group
Department of Management
London School of Economics and Political Science
London, UK.

Email: m.i.shaikh@lse.ac.uk

LERO-The Irish Software Engineering Research Centre, University of Limerick, Limerick, Ireland.

Email: Maha.Shaikh@lero.ie

Tony Cornford

Information Systems and Innovation Group
Department of Management
London School of Economics and Political Science
London, UK.

Email: t.cornford@lse.ac.uk

Submitted to the *Second Information Systems Workshop on Global Sourcing: Services, Knowledge
and Innovation*

OPEN-SOURCING: ON THE ROAD TO THE ULTIMATE GLOBAL SOURCE?

Abstract

Open-sourcing is a relatively new term and indicates a less explored theme within the overall perspective of global sourcing. Until recently the term has usually been taken to refer to commercially controlled and created proprietary software which switches, partially or fully to open source licenses – an example would be Netscape and the Mozilla browser. More recently the term has come to take a rather different meaning, implying a deeper link between the open source movement and the tradition of IT/IS outsourcing. Thus open-sourcing has been defined as ‘outsourcing to a global but largely unknown workforce’ (Ågerfalk *et al.*, 2006a).

This paper explores this emerging though tentative trend and charts eight basic sourcing models drawn from the outsourcing and open source domains and which converge towards open-sourcing. In the spirit of a substantial link between open source software processes and outsourcing of IT/IS activities, the paper analyses the characteristics of open-sourcing, and suggests when and why companies might adopt the approach. This draws on the broader literature on open source adoption, both of the products of open source and the processes. Based on this analysis we propose a detailed framework of open-sourcing. The framework places open-sourcing in perspective from the point of view of global perspectives on IT/IS outsourcing but also as an emerging strand of the open source movement.

Keywords

Global sourcing; open source; outsourcing; open-sourcing; globally distributed development.

INTRODUCTION

Open-sourcing - drawing on open source (OS¹) products and mass voluntary participation as a deliberate sourcing strategy for software and other IT/IS services - is a relatively unexplored concept. The term until very recently has been used to refer rather to cases of commercially controlled and created software switching partially or fully to open source licensing. An example would be Netscape's decision in 1998 to 'open' some of its browser code as Mozilla, the basis for the present day browser Firefox. Debate at the time was divided as to whether this was a capitulation to Microsoft and its dominance in browsers, or rather evidence of radical innovation in the software model. Subsequent history does not it seems reveal a clear answer. More recently the term open-sourcing has taken on a significantly different meaning (Ågerfalk *et al.*, 2006a; Anderson, 2005), implying a deeper link with fundamental sourcing options and decisions, and outsourcing strategies in particular. Thus open-sourcing is defined by Ågerfalk *et al.* (2006b) as 'outsourcing to a global but largely unknown workforce', a definition that we take as our starting point.

Open-sourcing in these terms has been initiated and become a potential option for organizations in large part due to the new understanding of Web 2.0 functionality (O'Reilly, 2005) as well as changes in attitude by commercial organizations towards OS software (Dickerson, 2004). The OS movement too has moved on from its original form (free software, counter-cultural ideology, hacker ethics) to embrace an increasingly sophisticated set of business models and strong commercial interrelationships. In a few specific cases, such as IBM's commitment to the LINUX operating system, an open-sourcing approach has been enacted on a huge, truly strategic scale. Reflecting such shifts Fitzgerald (2006) speaks of a new era of OSS 2.0 to reflect the extent to which the OS movement has reshaped itself allowing major sections of the software industry and their commercial clients to accept and work with OS software, its methods, licenses, processes, and (to a degree) its ideology of open innovation (von Hippel, 2005).

Open-sourcing also needs to be acknowledged in its 'global dimension' and as a means of bringing together diverse and distributed human, cultural and economic resources from across the world. It is feasible today, at least potentially, for a client organization to draw upon a truly global distributed collective of talent that can collaborate, communicate and coordinate to achieve outstanding results. The OS and the outsourcing communities both acknowledge that the Internet can bring parties closer together, allowing new configurations of resources and expertise within new and innovative markets. This in turn leads to new styles of working across boundaries, be they political, spatial, temporal, and not least cultural. Claiming a distinctive global character to open-sourcing is not just to repeat a tired

¹ To avoid confusion between the similar terms 'open source' and 'open-sourcing' in this paper we refer to open source as OS and write the full phrase 'open-sourcing' when we speak of the latter idea.

contemporary cliché, but helps reveal some of the truly distinctive aspects of this scheme of production and service delivery. In this open-sourcing is in some contrast to traditional outsourcing which is usually accomplished within a limited number of local contexts and situations – as in traditional software off-shoring - and pursues a strong degree of control as well as technical and cultural homogeneity (though it has been argued that culture is always local even within a global context and thus never truly homogenous (Walsham, 2002)).

The purpose of this paper is then three fold:

1. To trace the origins of open-sourcing ideas within both the OS and outsourcing movements.
2. To describe the various form of open-sourcing that are evident today, and the ways in which these ideas are discussed in both the academic and practitioner literature. In this aspect the evidence base upon which this paper is built includes much that is opinion, anecdotal and drawn from the grey literature (blogs, white papers etc.). Given the novelty of open-sourcing, and its native milieu on the web, we make no apologies for using such sources though they must be carefully assessed and not given the status of peer reviewed research.
3. On this basis to offer a framework for understanding open-sourcing that recognises its roots in both open source and offshore outsourcing and presents it in the context of global sourcing.

The next section clarifies the methodology used to collect and analyse our data. This is followed by an introduction to the open source (OS) movement and an exploration of the relationships, actual and potential, between OS and global sourcing strategies and practices. This leads on to our analysis of the various business models of open-sourcing proposed or in practice today. The discussion section that follows explores the strengths and weaknesses of open-sourcing approaches as a global sourcing strategy. The paper ends with conclusions which indicates areas and topics for future research.

Methodology

As stated in the introduction our aim in this paper is to map the various forms of open-sourcing currently in practice in order to provide a framework of open-sourcing within the context of various sourcing approaches. We seek to broaden the scope and boundary of global sourcing by the inclusion of open-sourcing as a recognised strategy. The purpose of our framework building, in line with Schwarz *et al.* (2007), is to ‘integrate’ material in the area of open-sourcing, to re-define the boundary of global sourcing to include open-sourcing, to indicate a future research agenda in this area.

In approaching this topic the research on open source in the past decade seems to offer the best opportunity for understanding how a diffuse and culturally diverse collective of developers or service providers might organize themselves to build technical products and provide services to client organizations. To research the relevant literature we loosely followed Schwarz *et al.* (2007) in their method of data collection. We began to search using key phrases that were used to yield academic research from online electronic libraries such as EBSCO, Swetswise, ISI Web of Knowledge, and the principal academic sites for OS research such as <http://opensource.mit.edu/> and <http://opensource.ucc.ie/> as well as FirstMonday². The search phrases included:

- open source AND outsourcing
- open source AND offshoring
- open-sourcing
- sourcing
- global sourcing

The papers we identified needed to have one of the above phrases in the abstract to be considered for inclusion (Swanson and Ramiller, 1993). As we identified more papers our search phrases evolved into different trajectories though we tried to keep the number under ten in order to stay focused. Due to a dearth of academic work in the area of open-sourcing we only found a handful of papers that qualified as specifically on open-sourcing. We then began to trace their references back in a historical manner. Most of the references that stemmed from these academic papers on the topic originated in online magazines, blogs and similar resources. Other phrases such as ‘global sourcing’ and ‘open source business models’ provided more academic work but not all was directly applicable for this research. The papers identified were read to check for deeper relevance. At this stage we narrowed our focus to those papers that revolved around key ideas of OS ideology, process, and licence adoption by commercial companies or public sector organizations in the context of sourcing relationships.

Alongside academic material we also directly searched online magazines, blogs, and websites through Google and Google Scholar. The number of hits were, not surprisingly, far greater. Websites of mentioned companies adopting OS software or OS process were also searched, both for available white papers or other material on the strategy and purpose of OS adoption, but also to assess as far as possible their business models. The steps taken to filter academic papers were repeated for online material as much as was possible (the nature of online and non-academic material does not map directly to academic studies).

GLOBAL SOURCING AND OPEN SOURCE

² <http://www.uic.edu/htbin/cgiwrap/bin/ojs/index.php/fm/index>

Phrases that include the term *source* or *sourcing* have proliferated in the past decade. Global sourcing itself (as in this workshop) has emerged as an umbrella term to encompass many other variants and innovations in providing knowledge based services and products, principally but not exclusively IT/IS services. Global sourcing is thus explained as “a term that includes all types of outsourcing, apart from onshore outsourcing. In other words, it refers to cases of nearshore, offshore and farshore outsourcing, when the responsibility for the delivery of IS/IT services resides in countries other than the country where the customer company is located, often overseas” (Tsotra and Fitzgerald, 2007), and as Erber and Sayed-Ahmed note, “Usually, it is characterized by the relocation of business processes to lower-cost locations outside national boundaries, assuming the perspective of the country of origin” (Erber and Sayed-Ahmed, 2005).

