Open Source as an Innovation Enabler: Case study of an Indian SME

A dissertation submitted to the University of Manchester
for the degree of 'Msc in ICTs for Development'
in the Faculty of Humanities

Ms. Katja Henttonen
School of Environment and Development
Year 2011

No portion of the work referred to in the dissertation has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning

ABSTRACT

Free and open source software (FOSS) has been advocated for its presumed capacity to support native software industries in developing countries. It is said to create new spaces for endogenous innovation and to lower entry barriers to mature software markets, for example. However, little empirical research has been conducted concerning FOSS business in a developing country setting and, thus, there is not much evidence to support or refute these claims. This dissertation presents a business case study which was conducted in India, a country where the software industry has increased rapidly in volumes, while its innovative capability has remained relatively low. The goal of this dissertation is to accumulate an understanding on how small and medium sized software companies (SMEs) in India can benefit from FOSS. The focus is on its impacts on their innovativeness and the consequent ability to value chain upgrade.

The study is guided by Chesbrough's (2003) theory on Open Innovation. The literature review takes a critical look at Western business research on FOSS-based innovation and assesses its applicability to the Indian context. The empirical part consists of an organizational case study of an Indian software SME, which bases most of its product and service offerings on FOSS. The data collection involved interviewing directors and developers in the case company, studying relevant documentation and monitoring the interaction of employees with FOSS communities on the Internet. The data analysis relies on two qualitative methods: the Value Network Analysis (Allee 2003) helped to model how Open Innovation works in the case company and a thematic coding method was used to analyse their experiences with FOSS-based innovation processes.

The results illustrate some benefits of FOSS-based software development for Indian SMEs, especially when compared to working with proprietary software products. FOSS does not only come with the price tag of zero, but also with unlimited customization options, allowing a third-party company to add more value in-house. By becoming a contributor to FOSS development projects, an SME can also accumulate 'vendor-like' deep level expertise and even create an image of itself as a 'shaper' or 'visionary' of given technologies. On the other hand, the findings also indicate several major challenges, such as a poor availability of recruits with FOSS competence in India, resulting in consequent additional training effort. The study also shows that launching one’s own FOSS projects could be particularly difficult for Indian SMEs, hinting that most could only increment FOSS products that originate from the global North.

Acknowledgements

I want to sincerely thank my supervisor, Dr. Richard Duncombe from the University of Manchester for patiently guiding me through this work. He always replied to my queries very promptly and his critical yet positive comments helped me to improve the work. My gratitude also belongs to all people from Mahiti Infotech Private Limited as they welcomed me warmly to visit their company and spent their valuable time being interviewed. In addition, I must thank my dear life partner Jussi who gave some useful comments and proof-read the entire text.

This dissertation work has been financially supported by the research project ITEI-VTT, which is co-funded by VTT Technical Research Centre of Finland and Finnish National Technology Agency (Tekes).The project is part of EUREKA ITEA (Information Technology for European Advancement 2) cluster project called “Information technologies supporting the execution of innovation projects”. I have been employed by VTT Technical Research Centre of Finland since 2006 and also want to acknowledge my manager, Dr. Pekka Savolainen, for creating a pleasant working environment.

Table of Contents

LIST OF ILLUSTRATIONS

FRONT COVER: Collage of FOSS logos[1]

MAIN TEXT:

APPENDICES:

WORD COUNT: 16823 (including footnotes,excluding references)

1 Subject to a Creative Commons license. Downloaded from the following URL: http://www.flickr.com/photos/grokcode/3084292004/sizes/m/in/photostream/

LIST OF ABBREVIATIONS

CEO	Chief Executive Office
CMS	Content Management System
EMC	Egan Marino Corporation
FOSS	Free and open source software
GIS	Geographical Information System
GNOME	GNU Network Object Model Environment
GPL	GNU General Public License
IP	Intellectual Property
IRC	Internet Relay Chat
NGO	Non-Governmental Organization
OS X	series of Operating Systems developed and marketed by Apple Inc
QA	Quality Assurance
PR	Public Relations
R&D	Research and Development
SME	Small and Medium Sized Enterprise
SNA	Social Networks Analysis
T&M	Time and Material
UML	Unified Modelling Language
VNA	Value Network Analysis
XML	eXtensible Markup Language

Open source software as an innovation enabler:
A case study of an Indian software company

1. INTRODUCTION

The purpose of this chapter is to explain why this dissertation has been undertaken, what it aims to accomplish and how the research is conducted. The first subsection explains the motives and the broad aims of the dissertation. The second subsection introduces the key terminology and the third defines the scope of the research. The forth subsection spells out the research questions and illustrates the overall research design.

1.1. MOTIVATION AND AIMS

Free and open source software (FOSS) has been widely advocated as a way to promote endogenous software innovation in developing countries (e.g. Dravis 2003; Wong 2004; Weerawarana and Weeratunge 2004). The developmental opportunities created by the FOSS phenomenon have been noticed both by international development institutions (e.g. World Bank and UNDP) and many of the developing countries themselves (ibid). It has been argued that the nature of FOSS enables an ICT environment based on local autonomy and ownership, thereby creating new spaces for exploration and innovation (e.g. Dravis 2003). It's often stated that FOSS lowers entry barriers to mature software markets, due to smaller up-front investments and shorter times-to-market (e.g. Wong 2004). Further, many of the FOSS projects bring together top software engineering talent from all around the world and this is said to provide unique learning opportunities for the companies of developing countries (e.g. Tapia and Maldonado 2009). Following such hopes, several developing countries (e.g. Brazil, India and South-Africa) have founded "open source competence centres" which provide FOSS-related advice to local ICT companies (see e.g. Hoe 2006).

However, despite the enthusiasm, there remains very little empirical research on the how developing country companies could successfully integrate FOSS efforts into their internal

innovative activities. Studies on commercial FOSS in the US and Europe abound, but the results may not be directly applicable to the diverse innovation environments in the global South. This dissertation introduces a case study conducted in India, the country with the most well-known software industry in the developing world. The aim is to understand the opportunities and challenges that FOSS based innovation creates for small and medium sized software enterprises (SMEs) in India. The question asked is not whether FOSS offers an overarching solution, but rather how, if anyhow, these companies could best derive value from the growing phenomenon. The results should be interesting to Indian software entrepreneurs who are wondering what FOSS could offer them business-wise. They could also prove to be useful to public or social sector organizations, which support or plan to support commercial FOSS in India.

The dissertation may also make a small contribution to the wider discussion on how sharing and 'openness' impact software innovation. Until quite recently, much of the existing research on FOSS companies has focused on how to protect innovation in a different licensing environment. However, as Henkel (2006) points out, the focus should be on appropriating profits from innovation, not on the protection per se. For a large number of successful FOSS companies, the free revealing of internal innovation is a deliberate decision, which is consistent with profit-maximizing behaviour (ibid). This study explores commercial FOSS, not through traditional business models, but rather in the context of the recent tendency of companies to open-up their innovation processes. This leads to some interesting questions e.g. How the commercial FOSS contents the picture of the innovative process driven by intellectual property rights.

1.2. KEY CONCEPTS

Per definition, *free and open source software* (FOSS) is distributed with source code and under a license, which allows the recipient to freely redistribute verbatim or modified copies. (Open Source Initiative 2010; Free Software Foundation 2010). FOSS developers typically communicate over the Internet, as they co-operate over geographical and organizational boundaries (Raymond 2001). It has traditionally been a volunteer-driven movement, yet, by today, a slight majority of the FOSS development is performed by private companies

(Fitzgerald 2006). Over the past two decades, company investments in for-profit FOSS endeavours have increased exponentially (Lerner and Tirole 2002; UNU-MERIT 2006). For the sake of brevity, the word pair "commercial FOSS" is occasionally used herein to refer to all scenarios where companies become involved with FOSS projects with a profit-making intensive.

Innovation refers to new ideas which are successfully put into practice (Pol and Carroll 2006). Following Schumpeter (1939) and others, it is typically distinguished from invention, which is a mere embodiment of an idea (ibid). Innovation has been defined as an invention which creates a commercial or public value (ibid) or, as Tuomi (2002:11) puts it, results in a "meaningful change in social practice." From a company viewpoint, innovation typically entails improvements to products, internal operations or a mix of markets (e.g. Johne 1999). The terms *innovativeness* and *innovative capability* are used to refer to the ability of a company to continuously generate innovations (see Subramaniam and Youndt 2005). The term *innovative activities* is used to refer to any significant participation in the discovery, incubation and/or commercialization of an innovation (see Tuomi 2002).

1.3. SCOPE AND CONTENT LIMITATIONS

As discussed previously (see section 1.1), the study focuses solely on software SMEs in India. FOSS business studies conducted in other countries are covered in the literature review, but their meanings to Indian SMEs are discussed separately. The study does not address questions that *only* concern large companies or micro-enterprises. It targets the *primary software sector*, i.e. companies that get most of their revenues from software developed and/or associated service provisions. While FOSS can enable other types of business models (e.g. in the fields of e-commerce and hardware sales), these are not addressed. Most of the legal issues concerning FOSS licensing terms and parents are also remain beyond the scope of the study.

As illustrated by **Figure 1,** Vanhaverbeke and Cloth (2006) propose that innovation can be investigated at five different levels of analysis**.** This study aims to *primarily* analyse innovation at the firm level; some other perspectives complement the analysis. Largely due to the underlying theory, the focus is on the product and market innovation. Costs and time

savings (essentially a process innovation) are also discussed, but the emphasis remains on their ability to remove the constraints to product development or improve market diversification. The study mostly talks about incremental innovations, which SMEs can generate on a regular basis, rather than radical innovations which transform entire industries.

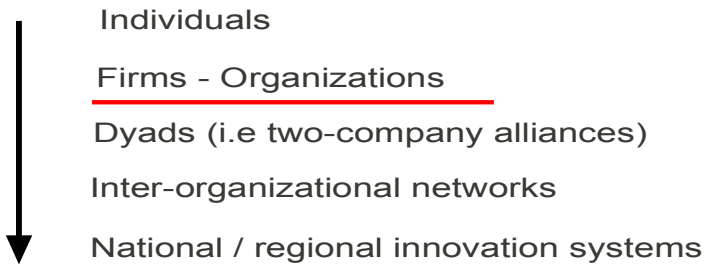

Figure 1: Levels of analysis in innovation research
(Vanhaverbeke and Cloth 2006)

Since the study targets private companies, the profitability and market performance of FOSS based innovations gains more attention than their public or social value. The connection between the innovativeness of private companies and socio-economic development is not discussed in detail. The study shares a common belief that commercial innovations are an important source of wealth creation in the current economic system (see Ulin and and Brown 2004, Pol and Carroll 2006). Due to wealth distribution aspects, the innovativeness of the SME sector can be seen as particularly important to socio-economic development (see e.g. World Bank 2004).

1.4. RESEARCH APPROACH AND DISSERTATION STRUCTURE

The overall research design of the dissertation is illustrated by **Figure 2** (on the next page). Its theoretical basis lies on Chesbrough's (2003) Open innovation paradigm, which is complemented by Allee's (2003) conception of value networks. The literature review looks at Western business research on FOSS and points out special characteristics of the Indian innovation environment, ending with an analysis of how Western research findings apply to the Indian context. The empirical part involves a single case study of an SME-sized Indian software producer, which is heavily engaged in FOSS activities. Data is collected by conducting semi-structured interviews, reading company documentation and observing

online interaction on FOSS forums. Two qualitative methods, namely Value Networks Analysis (Allee 2003) and Template Analysis (King 1998), are employed to analyse the primary data.

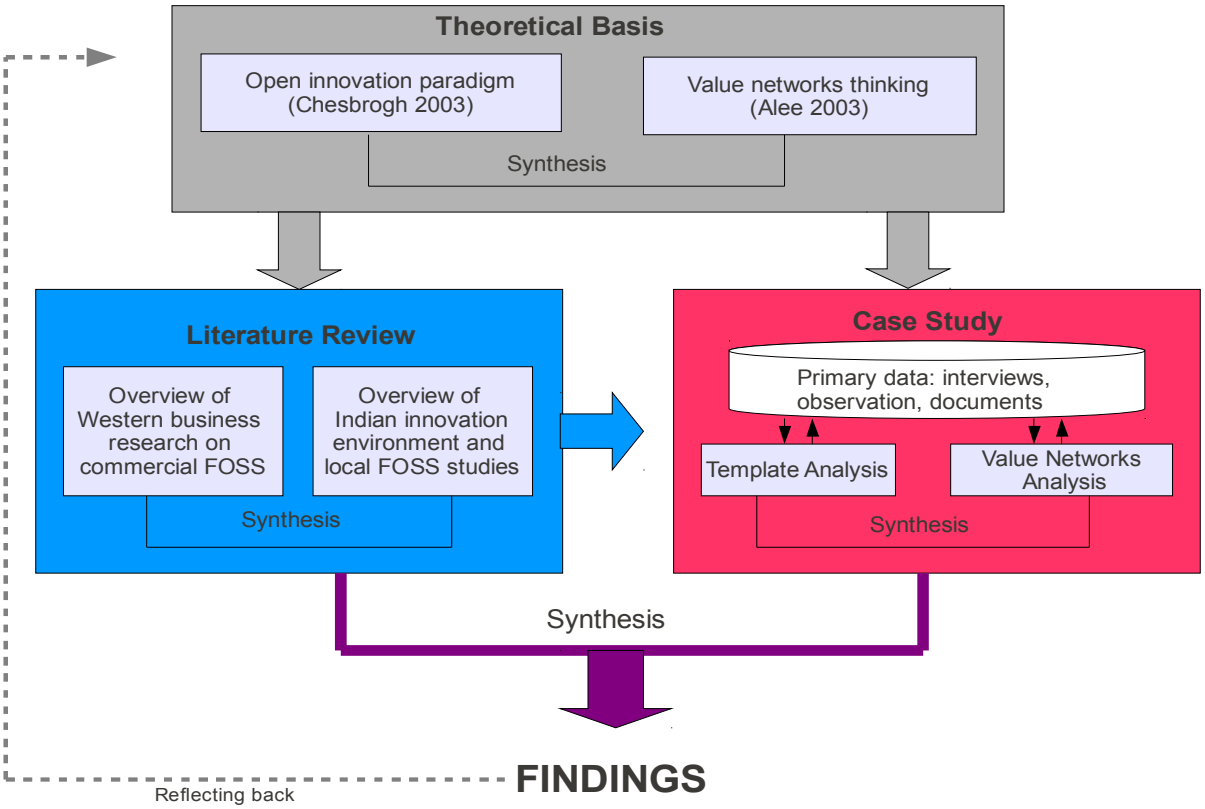

Figure 2: Research design of the dissertation

For the purposes of this study, the research questions are articulated as follows:

➔ *How can Indian software SMEs benefit from FOSS based innovation processes?*

➔ *What are the associated opportunities and challenges for them, especially considering the common constraints to software innovation in India?*

Obviously, a single case study cannot provide a conclusive answer to these questions. However, once empirical results of the case study are compared to the literature review findings and the underlying theory, they can contribute towards understanding the issues (cf. Yin 2009; Simons 2009).

The remainder of the dissertation is structured as follows:

- ➔ The second chapter introduces the theoretical underpinnings of the study.
- ➔ The third chapter presents the literature review findings.
- ➔ The fourth chapter spells out objectives for the case study and discusses methodological issues in more detail.
- ➔ The fifth chapter presents the actual case study results and analyses them against the literature review findings.
- ➔ The sixth chapter summarizes conclusions, along with associated limitations, and discusses their implications for further research.

2. THEORETICAL BASIS

This chapter introduces theoretical concepts which guide both literature review and methodology. The first subsection lists academic antecedents to Chesbrough's (2003) Open innovation theory and explains why it was selected as the basis of the dissertation. The second subsection describes the basics of the theory and introduces a way to categorize the Open innovation processes. Based on Allee (2003) and others, the third subsection explains how commercial FOSS can be understood as Open innovation.

2.1. COLLABORATION IN INNOVATION RESEARCH

Over the past three decades, collaboration and networks have come to the fore in innovation research (see Tuomi 2002 and Chesbrough 2006a for a review). A wide body of academic work has addressed the importance of external technology and knowledge sources to innovative organizations (ibid). Several theories and frameworks (e.g. von Hippel 1988; Arora and Gambardella 1994; Zahra and George 2002) have been created, concerning how companies can seek external technology and knowledge, e.g. from customers, supplies and competitors, and how they can develop what Cohen and Levithal (1989) termed 'absorptive capacity'. Building on insights emerging from social and cognitive sciences, researchers have also developed models (e.g. Gomes-Cassares, 1996; Dyer and Nobeoka 2000; Ahuja 2000) on how companies can best utilize social and inter-organizational networks in order to improve their innovative capability. Many of these ideas can and have been used in research on commercial FOSS (e.g. von Hippel 2001; Grand et al. 2005, Gloor 2006).