In Tables 1 and 2 below we summarise a number of relevant conceptualizations of sourcing. Table 1 refers principally to the outsourcing movement – where source refers to where a product or service comes from, while Table 2 refers to the OS movement – where source refers directly to source code or more generally to who has access to such code. In both tables the final entry is open-sourcing, and suggests a synthesis between the two streams and the two senses of the word ‘source’.

Indeed, tracing back into history, OS origins are found in the sourcing practices of computer users in the 1960s. These origins are usually traced to the pioneer hacker culture of the 1960s when software was sold bundled with hardware, and user developed code, macros and utilities were freely exchanged in user forums (Hars and Ou, 2000; Hars and Ou, 2002). However, from the 1970s software became more and more a separate and distinct business area, increasingly independent of hardware, with most substantial software being proprietary and closed, either in-house developed or bought in the market. This ‘closing’ of software frustrated many, particularly when it occurred in the areas of operating and systems software. One response in the early 1980s came when Richard Stallman, a researcher at MIT, started to write a free UNIX-like system called GNU, and in 1984 founded the Free Software Foundation (FSF)³ (Ljungberg, 2000). Stallman’s work is still recognized as providing the primary conceptual foundation for OS software as we know it today. Stallman saw free software as not having to do with price but with rights and freedoms, thus Stallman is often quoted saying “think of ‘free’ as in ‘free speech’, not as in ‘free beer’” (Stallman, 1999a; Stallman, 1999b). His definition of free software is about the ability of a user to have the freedom to “run, copy, distribute, study, change and improve the software” (Stallman, 1999a). This definition incorporates four specific freedoms;

- The freedom to run the program, for any purpose.

³ <http://www.fsf.org/>

- The freedom to study how the program works, and adapt it to your needs. Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help your neighbour.
- The freedom to improve the program, and release your improvements to the public, so that the whole community benefits. Access to the source code is a precondition for this” (Stallman, 1999a).

These freedoms are embodied in the copyleft method (Moody *et al.*, 2006). Copyleft uses copyright law but “instead of using it as a means of privatizing software, it becomes a means of keeping software free” (Ljungberg, 2000). The copyleft method is embodied in particular in the GNU/FSF General Public License, usually referred to as the GPL. This is the license under which much (though by no means all) OS software is licensed, and which requires that source code is freely distributed. The GPL is a viral or reciprocal license, otherwise termed a strong license; the implication of its use is that, if a person uses any part of GPL-ed code in new software then the new code created is also covered by the GPL. As Fitzgerald (2006) notes, however, as OS has developed its commercial sophistication, so it has developed its licensing schemes, both based on GPL and on other approaches. For example, the LGPL license (Lesser GPL) allows elements of code to be open (as in a library of functions), but then to be integrated into a proprietary product. BSD licenses offer similar variations. However, across the variations in licensing strength there remains a common theme; these license schemes makes it possible to *legally* take the software of others and change and improve it, in the case of a strong license such innovations cannot be owned, but weaker licenses do allow degrees of commercial exploitation (Tuomi, 2002).

By 1997 the word ‘free’ and its connotations led some people, including Eric Raymond and Bruce Perens to seek a new name for the movement (Perens, 1999). They felt that the ‘free’ in free software discouraged businesses from taking it up (Weber, 2004). Thus began the OS movement. Perens and others proposed the OS definition, and the name OS was registered as a trademark for the Open Source Initiative [OSI]. The OSI home page states its *raison d’être* in these terms, “Open source is a development method for software that harnesses the power of distributed peer review and transparency of process. The promise of open source is better quality, higher reliability, more flexibility, lower cost, and an end to predatory vendor lock-in⁴.”

Term	Definition	Authors
Outsourcing	“The handing over of assets, resources, activities and/or people to third party management to achieve agreed performance outcomes. This can be distinguished from the buying-in of	(Lacity and Willcocks, 2006)

⁴ <http://opensource.org/>

	external resources to work under in-house management, and in-house sourcing where internal management and operational resources are used almost exclusively” (Lacity and Willcocks, 2006).	
Insourcing	“The practice of evaluating the outsourcing option, but confirming the continued use of internal IT resources to achieve the same objectives of outsourcing” (Hirschheim and Lacity, 2000).	(Hirschheim and Lacity, 2000)
CoSourcing	“Cosourcing is when the vendor and client collaborate so closely that the vendor can replace or augment the client’s IT competencies. Project teams are mixed. And leadership can come from either one. Effectively, both organizations’ resources become part of a single team aimed at accomplishing the client’s needs” (Kaiser and Hawk 2004).	(Kaiser and Hawk, 2004; Thomas and Parish, 1999)
Netsourcing	“Is the practice of renting or "paying as you use" access to centrally managed business applications, made available to multiple users from a shared facility over the Internet or other networks via browser-enabled devices. Netsourcing allows customers to receive business applications as a service” (Kern <i>et al.</i> , 2002). <i>Note that this has been more often referred to as application service provision – (ASP)</i>	(Kern <i>et al.</i> , 2002)
Global Sourcing	“A term that includes all types of outsourcing, apart from onshore outsourcing. In other words, it refers to cases of nearshore, offshore and farshore outsourcing, when the responsibility for the delivery of IS/IT services resides in countries other than the country where the customer company is located, often overseas (Tsotra and Fitzgerald 2007). Usually, it is characterized by the relocation of business processes to lower-cost locations outside national boundaries, assuming the perspective of the country of origin” (Erber and Sayed-Ahmed 2005).	(Erber and Sayed-Ahmed, 2005; Tsotra and Fitzgerald, 2007)
Open-Sourcing	“Outsourcing to a global but largely <i>unknown</i> workforce”, or a “socioeconomic movement resulting from the marriage of the open source movement and the recent trend towards the international outsourcing of programming” (Agerfalk <i>et.al.</i> 2006a).	(Ågerfalk <i>et al.</i> , 2006a) (Anderson, 2005)

Table 1. Sourcing – Where Something Comes From

The Links between Open Source and Outsourcing

Opinions on the relationship between OS and outsourcing vary. Some believe that the outsourcing community can learn useful lessons from OS communities and practices (Babcock, 2007; Zellen, 2005). For example, Behlendorf, prominent in the OS movement, suggests that outsourcing companies may find it useful to study how talent is treated in OS communities, helping outsourcing organizations “....

to see themselves as talent brokers and pump up the individuals in their organizations as the key asset, rather than just saying 'Hey, we're ISO 9002 compliant'" (Udell, 2003). Recent analyses of outsourcing trends indicate that service companies that combine OS strategies with offshoring are able to source talent globally, and because developers are not in-house find it far easier to scale up and scale down when required (Harney, 2006). Other authors (Anderson, 2005; Minoli, 2004) approach the relationship from the other direction and see the outsourcing experience as addressing some of the often identified problematic aspects of OS; for example, weak requirements specifications, lack of documentation; feature-creep; achieving stability for nascent or bug-ridden products.

There may be many benefits for client and service provider from using OS processes and software in outsourced activities. OS software, being in general built on open standards, can allow for flexible switching of IT providers. More generally, the use of OS software as a component part of outsourced activity, and of associated OS processes and methods, may provide a greater atmosphere of trust and confidence in software since the source code is available and open to all, and the client may feel that some critical risks are thus addressed. In this way some argue that the use of OS code in an outsourcing solution "provides a shorter path to confidence and trust in outsourced software developers.... [thus] open source plays a positive part in the risk management of the decision to outsource" (Anonymous, 2006). Clients can have a more transparent method of system creation (more transparent than normal outsourcing) if "code is no longer hidden in a black box that can't be inspected" (Fox, 2004). A product that is in part or whole 'open' is visible and capable of review, and if all does not go well then there will at least be the source code remaining. Woods and Guliani (2005) also note that OS development, with its ideology of agile and iterative development, can help overcome the mindset that full and extensive requirements are always needed up front - a requirement often associated with the formal outsourcing contract. The result of the use of OS processes may then be faster and more flexible development allowing more opportunity for technical innovation (see also Metiu (2006) discussed below).

Term	Definition	Authors
Open Source	An open source program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed (Perens 1999).	(Perens, 1999)

Inner Source/ Corporate Source	Inner source refers to the development of open source software inside a bordered environment. Though similar to open source in that the code is made available it differs in the key aspect that inner source is a process of developing open source software within usually a commercial organization which is hierarchical in control structures. It makes use of organization mechanisms already in place, and enables flexible collaborations. Companies can use inner source development as an intermediate step towards the integration of open source in their products (Berreteaga 2005).	(Berreteaga, 2005; Gaughan, 2007a; Gaughan, 2007b)
Crowdsourcing	“Crowdsourcing represents the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call. This can take the form of peer production (when the job is performed collectively), but is also often undertaken by sole individuals. The crucial prerequisite is the use of the open call format and the large network of potential labourers... It is only crowdsourcing once a company takes that design, fabricates it in mass quantity and sells it” (Brabham 2007).	(Brabham, 2007; Howe, 2006a; Howe, 2006b; Howe, 2006c)
Progressive Source	Progressive open source requires a novel approach for large corporate software development. Instead of relying on a single-product, project-focused development method. With this method each employee of the corporation can potentially contribute to the development of any given software product. By restricting the openness of the software development to within the corporation or with select partners the corporation does not incur certain business costs of open source. If the corporation is large enough it can realize the main benefit of open source of a large tester base (Dinkelacker and Garg 2001).	(Dinkelacker and Garg, 2001; Dinkelacker <i>et al.</i> , 2001)
Open-Sourcing	“Socioeconomic movement resulting from the marriage of the open source movement and the recent trend towards the international outsourcing of programming”. or “Outsourcing to a global but largely <i>unknown</i> workforce” (Agerfalk <i>et al.</i> 2006a).	(Ågerfalk <i>et al.</i> , 2006a) (Anderson, 2005)

Table 2: Sourcing: Who has a say in it?

Outsourced or not information systems development and management, and software processes in particular, are fundamentally knowledge based activities. Hence one key to success is sourcing talent and gaining access to appropriate knowledge communities. Doing this well, better than competitors, will almost certainly require some innovation and challenge traditional means (Chesbrough, 2006; Chesbrough *et al.*, 2007). In other areas of business open innovation models linked to the internet have attracted attention (von Hippel, 2001; von Hippel, 2005; von Hippel and Krogh, 2003). For example,

Proctor & Gamble⁵ in part base their R&D strategy on a new open model, named as the “Connect and Develop” innovation model. Huston and Sakkab (2006) describe it as a process to “leverage external assets and capabilities.... [in a] relationship of co-invention-based interaction with outside resources”. This model of innovation, with its connections focus, seeks to tap into multiple knowledgeable communities across the globe (Huston and Sakkab, 2006; Huston and Sakkab, 2007; Sakkab, 2002), and notes that to “[leverage] people whom you don’t have direct control over, you have to build relationships and trust...”. These authors are clear that this is not a conventional method of outsourcing R&D, but rather “in-sourcing creativity”, and as in OS processes, aims to tap into a large pool of people, ideas, developers and testers.

An example of a company engaged in a combination of open source and offshore outsourcing is Intalio⁶, a US based company which supports both an open source and commercial version of its Business Process Management Suite (BPMS). The CEO, Ismael Ghalimi, explains the motivation of the company to have two versions of its BPMS, “having an OS version of our product lures the buyer to adoption and requires training, support, and maintenance contracts and then software licenses for more advanced versions of the product that we charge for” (Harney, 2006). The benefit to Intalio includes the possibility to build a ‘user base at a very low marketing cost’ (Harney, 2006), and customers gain free software. Indeed, within the open source movement there are many firms that pursue a similar blend of open source development, and proprietary software and service provision.

Open-sourcing is effective as a strategy for global sourcing in part because of its ability to exploit a commoditized and standardized conception of software (but with room for the odd tweak). OS commodification is driven by the understanding that ‘good enough’ solutions suffice so if software is open, free (in many cases as in free beer) and good enough to solve most of a problem, then alternatives which charge a premium price and lock you into their standard will be pushed out of the market (Asay, 2006b). Asay suggests that this commodification process in particular squeezes out of the market the middle-sized firms as there ‘...is no room for middling and muddling. Open source will commodify from the bottom up while “upmarket” vendors will dominate “up the stack”. Everything else will be wasteland’ (Asay, 2006a, p104).

Drawing from another strand in the OS literature we can understand open-sourcing as a new form of organizing, or the building of a novel type of organization to serve new knowledge needs. Metiu and Kogut (2001) studied a number of software companies in four different countries and identified two

⁵ <http://pg.t2h.yet2.com/t2h/page/homepage>

⁶ <http://bpms.intalio.com/index.pfhp>

distinct forms of organizing for global software development. Their analysis focuses on organizing of innovation and creativity in globally distributed work. The established model they term the 'global project model', but they see also a new model emerging – the 'open development model'. The 'global project model' at the most basic level implies that companies are able to take advantage of lower cost of labour by passing work (routine tasks) to offshore sites. This requires requirements specifications up front, and high degrees of control. In their analysis this model begins to translate into the 'open development model' as, over time, offshore firms and developers begin to not only follow requirements but also to build skills, innovate and create their own requirements for client companies. However, Kogut and Metiu (2001) question whether offshore developers are able to move very far away from simple specification following. The 'open development model' comes to push at the boundaries of the 'global project model' only when or if the motivation of contributors becomes different, actively seeking new experience, knowledge and skills. This is an analysis that echoes strongly the general understanding of motivation in OS communities as building human capital with OS developers are motivated to contribute to OS projects for the explicit purpose to learn and enhance their skills (Lakhani and Wolf, 2005; Shaikh, 2007).

Developing this theme, and the problems faced if and when developers do exercise more innovative capacity, a later study by Metiu (2006) of the (troubled) development by the 'global project model' of an innovative software product indicated how much more status was attached to ownership of design versus code. This study's ethnographic account of the governance of an offshoring relationship reveals how 'interactions both shape and are shaped by the status and roles assigned to groups in a society, and that they function to maintain and justify the structure of intergroup relations and intergroup behaviour' (Metiu, 2006). In this case the result was that innovations suggested by the Indian software engineers were not taken on board by the Californian clients.

OS ideology, giving as it does primacy to code and with personal status deriving directly from code production, suggests that an open development model might have revealed quite different outcomes. Thus, some business experts, while clear that both OS and offshoring outsourcing are phenomena that will prevail, recommend a cautious approach. Netke (2005), when considering if OS software will play a large role in the offshoring marketplace replies "I think it's going to become more popular, but the reality is that any kind of expertise in OS requires in-depth technical knowledge and engineering discipline, which not a lot of people have. So if you think about it, open source is not really about free software, it's about a sophisticated services model". Oren (2004) adds that "Open source and IT offshoring are the products of the same driving forces, two faces of the same coin. And they are feeding

off one another”, the greater the degree of offshoring then the better the infrastructure (human and technical) to support it. OS production and distribution can then free-load off the same infrastructure.

Business Models of Open-Sourcing

In the section above we have reviewed the potential for a link between OS and outsourcing in a global perspective. In this section we flesh out these ideas with data on the business models in use. We describe four umbrella open-sourcing models that are apparent from the literature and current commercial practice. We elaborate each model further to reveal two or more sub-models that provide a more detailed understanding of the practices involved for the companies that have adopted them. This analysis builds on the framework of Fitzgerald (2006) which proposes four dominant OS business models; value-added service, loss leader/market creating, leveraging community development, and leveraging the OS brand.

Here we elaborate this model and adapt it to our purpose. We rename ‘leveraging community development’ as ‘Resource and Expertise Mobilization’. We prefer to use the term community cautiously given the vague definition and usage of the word by both academics and laymen (Hillery, 1955; Jewkes and Murcott, 1996; Nisbet, 1966). More specifically our research indicates that companies are looking for ways to increase and improve flows of knowledge and mobilise of resources within and beyond their own organization, for example by inner sourcing. Thus we use the broader term to describe harnessing people and their knowledge across the range of global sourcing situations. Similarly we substitute for “Leveraging the Brand” the more general category of ‘Value Creation’. This reflects our identification of open-sourcing as a means of building capacity within organizations and within collaborative networks and alliances. In the rest of this section each of these four models is explained with examples from real organizations that have appropriated them, some with mixed results. This is summarised in Table 3, and elaborated in Table 4.