This dissertation, however, was selected to be based on a rather recent theory of the collaborative nature of innovation: Chesbrough's (2003; 2006b) Open Innovation paradigm. The theory differs from antecedent works in ways that make it particularly suitable for the purposes of the dissertation. One essential difference is the centrality of the business model to the Open innovation theory (Chesbrough 2006a). Chesbrough clearly takes the perspective of a single company (as in section 1.3) and does not overlook the basic question of “where does the money come from?” (cf. Vanhaverbeke 2006). Another distinction relates to the considerable attention given to *purposive* outbound IP flows; prior innovation theories

regard outward IP flows as mostly unwitting “spill-overs” (Chesbrough's 2006a). For these reasons, Chesbrough's ideas resonate well with FOSS business literature, which has been oriented towards identifying business models and explaining why companies voluntarily give out their own IP. While early publications on Open innovation (e.g. Chesbrough 2003; Gasman and Enkel 2004) were focused on large, incumbent companies, the theory has been applied to the SME context in several follow-up works (e.g. Enkel and Gassman 2008, Vrande et al 2009, see also Chesbrough 2006b). Unsurprisingly, it has also been quickly taken up by FOSS business researchers (e.g. Henkel 2006; West and Gallagher 2006b; Huurinen et al. 2006).

2.2. OPEN INNOVATION AND ITS ARCHETYPE PROCESSES

Chesbrough (2003; 2006a; 2006b) argues that the profits that companies can expect to gain from the traditional vertical innovation process are declining. In the traditional model of ”closed innovation”, internal research and development (R&D) activities feed the production pipeline and internally produced products (and/or services) are brought to the market by the company. In contrast, businesses that adopt the new Open innovation model draw on a wide range of external knowledge sources to feed their innovative activities. They incorporate internal and external ideas to create value and have internal mechanisms towards claiming a proportion of that value. Open innovators also believe that some internal ideas can be successfully commercialized through external path-ways to market, such as out-licensing or a spin-off venture company. **Figure 3** (on the next page) illustrates the difference between a closed and Open innovation model as depicted by Chesbrough (ibid). The defining characteristics of Open Innovation are highlighted with red colour.

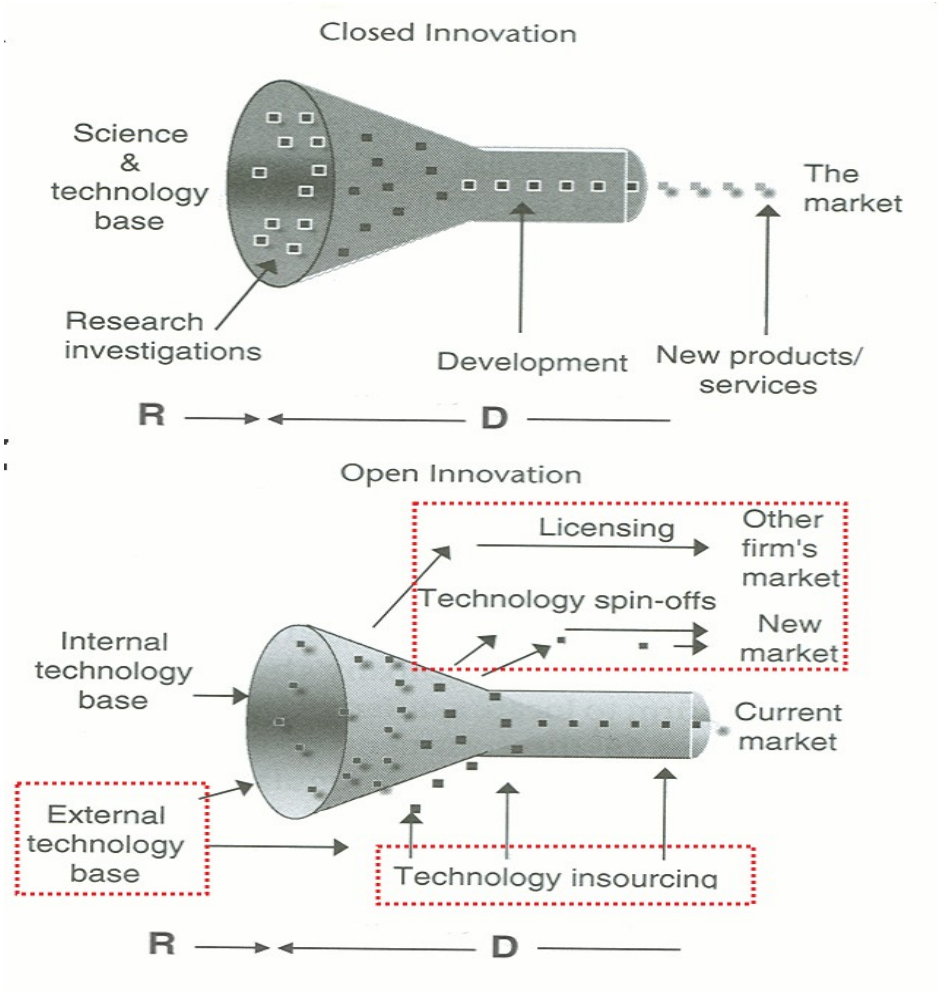

Figure 3: Closed and Open innovation as two different R&D pipelines (Cheshbrough 2006a; red highlighting added)

Based on extensive empirical studies involving companies of all sizes, Gassman and Enkel (2004; Enkel and Gassman 2008; Enkel et al. 2009) have identified three archetype processes in Open innovation. In the *outside-in process*, a company encourages and exploits inflows of external knowledge and integrates it with internal innovative activities. This is said to grow out from the realization that the locus of innovation can be different from the locus of knowledge creation. In the *inside-out process*, a company exploits external channels for bringing its own ideas to the market. It reflects the experience that the locus of exploitation can be different from the locus of innovation. In the *coupled process*, companies combine two previously introduced processes by collaborating tightly with other innovators.

This process is characterised through profound cooperation with the same partners over a long-period of time. **Figure 4** illustrates the three process archetypes. According to Gasman and Enkel (ibid), most companies select one primary process while still incorporating elements from the two others.

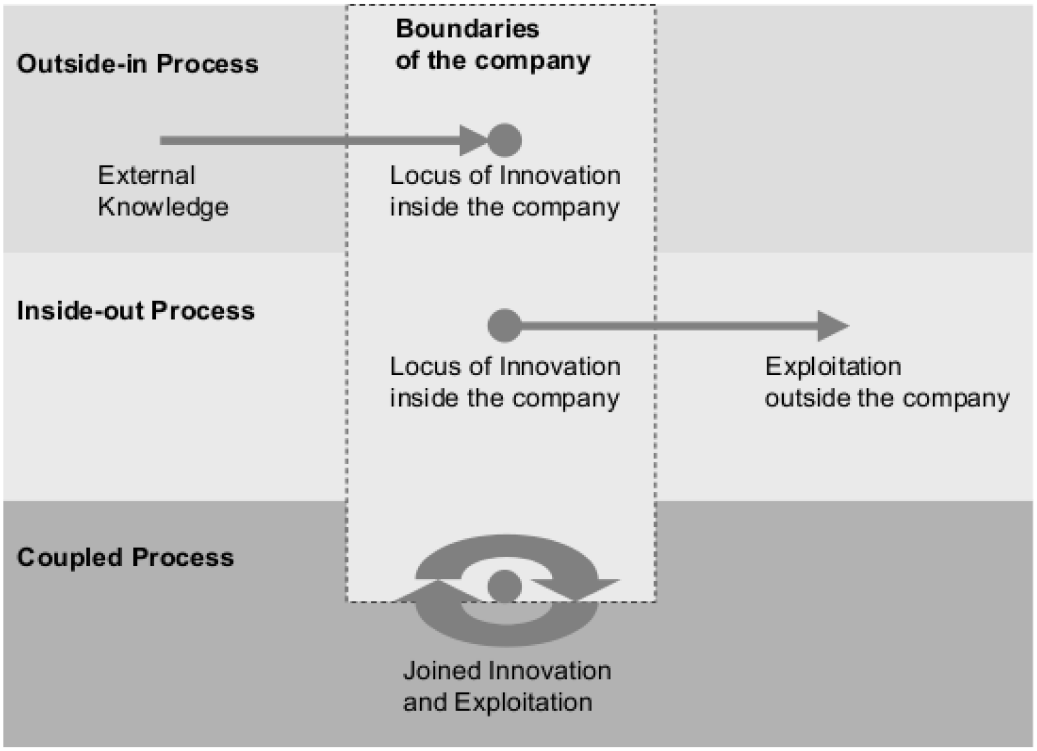

Figure 4: Archetype processes of Open innovation (Gassman and Enkel 2004)

2.3. OPEN SOURCE AS OPEN INNOVATION

This study does not presume an automatic link between FOSS and innovation. As West and Gallagher (2006b) point out, most FOSS projects cannot be described as innovative, as they never result in value creation or social change (cf. section 1.2). However, many well-known FOSS projects, such as Apache or WikiMedia, clearly create value for individuals, companies and the society at large (Tuomi 2002; Goldman and Gabriel 2005). From a company perspective, open source becomes Open innovation when it's combined with a sustainable business model (West and Gallagher 2006b). While commercial FOSS is seen as a significant example of Open innovation (e.g. Chesbrough 2006b; Henkel 2006), it also challenges the paradigm (West 2006). For instance, it is often presumed that Open

innovation necessarily requires stronger intellectual property (IP) laws in order to allow selling and buying IP in the market place (ibid). However, the success of commercial FOSS seems to counter this argument, as FOSS production depends on a “relaxed” IP regime instead (ibid).

Allee's (2003) theory on intangibles can help us understand the apparent contradiction. Her conception of value networks (ibid) acknowledges two ways to bring intangibles to the market: through monetary exchanges and through barter. She argues that any contemporary business activity involves highly-sophisticated barter systems for exchanging intangible assets such as knowledge. The concept of barter seems well-suited for analysing FOSS development, which has been repeatedly likened to both the gift economy (e.g. Bergquist and Ljungberg 2001) and barter economy (e.g. Hubbard 2004). In an extensive empirical study, Henkel (2006) found that, despite the existence of an explicit barter contract, commercially-motivated FOSS exchanges involve an anticipation of return and are based on rational cost-benefit considerations. A company may release some source code to the FOSS domain to gain design help or marketing visibility, for example (ibid).

Based on the above, it can be concluded that, while the Chesbrough's (2003) Open innovation paradigm places emphasis on the monetary exchanges of ideas and knowledge (e.g. licensing of patents), FOSS companies rely on barter systems in their Open innovation processes. By receiving and giving out bartered “gifts”, they attempt to exploit external innovation and find alternative ways to market for internal innovation (West and Gallargher 2006a). **Figure 5** (on the next page) illustrates how the Open Innovation model applies to a FOSS company; monetary and barter exchanges are high-lighted with separate symbols. The aforementioned archetype process of Open innovation by (see section 2.2) can also be applied to commercial FOSS, as will be shown in the next chapter.

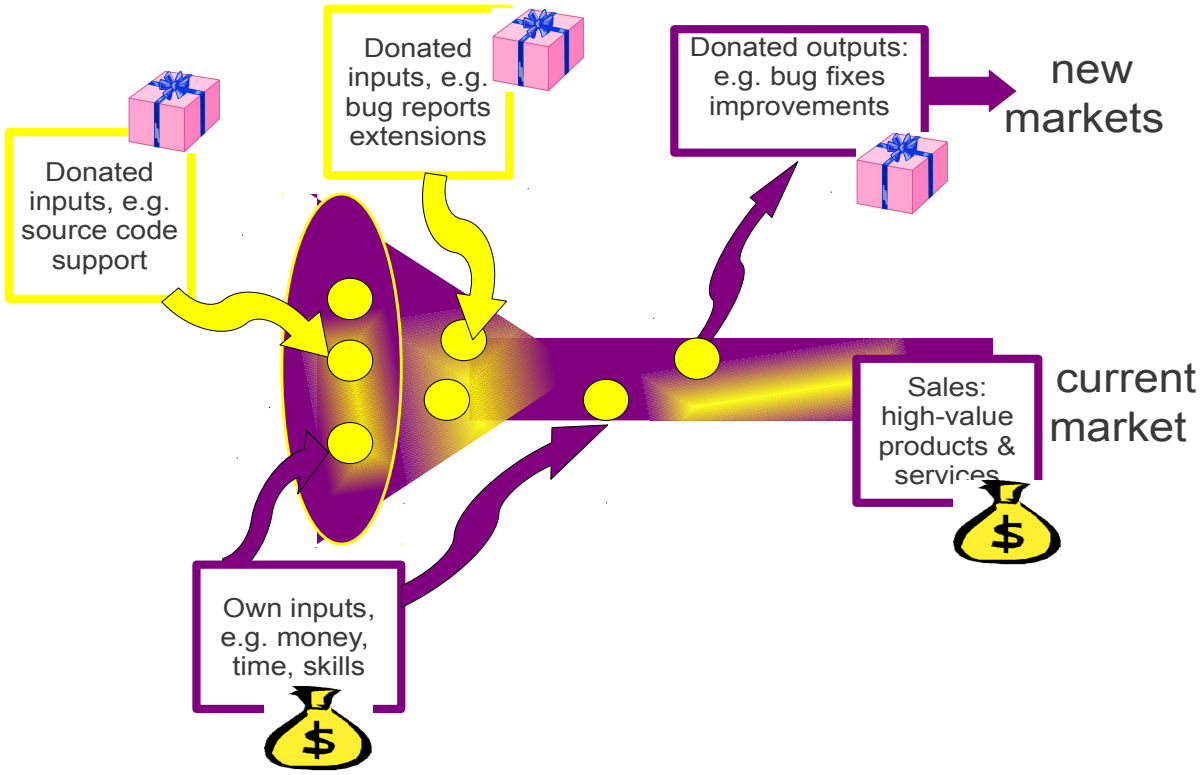

Figure 5: Open innovation in a FOSS company (adapted from Chesbrough 2003)

3. LITERATURE REVIEW

The theoretical concepts introduced in the previous chapter provide a starting point for this literature review. The chapter is composed of three parts. The first section describes FOSS based innovation processes, based on studies conducted in Europe and the US. The second section provides a brief overview of the Indian software innovation landscape and local innovation constraints. The third section assesses what FOSS based innovation processes could (not) offer for SMEs in the Indian context.

3.1. FOSS BASED INNOVATION PROCESSES

Based on West and Gallagher (2006a; 2006b) and others (e.g. Goldman and Gabriel 2005), this subsection explains how the three archetype processes of Open innovation (see section 2.2) apply to commercial FOSS. Further, a wide range of Western research and business literature is explored in order to identify the main advantages and drawbacks of each process archetype, especially from the SME perspective. **Table 1** (on the next page) summarizes the core characteristics of FOSS-based Open innovation processes. For those who are unfamiliar with FOSS business literature, **Appendix A** provides basic information on FOSS business models and clarifies how their map to the process archetypes introduced the Table 1.

Table 1: FOSS based innovation processes summarized (adapted from West and Gallagher 2006b; Gasman and Enkel 2004; Chesbrough 2006b)

	Outside-in process	**Inside-out process**	**Coupled process**
What a company does?	Takes FOSS code and incorporates it into one's own commercial products	Turns an internal development project into a new FOSS project	Contributes to an existing FOSS project on a long term basis
What is the purpose?	To save one's own resources and free them to more innovative activities	To advance promotional or strategic goals with non-commercial technology	To join forces to achieve a shared goal, build expertise, get visibility
What is the role of external innovation?	Free-of-charge IP as an alternative to 'buy or build'	One's own technology supplanted as a basis for ongoing innovation	Pooled contributions available to all, inter-organizational learning
How external innovation is motivated?	Motivation issue remains largely unsolved	Free access to valuable technology is supposed to motivate	Ongoing institutions establish continuity and legitimacy
What are the main challenges?	Incorporating external and internal IP . sustaining inbound IP flows	Affordability, sustaining third-party interest, recovering investments	Co-ordinating efforts, consolidating conflicting interests

The following subsections will expand on the three archetype processes described above.

3.1.1. Outside-in process: scouting for external IP

For software producers, the outside-in approach to FOSS typically entails the utilization of free assets (e.g. open source code and technical documentation) in the production of proprietary software products (see e.g. West 2003). The derived product(s) can go to market as an application service, as bespoke software or, FOSS licensing terms allowing, as closed packaged software (e.g. Koenig 2004, Krishnamurthy 2005, see also **Appendix A**). To exemplify the latter, Gentleware's commercial UML tools are almost entirely based on open source code developed by the ArgoUML community (Persson et al. 2006). In the long run, many of these companies make modest efforts to motivate external innovation[2], while

2 West and Gallagher (2006b) argue that, if nothing is done to address the motivation issue, one should not even talk about Open innovation, but use another term, such as "externally oriented innovation" instead.

maintaining low costs (see e.g. Dahlander and Mangnusson 2005). For example, Gentleware gives out free-of-charge software licenses to FOSS communities (Gentleware 2010).