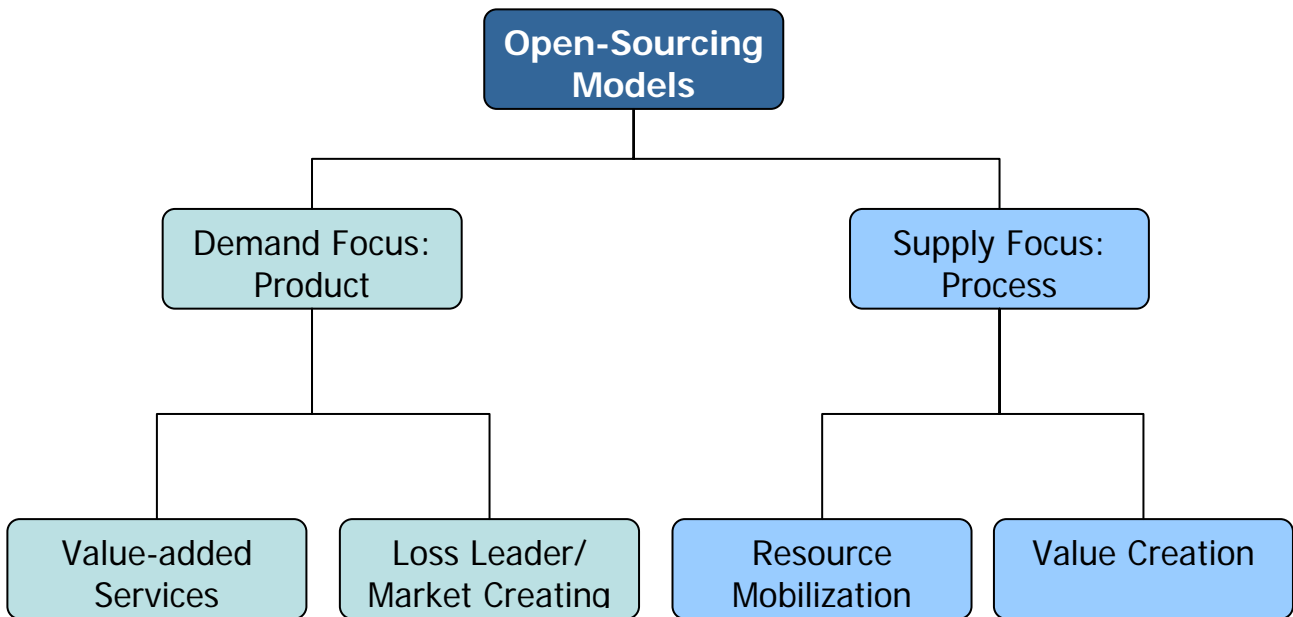


Table 3: Business Models for Open Sourcing

Value-added service: The value-added service model (Fitzgerald, 2006) focuses on the provision of services to customers that help them to acquire, install and manage software and provide other business service solutions. In the OS world there are many such organizations from small specialists to quite substantial players such as RedHat. Indeed, the sustaining of the open source movement in the past decade is very largely attributable to such businesses commitment and provision of a support channel to the wider market. Within this model we identify two sub-categories; charter-a-source and symbiotic source.

Charter-a-source is probably the least demanding model that companies can choose in order to source software or other services from a global workforce, and is appropriate where levels of innovation are low, and specification is strong. Small companies have mushroomed on the Internet and these developers seek to establish their different expertise and market their skills. Targeted web sites support this market place while clients can use the same sites to advertise their problems and needs. The website is usually owned by a third party which makes a profit through a small commission charged to both the company in need of help and the developer who steps in with a solution. This is an example of crowdsourcing (see Table 2), examples in the software industry include Rent-A-Coder (Royce, 2006, 29th October) and Code-with-Coder⁷.

Symbiotic source addresses systems integration and compatibility concerns across software architectures. Two similar but individually interesting examples are Gluecode (Gluecode was bought by

⁷ <http://www.code-with-coder.com/>

IBM in mid 2005⁸) and Specifix. Gluecode is an ‘an application server platform that encompasses several OS products and groups them together using a layered architecture, hence the name *Gluecode*’ (Barcia, 2005). It offers a middleware stack based on OS code elements and then allows customers to add their code to its CVS repository and will make sure that the new code added by the customer is compatible with the rest of the software and does not break the build. In this way the customer becomes ‘a development partner’ (Asay, 2006b). A user can then develop their own code base while receiving the support of a dedicated management service for the ‘standard’ elements.

Loss leader/Market creating: The loss-leader/market creating model (Fitzgerald 2006) distributes software or other services for free and relies on creating a market for complementary but closed software or services for which a fee can be charged, e.g. Sendmail and Sendmail Pro. We identify two sub-categories in the loss leader/market creating model; mature source and coalition sourcing.

Mature source refers to the opening of established products or services to free their owners of some development or maintenance burdens. This returns to the original meaning of the term open-sourcing, and the opening of the Netscape code in 1998. A more contemporary example is Open Workbench which Niku Corporation released the source code for under the Mozilla Public License⁹ handing it over to its users to take in any direction needed. Niku could thus ‘cut the resources [it needed to] devote to developing and maintaining Workbench and provide Workbench customers with the ability to make enhancements to the software on an as-needed basis’ (Greenemeier, 2004), described as ‘a weird form of outsourcing’. Niku released Open Workbench on SourceForge, a primary OS portal, and though it may not have had any major release since late December 2005, it can boast over 62000 downloads and still has feature requests being logged in 2008.

Coalition sourcing refers to situations in which partners work together to provide and service open products. Hamm (2007) relates the example of the long established American database company Ingres Corporation¹⁰, now with OS code (an example of mature source), aligning itself strategically with an Indian outsourcing company Satyam¹¹. Hamm’s analysis suggests that Ingres may hope to secure a portion of Oracle’s database market through such an OS comeback, while Satyam can use an OS alliance to strategically distinguish itself as it competes for outsourcing contracts. This move is fairly new, too new to know how successful it will be (the alliance was announced in February 2007¹²), but

⁸ <http://weblog.infoworld.com/techwatch/archives/001370.html>

⁹ <http://www.niku.com/products.asp?id=19>

¹⁰ <http://www.ingres.com/>

¹¹ <http://www.satyam.com/>

¹² http://www.ingres.com/press/2007-02-12_Satyam.php

history suggests that successful open source software needs service provider support so as to build market share, just as proprietary software like SAP does through its alliance partners.

	Open-sourcing Model	Characteristics	Possible Problems	Company Examples
Value-added Service	Charter-a-Source	<p>Relatively inexpensive software.</p> <p>Able to scale down quickly.</p> <p>Access to developers from world.</p> <p>Greater transparency and accountability.</p> <p>Agile development.</p> <p>Coders have access to repository of code.</p> <p>Not usually/always open source software.</p> <p>Form of crowdsourcing.</p>	<p>Limited loyalty to develop code further in future.</p> <p>Little of the open source code is contributed back to the collective.</p> <p>Commission of mediator is a deterrent to developers.</p> <p>Trend of undercutting bids by small amounts infuriating developers.</p> <p>Lower bids resulting in lower quality software.</p> <p>Little transparency in arbitration.</p> <p>Cultural and language difficulties with developers from different countries.</p>	<p>Rent-A-Coder¹³, Code-with-Coder¹⁴, FreelanceWebmarket¹⁵, GetaFreelancer¹⁶, LaunchPad Bounties¹⁷,</p>
	Symbiotic Source	<p>Provides supporting/needed code.</p> <p>Keeps version control.</p> <p>Collaborates in actual development – co-development partnerships.</p> <p>Provides additional services for fee.</p> <p>Greater transparency and accountability.</p> <p>Agile development.</p>	<p>Dependence and possible lock-in to middle man.</p> <p>Complexities of licensing and license proliferation.</p>	<p>Gluecode¹⁸, Specifix¹⁹ (Gluecode has been taken over by IBM)</p>

¹³ <http://rentacoder.com/RentACoder/default.asp> and <http://www.mybids.net/>

¹⁴ <http://www.code-with-coder.com/>

¹⁵ <http://freelancewebmarket.com/>

¹⁶ <http://www.getafreelancer.com/>

¹⁷ <https://launchpad.net/bounties>

¹⁸ http://www.ibm.com/developerworks/websphere/techjournal/0509_barcia/0509_barcia.html

¹⁹ <http://www.specifix.com/>

Loss Leader/Market Creating	Mature Source	Open source software. Commercially mature, established product. OSI approved license. Cut development costs. Provision of add-on services for fee. Build user base. Low marketing costs. Able to scale down quickly and inexpensively. Greater transparency and accountability.	Not all code is open source. Patented source is difficult to mix. Complexities of licensing and license proliferation. Limited impetus to develop code further. Quality of contributions often dubious.	Niku ²⁰ (releasing Workbench as Open Workbench), Intalio ²¹
	Coalition Sourcing	Alliance of strategic convenience. Open source adoption used as marketing device. Cooperate to restrain competition. Access to larger pool of talented developers.	Could escalate organizational costs. Rising production costs. Cultural and language difficulties with developers from different countries.	Ingres ²² aligning with Satyam ²³ (to restrain Oracle), IBM ²⁴
Resource and Expertise Mobilization	Talent Sourcing	Company adoption of OS collective ideology. Build relationship with external communities. Access to larger pool of talented developers. Agile development. Greater transparency and accountability.	Lose top management support. Unable to sustain enthusiasm beyond short-term. Complexities of licensing and license proliferation. Retaining collective links is tricky. Quality of contributions often dubious.	Proctor and Gamble ²⁵ (Connect & Develop strategy), HP ²⁶ , IBM

²⁰ <http://www.niku.com/>

²¹ <http://www.intalio.com/>

²² <http://www.ingres.com/>

²³ http://www.ingres.com/press/2007-02-12_Satyam.php

²⁴ <http://www.ibm.com/developerworks/opensource>

²⁵ <https://secure3.verticali.net/pg-connection-portal/ctx/noauth/PortalHome.do>

²⁶ <http://opensource.hp.com/>

	Portal Alliance Source	<p>Access to large pool of talented developers. Exploits distributed intelligence of global collective. Cuts development and some organizational costs. Offers a collaborative platform for mediation. Portal offers real time monitor of status of project. Provision of version control and repository. Greater transparency and accountability.</p>	<p>Overheads in management of filtering poor contributions. Subject to both good and bad code updates. Over-reliance on external collective can deteriorate in-house expertise.</p>	<p>Allstream²⁷ with SourceForge²⁸</p>
Value Creation	Open Outsourcing	<p>Open source philosophy. Open standards. Customers free to own the source. Limited to no vendor lock-in. Exploits distributed intelligence of global collective. Greater transparency and accountability.</p>	<p>Complexities of licensing and license proliferation. Little or no code contributed back into open source collective. Varying battle to gain acceptance for open source code for high security code like banking applications.</p>	<p>NatureSoft²⁹</p>

²⁷ <http://www.allstream.com/home/>

²⁸ <http://sourceforge.net/index.php>

²⁹ <http://www.nature-soft.com/index.html>

	Inner Source	Fosters in-house IT department. Greater transparency and accountability. Access to large pool of talented developers. Code releases and maturity more likely to match industry needs. Agile development. Keeps company-wide version control.	Need for restructuring in the organization can be onerous. Too many forks in the code and duplication of effort characteristic of open source projects. Free-riders on good code with little reciprocal contribution by all. Organizational acceptance for the change in mindset required for inner source to be successful. Complexities of licensing and license proliferation. Problem of all members abiding by standards set by industry.	Bell Labs ³⁰ , IBM, HP, Sun Microsystems ³¹
--	---------------------	---	---	---

Table 4: Categorization and Characterization of Open-sourcing Models

Resource and expertise mobilization brings the focus back to the source, not as in code, but of the resource that creates a product or service. The previous two models, ‘value added service’ and ‘loss leader/market creation’, focus on the demand side and the product or service, less on the process that makes a product happen. But the distinctive element of OS processes, and open-sourcing, is tapping into a large, distributed and unknown workforce. Many major companies now take the idea of tapping an ‘open’ or collective resource quite seriously, as indicated by schemes such as P&G’s ‘Connect and Develop” innovation model discussed above, or the multiple OS projects that are promoted and supported by the likes of HP, IBM and SUN. The two sub-categories we identify within resource and expertise mobilization are talent sourcing and portal source. The focus for both sub-categories is on attracting and managing collective expertise through building some form of bridge or agreement between a client with a problem or need, and a wider collective able and willing to help.

Talent sourcing describes the situation where an organization explicitly tries to build a communication channel between itself and an existing or nascent body of global workers so as to achieve access to the largest pool of potential experts in any given area, but also to new areas of expertise that may not have been up to then identified. Such a move almost inevitably comes to challenge accepted ideas of who

³⁰ <http://www.bell-labs.com/news/2000/june/7/2.html>

³¹ <http://www.sun.com/software/opensource/>

holds and owns knowledge, who has legitimate rights to participation in innovation processes, and who has the rights of decision making. As we know from the early history of the OS movement, to some people the very idea of mass participation, working for free, and openly sharing ideas smacks of Bolshevism and worse (Bezroukov, 1999a; Bezroukov, 1999b; Jones, 2000).

Portal source in contrast, refers to companies reaching out to existing OS community through an established OS portal, such as www.SourceForge.net. In this way companies can potentially access pre-formed collection of experts with OS experience through the various projects and products listed. The more challenging option is to establish and resource their own open project and invite others to participate, drawing in other potential users of the code or service proposed, as well as people who see opportunities to contribute or develop relevant skills. Participation of either kind (i.e. users or workers) is good and can validate the idea and refine it and thus draws yet more resources and skills to the effort. There is certainly some evidence that firms do use OS portals such as SourceForge as a way to entice others to build, adapt and support the systems they require (e.g. Allstream Corporation³²).

Value creation refers to the use of open-sourcing process to create value that allows a company or group of organizations to sustain themselves over time. Rather than concerning what product or service is needed by the client it emphasises how the process of participation in open-sourcing can be used to create value for clients. For example a number of companies use participation in OS projects to train staff, build reputation and locate new hires (Agerfalk and Fitzgerald, 2008, forthcoming; Grand *et al.*, 2004). The focus is then on building up in-house capability (e.g. in an IT department) and spreading new ideas across the organization. This approach may give rise, for example, to greater inter-departmental work or to revealing hidden talents or aspirations. This in turn may offer scope for learning and (re-)organizing. The key idea which emerges from this category and from 'resource and expertise mobilization' is the focus not on any product/service alone but on the process of support for innovation through open collaboration.

Open outsourcing refers to the adoption of OS philosophy and standards in support of an outsourcing strategy. We distinguish this from the more general value added service model by indicating how open outsourcing adds value but not at simply the product/service level. The aim in open outsourcing is to add value to the entire process of development and distribution (and even beyond with better after-sales service and greater innovation of new products/services). This model relies on the transparency that is provided by OS licenses, as well as adherence to standards. In these cases companies selling services are relying on OS communities to help in creating the product or service required by their client, and, as

³² http://www.allstream.com/enterprise_home/

the source code is open and visible, the client benefits and has a means to counter vendor lock-in e.g. NatureSoft.

Inner source is a model that adopts the rules of OS development to build a community and product collectively (Gaughan *et al.*, 2007). But, unlike the other methods classified as value creation, inner source does not look beyond the organization for expertise or contributions. Rather it concentrates on tapping in to in-house expertise and fostering a more collaborative environment *within* the company. A good example of a global company using inner source as a way to produce software is Nokia (Jaaksi, 2006; Jaaksi, 2007).

Discussion: Products and Services, Open Process, and the Collective

These business models that have emerged in response to organizational acceptance of OS ideas as part of a sourcing strategy of course need to be considered with regard to their sustainability and comparative effectiveness. Not all innovative business models work – indeed most fail! In this section we attempt to consolidate and assess these ideas in terms of three key aspects: the products and services offered through open sourcing, the work processes they imply, and the potential for a relevant global collective (community) that can support such activity (Fitzgerald, 2006; Shaikh, 2007). Using this framework we explore the advantages and problems that organizations may face as they come to assess and participate in open-sourcing activities.

Product and Service

Open-sourcing is about some degree of trust or belief in the ability of a ‘largely unknown workforce’ to deliver what a client wants. In the established area of open source this is usually understood as software, and most often software that has generic capabilities as part of an IS infrastructure and thus has obvious value to multiple users. In contrast, in open-sourcing the expectation must be that the product or service sought is in some degree unique or specific to the client’s needs. Of course this too is part of the open source model, open code allowing customization, but research seems to show that relatively little such customization actually takes place. But if such specific needs are to be served through an open-sourcing approach, then we must expect that product (as in code) will become more and more bound into a service, an observation that is born out by the examples cited in the section above.

Consideration of the product or service also leads us to consider the implications of the distinctive licensing of intellectual property that is central to OS. To be convincing for clients needs open-sourcing

will need to carefully address the issue of license (Clarke, 2007). The transparency that stems from adherence to such licenses is essential to allowing mass participation and peer review process. Current experience with open source code suggests that corporate lawyers are not always well-versed in all the different OS licenses (the Open Source Initiative (OSI) has approved 58 distinct licenses³³) or able to identify the implications of their use. It is however clear that the movement to less strong licenses such as the LGPL (Lesser GPL) and the BSD license and the practice of dual licensing may help build understanding.

In some areas OS software is recognised as of the highest quality. As a web server Apache is a match for its proprietary competitors, as arguably is LINUX in the operating systems market. However, beyond the top 20 OS products the quality of the product may be at least less easy to judge, notwithstanding the availability of code. The same may be true, by extension, of other open-sourced services and ideas, particularly if the peer review aspects are less than rigorous as may well be the case in global sourcing situations exploiting weaker licenses. Companies may also fear losing intellectual property rights to the software or service they source in this way and a switch to a different business model may not be accepted without effort or some desperate circumstance. There are also relevant concerns over security implications of wider access to code (Hoepman and Jacobs, 2007; Neumann, 2000). It is often proposed that OS software, because the code is visible, should be more secure through the scrutiny it has by a large body of testers and reviewers (Schneider, 2000; Schneier, 1999, September 15th). But there is the counter argument that when code is visible it can be manipulated in more subtle ways (Wheeler, 2003; Witten *et al.*, 2001).