The approach may seem optimal from the viewpoint of maximizing profits: external IP is acquired at zero or low cost, without an obligation to relieve any of one's own IP. Several empirical studies (e.g. Ajila and Wu 2007, Huurinen et al. 2006; Kenwood 2001) support the idea that the outside-in process entails a possibility for significant cost savings and a faster time-to-market. The cost savings can improve profit margins (e.g. Puhakka et. al. 2007), lower entry barriers to mature software markets (e.g. Bitzer 2004) and/or create new markets due to an increased affordability to customers (cf. Messerschmitt 2004). In addition, the literature mentions non-financial benefits such as improved customizability (e.g. Forge 2006) and architectural robustness (e.g. Norris 2004). These vary a great deal depending on how the FOSS solutions are selected and whether they are an alternative to buying software or building it in-house (e.g. ITEA 2004).

However, some popular FOSS licenses, most notably GPL, seriously limit some profit-making opportunities, especially selling packaged software. This can decrease the competitive benefits derived from the cost and time savings (e.g. Schmidt and Schnitzer 2003; Debroy and Morris 2004). Many companies also worry about not having the sufficient expertise to assess the quality of FOSS code or integrate it successfully with their own code base (e.g. ITEA 2004; Merilinna and Matinlassi 2006). Large-scale code reuse is known to require more advanced skills than working with commercial off-the-shelf components; many consider it more difficult than writing code from scratch (Spinellis 2003; Dibona 2005). The unavailability of commercial support may contribute to technical problems (see e.g. Fitzgerald and Kenny 2004). While some report very positive experiences of community support (e.g. Norris 2004), many companies view it as unreliable (Morgan and Finnegan 2007). Further, since the outside-in company does little to motivate inbound IP flows, there are major business risks (West and Gallagher 2006a). For example, the company has no 'say' if the FOSS community ceases existence or takes the development into a direction which is unfavourable to the company (Dahlander and Mangnusson 2005). To mitigate risks and ensure sustainability, many outside-in companies move towards the coupled process when the importance of a FOSS product to their business grows beyond a certain threshold (see e.g. Grand and al. 2004)

3.1.2. Inside-out process: why give out one's own IP for free?

The inside-out process entails what West and Gallagher (2006b) call a FOSS "spin-out": a company transfers an internal development project into an externally visible open source project. Economic researchers have been intrigued by the question on how such a spin-out can create value. One answer is that the donated IP creates a demand for the other products and services that the company is selling (e.g. West and Gallagher 2006a; Karels 2003). The popularity of the open source version of the MySQL database created the demand for commercial software libraries, for example (Rajala et al. 2007). Another answer is that a company can use FOSS as a means to advance complex strategic goals, e.g. establishing standards or undercutting enriched competition (Goldman and Gabriel 2005). For example, the goal of OpenOffice was to create an XML standard for file formats (ibid). A third answer lies in a so called "sell-it-free-it" model (Hecker 1999): a product may be transferred into the FOSS domain at a later part of it's life cycle, when it has stopped creating profits as a proprietary one. There may be promotional benefits from open sourcing or the company may seek to reduce maintenance costs, for example (Henkel 2006).

For SMEs, the greatest obstacle seems to relate to the high cost of FOSS spin-outs (Goldman and Gabriel 2005, Dahlander 2005). Fogel (2006) points out that there are hundreds of thousands of FOSS projects and it takes a lot to "stand out from the crowd". One needs to develop a very good quality base product and then market it intensively for both users and developers (ibid; see also Henttonen and Matinlassi 2007). Meanwhile, empirical studies (e.g. Harison and Koski 2008; Dahlander 2005) suggest that recovering the investment can be difficult. For example, with FOSS "loose leader" models, one needs to acquire a sizable user base to get even a few paying customers (see e.g. Rajala et al. 2007). While some large corporations, such as IBM, have benefited greatly by strategically using FOSS spin-outs (e.g. Gloor 2006), only a handful of SMEs seem to have succeeded with this approach (Wichmann et al. 2002). According to Wichmann (ibid), the same rule applies to basic research and FOSS spin-outs: large companies can typically capture indirect returns better than small ones (ibid). The aforementioned 'sell-it-free-it' model can first seem like a low-cost effort, but it still requires investments in infrastructure building, marketing and often the re-engineering of a monolithic software product into a modular one (Fogel 2006). Outsiders

will hardly have any interest in a FOSS project, which seems as if its a cemetery for unwanted code (e.g. Engelfriet 2009).

3.1.3. Coupled process: leveraging value through intense collaboration

The coupled process is all about collaboration. Private companies donate their R&D effort to a FOSS project, while exploiting the pooled R&D of all the contributors in their own innovative activities (West and Gallagher 2006b). For example, once Unix/Linux vendors were left without a supported Internet browser, they joined efforts to develop Mozilla Firefox (ibid). The coupled process can be seen as a combination of (or a compromise between) the outside-in and inside-out processes. Cost and time savings are achieved through inbound IP flows in largely the same way as in the outside-in process (see Dahlander and Magnuson 2005; Goldman and Gabriel 2004). Meanwhile, the participation in R&D gives a company more influence over the development of the FOSS product and long-term collaboration creates a degree of continuity (Dahlander and Magnuson 2005; West and Gallarger 2006b). The outbound IP flows of the coupled process can support promotional goals in a way which is similar to the inside-out process (see Goldman and Gabriel 2005; Henkel 2006). It is more about improving visibility on the existing market and less about creating new markets, however (ibid). An obvious difference is that joining an existing project is generally less costly and laborious than starting a new one (ibid).

As for inter-organizational learning, the coupled process holds a special importance. FOSS has often been credited for its effective training mechanism, where learning occurs through the gradual deepening process of participation (e.g. Reinhardt and Hemetsberge 2007, Ye and Kishida 2003). As illustrated by **Figure 6** (on the next page), newcomers start with peripheral participation and gradually proceed towards more advanced tasks and become experts. However, this commonly quoted “onion model” captures only one aspect of FOSS-related learning: building technical expertise. Domain and business knowledge is accumulated through participation in virtual networks, where users provide feedback on innovations or donate innovative complements (Hippel 2001; Gloor 2006). For example, many innovations, which ensure the nice 'look and feel' of the OS X operating system, are said to originate from the various GUI-intensive FOSS projects supported by Apple (Goldman and Gabriel 2005). In addition, different types of informal social ties between

FOSS developers, users and advocates enable learning for the innovation beyond project boundaries (e.g. Tuomi 2002). The diversity of participants can indeed boost innovation, yet it also contributes to the greatest challenge of the coupled process: consolidating conflicting interests (West and Gallagher 2006b; Chesbrough 2006). For example, tensions may escalate between for-profit companies, NGOs and volunteer individuals who each have very different skills, 'agendas' and motivations for participating in a FOSS project (Dahlander and Mangnusson 2005).

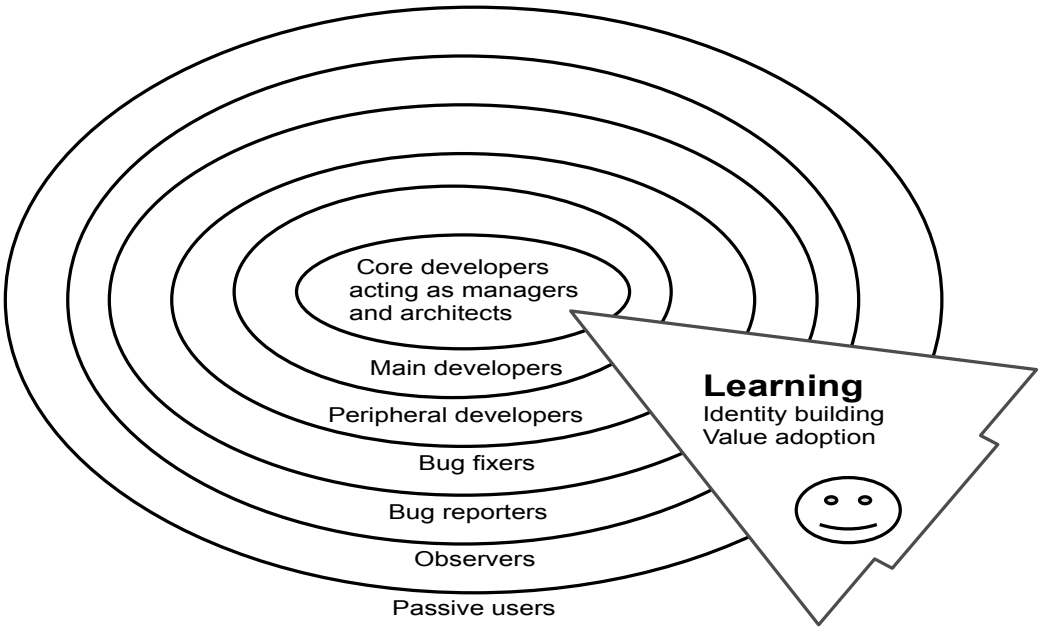

Figure 6: Participative learning in a FOSS community (adapted from Ye and Kishida 2003; Reinhardt and Hemetsberge 2007)

3.2. INDIAN SOFTWARE INNOVATION ENVIRONMENT

This section briefly describes how the Indian software innovation environment differs from that of the US or Europe. The first subsection provides some key figures on the production of proprietary and FOSS software in the country. The second subsection overviews the general innovation constraints faced by Indian software entrepreneurs. Building on the second subsection, the third one spells out the challenges that Indian software SMEs have to solve in order to move up the value chain.

3.2.1. Indian primary software sector in figures

Indian software exports have grown exponentially (by 20-99% each year) over the past two decades (Athreye 2005, Arora 2008). In 2007, India's software product and service exports, excluding back-office services, amounted to US$23000 million (IDMP 2010). India has also created itself a valuable country brand as the software power house of the developing world (Athreye 2005). However, observers (e.g. Kambhambati 2002; Heeks 2006; Arora 2008) have pointed to a tacit problem behind the success story: the innovative capability of India's software industry has remained relatively low. The vast majority of India's software exports consist of low-value development and testing services, such as the maintenance of legacy systems (D'Costa 2004; Illiyan, 2005; NASSCOM 2008). Less than 5 % of revenues come from the sales of packaged software and less than 10% come from what were described as “high value” services (ibid). Even though they have been slowly rising over the past decades, the average revenue-per-employer-ratio of Indian software companies ($34,000 in 2006) is still dozens, even hundreds, of times smaller than that of their Western counterparts (D'Costa 2002; Ahora 2008).

As for free and open source software, its usage have recently expanded in India, however, FOSS production has not become prolific in the country (UNU-MERIT2007d). While increasing the number of Indians joining FOSS communities, most participate in social activities rather than product development (ibid). For example, the largest open source development platform, SourgeForge, has approximately 22 000 Indian members (1-2% of all the members), but less than 200 of them contributed any source code (ibid). The vast majority of FOSS development (80-90%) is still performed in the US or Europe (Robles and Gonzalez-Barahona 2006**;** UNU-MERIT2007b). There are no statistics on commercial FOSS in India, but a survey concerning Indian FOSS community members (UNU-MERIT 2007c) suggests a weak phenomenon. Only 12% of the respondents said that they were paid for their FOSS involvement (ibid), compared to over 60 % in the US and over 40 % in Brazil (UNU-MERIT 2007a), for example.

3.2.2. Software innovation constraints in India

Studies suggest that Indian software companies are well aware of the need to upgrade in the value chain, but face many barriers to doing so (e.g. Ahora 2008). Due to scarce venture capital support and a small own revenue base, many cannot afford the large up-front investments required by the development and marketing of packaged software products (D'Costa 2002; Heeks 2004). This also relates to a late-comer disadvantage: a large number of modern software applications have taken thousands of man-years to develop and it's getting harder for new entrants to catch up (Wong 2004). Unable to compete with the salaries of multinationals, many indigenous companies also suffer from a drain of employees (D'Costa 2004; Illiyan 2005). Further, a limited domestic demand has caused the Indian software industry to focus on spatially distant markets (Parthasarathi and Joseph 2004; Ahora 2008). However, geographical and cultural distances limit the interaction between users and producers; something which is known to be a key factor in product innovation (ibid). The overt export orientation has also lead to weak ties between domestic companies (D'Costa 2002).

Due the constraints related to finances, human resources and geographical position, most Indian software companies are attracted to the low-entry-barrier off-shoring markets (Ahora 2008). The off-shoring work provides limited profit-making and learning opportunities, however (D'Costa 2002; Parthasarathi and Joseph 2004). To maximize own profits and mitigate risks, Western companies off-shore less critical, routine tasks, while keeping high value adding tasks to themselves (ibid). As the largest beneficiaries of the “international skills division of labour”, they are motivated to prevent knowledge spillovers (D'Costa 2005). Further, most off-shoring work is priced on the time and material used instead of the value-added, and the resulting intellectual property is assigned to the parent company (Ahora 2008). This provides little incentive to innovate and hinders the opportunities to benefit from the “economies of repetition”; many Indian companies reuse as little as 5% of the source code that they produce (ibid). D'Costa (2002) argues that this “low-value trajectory” can become self-enforcing, as off-shoring work rarely helps to build the financial and human capital required by more innovative activities.

3.2.3. Innovation challenges for Indian software SMEs

The aforementioned innovation constraints impact also Indian SMEs, probably even more than they impact large firms (Chaminade and Vang 2006; Nirjar and Tylecote, 2005). Typically, SMEs have even more limited financial and human resources and can therefore invest less in R&D and product marketing (ibid). The liabilities of smallness can also make it more difficult to build extensive external networks for learning or get high-value assignments at long distance (Nirjar and Tylecote, 2005). On the other hand, SME's have some advantages, for example, in a smaller unit, it can be easier to generate effective employee commitment (ibid). SME's can also be more 'agile' to make organizational changes required to make a value-chain upgrade (Ahora 2008) .

Nevertheless, Indian software SME need to find some creative solutions to their challenges. First, they should seek ways to overcome resource constraints, e.g. through domain specialization or collaboration with other SMEs (De'Costa 2004; Nirjar and Tylecote, 2005). Further, because alliances with foreign companies have not contributed to domestic innovative capability as much as many had hoped for, they need to find alternative routes to learning for innovation (D'Costa 2002; Parthasarathi and Joseph 2004). There is also a need to improve market diversification, especially to expand exports markets beyond the US and UK (Illian 2005). Ahora (2008) suggests targeting more software products to regional markets, where Indian companies have an advantage of “geographical and cultural proximity” over US competitors. One of the most successful Indian software producers, i-Flex, targeted its solutions towards East Asian banks, for example (ibid).

3.3. FOSS BASED INNOVATION IN AN INDIAN CONTEXT

This section discusses how previously introduced FOSS-based innovation processes could 'fit' into an Indian context. The emphasis is on what they can and cannot offer in terms of helping Indian SMEs to overcome the innovation challenges outlined in the previous section. This section brings forth some theoretical viewpoints presented by academics and reviews two empirical studies on commercial FOSS India[3]. Where empirical evidence from India is

3 The author was able to locate only these two empirical studies (Madanmohan and De and UNU-MERIT 2007 b/c) which address FOSS exploitation or production on the *primary software sector* in India.

non-existent, case examples from other developing countries are taken for illustrative rather than argumentative purposes.

3.3.1. Outside-in process: overcoming resource constraints

For the reasons previously outlined (see section 3.2.2), the cost and time savings (see section 3.1.1) should be particularly important for Indian software SMEs. Through multiple case studies conducted domestically, Madanmohan and De (2004) found evidence of significant cost savings from FOSS reuse. In some cases, the cost of acquiring a FOSS component was estimated to be as low as 1/1000 of acquiring a proprietary component. As FOSS was used as an alternative to buying rather than building, time-to-market advantages were not relevant. The interviewees seemed to have a positive view of FOSS community support and, in fact, were occasionally overtly dependent on it due to the lack of their own expertise. The majority of case companies used FOSS components or platforms as “black boxes”, lacking knowledge, skills and/or manpower to modify or even read the source code. The commercially-restrictive terms of some FOSS licenses were also a matter of concern to most case companies (ibid).

While providing positive evidence on cost savings, the above findings also highlighted some aforementioned problem areas in the outside-in process. (cf. section 3.1.1) They also raise questions concerning whether economic and skills constraints prevent Indian companies from moving towards the coupled process (as described in section 3.1.1) and 'lock' them to the passive consumption of FOSS components. As with GPL and alike FOSS licenses, some critics (e.g. Debroy and Moris 2006) have presented them as just another barrier potentially hindering the “economies of repetition” in developing country software industries. At the very least, since these licenses encourage business models based on service provisions, they can hardly help the Indian software industry to become more product oriented (see O'Donnel 2004).

The study by Madanmohan and De (2004) does not say how the cost savings impacted the profit-ratios or market position of the case companies. On a national level, there are subtle signs that FOSS-enabled cost savings can lower entry barriers to mature software markets. For example, local, open source based CMS (Content Management System) providers seem

to be increasing their share of the domestic market, partially at the expense of large vendors such as Oracle and EMC (IDC 2009). However, it seems unlikely that FOSS could create entirely new markets in India by making software more affordable (see also 3.1.1), as some have suggested. In comparison to pirated software, FOSS has the same “purchase price” and often higher maintenance expenses (Sharma and Adkins 2005). The cost of hardware, rather than software, is reportedly a key barrier to ICT access in India (e.g Venkatesh 2000).

3.3.2. Inside-out process: an unlikely route to fame

The literature review (see 3.1.2) paints a rather dark picture on the suitability of the outside-in process to SMEs. It may be even less applicable to Indian SMEs, due to heavier financial constraints (see section 3.2.2) and a smaller pool of volunteer developers available locally (see section 3.2.1). Even the sell-it-free model (see section 3.1.2) seems problematic considering that many Indian software companies don't own the IP that they produce (see section 3.2.2). Perhaps unsurprisingly, company-driven FOSS projects are almost non-existent in India; an international study scanning several sources (UNU-MERIT2007b) was able to find only three FOSS projects which were authored by Indian companies. However, some individuals and companies have, against all odds, launched both technically and commercial successful FOSS projects with very little resources (Wichmann et al. 2002). For example, a Mexican university-drop-out, Miguel Icaza, began to develop the GNOME environment as a pastime, but the project later achieved global fame and he made a fortune with an associated consultation company called Ximian (German 2002). Unfortunately, there is relatively little research on what has enabled exceptional successes of this type.

3.3.3. Coupled process: network building and learning

As discussed earlier, the coupled process can produce both cost savings and promotional benefits. To exemplify the former, ten Brazilian SMEs have joined forces with local universities and international research institutions to develop an open source GIS library called TerraLib (Câmara and Fonseca 2007). The significant cost and time savings have enabled them to enter the Brazilian GIS market, which was previously an exclusive duopoly of multinational companies (ibid). To exemplify the promotional benefits, a Brazilian SME,

Connectiva, became world famous due to their high-quality contributions into the Linux Kernel project (see Kroll 2000). Similar case evidence was not available from India, however. In the Indian context, it would be particularly interesting to know whether the coupled process could boost domestic co-operation (cf. section 3.2.3), for example.

The skills developmental benefits are also interesting from the Indian perspective. Unlike off-shoring parent companies (see section 3.2.2), FOSS communities have a strong incentive to share knowledge over geographical and organizational boundaries (see e.g. Krogh et al. 2003). According to an extensive survey conducted in India (UNU-MERIT 2007c), both individual community members and their employers highly appreciate FOSS as a training system, even over formal education. Out of the twelve social, technical and managerial software skills listed, ten were considered to be better learned through FOSS participation than by taking university courses (ibid). When respondents were asked to estimate their own or employee learning results, significant skills improvement was reported in several areas; interestingly, the very top ones related to domain and market knowledge (ibid).

However, the survey does not tell whether or how the perceived learning had an impact on the innovative capability of the organizations in question. Reflecting country-wide tendencies (see section 3.2.1), only a very small portion of the respondents made any direct contribution to FOSS product development. This re-raises questions on the barriers to FOSS participation (also see section 3.3.1). Since the coupled process requires investments in non-(directly) revenue generating activities, affordability can be a problem (O'Donnel 2004). Or, if cultural and linguistic barriers hinder learning through the participative process (Vaden and Vainio 2005), new entrants may not develop enough expertise to enter the 'inner development circles' (cf. Figure 6 on page 23). Further, as diverse interest conflicts often have an impact on the coupled processes (see section 3.1.3), the “North-South dimension” could add up to the list (ibid). For instance, Western FOSS developers can be reluctant to attend to development or support needs that arise from Southern contexts (ibid; see also Rejswoud and Mulo 2005).

3.4. SUMMARY OF THE LITERATURE REVIEW

The above literature review has described FOSS based innovation processes and discussed their meaning for Indian software SMEs. **Table 2** summarizes the literature review by listing opportunities and challenges associated with each process archetype and identifying concerns that rise specifically from the Indian context. These viewpoints guide the empirical part of the dissertation: specifically, they form the basis for the interview topic guide and provide a starting point for the template analysis. The next chapter will give detailed information on objectives and methods of the case study.

Table 2: Summary of the viewpoints in the literature review (process archetypes from Gasman and Enkel 2004, see section 2.2 and Table 1 on page 19 for an overview)

Process archetype	**Strengths / Opportunities**	**Weaknesses / Challenges**	**Domestic concerns**
Outside-in process	● Savings in development time and costs, leading to ▸ Improved profits margins ▸ Lowered market entry barriers ▸ New customers due to better affordability	● Technical difficulties in large scale code reuse ● Lack of commercial support ● Limitations some FOSS licenses (e.g. GPL) place on business ● Sustainability issues	● "Purchase price parity" with pirated software hinders market creation ● Skills barriers prevent switching to the coupled process?
Inside-out process	● Increased demand for one's own products and services ● Support for strategic goals (e.g. standardization) ● "Sell-it-free-it" model as a tool to cut expenses	● Affordability and financing problems ● Difficult to capture indirect returns, especially for SME's	● Affordability problems get emphasized (?) ● A small pool of developers available locally ● Lacking ownership of the source code
Coupled process	● Cost and time savings as in the outside-in process, but balanced with the expenses of contributing ● Marketing visibility ● Inter-organizational learning by technical "apprenticeship", feedback on innovations and informal socialization	● Difficult to co-ordinate interests between heterogeneous stakeholders ● Technical difficulties as in the outside-in process but eased by participative learning ● Licensing issues largely as in the outside-in process	● Linguistic and cultural barriers can hinder inter-organizational learning ● North-South polarity as a reason for interest conflicts? ● Affordability challenges? ● Opportunities for new domestic collaboration?

4. METHODOLOGY

The broad aims of the research, overall research design and research questions have already been introduced (see sections 1.1 and 1.4). Except for the first subsection, which denotes the philosophical footings of the study, this chapter focuses on the methodology used in the *empirical* part of the research. The second subsection describes the overall design and objectives of the case study. The third and fourth subsections explain, respectively, how the data was collected and how it was analysed. The fifth subsection describes what steps have been taken to ensure reliability and validity. The final section discusses problems in data collection and analysis.

4.1. PHILOSOPHICAL BASIS AND THE ROLE OF THEORY

The ontological and epistemological footing of this study falls into a broad category of perspectives described as critical realism (see e.g. Sayer 1992; Groff 2004). This positions combines *ontological realism*, a belief that some mind-independent reality is 'out there', with *epistemological relativism*, a position that our knowledge of the reality is always conceptually mediated and thus approximate or probabilistic at it's very best (ibid). Following from this, the study aims to describe and explain empirical realities *beyond* the research participants and takes them as a basis for assessing theoretical abstractions. Meanwhile, the author has tried to stay critically aware of how her own tacit assumptions and emotions, as well as those of the informants, interact with the research process[4].

This study can be described as theory-guided or theory-led (Simons 2009), but it tries not to be “theory-blinded”. The study employs both deductive and inductive circles of data collection; with the emphasis being on the former. Deduction[5] from the Open Innovation theory helped to focus the literature review. Deduction from the literature review helped to identify issues of interests and form some preliminary propositions that guided the collection of primary data. On the other hand, induction helped to form explanations from the primary

4 However, much of the associated 'soul searching' remains hidden from the reader because an impersonal (rather than self-reflexive) style of writing has been adopted.

5 The words “deduction” and “induction” are herein used to note the direction of reasoning (e.g. as in Grix 2004), not strictly as in formal logic.

data, which could then be assessed against the literature review and the theory (cf. Yin 2009). The process of combining deductive and inductive elements has been called “retroduction” (ibid) and is often associated with the critical realist paradigm (see e.g. Easton 2010). On the other hand, Ragin (1994) argues that all research is essentially “retroductive”: there is always an interplay between ideas and data.

4.2. JUSTIFICATION FOR THE RESEARCH APPROACH

The decision to select a qualitative case study instead of quantitative methods was based on three issues: the purpose of the study, the nature of the research questions and the (un)availability of prior research (cf. Bryman 2008). The study aims to benefit software entrepreneurs (see section 1.1) and analyse Open innovation at the level of a single company (see section 1.3). This alone seems to support the case study approach, which allows a versatile and detailed description of the studied organizations (see Chetty 1996). In contrast, quantitative methods confine company characteristics into a few variables and, as argued by Chetty (ibid), the lack of contextual information is likely to diminish the value of the findings to individual entrepreneurs. Further, the first research question (see section 1.4) is a ‘how’ question, which calls for understanding operational issues rather than reporting an incidence or frequency. Several authors (e.g. Rowley 2002, Yin 2009) regard case studies as particularly suitable for analysing 'why' and 'how' questions as these are explanatory in nature. The second research question (see section 1.4) could also be approached with a survey, but the lack of prior empirical research on FOSS business in India could cause problems to its design. Many of the viewpoints emerging from literature review (see section 3.4), especially those relating to Indian context, are highly speculative. While they can suggest potential areas of interest, they may not work as a sound basis for pre-determined, closed questions (cf. Bryman 2008). In the absence of statistics on FOSS companies in India, it would also be difficult to get a representative sample.

Qualitative case study approach was chosen in order to allow rich data collection in one or few companies and, thereby, to accumulate in-depth understanding on their FOSS activities and innovation processes. Some other qualitative approaches can also be applied into the

study of one or few organizations (see e.g. Bryman 1989), but case study was seen to have some practical advantages. Unlike action research or most ethnographic methods, it does not necessarily require a long-term commitment (Simons 2009; Bryman 1989) and could thus be adapted to the short time scale available for the dissertation work. Further, case study is one of the chief arenas for mixed methods research (Bryman 2008) and embedding some quantitative methods (e.g. an employee survey or financial score-carding) into this study was considered. However, due to the significant time constraints of both informants and the researcher, it was eventually decided that purely qualitative methods would best serve the purpose of 'maximizing what can be learned in the time available' (Dupé and Paré 2003:609).

4.3. CASE STUDY DESIGN AND OBJECTIVES

As for the case study design, one first had to decide on the number of cases. The purpose of this case study is instrumental, i.e. it is conducted for understanding a wider phenomenon rather than for the intrinsic interest in the case itself (cf. Simons 2009), and, therefore, it did seem ideal to include multiple cases to achieve a maximum variation of perspectives (e.g. Flygberg 2006). One could have selected 3-6 case companies which differ in the value chain position, innovation processes and/or domestic location, for example. However, a single case study was more implementable due to time and resource constraints. Considering that previous empirical research on FOSS business in India is almost non-existent, even a single case study has the potential to contribute worthwhile insights and provide leads for more comprehensive studies in the future (cf. Simons 2009; Flyberg 2006).

The second important decision concerned the criteria for site selection (cf. Stark and Torrance 2005). In line with the research scope and questions outlined previously (see sections 1.3 and 1.4), the case company had to meet with the following selection criteria: be privately owned, classify as an SME, operate in the primary software sector and engage in FOSS projects with a profit-making incentive. It was also decided that the case company should work with several FOSS products and integrate some elements of all of the three processes archetypes (the inside-out, outside-in and coupled process). This enables a more detailed inquiry and comparison of different innovation processes within the same company. **Figure 7** (on the next page) illustrates what has been described as an *embedded case study*

design (Rowley 2002; Yin 2009). The dashed line is used to indicate that the boundary between the case study and context is typically blurry (Yin 2009).

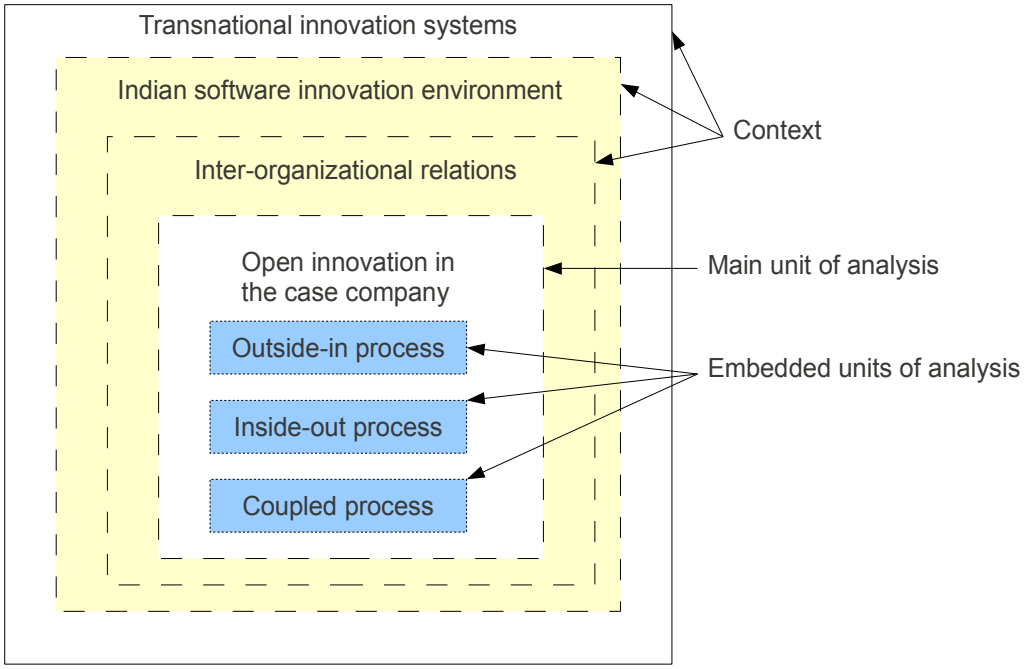

Figure 7: Embedded case study design (adapted from Yin 2009)

The objectives of the case study can be defined as follows. The first objective is to gain a holistic picture on the company's Open innovation processes, especially in relation to FOSS. To avoid becoming overly “blinded” by the predefined process archetypes, a broader networks perspective is used to complement the analysis. The second, and most important, objective of the case study is to collect and analyse the company's recent experiences, both positive and negative, in deploying FOSS based innovation processes. The purpose is to identify the successes, failures, opportunities and challenges that the company has faced when attempting to derive value from FOSS. These experiences are compared to the template emerging from the literature review and the theory (see e.g. Yin 2009; Easton 2010).

4.4. DATA COLLECTION METHODS

Multiple data collection strategies were employed to ensure the richness of data. From the five sources of evidence identified by Yin (2009), this case study will use three: interviews, documentation and non-intrusive observation. Semi-structured interviews were the most important data collection method. The interviewees included three managers with good understanding of the innovation processes and business models of the case company: the current CEO, marketing director and technical director. The interviews with the directors were recorded and selectively transliterated, resulting in approximately 20 pages of transcripts. The interviews also included discussions with developers who communicate with FOSS communities on a daily basis. The two key informants were female, senior developers who had both been with the company for few years. The employee interviews could not be recorded, however, as much notes were written as possible. All of the interviews were conducted in the premises of the case company in Bangalore. The interview topic guides were largely based on viewpoints from the literature review (see Table 2 on page 30). With open-ended questions, the interviewees were also encouraged to bring up (sub)topics of their own. In addition, some elements of the value network analysis (see section 1.4; more details later) were integrated into the interviews of the directors. To get a more unbiased picture on the inter-organizational collaboration involved (cf. Chetty 1996; Easton 2010), some interview questions were also made to representatives of partnering organizations. These included a competing Indian SME, who contributes to the same FOSS projects, and a local NGO, who co-operates with Mahiti Infotech. The partner interviews were also recorded and selectively transliterated. For the summary of the interviewed individuals, see **Appendix B**.

The two other sources of evidence, documentation and observation, were mainly used to collaborate and augment evidence which was collected in the main interviews or, in some cases, to get an initial picture before an interview. The studied documentation consists mostly of websites, videoed presentations and mailing list archives, which relate to the company's FOSS activities. The non-intrusive observation involved following synchronous on-line discussions about FOSS on Internet Relay Chat (IRC) channels, where the case company employees participate. IRC is the primary media for communication in most FOSS projects and the channels are publicly visible. Some field notes were also made while visiting the company premises in Bangalore.

4.5. DATA ANALYSIS METHODS

4.5.1. Value Network Analysis

In Chesbrough's (2003; 2006b) work, inter-organizational networks are not explicitly addressed. However, they are implicitly, and rather unavoidably, present in Open Innovation, because intellectual property (IP) is in-sourced, out-sourced or co-created with external partners. Several authors (e.g. Vanhaverbeke 2006; Maula et al. 2006) have underlined that Open Innovation cannot be fully understood without some kind of networks perspective. To better appreciate how Open Innovation works in the case company, Allee's (2003) Value Network Analysis (VNA) approach is employed herein. Unlike most other methods emerging from social network analysis (see Coulon 2005 for an overview of SNA in innovation research), VNA establishes a clear linkage between network patterns and value creation (Allee 2009). As discussed earlier (see section 2.3), it also provides equal emphasis on monetary and non-monetary exchanges, thus being well-suited to analyzing FOSS business.

The VNA approach is applied to a limited extent only. The focus is on a technique called *value exchange analysis*, which is used to assess the company's external value network. The task is to analyse the overall patterns of innovation and value exchange by using three simple elements: participants, transactions and deliverables (Allee 2003; 2009). These can be defined as follows: (ibid).

- ➔ *Participant* is an individual or a group of people carrying out a role in the value network. For the purposes of this study, the focus is on participants that either contribute to innovative activities or provide revenue directly.
- ➔ *Transaction* is an activity of passing a *deliverable*, i.e. an asset or favour, from one participant to another.
- ➔ *Tangible deliverables* include anything that is contractual or expected by the receipt, as part of a delivery of a product or service which generates revenue directly.
- ➔ *Intangible deliverables* include anything that is non-contractual and/or unpaid, e.g. favours exchanged in social interaction.