An Open Process

The OS process, in which code is reviewed and commented on by a larger community, where competing ideas are encouraged and which parallels development with debate, has been shown to be able to crack open some enduring software woes. More generally it is part of a broader movement towards open innovation that adopts a philosophy of mass participation, innovation through iteration, with a fast and fluid cycles from idea to critique to testing to use. This is supported by the ability to freely borrow and rework what already exists, based on a strong faith in peer review and adherence to (and creation of) standards. For many corporate IT/IS tasks these characteristics seem to offer an attractive possibility of accessing new levels of quality, responsiveness and timeliness.

But not surprisingly it is not so easy. Companies have endured many problems when adopting open-sourcing products, let alone processes, not least in the managerial and organizational change issues

³³ <http://www.opensource.org/licenses/category>

faced. These include basic lack of understanding, continuity concerns with experts and module designers, variable code quality and lack of full documentation, and lack of understanding of the range of support services. And even if higher management are comfortable with taking such a route to source major elements of their IT/IS, it may not be acceptable to front line staff who have a large investment in a different type of proprietary skill.

Collectives and Community

Finally, we must understand that fundamental to the open-sourcing idea is the existence of that 'global but largely unknown workforce', willing and able to undertake intensive knowledge work. Indeed many hands may make light work, or as the open source movement has it, 'Given enough eyeballs all bugs are shallow', but is the emphasis that OS scholars place on motivation - both community and individual - and the value it creates through knowledge creation and constant learning, as relevant in an open-sourcing model?

Sustained access to a collective or to organization-wide expertise is a necessary condition for open-sourcing, but is almost impossible to engineer and once achieved poses many problems to maintain long-term. Within the wider world there is increasing competition for the attention of the talented, and more and more opportunities form them to achieve realization of their goals through conventional work, while inner source strategies rely on a pool of experts that is not large. Other models of open-sourcing such as open outsourcing don't have the latter problem but bring other organizational concerns such as that of a mismatch between the hierarchical governance model of a client and the more distributed and flat governance model of most OS communities.

CONCLUSION

Open-sourcing is not a panacea for the problems of outsourcing, nor does it somehow guarantee access to high quality and low cost services that can deliver for clients. Yet as we have shown above, some companies large and small do adopt it and manage to achieve new and innovative outcomes by opening up work processes to mass participation. There seems to be promise in these novel combinations of OS ideas and outsourcing that may side-step at least some of the problems of both approaches when used independently. This paper has shown how and when companies have adopted open-sourcing and we have characterized this evidence into four overarching families of business models. We have further explored the interesting distinctions that emerge through the process, service and collective/community framework concerning issues of license, control, communication, governance and quality. The sections above have outlined the main advantages and disadvantages of open-sourcing. This seems to suggest that there are opportunities and if we are optimistic that most of the problems of open-sourcing can be

resolved with patience, careful strategy and adoption of new practices. Table 5 lists an expanded set of criteria that client companies considering open-sourcing may need to consider, and contrasts them to the existing sourcing options e.g. in-proprietary code, outsourcing development and direct sourcing of OS products.

From the data presented in this paper we know that open-sourcing is being practised, and we have started to note some of its advantages and problems, but there is still a substantial gap in the research literature in this area. We thus propose a tentative research agenda that focuses on the following five questions:

- What is the appropriate framework of risk and reward that client companies face when considering open-sourcing.
- Given the intangible nature of much service activity, what methods of expertise capture are available to use alongside open-sourcing?
- While we know the OS processes can work for product development (code), how can they be adapted to provide stronger service support (given that open source has long had a problem with the non-performance of the boring jobs that nobody wants to do)?
- How, and in what ways can clients for open sourcing make connections to potential of existing communities in ways that can build long term sustainable relationships?
- How can these relationships be supported, for example in the form of championing schemes, financial encouragement, expertise transfer? How effective are these various strategies?

APPENDIX

	Criteria	Proprietary Source	Outsourcing	Open Source	Open-Sourcing
Process	Communication	Face-to-face mostly.	Face-to-face, but also uses technology like email, telephone etc.	Mostly online, though limited face-to-face in conferences.	Combination of face-to-face and online.
	Control	Centralized, tight and rule based.	Centralized, tight and rule based – often implemented though the SLA.	Distributed, informal rules and norms.	Combination of control mechanisms used ranging from very centralized to quite distributed and informal.
	Infrastructure	Works on internal company infrastructure though this can be global.	Works on internal company infrastructure though this can be global.	Built on distributed OS structure and uses already in place Internet infrastructure.	Built on both global internal infrastructure and Internet.
	Governance model	Hierarchical, top down with strong management structure.	Hierarchical, top down with strong management structure, and often needs to consider governance model of client company.	Varies from OS project to project but often democratic and a combination of bottom up and top down.	Combination of hierarchical (and client based governance structure matching) and OS democratic style.
	Maintenance	Clear and distinct after sales phase with documentation.	Clear and distinct after sales phase with documentation.	Evolutionary and agile development that makes little distinction between phases of development.	Combination of clear phase and evolutionary type of development and after sales service.
	Distribution model	Software/product/service is often created for a large market so generalized product and then distributed widely through retail.	Made to measure software/product/service so distribution channel is closed and internal.	Distribution is carried out over the Internet as downloads (which are sometimes free of charge) but always with open access to product design and code. Internet provides large, cheap and effective distribution channel.	Mix of internal, closed channel with the use of the Internet (where companies often take from this latter channel but don't pour back their now copyright idea/product).

	Total cost of ownership	Clear methods to help quantify TCO in the company making decision making on this basis between products/services easier to compare.	Clear methods to help quantify TCO in the company making decision making on this basis between products/services easier to compare. Such factors have to be dealt with when creating the SLA and thus this is clear.	Too many factors that are hard to quantify thus making TCO difficult to measure in OS.	Depending on the open-sourcing route taken companies would face varying fuzziness of TCO. TCO is clearer in open-sourcing than OS as some elements must be quantified for a company wishing to sell its product/service.
	Marketing	Wide, open and global marketing strategy to pitch product/service at the largest audience.	More at the level of company ability and expertise to enrol other firms into long term contracts.	Via forums, word of mouth, use of product, and gaining critical mass of designers and developers in the community.	Global marketing strategy but also based on expertise of the company – however also rely on word-of-mouth and forums.
	Transparency of process	Limited transparency of process as proprietary license.	Limited transparency of process as proprietary license.	Greater transparency than other models and depends on the promiscuity of license.	Partly transparent – depends on license. If dual license then good level of transparency in process.
	Development model	Depends somewhat on size of project – large scale production usually entails clear phases and division of labour.	Depends somewhat on size of project – large scale production usually entails clear phases and division of labour.	Agile, evolutionary and more focused on parallel constant testing and building.	Combination of planned and clear phase production with agile methods and beta testing.
Product	License	Closed and proprietary	Closed and proprietary	Open source (with varying degrees of openness)	Dual licenses and some only OS or proprietary.
	Application type	Wide range of both products and services are covered.	Back-office and non-core applications/services.	Horizontal rather than vertical business applications and services. Mostly infrastructural and back-office.	Non-core applications and services but open-sourcing model used as a strategic device and propped up as strong PR for the company.
	Code/product/idea quality	Unable to reverse-engineer product or service so need to accept quality as provided by vendor.	Quality specifications somewhat written into SLA which ensures certain level of quality. Unable to reverse-engineer product or service.	Quality easier to test as large base of testers and bug fixers. Source and process are fairly transparent so if needed the product/service can be analyzed.	SLA specifies quality level so quite effective yet made even better when both product and process are transparent and open to scrutiny.
	Ownership	Owned solely by company or license holder.	Owned solely by company, license holder, or client company that pays for the product or service.	Owned by the collective and not by any one individual or company solely.	Partly owned by the collective and some strands of the product/service owned by a company.
	Architecture	Closed architecture.	Closed architecture.	Open architecture.	Early part of the product/service is

				open but the final product is often closed architecturally.	
Level of documentation	Detailed documentation is a requirement.	Detailed documentation is a requirement and specified in the SLA.	Often patchy or non-existent but sometimes this is mitigated by speedy help provided through discussion forums by experts.	Documentation of good quality and detail is a must and this counters a serious problem companies have with OS products/services.	
After sales service	Either part of the contract or non-existent.	Either part of the contract or non-existent – this can lead to issues of vendor lock-in just as the maintenance issue.	No real after sales service at all but discussion forums are often used to tap into community expertise and help.	Companies often bridge between the community and client and ensure that after sales services are provided through the company but tapping into community expertise.	
Reusability of code/idea/product	Reusability limited to development/designer of idea but often the company dictates what can be reused.	Reusability limited to client company as they pay for the product or service and thus own it. However they often don't have the expertise to reuse it so are locked into vendor company.	Great reusability of idea, code, product as the process and product is open and transparent – and available for anyone to reuse.	Good reusability level as company usually releases much of the design or product back into the community. Indeed this is encouraged as a way to build trust between company and community.	
Community/Organization	Motivation	Mostly financial, promotion and job-related.	Mostly financial, promotion and job-related.	Learning from each other, reputation, ego, potential job-seeking and creativity.	Community and company link building to retain expertise over time, promotion of product and company, and sustaining the community and its members.
	Contributor profile	Company based and can be a global company. Paid employees.	Company based and can be a global company where outsourcing and offshoring implies mixed cultural background of developers and designers. Paid employees.	More truly global than any other model, mixed culturally and mostly male. Mostly working in their free time/while on another job and not usually paid.	Global as the OS community it links too is very global but the company members are often mostly from one country, paid employees.
	Level of interest and contribution	Paid to work on project so interest varies and contribution is expected and dictated by senior management.	Paid to work on project so interest varies and contribution is expected and dictated by senior management and client company.	High level of interest but most contributions are small and can take the form of using the application/service and providing feedback/bug reports.	Good level of interest and contribution may be small but is consistent.
	Mobility of developers/	Mobility limited to company but company	Mobility limited to company but also to	Great mobility between projects and	Very good mobility between company

creators	maybe global.	client sites and often experts of the vendor are hired by the client.	often we see the same person is a member of more than one community.	sites, client sites and OS community.
Access to learning/training	Companies' offer training options but learning from other colleagues is limited by the number of people you have communication with and access too.	Training is offered by companies. Learning is enhanced through access to a larger population of people from working on client sites.	Training (traditional form of it) is not usually available but learning from colleagues is a key reason why people contribute and belong to a community.	Employees/members have both training and collaborative learning resources. Training is provided by the company and all members have access to a global community of OS experts outside the company walls.
Size of community	As large as the number of employees of the company.	As large as the number of company employees and often the client site too.	Varies from project to project but potentially this is a global and very large workforce.	A mix of both company employees and access to the global workforce accessible through the OS community.
Sanctions on rule breaking	Companies have clear rules and enforcement strategies.	There are clear rules and enforcement techniques which stem from both the company of origin and client/vendor companies.	Informal but very effective enforcement techniques to control the community.	Mix of informal and legally enforceable sanctions.
Status of core developer/creator group	Position of authority but still answerable to manager.	Position of authority but still answerable to manager and client/vendor management.	Position of control and authority which is based on recognition of expertise and thus very influential.	It is a position of control and authority yet there is some measure of answerability to the company.
Global distribution	Large companies have global offices which create a global presence and workforce.	Large companies have global offices which create a global presence and workforce, and this is enhanced when outsourcing is done in the form of offshoring.	Internet as infrastructure provides few temporal or spatial boundaries to such communities thus creating a more globally distributed community/organization than possible in any one company.	Beneficial combination of resources which gives rise to a considerably globally distributed presence.

Table 4: Comparative Characterizing of Open-Sourcing within Global Sourcing Types

REFERENCES

- Agerfalk, P. and B. Fitzgerald (2008, forthcoming) "Outsourcing to an Unknown Workforce: Exploring Opensourcing as a Global Sourcing Strategy", *MIS Quarterly*, (forthcoming).
- Ågerfalk, P. J., B. Fitzgerald, H. Holmström and E. Ó. Conchúir (2006a) "Open-Sourcing as Offshore Outsourcing Strategy". in *29th Information Systems Research Seminar in Scandinavia (IRIS 29) - Paradigms Politics Paradoxes*, LO-skolen, Denmark,
- Ågerfalk, P. J., B. Fitzgerald, H. Holmström and E. Ó. Conchúir (2006b) "Open-Sourcing in the Celtix Project: A Case of Outsourcing to an Unknown Workforce?" in *27th International Conference on Information Systems (ICIS2006)*, Milwaukee, Wisconsin, USA, December 10-13 2006,
- Anderson, K. (2005) *Open Outsourcing* Address: <http://kelly.anderson.name/openoutsourcing/>.
- Anonymous (2006) *Open Source and Offshore Development Flat World Software Development* Address: <http://flatworldsoftwaredevelopment.com/blog/2006/03/07/open-source-and-offshore-development>.
- Asay, M. (2006a) "Open Source and the Commodity Urge: Disruptive Models for a Disruptive Development Process" in *Open Sources 2.0: The Continuing Evolution*, (DiBona, C., D. Cooper and M. Stone eds) O'Reilly, Sebastopol, CA.
- Asay, M. (2006b) "Outsourcing and Open Source: A Match Made In...?" in *InfoWorld*, IDG Network (April 28, 2006).
- Babcock, C. (2007) *How to Tell the Open Source Winners from the Losers* InformationWeek Address: <http://www.informationweek.com/story/showArticle.jhtml?articleID=197002953>.
- Barcia, R. (2005) "What Is Gluecode? And Why?" in *IBM WebSphere Developer Technical Journal*, 21st September 2005.
- Benkler, Y. (2002) "Coase's Penguin, or, Linux and the Nature of the Firm", *Yale Law Journal*, **112 (3)**, pp. 369-446.
- Berreteaga, A. (2005) *Cosi - Co-Development with Inner and Open Source in Software Intensive Products* ESI - European Software Institute: Tecnalia Address: <http://www.esi.es/index.php?op=14.4>.
- Bezroukov, N. (1999a) "Open Source Software Development as a Special Type of Academic Research (Critique of Vulgar Raymondism)", *FirstMonday: Peer Reviewed Journal on the Internet*, **4 (10)**, pp. 1-23.
- Bezroukov, N. (1999b) "A Second Look at the Cathedral and the Bazaar", *FirstMonday: Peer Reviewed Journal on the Internet*, **4 (12)**, pp. 1-29.
- Brabham, D. (2007) "Crowdsourcing as a Model for Problem Solving: An Introduction and Cases" Working Paper, *University of Utah*,
- Chesbrough, H. W. (2006) *Open Business Models* Harvard Business School Boston, MA.
- Chesbrough, H. W., W. Vanhaverbeke and J. West (Eds.) (2007) *Open Innovation: Researching a New Paradigm* Oxford University Press London.

- Clarke, G. (2007) *Management 'Scared' by Open Source* Channel Register Address: http://www.channelregister.co.uk/2007/03/09/open_source_licensing/.
- Dickerson, C. (2004) "Open Source or Outsource?" in *InfoWorld*, IDG Network (October 22, 2004).
- Dinkelacker, J. and P. Garg (2001) "Corporate Source: Applying Open Source Concepts to a Corporate Environment (Position Paper)". in *International Conference for Software Engineering (ICSE) - 1st Workshop on Open Source Software Engineering*, Toronto, Canada, May 15th, 2001, ICSE and Hewlett-Packard.
- Dinkelacker, J., P. Garg, R. Miller and D. Nelson (2001) "Progressive Open Source" HPL-2001-233, *Hewlett-Packard*, Palo Alto, California.
- Erber, G. and A. Sayed-Ahmed (2005) "Offshore Outsourcing: A Global Shift in the Present It Industry", *Intereconomics*, **40 (2)**, pp. 100-112.
- Fitzgerald, B. (2006) "The Transformation of Open Source Software", *MIS Quarterly*, **30 (3)**, pp. 587-598.
- Fox, P. (2004) "Open-Source Model for Outsourcing". in *Computerworld*, January 26, 2004.
- Gaughan, G. (2007a) "D8.2: Inner Source Principles" D8.2, *University of Limerick*, Limerick, Ireland.
- Gaughan, G. (2007b) *Inner Source* (Personal communication).
- Gaughan, G., B. Fitzgerald, L. Morgan and M. Shaikh (2007) "An Examination of the Use of Inner Source in Multinational Corporations". in *1st OPAALS Workshop*, Rome, Italy, 26-27th November, 2007,
- Grand, S., G. Von Krogh, D. Leonard and W. Swap (2004) "Resource Allocation Beyond Firm Boundaries: A Multi-Level Model for Open Source Innovation", *Long Range Planning*, **37** pp. 591-610.
- Greenemeier, L. (2004) "Using Open Source as a Weird Form of Outsourcing". in *InformationWeek*, CMP Media (July 8, 2004).
- Hamm, S. (2007) "Open Source Meets Off-Shoring". in *BusinessWeek Online*, McGraw-Hill Companies (February 12, 2007).
- Harney, J. (2006) *Open Source Offshore Doubles Cost Cuts for Software Companies* OutsourcingCenter: An Everest Group Company Address: <http://www.outsourcing-information-technology.com/opensource.html>.
- Hars, A. and S. Ou (2000) "Why Is Open Source Software Viable? - a Study of Intrinsic Motivation, Personal Needs, and Future Returns". in *The 2000 Americas Conference on Information Systems (AMCIS 2000)*,
- Hars, A. and S. Ou (2002) "Working for Free? Motivations for Participating in Open-Source Projects", *International Journal of Electronic Commerce*, **6 (3)**, pp. 25-39.
- Hillery, G. A. (1955) "Definitions of Community: Areas of Agreement", *Rural Sociology*, **20** pp. 111-123.