The questions asked in the exchange analysis relate to the logic of value flow and patterns of reciprocity in the network; one is also encouraged to pay attention to the comparative proportions of tangible and intangible exchanges (Allee 2009). The concrete result of this analysis is a network topology, which is presented by using a VNA visualization technique called HoloMapping (Allee 2003). **Figure 8** presents symbols used in a HoloMap diagram.

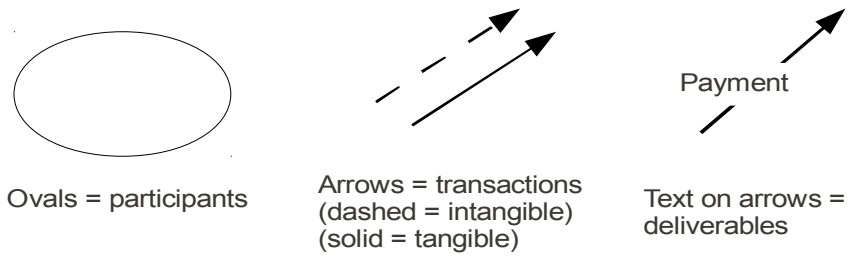

Figure 8: The elements of the HoloMap diagram (Allee 2003)

4.5.2. Template Analysis

While the value exchange analysis helps to get an overall picture on how the case company generates innovations and co-creates value with external partners, another technique was needed to analyse the company's positive and negative experiences on Open innovation processes (as stated in section 4.3). The Template analysis (King 1998) method was selected as a primary analytical technique for this dissertation. It is used for thematically analysing the interview transcripts and, to a much smaller extent, other data such as IRC discussions. It involved the development of a coding "template", which summarised the themes perceived to be important. The template was developed iteratively; the first version was largely based on the summary of the literature review (see Table 2 on page 30). Whilst the analytical process moved forwards, the template was modified to reflect any themes emerging from the data set. The final template served as a basis for the interpretation and illumination of data, as well as the writing-up of findings. The coding template was organized as a two-level hierarchy and parallel coding was allowed. Out of the several analytical techniques considered, Template Analysis was chosen as a reasonable 'compromise' between structure and flexibility. It allowed the viewpoints from the literature review to be integrated into the analytical process, but also provided space for new themes taken up by the informants. As a relatively 'light weight' method (ibid), it also seemed suitable for a small-scale study.

4.6. CREDIBILITY, DEPENDABILITY AND TRANSFERABILITY

The words 'reliability' and 'validity' refer largely to quantitative or experimental research, while this case study only uses qualitative methods. The terminology introduced by Guba (1981) is adopted herein; he suggests the following criteria to qualitative research: credibility (a parallel of internal validity), dependability (a parallel of reliability) and transferability (a parallel of external validity). To ensure *credibility*, the study relies on several different data sources (see section 4.4) and two different analysis methods (see section 4.5); one can talk about data triangulation and methodological triangulatio*n* respectively (e.g. Bryman 2008). Further, the case study report was also shown to key informants for confirmation. This is a form of respondent validation recommended by several authors (e.g. Rowley 2002; Dubé and Paré 2003). To ensure *dependability*, complete records was kept of the collected raw data so that other researchers or students have a possibility to check them; some refer this to as keeping a case study database (e.g. Rowley 2002; Yin 2009). For this study, this means having a memory stick containing interview recordings, transcripts of interviews and IRC discussions, field notes, and a list of websites and mailing list archives visited. As for *transferability,* the results from a single case study are obviously not generalisable to other situations. However, when analysed against existing research findings, they can contribute to the understanding of the target phenomena (see e.g. Yin 2009; Easton 2010). Qualitative researchers are also encouraged to provide rich contextual description, as this arguably helps others to make judgements on the transferability of findings to other settings (Bryman 2008). The VNA exchange analysis (see section 4.5.2) also plays a role herein: by giving a broader picture on how the case company creates value for itself, it may help assessing the applicability of their experiences for other Indian SME's. Disclosing the name of the case company[6] also provides access to additional contextual information (Liebenau and Smithsons 1993).

6 Researchers have argued both for and against disclosing the organization's name in case studies (see e.g. Simons 2009), but this is not debated herein. The company directors were given the choice and they selected recognition over anonymity.

4.7. METHODOLOGICAL CHALLENGES

The author of this dissertation did not have access to company's administrative or classified documentation; most of the studied documents are visible on the Internet. The interviewed directors were also likely to consider how their comments are going to affect the image of the company. Thus, there is a risk that the case study findings reflect how the company wants to presents itself, rather than how things are in reality. However, it must be stated that many of the same questions were made to each interviewee separately and an emerging consensus[7] suggests some degree of credibility. Partner interviewees, third-party websites and mailing list achievers also gave an opportunity to confirm certain facts from a non-company source. Nevertheless, due to the limited availability of other data, the study relies a bit too heavily on the interviews. When a given information could not be confirmed from another source, this is indicated with statements such as “directors said”.

The author of this dissertation lacked any previous experience in qualitative interviewing, which limited the amount of primary data collected in the field. A large amount of serious mistakes, e.g. failing to ask an obvious follow-up question, were easy to notice when listening to the recordings. The interviewer's inexperience may have been partially compensated for by the positive attitude of the company directors. They had 'googled' her name in advance and read about some of her voluntary FOSS activities. This clearly helped to build up a rapport: the interviewer felt welcomed as a fellow FOSS contributor, rather than as a suspicious outsider. This was in contrast with some partner interviews where the atmosphere was much more reserved, especially at the beginning. In turn, the interviewees of the developers were constrained by their limited knowledge of English. Observing non-verbal communication became important herein and statements such as “developers seemed genuinely enthusiastic...” reflect this in the results report. Especially considering cultural differences, such interpretations leave a lot of space for misunderstandings. Further, as most material was collected in the interviews of the directors, their views may dominate over those of other stakeholders.

7 This is to say that relevant factual accounts were remarkably consistent, not that interviewees answered similarly. They often approached the questions from very different perspectives, which helped to get a more wholesome picture.

As to the challenges related to data analysis, one key issue concerned depth versus coverage. Stark and Torrance (2005) point out that, within the logic of the case study approach, the emphasis should always be on the former. However, considering the wide range of theoretical viewpoints emerging from the literature review, none of these could be discussed in great detail. The variety of viewpoints also caused the initial version of the coding template to be relatively complex, which involves a danger of not staying open to issues emerging from the data (cf. King 1998). Due to presentational limitations, very difficult choices had to be made concerning which issues were relevant enough to be presented in the results section and in how much detail. In hindsight, focusing on only one aspect of Open Innovation, such as open co-creation in the coupled process, would have brought more depth to the study. As to thematic analysis, a further challenge was caused by certain themes, which were clearly very important to participants, but had been deliberately excluded from the research (cf. King 1998). The social implications of FOSS can be mentioned as an example (see section 1.3 for research scope)

5. CASE STUDY RESULTS

The previous chapter stated the objectives of the case study and described the data collection and analysis methods employed. This chapter reports the actual case study results. After providing some background information on the company, the first section presents the results of the value exchange analysis (see section 4.5.1), describing how the company co-creates innovations and value with external partners. Based on the thematic analysis (see section 4.5.2), the second section reports the company's experiences on FOSS based innovation processes. The final section summarizes the results of the case study.

5.1. THE CASE COMPANY AND IT'S APPROACH TO INNOVATION

5.1.1. A brief introduction to the case company

The case company, Mahiti Infotech Private Limited (Mahiti in short), is head-quartered in Bangalore, employs 70-90 people and has reportedly been soundly profitable for the past eight years. The company bases practically all of its offerings on free and open source software, adding value to FOSS products, mostly by customizing them to the needs of individual clients. The tailored end-products go to market either as bespoke software or though the ASP model. Technical consulting provides some additional revenue streams. The coupled process can be seen as Mahiti's primary approach to Open innovation. The company has strategic partnerships with several FOSS development communities and related advocacy groups. For example, they have an important role in the development of the Plone content management system. Mahiti also incorporates elements form the outside-in and inside-out processes. In order to save R&D costs and time, they intensively encourage the use of FOSS components in all of their software projects. They are also one of the very few Indian companies (see section 3.3.2) authoring their own FOSS “spin-outs”.

5.1.2. Patterns of innovation and value exchange

The value exchange map of the case company is presented by **Figure 9** (on the next page), please see Figure 8 (on page 37) for an explanation of symbols. The dominance of intangible exchanges seems to be the most striking characteristic of the network. As discussed earlier (see section 2.3), innovation and value creation in FOSS companies is largely based on barter exchanges, where reciprocity is expected but explicit contractual agreements are absent. This could explain why mutual trust and honesty were very highly valued and were the most frequently recurring themes in the interviews. One the directors described the centrality of 'trust' to their business model as follows:

> Many of our customers have never left us. This is not because they could not find somebody cheaper than us and not because they could not get somebody better than us. It is because they cannot find someone who is as transparent, as honest and as committed as we are.... They stay with us because they really trust us... (Sreekanth, see Appendix B)

While the trust and openness between customers and the company was emphasized the most, similar values were seen central to other exchanges, too. Interestingly, this small observation seems to be in line with Allee (2003), who theorizes about the growing importance of intangible exchanges and a consequent need for stronger 'ethical integrity' to ensure trust in modern business networks.

The entire logic of value flow is not verbally explained herein, but the following characteristics are particularly noteworthy. The company mostly works with direct end-clients; these can be divided into commercial and social sector customers. The commercial customers are either companies or large non-profits who can afford to pay market prices. Social sector customers are small, Southern non-profits who only pay a nominal fee. The rest of the expenses are subsidized by development institutions, such as EuropeAid, or, by the company itself. In the later case, the social sector customer would often participate in the design and beta-testing of new products which can be later sold to commercial customers. The map also shows so called “co-opetitors”[8], i.e. fellow software companies who contribute to the same FOSS projects, but compete on customer projects which are based on the co-created products. The 'co-opetiting' companies engage in little direct interaction. Most co-operation is intermediated by FOSS communities, yet they reportedly adhere to some

8 The term 'co-opetion' is used by Allee(2003), but was apparently first coined by Cherington (1913:144).

colleges plays an important role in recruiting volunteer developers to the company's own FOSS spin-out projects. In addition, free lecturing in a collages is a way to bring FOSS technologies into the curricula from the “back door”.

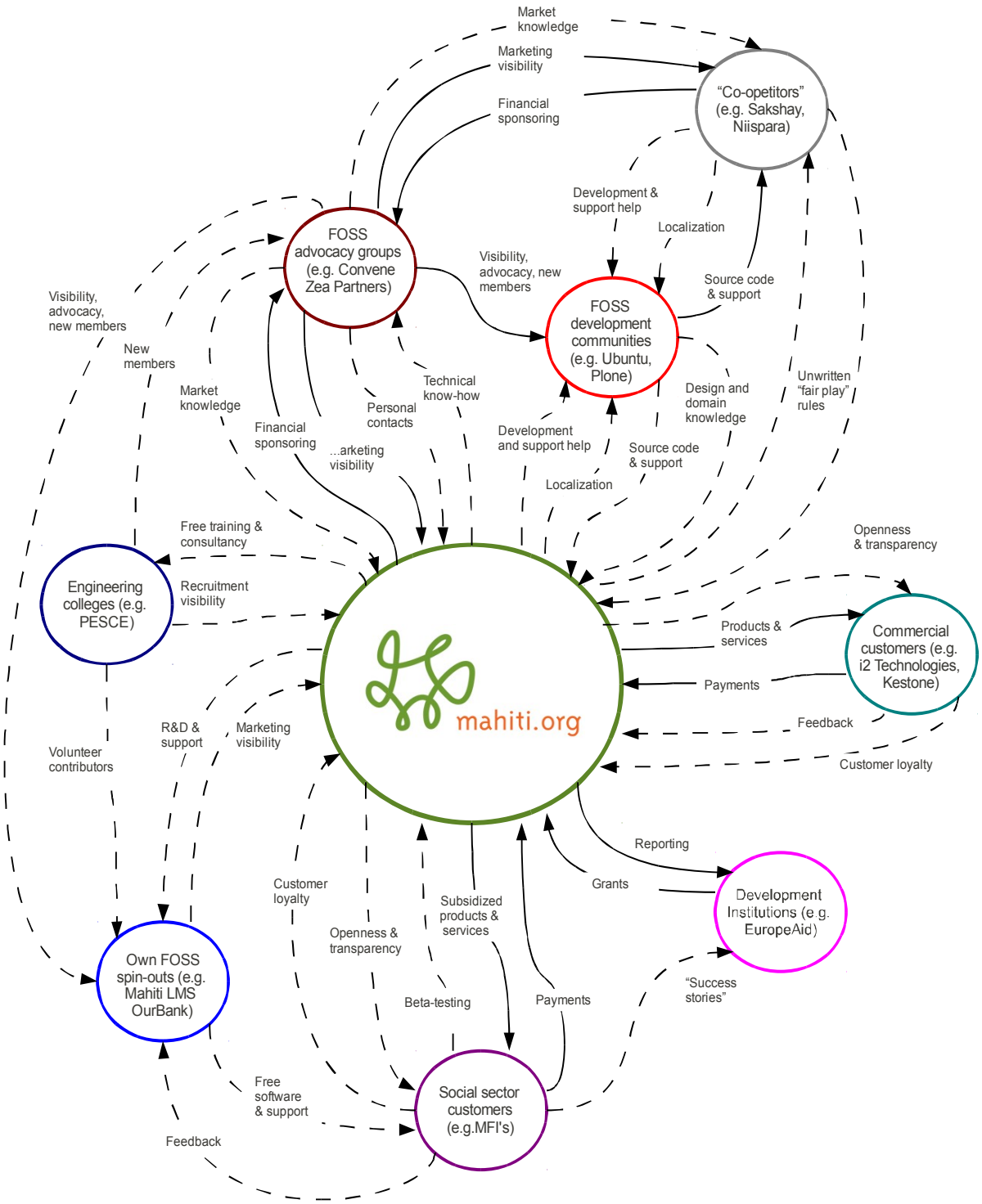

Figure 9: External value network of the case company

From the three Open innovation processes, the inside-out process and coupled process are explicitly visible in the value exchange map. The company collaborates with FOSS communities and exchanges assets with them, roughly in line with what was presented previously in connection with the coupled process (see section 3.1.3). This collaboration is supported by advocacy groups, who work to disseminate the innovations created by FOSS development communities. The pattern of value exchanges associated with the FOSS spin-outs also seems characteristic to the inside-out process (see 3.3.2). As for the outside-in process, the analysis revealed that Mahiti does not really deploy it in its archetype form. While the interviewees mentioned dozens of FOSS projects which are only 'exploited' without contributing back, the company was always found to be a contributor in one or more closely related communities. Thus, the outside-in process is not clearly distinct from the coupled process.

5.2. EXPERIENCES ON FOSS-BASED INNOVATION

This section expands on some issues which are visible in the above value network map (Figure 9 on page 43), reporting the case company's experiences with Open innovation. One subsection is dedicated to each process archetype. The 'outside-in subsection' reports experiences on FOSS reuse. These also apply to the coupled process as far as inbound IP flows are concerned (see section 3.3.3). The 'inside-out subsection' discusses the company's experiences with FOSS spin-outs. The 'coupled process subsection' reports experiences on FOSS participation. The focus is on inter-organizational collaboration, an aspect which differentiates the coupled process from the two other process archetypes. Table 2 (on page 30) may help in following the storyline of this section as it has greatly influenced the thematic template used in the analysis (as explained in section 4.5.2). The reader is also encouraged to keep the underlying theoretical concepts (see especially chapter 2 and Table 1 on page 19) in mind.

5.2.1. Outside-in process: experiences on FOSS reuse

One of the Mahiti founders gauged that savings from software licensing fees alone could make a FOSS company 15 % more profitable. According to his “rule of thumb estimate”, an average Indian software SME was spending one dollar on licensing fees for each eight dollars earned. In practice, the profitability implications of FOSS reuse varied greatly from one customer project to another. **Figure 10** illustrates two recent projects. In the first case, the company only needed to make minor modifications to an existing FOSS platform, but was still able to charge a “premium price”, higher than that of all proprietary software vendors participating in the bid. This is because the FOSS product met well with the needs of the customer as such and Mahiti could offer the fastest lead-time. Consequently, the profit margin was very high. In the second case, the company had to build almost half of the source code by itself before customer requirements were met. Despite this, the customer could only pay a relative low price and the project was not profitable alone. However, it was still worth doing because an extension developed for this customer was expected to be resold to several other customers over time. It is noteworthy that FOSS licensing terms do not hinder the “economies of repetition” (cf. section 3.3.1) of this type. The same piece of code, if only generic enough to be useful to many customers, could be reused in bespoke software projects for an unlimited number of times.

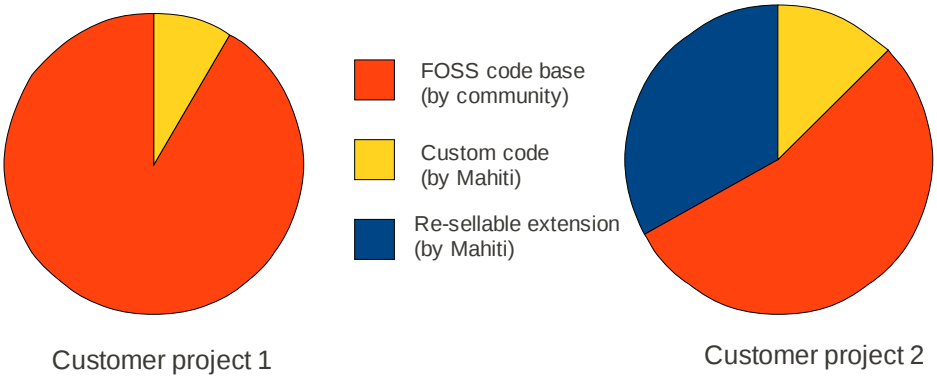

Figure 10: Proportions of FOSS and 'own' code in two software projects

The interviewees viewed proprietary software licensing, rather than GPL or other FOSS licenses, as restrictive and problematic. They said that proprietary licenses disallow repackaging, for example. In comparison, FOSS licensing terms place no restrictions on the current revenue models of the case company (see section 5.1.1 for revenue models, Appendix A for licensing issues). The company is not interested in entering the packaged product market, where GPL licensing would have major implications (as explained in section 3.1.1). In line with what was discussed in section 3.2, the software product market was seen as saturated and requiring very large investments: “basically the curve to recover the funds is very high and this kind of model is not viable for a company like ours”. It was underlined that a software company with their resources could not avoid sourcing external technology. Compared to in-licensing proprietary technology, FOSS comes at a lower cost and with unlimited customization options, allowing to add more value in-house. After the interviewer had insisted on FOSS-related “profit-making limitations” for a while, one of the company founders replied:

> Ok, you are very much right... if you choose to build your product with open source, you will most probably not become Bill Gates or Steve Jobs. But this is like any career choice, well, you can become a mortgage banker or a broker. […] Microsoft is what it is today because they have spent money on every single line of code that they have wrote. I cannot do that, start from scratch and build an operating system. I cannot achieve anything like that unless I do it with open source. And when I benefit from the efforts of others, I cannot expect to keep all of the profits alone. (Sreekanth, see Appendix B)

The source codes were given to customers, even when FOSS licensing terms did not require it. The directors were not worried of the fact that source code access would enable their customers to buy maintenance services from another software company or, in some cases, even sell the software to a third-party. Instead, they were proud to explain that their customers came back, not because of any vendor lock-in, but because of satisfaction with the work done. They also saw it as very unlikely that customers would resell the source code; most of them did not even operate in the ICT field. Mutual trust between the customers and the company was again strongly emphasized, taking us back to the dominant discourse on FOSS business being based on 'transparency, openness and trust' (see section 5.1.2).

The interviewees were also convinced that, through increased affordability, FOSS can make custom software available to new consumer groups. In line with previously presented views

(see section 3.3.1), they admitted that FOSS alternatives have a hard time competing with pirated copies of mass products, such as Microsoft Excel. However, a large number of under-resourced organizations, e.g. domestic non-profits and small companies, were said to need customized software. “If you are an NGO, you can always get some old computers [as donations] from industry, the real challenge is to get software that is suitable for your needs”, said one interviewee (Vijay, see Appendix B). While pirated mass products can be customized to a certain extent (e.g. Microsoft Excel with macros), such possibilities are limited. It was thus argued that FOSS based, low-cost solutions can increase the size of the bespoke software market in developing countries. Unlike most Indian software SMEs (see section 3.2.2), Mahiti has highly diversified export markets with customers in countries such as Mongolia, the Bahamas, Brazil and Tajikistan. They believed that this is partially because FOSS based solutions are more affordable to Southern customizers. They also reported that customer's receptiveness to FOSS solutions in India and some other developing countries had increased rapidly over the past five years. However, it still seems that the company's co-operation with development agencies is probably the major explanation for the geographical market diversity.

As for technical difficulties (cf. sections 3.1.1 and 3.3.1), it was mentioned that local education systems have traditionally ignored FOSS technologies, but this was said to be slowly changing. However, while technical expertise on certain FOSS products was making its way into the engineering curricula, code reuse skills were reportedly not. Most new employees were said to be unfamiliar with the whole idea of code reuse and had to be taught by hand. The complete “blind reuse” described by Madanmohan and De (2004; see section 3.3.1) was discouraged. The medicine used to address QA challenges with code reuse was “more reuse”. For example, a developer would wrap a FOSS component into an interface module which had already been tested in an earlier customer project and could thereby avoid method calls containing 'surprise bugs'. Mahiti had even built an Intranet solution where the developers can search for FOSS component interfaces which were developed in-house. Further, co-development with social sector customers (see also 5.1.2) plays an important role in improving the stability and usability of solutions over time.

The employees reported very positive experiences of community support, yet some explicitly attributed this to the fact that the company did not operate on the outside-in mode.

“Our support requests are always attended to as the first priority, because they know that we are also contributors”, said one (Hasita, see Appendix B). The directors believed that outside-in process alone would be a very unsustainable basis for business. FOSS communities were seen as forming a large 'ecosystem', which contributors kept alive and running. “If you are just a consumer, then the ecosystem will sooner of later die”, said one (Sreekanth, see Appendix B).

5.2.2. Inside-out process: experiences on FOSS spin-outs and “spin-offs”

One of the FOSS projects initiated and co-ordinated by Mahiti is OurBank, a community developing software for micro-finance institutions (MFI's). The project has attracted almost 100 volunteer developers, mostly engineering students, and, in addition, MFI's from as far away as Brazil have contributed localization efforts (cf. Figure 9 page 43). Following successes, the directors are convinced that authoring one’s own FOSS community is feasible for an SME. FOSS community building was perceived as mostly requiring “energy and passion”, while proprietary software development was seen as requiring (overly) large monetary investments. This partially contradicts with the literature review (see sections 3.1.2 and 3.3.2), but it was admitted that capturing returns can be difficult. The CEO explains:

> Creating a product, architecting it, developing it, convincing other people that it is good and building a community - it is a painful thing to do, but it is sustainable in the long run. However, it does not provide you with returns like these [FOSS customization projects]. (Sreekanth, see Appendix B)

These profitability challenges exist despite some institutional grants towards the development of OurBank. Besides, while all stakeholders share OurBank software under a FOSS license, its source code has not yet been put on-line. This could relate to what an interviewee called “open source piracy”: another company could appropriate the source code but illegally ignore the reciprocity terms of the FOSS license. This type of piracy was said to be common and can even destroy a new FOSS community (cf. Dahlander and Mangnusson 2005). So, there are clearly a large amount of difficulties to the inside-out process of this type. However, the company sees it as a temporary phase: the plan is to move to the coupled process in the long run. Interviewees saw FOSS spin-outs as a task of taking the community to a certain level and then waiting for others to take it further. They believed that a new

community cannot start “naturally evolving” unless the initiating organization gives away its leadership position at some point.

In addition to the full-fledged FOSS spin-outs, the case company also does what could be called “spin-offs”. These resemble the “sell-it-free-it-model” (see section 3.1.2), the software is open sourced as at a late stage of its life cycle, once it has stopped creating value commercially. However, nothing is invested in community building or even making a website, and thus, the cost of open sourcing is practically zero. Basically, it is just a practice of regularly making 'surplus' source code freely available on SourceForge or similar FOSS platform. As discussed previously (see section 3.1.2), such an activity would be regarded as useless or even harmful by most business researchers. However, the case company has very different experiences; they have obtained concrete and significant benefits. For example, the company open sourced a small business application, a Leave Management System, which was only meant to be used in-house. Later on, they were contacted by a large foundation, who had downloaded the software and wanted to have it extended. The company gained a very important customer in this way, but the marketing effort only consisted of a few mouse clicks. Knowing this goes against conventional wisdom, the marketing director said with a smile: “we have also been very surprised... but if we can get something so great [business benefits], then, something must be right” (Chethan, see Appendix B). However, there are also limitations: sometimes, the company does not own the IP created in the customer projects (cf. section 3.2.2; also more below) and thus, cannot make it open source.

5.2.3. Coupled process: experiences on FOSS participation

When asked about business benefits from contributing to FOSS projects, the global marketing benefits were emphasized and typically mentioned first. The company does not need to engage in conventional marketing, interviewees explained, because “FOSS gives us complete visibility”. Mahiti's partnership with the Plone content management system provides a good example. The company is listed as a key partner on the Plone website, which was said to bring in a large amount of customers alone. They also co-organize Plone conferences and, consequently, Mahiti is mentioned on dozens of blog spaces reporting on these events. Further, employees' code contributions and postings to Plone mailing lists are also made under the company name. This all results in excellent Internet visibility, especially

in regard to search engines[9]. However, interviewees explain that it is not only about having one's name visible, but also about being seen as “shapers” of the technology:

> people see that we are contributors and that Sree [CEO] is the member of the Plone Foundation and say 'hey, these are the people who vision the [Plone] software, it would be better working with them than with somebody who only has expertise (Chethan, see Appendix B).

FOSS communities are also specialized social networks, where “word-of-mouth” moves quickly. Recommendations by other Plone community members played an important role in getting new customers. While most of these conversations presumably happen in private, there was even evidence of this on public discussion forums. For example, a UK-based FOSS contributor recommends Mahiti to another organization on a Plone mailing list, saying “I've been told Mahiti has very good Plone/Zope skills and also knows the server side”.

As expected, another group of reported benefits relates to inter-organizational learning. The training of new employees is closely integrated into the participative learning methods of FOSS communities. In accordance with the aforementioned “onion model” (see Figure 6 on page 23), the new employees are first taught to use IRC and given time to follow conversations on FOSS development forums. They are then instructed to answer as many support questions as they can. Later on, they are encouraged to fix bugs or make small code contributions of their own; a specific time slot is devoted for this every week. If the contribution is not accepted, they should reflect on why. From the management viewpoint, the training method is cost effective, as new employees are coached by external experts, free-of-charge. However, FOSS project administrators were said to have limited time with 'newbies', for example, they are sometimes too busy to explain why a certain code contribution was rejected. Despite this, the interviewed employees seemed to be genuinely enthusiastic about this training method. One said that, while engineering education had mostly taught her to complete specific tasks, FOSS participation had taught her to find solutions independently.

Since the interviewed developers were not fluent in English, it seemed logical to ask about any language barriers to FOSS participation (see section 3.3.3). The question was received

9 For example, at the time of writing, a Google search with the key words “Plone” and Mahiti” brings almost 2000 entries and, with search words “Plone” and “India”, Mahiti appears at the top of the list.

with a friendly laugh; they explained that only very basic English language skills are needed on FOSS forums. “You only need to know technical words and then some IRC slang”, said one. This could be better understood after spending some time on the IRC channels in question. The majority of dialogue consists of excerpts from source code, configuration files or error logs, which are shared in the hope of feedback. The feedback is typically given in highly abbreviated technical language such as “lsof %PATH_TO_SOCKET 2ver” or ”edit selected page portion!mode raven”[10]. Even 'social niceties' are often replaced with abbreviations such as YNWIM for “you know what I mean” and YWVM for “you're welcome, very much”. The author of this study felt that she was facing an unfamiliar (sub)culture and, to some extent, a new language on the IRC challenges. The question of “linguistic and cultural barriers” started seeming more complicated that the literature review had suggested.

As for other forms of inter-organizational learning, the beta-testing and co-development with social sector clients (see also section 5.1.2) presents a noteworthy example of domestic co-operation. The interviewees saw it as an important continuum from FOSS development practices, which have always emphasized end-user participation in R&D (cf. section 3.1.3). A director explains the benefits:

> when I am giving the software to them [certain customers] at a low cost, I can do some amount of beta-testing on them, some R&D on them, which I do not get to do on the market...and the customer is happy because they are getting more than they would expect at a nominal cost... by pricing it low and reusing it, we bring in stable code (Vijay, see Appendix B)

Interestingly, this did not only help in developing solutions for commercial sector clients, but was also seen as important for making better FOSS contributions: “You cannot release something [to a FOSS community] and expect miracles, unless you have tested it with a customer” (Vijay, see Appendix B).

The interviewees had not experienced or, perhaps more likely, were not willing to share any specific interesting conflicts with other FOSS community members. Some large companies were said to 'prey' on FOSS communities, in order to find talented developers for recruitment, however, these were seen as outsiders rather than community members. The

10 In the first stament, “lfos” means 'list open files’; % refers to variable and, apparently, 2ver means simply “to verify”. The second expression is not understandable to the author.

latter were said to act differently (cf. Section 5.1.2):

> when there is another company in the [same FOSS] community in India, I would know the CEO and we are sharing something, you can call it a brotherhood or whatever... so if somebody from my organization applies there, they call me and ask 'hey this and this person has filled in the application for me, is it okay for me to hire him'. I may give them permission or I may say 'please wait for four months and then I will release him to you. (Chethan, see Appendix B)

It was also mentioned that there are practically no FOSS applications addressing Southern development needs, such as monitoring the quality of water or coordinating rural health-care. “All of these are possible with FOSS, but there are very few projects moving despite huge demands”, the technical director (Vijay, see Appendix B) said. This could be understood as a direct consequence of most FOSS communities having been born in the global North (see sections 3.2.1 and 3.3.3). However, as with relationships with existing communities, the employees seemed to feel sincerely proud of being well-recognized and respected members. For example, they very positively recalled that Joel Burton, a somewhat 'legendary' founder of the Plone Foundation, had visited Mahiti Infotech and socialized with them. This was understood to be evidence that their contributions are highly appreciated. “If we worked with Microsoft, Bill Gates would not come to chat with us”, compared one (Girija, see Appendix B)

The affordability of making one’s own FOSS contributions was also discussed; Mahiti has found several ways to keep expenses reasonable. Contributing to FOSS is integrated into employee training as explained earlier. The company also intermediates contributions made by others, for example, it helped local Myanmarian refugees translate Plone into Burmese and put their contribution online. Or, as in the “sell-it-free-it model” discussed in the context of spin-outs (see section 3.1.2), a certain module had often been sold to one or more customers before being contributed to a FOSS community. The case company seems to be in an exceptionally good position herein as it primarily works with direct end clients, almost half of whom are non-profit organisations. Such customers would often not mind a contractual clause allowing the source code to be open sourced later, interviewees told. In line with earlier discussions (see section 3.3.2), they said that when occasionally subcontracting for IT companies, they also lose ownership of the source code and cannot release it as open source.

5.3. A SUMMARY OF THE CASE STUDY RESULTS

This chapter has described how the case company derives value from FOSS. The coupled process holds the key role: the business is based on extending and customizing base products, which are developed collaboratively with FOSS communities. The 'Outside-in' process was applied to a very limited extent and was regarded as being unsustainable alone. The inside-out process seemed to be challenging business-wise, except for low-cost “spin-offs”. Full-fledged FOSS spin-outs require extensive R&D and marketing effort, but do not bring revenues from licensing fees. While the spin-offs brought notable benefits, they provide a complementary marketing strategy rather than a basis for business.

Table 3 (on the next page) summarizes the opportunities and challenges associated with FOSS based innovation processes in the case company. The reader is encouraged to compare them with the summary of the literature review (see Table 2 on page 30). As previously, issues reported for the outside-in process also apply to the coupled process, unless otherwise stated. In connection with some challenges, the specific solutions employed by the case company are also briefly mentioned. The next chapter concludes the dissertation by picking-up the most significant points from Table 3 and discussing their wider implications.

Table 3: The case company's experiences on FOSS based innovation processes (process archetypes from Gasman and Enkel 2004, see section 2.2 and Table 1 on page 19)

Process	**Strengths / Opportunities**	**Weaknesses / Challenges (+ Solutions)**
Outside-in process	● Savings mostly in licensing fees as FOSS is an alternative to in-sourcing proprietary technology ● The savings lead to improved profit margins and increased affordability to customers. The latter can can grow bespoke software markets in the South. ● Unlimited customization options due to source code access ● Reportedly good quality support by FOSS communities ● Customer receptiveness to FOSS has improved fast	● Additional training expenses, as FOSS technologies and general code reuse skills are often missing from engineering curricula; own co-operation with colleges aims to address this ● Some quality issues such as bugs in foreign code, but these are well-addressed by incremental product development and intensive reuse of own interfaces ● Basing business on outside-in process alone is seen as unsustainable (the FOSS "ecosystem" dies without contributors)
Inside-out process Full-fledged spin-outs	● Supported by grants from development institutions ● Creates a demand for customization and consultation services	● Takes a lot of effort and capturing returns is difficult, leading to profitability challenges ● "Open source piracy" is common and can cause significant damage ● Most volunteers are students and thereby relatively inexperienced
Low-cost "spin-offs"	● Very inexpensive and reportedly effective marketing	● The company must hold all the rights to the source code or ask permission from the original customer
Coupled process* *see outside-process for issues with inbound IP flows	● Significant marketing benefits relate to (a) search engine visibility (b) an image as a "shaper" of the technology and (c) word-of-mouth marketing ● Inter-organization learning in FOSS perceived beneficial (focused on technical expertise, most feedback on innovations comes from customers directly) ● Employees proud to be respected members of FOSS communities ● Fellow FOSS companies refrain from harming 'co-opetitors'	● Affordability is recognized as an issue, but effectively addressed by (a) integrating contributing to employee training (b) mediating the contributions of others (c) selling the module to one or more customers first ● FOSS project administrators have only a limited time to instruct 'newbies' ● Some large companies use FOSS communities for recruitment purposes ● A vast majority of FOSS projects originate from the North and do not target local development needs

6. DISCUSSION AND CONCLUSIONS

This chapter concludes the dissertation. The first section draws conclusions based on the case study results summarized above (see Table 3 on the preceding page), while the second section acknowledges some of the many problems and limitations of the study (see also section 4.7). The final section will discuss the implications of the study for further research, first summarizing its modest contributions to Open Innovation theory and then providing some pointers for those wishing to conduct further research on FOSS business in India.