- Hirschheim, R. and M. C. Lacity (2000) "The Myths and Realities of Information Technology Insourcing", *Communications of the ACM*, **43 (2)**, pp. 99-107.
- Hoepman, J.-H. and B. Jacobs (2007) "Increased Security through Open Source", *Communications of the ACM*, **50 (1)**, pp. 79-83.
- Howe, J. (2006a) *Crowdsourcing: A Definition: Tracking the Rise of the Amateur* [Weblog] WIRED Address: http://crowdsourcing.typepad.com/cs/2006/06/crowdsourcing_a.html.
- Howe, J. (2006b) *Customer-Made: The Site. The Site. Crowdsourcing: Tracking the Rise of the Amateur* [Weblog] WIRED Address: http://crowdsourcing.typepad.com/cs/2006/06/customer-made_th.html.
- Howe, J. (2006c) *The Rise of Crowdsourcing* WIRED Address: <http://www.wired.com/wired/archive/14.06/crowds.html>.
- Huston, L. and N. Sakkab (2006) "Connect and Develop: Inside Procter & Gamble's New Model for Innovation", *Harvard Business Review*, **84 (3)**, pp. 58-66.
- Huston, L. and N. Sakkab (2007) "Implementing Open Innovation", *Research - Technology Management*, **50 (2)**, pp. 21-25.
- Jaaksi, A. (2006) "Building Consumer Products with Open Source" White Paper, NOKIA,
- Jaaksi, A. (2007) "Experiences on Product Development with Open Source". in *Open Source Systems Conference IFIP WG 2.13*, Limerick, Ireland, June 11-14th, 2007, Springer.
- Jewkes, R. and A. Murcott (1996) "Meanings of Community", *Social Science and Medicine*, **43 (4)**, pp. 555-563.
- Jones, P. (2000) *Brooks' Law and Open Source: The More the Merrier? Does the Open Source Development Method Defy the Adage About Cooks in the Kitchen?* IBM Address: <http://www-4.ibm.com/software/developer/library/merrier.html>.
- Kaiser, K. M. and S. Hawk (2004) "Evolution of Offshore Software Development: From Outsourcing to Cosourcing", *MIS Quarterly Executive*, **3 (2)**, pp. 69-81.
- Kern, T., L. Willcocks and M. C. Lacity (2002) *Netsourcing: Renting Your Business Applications and Services over a Network*, Financial Times/Prentice Hall,
- Kogut, B. and A. Metiu (2001) "Open-Source Software Development and Distributed Innovation", *Oxford Review of Economic Policy*, **17 (2)**, pp. 248-264.
- Lacity, M. C. and L. P. Willcocks (2006) "Transforming Back Offices through Outsourcing" in *Global Sourcing of Business and It Services*, (Willcocks, L. P. and M. C. Lacity eds) Palgrave Macmillan, Basingstoke, Hampshire, pp. 1-34.
- Lakhani, K., B. Wolf, J. Bates and C. DiBona (2003) "The Boston Consulting Group Hacker Survey". in *Linux Conference Australia 2003*, Perth, Australia, January 22-25, 2003,

- Lakhani, K. R. and R. G. Wolf (2005) "Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects" in *Perspectives on Free and Open Source Software*, (Feller, J., B. Fitzgerald, S. Hissam and K. R. Lakhani eds) MIT Press.
- Ljungberg, J. (2000) "Open Source Movements as a Model for Organizing", *European Journal of Information Systems*, **9** (4), pp. 208-216.
- Metiu, A. (2006) "Owning the Code: Status Closure in Distributed Groups", *Organization Science*, **17** (4), pp. 418-435.
- Metiu, A. and B. Kogut (2001) "Distributed Knowledge and the Global Organization of Software Development" *Wharton School, University of Pennsylvania*, Philadelphia.
- Minoli, D. (2004) *A Networking Approach to Grid Computing*, Wiley-Interscience, New Jersey, US.
- Moody, G., R. Stallman, *et al.* (2006) "Gplv3: What the Hackers Said (Update)", *Linux Journal*.
- Netke, S. (2005) *Open Source Opens up the Offshore Market* ITBusinessEdge: NarrowCast Group Address: <http://www.itbusinessedge.com/item/?ci=17178>.
- Neumann, P. G. (2000) "Robust Nonproprietary Software". in *IEEE Symposium on Security and Privacy*, May 2000,
- Nisbet, R. (1966) *The Sociological Tradition*, Basic, New York.
- O'Reilly, T. (2005) *What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software* O'Reilly Media Last accessed: Last updated: Address: <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>.
- Oren, T. (2004) *Be Careful What You Wish For: Open Source and Off-Shoring* Tim Oren's Due Diligence: Letters from an inhabited dataspace Address: http://due-diligence.typepad.com/blog/2004/09/be_careful_what.html.
- Perens, B. (1999) "The Open Source Definition" in *Open Sources: Voices from the Open Source Revolution*, (DiBona, C., S. Ockman and M. Stone eds) O'Reilly, CA.
- Royce, J. (2006, 29th October) *Rent a Coder Going Downhill Fast* DEL.ISIO.US Address: <http://www.webtekconcepts.com/2006/10/29/rent-a-coder-going-downhill-fast/>.
- Sakkab, N. (2002) "Connect & Develop Complements Research & Develop at P&G", *Research - Technology Management*, **45** (2), pp. 38-45.
- Schneider, F. B. (2000) "Open Source in Security: Visiting the Bizarre". in *Proceedings of the 2000 IEEE Symposium on Security and Privacy (the ``Oakland Conference``)*, Berkeley, CA. Los Alamitos, May 14-17, pp. 126-127, IEEE Computer Society.
- Schneier, B. (1999, September 15th) *Open Source and Security* Crypto-Gram: Counterpane Internet Security, Inc Address: <http://www.schneier.com/crypto-gram-9909.html>.
- Schwarz, A., M. Mehta, N. Johnson and W. W. Chin (2007) "Understanding Frameworks and Reviews: A Commentary to Assist Us in Moving Our Field Forward by Analyzing Our Past ", *DATA BASE for Advances in Information Systems*, **38** (3), pp. 29 - 50.

Shaikh, M. (2007) *Version Control Software in the Open Source Process: A Performative View of Learning and Organizing in the Linux Collectif*. PhD Thesis, Information Systems and Innovation Group, London School of Economics and Political Science, London, UK..

Stallman, R. (1999a) *The Free Software Definition* Free Software Foundation Address: <http://www.gnu.org/philosophy/free-sw.html>.

Stallman, R. (1999b) "The Gnu Operating System and the Free Software Movement" in *Open Sources : Voices from the Open Source Revolution*, (DiBona, C., S. Ockman and M. Stone eds) O'Reilly.

Swanson, E. B. and N. C. Ramiller (1993) "Information Systems Research Thematics: Submissions to a New Journal, 1987-1992", *Information Systems Research*, **4 (4)**, pp. 299-330.

Thomas, C. W. and J. T. Parish (1999) "Co-Sourcing: What's in It for Me?" *Journal of Accountancy*, **187 (5)**, pp. 85-88.

Tsotra, D. and G. Fitzgerald (2007) "The Role of Culture in Global Is/It Sourcing". in *First Information Systems Workshop on Global Sourcing: Services, Knowledge and Innovation*, Val d'Isère, France, 13-15 March 2007,

Tuomi, I. (2002) *Networks of Innovation: Change and Meaning in the Age of the Internet*, Oxford University Press, Oxford.

Udell, J. (2003) "Open Source and Global Development". in *InfoWorld*, April 04, 2003.

von Hippel, E. (2001) "Innovation by User Communities: Learning from Open-Source Software", *MIT Sloan Management Review*, **42 (4)**, pp. 82-86.

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

von Hippel, E. and G. v. Krogh (2003) "Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science", *Organization Science*, **14 (2)**, pp. 209-223.

Walsham, G. (2002) "Cross-Cultural Software Production and Use: A Structural Analysis", *MIS Quarterly*, **26 (4)**, pp. 359-380.

Weber, S. (2004) *The Success of Open Source*, Harvard University Press,

Wheeler, D. A. (2003) *Secure Programming for Linux and Unix Howto*, (v3.010, 3 March 2003)

Witten, B., C. Landwehr and M. Caloyannides (2001) "Does Open Source Improve System Security?" *IEEE Software*, **September/October** pp. 57-61.

Woods, D. and G. Guliani (2005) *Open Source for the Enterprise*, O'Reilly, Sebastopol, California, US.

Zellen, B. (2005) *Three Sources of It Angst: Open Sourcing, Sole Sourcing & Outsourcing* EnterpriseInnovator Address: <http://enterpriseinnovator.com/index.php?articleID=4621§ionID=269>.