6.1. CONCLUSION ON THE RESEARCH RESULTS

The case study has shown one feasible model for how Indian SMEs may benefit from FOSS: by developing FOSS products collaboratively with communities and then adding value through customization. For the case company, the extensive reuse of own code emerged as an important success factor. Instead of planning customization projects on a one-time basis only, the company develops reconfigurable extensions, which are reused in several customer projects and thereby, developed iteratively. This increases profitability by the 'economies of repetition' and helps to bring in more stable solutions over time. This development of FOSS-based 'semi-packages' had been adopted as an alternative to project-based development with proprietary technologies. In comparison to the latter, it seemed to offer more opportunities for both profit-making and learning for innovation. Developing of own packaged software products (which may present most lucrative innovation opportunities) remains an unrealistic option for many Indian SMEs due to financial and other constraints.

The case study illustrates how FOSS can blur the boundary between the software vendor and third-party service provider, thus, levelling the associated value chain hierarchy. Source code access enables the case company to modify the software freely, without being confined to vendor-determined customization options. Further, participation in FOSS development has helped it to accumulate “vendor-like” in-depth expertise and build an image as a 'visionary' and co-creator of certain FOSS technologies. Thereby, the case company can add more value to FOSS products than a non-vendor can typically add to proprietary software products. In the absence of licensing fees, it can also retain a larger portion of the revenues from the

value-added. The example of the case company suggests that FOSS customization can be a viable option for some third-party service providers, who wish to upgrade in the value chain, but lack the resources to develop one's own software products from 'scratch'. The marketing benefits of FOSS participation, especially as to international visibility, were also strongly emphasized in the case study. This finding may interest those SMEs who want to find more direct customers at the long distance. The study also hints that, by making bespoke software more affordable, FOSS may open up opportunities for expanding markets domestically or to other Southern countries. This is presuming that customers are sufficiently receptive to FOSS solutions, which is, according to the experience of the case company, increasingly the case in those Southern markets where they operate.

As to challenges, the study demonstrates that it can be very difficult for Indian SMEs to profit from launching their own FOSS communities. Unless supported by grants from international donors or governments, their options are most likely limited to incrementing existing FOSS products. These mostly originate from the global North and are unlikely to address regional needs as well as locally created software could. Another challenge lies with domestic engineering education, which, despite recent positive developments, is reportedly failing to address the skills needs of FOSS companies. Consequently, new recruits are typically unfamiliar with FOSS technologies or development practices, causing additional training expenses. Further, the study hints that FOSS participation could be particularly difficult for those SMEs which serve large IT companies rather than direct customers. If they loose ownership of the developed source code, they do not have much to release as open source. As the case study suggests, making a good quality FOSS contribution can be unaffordable and also technically difficult unless the code is first sold and tested with an end-customer. While it is possible to exploit FOSS assets without participating in their production, this results in sustainability problems and lost opportunities for inter-organization learning.

6.2. PROBLEMS AND LIMITATIONS OF THE STUDY

Several problems and limitations arise from the single case study design, which was selected for pragmatic, rather than methodological/theoretical reasons (see section 4.3). With

reasonable caution, its findings may be transferable to other SME-sized *bespoke software companies with direct end-clients*. However, the case study says very little on how some other software SME's, e.g. subcontractors or mass software producers, could benefit from FOSS. Opportunities and challenges are likely to be very different for them. For example, if subcontractors only use T&M pricing (see section 3.2.2), FOSS-enabled cost and time savings presumably become insignificant or undesirable. Or, for packaged software producers, licensing terms are important (see section 3.1.1) and the growing popularity of more proprietary-friendly (i.e**.** non-GPL) FOSS licenses may open up new opportunities. Yet, the case study on Mahiti Infotech can shed little or no light on these issues. With hindsight, it may have been wise to narrow down the scope of the research after realizing that only one case study could be conducted.

Further problems arose from analysing the inside-out, outside-in and coupled processes separately *within* the same company. Gasman and Enkel (2004) originally used these concepts to *classify companies* according to their primary approach to Open innovation, *not* to categorize innovation processes inside a single organization. Although the latter application is not unprecedented, it brought quite a few challenges. Firstly, the outside-in process and the coupled process were not clearly separable from each other in the case company. Secondly, even if they had been clearly distinct, some overlap would have been unavoidable: the benefits and challenges associated with the outside-in and coupled process are quite similar as far as inbound IP flows are concerned. In turn, many characteristics of outbound IP flows apply to the inside-out and coupled process alike. For these and some other reasons, the three process archetypes were not ideal analytical categories for a single case study. To provide an alternative, it might have been better to identify opportunities and challenges in relation to inbound IP flows, outbound IP flows and bidirectional IP flows (derived from Chesbrough 2006b) instead. Besides, Mahiti is clearly a 'coupled process company' and so it's hardly surprising that the two process archetypes were pictured as troublesome. If the study had involved a company which *primarily* relies on the inside-out or outside-in process, contracting views would have most probably emerged.

6.3. IMPLICATIONS FOR FURTHER RESEARCH

6.3.1. Reflections on the Open innovation theory

The case study has exemplified that Open Innovation, even when commercially-motivated, can be based primarily on non-monetary exchanges. Thereby, it questions the common assumption (see section 2.3) that Open Innovation can *only* work through monetary exchanges and that it *necessarily* requires a legal environment where IP rights can be easily traded for money. The Allee's (2003) techniques for analysing barter exchanges, some of which were introduced in this study, could thus prove very useful for understanding Open Innovation. They appear to have significance beyond FOSS: one can easily name other innovation systems/networks (e.g. most industrial research consortia and professional learning communities) which operate on the basis of non-monetary exchanges. Arguably, equating Open innovation with IP trading represents a very limited view of innovative processes.

On the other hand, the study hints (with reservations discussed in section 6.2) that both the inside-out process and outside-in process are problematic in the context of commercial FOSS. The outside-in company presumes that external IP from FOSS communities keeps flowing in for free, but does nothing or little to motivate its creation. This can work for some time, but claiming long-term sustainability is difficult. The inside-out process involves a more subtle paradox. Companies typically aim to benefit by controlling the new FOSS community, but, meanwhile, third-party developers join to advance their own goals, not those of the originating company. The interviewees 'hit the nail' by arguing that an inside-out company must eventually give away its leadership in order to move towards balanced collaboration, i.e. the coupled process. Based on the study, one could hypothesize that FOSS development can only work through the coupled process in the long run and that the other two processes only work as temporary phases. This is an interesting speculation, but additional research is needed to make a founded argument.

The case company's habit of uploading 'surplus' source code to the Internet is also interesting from the viewpoint of Open Innovation theory. The practice is very much in line with one of the Chesbrough's (2003) main “ethos”: one should never 'sit' on the surplus intellectual

property. However, the channels that Chesbrough recommends for renting or selling the unused IP (e.g. through the patent system or own spin-off ventures) can be rather costly for an SME. As discussed previously, FOSS spin-outs are also regarded to be expensive. However, the case study suggests that SourgeForge-like platforms provide a low-cost route for releasing 'surplus' IP and that surprising benefits can emerge once the IP receives 'a new life' in the FOSS domain. This provides an interesting pointer for further research.

6.3.2. Pointers for further research on FOSS business in India

The dissertation has scanned several issues which have an impact on the ability of Indian SMEs to benefit from FOSS and none of these could be discussed in great depth (see section 4.7). However, the author hopes that this work could contribute to further researcher by highlighting the *wide range of perspectives*, which one should take into consideration when making arguments about FOSS business in India (or possibly other developing countries). Even in academic debate on the subject, one sometimes sees strong arguments which seem to fall for overly generalizing. For example, it is argued that FOSS business 'brings less profits' (see section 3.1.1) without mentioning what kind of business model one is talking about nor what are the likely alternatives for an SME in that particular country/region (i.e. less profitable than what?). Or, on the other side of the debate, the 'endogenous' nature of FOSS is often strongly advocated (see section 1.1) without addressing the difficulties that Southern organizations face when trying to launch their own FOSS projects or even contributing to existing ones. This study has brought forth these and some other perspectives which are not always well attended. Yet, as pointed out previously (see section 6.2), there are still a large number of issues which have not been addressed. From here, the next step could be to conduct multiple case studies or a range of qualitative interviews in several companies. Based on these, one could develop a framework which captures the opportunities and challenges that FOSS creates for different types of software SMEs in India.

The study also points to some specific areas that could be interesting to research in much more depth. For example, the strong emphasis that the interviewees placed on the marketing benefits of FOSS participation was surprising, as this issue clearly takes a 'back seat' in Western business literature. If other Indian SMEs are found to have similar views, then one

could, very hypothetically, see some explanation in the cost of international marketing (e.g. adverts in global magazines) being proportionally higher than the cost of R&D labour (i.e. FOSS participation) for Indian companies. Another interesting subtopic is the proposed ability of FOSS to expand bespoke software markets in the South and thereby improve the market diversity of Indian SMEappeas. This issue is becoming more topical as Southern organizations are reportedly getting more receptive to FOSS solutions. These are just two examples of many subtopics deserving more attention. Overall, FOSS has been studied a great deal from the perspective of developing country end-users, but very little from the perspective of the local software companies. The two perspectives are quite different. As the case study suggests, something that is only an additional expense to end-users, may present extra income, and valuable learning opportunities to local software SMEs.

7. REFERENCES

Ahuja, G.(2000) Collaboration networks, structural holes, and innovation: A longitudinal study, *Administrative science quarterly*, 45(3), pp. 425-455.

Ajila S. and Wu J.(2007) Empirical study of the effects of open source adoption on software development economics, *Journal of Systems and Software*, 80(9), pp. 1517-1529

Allee, V. (2008) Value Network Analysis and Value Conversion of Tangible and Intangible Assets. *Journal of Intellectual Capital* 9(1), pp. 5-24.

Allee, V. (2003) *The Future of Knowledge: Increasing Prosperity Through Value Network*s. Butterworth-Heinemann, New York, 2003.

Arora, A. (2008) The Indian software industry and its prospects, in Bhagwati, J. and Calomiris, C. (eds) *Sustaining India's growth miracle*, New York: Columbia University Press, pp:166-215.

Arora, A. and Gambardella (1994), Evaluating technological information and utilizing it: Scientific knowledge, technological capability, and external linkages. *Journal of Economic Behavior and Organization*. 24(1), pp. 91—114.

Athreye S. (2005) The Indian software industry. In Arora A. and Gambardella (Eds.) *From Underdogs to Tigers: The Rise and Growth of the Software Industry in Brazil, China, India, Ireland, and Israel*, New York: Oxford University Press, pp. 7-4

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

Bryman A. (2008) *Social Research Methods.* Third Edition. New York: Oxford University Press.

Bryman A. (1989) *Research Methods and Organization Studie*s. London: Unwin Hyman.

Câmara, G. and Fonseca F. (2007). Information Policies and Open Source Software in Developing Countries. *Journal of the American Society for Information Science and Technology* 12(4): 255-272.

Chaminade, C. and Vang, J. (2006) Innovation policy for Asian SMEs: exploring cluster differences. Proceedings of the *DRUID Summer Conference 2006 on Knowledge, Innovation and Competitiveness*,Copenhagen, Denmark, June 18-20, 2006

Cherington P. (1998 *Advertising as a Business Force: A Compilation of Experience Records*, Doubleday, for the Associated advertising clubs of America, 1913

Chesbrough, H. (2003) *Open innovation : the new imperative for creating and profiting from technology*. Boston: Harvard University Press.

Chesbrough, H (2006a). Open innovation: A new paradigm for understanding industrial revolution. In H. Chesbrough, W. Vanhaverbeke and J. West (eds.) *Open innovation : researching a new paradigm*, Oxford : Oxford University Press.

Chesbrough, H (2006b). *Open Business Models: How to Thrive in the New Innovation Landscape*. Boston: Harvard University Press.

Cohen, W. and Levinthal, D. (1989) Innovation and learning: the two faces of R & D, *The Economic Journal*, 99(397), pp. 569—596.

Coulon, F. (2005). The use of social network analysis in innovation research: A literature review. In *DRUID Academy Winter 2005 PhD Conference*, Aalborg, Denmark, January 27-29.

Dahlander, L. (2005) Appropriation and appropriability in open source software. *International Journal of Innovation Management*, 9(3). pp. 259-285

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

D'Costa A. (2004). The Indian software industry in the global division of labor. In A. D'Costa and E. Sriharan (eds.) *India in the Global Software Industry: Innovation, Firm Strategies and Development*, New York: Palgrave Macmillan.

D'Costa A. (2002). Export Growth and Path Dependency: Locking Innovations in the Software Industry, *Science, Technology and Society*, 7(1), pp 51-81.

Debroy, B. and Morris, J. (2004). Open to development: *Open-Source software and economic development*. London: International Policy Network.

Dibona, C. (2005). OS and Proprietary Software Development. In C. Dibona, D. Cooper and M. Stone (eds.) *Open Sources 2.0*. Sebastopol: O'Reilly.

Dravis P. (2003). *Open source software. Perspectives for development*. Washington: Global information and communication technologies department, the World Bank.

Dupé L. and Paré G. (2003) Rigor in Information Systems Positivist Case Research: Current Practices, Trends and Recommendations. *MIS Quarterly* 27(4), pp. 597-636.

Dyer, J. and Nobeoka, K. (2002) Creating and managing a high-performance knowledge-sharing network: the Toyota case. *Strategic Management Journal*, 21(3), pp—357.

Easton G. (2010), Critical realism in case study research, *Industrial Marketing Management*, 39(1), 1, pp. 118-128.

Enkel, E. and Gassmann, O. (2008) Driving Open innovation in the front end. Working Paper , University of St. Gallen and Zeppelin University, St. Gallen and Friedrichshafen.

Enkel, E. and Gassmann, O. and Chesbrough, H. (2009) Open R&D and Open innovation: exploring the phenomenon. *R&D Management* 36(4), pp. 311--316

Engelfriet, A. (2009). Choosing an Open Source License, *IEEE Software*, 27(1), pp. 48-49.

Fitzgerald B. (2006). The Transformation of Open Source Software. *MIS Quarterly*, 30(3), pp. 587–598

Fitzgerald B. and Kenny T. (2004), Developing an information systems infrastructure with open source software, *Software, IEEE* **21** (1) (2004), pp. 50–55.

Fogel, K. (2006). Producing open source software: how to run a successful free software project, Sabastopol: O'Reilly.

Forge, S. (2006) The rain forest and the rock garden: the economic impacts of open source software, Journal of Policy, Regulation and Strategy for Telecommunications, Information and Media 8(3) , pp. 12-31.

Free Software Foundation (2010) The Free Software Definition. http://www.gnu.org/philosophy/free-sw.html. Retrieved on December 2010.

Gassmann, O. and Enkel, E. (2004). Towards a theory of Open innovation: three core process archetypes. In the proceedings of the R&D Management Conference, Sesimbra, Portugal, 7-9 July 2004.

Gentleware (2010). Open Source Sponsorship. http://www.gentleware.com/opensource.html Retrieved on 12 August 2010.

German D. (2002). The evolution of the GNOME Project. *Proceedings of the 2nd Workshop on Open Source Software Engineering*, Boston, MA .

Gloor, P. (2006) *Swarm Creativity – Competitive advantages through collaborative innovation networks*. New York: Oxford University Press.

Goldman G. and Gabriel R. (2005). I*nnovation happens elsewhere. Open source as business strategy*. San Fransisco: Elsivier.

Gomes-Casseres (1996), The alliance revolution: the new shape of business rivalry, New York: Harvard Univ Press.

Grand, S., Krogh, G., Leonard, D. and Swap, W. (2005) Resource allocation beyond firm boundaries:: A multi-level model for Open Source innovation, Long Range Planning, 37(6), pp. 591-610

Grix. J. (2004). *The Foundations of Research*. New York: Palgrave Macmillan.

Groff, R. (2004). *Critical realism, Post-positivism and the Possibility of Knowledge.* London: Routledge.

Guba, E. (1981). Criteria for assessing the trustworthiness of naturalistic inquiries *Educational Technology Research and Development* , 29(2), pp. 75-91.

Harison E. and Koski H. (2006) Does Open innovation foster productivity. Evidence from open source software firms. Discussion paper #1135. The Research Institute of Finnish Economy (ETLA).

Hecker, 1999. F. Hecker, Setting up shop: the business of Open-Source software. *IEEE Software* 16(1) (1999), pp. 45–51.

Heeks R. (2004). Software Export Success Factors and Strategies in 'Follower' Nations. Competition and Change 8(3), pp. 267–303

Heeks. R. (2006). Using competitive advantage theory to analyze IT sectors in developing countries: a software industry case analysis. *Information Technologies and International Development*, 6(3), pp. 5-34.

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

Henttonen, K. and Matinlassi, M. (2007) Contributing to Eclipse - a case study. Proceedings of *the Software Engineering 2007 Conference (*SE2007). Hamburg, Germany, 27-30 March 2007.

Hippel, E. (1998). The sources of innovation. New York: Oxford University Press.

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

Hoe, N. (2006) Breaking Barriers: The Potential of Free and Open Source Software for Sustainable Human Development. New Delhi: UNDP and Elsevier.

Huurinen J., Torkkell M., Viskari S. and Salmi P. (2006). *Motives, circumstances and driving forces for Open innovation: using open source to drive profitable businesses*. Research report #174. Lappeenranta University of Technology, The department of industrial management.

IDC (2009) Asia/Pacific (Excluding Japan) Content Management Pulse Check, Adoption Drivers, and Barriers in 2009: What Are Buyers Prioritizing? *IDC Report* # AP322105S, Aug 2009

IDPM (2010). Indian software exports statistics. The Institute of Development and Management, The University of Manchester.
http://www.sed.manchester.ac.uk/idpm/research/is/isi/isiexpt.htm
Retrieved on 12 August 2010.

Illiyan, A.(2005) Performance challenges and opportunities of Indian software export, *Journal of Theoretical and Applied Information Technology*, 4(11), pp. 1088-1106.

ITEA (2004). ITEA Report on Open Source Software, Eidhoven: ITEA Office Association.

Johne, A. (1999) *Successful market innovation. European Journal of Innovation Management,* 2(1), pp. 6–11

Kambhampati, U.S. (2002) The Software Industry and Development: the Case of India, *Progress in Development Studies*, 2(1): 23-45.

Karels, M. (2003) Commercializing Open Source Software, *ACM Queue*, 1(5).

Kenwood C. (2001) *A Business Case Study of Open Source Software* . MITRE. Available at http://www.mitre.org/
support/papers/tech_papers_01/kenwood_software/kenwood_software.pdf

King, N. (1998). Template analysis. G. Symon, and C. Cassell (Eds) In *Qualitative Methods and Analysis in Organizational Research: A Practical Guide*. Sage, Thousand Oaks, CA. pp. 118-134.

Koenig, J. (2004) Seven Open Source Business Strategies for Competitive Advantage, *IT Manager's Journal*, 2004.

Krishnamurthy, S. (2005) An analysis of open source business models. In J. Feller, B. Fitzgerald, S. Hissam and K. Lakhani (eds.) *Perspectives on Free and Open Source Software*, Massachusetts: MIT Press.

Krogh G., Spaeth S. and Lakhani K. (2003) Community, joining, and specialization in open source software innovation: a case study. *Research Policy*, 32 (7), pp. 1217–1241

Kroll J. (2000) Conectiva Linux. Linux Journal, No. 73

Lerner J. and Tirole J (2002). Some Simple Economics of Open Source. *The Journal of Industrial Economics*, 50(2), pp. 197-234

Liebenau J. and Smithsons S. (1993) Editorial: secrecy threatens research. *European Journal of Information Systems* 2(4), p. 239.

Maula M., Keil, T. and Salmankanta J. (2006). Open innovation in systemic innovation contexts. In H. Chesbrough, W. Vanhaverbeke and J. West (eds.) *Open innovation : researching a new paradigm*, Oxford : Oxford University Press

Merilinna, Janne and Matinlassi, Mari. (2006). State of the art and practice of open source component integration. Proceedings of the 3*2th Euromicro Conference on Software Engineering and Advanced Applications* (SEAA) Component-Based Software Engineering Track Cavtat/Dubrovnik (Croatia), August 28-September 1, 2006, pp. 170 - 177.

Messerschmitt, D. (2004) Marketplace issues in software planning and design, *IEEE Software*, 21 (3) (2004), pp. 62–70.

Morgan, L. and Finnegan, P. (2007) Benefits and drawbacks of open source software: an exploratory study of software firms. In J. Feller, B. Fitzgeral, W. Scacchi, and A. Sillitti (eds.) *International Federation for Information Processing: Open Source Development, Adoption and Innovation*, Boston: Springer, pp. 307-312

NASSCOM 2008. The IT Industry in India: Strategic Review. New Delhi: NASSCOM

Nirjar, A. and Tylecote, A. (2005). Breaking out of lock-in: Insights from case studies into ways up the value ladder for Indian software SMEs. *Information Resources Management Journal*, 18(4), pp. 40-61.

Norris J., (2004) Mission-critical development with open source software: lessons learned, *IEEE Software* **21** (1) (2004), pp. 42–49.

O'Donnell C. (2004). A case for Indian outsourcing: open source interests in IT jobs. *First Monday* 9(11).

Open Source Initiative (2010). The open source definition. http://www.opensource.org/docs/osd. Retrieved on December 2010.

Parthasarathi A. and Joseph K. (2004) Innovation and under export orientation. In A. D'Costa and E. Sriharan (eds.) *India in the Global Software Industry: Innovation, Firm Strategies and Development*,New York: Palgrave Macmillan.

Persson A., Gustavsson H, Lings B, Lundell B., Mattsson A. and Ärlig U. (2005) OSS tools in a heterogeneous environment for embedded systems modelling. In Proceedings of the *5th Workshop on Open Source Software Engineering*, held in conjunction with the 27th International Conference on Software Engineering (ICSE 2005), May 2005.

Pol, E. and Carroll, P. (2006). *An Introduction to Economics with Emphasis on Innovation*, Melbourne: Innovation Planet.

Puhakka, M., Jungman H. and Seppänen M. (2007). Investing in open source companies. Deal making from the venture capitalist viewpoint. In K. Amant and B. Still (eds) *Handbook*

of Research on Open Source Software: Technological, Economic, and Social Perspectives. London: Information science reference.

Ragin C. (1994). *Constructing Social Research. The Unity and Diversity of Method.* Thousand Oaks: Pine Forge Press.

Rajala R., Nissilä, J. Westernlund, M. (2007) Revenue models in open source software business. In K. St.Amant and B. Still (eds). Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives. London: Information science reference.

Raymond, E. 2001. *The Cathedral and the Bazaar*. Sepastobol: O'Reilly

Reinhardt C. and Hemetsberge A. (2007) Of experts and apprentices: Learning from the KDE community. In M. Lytras and A. Naeve (eds) *Open source for knowledge and learning management: Strategies beyond tools*. London: Idea Group Publishing.

Rejswoud, V. and Mulo, E (2005). Applying Open Source Software in a Development Context: Expectations and Experiences. Case study of a University in Uganda. International Institute Communication and Development (IICD).

Robles, G. and Gonzalez-Barahona, J.M.(2006) Geographic location of developers at SourceForge In the proceedings of *the 2006 International Workshop on Mining Software Repositories*, Shanghai, China, pp. 144-150

Rowley J. (2002) Using case studies in research, *Management Research News*, 25(1), pp.16 – 27

Sayer, A. (1992). *Method in social science: A realist approach*. 2nd Edition. London: Routledge

Schmidt, K. and Schnitzer, M. (2003). Public subsidies for open source? Some economic policy issues of the software market. *Harvard Journal in Law and Technology*. 16, pp 474–505.

Schumpeter (1939). *Business Cycles*. New York: McGrew-Hill.

Sharma A. and Adkins R. (2005) OSS in India. In Dibona C., Cooper D and Stone M. (eds) *Open sources: the Continuing Evolution*. Sebastobol: O'Reilly.

Simons H. (2009) *Case Study Research in Practise*. London: Sage

Spinellis, D. (2003) *Code reading: the open source perspective*. London: Addison Wesley Publishing Company.

Stark, S. and Torrance H. (2005). Case study. In B. Somekh and C. Levin (eds), *Research Methods in the Social Sciences*. London: Sage Publications.

Subramaniam M. and Youndt S. (2005) The influence of intellectual capital on the type of

innovative capabilities. *Academy of Management Journal* , 48(3), pp. 450–463.

Tapia, A. and Maldonado E. (2009) An ICT Skills Cascade: Government-Mandated Open Source Policy as a Potential Driver for ICT Skills Transfer. Information Technologies and International Development, 5(2), pp. 31–51

Tuomi, I. (2002) *Networks of Innovation*, New York: Oxford University Press.

Ulijn, J. and Brown, T. (2004). Innovation, entrepreneurship and culture, a matter of interaction between technology, progress and economic growth? In Brown T. and Ulinj J. (eds). *Innovation, entrepreneurship and culture.* Massachusetts: Edward Elgar Publishing.

UNU-MERIT 2006. Open source as open standards: A new frontier for economic development?. Policy Brief 2006(1), Maastricht: United Nations University.

UNU-MERIT (2007a). Free/Libre and Open Source Software: Worldwide Impact Study . D31: Track 1 International Report. Skills Study. Maastricht : United Nations University.

UNU-MERIT (2007b). Free/Libre and Open Source Software: Worldwide Impact Study . D31: Track 1 International Report. Software Study. Maastricht : United Nations University.

UNU-MERIT (2007c). Free/Libre and Open Source Software: Worldwide Impact Study D7: Track 1 Survey Report - India. Skils Study. Maastricht : United Nations University.

UNU-MERIT (2007d). Free/Libre and Open Source Software: Worldwide Impact Study D7: Track 2 Survey Report - India. Software Study. Maastricht : United Nations University.

Vaden T. and Vainio N. (2005). Free and Open Source Software Strategies for Sustainable Information Society. In O. Hietanen (ed.) *University partnerships for international development: Finnish development knowledge*, Turku: Finland Futures Research Centre.

Vanhaverbeke W. (2006). Interorganizational context of Open innovation. In H. Chesbrough, W. Vanhaverbeke and J. West (eds.) Open innovation : researching a new paradigm, Oxford : Oxford University Press

Vanhaverbeke W. and Cloth M. (2006). Open innovation in Value networks. In H. Chesbrough, W. Vanhaverbeke and J. West (eds.) *Open innovation : researching a new paradigm*, Oxford : Oxford University Press

Vankatesh, A. (2000) Computers and New Media Technologies in Indian Households: Based on a Study of Eight Major Cities in India , NOAH Research Report, Center for Research on Information Technology and Organizations (CRITO) , The University of California.

Vrande et al., (2009). Open innovation in SME's: trends, motives and management challenges, *Technovation* **29** (2009), pp. 423–437.

Weerawarana, S. and Weeratunge J. (2004). Open Source in Developing Countries. Stockholm: SIDA.

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

West J. (2006). Does appropriability enable or retard Open innovation. In H. Chesbrough, W. Vanhaverbeke and J. West (eds.) *Open innovation : researching a new paradigm*, Oxford : Oxford University Press.

West J. and Gallagher S. (2006a). Challenges of Open innovation: the paradox of firm investment in open source software. *R&D Management* 36(2), pp. 316-331.

West J. and Gallagher S. (2006b). Patterns of Open innovation in open source software development. In H. Chesbrough, W. Vanhaverbeke and J. West (eds.) *Open innovation : researching a new paradigm*, Oxford : Oxford University Press

Wichmann T. and others (2002) Free/Libre Open Source Software: Survey and Study . Part 2A:Firms‘ Open Source Activities: Motivations and Policy Implications . Berlin: Berlecon Research.

Wong K. (2004). Free/open source software : government policy; New Delhi: UNDP Asia Pacific Development Information Program in co-operation with Elsevier.

World Bank. (2004). SME. World Bank Group Review of Small Business Activities. Washington, DC: World Bank.

Ye, Y. and Kishida, K. (2003) Toward an understanding of the motivation of open source software developers. The proceedings of the *25th International Conference on Software Engineering* (ICSE'03).

Zahra S. and George G. Absorptive capacity: A review, reconceptualization, and extension, *Academy of management review*, 27(2), pp. 185-203.

APPENDIX A: FOSS BUSINESS MODELS AND LICENSES

This appendix clarifies the relationship between Open innovation processes (see section 2.2 in the main text), FOSS business models and FOSS licenses. The aim is to assists especially those readers who are not previously familiar with FOSS business literature. The first section of the literature review (see section 3.1 in the main text) proposes a strong connection between Open innovation processes archetypes and FOSS business models. **Table 4** (on the next page) presents this mapping more explicitly by listing the revenue options associated with each process archetype and linking them to the FOSS business models introduced in 'classic' articles by Koenig (2004), Hecker(1999), Krishnamurthy (2005) and Karels(2003). Following the scope of this study (see section 1.3), only software-intensive business models are presented. Any of the these may include technical consulting services as an additional revenue source. Hecker's (1999) 'sell-it-free-it' model (see section 3.1.2 in the main text) is not listed separately because it just refers to replacing a proprietary business model with one of the FOSS business models, e.g. with the loose-leader model.

Based on the aforementioned authors, the table also shows how FOSS licensing terms relate to particular revenue models. It is important to understand that, against a common misconception, none of the FOSS licenses place restrictions on how software can be used, commercially or non-commercially. Any terms and conditions only apply to its redistribution. Some licenses, mostly famously GPL, say that the source code of any derived or interlinked work must be given to anybody who receives the binary version of the software, but they do *not* ask to share the source code with third-parties. Consequently, none of the many FOSS licenses have an impact on the application service provision (ASP) or other models where software is not distributed, but hosted instead. They also have limited (though possibly still significant) impacts on business models, where the software is only distributed to a small number of parties, such as bespoke software development. The effects are most 'dramatic' on packaged software production, where GPL licensing terms cause the software to be shared widely and, thereby, its price drops quickly close to zero. For business models which rely on open sourcing internally developed software (i.e. the inside-out process), GPL is often the preferred choice of license. This is because, by enforcing reciprocity, its terms make it more difficult for competitors to 'hijack' the FOSS product and

use it for their own ends.

Table 4: Mapping Open Innovation processes to FOSS business models literature (process archetypes from Gasman and Enkel 2004, see section 2.2 in the main text)

Process arthetype	**The purpose of FOSS involvement**	**Offering (Revenue source)**	**Possible licensing scheme**	**Business models from literature**[11]
Outside-in process	To sell an improved or extended version of a FOSS product	Packaged software (License fees)	Any, except GPL; BSD and alike preferred	Non-GPL software producer (Krishnamurthy 2005) Commercial enhancement or value-added (Karels 2003)
		Bespoke software (Payments by contract)	Any, but non-GPL sometimes preferred	Optimization (Koenig 2004) Contract development (Karels 2003) GPL-mode software producer (Krishnamurthy 2005)
		Application service provision (ASP fees)	Any	Hosted (Koenig 2004) Service Enabler (Hecker 1999)
Inside-out process	To use FOSS as promotional tactics	Any, but often packaged software (License fees)	Any, but GPL often preferable	Loose leader (Hecker 1999) Dual licensing (Koenig 2004) Poison Pill (Karels 2003)
	To advance diverse strategic interests through FOSS	Any (No direct revenue targeted)	Any, depends on goals	Patronage (Koenig 2004) Standards development or market building (Goldman and Gabriel 2005)
Coupled process	Any of the above			

11 These sources have also been cited in the main text. For bibliographic information, please see the main reference list of the dissertation (starts from page 61)

APPENDIX B: INTERVIEWEE SUMMARY

Table 5 shows the summary of people who were interviewed for this study. The directors from both Mahiti Infotech and a "co-opetiting" company, Sakshey Web Technologies, agreed to be identified with their own name. The names of other respondents have been changed. The two female developers mentioned in the table were interviewed individually, but the researcher also interacted with some other employees in an informal group setting. The study also involved watching videoed presentations by some people who are not listed herein. For information on participant selection and interview topic guide, see section 4.3 in the main text.

Table 5: The summary of the interviewees

Name/Alias	Organization	Position	Responsibilities in the organization
Sreekanth	Mahiti Infotech	Chief Executive Officer	Strategic Planning Growth/Change management Learning and Innovation management
Chethan	Mahiti Infotech	Marketing Director	Analyzing client's business requirements Marketing and PR communication
Vijay	Mahiti Infotech	Technical Director	Management of development projects Training of new employees Consulting clients on technology choices
Hasita	Mahiti Infotech	Senior Developer	Software Architecture Design Support for junior developers Communication with FOSS communities
Girija	Mahiti Infotech	Senior Developer	Software design and development Communication with FOSS communities
Ravi	Sakshay Web Technologies	Chief Executive Officer	Strategic Planning Management of R&D projects
Jagreet	Local NGO for social development	Director and Research fellow	Management of education related development and advocacy programmes Background research and evaluation